



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**  
**ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ**  
**ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ**  
**КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ**  
Office communicator

Виконав

студент групи ІА – 22:

Ільніцький П А

Перевірив:

Мягкий Михайло Юрійович

**Тема:** Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей.

**Мета:** Розробити діаграму розгортання, діаграму компонентів, діаграму взаємодій та послідовностей для проекрованої системи.

**Варіант:**

**..13 Office communicator (strategy, adapter, abstract factory, bridge, composite, client-server)**

Мережевий комунікатор для офісу повинен нагадувати функціонал програми Skype з можливостями голосового / відео / конференц-зв'язку, відправки текстових повідомлень і файлів (можливо, оффлайн), веденням організованого списку груп / контактів.

## Короткі теоритичні відомості

### Діаграма розгортання

Діаграми розгортання (Deployment Diagram) відображають фізичне розташування компонент системи та показують, на якому обладнанні запускається програмне забезпечення.

Основними елементами є вузли, що можуть бути:

- *Пристроями* (фізичне обладнання, наприклад, комп'ютери)
- *Середовищами виконання* (програмне забезпечення, яке запускає інші компоненти, наприклад, операційна система).

Вузли можуть бути з'єднані шляхами, що показують взаємодію між ними, з використанням різних протоколів чи технологій. У середині вузлів можуть бути артефакти — файли, які реалізують компоненти системи (виконувані файли, дані тощо).

Діаграми розгортання можуть бути:

- *Описовими*, які показують загальну структуру без деталей обладнання.
- *Екземплярними*, що включають конкретні приклади обладнання та програмних компонентів, і використовуються на пізніх етапах розробки для планування розгортання системи.

## Діаграма компонентів

Діаграма компонентів UML відображає систему, розбиту на окремі модулі. Виділяють три типи діаграм:

1. Фізичні
2. Виконувані
3. Логічні

Найчастіше використовують *логічне* розбиття — система представлена як набір незалежних компонентів, що взаємодіють між собою. Наприклад, система продажу продуктів може мати такі компоненти: каса, сервер продаж, система обліку тощо. Компоненти можуть бути розміщені на різних фізичних пристроях або в різних процесах.

*Фізичне* розбиття фокусується на розподілі компонент між серверами та комп'ютерами, що частіше відображається на діаграмі розгортання.

*Виконване* розбиття показує компоненти як файли (наприклад, .exe, html, бази даних), даючи інший ракурс системи. Однак цей підхід не має широкого використання через наявність діаграм розгортання.

Компоненти мають забезпечувати гнучкість для клієнтів, дозволяючи оновлювати систему частинами та забезпечувати сумісність між різними виробниками, хоча це важко реалізувати.

## Діаграма послідовностей

Діаграми послідовностей в UML використовуються для моделювання виконання операцій і взаємодії між об'єктами в системі. Вони відображають послідовність повідомлень між об'єктами у процесі виконання операцій, деталізуючи алгоритми і логіку.

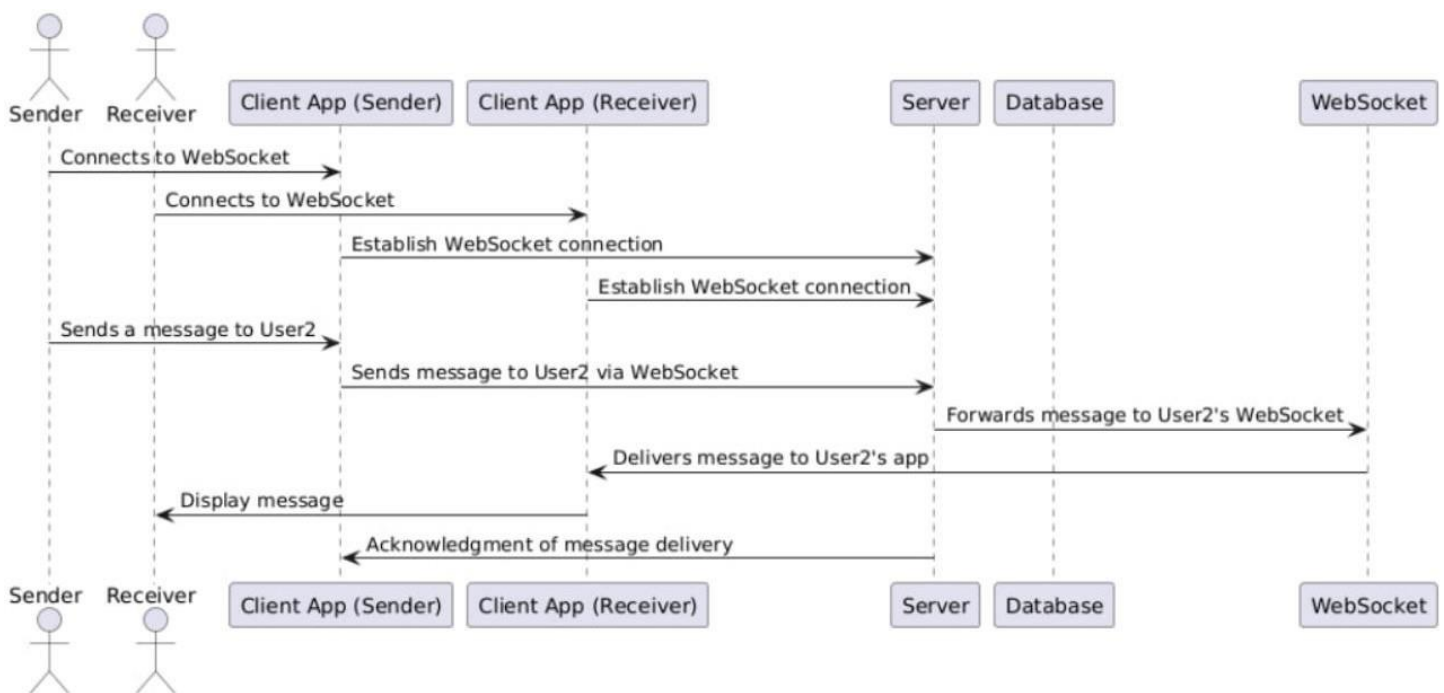
Основні елементи діаграми послідовностей:

- *Актори або об'єкти*, які взаємодіють між собою.
- *Повідомлення*, що передаються між ними.
- *Лінії життя*, які показують час існування об'єкта або актора під час взаємодії.

Діаграми послідовностей використовуються для візуалізації порядку виконання дій і

детальної реалізації алгоритмів, відображаючи, як операції переходять із одного стану в інший.

## Діаграма послідовностей

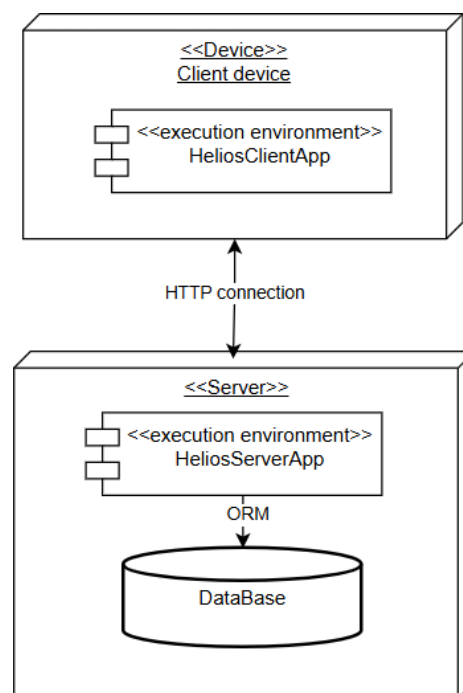


### Кроки процесу:

1. Підключення користувачів до веб-сокетів:
  - User1 (відправник) підключається до веб-сокетів через клієнтський додаток (Client1).
  - User2 (отримувач) також підключається до веб-сокетів через свій клієнтський додаток (Client2).
2. Створення з'єднання з сервером:
  - Клієнт User1 (Client1) встановлює з'єднання з сервером через веб-сокет.
  - Клієнт User2 (Client2) також встановлює з'єднання з сервером через веб-сокет.
3. Відправка повідомлення:
  - User1 через свій клієнт відправляє повідомлення до User2.
  - Client1 передає це повідомлення на сервер (Server) через веб-сокет.
4. Обробка та пересилання повідомлення сервером:

- Сервер отримує повідомлення від Client1 і передає його через веб-сокет на User2's WebSocket.
  - Веб-сокет на сервері пересилає повідомлення на клієнт User2 (Client2).
5. Отримання та відображення повідомлення:
- Client2 отримує повідомлення через веб-сокет і відображає його на екрані користувача User2.
6. Підтвердження доставки повідомлення:
- Сервер відправляє підтвердження доставки повідомлення назад до Client1, що повідомлення успішно доставлено.

### Діаграма розгортання

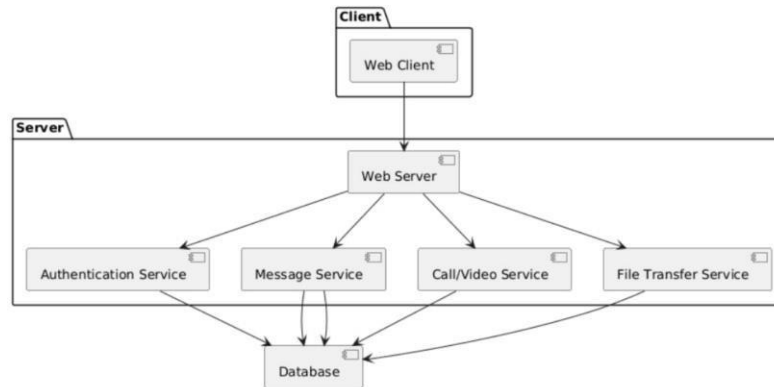


### Опис компонентів діаграми:

1. Client device (Пристрій клієнта):
  - Це фізичний пристрій, на якому виконується клієнтська частина додатку.
  - Він представлений як компонент типу Device. На ньому працює програмне середовище HeliosClientApp, що відповідає за виконання клієнтської логіки програми.
2. HTTP connection (HTTP-з'єднання):
  - Клієнтський додаток і сервер з'єднані через HTTP-протокол, що забезпечує передачу даних між пристроєм клієнта та сервером.
3. Server (Сервер):
  - Це серверна частина системи, що забезпечує обробку запитів від клієнта. Він представлений компонентом типу Server.
  - На сервері працює програмне середовище HeliosServerApp, яке обробляє логіку серверної частини.

- Сервер також взаємодіє з базою даних через ORM (Object-Relational Mapping) для збереження та отримання даних з бази.
4. Database (База даних):
- Це система зберігання даних, з якою сервер взаємодіє через ORM для виконання операцій з базою даних.

### Діаграма компонентів



1. Client (Клієнт):
  - Web Client — це клієнтська частина програми, яка працює через веб-браузер. Клієнт взаємодіє з сервером через HTTP-з'єднання, надсилаючи запити та отримуючи відповіді.
2. Server (Сервер):
  - Web Server — центральний сервер, який обробляє запити від клієнта. Він взаємодіє з кількома внутрішніми сервісами для забезпечення функціональності програми.

Сервер включає кілька специфічних сервісів:

3. Authentication Service (Сервіс автентифікації):
  - Цей сервіс відповідає за перевірку користувачів і надання доступу до програми після автентифікації. Клієнт запитує цей сервіс для перевірки логіну та пароля.
4. Message Service (Сервіс повідомлень):
  - Цей сервіс обробляє текстові повідомлення, які надсилаються від користувачів через веб-клієнт. Він відповідає за збереження та доставку повідомлень між користувачами.
5. Call/Video Service (Сервіс дзвінків/відео):
  - Цей сервіс відповідає за управління голосовими та відео дзвінками між користувачами, забезпечуючи функціональність конференц-зв'язку, якщо це необхідно.
6. File Transfer Service (Сервіс передачі файлів):
  - Цей сервіс дозволяє користувачам передавати файли через веб-клієнт. Він відповідає за завантаження, збереження та передачу файлів між

користувачами.

## 2. Database (База даних):

- Database — це центральна база даних, в якій зберігається інформація про користувачів, повідомлення, файли та інші дані, необхідні для функціонування сервісів.
- База даних взаємодіє з усіма сервісами для збереження або отримання необхідної інформації (наприклад, автентифікація користувачів, зберігання повідомлень тощо)

**Висновки:** У даній роботі було створено діаграму послідовностей, яка ілюструє основні процеси взаємодії користувача з додатком, діаграму розгортання, що описує фізичну архітектуру системи, та діаграму компонентів, яка демонструє модульну структуру додатку і взаємозв'язки між його частинами. Ці діаграми забезпечують візуалізацію архітектури та логіки роботи додатку для подальшої ефективної розробки та підтримки.