



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

з дисципліни «Технології розроблення програмного забезпечення»

Тема: «Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей»

Варіант №5

Виконала:  
студентка групи ІА-23  
Архип'юк Катерина

Перевірив:  
Мягкий Михайло Юрійович

Київ 2024

## Зміст

Завдання.....	2
Тема (Варіант №5).....	2
Хід роботи .....	3
1. Короткі теоретичні відомості .....	3
2. Діаграма розгортання .....	4
3. Діаграма компонентів.....	5
4. Діаграма послідовностей.....	6
Висновки: .....	8
Посилання на репозиторій з кодом проєкту .....	8

## Завдання

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

## Тема (Варіант №5)

### **..5 Аудіо редактор (singleton, adapter, observer, mediator, composite, client-server)**

Аудіо редактор повинен володіти наступним функціоналом: представлення аудіо даних будь-якого формату в WAVE-формі, вибір і подальші операції копіювання / вставки / вирізання / деформації по сегменту аудіозапису, можливість роботи з декількома звуковими доріжками, кодування в найбільш поширених форматах

(ogg, flac, mp3).

## Хід роботи

### 1. Короткі 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: пристрої (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

дії зображається прямокутником із заокругленими 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 знаходиться зліва. Об'єкти можуть бути створені або знищені в будь-який момент. Знищення позначається символом "X". Лінії взаємодії між об'єктами відображають обмін повідомленнями, їх порядок і час виконання.

## 2. 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 <<device>>). 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.

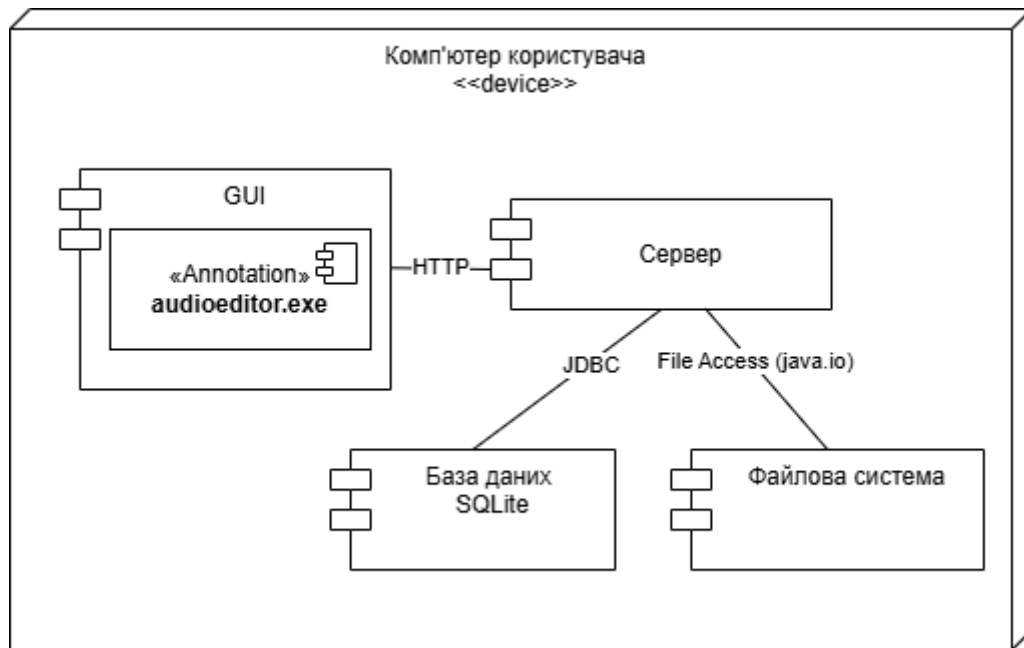


Рисунок 1 – Діаграма розгортання

### 3. 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.

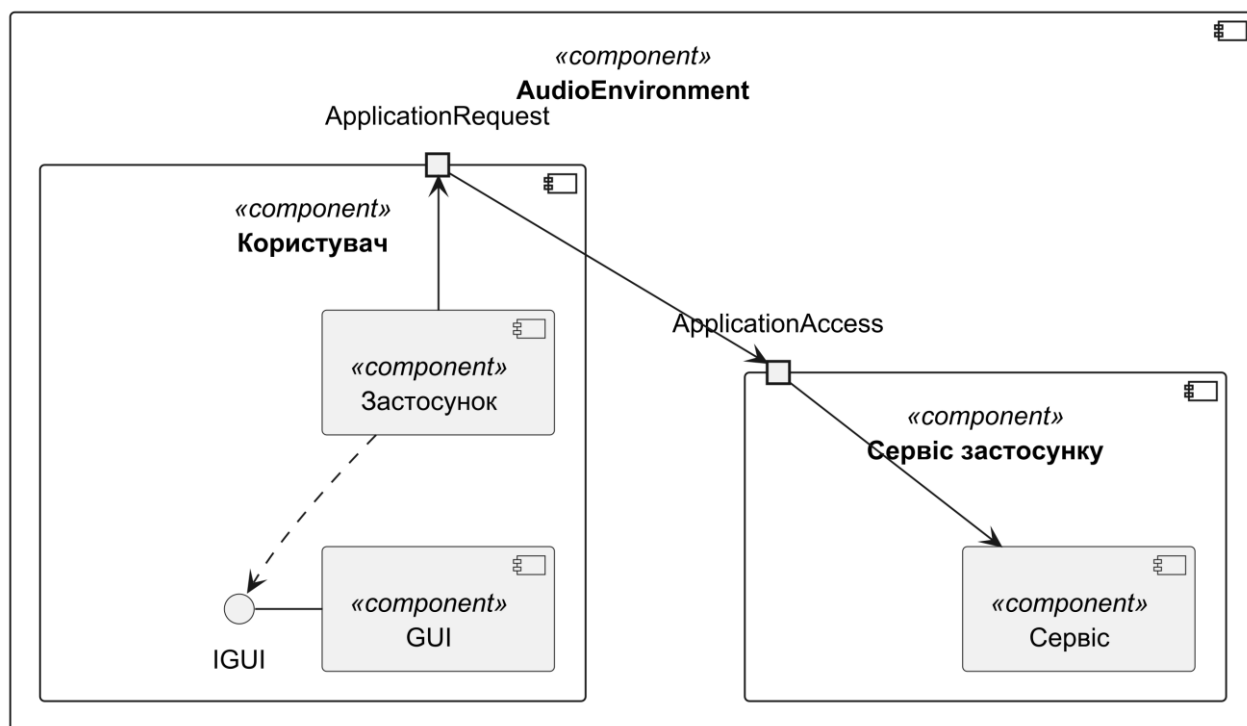


Рисунок 2 – Діаграма компонентів

#### 4. Діаграма послідовностей

На діаграмі послідовностей зображено 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

конвертація, після чого файл зберігається в новому форматі в базі даних або файловій системі. Сервер підтверджує успішне збереження і передає шлях до збереженого файлу застосунку, який повідомляє користувача про завершення операції. Після цього користувач закриває застосунок, завершуючи роботу. Описаний процес зображено на Рисунку 3.

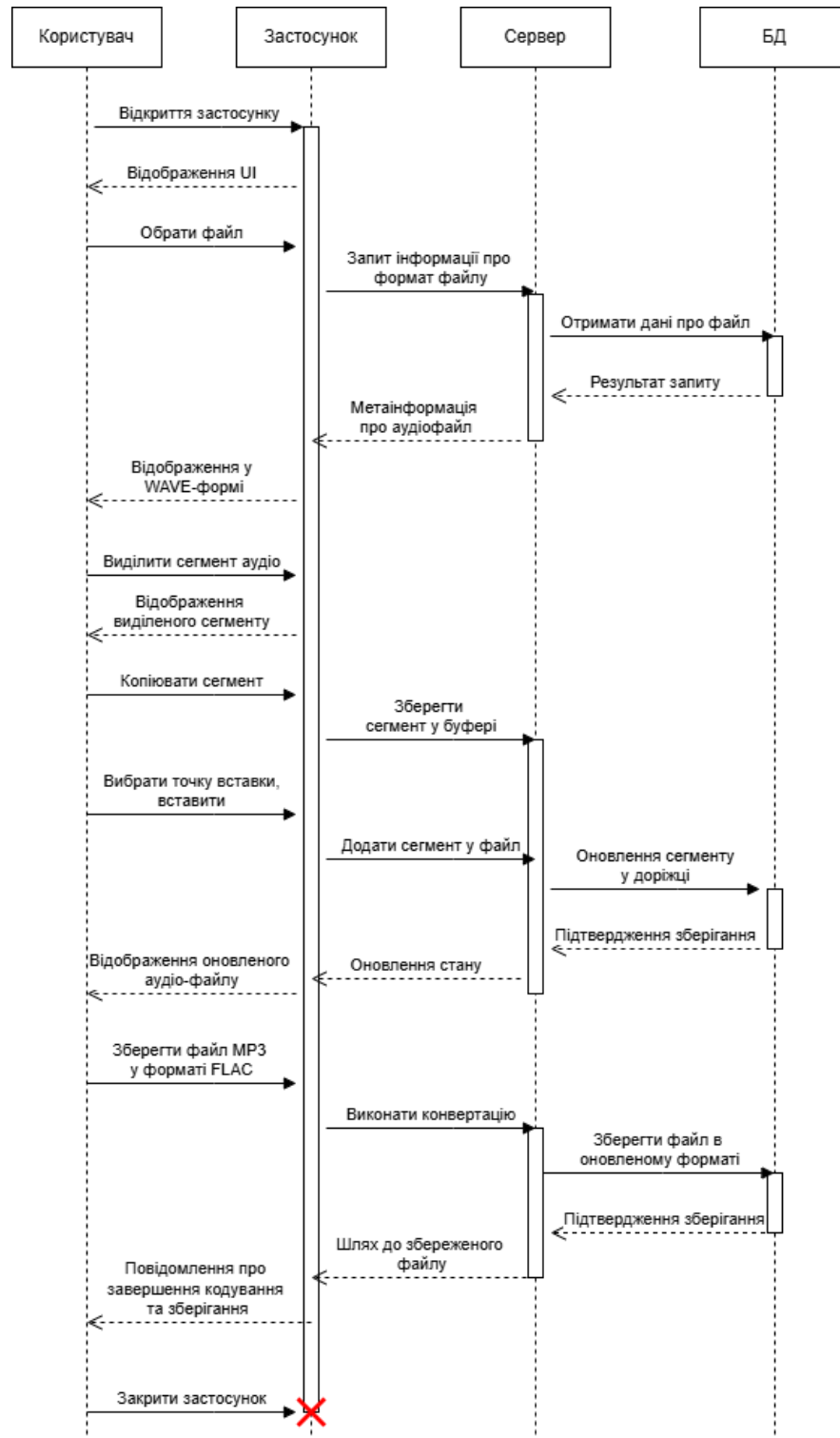


Рисунок 3 – Діаграма послідовностей

**Висновки:** При виконанні цієї лабораторної роботи я закріпила навички створення UML-діаграм, таких як діаграми розгортання, компонентів, взаємодій та послідовностей. Діаграма розгортання дозволяє моделювати фізичну архітектуру системи. Діаграма компонентів показує взаємозв'язки між окремими модулями програмного забезпечення, що дає чітке розуміння їхньої структури. Діаграми взаємодій та послідовностей демонструють, як елементи системи взаємодіють у часі для аналізу логіки процесів і сценаріїв використання.

**Посилання на репозиторій з кодом проєкту:**

<https://github.com/KatiaArkhyp/AudioEditor>