Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Інструментальні засоби розробки програмного забезпечення

Лабораторна робота

Виконала студентка 2-го курсу

Групи ІПС-22

Філоненко Дарина Дмитрівна

Київ - 2025

**Тема:** Документація коду (Doxygen/JavaDoc/…) і автоматична публікація через GitHub Actions та GitHub Pages

**Мета:** Набуття навичок у

1. документуванні коду Python за допомогою стандартних коментарів формату Doxygen/JavaDoc;
2. локальній генерації HTML-документації з коду;
3. налаштуванні безперервної інтеграції (CI/CD) за допомогою GitHub Actions для автоматичної збірки документації;
4. автоматичній публікації згенерованої HTML-документації на статичний хостинг GitHub Pages.

**Теорія:** *Doxygen —* це потужний інструмент для генерації документації безпосередньо з вихідного коду. Він парсить спеціально відформатовані коментарі (з тегами @brief, @param, @return тощо) і автоматично будує HTML-сайт, PDF (через LaTeX) або інші формати, що містять опис класів, методів та їхніх взаємозв'язків. Для візуалізації ієрархії класів та діаграм Doxygen інтегрується з інструментом Graphviz.

*GitHub Actions —* це вбудована в GitHub платформа CI/CD, що дозволяє автоматизувати робочі процеси розробки. Вона працює на основі .yml файлів, розміщених у папці .github/workflows. Ці файли описують "події" (наприклад, push у гілку) та "завдання" (jobs), які потрібно виконати, наприклад: встановити Doxygen, запустити генерацію та завантажити артефакти.

*GitHub Pages —* це сервіс статичного хостингу від GitHub. Він дозволяє опублікувати веб-сайт (наприклад, згенеровану Doxygen HTML-документацію) безпосередньо з репозиторію. У поєднанні з GitHub Actions, це дозволяє створити повністю автоматизований процес, де кожна зміна в коді автоматично оновлює живу документацію в Інтернеті*.*

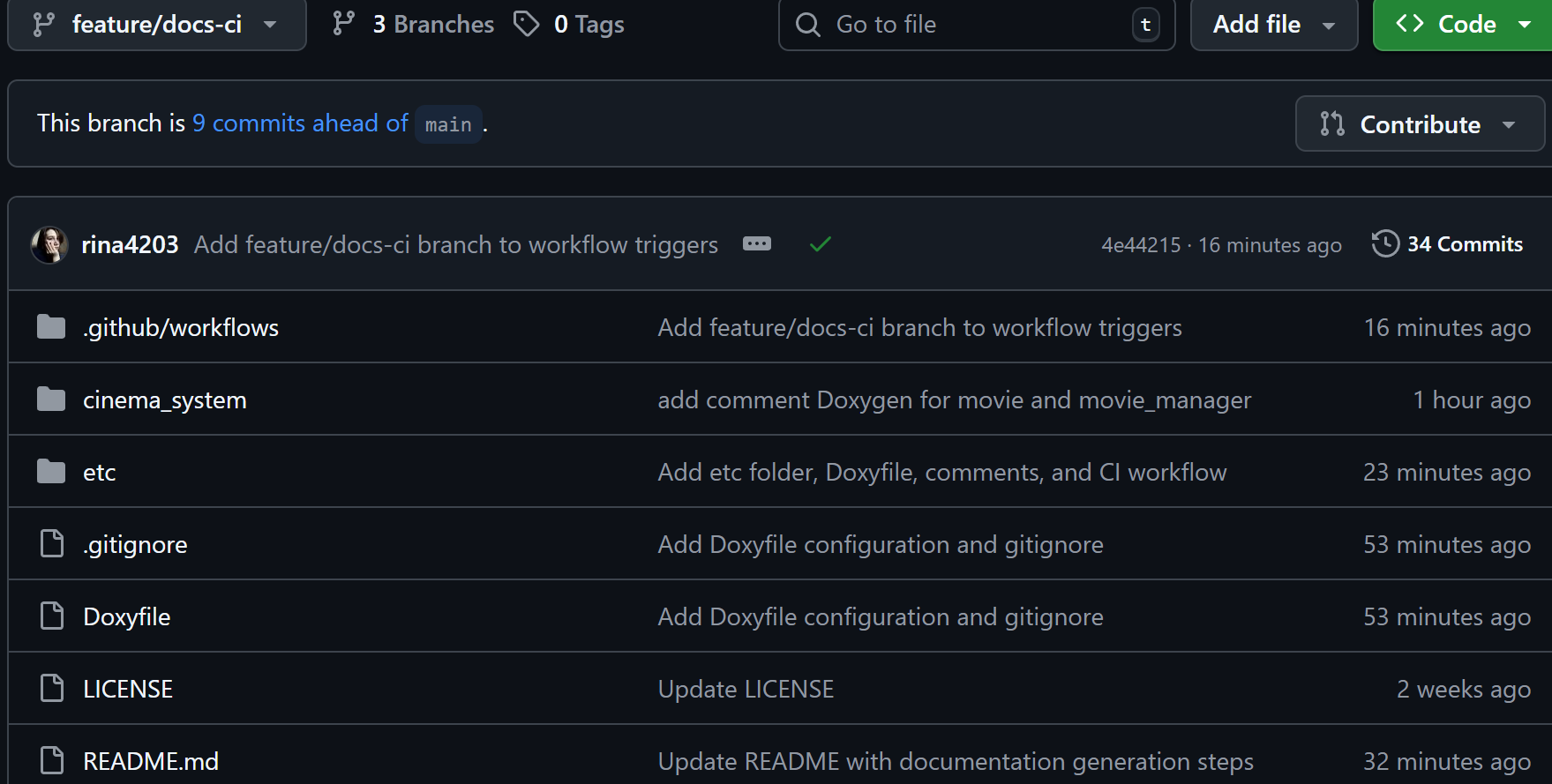
[Посилання на репозиторій GitHub](https://github.com/rina4203/unit-testing-laba.git)

[Посилання на сторінку GitHub Pages із документацією](https://rina4203.github.io/unit-testing-laba/)

**Скріншоти основних етапів роботи**

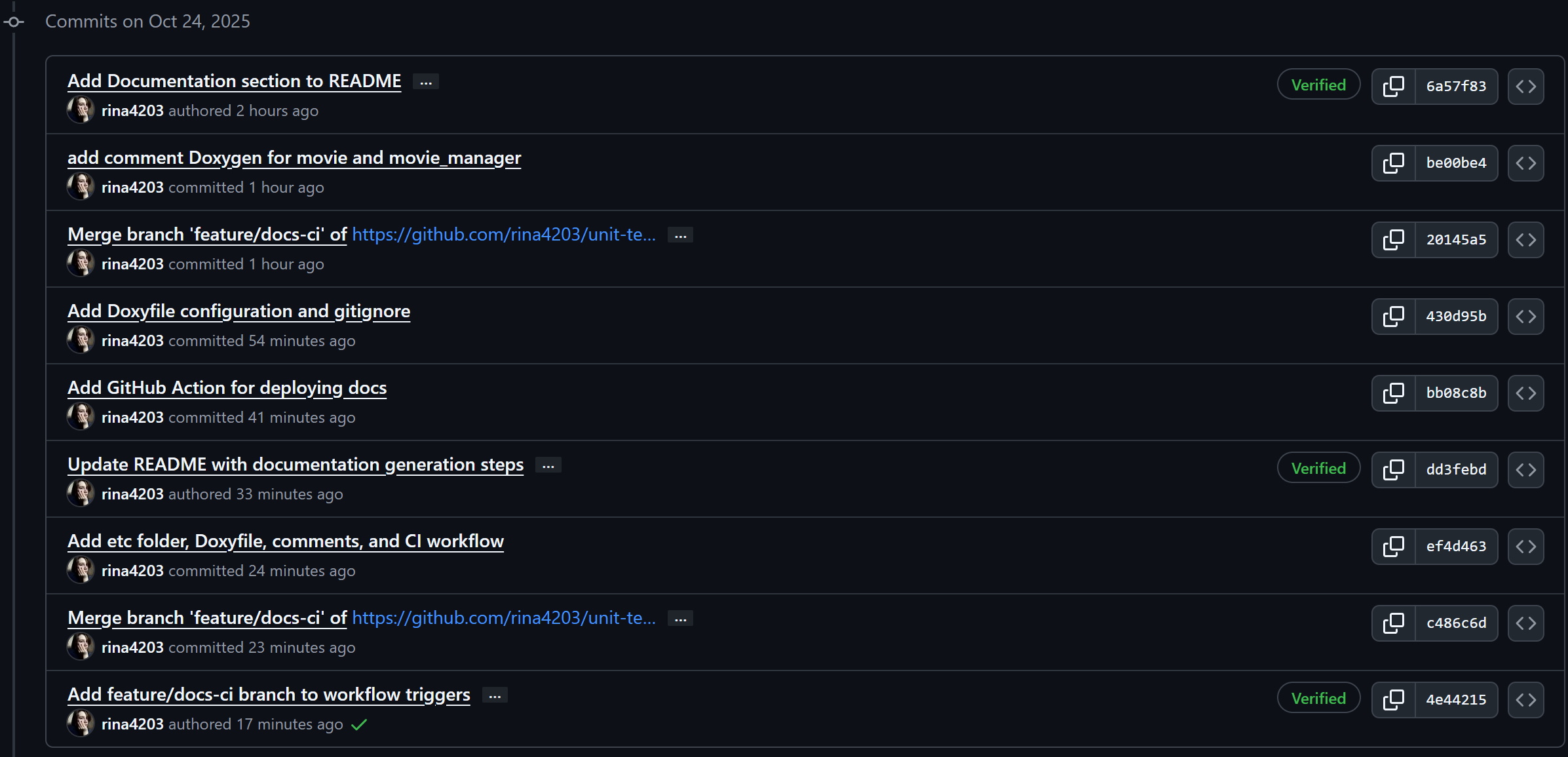
1. Створення нової гілки feature/docs-ci

Робота велася в окремій гілці, щоб ізолювати її від основної гілки main.



2. Серія логічних комітів

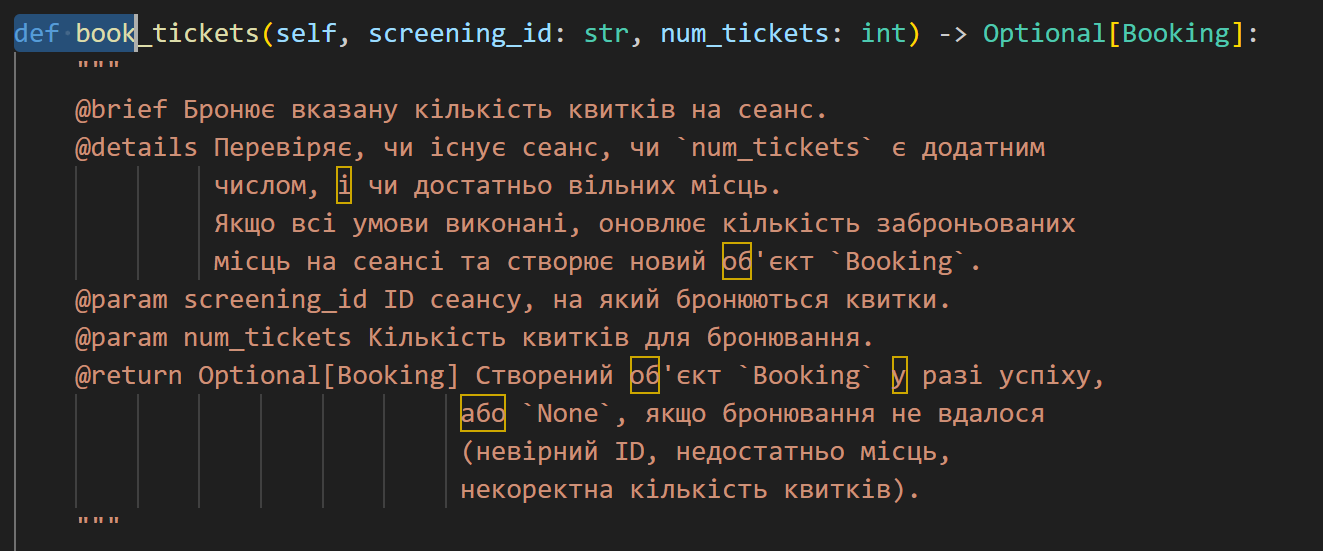
Усі зміни зберігалися через невеликі, логічні коміти з описовими повідомленнями, що відображають прогрес роботи.



**1. Додавання Doxygen-коментарів до коду**

До основних класів (CinemaManager, Movie, Screening, Booking) та їхніх методів було додано коментарі у стилі Doxygen. Це включало опис класів, призначення методів та параметри, які вони приймають.

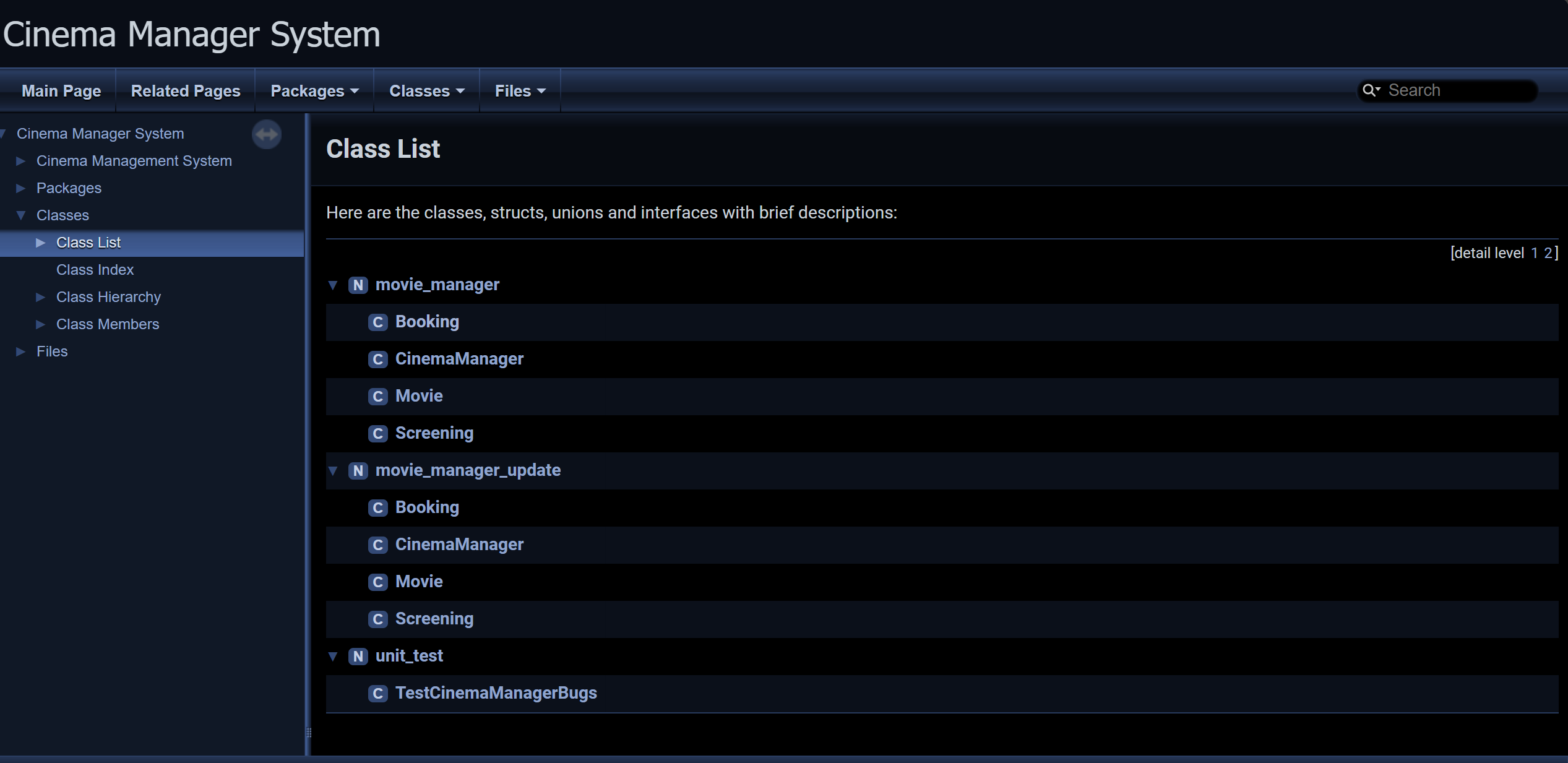
Приклад коментаря до методу:



**2. Генерація документації локально (Doxyfile)**

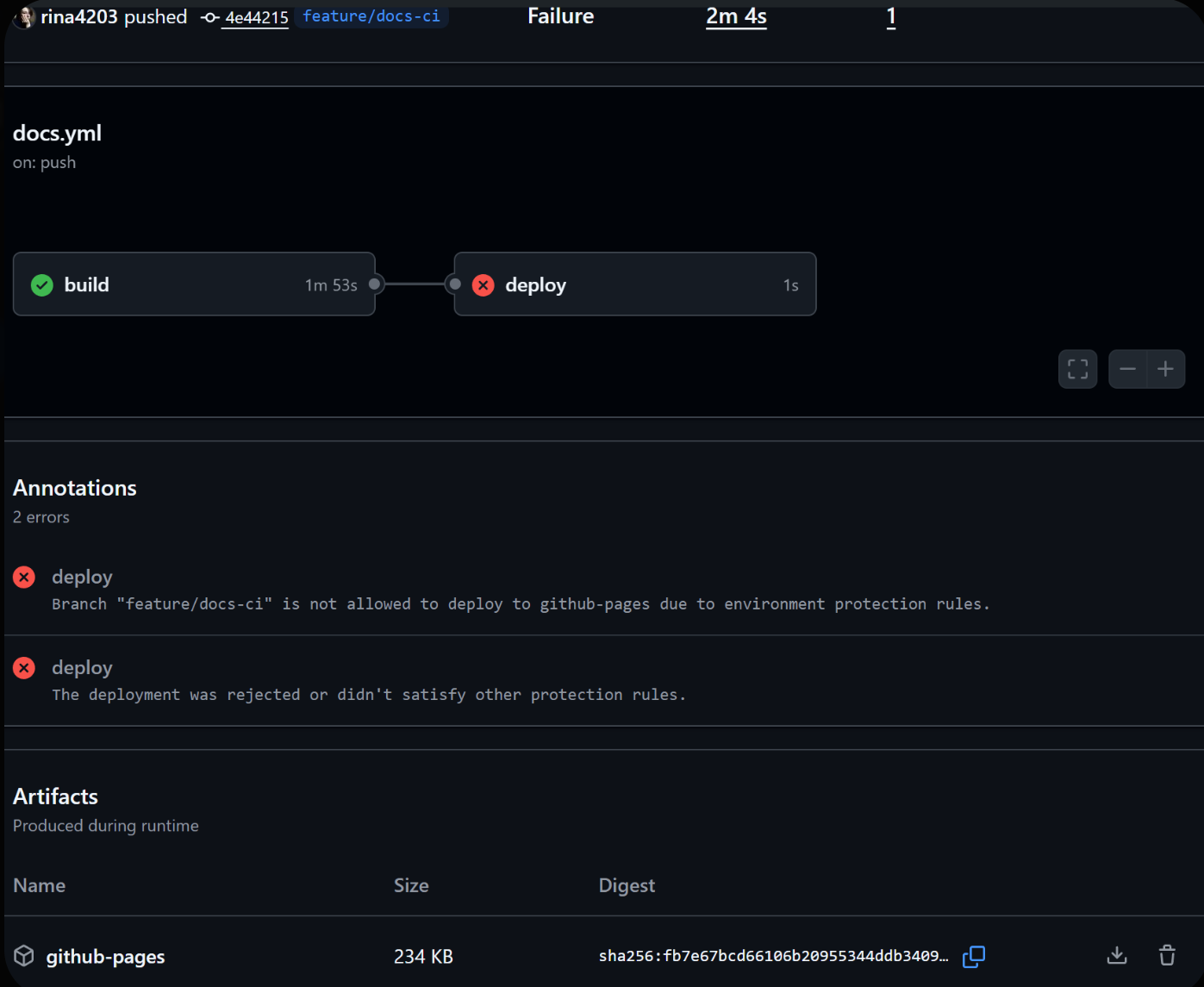
За допомогою команди doxygen -g Doxyfile було створено конфігураційний файл. Ключові налаштування у Doxyfile були змінені для коректної роботи з Python та Graphviz:

* PROJECT\_NAME = "Cinema Manager System"
* OPTIMIZE\_OUTPUT\_JAVA = YES (це налаштування також оптимізує вивід для Python)
* HAVE\_DOT = YES (для увімкнення діаграм)
* RECURSIVE = YES (для пошуку файлів у підпапках)
* EXCLUDE = \*/tests/\* \*/\_\_pycache\_\_/\* (щоб ігнорувати тестові файли)

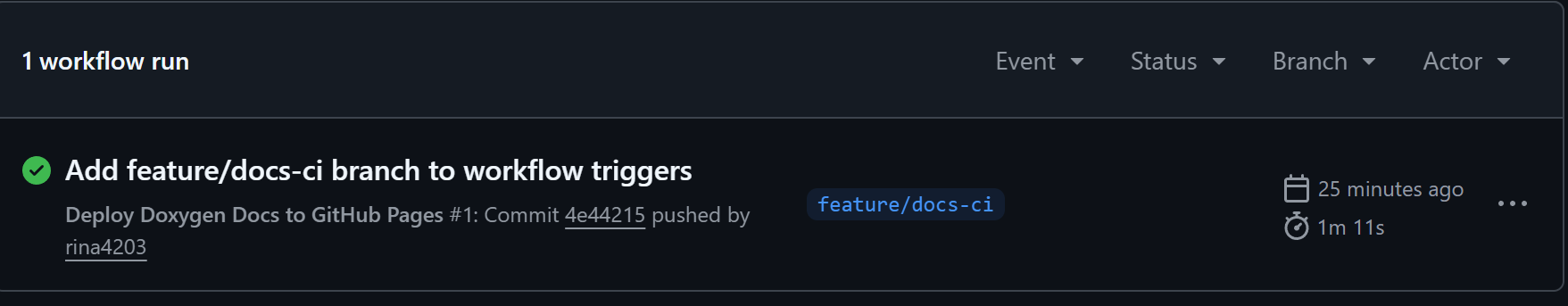


**3. Налаштування GitHub Pages та вирішення проблем**

Під час тестування у гілці feature/docs-ci виникла помилка доступу.

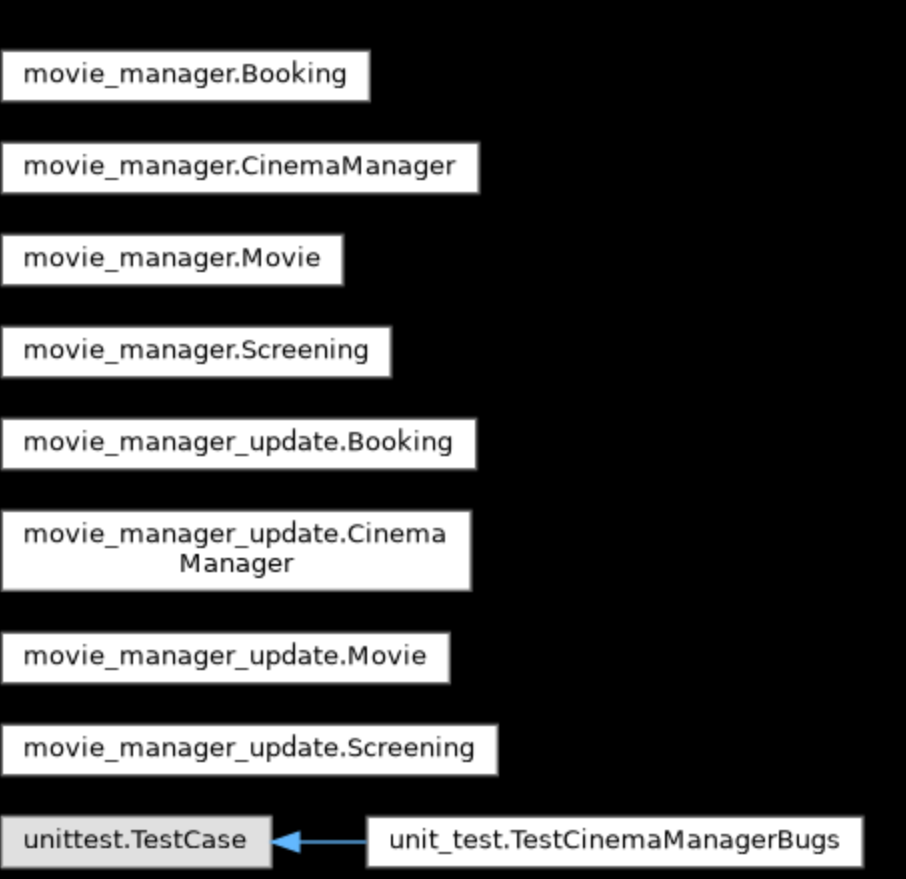


**Рішення:** Помилка була виправлена шляхом додавання правила у Settings > Environments > github-pages



**5. Результат: Опублікований сайт**

Після успішного виконання Action, документація стала доступна онлайн.



**Висновки про набуті навички**

Під час виконання цієї лабораторної роботи я набула таких практичних навичок:

1. **Документування коду:** Освоїла синтаксис Doxygen/JavaDoc (@brief, @param, @return та ін.) для створення змістовних коментарів, які можуть бути розпарсені автоматично.
2. **Використання Doxygen:** Навчилася генерувати HTML-документацію локально, налаштовувати Doxyfile для роботи з Python та вмикати генерацію діаграм класів за допомогою Graphviz.
3. **Основи CI/CD:** Зрозуміла фундаментальні принципи CI/CD — автоматизацію рутинних завдань (як-от генерація документації) при зміні коду.
4. **Робота з GitHub Actions:** Набула практичного досвіду у створенні .yml файлів воркфлоу, налаштуванні кроків (steps), встановленні залежностей (apt-get install) у середовищі Linux та передачі результатів (артефактів) між завданнями (jobs).
5. **Налаштування GitHub Pages:** Навчилася налаштовувати репозиторій для публікації статичних сайтів та пов'язувати його з GitHub Actions.
6. **Вирішення проблем (Troubleshooting):** Отримала цінний досвід у діагностиці та виправленні помилок у CI/CD пайплайнах, зокрема, проблем зі шляхами до виконуваних файлів (локальний DOT\_PATH) та помилок доступу, пов'язаних з правилами середовища (Environment protection rules).