

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЗАМОВЛЕННЯМИ У ЗАКЛАДАХ ХАРЧУВАННЯ З КРОСПЛАТФОРМЕННИМ ДОСТУПОМ

Корчев Є.А.¹, Божуха Л.М.²

¹ДНУ імені Олеся Гончара, студент, Україна

²ДНУ імені Олеся Гончара, кандидат фізико-математичних наук, доцент, Україна

Анотація. У роботі розглянуто проєктування та реалізацію інформаційної системи управління замовленнями для закладів харчування, орієнтованої на підвищення ефективності обслуговування, зручність користування та дотримання вимог інформаційної безпеки. Запропоноване рішення забезпечує взаємодію користувачів із різними рівнями доступу, підтримує обробку замовлень, бронювань і пов'язаних бізнес-процесів, а також дозволяє працювати з різних типів пристроїв у єдиному середовищі. Особливу увагу приділено побудові безпечної архітектури доступу, уніфікації каналів взаємодії, валідації даних та адаптивності інтерфейсів. Розроблена система демонструє гнучке рішення, що може бути інтегроване для будь-якого сучасного закладу харчування та адаптоване до майбутніх функціональних і технологічних змін. Архітектура системи підтримує масштабування, майбутню інтеграцію нових модулів і забезпечує високий рівень зручності для користувачів незалежно від платформи чи пристрою.

Ключові слова: інформаційна система, FastAPI, React, Android, клієнт-серверна архітектура, кросплатформенність, API, фреймворк.

Вступ. У сучасних умовах сфера обслуговування має тенденцію стрімкого зростання попиту на автоматизовані рішення, які здатні оперативно обробляти замовлення та надавати доступ із різних пристроїв для користувачів різних рівнів доступу. Для закладів харчування особливо важливо забезпечити безперебійне управління процесами, водночас дотримуючись вимог захисту даних та контролю доступу. Кросплатформенні технології, що дають змогу працювати через веб-інтерфейси й мобільні додатки, відкривають широкі можливості для створення інтелектуальних систем керування, орієнтованих на ефективність, безпеку й зручність користувачів. У цій роботі розглянуто проєктування подібної системи із застосуванням сучасних клієнт-серверних технологій.

Основний матеріал. Інформаційна система управління замовленнями розроблена за принципом клієнт-серверної архітектури. Основу серверної частини складає REST API, реалізований з використанням фреймворку FastAPI, який забезпечує взаємодію з базою даних та клієнтськими застосунками [1]. Система підтримує асинхронну обробку запитів завдяки використанню ASGI. Такий підхід здатен забезпечити масштабоване та максимально ефективне рішення для роботи з великою кількістю одночасних користувачів. На рис. 1 наведено загальну архітектуру системи.

Клієнтська частина включає веб-інтерфейс, реалізований з використанням React разом з додатковими бібліотеками для навігації та інтерфейсної взаємодії, а також мобільні застосунки для Android, створені з використанням мови програмування Kotlin. Вебзастосунок забезпечує багатосторінкову навігацію завдяки React Router, що дозволяє ефективно розділяти логіку інтерфейсу та підвищує зручність користування. Таке рішення дозволяє реалізувати динамічну маршрутизацію на стороні клієнта, що дає змогу створити односторінковий застосунок (SPA) із навігацією без перезавантаження сторінки. Компоненти інтерфейсу відображаються згідно зі структурою маршрутів, що дозволяє будувати гнучкий та масштабований UI.

Android додатки орієнтовані на клієнтів та персонал закладу, і забезпечують доступ до основних функцій системи шляхом інтеграції з розробленим API. Завдяки обраній архітектурі вдалося досягти того, що два окремих застосунки на базі різних технологій забезпечують однакову функціональність із вебверсією. Користувачі можуть оформлювати замовлення, переглядати їх статус, залишати відгуки, бронювати столи, використовувати бонуси та взаємодіяти з іншими компонентами системи – незалежно від того, чи вони працюють у браузері, чи через мобільний застосунок. Обидві клієнтські платформи взаємодіють із сервером через HTTP-запити до REST API. Обмін даними відбувається у форматі JSON, із застосуванням токен-авторизації. Такий підхід забезпечує кросплатформенність та уніфіковану логіку доступу, незалежно від типу пристрою користувача [2].

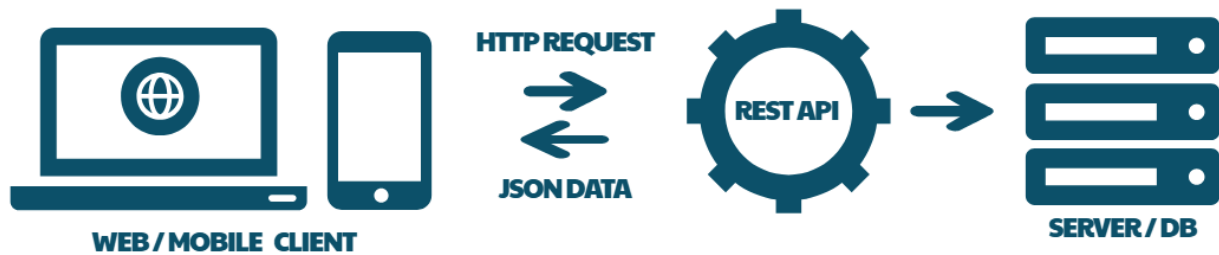


Рисунок 1 – Архітектура системи

Оскільки система працює з персональними даними користувачів та замовленнями, одним із ключових аспектів її розробки стало впровадження надійної моделі доступу до ресурсів. Для цього реалізовано рольову модель авторизації, яка включає чотири основні типи користувачів:

- Гість має лише ознайомчий доступ,
- Клієнт може оформлювати замовлення, використовувати бонуси, переглядати історію, залишати відгуки, бронювати столики,
- Офіціант обробляє замовлення, змінює їх статуси, взаємодіє зі столами,
- Адміністратор має повний контроль над даними, включно з управлінням користувачами, меню, тощо.

Доступ до ресурсів обмежується на рівні серверної частини за допомогою JWT-токенів, які видаються після успішної авторизації. Кожен запит до захищених маршрутів перевіряється на наявність, коректність та надійність токена, а також на відповідальність ролі користувача до дозволених дій. Такий підхід дозволяє уникнути несанкціонованого доступу до ресурсів, та уникнути зберігання сесій на сервері, що значно спрощує масштабування системи [3].

Для додаткового захисту використано перевірку даних за допомогою моделей Pydantic, що забезпечують валідацію типів, обов'язкових полів і допустимих значень. Застосування цієї бібліотеки дозволило позбутися потенційних проблем які можуть виникнути через динамічну типізацію мови Python. Обмін даними між клієнтами та сервером здійснюється у форматі JSON через зашифровані HTTPS з'єднання, що в свою чергу гарантує конфіденційність та цілісність інформації.

З точки зору безпеки з яких джерел можуть надходити запити до API, було застосовано обмеження CORS-запитів, що дозволяє контролювати джерела з

яких здійснюється доступ до API, і таким чином запобігати несанкціонованим зверненням з небажаних доменів [4].

Висновки. Розроблена інформаційна система демонструє можливість побудови масштабованого та безпечного рішення для автоматизації процесів обслуговування у закладах харчування. Завдяки застосуванню сучасних інструментів та фреймворків вдалося реалізувати клієнт-серверну архітектуру з кросплатформеним доступом, розмежуванням прав користувачів та захищеним обміном даних.

ЛІТЕРАТУРА

1. McLaughlin P. FastAPI. FastAPI: Modern Python Web Development. – O'Reilly Media, 2021. – С. 27–47.
2. Kleppmann M. Розробка надійних, масштабованих та підтримуваних додатків. – К.: Видавництво "Наш Формат", 2020. – 592 с.
3. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). [Електронний ресурс]. – 2015. – 33 р. – Режим доступу: [<https://datatracker.ietf.org/doc/html/rfc7519>].
4. van Kesteren A. Cross-Origin Resource Sharing (CORS). [Електронний ресурс]. – W3C Recommendation, 2014. – 13 р. – Режим доступу: [<https://www.w3.org/TR/cors/>].

DESIGN AND DEVELOPMENT OF AN ORDER MANAGEMENT INFORMATION SYSTEM FOR FOOD SERVICE ESTABLISHMENTS WITH CROSS-PLATFORM ACCESS

Yevhenii Korchev, Liliya Bozhukha

Abstract. *The paper considers the design and implementation of an order management information system for catering establishments focused on improving service efficiency, ease of use, and compliance with information security requirements. The proposed solution provides user interaction with different levels of access, supports the processing of orders, reservations, and related business processes, and allows working from different types of devices in a single environment. Particular attention was paid to building a secure access architecture, unifying interaction channels, data validation, and interface adaptability. The developed system demonstrates a flexible solution that can be integrated for any modern food establishment and adapted to future functional and technological changes. The system architecture supports scalability, future integration of new modules, and provides a high level of user experience regardless of platform or device.*

Keywords: *information system, FastAPI, React, Android, client-server architecture, cross-platform, API, framework*

REFERENCE

1. McLaughlin P. FastAPI. FastAPI: Modern Python Web Development. – O'Reilly Media, 2021. – P. 27–47. [in English].
2. Kleppmann M. Designing Data-Intensive Applications. – Kyiv: Nash Format, 2020. – 592 p. [in English].
3. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). [Online resource]. – 2015. – 33 p. – Available at: [<https://datatracker.ietf.org/doc/html/rfc7519>]. [in English].
4. van Kesteren A. Cross-Origin Resource Sharing (CORS). [Online resource]. – W3C Recommendation, 2014. – 13 p. – Available at: [<https://www.w3.org/TR/cors/>]. [in English].