

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. СЦЕНАРІЇ

ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ

КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ

Особиста Бухгалтерія

­­

Виконала:

Студент групи ІА-22

Вдовиченко А.Х.   
  
Перевірив:  
Мягкий М. Ю

Київ 2024

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

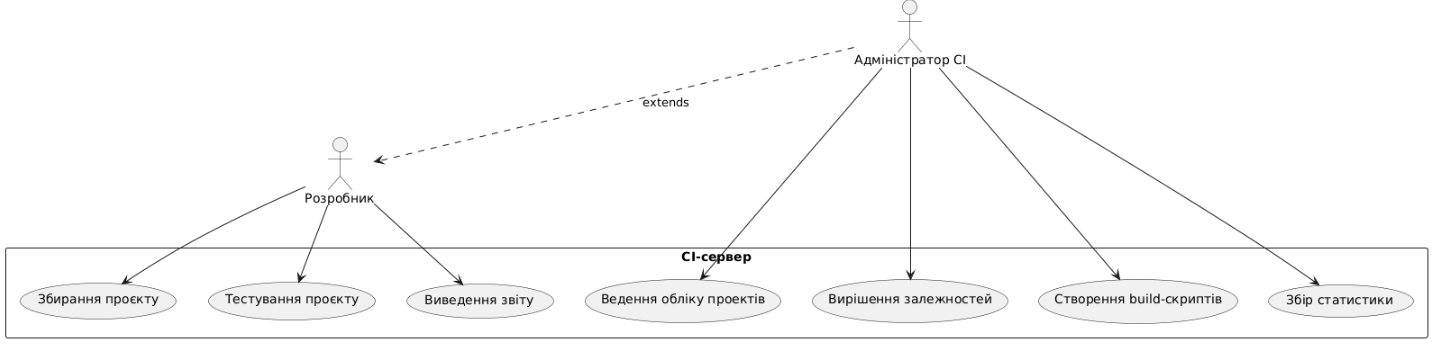
**Мета**: Проаналізувати тему, намалювати схему прецеденту, діаграму класів,

розробити основні класи і структуру бази

**Хід роботи**

1. **Схема прецедентів**

Схема прецедентів, що відповідає темі «СІ-сервер».



### На основі діаграми опишемо 3 сценарії використання:

**Сценарій 1: Збирання проєкту**

**Передумови**: Проект знаходиться у системі контролю версій, і до нього були внесені зміни. **Постумови**: Проект успішно зібраний, або створено повідомлення про помилку. **Взаємодіючі сторони**: Розробник, CI-сервер.  
**Короткий опис**: Даний сценарій описує процес автоматичного запуску збірки проєкту після внесення змін до репозиторію.

**Основний потік подій**:

1. Розробник вносить зміни до репозиторію та робить push змін.
2. CI-сервер автоматично виявляє зміни в репозиторії.
3. CI-сервер ініціює процес збирання проєкту.
4. CI-сервер завантажує залежності та починає компіляцію проєкту.
5. Після завершення збірки CI-сервер відправляє повідомлення про успішну або невдалу збірку.

**Виключення**: Невдала збірка: Якщо в процесі компіляції виникає помилка, CI-сервер зупиняє збірку і надсилає повідомлення про помилку розробнику.

**Примітки**: Цей сценарій може бути ініційований вручну або автоматично, залежно від налаштувань CI-сервера.

**Сценарій 2: Створення нового проекту на CI-сервері**

**Передумови**: Адміністратор CI має доступ до CI-сервера, і репозиторій з вихідним кодом проєкту доступний для інтеграції.  
**Постумови**: Проект успішно створений на CI-сервері, налаштовані параметри збірки, і сервер готовий автоматично збирати проект.  **Взаємодіючі сторони**: Адміністратор CI, CI-сервер.  
**Кроткий опис**: Сценарій описує процес налаштування нового проекту на CI-сервері для його автоматичної збірки, тестування і моніторингу.

**Основний потік подій**:

1. Адміністратор CI заходить у веб-інтерфейс або консоль CI-сервера.
2. Адміністратор натискає на опцію створення нового проекту.
3. CI-сервер запитує основні параметри проекту, такі як назва проекту, репозиторій з вихідним кодом (URL Git, SVN тощо), гілка для відстеження.
4. Адміністратор вводить дані проекту, включаючи URL репозиторію та гілку, яку потрібно збирати.
5. CI-сервер запитує вибір технології збірки або build-скрипта (наприклад, Maven, Gradle для Java, MSBuild для C#, або інший інструмент залежно від мови).
6. Адміністратор вказує відповідний build-скрипт або конфігураційний файл для збирання (наприклад, pom.xml для Maven або build.gradle для Gradle).
7. Адміністратор налаштовує тригери для автоматичного запуску збірки (наприклад, на кожен push у репозиторій або за розкладом).
8. CI-сервер створює проєкт і перевіряє доступ до репозиторію.
9. Адміністратор зберігає конфігурацію проєкту, і CI-сервер готовий до запуску збірки при внесенні змін у код.

**Виключення**:

* **Невірний URL репозиторію**: Якщо вказаний репозиторій недоступний або URL невірний, CI-сервер повідомляє про помилку і пропонує перевірити дані.
* **Невідповідний build-скрипт**: Якщо CI-сервер не може знайти або розпізнати вказаний build-скрипт (наприклад, помилка у конфігурації файлу), сервер зупиняє налаштування і відправляє повідомлення про помилку адміністратору.

**Примітки**: Після створення проєкту адміністратор може також налаштувати додаткові параметри, такі як інтеграція з тестовими фреймворками або налаштування середовищ для деплою.

**Сценарій 3: Ведення статистики збірок**

**Передумови**: У системі вже виконувались кілька збірок проектів.  
**Постумови**: Статистика збірок збережена і доступна для перегляду.  
**Взаємодіючі сторони**: Адміністратор CI, CI-сервер.  
**Короткий опис**: Сценарій описує процес збирання та збереження статистики успішних і невдалих збірок для аналізу продуктивності.

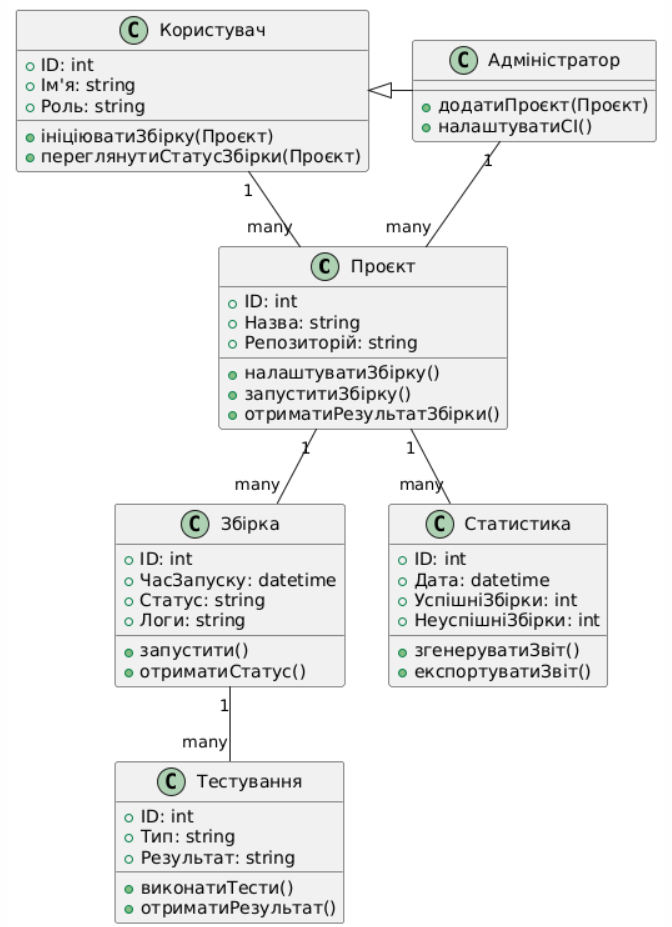
**Основний потік подій**:

1. CI-сервер автоматично зберігає дані про кожну виконану збірку.
2. Адміністратор CI може переглядати звіти з успішних та невдалих збірок у вигляді статистичних даних.
3. CI-сервер відображає детальну інформацію про кожну збірку, включаючи дату, час, статус (успішна або невдала) та причини можливих помилок.
4. Адміністратор CI використовує ці дані для покращення процесу збірки.

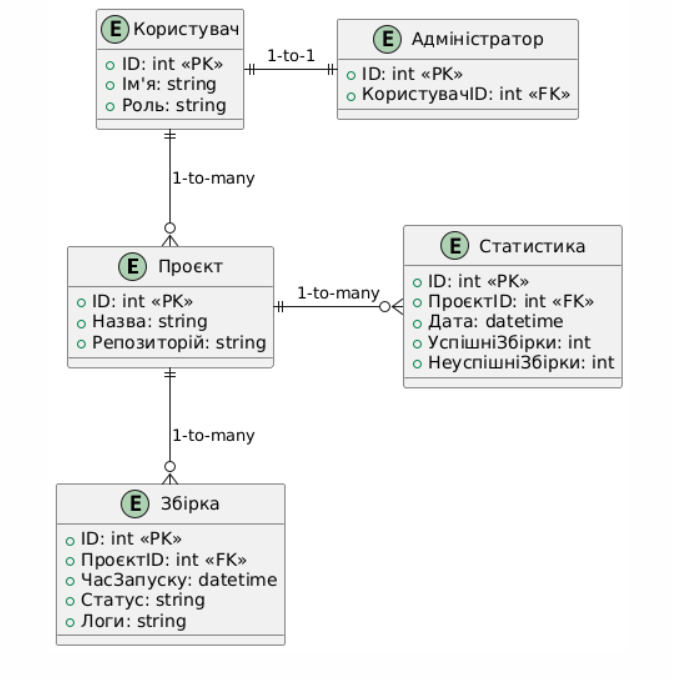
**Виключення**. Відсутність даних: Якщо сервер з якоїсь причини не зміг зберегти дані збірки, адміністратору буде надіслано повідомлення про цю помилку.

**Примітки**: немає.

1. **Діаграма класів**

****

1. **Cхема бази даних**

****

**Висновки**: на цій лабораторній роботі я проаналізував свою тему, створив діаграму прецедентів, діаграму класів та спроектував базу даних.