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 РОЗГОРТАННЯ. ДІАГРАМА КОМПОНЕНТІВ. ДІАГРАМА ВЗАЄМОДІЙ ТА ПОСЛІДОВНОСТЕЙ»

Виконала:

студентка групи ІА-23

Єрмак Д. Р

Перевірив:

Мягкий М. Ю.

Київ 2024

Тема лабораторних робіт:

IRC client (singleton, builder, abstract factory, template method, composite, client-server)

Клієнт для IRC-чатів з можливістю вказівки порту і адреси з'єднання, підтримка базових команд (підключення до чату, створення чату, установка імені, реєстрація, допомога і т.д.), отримання метаданих про канал.

Завдання:

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.

Зміст

Хід роботи:

Крок 1. Створення діаграми розгортання для проектованої системи

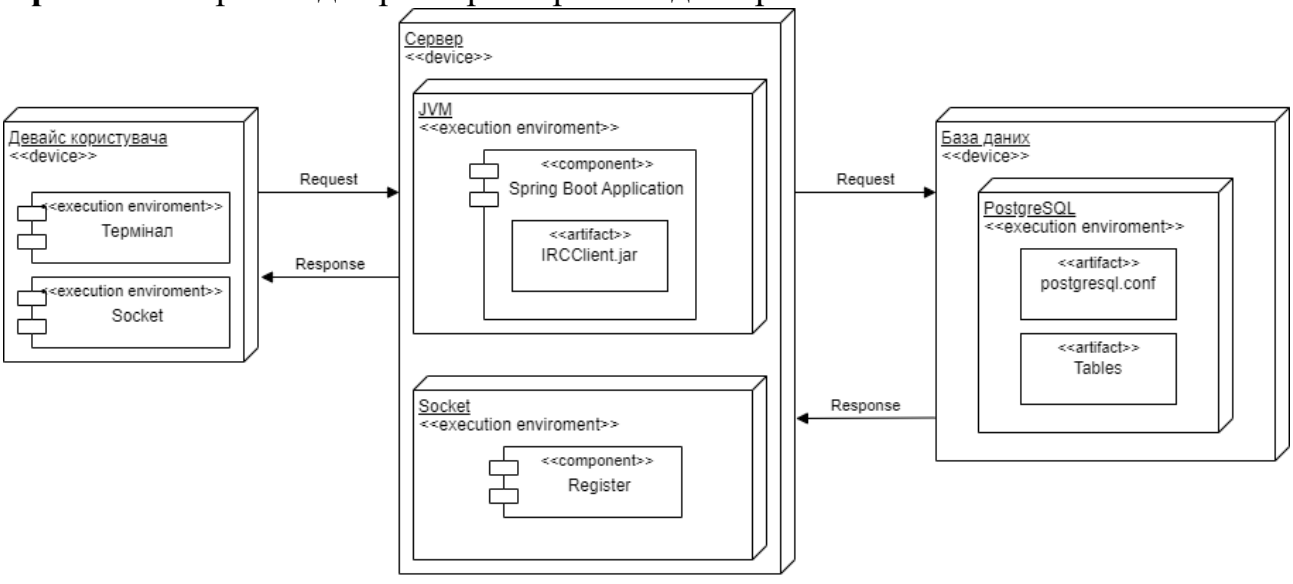

Рис. 1 – Діаграма розгортання

Ця діаграма розгортання описує архітектуру системи, що складається з трьох основних компонентів: пристрою користувача, сервера та бази даних.

Пристрій користувача

Пристрій користувача <<device>> містить два середовища виконання (<<execution environment>>): Термінал і Socket.

- Термінал відповідає за ініціювання запитів до сервера.

- Socket використовується для підтримки з'єднання з сервером.

Сервер

- Сервер <<device>> включає кілька середовищ виконання:

- JVM (Java Virtual Machine) — середовище виконання, в якому розміщено:

- Spring Boot Application<<component>> — додаток, розроблений за допомогою Spring Boot.

- Артефакт IRCClient.jar<<artifact>>, який представляє собою згенерований файл програми.

- Socket — середовище виконання, яке включає компонент Register <<component>>, що підтримує функціональність зберігання інформації про підключення.

База даних

- База даних<<device>> має середовище виконання PostgreSQL, яке містить:

- Артефакт postgresql.conf <<artifact>>, що є конфігураційним файлом PostgreSQL.

- Артефакт Tables <<artifact>>, що представляє структуру таблиць бази даних для зберігання необхідної інформації.

Взаємодія між компонентами

- Request — запит, що відправляється з пристрою користувача на сервер.

- Response — відповідь, що повертається сервером на пристрій користувача.

- Сервер також надсилає запити до бази даних, отримуючи від неї відповіді.

Діаграма відображає архітектуру системи, показуючи взаємодію між клієнтом,

сервером і базою даних через запити і відповіді, що проходять через різні середовища виконання.

Крок 2. Створення діаграми компонентів для проектованої системи.

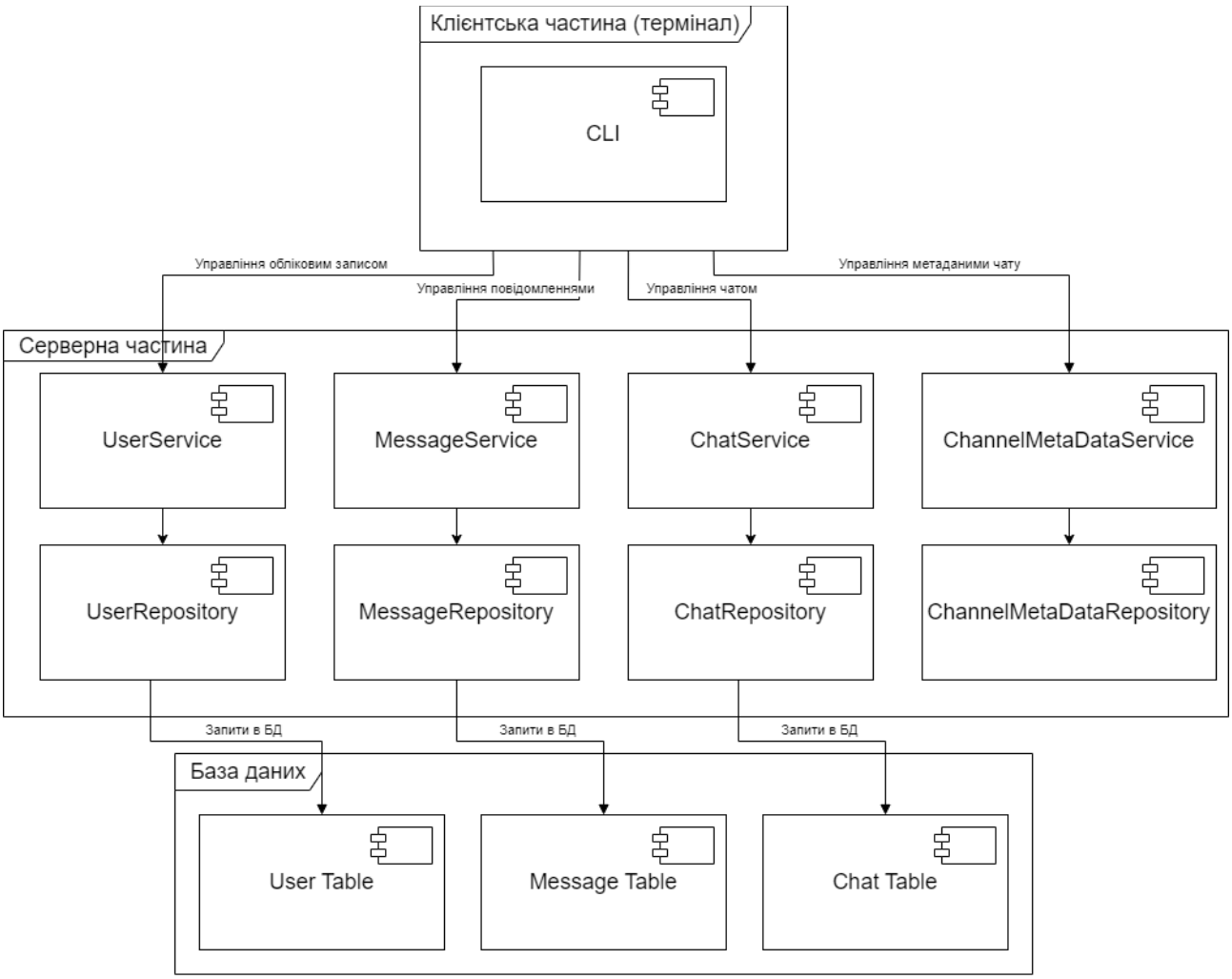

Рис. 2 – Діаграма компонентів

Діаграма компонентів описує архітектуру системи, що складається з трьох основних частин: клієнтської частини, серверної частини та бази даних.

Клієнтська частина

- Терміналом (CLI - Command Line Interface), що є інтерфейсом користувача для взаємодії із системою.

- Клієнтська частина підтримує управління обліковим записом, повідомленнями та метаданими чату, надсилаючи відповідні запити до

серверної частини.

Серверна частина складається з таких компонентів:

- UserService — компонент, що відповідає за управління обліковими записами користувачів.
  - Взаємодіє з UserRepository, який обробляє запити до таблиці користувачів у базі даних.
- MessageService— компонент для обробки повідомлень.
  - Працює разом з MessageRepository, який зберігає та отримує дані з таблиці повідомлень у базі даних.
- ChatService — відповідає за управління чатами, включаючи створення та підключення до чату.
  - Використовує ChatRepository для доступу до даних у таблиці чатів у базі даних.
- ChannelMetaDataService — компонент для управління метаданими каналів.
  - Підключений до ChannelMetaDataRepository, який працює з метаданими каналів.

База даних складається з трьох таблиць:

- User Table — таблиця для зберігання інформації про користувачів.
- Message Table — таблиця для зберігання повідомлень, що надсилаються між користувачами.
- Chat Table — таблиця для зберігання даних про чати

Взаємодія між компонентами

- Клієнтська частина надсилає запити на сервер для управління обліковими записами, чатами, повідомленнями та метаданими чату.

- Серверна частина обробляє ці запити через відповідні сервіси, які взаємодіють зі сховищами для отримання або зберігання даних у базі.

- База даних відповідає на запити серверної частини, забезпечуючи постійне зберігання та доступ до інформації.

Діаграма відображає розподіл функціональності між клієнтською частиною, серверною частиною і базою даних, показуючи чітку структуру взаємодії між компонентами.

Крок 3. Створення діаграми послідовностей для проектованої системи

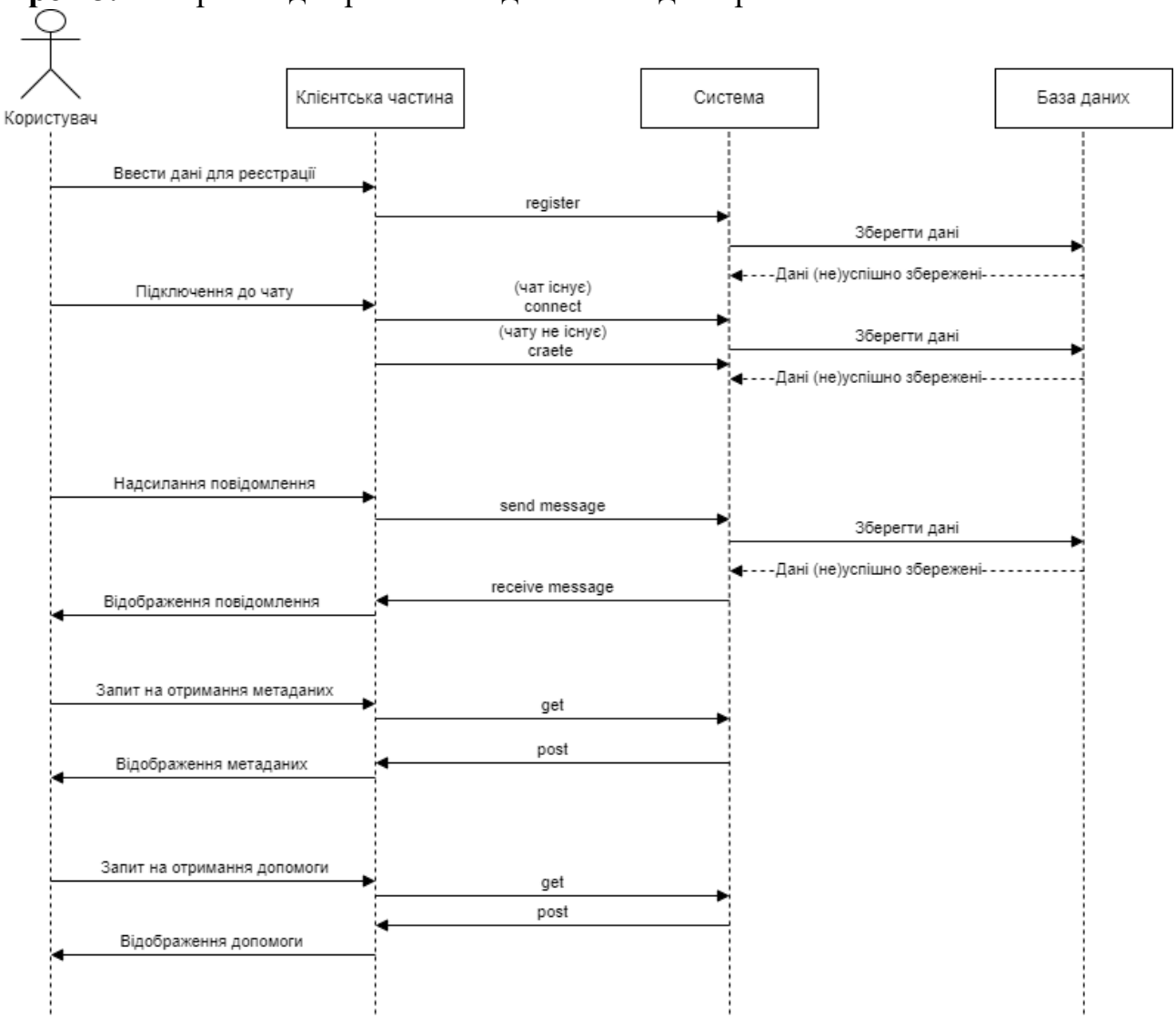

Рис. 3 – Діаграма послідовностей

На діаграмі послідовностей зображено процеси взаємодії між користувачем, клієнтською частиною, системою та базою даних під час виконання різних дій.

Користувач вводить дані для реєстрації у клієнтській частині, яка надсилає запит на реєстрацію до системи. Система передає запит на збереження даних у базу даних, і база даних повертає відповідь про успішне або неуспішне збереження.

Для підключення до чату користувач надсилає запит через клієнтську частину. Клієнтська частина перевіряє, чи існує чат. Якщо чат існує, відбувається підключення, якщо ні — створення нового чату. Система зберігає інформацію у базі даних, отримуючи від неї підтвердження або відмову.

Під час надсилання повідомлення користувач вводить текст у клієнтській частині, яка відправляє запит на надсилання повідомлення до системи. Система зберігає повідомлення в базі даних, яка відповідає результатом операції.

Отримання повідомлення відбувається після надсилання: система надсилає запит до клієнтської частини, яка відображає повідомлення для користувача.

Для запиту метаданих користувач звертається до клієнтської частини, яка надсилає запит на отримання інформації до системи. Система відповідає, передаючи метадані до клієнтської частини для їх відображення.

Для отримання допомоги користувач надсилає запит через клієнтську частину, яка звертається до системи. Система передає інформацію назад у клієнтську частину для відображення допомоги користувачу.

Діаграма показує послідовність дій під час реєстрації, підключення до чату, обміну повідомленнями, запиту метаданих та отримання допомоги.

Висновок: У ході виконання даної лабораторної роботи було виконано розроблення та моделювання архітектури системи. Спочатку було проведено аналіз теоретичних відомостей. Далі було розроблено діаграми розгортання, компонентів та послідовностей для забезпечення легшої реалізації проєкту.