

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 «ТРПЗ»

Тема: Основи проектування.

Виконав:

Студент групи IA-34

Ковальчук Станіслав

Перевірив:

Мягкий М.Ю.

Київ — 2025

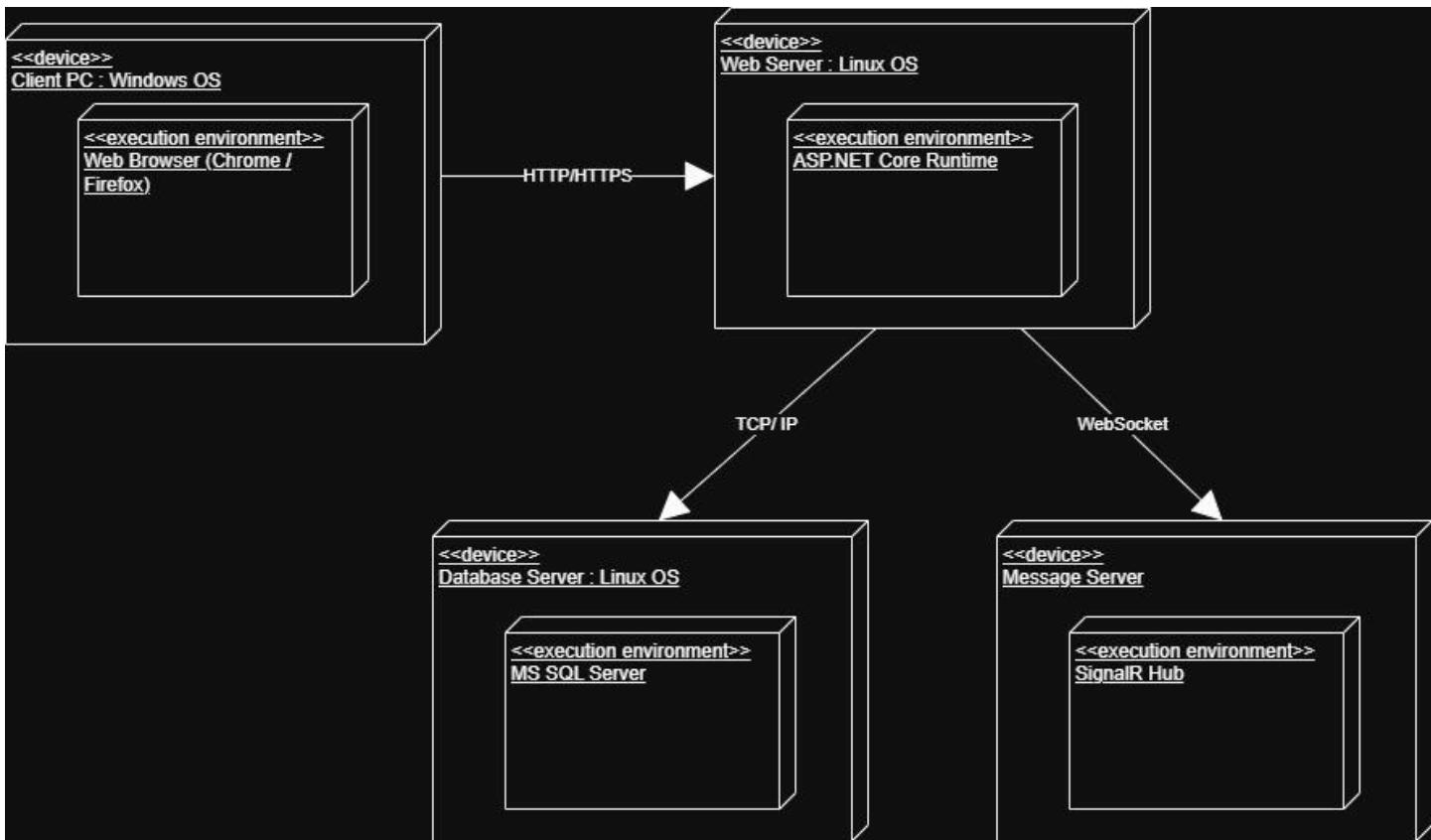
Тема: Основи проектування розгортання.

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.

Завдання

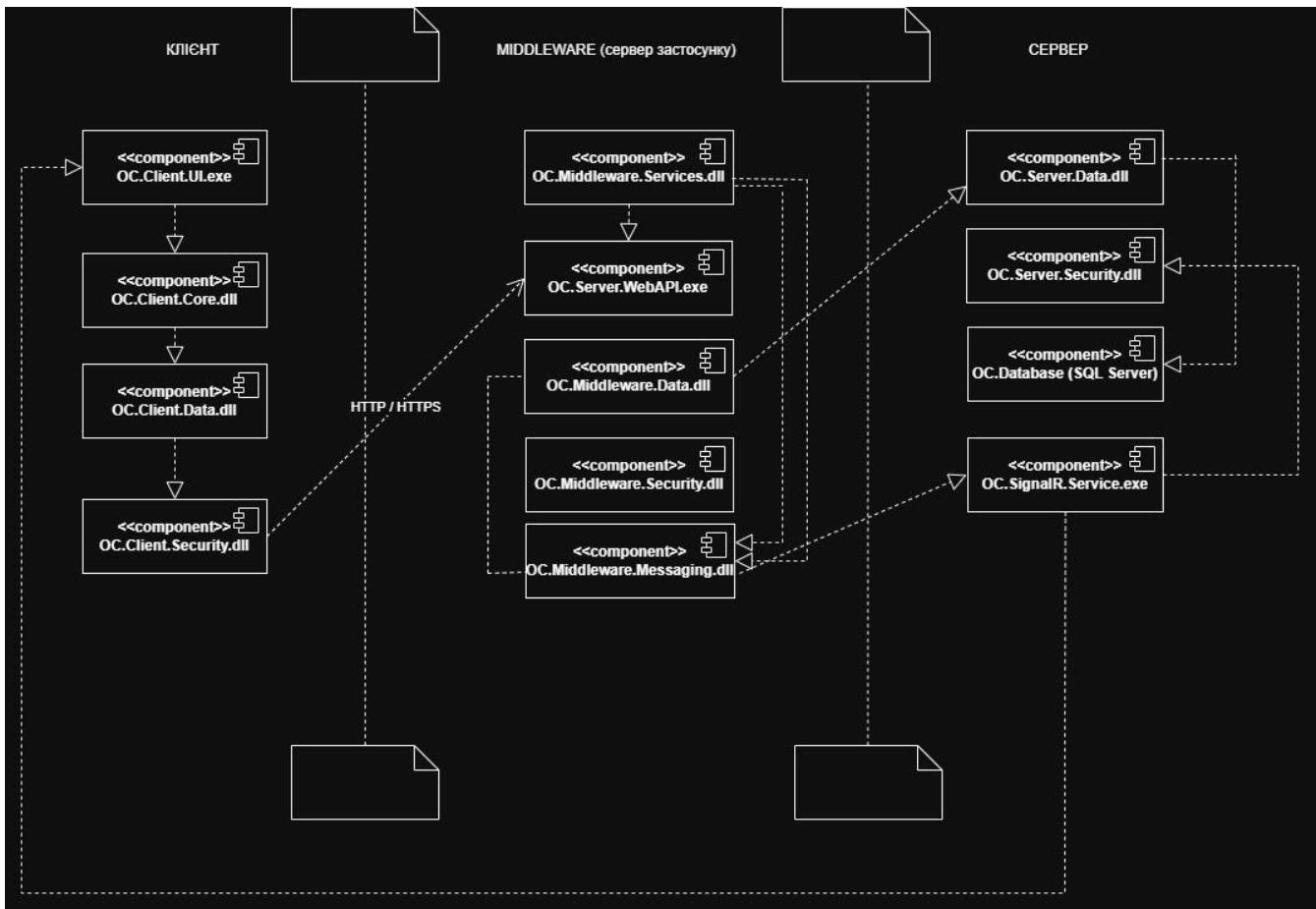
1. Ознайомитись з короткими теоретичними відомостями.
2. Проаналізувати діаграми створені в попередній лабораторній роботі а також тему системи та спроєктувати діаграму розгортання використання відповідно до обраної теми лабораторного циклу.
3. Розробити діаграму компонентів для проєктованої системи.
4. Розробити діаграму розгортання для проєктованої системи.
5. Розробити як мінімум дві діаграми послідовностей для сценаріїв прописаних в попередній лабораторній роботі.
6. На основі спроєктованих діаграм розгортання та компонентів доопрацювати програмну частину системи. Реалізація системи, додатково до попередньої реалізації, повинна містити як мінімум дві візуальні форми. В системі вже повинен бути повністю реалізована архітектура (повний цикл роботи з даними від вводу на формі до збереження їх в БД і подальшій виборці з БД та відображенням на UI).
7. Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен містити: діаграму розгортання з описом, діаграму компонентів системи з описом, діаграми послідовностей, а також вихідний код системи, який було додано в цій лабораторній роботі.

Діаграма розгортання:



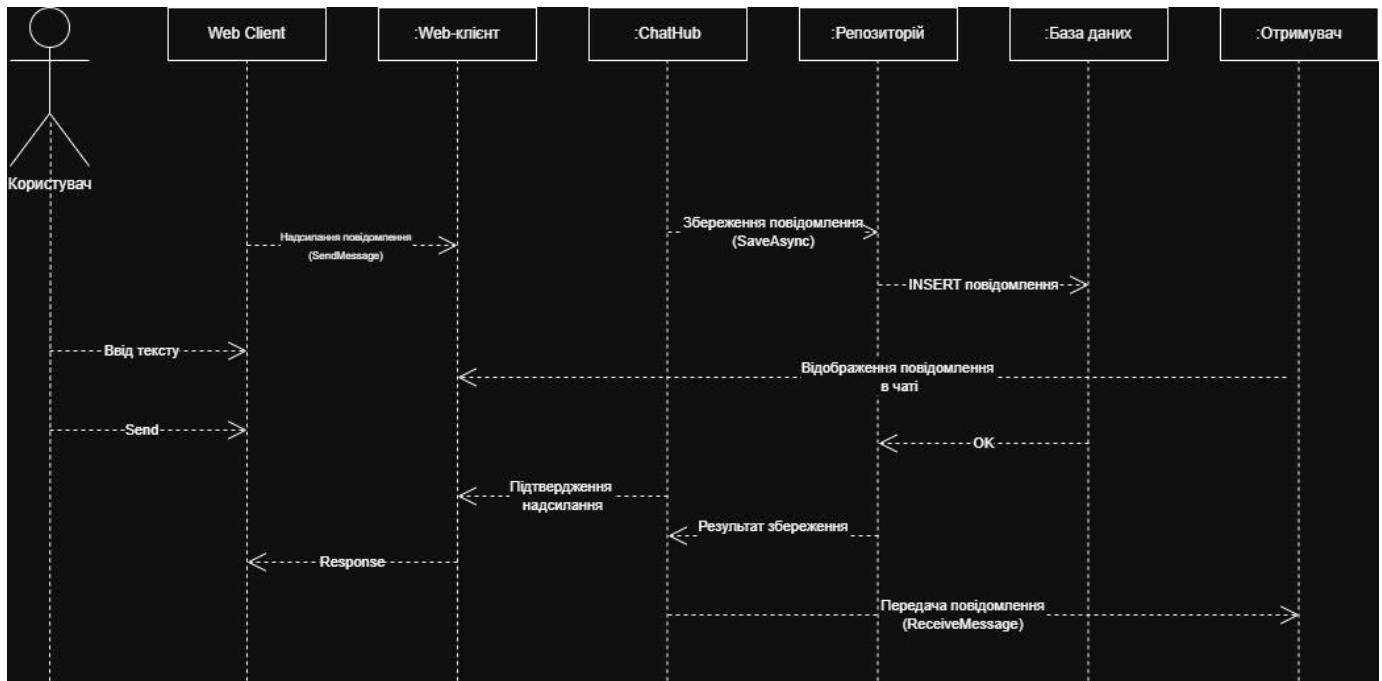
Опис: Діаграма розгортання відображає фізичну архітектуру системи Office Communicator. Користувачі працюють із системою через веб-браузер на клієнтських комп'ютерах під управлінням операційної системи Windows. Серверна частина застосунку розгорнута на веб-сервері під управлінням Linux з використанням платформи ASP.NET Core. Збереження даних здійснюється на окремому сервері бази даних MS SQL Server. Для доставки повідомлень у реальному часі використовується сервер повідомлень SignalR.

Діаграма компонентів:



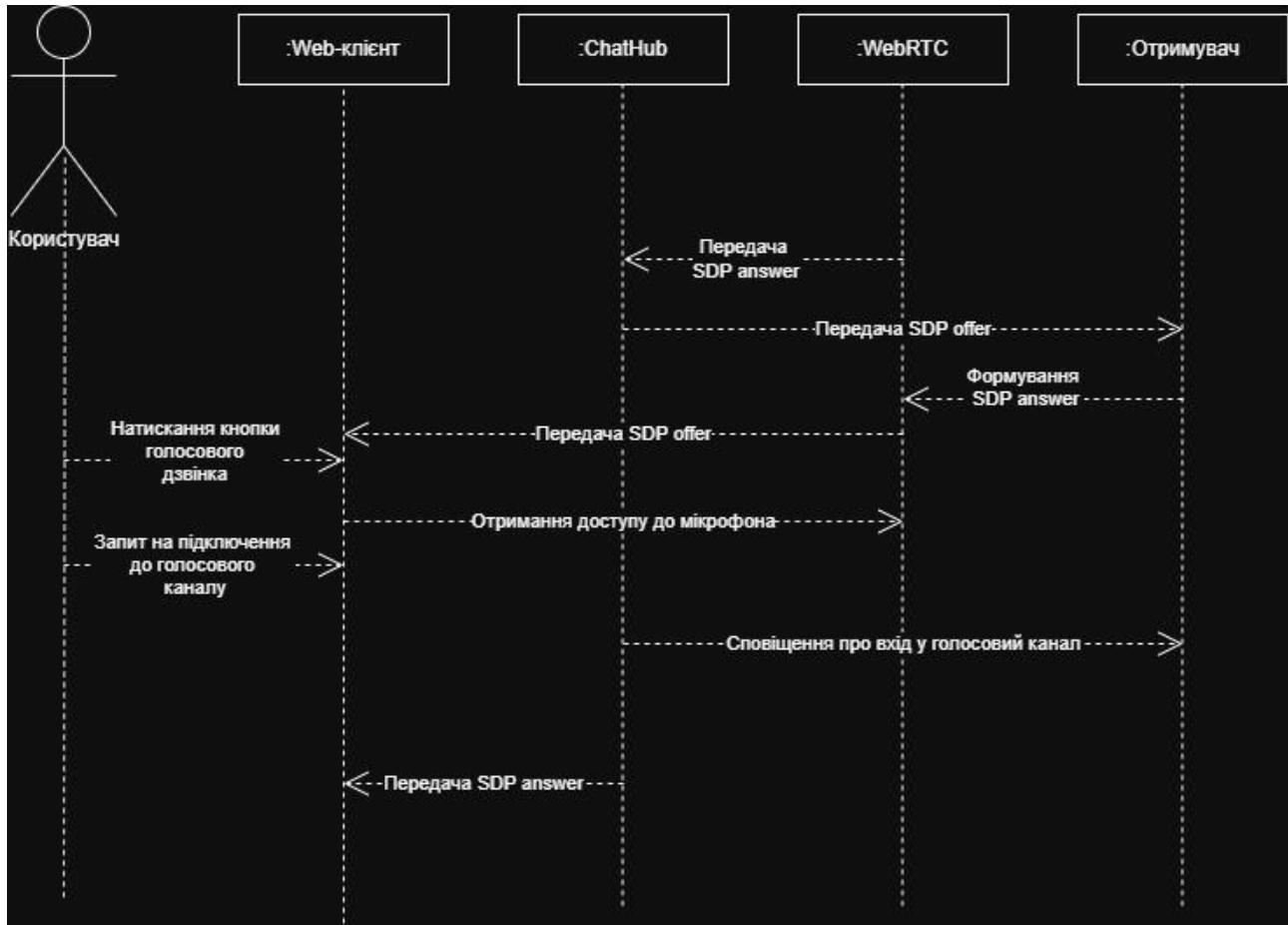
Діаграма компонентів відображає програмну архітектуру системи Office Communicator, яка складається з клієнтської частини, серверної логіки (middleware) та сервера зберігання даних. Клієнтська частина відповідає за взаємодію з користувачем, middleware забезпечує обробку бізнес-логіки та повідомлень, а серверна частина виконує зберігання та доставку даних. Такий поділ компонентів підвищує масштабованість та зручність супроводу системи.

Діаграма послідовностей(Надсилання повідомлення):



Дана діаграма послідовностей відображає процес надсилання текстового повідомлення користувачем у чаті системи Office Communicator. Процес починається з того, що користувач у веб-клієнті вводить текст повідомлення та ініціює його відправлення шляхом натискання кнопки Send. Після цього веб-клієнт формує запит на серверну частину системи та передає повідомлення до компонента ChatHub, який відповідає за обробку повідомлень у реальному часі. Отримавши повідомлення, ChatHub ініціює процес його збереження, звертаючись до шару доступу до даних — Репозиторію. Репозиторій виконує операцію збереження повідомлення в базі даних шляхом виконання SQL-операції типу INSERT. Після успішного збереження база даних повертає підтвердження виконання операції. Далі репозиторій повідомляє ChatHub про успішне завершення збереження, після чого сервер надсилає підтвердження веб-клієнту відправника. Паралельно ChatHub передає повідомлення клієнтській частині отримувача, де воно відображається у відповідному чаті. Таким чином, діаграма демонструє повний цикл обробки текстового повідомлення: від введення користувачем до збереження в базі даних та доставки іншому користувачу в режимі реального часу.

Діаграма послідовностей(Ініціація голосового дзвінка):



Дана діаграма послідовностей відображає процес ініціації голосового дзвінка між користувачами системи Office Communicator. Користувач ініціює дзвінок через веб-клієнт, після чого система звертається до серверного компонента ChatHub для організації голосового каналу. Обмін сигналами WebRTC (SDP offer та SDP answer) здійснюється через сервер, а безпосередня передача аудіоданих відбувається між клієнтами напряму, що забезпечує мінімальні затримки зв'язку.

Висновок: У ході виконання роботи було досягнуто поставленої мети: опановано підходи до проектування UML-діаграм розгортання та компонентів для системи, що проєктується, а також розроблено діаграми взаємодії — зокрема діаграми послідовностей — на основі сценаріїв, сформованих у попередній лабораторній роботі. У результаті визначено склад програмних компонентів і їхні залежності, окреслено цільове середовище

розміщення системи та відображену логіку обміну повідомленнями між учасниками процесів у ключових сценаріях. Отримані діаграми уточнюють архітектурні рішення, забезпечують узгодженість вимог і реалізації та слугують основою для подальшої розробки й тестування системи.

Контрольні запитання:

1. Що собою становить діаграма розгортання?

Діаграма розгортання (Deployment Diagram) відображає фізичне розташування системи: на якому обладнанні та в яких середовищах виконання розгортаються складові ПЗ, а також зв'язки (інформаційні шляхи) між вузлами.

2. Які бувають види вузлів на діаграмі розгортання?

Вузли бувають двох типів:

- Пристрій (device) — фізичне обладнання (ПК, сервер, інший пристрій, пов'язаний із системою).
- Середовище виконання (execution environment) — програмне середовище, що може містити інше ПЗ (ОС, процес-контейнер, вебсервер тощо).

3. Які бувають зв'язки на діаграмі розгортання?

Зв'язки між вузлами (інформаційні шляхи) зазвичай зображаються прямою лінією та можуть мати:

- множинність (наприклад, кілька клієнтів підключені до одного сервера);
- назву, яка часто містить протокол або технологію взаємодії (HTTP, IPC, WCF, .NET Remoting тощо).

4. Які елементи присутні на діаграмі компонентів?

На діаграмі компонентів відображають:

- компоненти (модулі) системи;
- залежності між компонентами;

- у разі виконуваного поділу — компоненти як файли/одиниці поставки (.exe, .dll, вихідні файли, HTML-сторінки, БД і таблиці тощо).

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

Зв'язки (залежності) показують, що класи/функціональність одного компонента використовують класи/функціональність іншого компонента, а також допомагають оцінити, як зміни в одному компоненті впливають на інші.

6. Які бувають види діаграм взаємодії?

У UML до діаграм взаємодії належать:

- Діаграма послідовностей (Sequence Diagram);
- Діаграма комунікації/співпраці (Communication/Collaboration Diagram);
- Оглядова діаграма взаємодії (Interaction Overview Diagram);
- Часова діаграма (Timing Diagram).

7. Для чого призначена діаграма послідовностей?

Діаграма послідовностей призначена для моделювання взаємодії між об'єктами в часі: вона показує порядок обміну повідомленнями, логіку виконання операцій і послідовність викликів у межах сценарію.

8. Які ключові елементи можуть бути на діаграмі послідовностей?

Основні елементи:

- Актори (Actors) — користувачі або зовнішні системи;
- Об'єкти/класи (учасники взаємодії) та їх лінії життя;
- Повідомлення (виклики методів/передача даних, синхронні й асинхронні, та повернення результату пунктиром);
- Активності (активації) — проміжки виконання дій об'єктом;
- Контрольні структури (фрагменти) для умов/циклів/альтернатив: alt, loop тощо.

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

Діаграма варіантів використання (Use Case) описує що має робити система (функціональні вимоги), а діаграма послідовностей деталізує як саме виконується конкретний сценарій цього варіанта використання: які учасники залучені та які повідомлення вони передають у часі. Зазвичай для одного Use Case роблять окремі діаграми послідовностей для основного та альтернативних потоків.

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

Діаграма класів задає структуру (класи, їх методи та зв'язки), а діаграма послідовностей показує поведінку (виклики між об'єктами). Типово:

- учасник на послідовностях є екземпляром певного класу;
- повідомлення на діаграмі послідовностей відповідають методам/операціям класів;
- якщо на послідовностях з'являються виклики, яких немає на діаграмі класів, це означає потребу уточнити методи/зв'язки на діаграмі класів.