

к.т.н. Барабан М. В.

к.т.н. Бевз О. М.

студент Довгалець К. С.

студент Щишов О. С.

Вінницький національний технічний університет

Розробка програмного забезпечення для миттєвого обміну повідомленнями між декількома користувачами

Актуальність задачі

У сучасному світі все більше людей користуються системами миттєвої відправки повідомлень. Це є зручною альтернативою листам, які писали декілька століть тому, чи телефонним дзвінкам, чи листам на електронну пошту. Такі системи обміну інформацією пришвидшують наше спілкування, вирішення життєвих проблем чи обмін цінною інформацією, фотографіями, відео. Із кожним днем все більше людей починають користуватись програмами обміну повідомленнями.

Опис підходу до вирішення задачі

Ідея написання програмного забезпечення для обміну повідомленнями не є новою чи надто сучасною. Вона має декілька основних рішень. Одне із найбільш поширених – це використання бази даних для зберігання повідомлень та використання сучасних фреймворків для реєстрації та авторизації. Також використання WebSocket для миттєвого обміну інформацією та використання технології Rest API для надсилання запитів про реєстрацію, авторизацію, надсилання, чи отримання повідомлень. При розробці програмного забезпечення для миттєвого обміну повідомленнями постає питання як в кожен момент часу відправляти та отримувати повідомлення та відображати їх без перезавантаження сторінки. Звісно є багато реалізацій цієї проблеми. Один із них є WebSocket, який дозволяє дуже гнучко та без зайвих зусиль вирішити

проблему. Архітектурний стиль Rest дозволяє легко отримувати та обробляти HTTP запити, що полегшує реєстрацію, авторизацію та обмін повідомленнями. Також написання графічного інтерфейсу за допомогою технології JSP мовою Java Script.

Етапи розробки та використані технології

При розробці даного WEB-додатку необхідно створити реляційну базу даних, яка міститиме дві таблиці: користувачі та повідомлення. Таблиці повинні бути зв'язані між собою зовнішніми ключами. У реалізації даного програмного забезпечення було використано базу даних PostgreSQL.

Обмін повідомленнями між користувачами реалізується за допомогою фреймворку Spring MVC, який забезпечує реалізацію шаблону Model-View-Controller (MVC). Схема шаблону представлена на Рисунку 1.

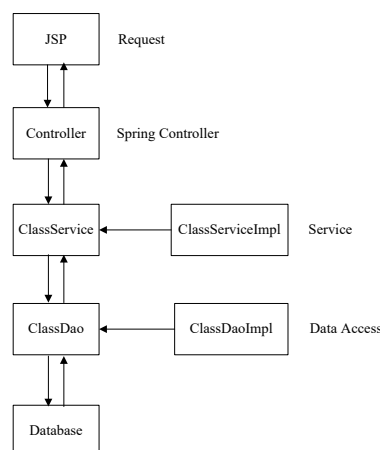


Рисунок 1 – Схема шаблону MVC

Модель MVC полягає у реалізації декількох рівнів доступу до даних. На рівні Controller приймаються HTTP запити та робиться найпростіша перевірка на правильність даних. Далі дані передаються у Service де обробляються у необхідний формат, якщо це необхідно, та передаються на рівень Dao, де реалізуються запити до бази даних через інструменти бібліотеки JdbcTemplate, яка є частиною Spring. Дані, які отримуються із бази даних передаються від Dao до Service і до Controller, звідки надсилаються у файли JSP та відображаються на WEB-сторінці.

Для реалізації реєстрації та авторизації необхідно налаштувати Spring

Security фреймворк. Для цього у папку із конфігурацією проекту додається клас `WebSecurityConfig.java` де описуються необхідні конфігурації та кодування паролей. Також необхідно реалізувати графічне відображення поля реєстрації та авторизації. Із використанням технології JSP це реалізується просто, оскільки розмітка схожа на HTML і в неї передаються введені користувачем дані через спеціальні змінні.

Для отримання повідомлень у кожен момент часу необхідно налаштувати `WebSocket`, створивши клас `WebSocketConfig` у папці із конфігурацією та написати там необхідний конфігураційний код. Також необхідно реалізувати слухача, який буде спрацьовувати на підключенні чи відключенні `WebSocket`. Реалізація миттєвого отримання повідомлень виконана мовою програмування JavaScript. Повинні бути реалізовані методи підключення сокета, отримання та обробки повідомлень, передачі повідомлення та користувача серверу для збереження у базі та відображення повідомлення на сторінці. Також, якщо користувач ввійшов у свій профіль, необхідно завантажувати його попередні повідомлення, зробивши REST запит до сервера, який повертає дані у форматі JSON. Також необхідно створити файл із властивостями всього додатку, де необхідно вказати порт, підключення до бази та деякі налаштування файлів, які відносяться до відображення на WEB-сторінці. Файл необхідно помістити у папку ресурсів проекту та назвати `application.properties`.

Література

- [1] Уомс К. Spring в действии. М., Россия: Форзац, 2015.
- [2] Брюс Е. Філософія Java. СПб., Россия: Питер, 2016.
- [3] Бьюи А. Изучаем SQL. СПб., Россия: Сивол-Плюс, 2007.
- [4] [Електронний ресурс]: Режим доступу: <https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>. Дата звернення: Вересень 03, 2019.
- [5] [Електронний ресурс]: Режим доступу: <https://www.baeldung.com/java-websockets>. Дата звернення: Серпень 29, 2019.