

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інформаційно-технологічної платформи для інтернет-торгівлі
на базі Instagram

Виконав: студент IV курсу, групи СНз-41
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис) Яцюк Я. О.
(прізвище та ініціали)

Керівник _____
(підпис) Гром'як Р.С.
(прізвище та ініціали)

Нормоконтроль _____
(підпис) Шимчук. Г.В.
(прізвище та ініціали)

Завідувач кафедри _____
(підпис) Боднарчук І.О.
(прізвище та ініціали)

Рецензент _____
(підпис) Стухляк П.Д.
(прізвище та ініціали)

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 13 » червня 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Яцюку Ярославу Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

Керівник роботи Гром'як Роман Сильвестрович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 23 » березня 2022 року № 4/7-172

2. Термін подання студентом завершеної роботи 13 червня 2022р.

3. Вихідні дані до роботи Літературні та інтернет джерела щодо побудови розподілених систем у хмарах

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. Розділ 1. Постановка завдання та проектування інформаційно-технологічної для інтернет-торгівлі на базі Instagram. 1.1 Формування бізнес задачі. 1.2 Формування вимог до функціоналу платформи. 1.3 Визначання стеку інформаційних технологій для реалізації платформи. 1.4 Проектування узагальненої архітектури інформаційно-технологічної платформи. 1.5 Проектування бази даних інформаційно-технологічної платформи. 1.6 Проектування користувацького інтерфейсу інформаційно-технологічної платформи.

Розділ 2. Розробка та валідація інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram. 2.1 Розробка серверної частини інформаційно-технологічної платформи. 2.2 Реалізація користувацького інтерфейсу інформаційно-технологічної платформи. 2.3 Тестування інформаційно-технологічної платформи.

3. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів) Тема мета та задачі дослідження. Опис необхідного функціоналу платформи. Діаграма використання. Високорівнева архітектура платформи. Стек інформаційних технологій. ER-діаграма бази даних. Архітектура платформи у AWS Cloud. Архітектури веб-серверу платформи. Діаграма послідовності обробки веб-сервером запиту на отримання інформації про замовлення. Веб-інтерфейс, сторінка всіх замовлень. Веб-інтерфейс, деталі замовлення. Веб-інтерфейс, редагувати товар. Веб-інтерфейс, налаштування акаунта. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., к.т.н., доцент.	04.04.22	01.05.22

7. Дата видачі завдання 24 січня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	24.01.2022	Виконано
2.	Підбір джерел про взаємодію з Instagram API	04.01.2022-30.01.2022	Виконано
3.	Переклад та опрацювання джерел про взаємодію з Instagram API, хмарні сервіси, міжсайтову взаємодію	31.01.2022-06.02.2022	Виконано
4.	Виконання дослідження щодо хмарної архітектури інформаційно-технологічної платформи	07.02.2022-13.02.2022	Виконано
5.	Оформлення розділу «Постановка завдання та проєктування інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram»	14.02.2022-06.03.2022	Виконано
6.	Оформлення розділу «Розробка та валідація інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram»	07.03.2022-03.04.2022	Виконано
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	04.04.2022-17.04.2022	Виконано
8.	Виконання завдання до підрозділу «Основи охорони праці»	18.04.2022-01.05.2022	Виконано
9.	Оформлення кваліфікаційної роботи	02.05.2022-15.05.2022	Виконано
10.	Нормоконтроль	16.05.2022-22.05.2022	Виконано
11.	Перевірка на плагіат	03.06.2022	Виконано
12.	Попередній захист кваліфікаційної роботи	07.06.2022	Виконано
13.	Захист кваліфікаційної роботи	14.06.2022	

Студент

(підпис)

Яцюк Я.О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гром'як Р.С.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram // Кваліфікаційна робота освітнього рівня «Бакалавр» // Яцюк Ярослав Олегович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНз-41 // Тернопіль, 2022 // С. – 44 , рис. – 17, слайдів – 16, додат. – 2, бібліогр. – 33.

Ключові слова: інформаційно-технологічна платформа, інтернет-торгівля, автоматизація продаж, хмарна архітектура, instagram, java, aws, spring framework, react.

Кваліфікаційна робота присвячена розробці інформаційно-технологічної платформи для автоматизації торгівлі в мережі Instagram. Платформа дозволяє власникам магазинів ефективно здійснювати облік продаж у Instagram, ефективно відслідковувати статуси замовлень.

У першому розділі кваліфікаційної роботи розглянуто основні функціональні особливості проєктованої інформаційно-технологічної платформи. Спроектовано серверну, клієнтську, хмарну та загальну архітектуру платформи. Вибрано та обґрунтовано використання стеку технологій.

У другому розділі кваліфікаційної роботи розглянуто деталі реалізації та тестування серверної та клієнтської частин інформаційно-технологічної платформи. Зокрема, практичну реалізацію трьох-рівневої архітектури створеного сервера та користувачького Single Page Application інтерфейсу. Описано процес тестування інформаційно-технологічної платформи за допомогою модульних та інтеграційних тестів.

ANNOTATION

Development of an information technology platform for online Instagram based commerce // Qualification work of educational level «Bachelor» // Yatsiuk Yaroslav Olehovych // Ternopil Ivan Puluj national Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, group CHz-41 // Ternopil, 2022 // pages – 39, figures – 20, chair. – 16 annexes – 2, references – 33.

Key words: information technology platform, e-commerce, sales automation, cloud architecture, instagram, java, aws, spring framework, react.

The qualification work is devoted to the development of an information technology platform for online Instagram based commerce. The platform allows store owners to effectively track Instagram sales and order statuses.

In the first chapter of the qualification work, the main functional features of the designed information technology platform are considered. Was designed a server, client, cloud, and general platform architectures. The technology stack was selected and justified.

In the second chapter of the qualification work, the details of implementing and testing the service and client parts of the information technology platform are considered. In particular, the practical implementation of the three-layered architecture of the created server and the user single page application interface. Also, was described a process of testing an information technology platform using unit and integration tests.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CSS (англ. Cascading Style Sheets) – Каскадні таблиці стилів.

HTML (англ. Hyper Text Markup Language) – Мова розмітки гіпертекстових документів.

API (англ. Application Programming Interface) – Прикладний програмний інтерфес.

JS (англ. Java Script) – Скриптова мова програмування.

UI (англ. User Interface) – Інтерфейс користувача.

UX (User Experience) – Користувацький досвід, зазвичай це означає зручність користування певним програмним забезпеченням із точки зору кінцевого користувача.

AWS (англ. Amazon Web Services) – дочірня компанія «Amazon.com», що надає в оренду платформу хмарних обчислень на основі платної підписки.

IdP (англ. Identity Provider) – Провайдер ідентифікації.

DTO (англ. Data Transfer Object) – Об'єкт трансферу даних.

CDN (Content Delivery Network) – Мережа доправлення контенту.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО-ТЕХНОЛОГІЧНОЇ ПЛАТФОРМИ ДЛЯ ІНТЕРЕТ-ТОРГІВЛІ НА БАЗІ INSTAGRAM.....	9
1.1 Формування бізнес задачі.....	9
1.2 Формування вимог до функціоналу платформи	9
1.3 Визначення стеку інформаційних технологій для реалізації платформи.....	12
1.4 Проєктування узагальненої архітектури інфомаційно- технологічної платформи.....	15
1.5 Проєктування бази даних інформаційно-технологічної платформи.....	18
1.6 Проєктування користувацького інтерфейсу.....	19
1.7 Проєктування хмарної архітектури інформаційно-технологічної платформи.....	21
РОЗДІЛ 2. РОЗРОБКА ТА ВАЛІДАЦІЯ ІНФОРМАЦІЙНО- ТЕХНОЛОГІЧНОЇ ПЛАТФОРМИ ДЛЯ ІНТЕРНЕТ-ТОРГІВЛІ НА БАЗІ INSTAGRAM	23
2.1 Розробка серверної частини інформаційно-технологічної платформи.....	23
2.2 Реалізація користувацького інтерфейсу інформаційно- технологічної платформи.....	31
2.3 Тестування інформаційно-технологічної платформи.....	34
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	36
3.1 Долікарська допомога при контузіїх	36

3.2 Вимоги пожежної безпеки при гасінні електроустановок.	37
ВИСНОВКИ.....	39
ПЕРЕЛІК ДЖЕРЕЛ.....	40
ДОДАТКИ	

ВСТУП

Актуальність теми. На даний час на ринку обмежена кількість програмного забезпечення для автоматизації продаж в мережі Instagram. При цьому воно зазвичай є англomовними та платними. Водночас Instagram набуває популярності не лише як соціальна мережа та хостинг фотографій, а також як торгова площадка. Багато людей публікують товари для продажу в Instagram, та успішно продають за допомогою угод в приватних повідомленнях, або через посилання на веб-сайти. Коли продавець не має власного інтернет-магазину, або має бажання продавати товари через приватні повідомлення з ростом кількості замовлень виникне проблема обліку, аудиту та автоматизації продаж. Важливим є формування та накопичення клієнтської бази для здійснення таргетової реклами та розповсюдження інформації про новинки або акційні пропозиції. Для будь-якого інтернет-магазину вкрай важливо зберігати історичні дані щодо продажів для своєчасного виявлення та відслідковування ринкових тенденцій, оперативного прийняття рішення на основі накопичених статистичних даних та ефективного запроваджувати акційних пропозицій в залежності від сезонної активності продажів. Тому розробка інформаційно-технологічної платформи є дуже актуальною.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є підвищення поінформованості споживачів щодо товарів та послуг та покращення зручності роботи торговельних представників та громадян в процесі продаж з використанням соціальної мережі Instagram. Для досягнення поставленої мети було сформовано ряд наступних завдань:

- Провести аналіз основних бізнес-потреб та сформуванати перелік основних функціональних можливостей інформаційно-технологічної платформи.
- Обґрунтувати стек технологій розробки.

- Спроекувати архітектуру, структуру бази даних інформаційно-технологічної платформи.
- Визначити дизайн користувацького інтерфейсу платформи.
- Визначити хмарний провайдер для розгортання платформи.
- Здійснити розробку клієнтської та серверної частин, провести валідацію та тестування платформи;
- Розробити плани заходів щодо розгортання у хмарі, архівного копіювання та забезпечення інформаційної безпеки платформи.

Практичне значення одержаних результатів. Результат дослідження та розробки – це інформаційно-технологічна платформа для торгівлі в мережі Instagram з адаптивним та мультимовним інтерфейсом.

РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО-ТЕХНОЛОГІЧНОЇ ПЛАТФОРМИ ДЛЯ ІНТЕРЕТ- ТОРГІВЛІ НА БАЗІ INSTAGRAM

1.1 Формування бізнес задачі

На даний час поширеною сферою дрібного та середнього бізнесу є продаж різнотипового одягу в Instagram за допомогою сервісу приватних повідомлень. Зі зростанням популярності Instagram -сторінок збільшується кількість онлайн-замовлень і, як наслідок, суттєво ускладнюються процеси відстеження та управління замовлень. На даний час переважною більшістю користувачів для зберігання замовлень, ведення статистики та аналітики здебільшого використовується Microsoft Excel. Таке рішення погано масштабується і не зручне. Наприклад, на мобільному пристрої заповнювати Excel таблиці не зручно, а замовлення можуть надходити впродовж цілого дня і навіть пізно у вечері. Відповідно комп'ютер не завжди під рукою, а записати інформацію про замовлення потрібно відразу після його отримання.

Для розв'язання такого класу бізнес-задач доцільно розробити спеціалізовану інформаційно-технологічну платформу з адаптивним веб-інтерфейсом. Адаптивність клієнтської частини платформи дозволить її використання на будь-якому типі електронно-обчислювальних пристроїв, починаючи від стаціонарного комп'ютера та закінчуючи смартфоном під керуванням будь-якої операційної системи.

1.2 Формування вимог до функціоналу платформи

Базуючись на сформованій у попередньому параграфі бізнес задачі можна сформулювати перелік функціональних особливостей, який повинна містити зазначена інформаційно-технологічна платформа:

- забезпечення безпечної аутентифікації користувачів;
- безпечну реєстрацію менеджерів та адміністраторів;
- додавання облікових записів клієнтів до бази даних;
- відображення розширеної інформації про клієнта;
- перегляд множини останніх покупок клієнта;
- перегляду статистики покупок, зокрема, кількості здійснених замовлень та сумарних витрат по категоріях;
- додавання інформації про нові товари;
- створення нового замовлення;
- автоматичне формування накладної на замовлення;
- автоматичне відстеження статусу доставки замовлення;
- видалення або скасування замовлення;
- редагування інформації про товар;
- створення нотаток про клієнта та замовлення;
- пошук та фільтрування інформації про клієнтів, замовлення та товари;
- сортування інформації по клієнтах, замовленнях та товарах;
- управління обліковим профілем адміністратора;
- зміна пароля для входу в систему;
- відновлення паролю;
- денну та нічну теми користувацького інтерфейсу;
- адаптивність інтерфейсу до різних розмірів, та типів екранів;
- багатомовність користувацького інтерфейсу, зокрема, українська та англійська мовні інтерпретації;
- валідація полів вводу для елементів користувацького інтерфейсу;
- підтвердження виконання подій та операцій за допомогою додаткових інтерактивних вікон.

Для візуального представлення ключових моментів використання інформаційно-технологічної платформи розроблено діаграма використання [3], подана в нотації «Uses Cases» на рисунку 1.1. Адміністратор – це

користувач системи, який де-факто є адміністратором певного інтернет-магазину в Instagram.

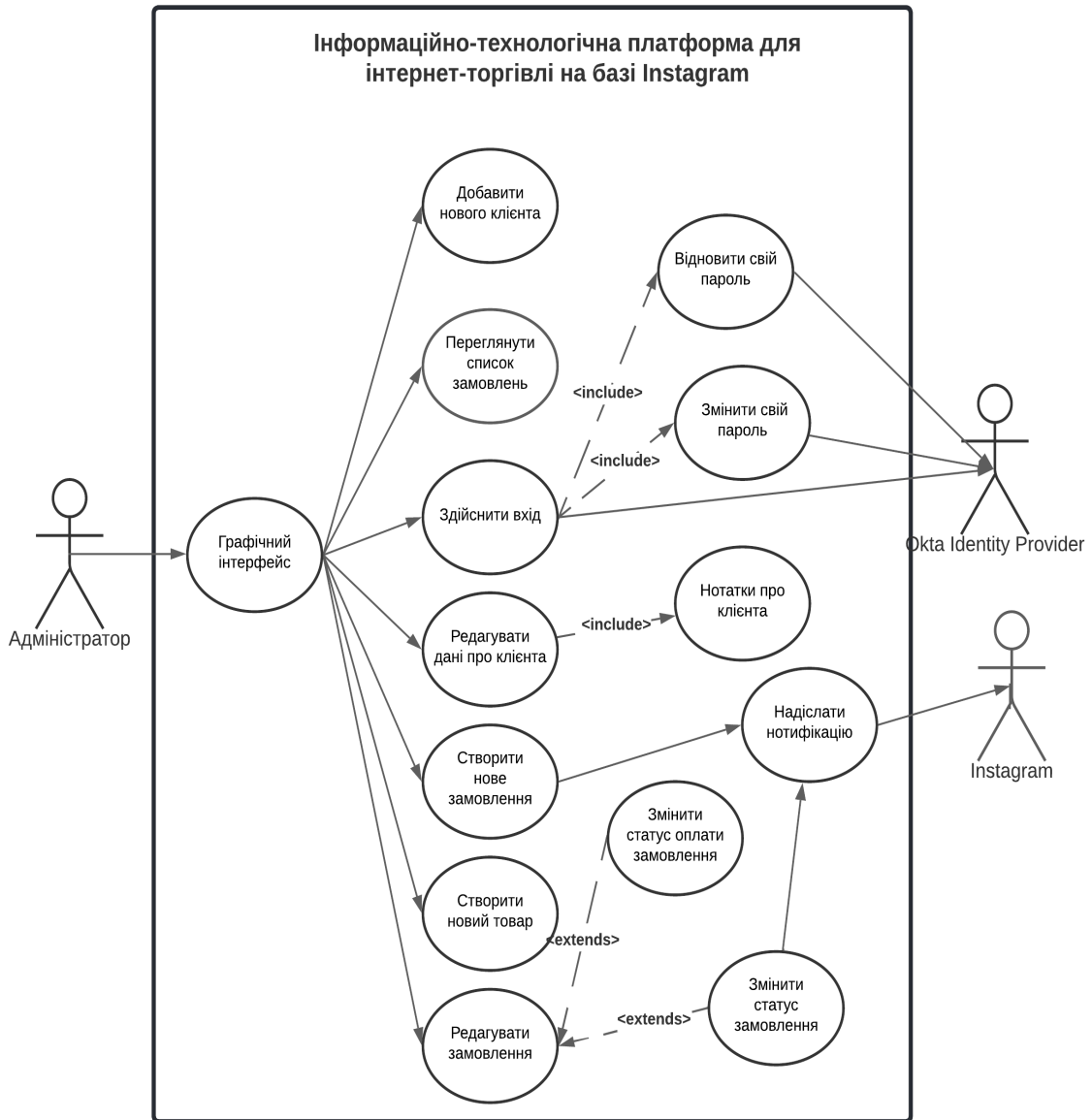


Рисунок 1.1 – Діаграма використання інформаційно-технологічної платформи для інтернет торгівлі на базі Instagram

«Okta Identity Provider» – це третьо-сторонній сервіс (англ. third-party service), який виконує роль авторизаційного сервера та бази даних адміністраторів системи. Авторизаційний сервер «Okta» зручно використовувати тому, що це повністю готовий та протестований сервіс, який

ефективно несе відповідальність за надійне збереження паролів та адміністративної інформації.

В «Okta» інтегровано готовий функціонал для зміни та відновлення паролів, активації та блокування облікових записів користувачів, делегування повноважень на здійснення операцій в проєктованій інформаційно-технологічній платформі.

Інформаційно-технологічна платформа повинна бути захищеною та стійкою до втрати даних. Обчислювальна архітектура інформаційно-технологічної платформи повинна відповідати сучасним стандартам та бути легко масштабованою в умовах високого навантаження [1], [2]. Користувацький інтерфейс повинен бути зручним у користуванні та підтримувати українську та англійську мови інтерфейсу.

1.3 Визначення стеку інформаційних технологій для реалізації платформи

Для реалізації даної інформаційно-технологічної платформи, після детального аналізу вимог, було прийнято рішення щодо використання обширного стеку інформаційних технологій.

Зокрема для користувацького інтерфейсу:

- React 17.
- Material UI.
- React Router 6.
- i18next.
- Yup.
- Formik.
- Npm.

Для серверної частини:

- Java 17 (LTS).

- Spring Boot 2.6.2.
- Spring Data JPA.
- Spring Security OAuth2.
- Logback.
- Lombok.
- Maven.
- PostgreSQL.

Для розгортання інформаційно-технологічної платформи:

- AWS Cloud Provider.
- Docker.

IdP:

- Okta.

Для моніторингу доцільно використати «AWS CloudWatch».

1.3.1 Обґрунтування списку вибраних технологій

React – це популярна на даний момент часу JS-бібліотека з відкритим вихідним кодом для створення користувацьких інтерфейсів [4]. Вибір React, як клієнтської бібліотеки є доволі стандартним рішенням на даний час адже, він є лідером серед програмно-алгоритмічних рішень для створення односторінкових веб-застосунків (англ. SPA – Single Page Application). Зокрема всесвітньо-відомі компанії Meta (ex. Facebook), Netflix, Yahoo, Airbnb, Sony активно використовують зазначену бібліотеку для розробки та впровадження інноваційних веб-інтерфейсів та застосунків. Це гарантує довготривалу підтримку JS-бібліотеки.

Водночас, Java – є лідером Українського та світового IT-ринку. Ця мова програмування тривалий час доводить свою високу ефективність. Програми написані на Java легко супроводжувати, підтримувати, масштабувати та убезпечувати [5].

Безсумнівна перевага мови програмування Java – вона компілюється в байт код – спеціальний код, який потім інтерпретується в машинні інструкції, в залежності від процесора та операційної системи тієї програмно-апаратної платформи на якій буде відбуватись Java-програма. Потім скомпільований застосунок виконується віртуальною машинною JVM – Java Virtual Machine. Це забезпечує кросплатформність написаного Java-коду. Він не потребує перекомпіляції для різних ОС та процесорів [6].

На чистій Java написати серверну частину проекту, було б дуже довго та складно. Тому для автоматизації процесів розроблення виберемо серверний веб-фреймворк «Spring Boot». Він дозволяє суттєво пришвидшити процес розробки серверної частини програмного забезпечення. Адже це не просто набір реалізованих методів, а ціла програмно-алгоритмічна інфраструктура, для роботи із безпекою, базами даних, транзакціями, Web API тощо [7].

«Spring Framework», завдяки вбудованому патерну «Dependency Injection», спонукає веб-розробників формувати слабо зав'язаний на інші компоненти – англ. loosely coupled – програмний код. В перспективі це дозволить спростити процедури розширення та внесення змін до елементів інформаційно-технологічної платформи.

Для зберігання даних прийнято рішення щодо використання реляційної бази даних PostgreSQL. Ця база даних здобула високу популярність за останні роки завдяки високим показникам:

- продуктивності;
- надійності;
- гнучкості налаштувань.

Зазначена база даних поширюється із відкритим вихідним кодом – це робить її абсолютно безплатною. Не зважаючи на безкоштовність, вона по більшості критеріїв абсолютно не поступається платним базам на кшталт як «MSSQL» або «Oracle Database». По багатьох критеріях навіть переважає конкурентів [8].

Для розгортання інформаційно-технологічної платформи було вирішено використовувати «Amazon Web Services (AWS)» [9], оскільки цей провайдер на даний час вважається де-факто стандартом для переважної більшості бізнес-застосунків – англ. enterprise applications. При цьому він суттєво перевершує конкурентів, зокрема GCP – «Google Cloud Provider» та «Microsoft Azure». Водночас вартість обраного хмарного інформаційно-технологічного рішення є оптимальною, адже, використовуючи AWS платити потрібно лише спожиті ресурси. Тому, поки платформа є достатньо не великого розміру та не має обширної аудиторії користувачів, ціна розгортання на AWS буде прийнятно-низькою [10].

1.4 Проектування узагальненої архітектури інформаційно-технологічної платформи

Архітектура проектованої інформаційно-технологічної платформи для інтернет торгівлі на базі Instagram повинна бути:

- максимально простою;
- ефективною;
- захищеною від кібератак;
- захищеною від втрати даних;
- стійкою до непередбачуваних обставин на серверах;
- доступною у різних географічних регіонах;
- гнучкою до змін.

Відповідно до призначення, інформаційно-технологічна платформа не буде високонавантаженою, тому не потрібно розбивати її на множину мікро-сервісів. При цьому буде цілком достатньо сформувати декілька функціонально-розділених сервісів.

Серверна частина платформи повинна бути розроблена з дотриманням вимог «RESTful»-архітектури. Комунікація між клієнтом та сервером повинна

відбуватися засобами HTTP (HTTPS) протоколу. Така архітектура зробить клієнтську та серверну частину повністю незалежними один від одного. В свою чергу, це дозволить здійснювати незалежне розгортання будь-яких компонентів інформаційно-технологічної платформи та зберігати кодові бази у різних онлайн-репозиторіях.

Клієнтська частина інформаційно-технологічної платформи повинна бути SPA-застосунком – це дозволить сформувати максимально-швидкий та ефективний користувацький інтерфейс, водночас забезпечуючи гнучкість до змін та використання переваг «RESTful» сервісів (Див. рисунок 1.2).

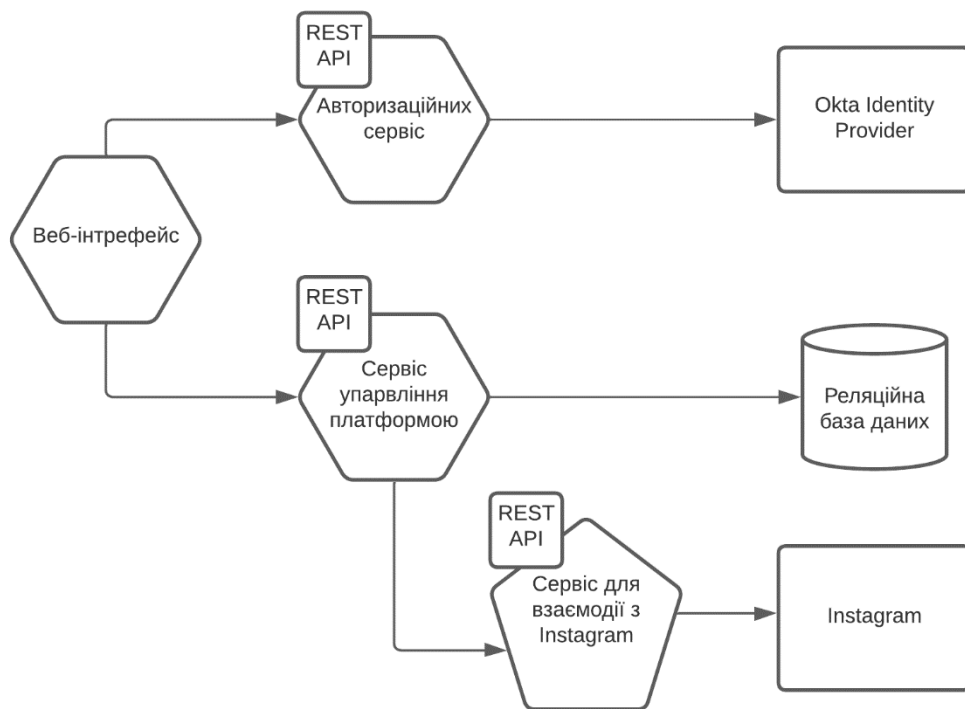


Рисунок 1.2 – Узагальнена структура інформаційно-технологічної платформи для інтернет торгівлі на базі Instagram

Доцільно виділити основні структурні інформаційно-технологічні компоненти платформи:

– Сервіс управління платформою – програмно-алгоритмічний комплекс, розроблений засобами мови програмування Java.

– Сервіс автентифікації користувачів – застосунок, розроблений засобами мови програмування Java.

– IdP – третьо-сторонній сервіс який надаватиме «JWT»-токени та зберігатиме адміністраторські дані, зокрема інформацію щодо електронної пошти та паролів.

– Програмно-алгоритмічні засоби для практичної реалізації функцій та елементів клієнтського інтерфейсу інформаційно-технологічної платформи – React-програма.

– Реляційна база даних.

Узагальнена архітектура інформаційно-технологічної платформи подана (див. рис. 1.3) – це типова «3-tier architecture» в якій визначено три основні рівні:

- «Presentation Tier» – клієнтська React програма;
- «Logic Tier» – серверна програма яка містить основні елементи бізнес-логіки – «RESTful» веб-сервіси на мові програмування Java.
- «Data Tier» – основна база даних «PostgreSQL» та «Okta Identity Provider».

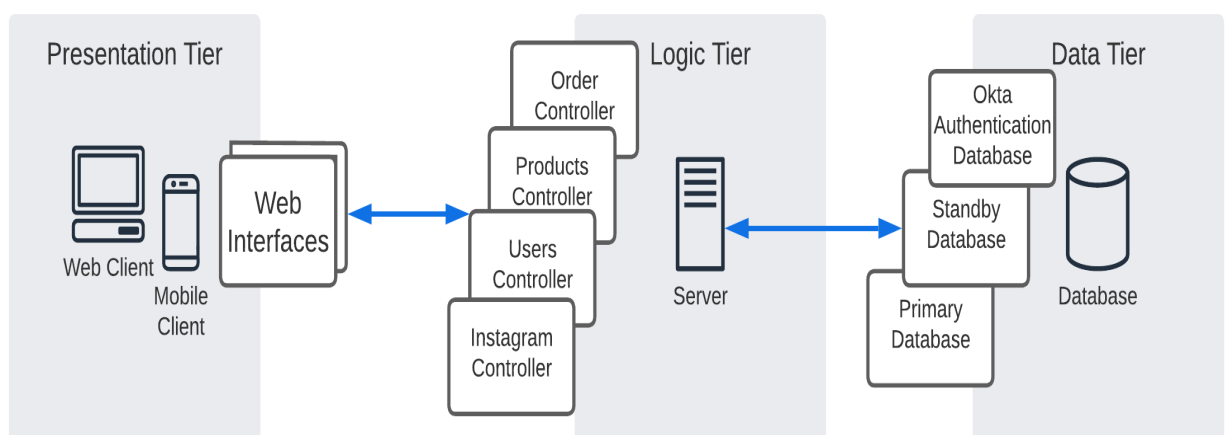


Рисунок – 1.3 – Трьох-рівнева архітектура інформаційно-технологічної платформи для інтернет торгівлі на базі Instagram

1.5 Проектування бази даних інформаційно-технологічної платформи

Для проектування реляційної бази даних зручно використати ER діаграму – «Entity Relationship Diagram» [11]. Проаналізувавши функціональні потреби розроблюваної інформаційно-технологічної платформи спроектовано подану на рисунку 1.4 реляційну модель бази даних.

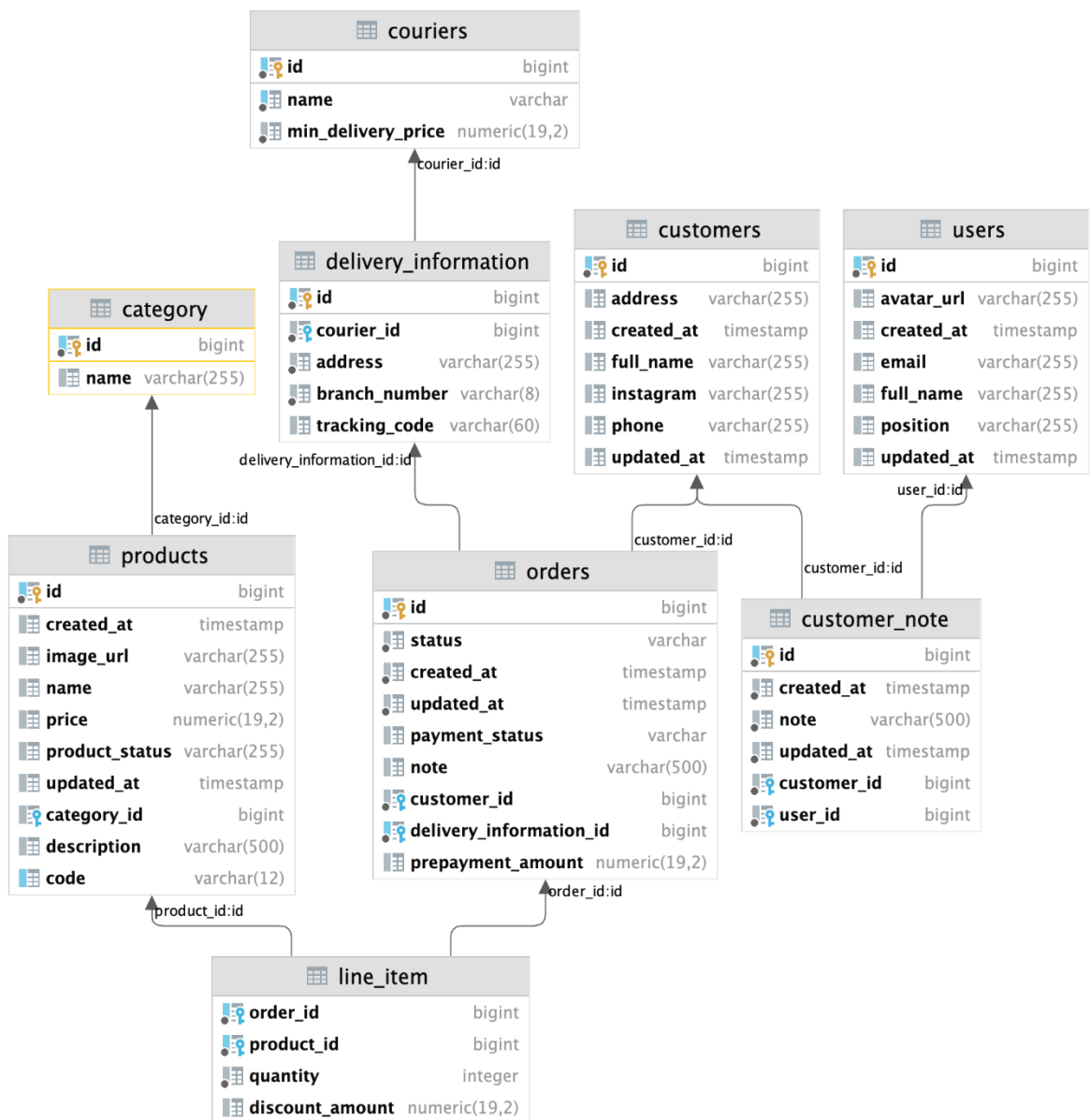


Рисунок 1.4 – ER-діаграма інформаційно-технологічної платформи для інтернет торгівлі на базі Instagram

Подана на рис. 1.4 база даних інформаційно-технологічної платформи – нормалізована. Вона приведена до третьої нормальної форми. Така форма подання структури таблиць дозволяє легко формувати нові зв'язки та динамічно змінювати дані. При потребі можна буде будь коли розширити функціональні можливості інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram. Водночас, структура бази даних дозволить підвищити ефективність виконання складних аналітичних запитів для формування оперативних та періодичних звітів щодо продажів, спрощуючи практичну реалізацію інтелектуальних аналітичних засобів.

Проте, нормалізація даних має свою ціну – збільшує кількість перехресних складних запитів тощо. Зокрема, щоб отримати повну інформацію про товар, потрібно виконати запити до таблиці з категоріями товарів. Відповідно, потрібно додати JOIN-операцію в запит – це спричинить зменшення швидкості виконання запитів.

В окремих випадках доцільно використовувати комбінацію з реляційної та не реляційної баз даних для досягнення вищої продуктивності системи. Для проєктованої інформаційно-технологічної платформи немає потреби в денормалізації даних або одночасному використанні різних типів баз даних. Тому, що заплановане навантаження на інформаційно-технологічну платформу для інтернет-торгівлі на базі Instagram є відносно невеликим.

В даному випадку важливішим фактором за швидкість виконання звичайних запитів є можливість здійснювати надшвидкі аналітичні запити, для пришвидшення процесів відслідковування динаміки бізнес-операцій та покращення розуміння тенденцій зміни потреб клієнтів.

1.6 Проєктування користувацького інтерфейсу

Користувацький інтерфейс інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram повинен бути:

- Простим.
- Зрозумілим.
- Інтуїтивним.
- Адаптивним.
- Кросплатформним.

На рисунку 1.5 зображені основні функціональні компоненти інтерфейсу інформаційно-технологічної платформи.

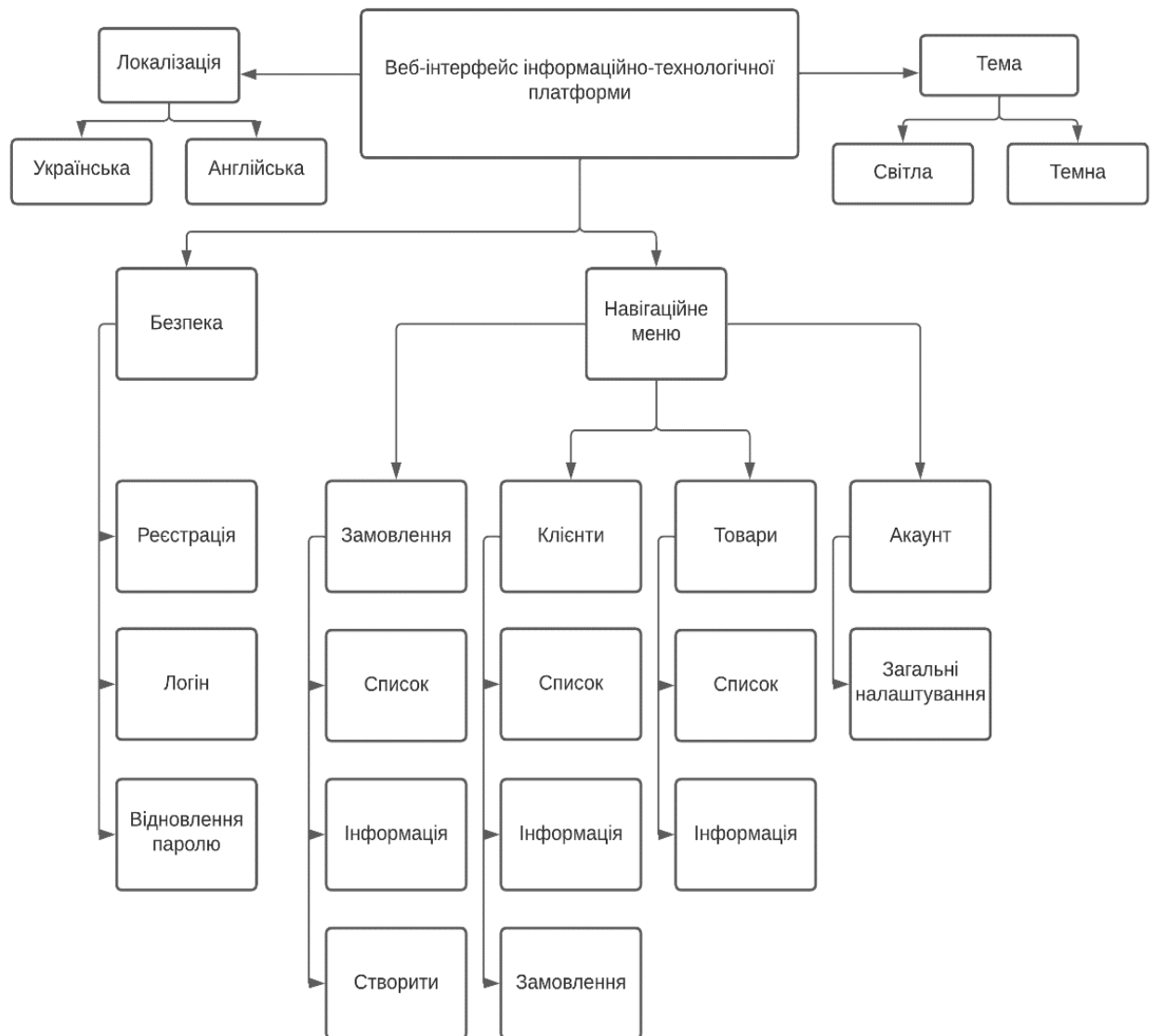


Рисунок 1.5 – Функціональна схема клієнтської частини інформаційно-технологічної платформи

Всі компоненти вводу повинні забезпечувати інтерактивну валідацію даних з інтуїтивно-зрозумілими користувачам елементами та підказками. Таблиці повинні мати можливість пагінації, оперативного сортування та фільтрування даних.

Для уникнення не авторизованого доступу клієнтська частина інтерфейсу інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram повинна мати безпечну та захищену посторінкову маршрутизацію та навігацію.

1.7 Проєктування хмарної архітектури інформаційно-технологічної платформи

Оскільки інформаційно-технологічна платформа буде розгортатися на базі «AWS Services», то потрібно спроектувати хмарну інформаційно-технологічну архітектуру платформи з врахуванням вимог безпеки. Адже її програмно-алгоритмічні компоненти, модулі, сервіси та елементи будуть загальнодоступними в мережі Інтернеті.

Також потрібно забезпечити доступність інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram. Оскільки, якщо вона буде розгорнута лише на одному сервері, то буде високий ризик не доступності сервера у певний момент часу. Як наслідок, користувачі системи можуть втратити доступ до онлайн-ресурсів.

Щоб забезпечити оптимально-високу швидкість завантаження для географічно-віддалених користувачів, статичні елементи інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram повинні зберігатися та розповсюджуватися засобами мереж постачання контенту – CDN [12]. На рисунку 1.6 зображена хмарна архітектура інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram, спроектована для реалізації засобами хмарного провайдера «AWS».

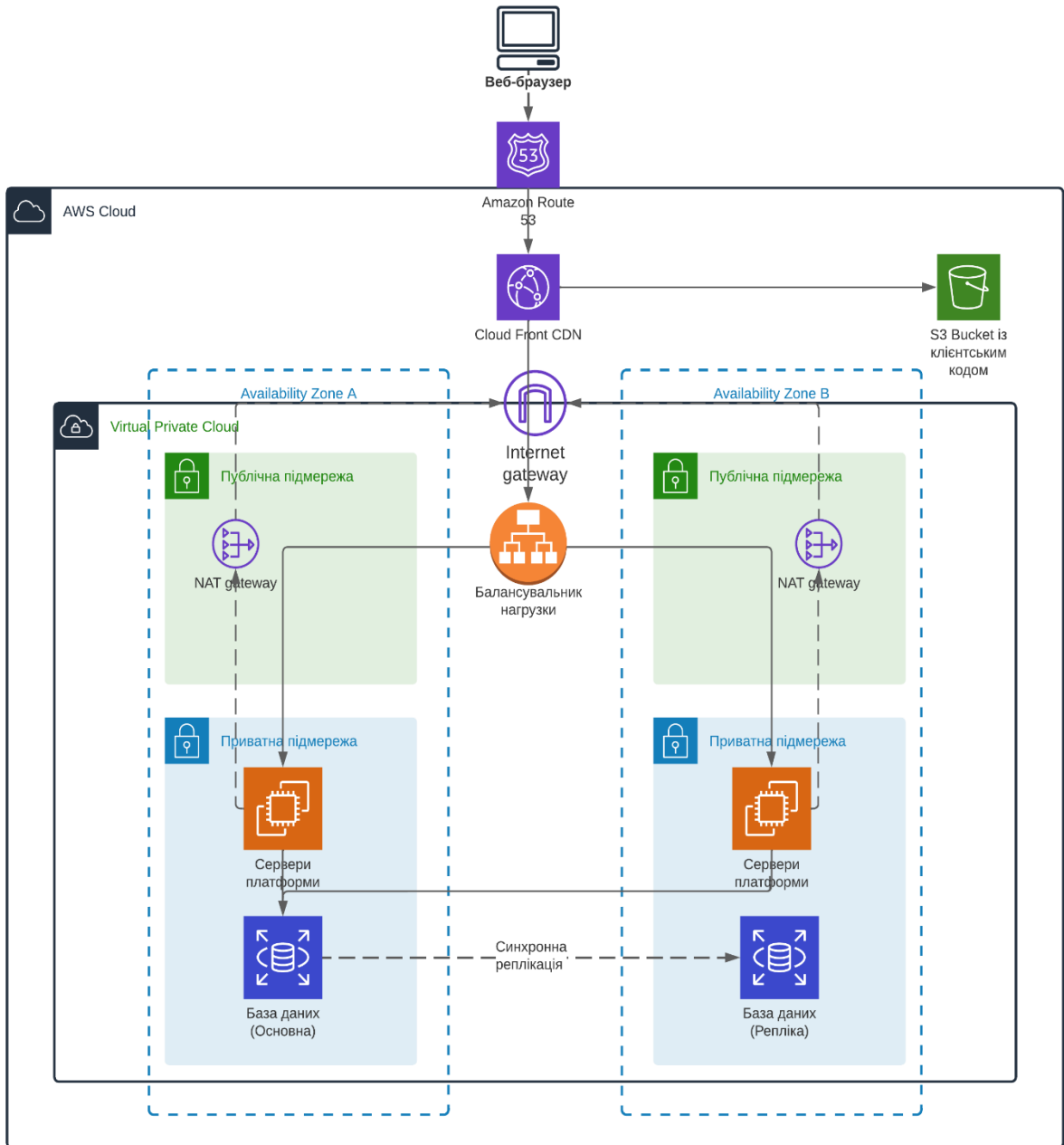


Рисунок 1.6 – Хмарна архітектура інформаційно-технологічної платформи

Точкою входу до хмарної інфраструктури є DNS-сервіс «Amazon Route 53». Потім трафік надходить до CDN з клієнтською частиною інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram.

РОЗДІЛ 2. РОЗРОБКА ТА ВАЛІДАЦІЯ ІНФОРМАЦІЙНО-ТЕХНОЛОГІЧНОЇ ПЛАТФОРМИ ДЛЯ ІНТЕРНЕТ-ТОРГІВЛІ НА БАЗІ INSTAGRAM

2.1 Розробка серверної частини інформаційно-технологічної платформи

Серверна частина платформи розроблена із використанням Spring Boot Web для реалізації веб-рівня (англ. Web Layer) та Spring Data JPA для реалізації рівня доступу до даних (англ. Data Access Layer). Загалом можна виділити три основні рівні [13] (див. рисунок 2.1) на які розділено веб-сервер платформи.

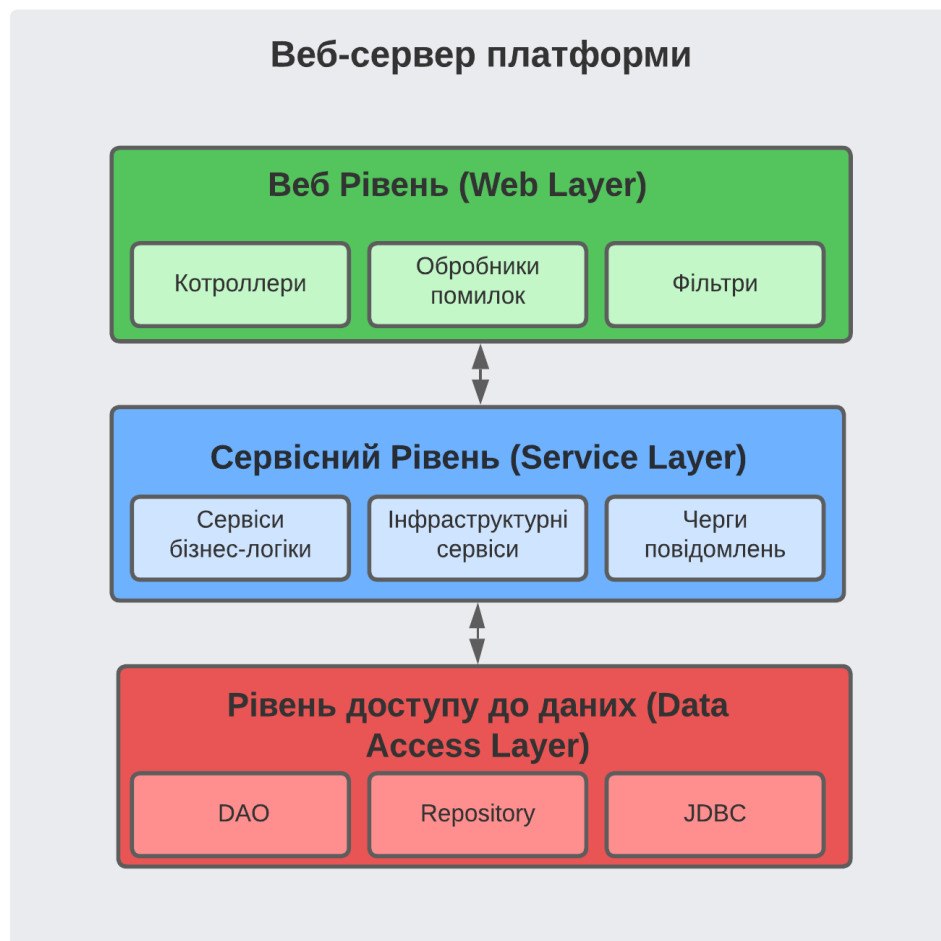


Рисунок 2.1 – Рівні веб-серверу інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

Ці три рівні також притаманні більшості Spring Boot застосунків розроблених впродовж останнього періоду часу. Розбивання кодової бази сервера на шари полегшує тестування та підтримку коду, а також робить компоненти серверу не зав'язаними один на одному (англ. loosely coupled) [16], що дозволяє використовувати окремі компоненти в різних частинах коду. На рисунку 2.2 зображено діаграму послідовності обробки запиту на отримання інформації про замовлення по унікальному ідентифікатору замовлення (orderId).

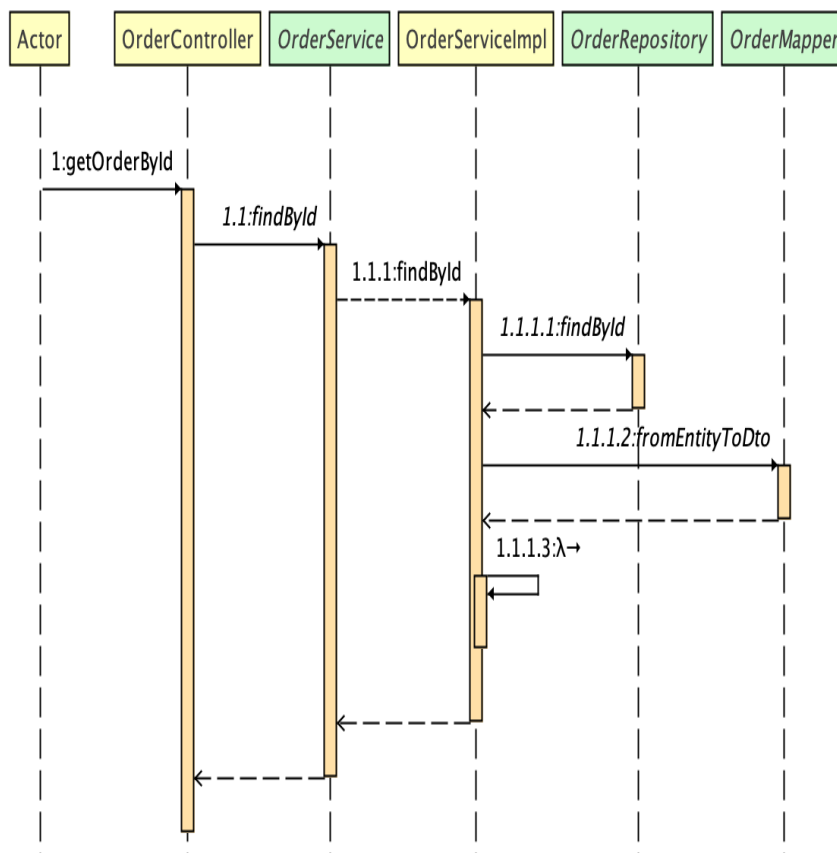


Рисунок 2.2 – Діаграма послідовності обробки запиту на отримання інформації про замовлення

Actor – клієнтська програма яка надсилає запит на отримання інформації про замовлення, OrderController – компонент веб рівня, OrderService – компонент сервісного/бізнес рівня, OrderRepository – компонент рівня доступу до даних.

2.1.1 Розробка веб-рівня серверної частини платформи

Веб рівень серверної частини платформи складається із контролерів та обробника виняткових ситуацій. Контролери – це Java класи які призначені для отримання HTTP запитів [14] від клієнтської частини платформи та відправлення відповідей із результатом опрацювання цих запитів назад на клієнт. Кожен контролер відповідає за певний ресурс [17], таких ресурсів у платформі є п'ять:

- Клієнти – CustomerController клас;
- Рисунки/фото – ImageController клас;
- Замовлення – Order Controller клас;
- Товари – ProductController клас;
- Користувачі – UserController клас.
- Автентифікація – AuthController.

На рисунку 2.3 зображено діаграму класів «CustomerController», «UserController» та «AuthController».

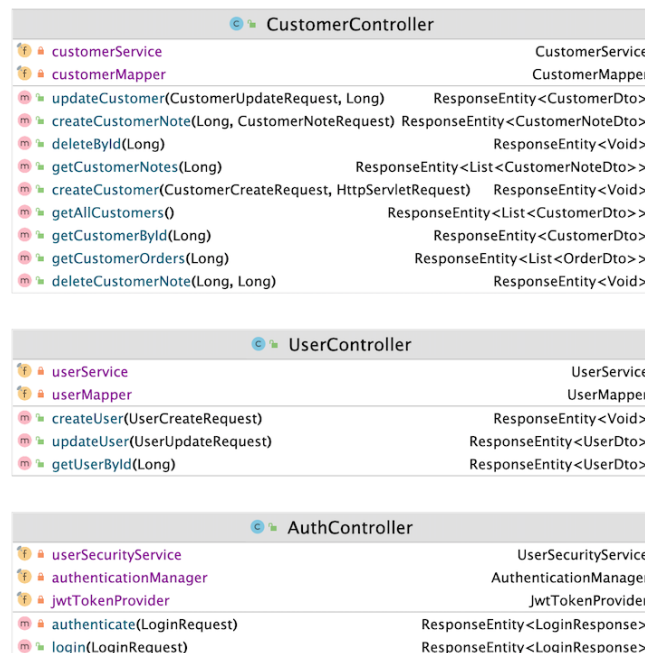


Рисунок 2.3 – Діаграма класів веб-рівня інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

На рисунку 2.4 зображено діаграму класів «ProductController», «OrderController» та «ImageController».

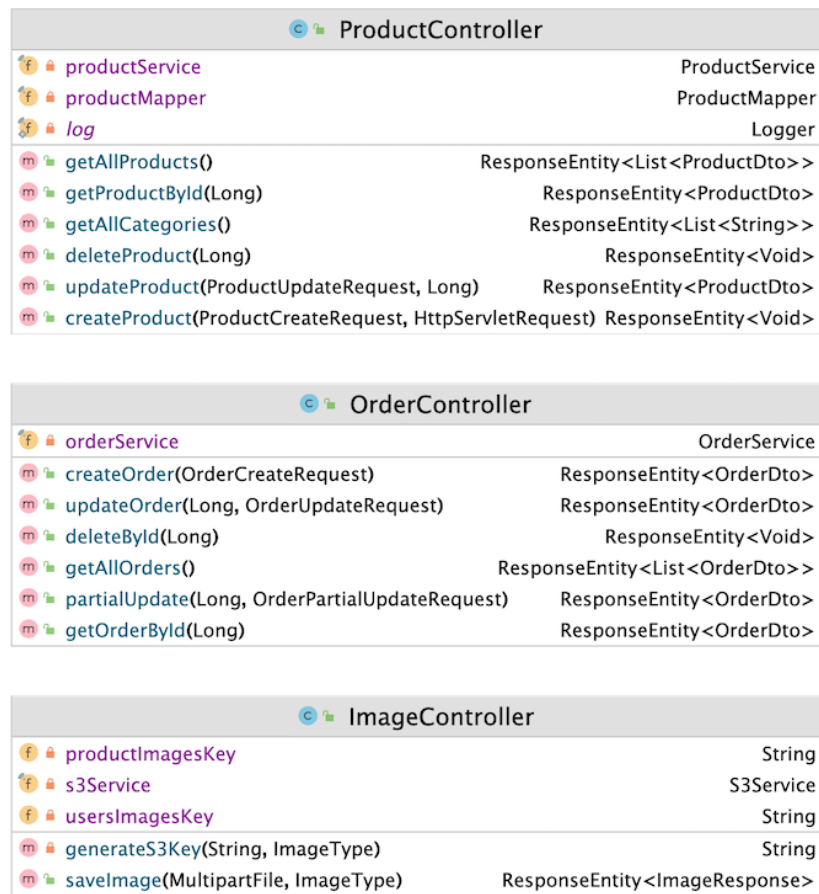


Рисунок 2.4 – Діаграма класів веб рівня інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

Логіка контролерів є дуже простою та не містить методів пов'язаних із бізнес-логікою платформи. Все що робить кожен контролер це приймає запит від клієнта, перетворює дані в потрібний формат та передає їх далі на сервісний рівень, який містить всю основну бізнес-логіку [16]. В лістингу 2.1 показано уривок коду із класу CustomerController.

Лістинг 2.1 – CustomerController.java

```
@RestController
@RequestMapping("/api/customers")
@RequiredArgsConstructor
```

```

public class CustomerController {

    private final CustomerMapper customerMapper;
    private final CustomerService customerService;

    @PutMapping("/{id}")
    public ResponseEntity<CustomerDto>
updateCustomer(@RequestBody CustomerUpdateRequest payload,
@PathVariable Long id) {
        CustomerDto dto =
customerMapper.fromUpdateRequestToDto(payload, id);
        return ResponseEntity.ok(customerService.update(dto));
    }

    // Інші методи.....
}

```

Обробник виняткових ситуацій – це Java клас основне завдання якого перехоплювати виняткові ситуації в які можуть виникнути в програмі та в залежності від типу помилки повертати відповідні HTTP статуси та повідомлення на клієнт [17]. В лістингу 2.2 показано фрагмент коду із класу обробника виняткових ситуацій.

Лістинг 2.2 – CustomerController.java

```

@ControllerAdvice
@Log4j2
public class APIExceptionHandler extends
ResponseEntityExceptionHandler {

    @ExceptionHandler(EntityNotFoundException.class)
    public ResponseEntity<Response>
handleEntityNotFoundException(final EntityNotFoundException ex)
{
        log.info(ex.getMessage());
        return ResponseEntity.notFound().build();
    }

    // Інші методи.....
}

```

Показаний в лістингу 2.2 метод `handleEntityNotFoundException` перехоплює помилку `EntityNotFoundException`, яка сповіщає про те, що об'єкт не був знайдений у базі даних, та повертає HTTP статус 404 (не знайдено) [18]

на клієнт. Такий механізм забезпечує захист від розкриття внутрішньої системи веб-серверу, адже в тексті помилки може міститися інформація про бібліотеки чи фреймворк які було використано при обробці запиту, також такий механізм спрощує обробку помилок на клієнті.

2.1.2 Розробка сервісного рівня інформаційно-технологічної частини платформи

Сервісний рівень веб-серверу платформи складається із інтерфейсів бізнес-логіки, їхніх імплементацій та інфраструктурних класів. Саме цей рівень є серцем всієї бізнес-логіки платформи і всі доменні об'єкти (англ. Domain objects) [19] обробляються на цьому рівні. Для передачі даних між веб рівнем та сервісним рівнем використовуються спеціальні об'єкти – DTO [20], такий механізм дозволяє ізолювати логіку роботи з доменними об'єктами в одному місці. На рисунку 2.5 показано діаграму класів «Customer», «User» та «Auth» сервісів.

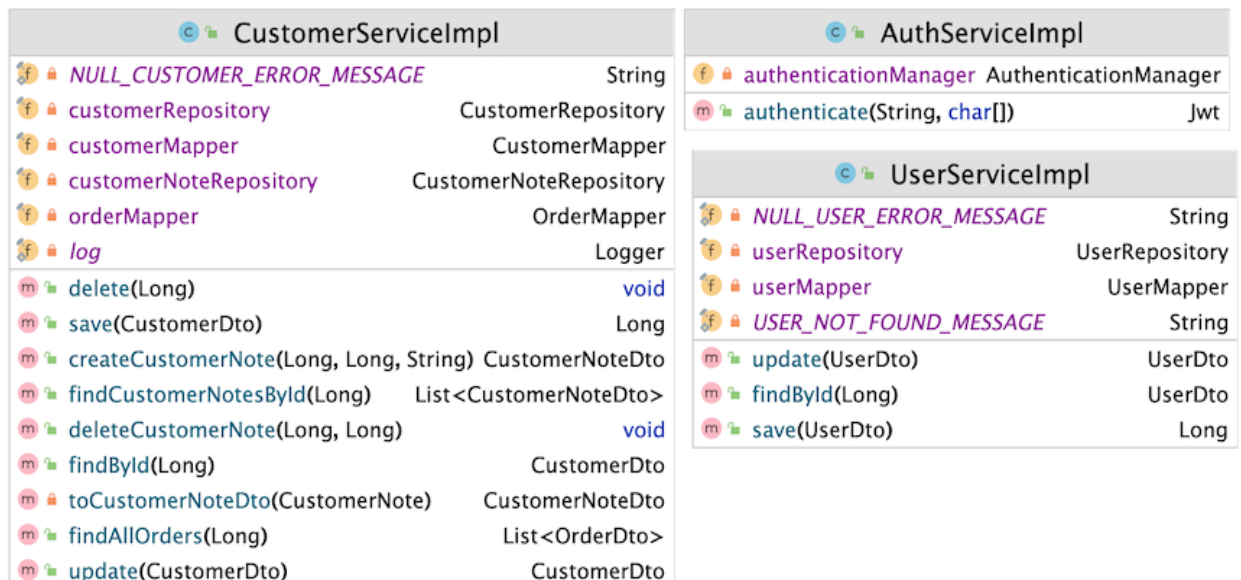


Рисунок 2.5 – Діаграма класів сервісного рівня інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

На рисунку 2.6 показано діаграму класів «Product», «Order» та «S3» сервісів.

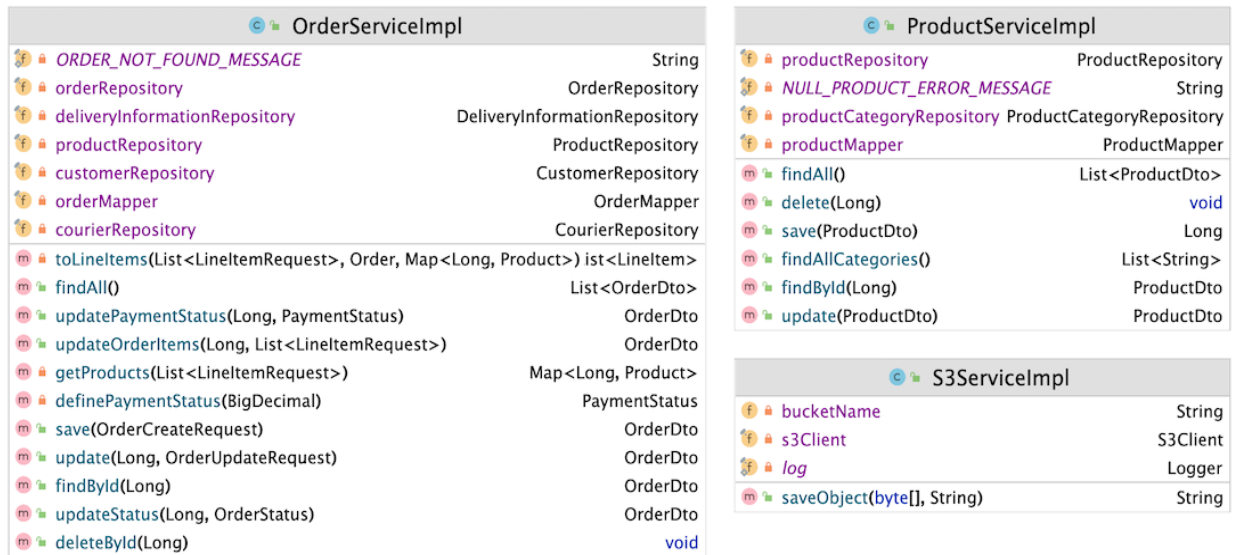


Рисунок 2.6 – Діаграма класів сервісного рівня інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

В лістингу 2.3 показано код класу `S3ServiceImpl` який реалізовує інтерфейс `S3Service` і відповідає за взаємодію веб-серверу платформи із AWS S3 [21].

Лістинг 2.3 – `S3ServiceImpl.java`

```
@Log4j2
@Service
@RequiredArgsConstructor
public class S3ServiceImpl implements S3Service {

    @Value("${aws.s3.bucket}")
    private String bucketName;

    private final S3Client s3Client;

    @Override
    public String saveObject(byte[] data, String key) {
        PutObjectRequest putObjectRequest =
        PutObjectRequest.builder()
            .bucket(bucketName)
            .key(key)
```

```

        .build();

        log.info("Uploading image: {} to the s3 bucket", key);
        s3Client.putObject(putObjectRequest,
            RequestBody.fromBytes(data));
        return s3Client.utilities().getUrl(
            GetUrlRequest.builder().bucket(bucketName).key(key).build()).toString();
    }
}

```

2.1.3 Розробка рівня доступу до даних платформи

Рівень доступу до даних відповідає за взаємодію із базою даних, аболюбим іншим типом сховищ даних. Дані платформи знаходяться у базі даних PostgreSQL, для взаємодії із базою використовуються інтерфейси репозиторіїв (шаблон проектування Repository) [22] більшість методів яких реалізовує «із коробки» Spring Data JPA [23]. Кожен репозиторій відповідає за певну сутність (entity) у базі даних, а кожна сутність відноситься до певної таблиці у базі даних, для встановлення зв'язків між сутностями та таблицями у базі даних використовуються анотації із специфікації Java Persistence API (Див. Лістинг 2.5).

Лістинг 2.5 – ProductCategory.java

```

@Entity
@Getter
@Setter
@NoArgsConstructor
@Table(name = "category")
public class ProductCategory {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "name", unique = true)
    private String name;

    @OneToMany(mappedBy = "category")
    private List<Product> products;
}

```


Анотація @Entity [24] сповіщає ORM фреймворк про те, що даний клас відноситься до певної таблиці у базі даних, анотація @OneToMany вказує на те, що зв'язок між категорією товару та товаром є один до багатьох.

Інтерфейси репозиторіїв завдяки Spring Data JPA фактично не потребують імплементації, адже фреймворк надає готові реалізації основних методів, також за допомогою анотації @Query (Див. Лістинг 2.6) можна створювати кастомізовані запит до бази даних використовуючи мову JPQL [25].

Лістинг 2.6 – CustomerRepository.java

```
@Repository
public interface CustomerRepository extends
JpaRepository<Customer, Long> {
    @Query("SELECT c.instagram from Customer c WHERE c.createdAt
> :createdAt AND c.orders.size > :ordersCount")
    List<String> findCustomerBy(LocalDateTime createdAt, int
ordersCount);
}
```

2.2 Реалізація користувацького інтерфейсу інформаційно-технологічної платформи

Для реалізації сучасного та адаптивного інтерфейсу користувача була використана бібліотека React JS у зв'язці з Material UI. Material UI - це бібліотека розроблена компанією Google, яка надає широкий асортимент компонентів для користувацького інтерфейсу [26]. На рисунку 2.7 зображено сторінку яку бачить користувач одразу після входу в систему, це сторінка замовлень. Таблиця замовлень є функціональною і дозволяє сортувати дані по різним колонкам та робити посторінковий перегляд, що є зручно при великій кількості інформації. Також у шапці таблиці є можливість перегляду замовлень фільтрування замовлень по статусу готовності замовлення.

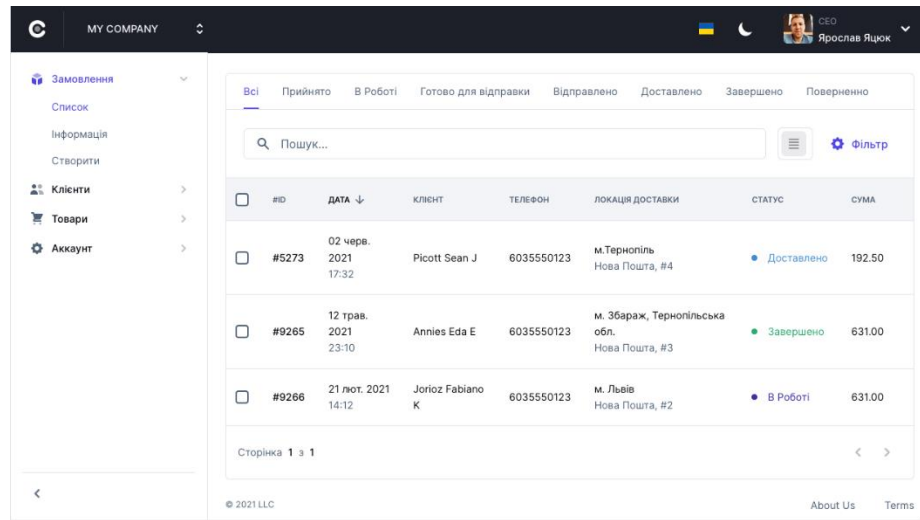


Рисунок 2.7 – Сторінка замовлень інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

На рисунку 2.8 можна побачити як реалізовано сторінку інформації про замовлення. Користувач платформи на цій сторінці може редагувати статус замовлення, інформації про замовлення та список товарів у замовленні, також МОЖЛИВО ПОЗНАЧИТИ ЗАМОВЛЕННЯ ЯК ОПЛАЧЕНЕ.

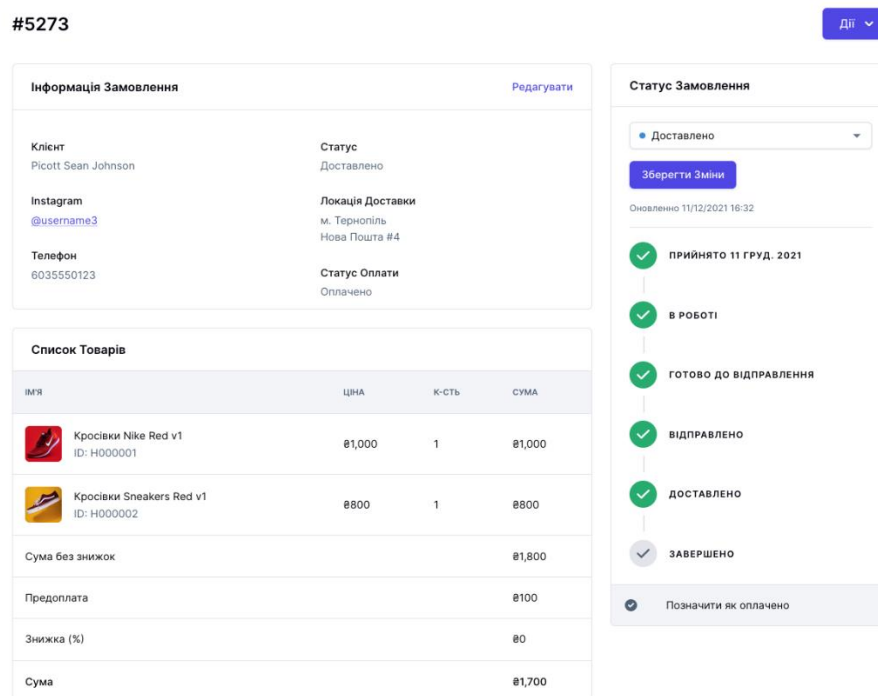


Рисунок 2.8 – Сторінка інформації про замовлення інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

Для редагування даних клієнта була розроблена спеціальна сторінка (Див рисунок 2.9) із інформацією про кожного окремого клієнта який є збережений у системі. На даній сторінці можливо також залишати коментарі про клієнта, наприклад коли краще зв'язатися із певним клієнтом, чи були проблеми із клієнтом тощо. Для відображення статистичної інформації по покупкам окремого клієнта була розроблена спеціальна панель із інформацією про дату реєстрації, кількості покупок та загальної кількості витрат цього клієнта.

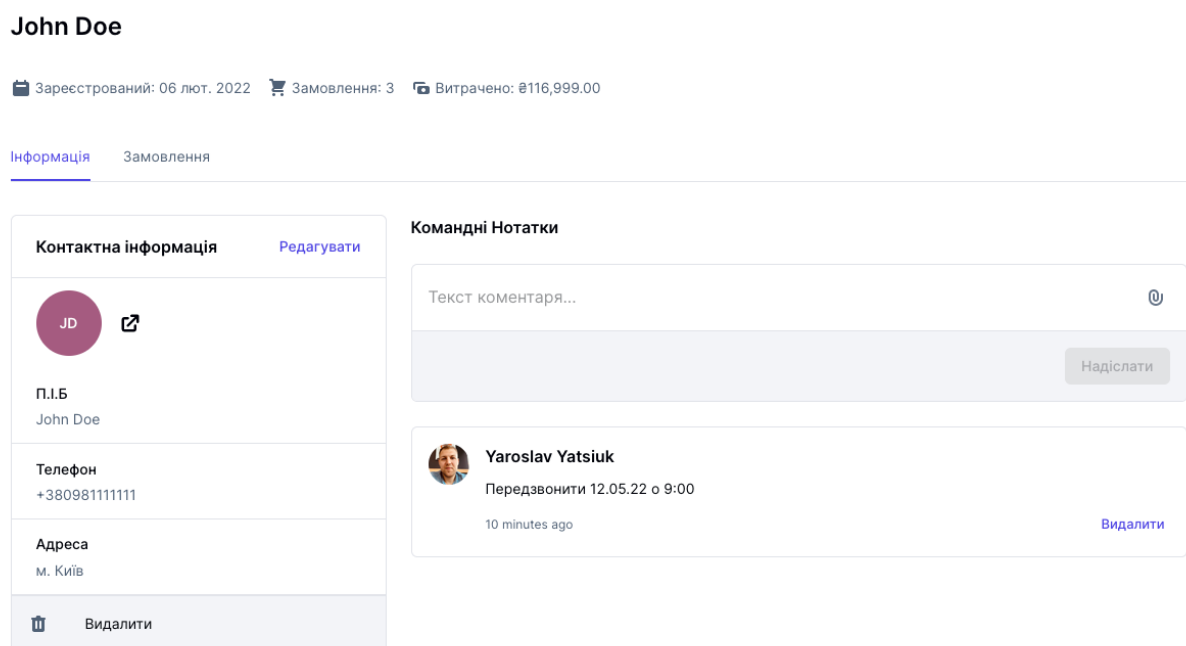


Рисунок 2.9 – Сторінка інформації про клієнта інформаційно-технологічної платформи для інтернет-торгівлі на базі Instagram

Хорошою практикою є валідація полів вводу даних [27] як на серверній частині так і на стороні клієнта, це дозволяє попередити помилки в системі ще на етапі введення даних на клієнті, також валідація робить інтерфейс зрозумілим для користувача, адже система миттєво повідомляє якщо якийсь поле було введено не вірно, або поле є обов'язковим для введення. Вікна вводу повинні змінювати колір та надавати інформативні повідомлення при не

вірному вводу даних. На рисунку 2.10 показано сторінку створення замовлення з англійською локалізацією при ситуації не вірного вводу даних.

← Order

Create Order

Customer
Shelby O'Docherty

Delivery Address
City/Village, District, Region
Delivery Address is required

Courier
Nova Poshta

Branch Number
12
Branch number is required

Phone
+38068111111111asd
phone must be a valid phone number for region UA

Prepayment
-100
Value cannot be less than 0

Item	Qty	Total
Timely Cyan Strap 7	1	€ 1027.45
Unbranded Sneakers 4	1	€ 114.4

+ Add Item

Final Total €1,241.85

Additional Notes

Create Order

Рисунок 2.10 – Сторінка створення замовлення, англійська версія

Всі текстові підказки та застереження повинні відповідати мові інтерфейсу, дана платформа має підтримку української та англійської мов інтерфейсу тому відповідно текст підказок змінюється в залежності від вибраної мови.

2.3 Тестування інформаційно-технологічної платформи

Для тестування серверної частини платформи були написані модульні (unit) та інтеграційні (integration) тести. Для написання інтеграційних тестів було використано популярний зараз фреймворк testcontainers [29] який налаштовує та запускає перед стартом тестів всю необхідну інфраструктуру,

таку як бази даних, брокери повідомлень, тощо. Модульними тестами [28] була покрита бізнес логіка платформи, тобто сервісний рівень, всі решту рівнів були mocked за допомогою бібліотеки Mockito. В лістингу 2.7 зображено код модульного тесту на додавання нового клієнта.

Лістинг 2.7 – CreateCustomerTest.java

```
@Test
void shouldCreateNewCustomer() {
    CustomerDto customerDto = new CustomerDto();
    customerDto.setFullName(FULL_NAME);
    customerDto.setInstagram(INSTAGRAM);
    customerDto.setPhone(PHONE);

    Long id = customerService.save(customerDto);
    assertNotNull(id);

    verify(customerRepository).save(customerDtoArgumentCaptor.capture());
    Customer customer = customerDtoArgumentCaptor.getValue();
    assertEquals(FULL_NAME, customer.getFullName()),
    assertEquals(INSTAGRAM, customer.getInstagram()),
    assertEquals(PHONE, customer.getPhone())
};
}
```

Рівень доступу до даних було протестовано інтеграційними тестами, які допомагають виявляти n+1 проблеми в JPA, та помилки в SQL запитах. Незважаючи на те, що інтеграційні тести займають набагато більше часу на тестування в порівнянні з модульними тестами вони перевіряють взаємодію відразу декількох модулів системи і максимально точно імітують роботу програмних засобів в реальних умовах.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Долікарська допомога при контузіях

Контузія може з'явитися від ударної хвилі від вибуху, або удару людини об воду чи землю. Контузія може супроводжуватися наступними травмами: забій головного мозку, розрив барабанної перетинки, легень або інших внутрішніх органів [30]. Залежно від ступеня тяжкості виділяють три ступеня контузії:

- Легка – заїкання, тремтіння або поколювання кінцівок, погіршення слуху.
- Середньої тяжкості – погіршення зору, відсутність реакції зіниць на світло, повна глухота, частковий параліч кінцівок.
- Важка – втрата свідомості, переривчасте дихання, судоми чи мимовільні рухи кінцівками, може спостерігатися кровотеча з рота і вух.

Легка контузія зазвичай проходить без негативних наслідків для організму. Середня і важка – ні, тому якщо постраждалу особу вдалося врятувати, на неї чекає тривалий період реабілітації. У будь-якому випадку потрібно викликати швидку допомогу.

Перед приїздом швидкої допомоги потрібно зробити наступні дії:

- Перш за все, покинути небезпечну зону, якщо удар спричинено вибухом.
- Розстебнути одяг жертви, що може стискати дихальні шляхи або ж обмежувати рухливість.
- Бажано повернути постраждалу особу набік, аби запобігти можливому потраплянню блювотних мас в дихальні шляхи.
- Очистити носову та ротову порожнини жертви від води, пилу, піску чи осколків вибуху.

– Якщо жертва не дихає або має проблеми з цим, варто зробити штучне дихання. Непрямий масаж серця в такому разі протипоказаний — внутрішні органи можуть бути пошкоджені, тому ритмічні натискання в область грудної клітини можуть лише погіршити стан постраждалого.

3.2 Вимоги пожежної безпеки при гасінні електроустановок.

Гасіння пожежі в електроустановках необхідно проводити тільки вуглекислотними вогнегасниками з попереднім зняттям напруги з електроустановки. Гасіння пожежі компактними і розпиленими водяними струменями без зняття напруги з електроустановок допускається тільки у відкритих для огляду ствольщика електроустановках, в т. ч. кабелів, що горять, при номінальній напрузі до 10 кВ. При цьому ствол повинен бути заземлений, а ствольщик повинен працювати в діелектричних ботах та діелектричних рукавицях і знаходитись на віддалі до електроустановки, що горить, не менше:

- 4 м в електроустановках до 1000 В;
- 8 м в електроустановках до 6-10 кВ.

Використання морської і сильно забрудненої води при цьому не допускаються. Гасити пожежу ручними засобами без зняття напруги з електроустановок в сильно задимлених приміщеннях електроустановок не допускається. Гасіння пожежі в електроустановках, що знаходяться під напругою до 10 кВ, всіма видами пін за допомогою ручних засобів забороняється, так як піна і розчин піноутворювача у воді має підвищену електропровідність [31].

В окремих випадках дозволяється гасіння пожежі в електроустановках, що знаходяться під напругою до 10 кВ, повітряно-механічною піною:

- якщо генератор високочастотної піни, встановлений попередньо, надійно заземлений;

- заземлені насоси пожежних машин.

Якщо по умовах режиму роботи електроустановки при гасінні пожежі частина приєднань не відключена, то для недопущення ураження електрострумом (при випадковому дотику або недопустимому наближенню людини до струмоведучих частин) віддаль повинна бути не менше:

- при напрузі 6-10 кВ – 0,7 м;
- при напрузі 35 кВ – 1 м;
- при напрузі 110 кВ – 1,5 м.

Пожежні підрозділи приступають до гасіння пожежі в електроустановці після проведення їм інструктажу головним енергетиком, інженер-електриком (черговим електриком) і одержання від них дозволу на гасіння пожежі, при цьому повинні оцінити всі можливі ситуації і наслідки від можливих коротких замикань [32].

При гасінні пожежі робота пожежних підрозділів проводиться з урахуванням вказівок головного енергетика, інженер-електрика (чергового електрика), які зобов'язані давати порадивказівки керівникові гасіння пожежі по дотриманню правил безпеки, про можливе загорання поруч розміщеної апаратури і поширення пожежі. Керівник гасіння пожежі повинен узгоджувати з головним енергетиком, інженер-електриком. (черговим електриком) свої дії по розстановці сил і засобів пожежогасіння, зміну позицій пожежних стволів і т. д.

Недопустимо проникнення особового складу пожежних підрозділів за огороження струмоведучих частин, що знаходяться під напругою. Особовий склад пожежних підрозділів під час гасіння пожежі повинен вважати, що всі струмоведучі частини електроустановки знаходяться під напругою, за виключенням тих струмоведучих частин, відсутність напруги на яких підтверджено оперативним персоналом [33].

ВИСНОВКИ

В кваліфікаційній роботі було розроблено інформаційно-технологічну платформу для автоматизації торгівлі у мережі Instagram. Розроблена платформа має всі функціональні можливості для забезпечення обліку продаж, відслідковування статусу замовлень, створення нових замовлень, редагування існуючих замовлень, та управління товарами та клієнтами.

У першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- розглянуто основні функціональні можливості платформи;
- спроектовано серверну, клієнтську, хмарну та узагальнену архітектуру платформи;
- обґрунтовано вибір стеку технологій для реалізації функціоналу та архітектури платформи.

У другому розділі кваліфікаційної роботи розглянуто деталі програмно-алгоритмічної реалізації і тестування серверної та клієнтської частин інформаційно-технологічної платформи, а саме:

- спроектовано трьох-рівневу архітектуру сервера;
- розроблено користувацький інтерфейс Single Page Application (SPA);
- проведено тестування платформи за допомогою модульних та інтеграційних тестів.

У розділі «Безпека життєдіяльності, основи охорони праці» було розглянуто:

- основи надання першої медичної допомоги при контузіях;
- вимоги пожежної безпеки при гасінні електроустановок.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Richardson C. *Microservices Patterns: With examples in Java*. Manning Publications, 2018. 520 стр.
- 2 Kleppmann M. *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media, 2017. 624 p.
- 3 Deutch K. *What Is a POS System And How Does It Work?* Square. URL: <https://squareup.com/us/en/townsquare/what-pos-system>
- 4 MUI: The React component library you always wanted. URL: <https://mui.com/>
- 5 Getting Started – React. React – A JavaScript library for building user interfaces. URL: <https://reactjs.org/docs/getting-started.html>
- 6 Bloch J. *Effective Java*. Addison-Wesley Professional, 2018. 412 стр.
- 7 Spring Framework Documentation. Spring | Home. URL: <https://docs.spring.io/spring-framework/docs/current/reference/html/>
- 8 PostgreSQL: Documentation. PostgreSQL: The world's most advanced open source database. URL: <https://www.postgresql.org/docs/>
- 9 Wittig A., Wittig M. *Amazon Web Services in Action*. Manning Publications, 2015. 424 стр.
- 10 dbdiagram.io - Database Relationship Diagrams Design Tool. URL: <https://dbdiagram.io/>
- 11 Wittig A., Wittig M. *Amazon Web Services in Action*. Manning Publications, 2015. 424 стр.
- 12 Richards M., Ford N. *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly Media, 2020. 432 p.
- 13 What is a Loosely Coupled System?. The Customize Windows. URL: <https://thecustomizewindows.com/2021/01/what-is-a-loosely-coupled-system/>
- 14 RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1. IETF Datatracker. URL: <https://datatracker.ietf.org/doc/html/rfc2616>

- 15 What Are RESTful Web Services? - The Java EE 6 Tutorial. Moved. URL: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>
- 16 P of EAA: Service Layer. martinowler.com. URL: <https://martinowler.com/eaCatalog/serviceLayer.html>
- 17 What is a web server? - Learn web development | MDN. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server
- 18 HTTP Status Codes - REST API Tutorial. REST API Tutorial. URL: <https://restfulapi.net/http-status-codes/>
- 19 P of EAA: Domain Model. martinowler.com. URL: <https://martinowler.com/eaCatalog/domainModel.html>
- 20 The DTO Pattern (Data Transfer Object) | Baeldung. Baeldung. URL: <https://www.baeldung.com/java-dto-pattern>
- 21 What is Amazon S3? - Amazon Simple Storage Service. URL: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>
- 22 Bergman P.-E. Repository Design Pattern. Medium. URL: <https://medium.com/@pererikbergman/repository-design-pattern-e28c0f3e4a30>
- 23 Spring Data JPA - Reference Documentation. Spring | Home. URL: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#reference>
- 24 JPA Tutorial. Biggest Online Tutorials Library. URL: <https://www.tutorialspoint.com/jpa/index.htm>
- 25 JPQL - How to Define Queries in JPA and Hibernate. Thorben Janssen. URL: <https://thorben-janssen.com/jpql/>
- 26 MUI: The React component library you always wanted. URL: <https://mui.com/>
- 27 Validation | Formik. Formik: Build forms in React, without the tears. URL: <https://formik.org/docs/guides/validation>
- 28 Khorikov V. Unit Testing Principles, Practices, and Patterns. Manning Publications Company, 2020. 304 p.

29 Integration Testing with Testcontainers :: Debezium Documentation. Debezium. URL: <https://debezium.io/documentation/reference/stable/integrations/testcontainers.html>

30 Редакція Т. Контузія - симптоми та перша допомога. ТСН.ua. URL: <https://tsn.ua/ukrayina/kontuziya-simptomi-ta-persha-dopomoga-2022373.html>

31 Гасіння пожеж на енергетичних об'єктах під напругою. Охорона праці і пожежна безпека. URL: <https://oppb.com.ua/docs/gasinnya-pozhezh-na-energetichnih-objektah-pid-naprugoyu>

32 Яким вогнегасником можна гасити електрообладнання з техніки безпеки? - Zprim. Zprim. URL: <http://zprim.com.ua/yakim-vognegasnikom-mozhna-gasiti-elektroobladnannya-z-tehniki-bezpeki/>

33 Про затвердження Інструкції з гасіння пожеж на енергетичних об'єктах України. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/z0013-12#>

ДОДАТКИ

Програмний код лістингу pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.2</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.yyatsiuk.api.core</groupId>
  <artifactId>hs-pos-core-api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>hs-pos-core-api</name>
  <description>hs-pos-core-api</description>
  <properties>
    <java.version>17</java.version>
    <org.projectlombok.version>1.18.22</org.projectlombok.version>
    <amazon.sdk.version>2.17.181</amazon.sdk.version>
    <org.mapstruct.version>1.5.0.Beta2</org.mapstruct.version>
    <org.projectlombok.version>1.18.20</org.projectlombok.version>
    <lombok-mapstruct-binding.version>0.2.0</lombok-mapstruct-
binding.version>
    <thumbnailator.version>0.4.17</thumbnailator.version>
    <google.guava.version>31.0.1-jre</google.guava.version>
    <jsonwebtoken.version>0.9.1</jsonwebtoken.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-actuator</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>>true</optional>
    </dependency>
    <dependency>
      <groupId>org.hibernate.validator</groupId>
      <artifactId>hibernate-validator</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-security</artifactId>
    </dependency>
  </dependencies>

```

```

<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt</artifactId>
  <version>${jsonwebtoken.version}</version>
</dependency>

<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-configuration-processor</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>s3</artifactId>
  <version>${amazon.sdk.version}</version>
</dependency>
<dependency>
  <groupId>net.coobird</groupId>
  <artifactId>thumbnailator</artifactId>
  <version>${thumbnailator.version}</version>
</dependency>
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
</dependency>
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>${google.guava.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.mapstruct</groupId>
  <artifactId>mapstruct</artifactId>
  <version>${org.mapstruct.version}</version>
</dependency>
</dependencies>
</project>

```

Програмний код лістингу package.json

```
{
  "name": "hs-pos-ui",
  "author": "yyatsiuk",
  "version": "1.1.0",
  "scripts": {
    "lint": "eslint ./src",
    "lint-fix": "eslint --fix ./src",
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "babel": {
    "presets": [
      "@babel/preset-react"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "dependencies": {
    "@auth0/auth0-spa-js": "^1.16.0",
    "@emotion/cache": "^11.4.0",
    "@emotion/react": "^11.4.0",
    "@emotion/styled": "^11.3.0",
    "@material-ui/core": "^5.0.0-beta.2",
    "@material-ui/icons": "^5.0.0-beta.1",
    "@material-ui/lab": "^5.0.0-alpha.41",
    "@react-pdf/renderer": "^2.0.16",
    "apexcharts": "^3.27.1",
    "date-fns": "^2.22.1",
    "formik": "^2.2.9",
    "gray-matter": "^4.0.3",
    "history": "^5.0.0",
    "il8next": "^20.3.5",
    "jwt-decode": "^3.1.2",
    "nprogress": "^0.2.0",
    "numeral": "^2.0.6",
    "prop-types": "^15.7.2",
    "react": "^17.0.2",
    "react-apexcharts": "^1.3.9",
    "react-beautiful-dnd": "^13.1.0",
    "react-dom": "^17.0.2",
    "react-dropzone": "^11.3.2",
    "react-helmet-async": "^1.0.9",
    "react-hot-toast": "^2.0.0",
    "react-il8next": "^11.11.4",
```



```
"react-markdown": "^5.0.3",
"react-router-dom": "^6.2.1",
"react-syntax-highlighter": "^15.4.3",
"simplebar": "^5.3.3",
"simplebar-react": "^2.3.3",
"stylis-plugin-rtl": "^2.1.0",
"yup": "^0.32.9",
"yup-phone": "^1.3.2"
},
"devDependencies": {
  "@babel/core": "^7.14.8",
  "@babel/eslint-parser": "^7.14.7",
  "@babel/preset-react": "^7.14.5",
  "eslint": "^7.31.0",
  "eslint-config-airbnb": "^18.2.1",
  "eslint-config-airbnb-typescript": "^12.3.1",
  "eslint-plugin-import": "^2.23.4",
  "eslint-plugin-jsx-ally": "^6.4.1",
  "eslint-plugin-react": "^7.28.0",
  "eslint-plugin-react-hooks": "^4.2.0",
  "react-scripts": "4.0.3"
}
}
```