



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 №2
Технології розроблення програмного забезпечення
ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ.
СЦЕНАРІЙ ВАРІАНТІВ ВИКОРИСТАННЯ.
ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ.
КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ

Project Management software

Виконав
студент групи ІА–22:
Неїжсал Артем

Київ 2024

Обрана тема:	3
Короткі теоретичні відомості	3
Виконання роботи	5
Діаграма прецедентів	5
Опис 3 прецедентів	5
Структура бази даних	8
Діаграма реалізовних класів	9

Тема: Діаграма варіантів використання. Сценарії варіантів використання. Діаграми UML. Діаграми класів. Концептуальна модель системи

Мета: Проаналізувати тему, намалювати схему прецеденту, діаграму класів, розробити основні класи і структуру бази

Обрана тема:

..23 Project Management software (proxy, chain of responsibility, abstract factory, bridge, flyweight, client-server)

Програмне забезпечення для управління проектами повинно мати наступні функції: супровід завдань/вимог/проектів, списків команд, поточних завдань, планування за методологіями agile/kanban/gup (включаючи дошку завдань, ітерації тощо), мати можливість прикріплювати вкладені файли до завдань та посилатися на конкретні версії програми, зберігати виконувані файли для кожної версії.

Короткі теоретичні відомості

У даній лабораторній роботі розглядається використання діаграм UML для моделювання різних аспектів системи. Ключовими елементами є діаграми варіантів використання (Use Case Diagrams), які допомагають визначити основні сценарії взаємодії користувачів із системою.

1. UML (Unified Modeling Language) – це універсальна мова візуального моделювання, призначена для опису структури і поведінки програмних систем. Вона дозволяє створювати діаграми для різних аспектів системи, забезпечуючи чітке та наочне уявлення про її роботу. UML використовується на всіх етапах розробки: від аналізу вимог до тестування.

2. Діаграми варіантів використання (Use Case Diagrams) – це інструмент для моделювання функціональності системи з точки зору користувача. Вони відображають, як зовнішні актори (користувачі,

інші системи) взаємодіють із системою для досягнення своїх цілей. Основні елементи діаграми: актори, варіанти використання (use cases) і зв'язки між ними.

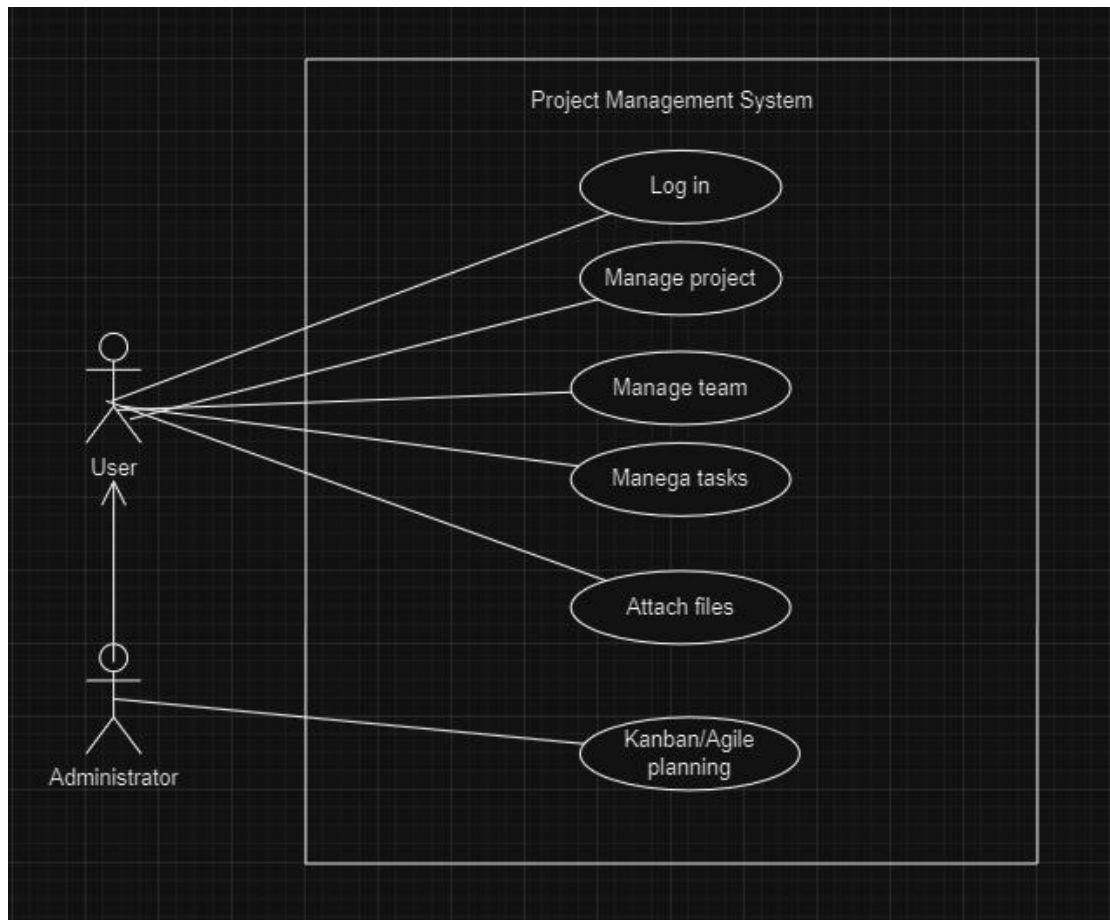
3. Діаграма класів (Class Diagrams) – це інший важливий інструмент UML, що моделює структуру системи на рівні класів і їх взаємодій. Вона відображає класи, їх атрибути, методи, а також зв'язки між ними (асоціації, агрегації, композиції). Ця діаграма є основою для реалізації об'єктно-орієнтованого програмування, оскільки дозволяє чітко визначити структуру об'єктів у системі.

4. Концептуальна модель системи – це початковий етап побудови моделі, який допомагає визначити ключові компоненти системи, їх взаємодії та ролі. Вона може містити як діаграми варіантів використання, так і діаграми класів, забезпечуючи базу для подальшого аналізу та реалізації.

5. Репозиторій – це шаблон проєктування, який забезпечує абстракцію над операціями з базою даних. Він дозволяє взаємодіяти з базою даних через інтерфейси, що спрощує обслуговування та тестування коду. Репозиторій приховує деталі доступу до даних, що робить код більш організованим і підтримуваним

Виконання роботи

Діаграма прецедентів



Опис 3 прецедентів

а) Прецедент: Створення завдання

Актори: Користувач

Опис: Користувач створює нове завдання в рамках проекту, яке потім можна відстежувати і виконувати.

Основний хід подій:

- i. Користувач обирає проект, в якому хоче створити завдання.
- ii. Користувач натискає кнопку «Створити завдання».

- iii. Вказує назву, опис та інші деталі завдання (наприклад, терміни виконання, пріоритет).
- iv. Користувач призначає завдання на відповідного користувача або команду.
- v. Система валідує введені дані та зберігає завдання в базі даних.
- vi. Нове завдання додається до списку завдань для даного проекту.

Виключення: Якщо якісь дані введено не вірно або порожні обов'язкові поля, система видасть сповіщення про помилку.

b) Прикріплення файлу до завдання

Актори: Користувач

Опис: Користувач прикріплює файл до існуючого завдання для надання додаткових матеріалів або документів, що стосуються завдання.

Основний хід подій:

- i. Користувач відкриває потрібне завдання.
- ii. Користувач натискає кнопку «Прикріпити файл».
- iii. Обирає файл зі свого комп'ютера для завантаження.
- iv. Користувач натискає «Завантажити».
- v. Система перевіряє коректність файлу (наприклад, тип і розмір).
- vi. Файл прикріплюється до завдання і зберігається на сервері.

Виключення: Якщо формат, розмір файлу невідповідає вимогам, система сповістить про проблему. Якщо виникла помилка завантаження, система запропонує спробувати ще раз.

c) Прецедент: Управління командою (Team Management)

Актори: Користувач (менеджер або адміністратор)

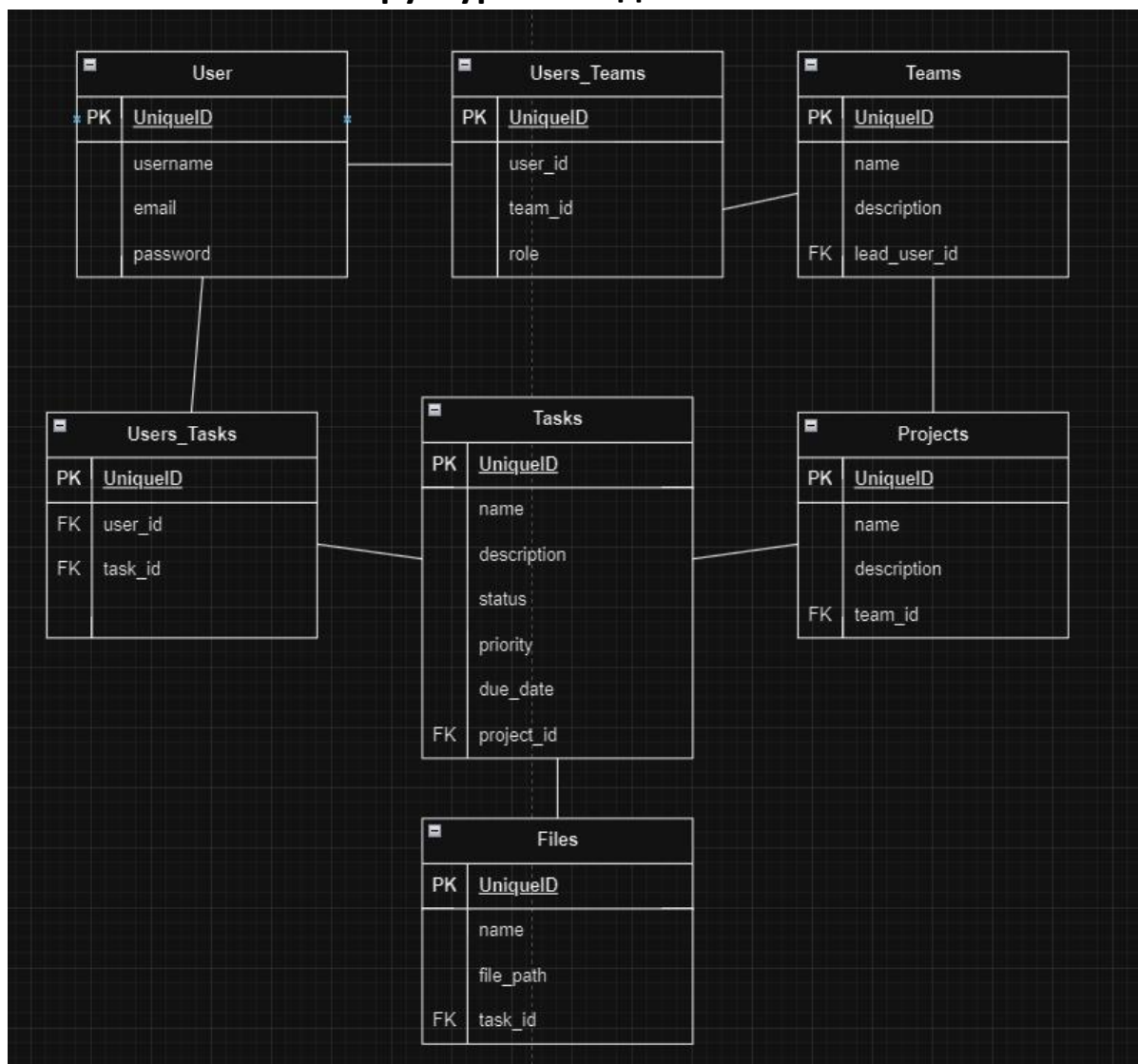
Опис: Користувач управляє командою, додаючи або видаляючи членів команди, а також призначаючи ролі.

Основний хід подій:

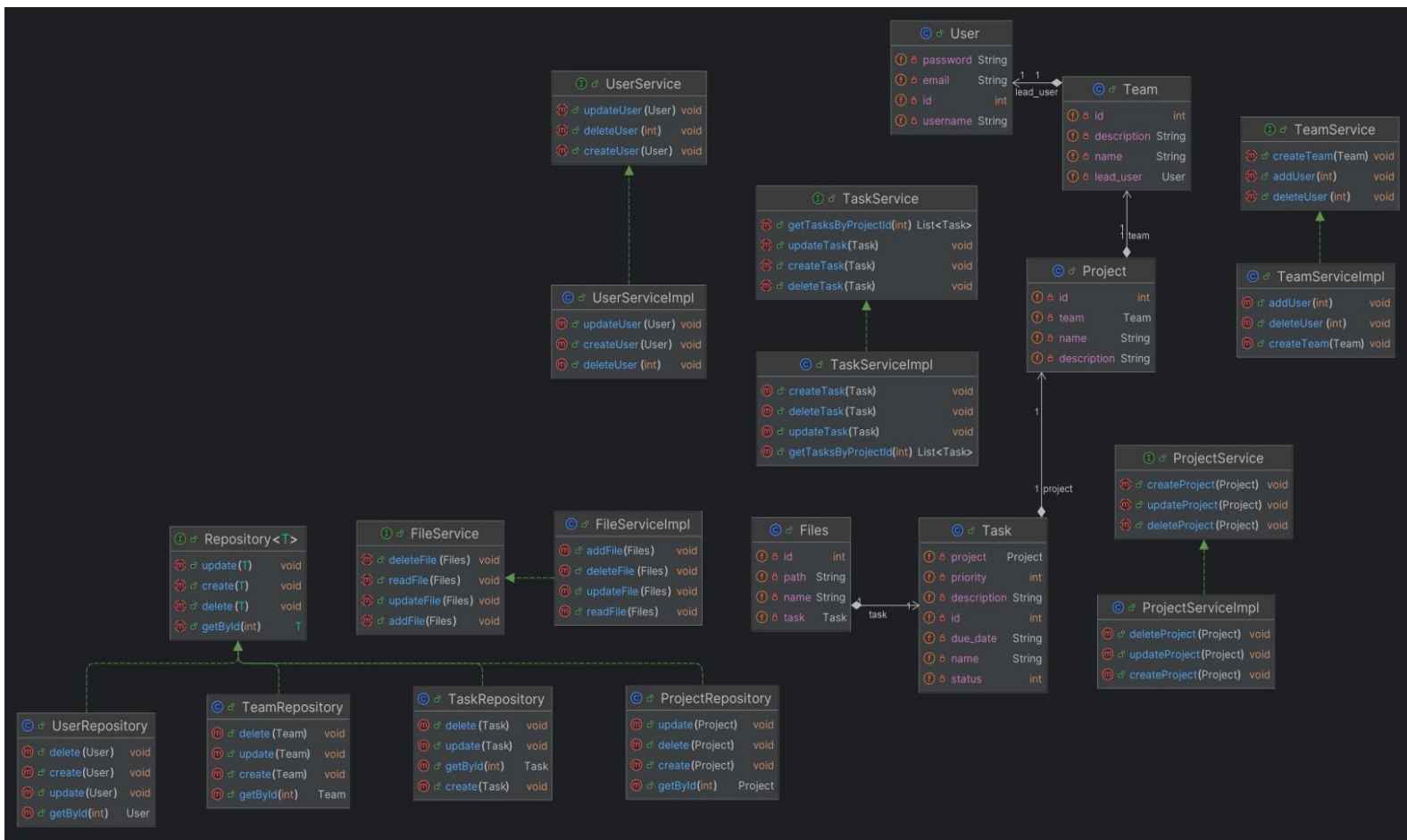
- i. Користувач відкриває розділ управління командами.
- ii. Користувач обирає команду для редагування.
- iii. Користувач натискає кнопку «Додати користувача» або «Видалити користувача».
- iv. Вибирає користувача зі списку для додавання або видалення.
- v. Користувач може змінити роль користувача (наприклад, «лідер команди», «учасник»).
- vi. Система валідує зміни та оновлює інформацію про команду в базі даних.

Виключення: Система повідомляє, що користувач уже є учасником команди, і не дозволяє додати його повторно. Система повідомляє, що в системі немає обраних користувачів.

Структура бази даних



Діаграма реалізовних класів



Посилання на GitHub: <https://github.com/BeP1/trpz>

Висновок: Проаналізував тему, намалював схему прецеденту, діаграму класів і структуру бази