



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні системи та технології

Лабораторна робота №3
Технології розроблення програмного забезпечення
*Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій
та послідовностей.*

Варіант 20 (Mind-mapping software)

Виконав студент групи ІА-13:

Пасинчук Г.Б.

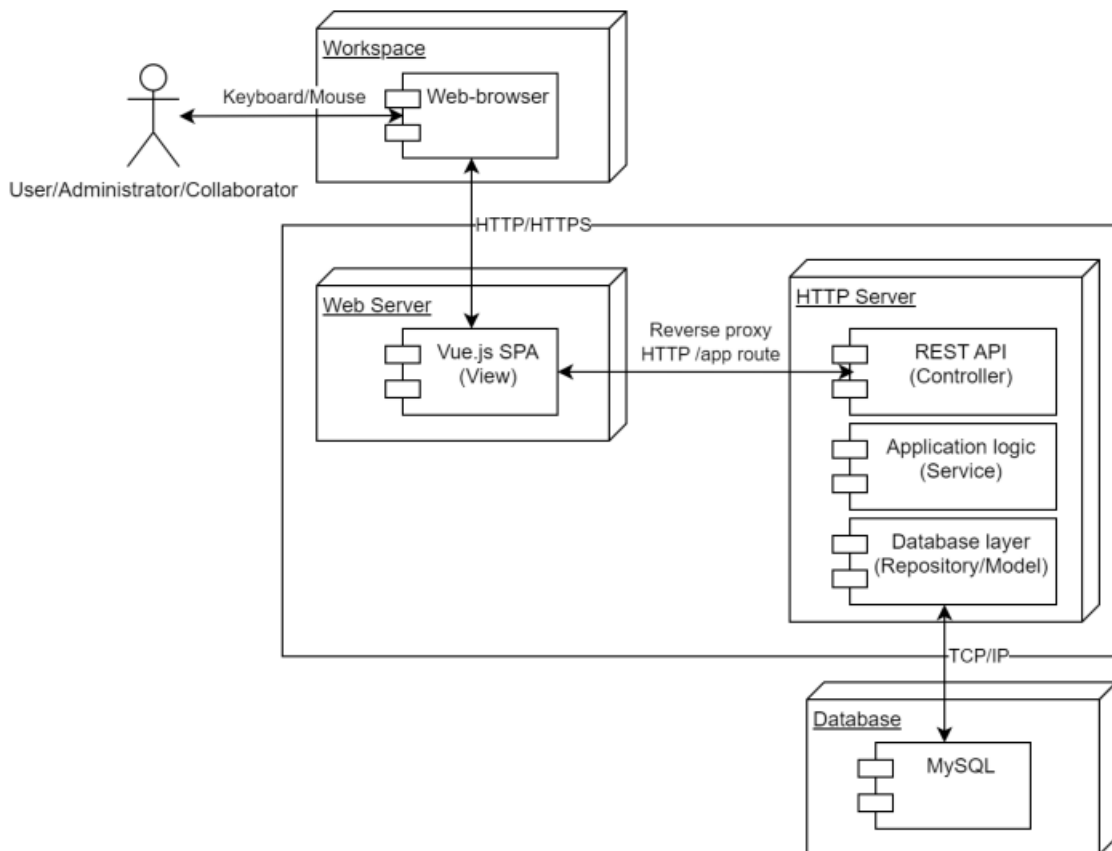
Перевірив:

Мягкий М.Ю.

Київ 2023

Мета роботи: навчитися створювати діаграму розгортання, компонентів, взаємодії та послідовностей.

Хід роботи:

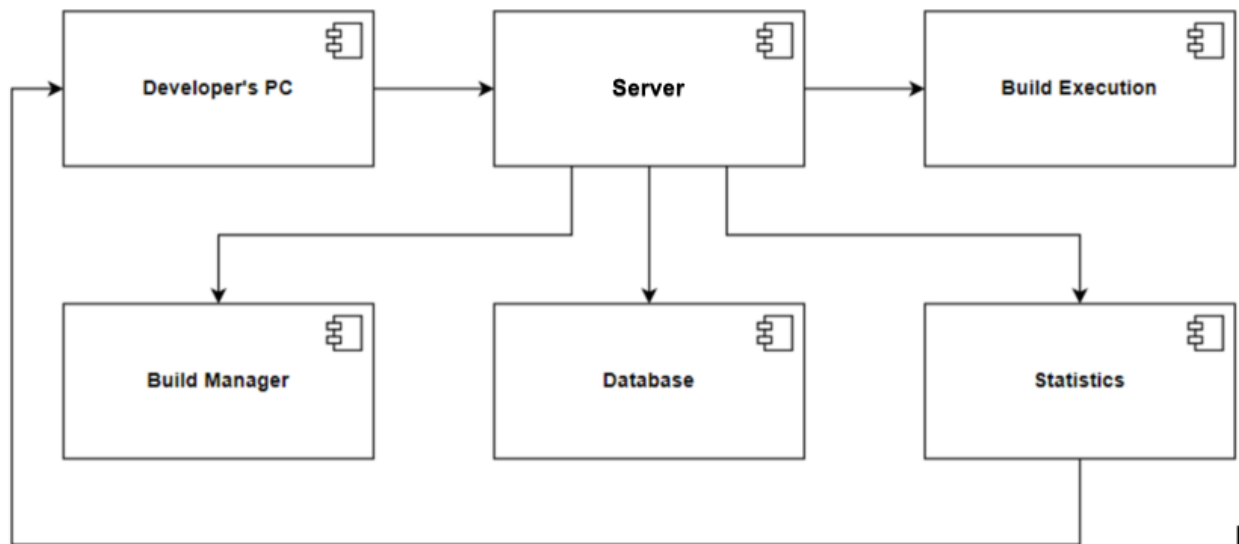


Діаграма розгортання

На схемі схематично зображено блок основної частини застосунку, що насправді складається з двох взаємопов'язаних блоків через Reverse-проху від сервера з Vue.js до HTTP сервера. Завдяки цьому ми можемо закрити доступ до серверної частини від стороннього користувача, та дозволяти лише шляхи, які використовуються frontend-сервером, а також з'являється чітке розділення шарів .

Архітектурний стиль серверної частини побудований на принципах REST API [2], [6]. До цього відносяться принципи уніфікованості інтерфейсу взаємодії, відсутність стану, роздільність системи на декілька шарів тощо. При створенні системи з мінімальними міркуваннями безпеки, але з високими вимогами до швидкості, REST є відмінним вибором. REST API-інтерфейси мають менше вимог до безпеки, покращують сумісність з клієнтом браузеру, краще відкриття, працездатність даних і масштабованість – речі, які дійсно застосовні до веб-сервісів.

Також на схемі зображено вузол з абстракцією бази даних, що працює з СКБД, а також його зв'язок із застосунком по протоколу TCP/IP. Користувач взаємодіє з системою через веб-браузер.



Діаграма компонентів

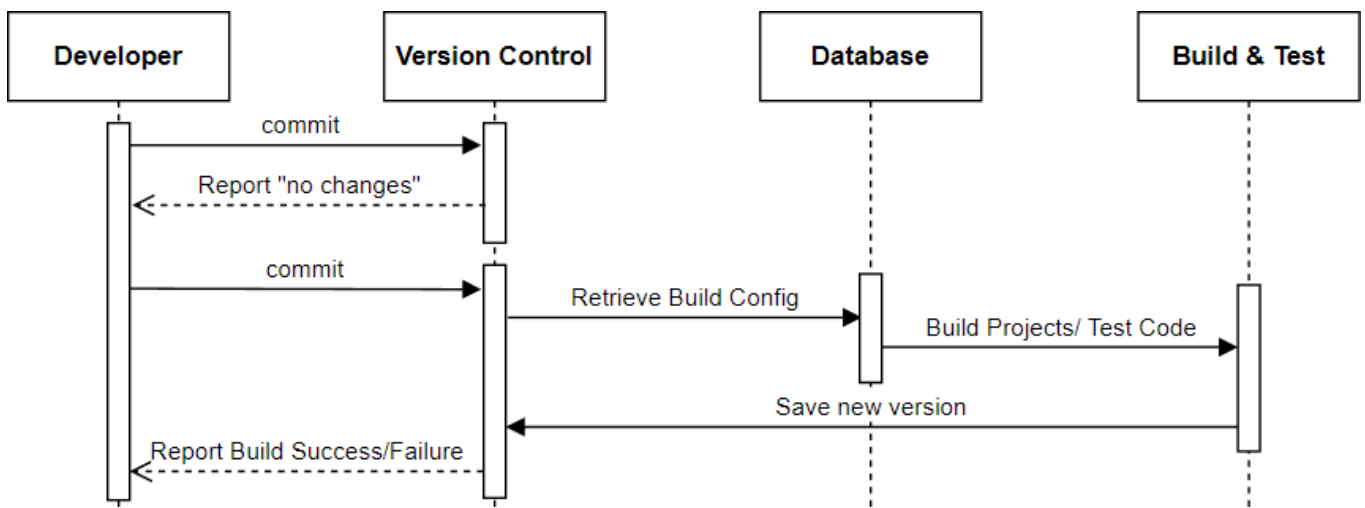
У цій діаграмі:

Build Manager - цей компонент відповідає за ведення обліку проектів та їх залежностей, розпізнавання build-скриптів, виконання компіляції та тестування, а також за ведення статистики.

Project Database - база даних проектів, яка зберігає інформацію про проекти та їх залежності.

Build Execution - компонент, який фактично виконує компіляцію, тестування та розпізнавання build-скриптів відповідно до вказівок, отриманих від Build Manager.

Statistics - цей компонент веде статистику про успішні та невдалих збірок проектів.



Діаграма послідовностей

Надана діаграма послідовності ілюструє роботу програми «сервер CI», коли користувач запускає збірку. Ось покроковий опис операцій, зображених на схемі:

1. Розробник/Користувач запускає збірку (commit):
 - Послідовність починається з того, що актор Розробник(Developer) ініціює збірку, ініціюючи запит на збірку.
2. Перевірка версії збірки:
 - Якщо версія співпадає з версією попередньої збірки то відправляється сповіщення Розробнику/Користувачу
3. Отримання конфігурації збірки:
 - Отримання конфігурації збірки з бази даних (Database), необхідну для процесу збірки.
4. Створення та тестування (Build & Test):
 - Створення проекту вхідного коду
 - Виконання тестів для перевірки правильності коду
5. Збереження збірки
 - Після виконаних тестів оновлюється версія нової збірки
6. Сповіщення Розробника/Користувача:
 - Сповіщення Розробника/Користувача про успішні/неуспішні тести збірки.

Ця діаграма послідовності надає загальнорівневий огляд взаємодії між користувачем, сервером CI та базою даних під час процесу запуску та завершення збірки.

Висновок: виконуючи дану лабораторну роботу, я побудував діаграму розгортання, побудував діаграму компонентів та діаграму послідовностей для Mind-mapping software.