Василь Васильчук, Олександр Овсяк*
[1] НЛТУ України, Львів, Україна, ORCID: ID: 0009-0003-1592-2411,
122.24vasylchuk.v@nltu.lviv.ua
[2] НЛТУ України, Львів, Україна. ORCID: ID: 0000-0003-2620-1938,
ovsjak@nltu.edu.ua

Автоматизована система керування сервісом оренди автомобілів засобами Java Spring Boot

Анотація. *Представлено процес розроблення автоматизованої системи керування сервісом оренди автомобілів, створеної із застосуванням Java Spring Boot. Система призначена для автоматизації ключових бізнес-процесів компанії - управління автопарком, бронювання, оплат, комунікації з клієнтами та формування звітів. Реалізовано багаторівневу систему автентифікації користувачів із розмежуванням ролей доступу, а також інтеграцію з платіжним шлюзом Stripe для безпечних онлайн-оплат. Для зручності користувачів створено WhatsApp-бот, який забезпечує обмін повідомленнями, нагадуваннями та сповіщеннями про стан замовлень.*

Використання Spring Security, Liquibase і MySQL гарантує стабільність, безпеку та масштабованість системи. Розроблена система може бути адаптована до потреб реальних компаній у сфері автопрокату, забезпечуючи ефективність управління й підвищення рівня обслуговування клієнтів.

Ключові слова *– автоматизація, оренда автомобілів, Java Spring Boot, WhatsApp-бот, Stripe, інформаційна система, Liquibase, MySQL, Spring Security.*

Вступ. Сфера оренди автомобілів динамічно розвивається завдяки зростанню попиту на послуги короткострокового транспортування. У таких умовах ефективність бізнесу безпосередньо залежить від рівня автоматизації процесів: управління автопарком, бронювання, обробки оплат і комунікації з клієнтами. Багато малих компаній досі використовують неінтегровані системи (Excel, ручні таблиці), що ускладнює роботу та збільшує ризики помилок [1].

Застосування вебтехнологій та фреймворків, таких як Spring Boot і React, дозволяє створити масштабовані системи, які поєднують бекенд- і фронтенд-частини, забезпечуючи сучасний рівень продуктивності та безпеки.

* науковий керівник

Математична модель. Розроблена система оренди автомобілів описується як динамічна модель взаємодії користувачів, ресурсів та транзакцій, де основними сутностями є множини:

$$U = \{u_1, u_2, \ldots, u_n\} \text{ — користувачі,}$$
$$C = \{c_1, c_2, \ldots, c_m\} \text{ — автомобілі,} \quad (1)$$
$$B = \{b_1, b_2, \ldots, b_k\} \text{ — бронювання.}$$

Для кожного користувача u_i визначається функція ролі:

$$r(u_i) \in \{\text{admin, manager, client}\}, \quad (2)$$

що задає рівень доступу в системі.

Функція бронювання описується як:

$$b_j = f(u_i, c_p, t_s, t_e, s), \quad (3)$$

де t_s і t_e - час початку та завершення оренди, а s - статус бронювання (активне, підтверджене, завершене).

Система платежів моделюється відображенням:

$$P : B \to \{0, 1\}, \quad (4)$$

де 1 - підтверджена оплата через Stripe, 0 — очікує виконання.

Інформаційна взаємодія з клієнтами визначається функцією повідомлень:

$$N : B \to M, \quad (5)$$

де M - множина повідомлень, що надсилаються користувачу через WhatsApp API відповідно до зміни статусу бронювання.

Таким чином, загальна модель функціонування системи описується кортежем:

$$S = \langle U, C, B, P, N, R \rangle, \quad (6)$$

де R - набір бізнес-правил, що визначають логіку переходів між станами об’єктів.

У цій моделі забезпечується цілісність даних і виключення конфліктів бронювань, оскільки для будь-яких $b_i, b_j \in B$ виконується виконується умова:

$$[t_s(b_i), t_e(b_i)] \cap [t_s(b_j), t_e(b_j)] = \emptyset, \text{ якщо } c(b_i) = c(b_j). \quad (7)$$

Це означає, що один автомобіль не може бути заброньований одночасно кількома клієнтами.

Запропонована модель дозволяє формалізувати основні процеси системи: реєстрацію користувачів, бронювання, обробку платежів та сповіщення, що створює основу для подальшої програмної реалізації.

* науковий керівник

Програмна реалізація. Програмну реалізацію системи виконано мовою Java з використанням фреймворку Spring Boot, який забезпечує швидку конфігурацію, масштабованість і інтеграцію з іншими технологіями [2].

Серверна частина включає модулі:

- Spring Data JPA — для роботи з базою даних MySQL;
- Spring Security — для автентифікації користувачів і контролю доступу;
- Stripe API — для здійснення онлайн-платежів;
- WhatsApp API — для сповіщення користувачів про зміни у статусах бронювання.

Клієнтська частина реалізована на React.js. Вона дозволяє користувачам створювати облікові записи, переглядати автопарк, здійснювати бронювання та оплату послуг. Комунікація з сервером відбувається через REST API із передачею даних у форматі JSON.

Для забезпечення безпеки використано JWT (JSON Web Token) — технологію, що дозволяє автентифікувати користувачів без зберігання сесій на сервері. Паролі користувачів шифруються алгоритмом BCrypt, що гарантує їх надійний захист.

База даних MySQL містить таблиці для користувачів, автомобілів, бронювань, оплат і сповіщень. Її структура контролюється за допомогою Liquibase, який зберігає всі зміни схеми у вигляді XML- або YAML-файлів, що забезпечує контроль версій та автоматичну синхронізацію при оновленнях системи[3].

Інтерфейс користувача має адаптивний дизайн, який забезпечує комфортне користування як на комп’ютерах, так і на мобільних пристроях. Реалізовано модуль адміністратора для моніторингу замовлень і звітності, що дає змогу оперативно контролювати роботу сервісу.

Під час тестування система показала стабільну роботу при одночасному використанні кількома десятками користувачів, що підтверджує її масштабованість і ефективність.

Список використаних літературних джерел

* науковий керівник

[1] Романів Є., Війтович М. (2023). Сучасний стан та напрямки удосконалення інформаційних систем обліку. Молодий вчений, 10(122), 229–232.
[2] Walls C. Spring in Action, 6th Edition. Manning Publications, 2022.
[3] Сміт А. Основи оброблення даних у базах даних. Л., 2018.

Automated car rental service management system using Java Spring Boot: analytics

Vasyl Vasylchuk, Oleksandr Ovsyak

A description presents the development of an automated car rental management system using Java Spring Boot. The system automates major business processes such as car management, booking, payment, and customer communication. It integrates Stripe API for online payments and WhatsApp Business API for customer notifications. The architecture follows a multi-layer client–server model, ensuring modularity and scalability. Spring Security provides reliable user authentication via JWT, while Liquibase manages database versioning in MySQL. Testing confirmed system stability, security, and performance, making it suitable for real-world deployment in rental companies.

* науковий керівник