



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

із дисципліни «**Технології розроблення програмного забезпечення**» Діаграма
розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей

Варіант №5. Аудіоредактор

Виконала
студентка групи ІА-31
Биковська О. І.

Перевірив
викладач
Мягкий М.Ю.

Київ 2025 Мета	Ошибка! Закладка не определена.
Теоретичні відомості.....	3
Хід роботи	5
Діаграма розгортання	6
Діаграма компонентів.....	7
Діаграма послідовностей	8
Висновок.....	9

Зміст

Мета: Вивчення основних принципів побудови та проектування програмних систем з використанням діаграм розгортання, діаграм компонентів, діаграм взаємодій та послідовностей

Теоретичні відомості

Діаграма розгортання (Deployment Diagram)

Діаграми розгортання показують фізичне розташування елементів системи, демонструючи, на якому обладнанні працює програмне забезпечення. Головні елементи діаграми — вузли, які пов’язані між собою інформаційними шляхами. Вузли бувають двох типів: пристрой (фізичне обладнання, як-от комп’ютери) і середовище виконання (програмне забезпечення, яке може включати інше). В узлах можна деталізувати артефакти, наприклад, компоненти чи класи. Проте така деталізація рідко потрібна, оскільки вона може змістити фокус із розгортання системи на її структуру.

Діаграма компонентів (Component Diagram)

Ця діаграма показує систему, розбиту на модулі. Є три типи діаграм компонентів: логічні, фізичні й виконавчі. Найчастіше використовують логічне розбиття, коли система уявляється як набір автономних модулів, які можуть взаємодіяти між собою. Наприклад, система продажів може складатися з каси, черги повідомлень, сервера продажів і системи обліку. Компоненти можуть належати до різних фізичних вузлів, проте взаємозамінність робить їх гнучкими для клієнтів. У фізичному поділі кожен компонент може бути розташований на окремому сервері чи комп’ютері, але такий підхід застарів і зазвичай замінюється діаграмами розгортання. Виконавчі діаграми описують компоненти як файли, наприклад, .exe чи бази даних.

Діаграма діяльності (Activity Diagram)

Цей тип діаграм моделює виконання операцій, їх логіку і порядок переходів між діями. Діяльність — це набір обчислень, які приводять до певного результату. Графічно вона схожа на діаграми станів, але кожне її стан — це виконання конкретної операції. Дія, завершуючись, передає керування наступному стану. Стан 4 дії зображається прямокутником із закругленими кутами, всередині якого вказується унікальне ім'я дії. Якщо дія складна, вона може бути представлена як піддіяльність, яка позначається спеціальною піктограмою.

Діаграма послідовності (Sequence Diagram)

Ця діаграма показує взаємодію об'єктів у часі. На вертикальній осі розташовані лінії життя об'єктів, які позначають час їх існування в системі. Ініціатор взаємодії знаходитьться зліва. Об'єкти можуть бути створені або знищенні в будь-який момент. Знищенння позначається символом "X". Лінії взаємодії між об'єктами відображають обмін повідомленнями, їх порядок і час виконання.

Хід роботи

1. Ознайомитися з короткими теоретичними відомостями.
2. Розробити діаграму розгортання для проектованої системи.
3. Розробити діаграму компонентів для проектованої системи.
4. Розробити діаграму послідовностей для проектованої системи.
5. Скласти звіт про виконану роботу..

Діаграма розгортання

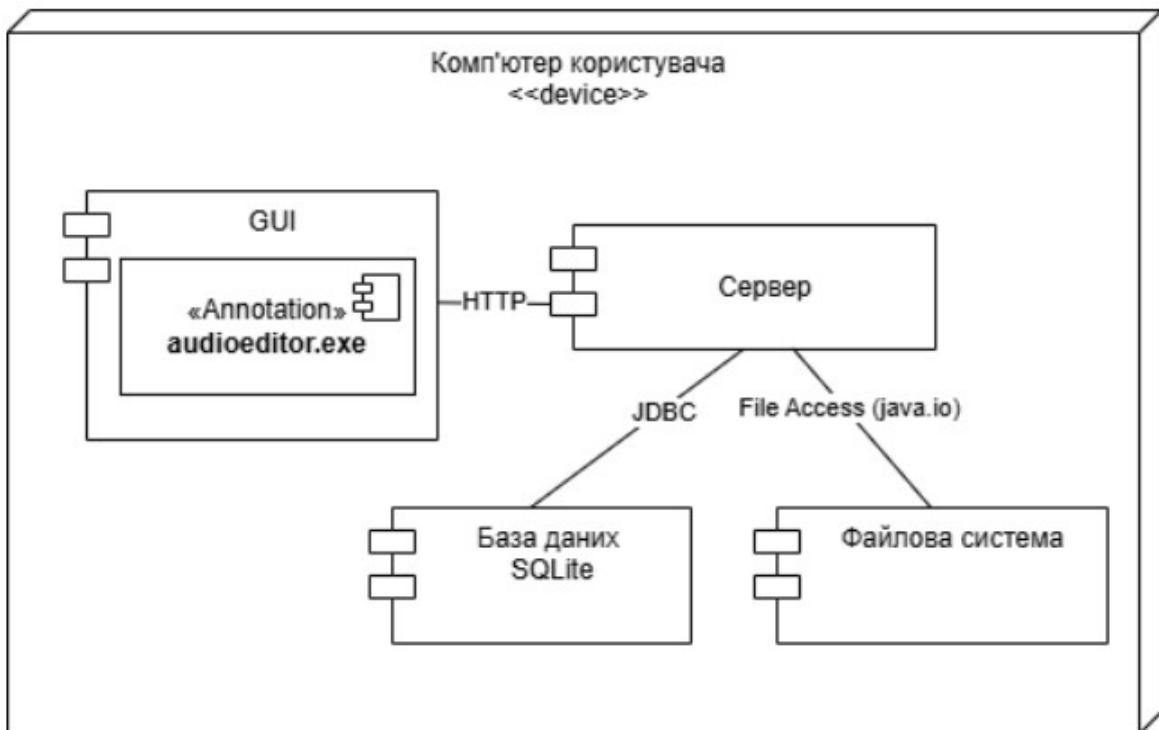


Рисунок 1. Діаграма розгортання

Діаграма розгортання демонструє архітектуру аудіоредактора, який розгорнуто на комп'ютері користувача, із взаємодією між різними компонентами. Основний пристрій позначено як Комп'ютер користувача (позначений $\langle\rangle$). На цьому пристрої розгорнуто GUI (графічний інтерфейс користувача), який взаємодіє із серверною частиною. GUI включає виконуваний файл аудіоредактора, що забезпечує інтерфейс та основний функціонал для користувача. GUI надсилає запити до компонента Сервер, який відповідає за виконання основної бізнес-логіки та обробку даних. Сервер має доступ до двох ресурсів: База даних SQLite, до якої сервер звертається через JDBC для зберігання та отримання даних; файлова система, з якою сервер працює через механізми доступу до файлів, наприклад, для читання та збереження аудіофайлів.

Діаграма компонентів

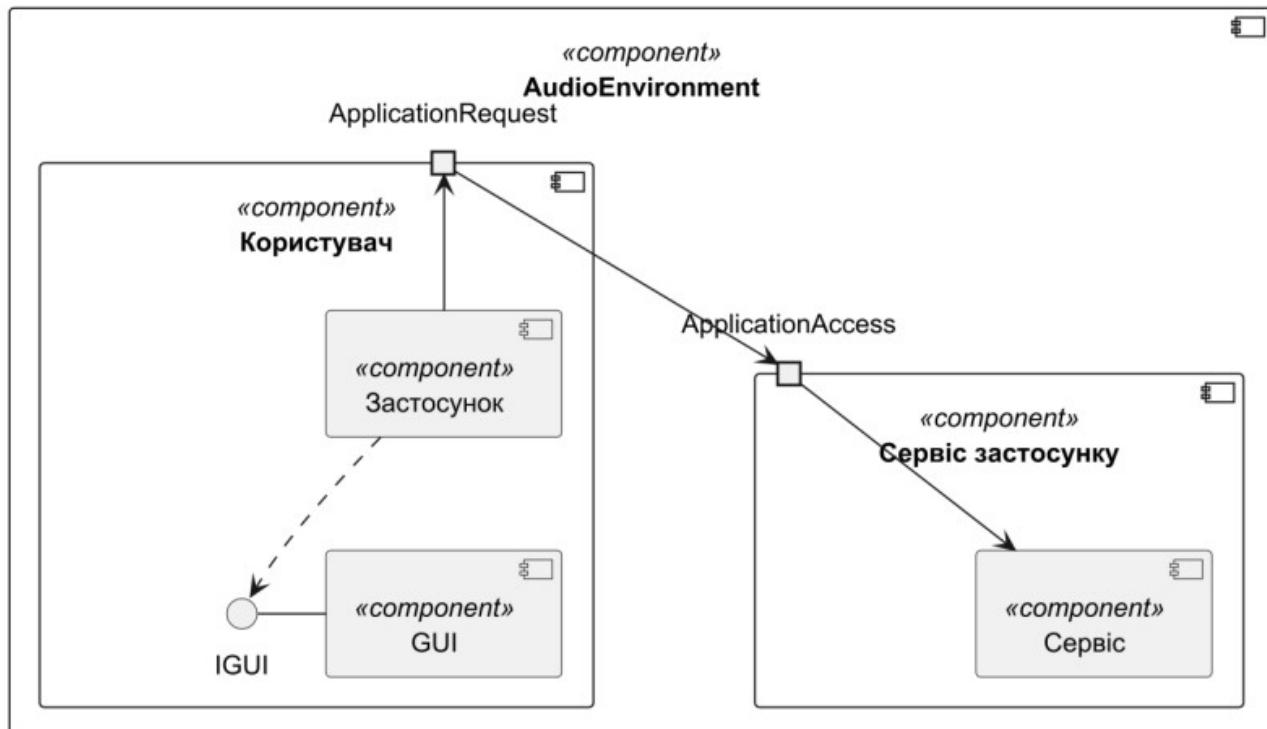


Рисунок 2. Діаграма компонентів

Ця діаграма відображає взаємодію між користувачем, застосунком, GUI, та сервісною частиною системи у вигляді компонентної архітектури. У складі компонента **AudioEnvironment** виділено три основні частини: Користувач, Застосунок, і Сервіс застосунку. Користувач взаємодіє із системою через компонент **GUI**, який інкапсулює графічний інтерфейс користувача. **GUI** реалізує інтерфейс **IGUI**, забезпечуючи доступ до функціоналу, який представлений у застосунку. Застосунок через **ApplicationRequest** надсилає запити на доступ до можливостей системи, зокрема, до сервісної частини. Застосунок напряму пов'язаний із **GUI** через інтерфейс **IGUI**, що дозволяє оновлювати відображення та отримувати дані від користувача. У сервісній частині виділено компонент **Сервіс**, який відповідає за виконання запитів застосунку. Застосунок взаємодіє з цим компонентом через **ApplicationAccess**, забезпечуючи доступ до внутрішніх ресурсів або логіки системи.

Діаграма послідовностей

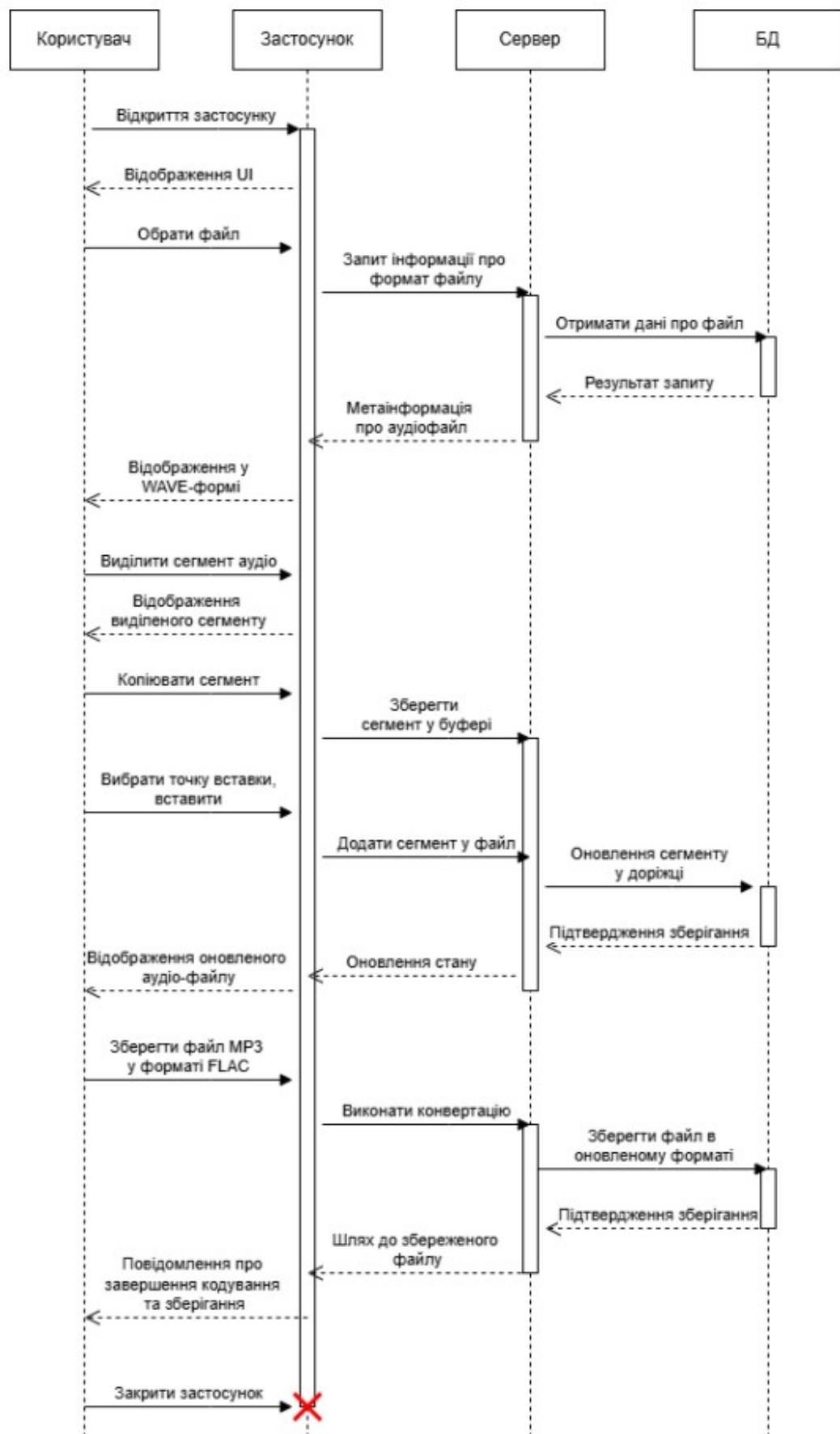


Рисунок 3. Діаграма послідовностей

На діаграмі послідовностей зображенено процес роботи користувача з аудіоредактором, який включає взаємодію між користувачем, застосунком, сервером і базою даних. Процес починається з відкриття застосунку, після чого відображається графічний інтерфейс. Користувач обирає аудіофайл для роботи, і застосунок надсилає запит на сервер для отримання метаінформації про формат файлу. Сервер звертається до бази даних, отримує дані про файл і передає їх назад до застосунку. Далі застосунок відображає аудіофайл у вигляді WAVE-форми. Користувач виконує різні операції над аудіофайлом, зокрема виділення, видалення сегментів, копіювання сегментів у буфер, а також вибір точки вставки і додавання сегменту до іншої частини файлу. Усі зміни відображаються у застосунку, оновлюючи стан файлу. Після редагування користувач може зберегти файл у форматі FLAC. Для цього застосунок передає дані серверу, де виконується 7 конвертація, після чого файл зберігається в новому форматі в базі даних або файловій системі. Сервер підтверджує успішне збереження і передає шлях до збереженого файла застосунку, який повідомляє користувача про завершення операції. Після цього користувач закриває застосунок, завершуючи роботу. Описаний процес зображенено на Рисунку 3

Висновок

У ході виконання роботи було проведено аналіз основних теоретичних аспектів, пов'язаних із розробкою діаграм у рамках моделювання систем за допомогою UML.

Було виконано наступні завдання:

1. Розроблено діаграму розгортання, яка показує фізичну архітектуру системи, зокрема розподіл програмних компонентів між апаратними вузлами та зв'язки між ними.
2. Розроблено діаграму компонентів, яка відображає логічну архітектуру системи, демонструючи залежності між основними модулями та компонентами програмного забезпечення.

3. Розроблено діаграму послідовностей, що деталізує взаємодію об'єктів у системі під час виконання ключових сценарій.
4. Сформовано звіт, у якому узагальнено результати роботи, представлено створені діаграми та пояснено їхнє призначення.

Завдяки виконанню цих завдань вдалося досягти таких результатів:

- Візуалізовано ключові аспекти архітектури системи, що дозволяє краще зрозуміти її структуру та взаємодію компонентів.
- Створені діаграми можуть бути використані для комунікації між членами команди, документації проекту та подальшого впровадження системи.
- Розроблені моделі слугуватимуть основою для вдосконалення проектованої системи на етапі її впровадження.

Таким чином, виконання цієї роботи сприяло поглибленню знань у сфері моделювання інформаційних систем, а також практичному застосуванню UML для проектування архітектури систем.

1. **Що собою становить діаграма розгортання?** Це структурна діаграма UML, яка моделює фізичну архітектуру системи, відображаючи розподіл програмних артефактів по апаратних обчислювальних ресурсах.
2. **Які бувають види вузлів на діаграмі розгортання?** Вузли поділяються на два основні типи: пристрой (фізичне обладнання з обчислювальними потужностями) та середовища виконання (програмні платформи, що функціонують усередині пристройів).
3. **Які бувають зв'язки на діаграмі розгортання?** Зв'язки на діаграмі розгортання представлені переважно шляхами комунікації (фізичні або мережеві з'єднання) та залежностями, що вказують на розміщення компонентів на вузлах.

4. **Які елементи присутні на діаграмі компонентів?** Основними елементами діаграми є компоненти (модульні частини системи), інтерфейси (надані та необхідні), порти та артефакти, що їх реалізують.
5. **Що становлять собою зв'язки на діаграмі компонентів?** Зв'язки на діаграмі компонентів відображають логічну взаємодію між модулями, зокрема через реалізацію інтерфейсів, використання необхідних інтерфейсів та делегування запитів.
6. **Які бувають види діаграм взаємодії?** До основних видів діаграм взаємодії в UML належать діаграма послідовностей, діаграма комунікації (кооперації), діаграма огляду взаємодії та діаграма часових параметрів.
7. **Для чого призначена діаграма послідовностей?** Діаграма послідовностей призначена для візуалізації взаємодії між об'єктами системи в часі, акцентуючи увагу на хронологічному порядку обміну повідомленнями.
8. **Які ключові елементи можуть бути на діаграмі послідовностей?** Ключовими елементами діаграми є лінії життя (учасники), смуги активації (фокус керування), повідомлення (виклик, повернення, створення) та комбіновані фрагменти для керування логікою взаємодії.