

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

Технології розроблення програмного забезпечення

Тема: «Основи проектування розгортання»

Виконала:

Студентка групи ІА-34

Сизоненко А.О

Перевірив:

Мягкий Михайло Юрійович

Мета: Навчитися проєктувати діаграми розгортання та компонентів для системи що проєктується, а також розробляти діаграми взаємодії, а саме діаграми послідовностей, на основі сценаріїв зроблених в попередній лабораторній роботі.

Тема роботи: Текстовий редактор (strategy, command, observer, template method, flyweight, SOA)

Текстовий редактор повинен вміти розпізнавати текстові файли в будь-якій кодуванні, мати розширені функції редагування: макроси, сніппети, підказки, закладки, перехід на рядок / сторінку, підсвічування синтаксису (для однієї мови програмування або розмітки на розсуд студента).

Теоретичні відомості

Діаграма розгортання (Deployment Diagram) – важливий інструмент у моделюванні архітектури програмних систем. Вона показує, як програмні компоненти (якщо вони є частинами системи) розгортаються на фізичних пристроях, таких як сервери, комп'ютери або інші мережеві вузли. Діаграма розгортання відображає фізичну структуру системи, вказуючи на те, як різні частини програмного забезпечення взаємодіють між собою через апаратні компоненти. Вона допомагає зрозуміти конфігурацію мережі, з'єднання між вузлами та розподіл ресурсів на різних пристроях.

Діаграма компонентів (Component Diagram) – показує структуру системи з точки зору її компонентів та їх взаємодії. Вона використовується для візуалізації фізичних компонентів програмної системи, таких як модулі, бібліотеки, сервіси або інші програмні одиниці, та їх зв'язків. Діаграма компонентів корисна для визначення того, як різні частини системи будуть взаємодіяти між собою на етапі реалізації та для аналізу масштабованості і модульності системи.

Діаграма послідовностей (Sequence Diagram) – ілюструє, як об'єкти або компоненти взаємодіють один з одним протягом певного сценарію або процесу. Вона зображує обмін повідомленнями між об'єктами в часі, де час йде вертикально, а повідомлення між об'єктами — горизонтально. Діаграма послідовностей дозволяє візуалізувати порядок виконання дій і процесів, що допомагає у розумінні логіки роботи системи на етапі розробки та тестування.

Хід роботи

1. Діаграма розгортання Deployment Diagram.

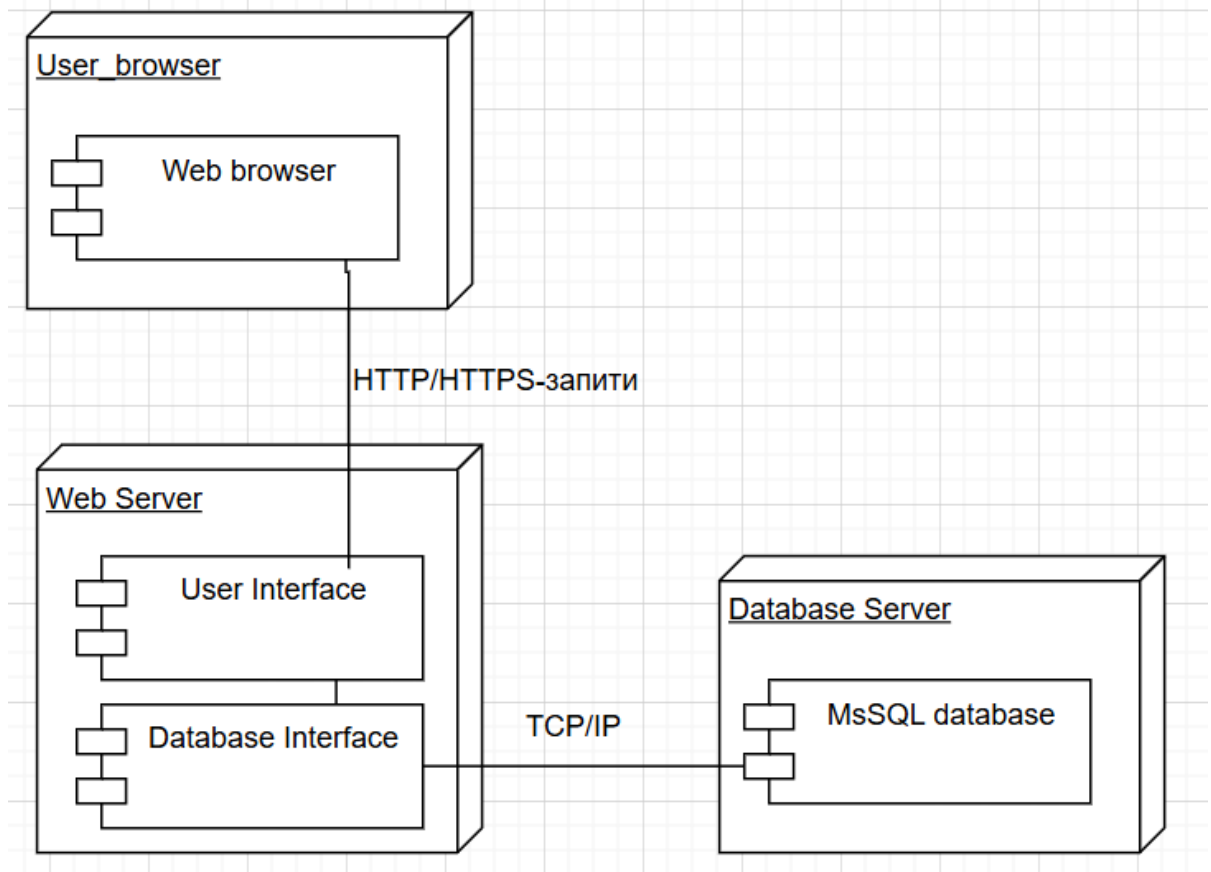


Рисунок 1 – Діаграма розгортання

1. User_browser

Компоненти: Web browser.

Призначення: Клієнтський вузол, через який користувач взаємодіє з текстовим редактором.

Функції:

- Відображення інтерфейсу редактора.
- Надсилання HTTP/HTTPS-запитів до Web Server для відкриття, редагування та збереження файлів.
- Підтримка інтерактивних елементів: закладки, підказки, перехід на рядок/сторінку, підсвічування синтаксису.

2. Web Server

Компоненти:

- User Interface: обробляє запити від браузера, відповідає за відображення редактора, макросів, сніпетів і підказок.
- Database Interface: забезпечує взаємодію з базою даних для збереження та отримання текстових файлів і закладок.

Призначення: Посередник між клієнтом та базою даних, керує логікою роботи редактора.

Комунікації:

- Отримує HTTP/HTTPS-запити від клієнта.
- Працює з базою даних через TCP/IP для збереження і читання даних.

3. Database Server

Компоненти: MsSQL database.

Призначення: Зберігає всі дані редактора, включно з текстовими файлами, закладками, макросами та іншими налаштуваннями користувача.

Комунікації: Підключається до Web Server через TCP/IP.

Загальна логіка роботи

1. Користувач відкриває веб-браузер і заходить у текстовий редактор.
2. Браузер відправляє HTTP/HTTPS-запити на Web Server для роботи з файлами.
3. Web Server обробляє запити, реалізує функції редагування (макроси, сніпети, підсвічування синтаксису, закладки тощо).
4. При потребі зберігання або отримання даних Web Server звертається до Database Server через TCP/IP.
5. Database Server виконує запити і повертає дані на Web Server, який оновлює інтерфейс користувача.

2. Діаграма компонентів.

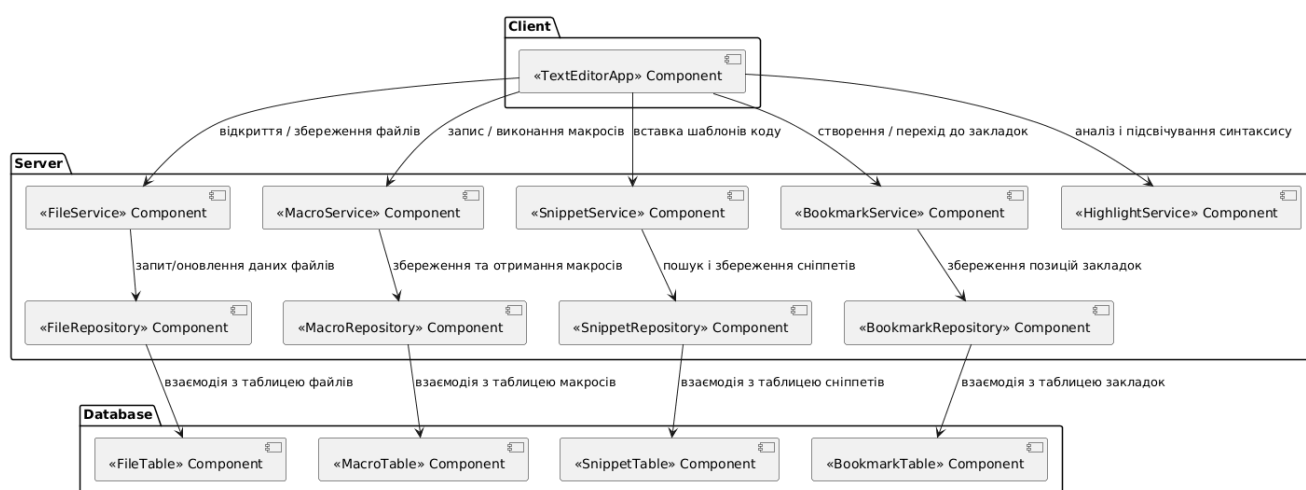


Рисунок 2 –діаграма компонентів

На діаграмі зображено архітектуру текстового редактора, де система поділена на три основні частини: Client, Server та Database.

1. Client

Компонент TextEditorApp представляє клієнтську частину системи — графічний інтерфейс користувача (GUI). Він забезпечує взаємодію користувача з

редактором: відкриття файлів, запуск макросів, вставку сніппетів, керування закладками та підсвічуванням синтаксису. Усі запити від користувача передаються на сервер через відповідні сервіси.

2. Server

Серверна частина містить п'ять сервісів, кожен з яких відповідає за окрему функціональність редактора:

- FileService – відкриття, збереження, кодування текстових файлів;
- MacroService – запис, виконання та управління макросами;
- SnippetService – керування сніппетами та шаблонами коду;
- BookmarkService – створення, редагування та збереження закладок;
- HighlightService – аналіз тексту та підсвічування синтаксису.

Кожен сервіс взаємодіє зі своїм Repository, який відповідає за доступ до даних у базі.

3. Repositories

Усі репозиторії розміщені на сервері. Вони реалізують шар доступу до даних і забезпечують взаємодію між сервісами та базою даних:

- FileRepository, MacroRepository, SnippetRepository, BookmarkRepository.
Репозиторії отримують, зберігають і оновлюють відповідні дані у своїх таблицях.

4. Database

База даних містить таблиці:

- FileTable – збереження файлів і метаданих (назва, кодування, шлях);
- MacroTable – дані про користувацькі макроси;
- SnippetTable – шаблони коду та фрагменти тексту;
- BookmarkTable – інформація про закладки (позиція, файл, коментар).

Сервер взаємодіє з базою через репозиторії, виконуючи запити на читання або запис даних.

3. Діаграма послідовностей.

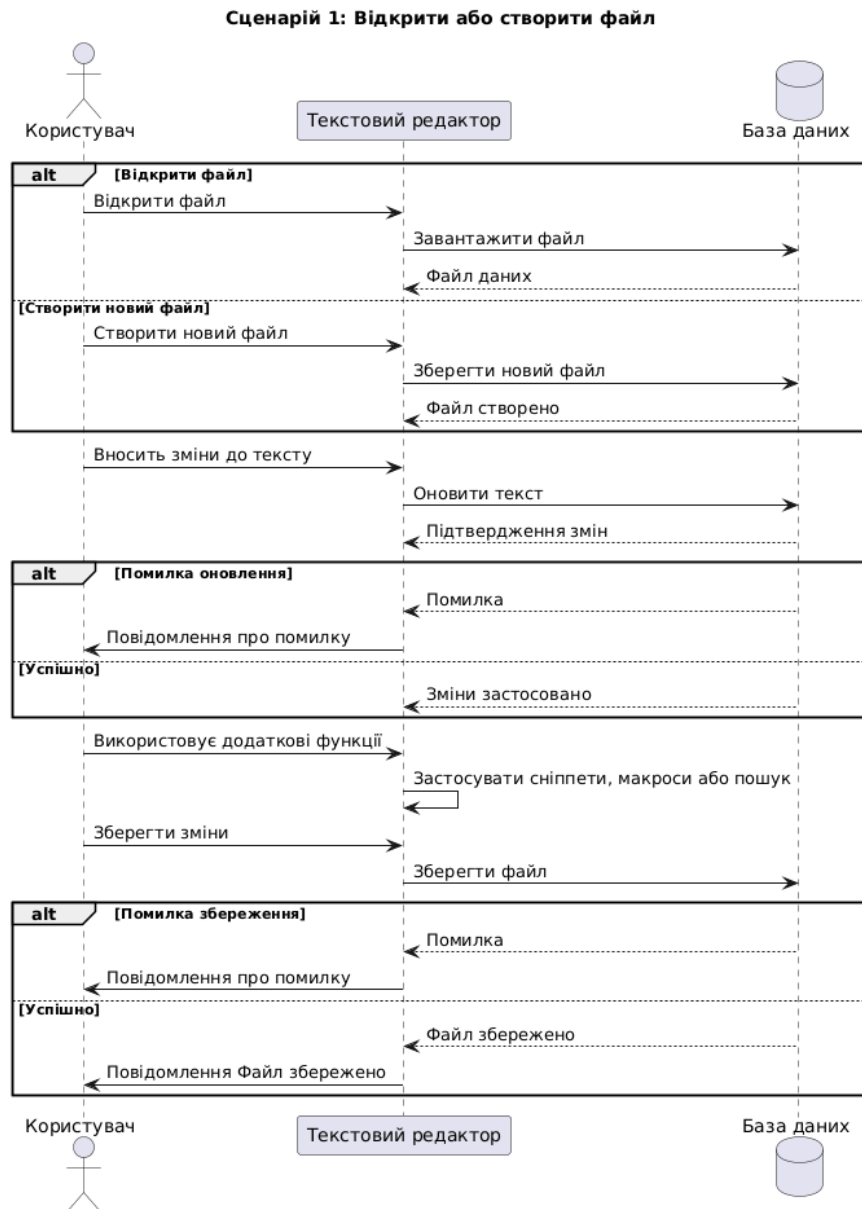


Рисунок 3 –діаграма послідовності “Редагування файлу”

Діаграма показує взаємодію користувача з текстовим редактором та базою даних під час роботи з файлом.

Користувач може виконати одну з двох альтернативних дій: відкрити існуючий файл або створити новий. У випадку відкриття редактор запитує файл у бази даних, яка повертає дані, а у випадку створення редактор зберігає новий файл у бази даних і підтверджує його створення.

Далі користувач вносить зміни до тексту, редактор оновлює дані у базі, яка підтверджує зміни або повертає помилку, яку редактор передає користувачу. Користувач може застосовувати додаткові функції, такі як сніппети, макроси або пошук, що виконуються всередині редактора.

Наприкінці користувач зберігає зміни, редактор відправляє файл у базу даних, яка повертає підтвердження або помилку, а редактор повідомляє користувача про результат збереження.

Сценарій 3: Управління закладками

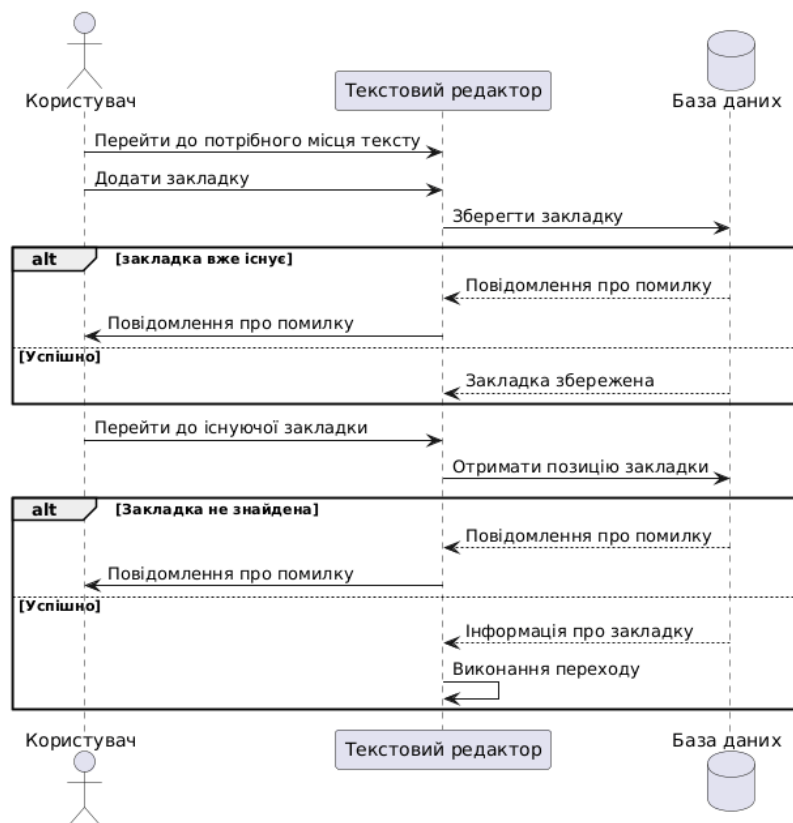


Рисунок 4 –діаграма послідовності “Управління закладками”

Діаграма показує процес взаємодії користувача з текстовим редактором та базою даних під час управління закладками. Користувач спочатку переходить до потрібного місця тексту у документі через редактор.

Далі користувач додає закладку, редактор надсилає запит до бази даних для збереження. Якщо закладка з таким ім'ям уже існує, база даних повертає повідомлення про помилку, яке редактор передає користувачу. У випадку успішного збереження база підтверджує, що закладка збережена.

Потім користувач може перейти до існуючої закладки. Редактор отримує позицію закладки з бази даних. Якщо закладка не знайдена, база повідомляє про помилку, і редактор передає її користувачу. Якщо ж закладка існує, база даних повертає дані закладки і редактор виконує перехід до зазначеного місця тексту.

Висновок: Під час виконання лабораторної роботи я навчилася створювати діаграми розгортання, компонентів і послідовностей. Діаграма розгортання показує, як система працює на різних пристроях і як вони між собою взаємодіють. Діаграма компонентів допомагає зрозуміти, з яких частин складається програма, які завдання вони виконують і як пов'язані між собою. Діаграма послідовностей показує, у якому порядку виконуються дії користувача та як система на них реагує. Завдяки цим діаграмам стало зрозуміліше, як працює система в цілому і як її реалізувати.

Відповіді на контрольні питання

1. Що собою становить діаграма розгортання?

Діаграма розгортання показує фізичне розташування компонентів системи на апаратних вузлах і взаємодію між ними.

2. Які бувають види вузлів на діаграмі розгортання?

Вузли бувають двох типів: пристрій — фізичне обладнання, наприклад комп'ютер або сервер, і середовище виконання — програмне забезпечення, яке може включати інші програми, як операційна система або вебсервер.

3. Які бувають зв'язки на діаграмі розгортання?

На діаграмі розгортання використовуються зв'язки, такі як асоціації, залежності та шляхи комунікації між вузлами.

4. Які елементи присутні на діаграмі компонентів?

Діаграма компонентів відображає структуру системи у вигляді компонентів, їхніх інтерфейсів та пакетів.

5. Що становлять собою зв'язки на діаграмі компонентів?

Зв'язки між компонентами показують залежності, асоціації або реалізацію інтерфейсів.

6. Які бувають види діаграм взаємодії?

Діаграми взаємодії можуть бути різних видів, зокрема діаграми послідовностей, комунікацій, часу та огляду взаємодії.

7. Для чого призначена діаграма послідовностей?

Діаграма послідовностей призначена для відображення часової послідовності повідомлень між об'єктами під час виконання сценарію.

8. Які ключові елементи можуть бути на діаграмі послідовностей?

Основні елементи діаграми : актори позначають користувачів або зовнішні системи, що взаємодіють із системою. Об'єкти або класи розміщені горизонтально та мають лінію життя, яка показує їх існування в часі.

Повідомлення зі стрілками відображають виклики методів або обмін даними, синхронні або асинхронні, з пунктирними лініями для повернення результату.

Активності показують періоди виконання дій об'єктом і зображуються прямокутниками на лінії життя. Контрольні структури, як alt або loop, відображають умови, альтернативи та цикли.

9. Як діаграми послідовностей пов'язані з діаграмами варіантів використання?

Діаграми послідовностей деталізують сценарії, які представлені на діаграмах варіантів використання, показуючи порядок обміну повідомленнями для конкретних випадків.

10. Як діаграми послідовностей пов'язані з діаграмами класів?

Вони пов'язані з діаграмами класів, оскільки показують, як методи класів викликаються під час виконання сценарію, відображаючи динамічну поведінку структури класів.