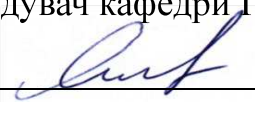


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

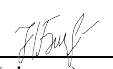
на здобуття ступеня вищої освіти "бакалавр"

на тему: Розроблення підсистеми бекенду інформаційної системи мережі
магазинів одягу "Південний стиль"

Виконав: студент групи


_____ Новицький В.К.
(підпис)

Керівник роботи:

(посада, науковий ступінь вчене звання)

_____ Беркунська Н.М.
(підпис)

м. Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

Спеціальність 124 "Системний аналіз"

Освітня програма "Системний аналіз"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми



к.т.н, доц. Гайдаєнко О. В.

" ____ " _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "бакалавр"

Студенту Новицькому Валерію Костянтиновичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення системи бекенду інформаційної системи мережі магазинів одягу "Південний стиль" _____

Керівник роботи Беркунська Наталія Михайлівна _____

Затверджені наказом ректора № 295-уч від "27" березня 2025 року.

2. Термін подання роботи: 20 червня 2025 р. _____

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем. _____

4. Перелік питань, що належать до розробки (найменування розділів): Розробка концепції інформаційної системи; Розробка проекту; Реалізація проекту; Розділ з охорони праці; Висновки; Додатки (Технічне завдання, Інструкція для системного адміністратора з розгортання та налаштування системи, Повний перелік ендпоінтів) _____

5. Перелік презентаційних матеріалів: 1. Мета роботи. 2. Завдання роботи. 3. Об'єкт дослідження. 4. Організаційна структура мережі магазинів одягу «Південний стиль». 5. Концепція інформаційної системи. 6. Діаграма варіантів використання. 7. Концептуальна модель даних. 8. Логічна модель даних. 9. Фізична модель даних. 10. Програмні засоби. 11. Робота застосунку. 12. Тестування застосунку. 13. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Беркунська Н.М.	03.02.2025	22.02.2025
2	Беркунська Н.М.	22.02.2025	21.03.2025
3	Беркунська Н.М.	21.03.2025	25.04.2025
4	Беркунська Н.М.	25.04.2025	07.06.2025

7. Дата видачі завдання 03 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі	13.02.2025	
2	Постановка задачі	15.02.2025	
3	Розробка концепції	22.02.2025	
4	Розробка проекту ІС	21.03.2025	
5	Реалізація проекту	25.04.2025	
6	Розробка питань з охорони праці	07.06.2025	
7	Розробка графічних матеріалів	12.06.2025	
8	Подання роботи на перевірку на плагіат	10.06.2025	
9	Подання роботи рецензенту	15.06.2025	
10	Подання роботи на захист	20.06.2025	

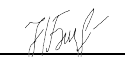
Студент


(підпис)

Новицький В.К.

(ПІБ)

Керівник роботи


(підпис)

Беркунська Н.М.

(ПІБ)

АНОТАЦІЯ

Тема дипломного проекту: «Розроблення підсистеми бекенду інформаційної системи мережі магазинів одягу "Південний стиль".»

Актуальність даної теми зумовлена сучасними тенденціями цифрової трансформації бізнесу, що вимагають автоматизації ключових процесів у роздрібній торгівлі для підвищення ефективності та рентабельності.

Мета дипломного проекту полягає в тому, щоб розробити надійну підсистему бекенду, яка автоматизує управління товарними запасами, продажами, закупівлями та аналітикою для мережі магазинів одягу "Південний стиль".

Мережа "Південний стиль" є роздрібною торговельною організацією з 5 магазинами, 2 складами та головним офісом, що спеціалізується на продажу чоловічого, жіночого та дитячого одягу.

У ході роботи проведено аналіз предметної області, сформульовано вимоги до системи, розроблено концепцію і реалізовано необхідний функціонал системи.

Робота виконана на 73 сторінки, містить 13 рисунків, 26 таблиць, 3 додатки, список використаних джерел з 17 найменувань.

Робота виконана українською мовою.

Ключові слова: бекенд, інформаційна система, Spring Boot, Java, MySQL, торгівля одягом, API, автоматизація обліку.

SUMMARY

The topic of the diploma project: "Development of a backend subsystem of the information system of the clothing store chain "Southern Style".

The relevance of this topic is due to modern trends in digital business transformation, which require automation of key processes in retail trade to increase efficiency and profitability.

The goal of the diploma project is to develop a reliable backend subsystem that automates inventory management, sales, purchases and analytics for the clothing store chain "Southern Style".

The chain "Southern Style" is a retail organization with 5 stores, 2 warehouses and a head office, specializing in the sale of men's, women's and children's clothing.

During the work, an analysis of the subject area was conducted, system requirements were formulated, a concept was developed and the necessary system functionality was implemented.

The work is 91 pages long, contains 26 figures, 6 tables, 4 diagrams, 3 models, 4 appendices, a list of sources used with 11 titles.

The work is done in Ukrainian.

Keywords: backend, information system, Spring Boot, Java, MySQL, clothing trade, API, accounting automation.

ЗМІСТ

АНОТАЦІЯ	4
SUMMARY	5
ЗМІСТ	6
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП	8
РОЗДІЛ 1. РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕРЕЖІ МАГАЗИНІВ ОДЯГУ «ПІВДЕННИЙ СТИЛЬ»	9
1.1 Опис об'єкта дослідження та аналіз предметної області	10
1.2 Аналіз існуючих рішень	13
1.3 Формування вимог до системи	19
1.4 Розробка концепції інформаційної системи.....	22
1.5 Постановка задачі	25
РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ ПІДСИСТЕМИ БЕКЕНДУ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕРЕЖІ МАГАЗИНІВ ОДЯГУ «ПІВДЕННИЙ СТИЛЬ»	27
2.1 Загальносистемні рішення.	27
2.2. Рішення з інформаційного забезпечення.....	29
2.3 Рішення щодо програмного забезпечення.....	41
2.4 Рішення з технічного забезпечення	44
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ ПІДСИСТЕМИ БЕКЕНДУ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕРЕЖІ МАГАЗИНІВ ОДЯГУ «ПІВДЕННИЙ СТИЛЬ»	46
Розділ 4. Охорона праці.....	49
ВИСНОВОК.....	53
ПЕРЕЛІК ПОСИЛАНЬ	54
Додаток А.....	56
Додаток Б	59
Додаток В.....	64

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

СМ – системний аналіз.

СУБД - система управління базами даних.

БД – база даних.

ІС – інформаційна система.

ОС – операційна система.

ВСТУП

У контексті сучасної цифрової трансформації бізнес-середовища суб'єкти роздрібної торгівлі, зокрема мережі магазинів одягу, активно інтегрують інформаційні системи з метою автоматизації ключових операційних процесів. Рациональне управління товарними ресурсами, обробка замовлень, ведення клієнтської бази, аналітика продажів і організація внутрішньої комунікації потребують комплексного та системного підходу до проєктування програмного забезпечення. Особливої актуальності набуває розробка адаптивних, масштабованих та ефективних архітектурних рішень, які забезпечують цілісність і узгодженість функціонування інформаційної інфраструктури підприємства.

Актуальність обраної тематики зумовлена зростаючою потребою комерційних структур у впровадженні індивідуалізованих цифрових рішень, здатних враховувати галузеву специфіку та мінімізувати залежність від стороннього програмного забезпечення, що, у свою чергу, сприяє підвищенню рівня автономності й ефективності управління. Аналіз наукових джерел та прикладних реалізацій підтверджує тенденцію активного розвитку технологій побудови бекенд-архітектур, зокрема використання REST-підходів, мікросервісної архітектури, інструментів аутентифікації та взаємодії з базами даних. Проте більшість наявних рішень вимагають адаптації або подальшої розробки відповідно до конкретних вимог бізнес-логіки, що зумовлює доцільність створення спеціалізованого програмного продукту.

Об'єктом дослідження у межах даної роботи виступає інформаційна система для управління мережею роздрібної торгівлі одягом. Предметом дослідження є процес проєктування та реалізації серверної частини (бекенду) зазначеної системи.

Метою дипломного дослідження є розробка ефективного серверного компонента інформаційної системи для мережі магазинів одягу «Південний

стиль», що забезпечуватиме автоматизацію процесів управління товарними залишками, продажами, закупівлями та бізнес-аналітикою.

Для досягнення поставленої мети були сформульовані наступні завдання:

- провести аналіз предметної області та огляд існуючих рішень;
- визначити вимоги до архітектури майбутньої системи;
- дослідити і обрати засоби розробки;
- спроектувати структуру бази даних і архітектуру системи;
- реалізувати RESTful API для основних функціональних компонентів;
- інтегрувати механізми захисту даних та авторизації користувачів;
- виконати тестування і валідацію розробленого рішення.

Реалізація вищезазначених завдань забезпечить створення технологічно цілісної та функціонально повної основи для подальшої інтеграції з фронтенд-частиною, модулями аналітики й іншими підсистемами, а також сприятиме оптимізації діяльності підприємства й підвищенню ефективності управління мережею магазинів одягу «Південний стиль».

РОЗДІЛ 1. РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕРЕЖІ МАГАЗИНІВ ОДЯГУ «ПІВДЕННИЙ СТИЛЬ»

1.1 Опис об'єкта дослідження та аналіз предметної області

Мережа магазинів одягу «Південний стиль» здійснює діяльність у сфері роздрібної торгівлі одягом, забезпечуючи споживачів асортиментом чоловічого, жіночого та дитячого одягу. Основною метою функціонування мережі є задоволення попиту клієнтів на якісний, сучасний та доступний за ціною одяг.

Мережа має у своєму розпорядженні 5 магазинів, 2 склади та головний офіс.

До основних функцій магазину належать:

- роздрібний продаж одягу;
- організація закупівель продукції у постачальників;
- маркетингова діяльність та просування бренду;
- ведення фінансового та бухгалтерського обліку;
- управління складськими запасами;
- кадрове забезпечення діяльності компанії.

Організаційна структура мережі магазинів одягу «Південний стиль» представлена на рисунку 1.1

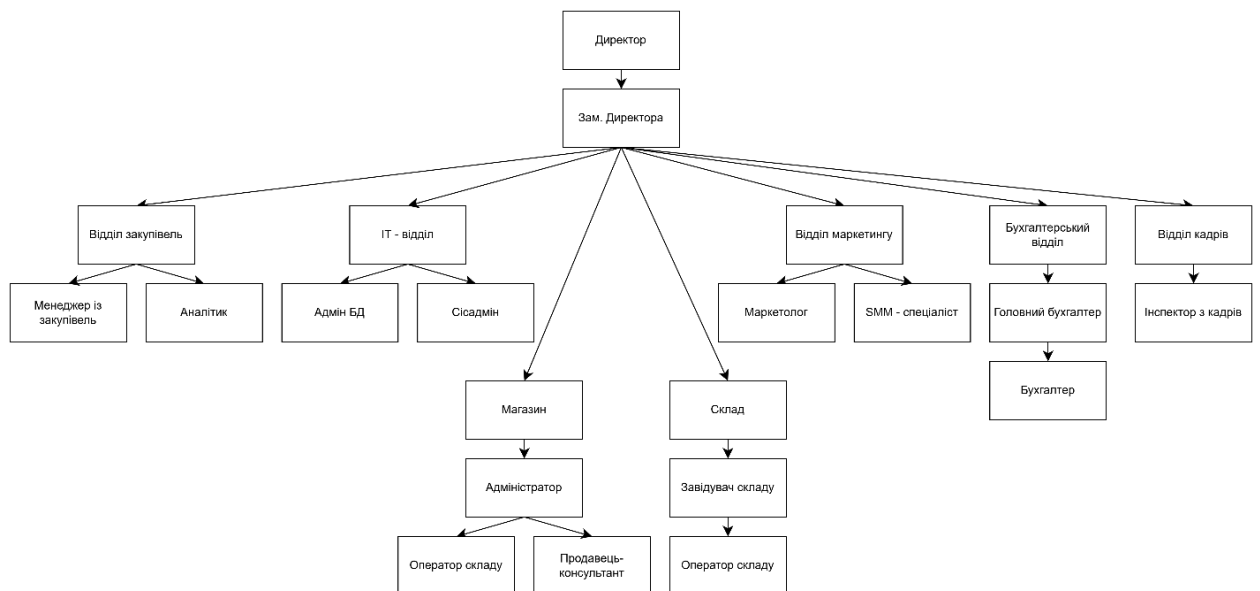


Рисунок 1. 1 Організаційна структура мережі магазинів одягу
«Південний стиль»

Директор керує загальною діяльністю компанії, визначає стратегічні цілі та приймає ключові управлінські рішення.

Замісник директора координує оперативну діяльність, контролює роботу всіх структурних підрозділів, сприяє реалізації стратегічних цілей компанії.

Відділ закупівель здійснює аналітику ринку, взаємодію з постачальниками, оформлення замовлень та контроль їх виконання.

ІТ-відділ складається з системного адміністратора, який забезпечує функціонування технічної інфраструктури, та адміністратора баз даних, відповідального за збереження і цілісність даних у електронних таблицях, а також за резервне копіювання й контроль доступу.

Маркетинговий відділ займається формуванням рекламних кампаній, просуванням бренду та підтримкою іміджу компанії в інформаційному просторі.

Бухгалтерський підрозділ відповідає за ведення фінансової документації, складання звітності, розрахунок заробітної плати та податків. Відділ кадрів здійснює підбір персоналу, ведення кадрової документації та контроль дотримання трудового законодавства.

На рівні окремих магазинів продавці-консультанти безпосередньо обслуговують клієнтів, здійснюючи продаж товарів та надаючи консультації. Адміністратори забезпечують безперебійну роботу магазинів і своєчасну звітність. Оператори складів відповідають за облік товару та його розміщення, а також надають підтримку продавцям за потреби.

Склади підприємства виконують функції прийому, зберігання та видачі продукції, а також ведення обліку товарних залишків.

Об'єктом дослідження є бізнес-процеси підприємства, до яких належать: продажі, закупівлі, маркетингова діяльність, фінансовий облік, управління запасами та кадрове адміністрування. Попри сформовану організаційну структуру, підприємство стикається з проблемою низької рентабельності, що свідчить про недосконалість існуючої моделі управління та низький рівень автоматизації.

Розглянемо основні проблеми, що негативно впливають на ефективність функціонування підприємства.

Більшість внутрішньої документації ведеться вручну із застосуванням текстових і табличних редакторів (Excel, Word), що спричиняє значні часові витрати, підвищує ризики помилок і ускладнює контроль за виконанням операцій.

Відсутність єдиної інтегрованої платформи для обміну даними між відділами призводить до фрагментарної комунікації, яка здійснюється переважно через електронну пошту або телефон, що ускладнює оперативне управління.

Формування звітності відбувається вручну, що не лише уповільнює процес прийняття управлінських рішень, а й підвищує ризик інформаційних затримок і неточностей.

Недостатня автоматизація складського обліку зумовлює втрати продукції, затримки в постачанні й порушення обліку товарних залишків.

Відсутність системного аналізу попиту ускладнює планування поставок і своєчасне поповнення асортименту, особливо в сезонний період, що призводить до втрати потенційних продажів

Визначення доцільності знижок і маркетингових акцій здійснюється без належного обґрунтування, що знижує ефективність цінової політики та не дозволяє вчасно реагувати на зміни ринку.

Виявлені проблеми підтверджують необхідність впровадження автоматизованої інформаційної системи, яка дозволить забезпечити інтеграцію всіх бізнес-процесів, підвищити ефективність управління ресурсами та покращити загальні фінансово-економічні показники підприємства.

1.2 Аналіз існуючих рішень

Для кращого розуміння вимог до системи важливо дослідити популярні рішення, що є на ринку

Dilovod – це український хмарний сервіс, призначений для ведення бухгалтерського та управлінського обліку, орієнтований на малі та середні підприємства, зокрема в сфері роздрібно́ї торгівлі, виробництва та послуг. Основна мета сервісу – спростити облік і звітність, ставши альтернативою таким системам, як 1С чи BAS. [1]

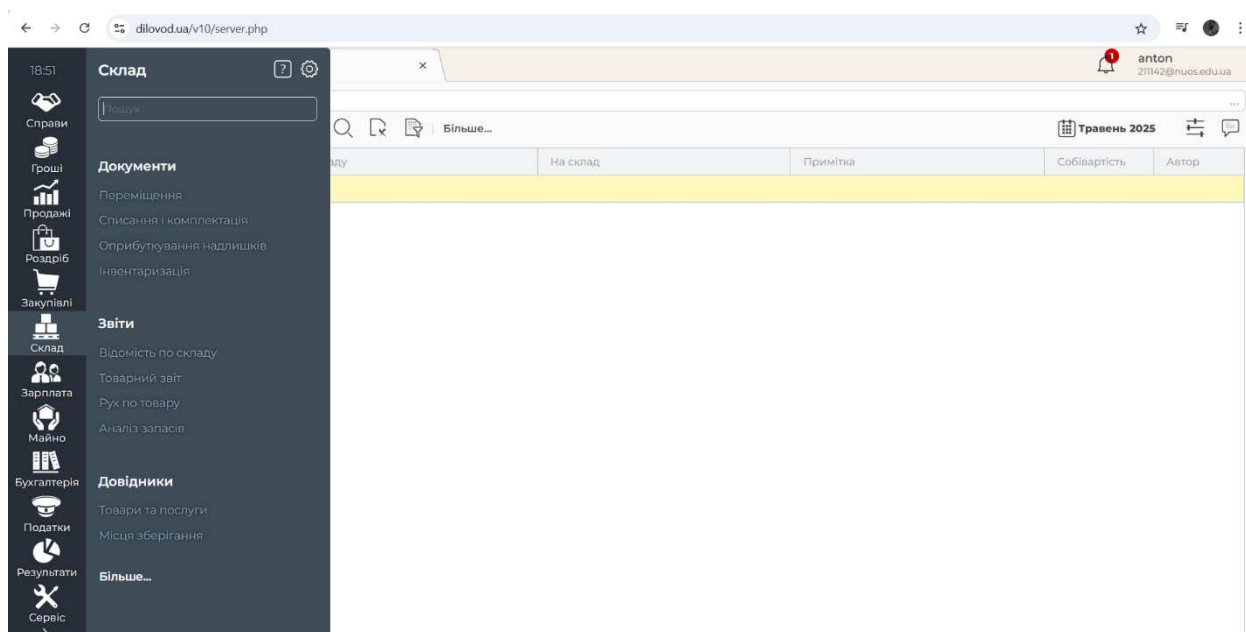


Рисунок 1. 1 Сторінка сайту Dilovod

Dilovod забезпечує повний цикл бухгалтерського обліку – від первинної документації (накладні, рахунки, акти) до здачі звітності та фіскалізації продажів через інтеграцію з програмними РРО. Користувачі можуть формувати звіти у форматах Excel, PDF або CSV, що спрощує їх подальшу обробку.

Сервіс дозволяє завантажувати номенклатуру товарів і залишки з Excel, вести облік за собівартістю, автоматично розраховувати вартість при переміщенні чи списанні товарів. Звіт "Аналіз запасів" допомагає оцінити поточні залишки, собівартість і потребу в товарах на різних складах.

Інтерфейс інтуїтивний, з боковим меню для швидкого доступу до документів і звітів, а також "робочим столом" для налаштування ярликів. Дані зберігаються на серверах у Німеччині з шифруванням рівня інтернет-банкінгу, що забезпечує високий рівень безпеки.

Переваги: Зручність для користувачів без глибоких бухгалтерських знань, можливість автоматизації первинної документації та звітності, інтеграція з РРО для фіскалізації.

Недоліки: Обмежений функціонал для складних виробничих процесів чи великих компаній з понад 20 користувачами. Деяка інформація про

можливості доступна лише після реєстрації, що може ускладнити ознайомлення.

USAP.ONLINE – це хмарна платформа для автоматизації бізнес-процесів, орієнтована на малий і середній бізнес, зокрема у сфері торгівлі та послуг. Сервіс пропонує інструменти для управління фінансами, обліку та звітності, з акцентом на гнучкість і адаптацію до потреб компанії. [2]

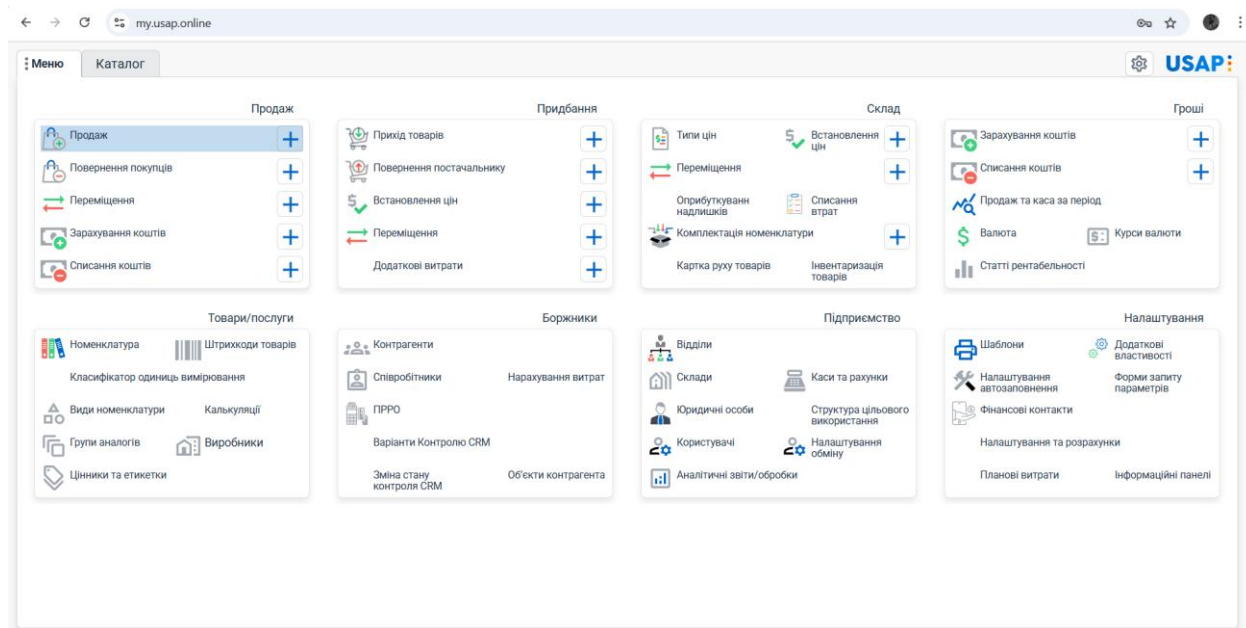


Рисунок 1. 2 Сторінка сайту USAP.ONLINE

Система складається з модулів, які можна налаштувати під конкретні бізнес-процеси, наприклад, облік продажів, закупівель чи фінансів. Це дозволяє адаптувати платформу до різних галузей.

USAP.ONLINE підтримує облік товарів, управління запасами та фінансовими операціями. Є можливість інтеграції з ПРРО для фіскалізації продажів, що відповідає українському законодавству.

Платформа дозволяє формувати різноманітні звіти для аналізу продажів, фінансових показників і запасів, що допомагає в прийнятті управлінських рішень.

Дані зберігаються в хмарі, що забезпечує доступ з будь-якого місця, але інформація про безпеку (наприклад, шифрування) недоступна без реєстрації.

Переваги: Гнучкість налаштувань, хмарний доступ, інтеграція з ПРРО, широкий вибір звітів для аналізу бізнесу.

Недоліки: Інформація про функціонал обмежена без попередньої реєстрації, що ускладнює оцінку можливостей. Інтерфейс може бути складним для новачків, а повний набір функцій може виявитися дорогим для невеликих компаній.

RemOnline – це хмарний сервіс, розроблений для автоматизації бізнес-процесів у сфері торгівлі, послуг і невеликих виробництв. Він фокусується на управлінні замовленнями, складським обліком, фінансами та клієнтськими даними, що робить його придатним для роздрібних мереж, таких як "Південний стиль".

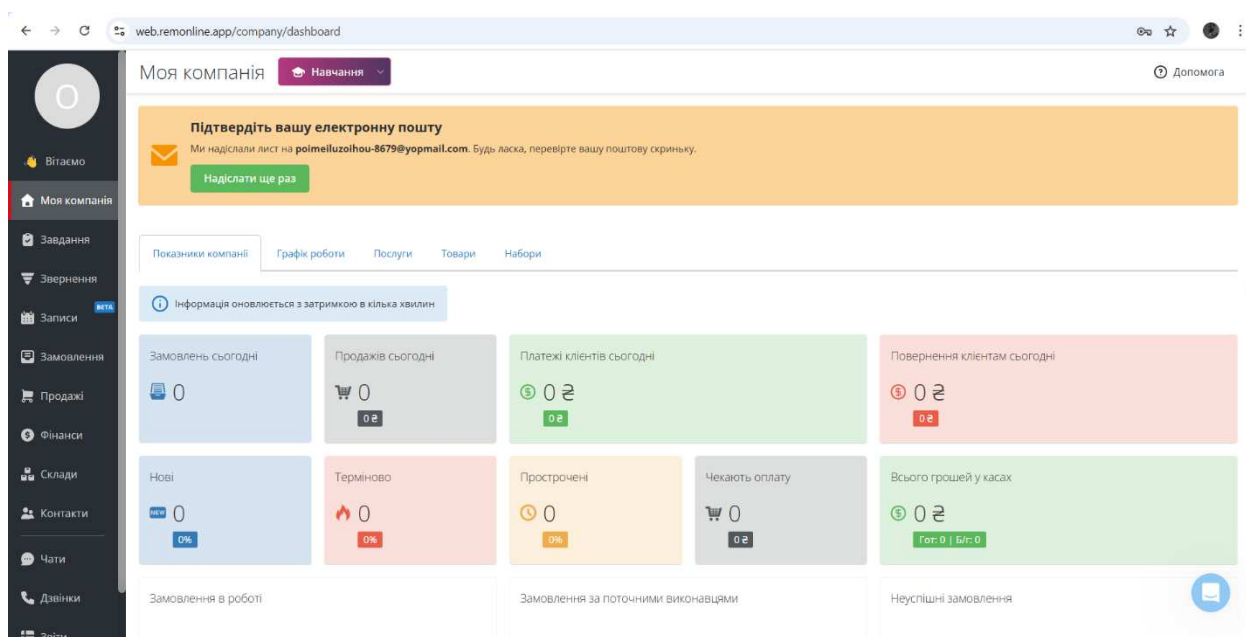


Рисунок 1. 3 Сторінка сайту RemOnline

RemOnline автоматизує цикл замовлення – від створення заявки до завершення угоди, дозволяючи контролювати виконання робіт і касові операції.

Сервіс забезпечує облік товарів, переміщення між складами, інвентаризацію та управління запасами. Є можливість інтеграції з платіжними системами (наприклад, LiqPay) та логістичними сервісами (Nova Poshta).

Вбудована CRM-система дозволяє вести базу клієнтів, відстежувати історію взаємодії та надсилати SMS-повідомлення для інформування про замовлення чи акції.

Платформа пропонує понад 20 звітів для аналізу продажів, фінансових показників і складських запасів, що полегшує прийняття управлінських рішень.

RemOnline має мобільні додатки, що дозволяють керувати бізнесом у дорозі, а також підтримує інтеграцію з Instagram для маркетингових цілей. [3]

Переваги: Інтуїтивний інтерфейс, розвинена аналітика, інтеграція з популярними сервісами, наявність CRM і мобільних додатків.

Недоліки: Широкий функціонал може вимагати часу на освоєння, а повна підписка може бути дорогою для невеликих компаній. Деякі функції, як-от інтеграція з платіжними системами, можуть потребувати додаткових налаштувань.

По результатам аналізу зроблено порівняльну таблицю

Таблиця 1.1

Порівняльний аналіз ІС

Функціональність	Dilovod	USAP.ONLINE	RemOnline
Бухгалтерський облік	✓	✓	✓
Управління запасами	✓	✓	✓
Формування звітів	✓	✓	✓
Інтеграція з ПРРО (фіскалізація)	✓	✓	✗
CRM (клієнтська база)	✗	✗	✓
Управління замовленнями	✗	✓	✓
Інтеграція з платіжними системами	✗	✗	✓
Інтеграція з логістичними сервісами	✗	✗	✓
Мобільні додатки	✗	✗	✓
Аналітика продажів	✓	✓	✓
Інтеграція з маркетплейсами	✗	✗	✓
Автоматизація первинної документації	✓	✓	✗

На основі проведеного аналізу можна зробити висновок, що жодна з розглянутих систем не відповідає необхідним вимогам

1.3 Формування вимог до системи

Для забезпечення ефективного функціонування інформаційної системи мережі магазинів одягу "Південний стиль" необхідно сформувати комплекс вимог, які враховують потреби користувачів, технічні можливості та специфіку бізнес-процесів. На основі аналізу предметної області та проблем, виявлених у підрозділі 1.1 (ручний документообіг, відсутність єдиного інформаційного простору, ручне складання звітів), вимоги поділено на кілька категорій: вимоги користувачів, системні, функціональні, вимоги до інформації, вимоги до програмних і технічних засобів, а також атрибути якості.

Вимоги користувачів

Користувачі системи включають різні категорії працівників мережі: продавців-консультантів, адміністраторів магазинів, операторів складу, працівників відділу закупівель, бухгалтерів, маркетологів та ІТ-персонал. Основні вимоги користувачів:

Продавці-консультанти: Зручний доступ до інформації про наявність товарів, можливість оформлення продажів і замовлень, перегляд акцій та знижок.

Адміністратори магазинів: Формування звітів про продажі, контроль роботи персоналу, управління локальними складськими запасами.

Оператори складу: Облік руху товарів, автоматизоване оновлення даних про запаси, формування заявок на поповнення.

Відділ закупівель: Управління постачальниками, створення та відстеження замовлень, аналіз потреб у товарах.

Бухгалтери: Автоматизоване формування фінансових звітів, облік продажів і витрат, інтеграція з ПРРО для фіскалізації.

Маркетологи: Доступ до аналітики продажів, управління акціями та клієнтськими даними для персоналізованих пропозицій.

ІТ-персонал: Можливість адміністрування системи, управління доступом користувачів, моніторинг безпеки.

Функціональні вимоги

Функціонал системи має охоплювати ключові бізнес-процеси мережі магазинів:

- Реєстрація та авторизація користувачів із різними ролями (адміністратор, продавець, бухгалтер тощо).
- Управління каталогом товарів: додавання, редагування, видалення позицій, облік характеристик (розмір, колір, бренд).
- Обробка замовлень: створення, редагування, відстеження статусу, інтеграція з касовими операціями.
- Управління складськими запасами: облік руху товарів, автоматичне оновлення даних, формування звітів про наявність.
- Управління клієнтськими даними: створення профілів клієнтів, збереження історії покупок, підтримка програм лояльності.
- Формування звітів: звіти про продажі, закупівлі, складські запаси, фінансові операції.
- Інтеграція з зовнішніми сервісами: платіжні шлюзи, служби доставки, ПРРО.

Вимоги до інформації

Інформація в системі повинна бути структурованою, актуальною, зрозумілою та захищеною:

- Формат обміну даними — JSON через REST API, що забезпечує сумісність із сучасними веб-технологіями. [4]
- Конфіденційна інформація (наприклад, персональні дані клієнтів) шифрується, а автентифікація здійснюється через JWT. [5]
- Захист від SQL-ін'єкцій забезпечується через використання Hibernate. [6]

- Автоматичне резервне копіювання бази даних здійснюється не рідше одного разу на добу для відновлення даних у разі збоїв.

Вимоги до складу та параметрів технічних засобів

Система розгортається на сервері в головному офісі, який забезпечує централізовану обробку даних і доступ через Інтернет для всіх магазинів і складів. Сервер має відповідати наступним характеристикам:

- Процесор: Intel Xeon E5 або вище (наприклад, Xeon E5-2620 v4 з 8 ядрами та частотою 2.1 ГГц)
- Оперативна пам'ять: 32 ГБ DDR4 ECC
- Диск: SSD об'ємом 500 ГБ або більше
- Мережевий адаптер: Gigabit Ethernet (1 Гбіт/с) для швидкого і стабільного підключення до Інтернету, що забезпечить низьку затримку при обміні даними між сервером і клієнтськими комп'ютерами.
- Джерело безперебійного живлення (ДБЖ): Потужністю від 1000 ВА для захисту від перебоїв у електропостачанні, що гарантує безперервну роботу сервера.
- Операційна система: Ubuntu Server 22.04 LTS або новіша [7]
- Монітор, клавіатура, миша: Для адміністрування сервера на місці в головному офісі (можуть бути тимчасово підключені за потреби).
- Захист і безпека: Фізичний доступ до сервера обмежений (серверна кімната з контролем доступу), встановлено антивірусне ПЗ (наприклад, ClamAV для Ubuntu), налаштовано фаєрвол (ufw або Windows Defender Firewall).

Доступ до сервера відбуватиметься через інтернет, кожен магазин та склад матиме комп'ютер з такими характеристиками

- Процесор: Intel i5 або вище.
- Оперативна пам'ять: 16 ГБ і вище.
- Диск: SSD із 50 ГБ вільного місця.
- Монітор, клавіатура, миша з ергономічним дизайном.

- Підключення до мережі з доступом до Інтернету через Wi-Fi чи Ethernet.
- Принтер для роздрукування звітів.

Програмні вимоги

Система базується на сучасному технологічному стеку та працює через веб-браузер:

- Мова програмування: Java 21.
- Фреймворк: Spring Boot із вбудованим сервером Tomcat. [8]
- СУБД: MySQL для зберігання даних. [9]
- Безпека: Spring Security із підтримкою JWT-аутентифікації. [10]
- Тестування: JUnit [11] і Postman [12].
- Середовище розробки: IntelliJ IDEA з плагінами (Maven, JPA Buddy, Thymeleaf Plugin). [13]
- Інструменти бази даних: HeidiSQL. [14]
- Локальний сервер: XAMPP для тестування. [15]
- Фронтенд-інтеграція: Thymeleaf, Bootstrap, jQuery.
- Програмне забезпечення адаптовано до операційних систем Windows 10/11 та Linux (Ubuntu Server).

1.4 Розробка концепції інформаційної системи

Модель концепції інформаційної системи наведено на рисунку 1.5

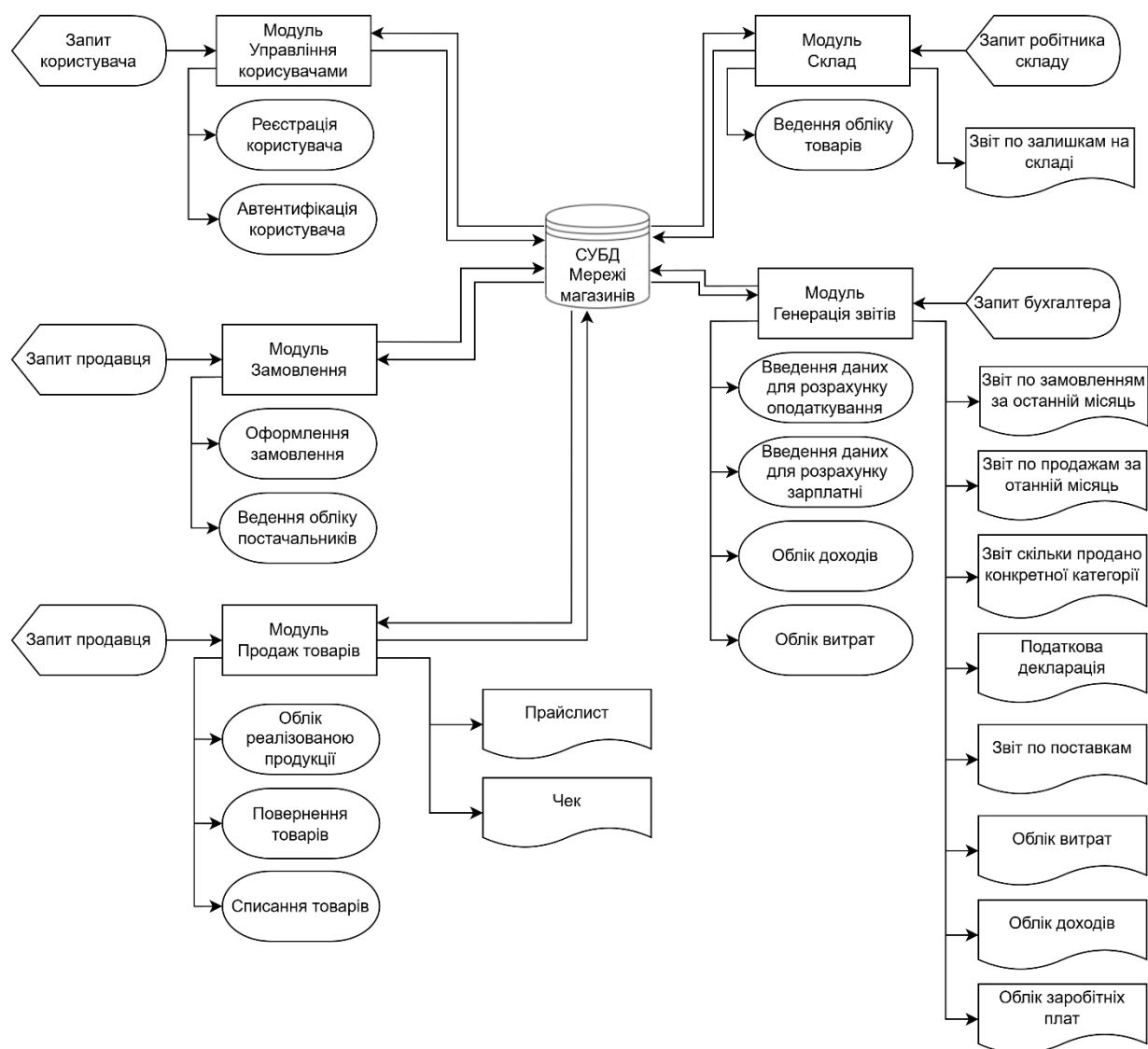


Рисунок 1. 5 Модель концепції інформаційної системи

Функціонал системи розподілений між кількома бекенд-модулями, кожен із яких відповідає за окрему сферу бізнес-процесів.

Модуль управління користувачами: Реєстрація, автентифікація, управління ролями та правами доступу (адміністратор, продавець, бухгалтер, робітник складу).

Модуль управління складом: Ведення обліку товарів (додавання, редагування, видалення товарів, редагування характеристик, пошук і сортування, автоматичне оновлення даних про запаси)

Модуль продажу товарів: Облік реалізованою продукції, списання та повернення товарів.

Модуль управління замовленнями: Створення, редагування, відстеження статусу замовлень, ведення обліку постачальників.

Модуль генерації звітів: Звіт по замовленням за останній місяць, звіт по продажам за останній місяць, звіт скільки продано конкретної категорії, податкова декларація, звіт по поставкам, облік витрат, облік доходів, облік заробітніх плат.

Концепція системи спрямована на вирішення виявлених проблем. Ручний документообіг замінюється автоматизованим управлінням даними через централізовану базу MySQL, що усуває помилки та прискорює доступ до інформації. Відсутність єдиного інформаційного простору вирішується завдяки REST API, яке забезпечує обмін даними між магазинами, складами та офісом у реальному часі. Ручне складання звітів автоматизується за допомогою модуля генерації звітів, що економить час бухгалтерів і адміністраторів. Низька безпека даних усувається впровадженням шифрування, JWT-автентифікації та захисту від SQL-ін'єкцій, а резервне копіювання гарантує захист від втрати інформації. Система враховує потреби різних категорій користувачів, таких як адміністратори, продавці, оператори складу, бухгалтери та маркетологи, а також забезпечує масштабованість для майбутнього розвитку.

Система працює за принципом клієнт-серверної взаємодії через REST API, що забезпечує гнучкість і масштабованість. Користувачі надсилають HTTP-запити (GET, POST, PUT, DELETE) до API-шлюзу, який після автентифікації перенаправляє їх до відповідного бекенд-сервісу. Наприклад, продавець-консультант може надіслати запит на отримання інформації про наявність товару, який обробляється сервісом управління товарами, а бухгалтер може запросити фінансовий звіт через сервіс генерації звітів. Дані зберігаються в MySQL, а їх обробка автоматизована завдяки ORM-фреймворку Hibernate, що зменшує ризик помилок і спрощує розробку. Безпека забезпечується на всіх рівнях: від автентифікації (JWT) до шифрування конфіденційних даних, таких як персональна інформація

клієнтів. Для захисту від збоїв передбачено автоматичне резервне копіювання бази даних із періодичністю один раз на добу.

Для тестування використовуються JUnit і Postman, що гарантує відповідність системи функціональним вимогам. Система розгортається на вбудованому сервері Spring Boot, що забезпечує гнучкість розгортання.

1.5 Постановка задачі

Метою даної дипломної роботи є розробка підсистеми бекенду інформаційної системи для мережі магазинів одягу "Південний стиль". Основне завдання цієї підсистеми полягає в автоматизації критично важливих бізнес-процесів, зокрема управління товарними залишками, обробки продажів, організації закупівель та аналітичної обробки даних. Окрім того, система має вирішувати низку існуючих проблем, пов'язаних із відсутністю централізованого інформаційного середовища, надмірною залежністю від ручного документообігу та складністю формування звітності.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Провести аналіз предметної області та огляд існуючих аналогічних систем для визначення їхніх переваг, недоліків та можливостей для вдосконалення, що дозволить сформулювати вимоги до розроблюваної системи.
2. Визначити архітектурні принципи майбутньої системи, орієнтуючись на забезпечення її масштабованості, інформаційної безпеки та сумісності з іншими цифровими сервісами, такими як платіжні платформи та логістичні рішення.
3. Вибрати відповідні інструменти та технології розробки, зокрема мову програмування Java, фреймворк Spring Boot, систему управління базами даних MySQL та REST API, для створення надійного та ефективного бекенду.

4. Розробити структуру бази даних та загальну архітектуру системи, які забезпечать централізоване оброблення даних та обмін інформацією в реальному часі між магазинами, складами та головним офісом.
5. Реалізувати RESTful API, яке покриватиме основні функції системи, зокрема управління обліковими записами користувачів (реєстрація, аутентифікація), адміністрування товарного каталогу, обробку замовлень, моніторинг запасів та генерацію звітної документації.
6. Інтегрувати механізми безпеки, такі як JWT-аутентифікація та захист від SQL-ін'єкцій, для забезпечення конфіденційності даних та контролю доступу користувачів.
7. Провести комплексне тестування розробленої підсистеми для перевірки її функціональності, продуктивності та відповідності встановленим вимогам.

Реалізація вищезазначених завдань дозволить створити сучасну бекенд-підсистему, яка виступатиме ядром корпоративної інформаційної системи мережі «Південний стиль». Вона сприятиме підвищенню ефективності операційної діяльності підприємства завдяки автоматизації бізнес-процесів, підвищенню безпеки обробки даних та створенню основи для майбутньої модернізації, включаючи можливість інтеграції з платіжними сервісами, розробки мобільних застосунків і впровадження модулів аналітики.

Висновок

Виконано аналіз предметної області мережі магазинів одягу "Південний стиль", визначено ключові проблеми, такі як ручний документообіг та відсутність єдиного інформаційного простору, та сформульовано вимоги до бекенд-підсистеми. Розроблено концепцію системи на основі сучасних технологій (Java, Spring Boot, MySQL, REST API), яка забезпечує автоматизацію бізнес-процесів, безпеку даних і масштабованість. Постановка задачі чітко окреслила мету та завдання, створивши основу для подальшого проектування та реалізації системи, що сприятиме підвищенню ефективності роботи мережі.

РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ ПІДСИСТЕМИ БЕКЕНДУ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕРЕЖІ МАГАЗИНІВ ОДЯГУ «ПІВДЕННИЙ СТИЛЬ»

2.1 Загальносистемні рішення

Підсистема бекенду, розроблена для мережі магазинів одягу «Південний стиль», являє собою інтегровану інформаційну систему, основне призначення якої полягає в автоматизації та оптимізації основних бізнес-процесів підприємства. Її функціональність орієнтована на забезпечення єдиного інформаційного середовища для п'яти роздрібних точок, двох складів та центрального офісу компанії, що спеціалізується на реалізації чоловічого, жіночого та дитячого одягу. Ключова мета впровадження цієї підсистеми — підвищення операційної ефективності та загальної прибутковості підприємства через автоматизоване виконання таких процесів, як управління товарними запасами, моніторинг продажів, обробка замовлень та формування аналітичної звітності.

У своїй основі система забезпечує управління користувачами з надійним контролем доступу на основі ролей, гарантуючи, що адміністратори, продавці, бухгалтери та складські працівники можуть ефективно виконувати свої обов'язки в межах своїх призначених дозволів. Вона підтримує повний цикл управління товарними залишками: від моніторингу рівня запасів у реальному часі на всіх локаціях до адміністрування товарного каталогу й обліку внутрішніх переміщень продукції між складами та магазинами. Модуль обробки замовлень забезпечує створення, оновлення та відстеження замовлень, інтегруючись із касовими операціями та, за потреби, з зовнішніми службами, що сприяє формуванню безперервного та зручного користувацького досвіду. Крім того, система включає розширені можливості звітності, генеруючи детальні звіти про продажі, запаси, фінансові транзакції

та податкові зобов'язання, які є важливими для прийняття обґрунтованих рішень.

Діаграма варіантів використання для підсистеми бекенду інформаційної системи "Південний стиль" представлена на рисунку 2.1.

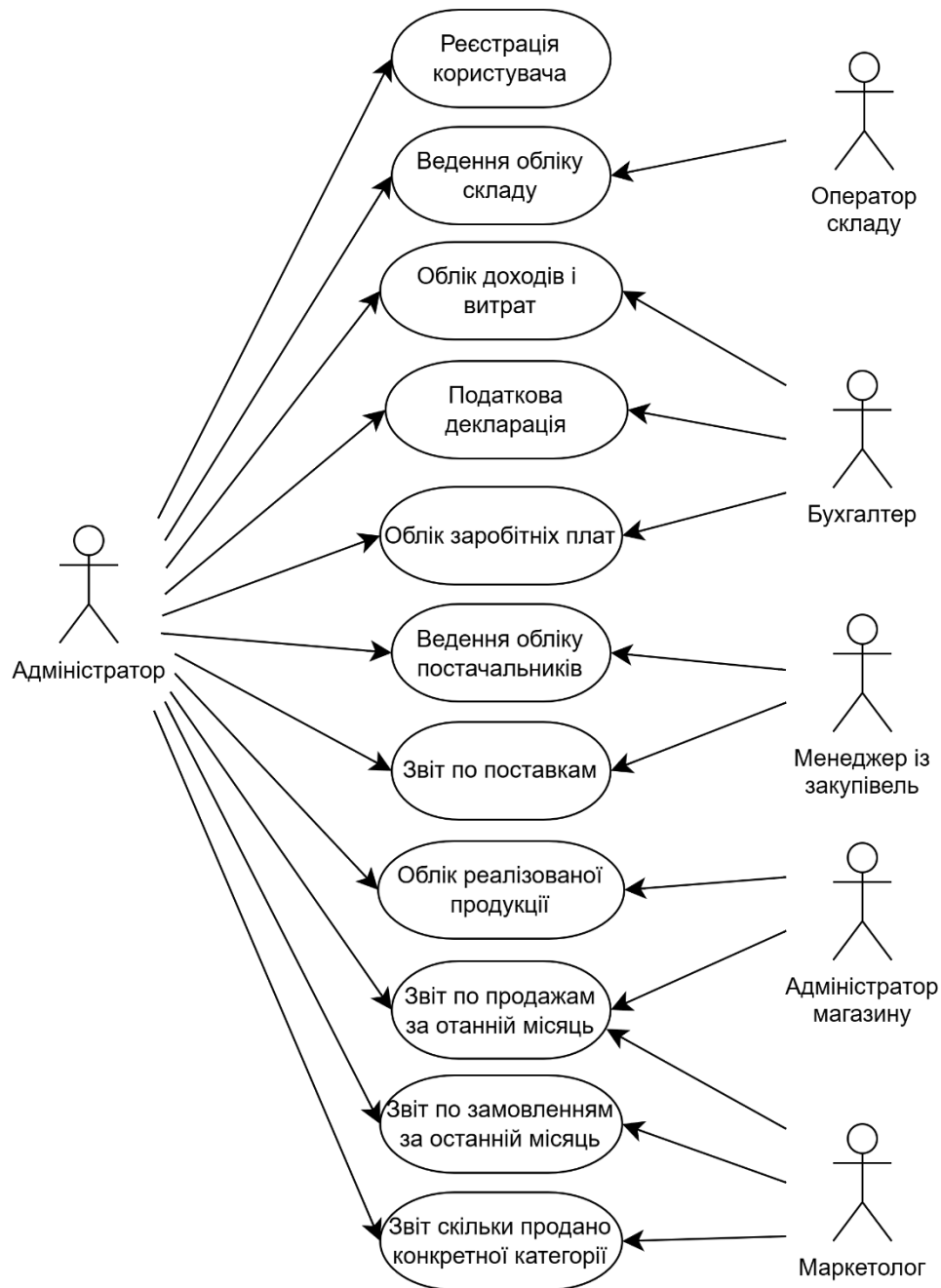


Рисунок 2. 1 Діаграма варіантів використання

Створена діаграма варіантів використання моделює різні можливі дії, які можуть виконувати користувачі системи мережі магазинів одягу. У цій

діаграмі представлені основні типи акторів: адміністратор, продавець-консультант, адміністратор магазину, оператор складу, бухгалтер, маркетолог, кожен з яких має свій власний набір варіантів використання.

Адміністратор системи може користуватись всіма функціями системи, зокрема управляти обліковими записами користувачів. Адміністратор магазину може генерувати звіти по магазину, робити записи про повернення та списання товарів. Оператор складу може керувати запасами на складі, робити інвентаризацію. Менеджер із закупівель веде облік поставок та постачальників. Бухгалтер може генерувати фінансові звіти до податкової, вести доходи та витрати, робити розрахунок заробітної плати. Маркетолог аналізує дані продажів.

Кожен з цих акторів взаємодіє з системою через відповідні сервіси, такі як управління користувачами, обробка продажів, управління запасами, генерація звітів. Діаграма допомагає уявити різноманітність можливостей взаємодії користувачів з системою та є важливим інструментом для забезпечення повноти та коректності проектування системи.

2.2. Рішення з інформаційного забезпечення

Розробка концептуальної моделі. Концепція являє собою вхідну і вихідну інформацію і визначає склад концептуальної моделі.

Вхідна інформація системи:

- Дані про категорії: назва
- Дані про продукти: назва, ціна, розмір, колір, артикул, бренд
- Дані про клієнтів: ім'я, прізвище, електронна пошта, телефон, адреса, бали лояльності
- Дані про замовлення: дата замовлення, статус, загальна сума
- Дані про деталі замовлення: кількість, ціна (для кожного продукту в замовленні)

- Дані про доставки: дата доставки, служба, номер відстеження, статус
- Дані про платежі: дата платежу, сума, метод, статус
- Дані про співробітників: ім'я, прізвище, роль
- Дані про магазини: назва, місцезнаходження, телефон
- Дані про запаси в магазинах: кількість продуктів
- Дані про склади: назва, місцезнаходження
- Дані про запаси на складах: кількість продуктів
- Дані про постачальників: назва, контактна особа, телефон, електронна пошта
- Дані про поставки: дата поставки, кількість (для кожного продукту)
- Дані про продажі: дата продажу, загальна сума
- Дані про деталі продажу: кількість, ціна (для кожного продукту в продажу)

Вихідна інформація:

- Списки продуктів за категоріями
- Списки клієнтів
- Списки замовлень
- Детальні звіти про замовлення
- Звіти про доставки
- Звіти про платежі
- Списки співробітників по магазинах
- Звіти про запаси в магазинах
- Звіти про запаси на складах
- Звіти про поставки
- Звіти про продажі
- Фінансовий облік доходів від продажів і платежів

- Звіти про продажі (щомісячні продажі, продажі за категоріями)
- Звіти про закупівлі
- Фінансові операції

На основі вхідних та вихідних документів та потоків даних побудуємо концептуальну модель (рисунок 2.2). [15]

Концептуальна модель - модель предметної області, що складається з переліку взаємопов'язаних понять, що використовуються для опису цієї області, разом з властивостями і характеристиками, класифікацією цих понять, за типами, ситуацій, ознаками в даній області і законів перебігу процесів у ній.

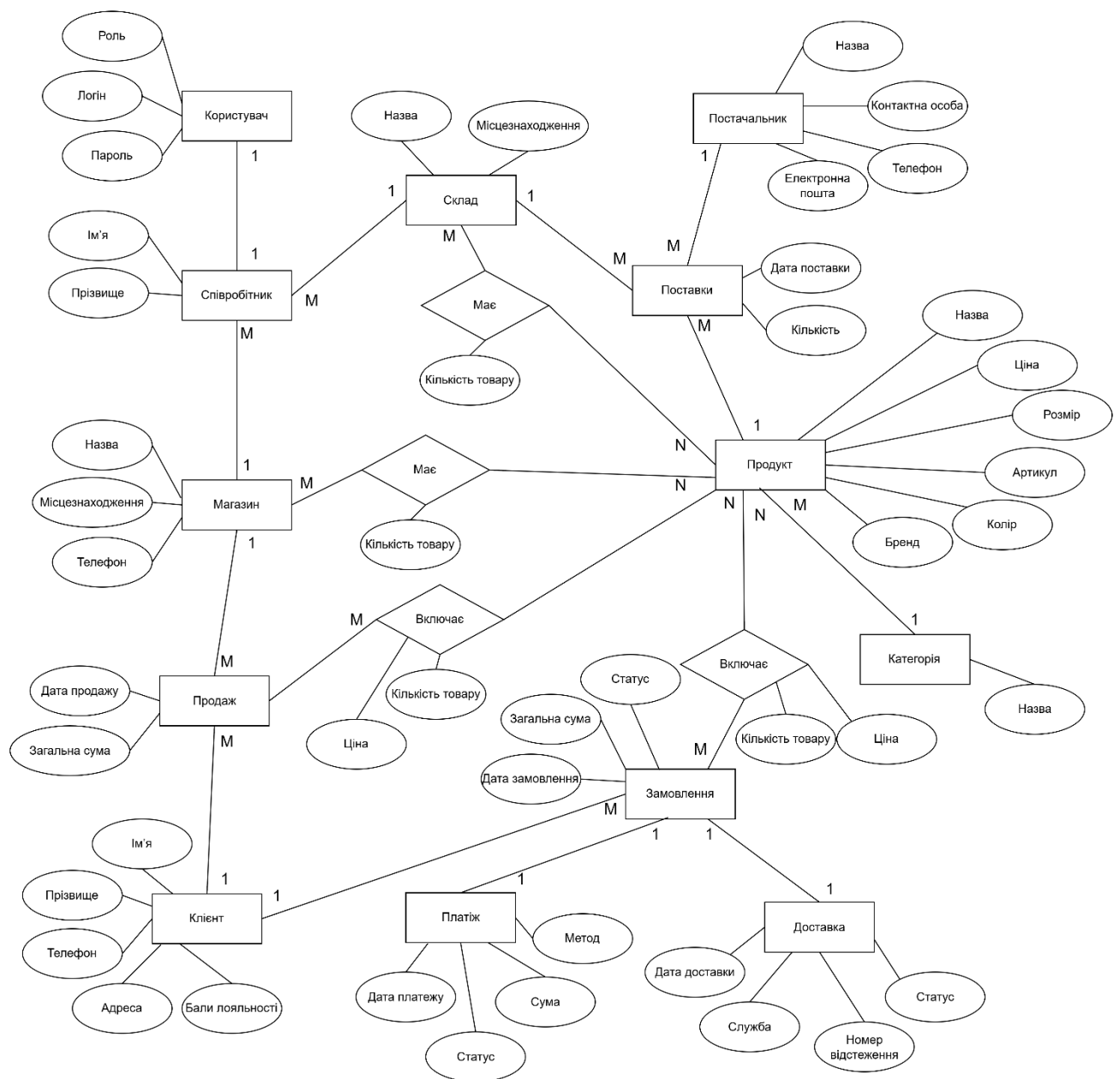


Рисунок 2. 2 Концептуальна модель даних

Розробка логічної моделі даних. Проектування логічної моделі бази даних в основному визначається специфікацією функціональних завдань, а саме визначення числа і структури таблиць, формування запитів до БД автоматизованої системи мережі магазинів одягу "Південний стиль", розробки алгоритмів обробки інформації, створення форм для введення і редагування даних в базі необхідно забезпечити цілісність зв'язків.

Концептуальна модель є базою для побудови логічної моделі даних. Логічна модель є більш детальним представленням даних. В ній сутності трансформуються в таблиці даних, одна сутність може представлятися декількома таблицями, атрибути сутностей стають полями таблиць, виділяються відношення між таблицями та первинні ключі.

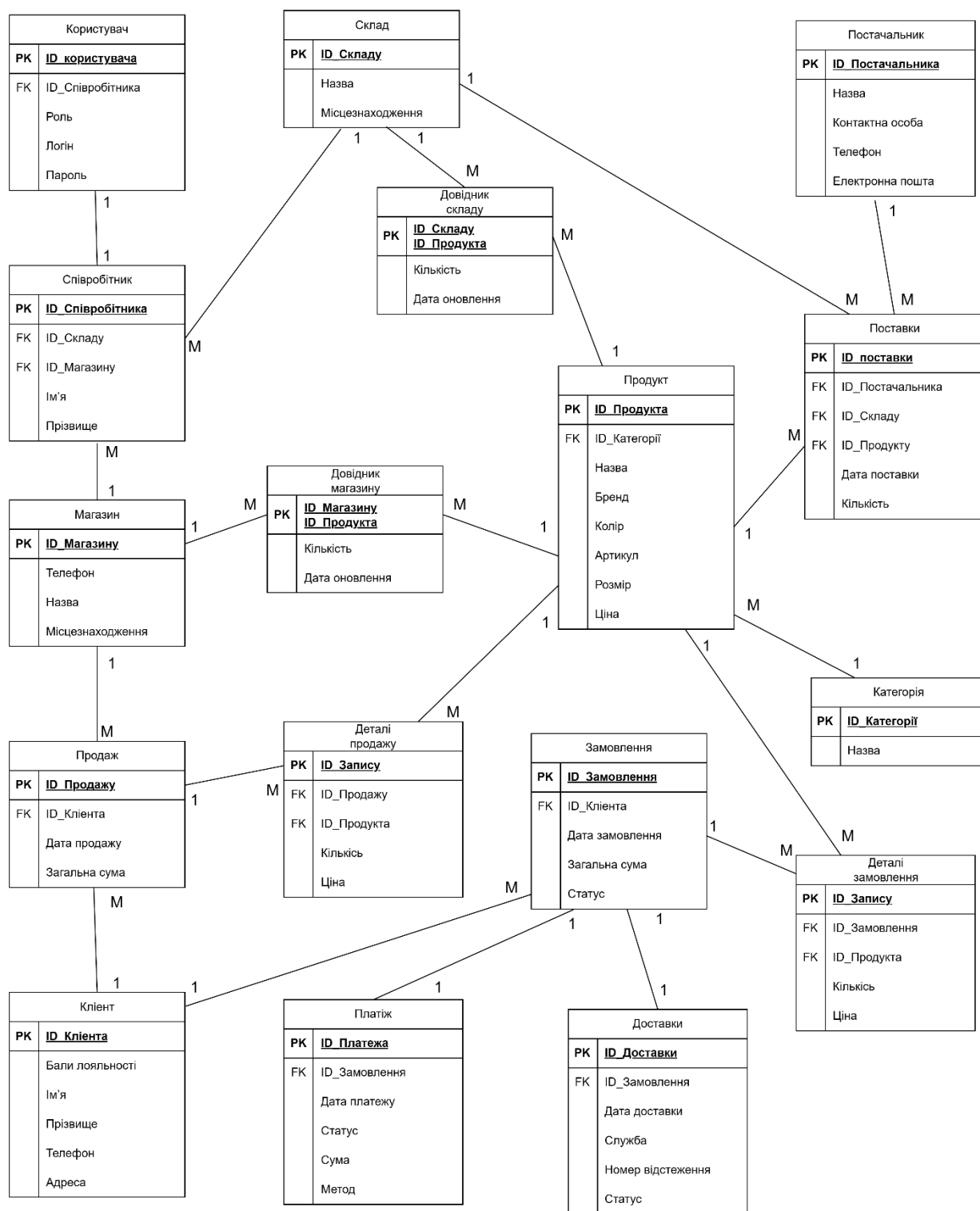


Рисунок 2. 3 Логічна модель даних

Розробка фізичної моделі даних - створення схеми бази даних для СУБД автоматизованої системи. Фізична модель будується з урахуванням логічної з урахуванням обмежень, накладених можливостями обраної СУБД. Фізична модель бази даних містить усі деталі, необхідні СУБД для створення бази:

найменування таблиць та стовпців, типи даних, визначення первинних та зовнішніх ключів тощо.

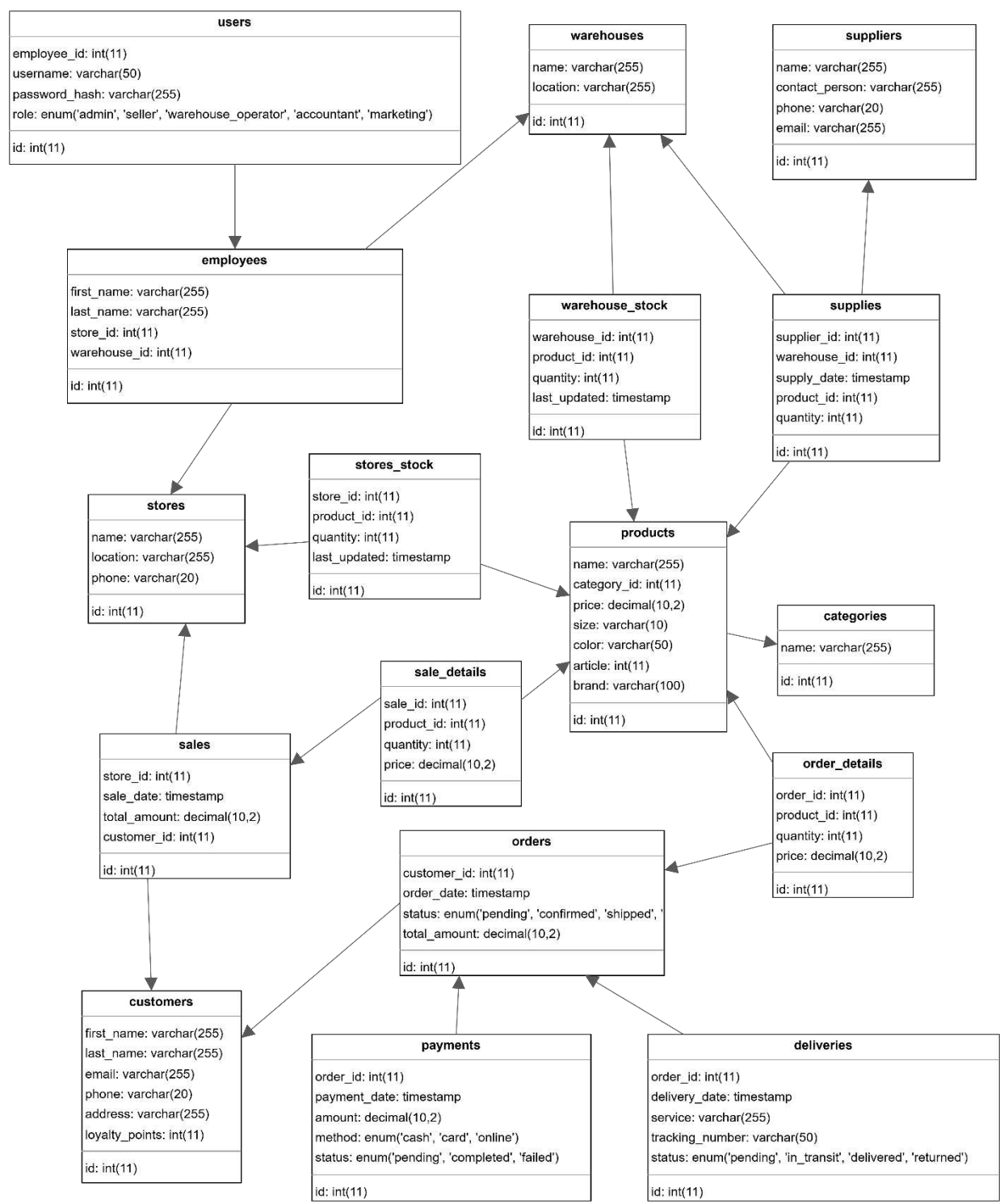


Рисунок 2. 4 Фізична модель даних

Нижче на таблицях 2.1-2.17 наведено повний опис таблиць новоствореної бази даних

Таблиця 2.1

Структура таблиці «categories»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
name	VARCHAR(255)	NOT NULL	No	No
created_at	TIMESTAMP	NOT NULL	No	No

Таблиця 2.2

Структура таблиці «customers»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
first_name	VARCHAR(255)	NOT NULL	No	No
last_name	VARCHAR(255)	NOT NULL	No	No
email	VARCHAR(255)	NULL	No	No
phone	VARCHAR(20)	NULL	No	No
address	VARCHAR(255)	NULL	No	No
loyalty_points	INT(11)	NOT NULL	No	No
created_at	TIMESTAMP	NOT NULL	No	No

Таблиця 2.3

Структура таблиці «deliveries»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
order_id	INT(11)	NOT NULL	No	Yes
delivery_date	TIMESTAMP	NOT NULL	No	No
service	VARCHAR(255)	NOT NULL	No	No
tracking_number	VARCHAR(50)	NULL	No	No
status	ENUM	NOT NULL	No	No

Таблиця 2.4

Структура таблиці «employees»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
first_name	VARCHAR(255)	NOT NULL	No	No
last_name	VARCHAR(255)	NOT NULL	No	No
role	ENUM	NOT NULL	No	No
store_id	INT(11)	NULL	No	Yes
warehouse_id	INT(11)	NULL	No	Yes

Таблиця 2.5

Структура таблиці «orders»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
customer_id	INT(11)	NOT NULL	No	Yes
order_date	TIMESTAMP	NOT NULL	No	No
status	ENUM	NOT NULL	No	No
total_amount	DECIMAL(10,2)	NOT NULL	No	No

Таблиця 2.6

Структура таблиці «order_details»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
order_id	INT(11)	NOT NULL	No	Yes
product_id	INT(11)	NOT NULL	No	Yes
quantity	INT(11)	NOT NULL	No	No
price	DECIMAL(10,2)	NOT NULL	No	No

Таблиця 2.7

Структура таблиці «payments»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
order_id	INT(11)	NOT NULL	No	Yes
payment_date	TIMESTAMP	NOT NULL	No	No
amount	DECIMAL(10,2)	NOT NULL	No	No
method	ENUM	NOT NULL	No	No
status	ENUM	NOT NULL	No	No

Таблиця 2.8

Структура таблиці «products»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
name	VARCHAR(255)	NOT NULL	No	No
category_id	INT(11)	NULL	No	Yes
price	DECIMAL(10,2)	NOT NULL	No	No
size	VARCHAR(10)	NULL	No	No
color	VARCHAR(50)	NULL	No	No
article	INT(11)	NULL	No	No
brand	VARCHAR(100)	NULL	No	No
created_at	TIMESTAMP	NOT NULL	No	No

Таблиця 2.9

Структура таблиці «sales»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
store_id	INT(11)	NOT NULL	No	Yes
sale_date	TIMESTAMP	NOT NULL	No	No
total_amount	DECIMAL(10,2)	NOT NULL	No	No
customer_id	INT(11)	NULL	No	Yes

Таблиця 2.10

Структура таблиці «sale_details»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
sale_id	INT(11)	NOT NULL	No	Yes
product_id	INT(11)	NOT NULL	No	Yes
quantity	INT(11)	NOT NULL	No	No
price	DECIMAL(10,2)	NOT NULL	No	No

Таблиця 2.11

Структура таблиці «stores»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
name	VARCHAR(255)	NOT NULL	No	No
location	VARCHAR(255)	NOT NULL	No	No
phone	VARCHAR(20)	NULL	No	No
created_at	TIMESTAMP	NOT NULL	No	No

Таблиця 2.12

Структура таблиці «stores_stock»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
store_id	INT(11)	NOT NULL	No	Yes
product_id	INT(11)	NOT NULL	No	Yes
quantity	INT(11)	NOT NULL	No	No
last_updated	TIMESTAMP	NOT NULL	No	No

Таблиця 2.13

Структура таблиці «suppliers»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
name	VARCHAR(255)	NOT NULL	No	No
contact_person	VARCHAR(255)	NULL	No	No
phone	VARCHAR(20)	NULL	No	No
email	VARCHAR(255)	NULL	No	No
created_at	TIMESTAMP	NOT NULL	No	No

Таблиця 2.14

Структура таблиці «supplies»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
supplier_id	INT(11)	NOT NULL	No	Yes
warehouse_id	INT(11)	NOT NULL	No	Yes
supply_date	TIMESTAMP	NOT NULL	No	No
product_id	INT(11)	NOT NULL	No	Yes
quantity	INT(11)	NOT NULL	No	No

Таблиця 2.15

Структура таблиці «users»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
employee_id	INT(11)	NOT NULL	No	Yes
username	VARCHAR(50)	NOT NULL	No	No
password_hash	VARCHAR(255)	NOT NULL	No	No
role	ENUM	NOT NULL	No	No
created_at	TIMESTAMP	NOT NULL	No	No

Таблиця 2.16

Структура таблиці «warehouses»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
name	VARCHAR(255)	NOT NULL	No	No
location	VARCHAR(255)	NOT NULL	No	No
created_at	TIMESTAMP	NOT NULL	No	No

Таблиця 2.17

Структура таблиці «warehouse_stock»

Name	Datatype	Null Option	Is PK	Is FK
id	INT(11)	NOT NULL	Yes	No
warehouse_id	INT(11)	NOT NULL	No	Yes
product_id	INT(11)	NOT NULL	No	Yes
quantity	INT(11)	NOT NULL	No	No
last_updated	TIMESTAMP	NOT NULL	No	No

2.3 Рішення щодо програмного забезпечення

Автоматизована підсистема бекенду інформаційної системи мережі магазинів одягу «Південний стиль» реалізується у вигляді єдиного програмного комплексу, структурованого на низку функціональних модулів, кожен з яких адаптований до вимог конкретної категорії користувачів. До таких категорій належать: продавці-консультанти, адміністратори роздрібних точок, оператори складів, співробітники відділу закупівель, бухгалтерський персонал, фахівці з маркетингу та ІТ-працівники. Кожен модуль забезпечує автоматизацію специфічного набору бізнес-процесів, зокрема: облік товарів, обробка замовлень, формування аналітичної звітності, взаємодія з зовнішніми інформаційними сервісами, а також управління правами доступу користувачів до ресурсів системи.

У процесі розробки програмного забезпечення підсистеми було прийнято низку проектних рішень, обґрунтованих вимогами до масштабованості, продуктивності та безпеки. Зокрема, обрані наступні технології:

- Мова програмування Java 21;

Java є однією з найпоширеніших мов програмування завдяки своїй кросплатформності, надійності та широкій екосистемі. Вона підтримує об'єктно-орієнтоване програмування, що дозволяє створювати масштабовані та модульні додатки. Java 21 обрана через її довгострокову підтримку (LTS) та нові функції, які підвищують продуктивність і безпеку.

- Фреймворк Spring Boot із вбудованим сервером Tomcat;

Spring Boot надає розробникам набір засобів для спрощення створення веб-застосунків, включаючи підтримку конфігурацій за замовчуванням, зручне створення RESTful сервісів та інтеграцію зі сторонніми модулями. Вбудований у фреймворк сервер Tomcat дозволяє розгортати додаток без необхідності додаткового встановлення або налаштування зовнішнього веб-сервера. Крім того, Spring Boot підтримує роботу з ORM-фреймворком

Hibernate для ефективної взаємодії з реляційними базами даних, а також забезпечує засоби реалізації авторизації та автентифікації через Spring Security.

- Система управління базами даних MySQL;

MySQL є популярною реляційною системою керування базами даних, яка функціонує як сервер, обробляючи запити від клієнтів — як локальних, так і віддалених. Основні переваги MySQL полягають у високій швидкодії, простоті адміністрування, високому рівні безпеки та добрій масштабованості. Вона є оптимальним вибором для зберігання структурованих даних, включаючи інформацію про номенклатуру товарів, замовлення, клієнтів, фінансові операції та звітність.

- Spring Security із підтримкою JWT-автентифікації;

Spring Security забезпечує багаторівневий захист веб-застосунків, включаючи автентифікацію, авторизацію та контроль доступу до ресурсів. Впровадження JWT (JSON Web Token) дозволяє реалізувати безпечну та масштабовану передачу облікових даних між клієнтом і сервером, а також гнучке управління правами доступу залежно від ролі користувача (адміністратор, продавець, бухгалтер тощо).

- JUnit і Postman;

JUnit використовується для модульного тестування окремих компонентів Java-додатку, що дозволяє своєчасно виявляти помилки на етапі розробки та гарантувати коректність роботи бізнес-логіки. Postman слугує для інтеграційного тестування RESTful API, забезпечуючи перевірку відповідей серверу, обробку HTTP-запитів і тестування сценаріїв взаємодії між компонентами системи.

- Середовище розробки IntelliJ IDEA з плагінами (Maven, JPA Buddy, Thymeleaf Plugin);

IntelliJ IDEA є потужним середовищем розробки, яке підтримує автодоповнення коду, рефакторинг і дебагінг. Плагіни, такі як Maven,

спрощують управління залежностями, JPA Buddy полегшує роботу з Hibernate, а Thymeleaf Plugin підтримує інтеграцію з фронтендом.

- HeidiSQL;

HeidiSQL є зручним графічним інтерфейсом для адміністрування бази даних MySQL. Цей інструмент дозволяє створювати й редагувати таблиці, здійснювати міграцію даних, виконувати SQL-запити та візуально аналізувати структуру бази, що значно полегшує супровід і тестування даних на етапі розробки.

- Локальний сервер XAMPP для тестування.

XAMPP забезпечує локальне середовище для тестування системи, включаючи сервер Apache і MySQL, що дозволяє розробникам перевіряти функціонал перед розгортанням на основному сервері.

Для функціонування системи на сервері в головному офісі встановлено наступне програмне забезпечення:

- Операційна система Ubuntu Server 22.04 LTS.
- Java 21 і Spring Boot для роботи бекенду.
- MySQL для управління базою даних.
- Антивірусне ПЗ (ClamAV) і фаєрвол (ufw) для забезпечення безпеки.

На клієнтських комп'ютерах у магазинах і складах встановлено:

- Операційна система Windows 10/11.
- Веб-браузер Google Chrome для доступу до системи.
- Драйвери для принтерів для друку звітів.

Кожен комп'ютер у магазинах і складах підключений до Інтернету через Wi-Fi або Ethernet, що забезпечує доступ до сервера в головному офісі.

2.4 Рішення з технічного забезпечення

Для забезпечення роботи підсистеми бекенду інформаційної системи мережі магазинів одягу "Південний стиль" необхідний сервер у головному офісі та комп'ютери в кожному з 5 магазинів і 2 складів із доступом до Інтернету. Сервер централізовано обробляє дані, а клієнтські комп'ютери забезпечують доступ до системи через веб-інтерфейс.

Сервер у головному офісі повинен мати наступні характеристики:

- Процесор Intel Xeon E5-2620 v4 з 8 ядрами та частотою 2.1 ГГц.
- Оперативна пам'ять 32 ГБ DDR4 ECC.
- Диск SSD об'ємом 500 ГБ.
- Мережевий адаптер Gigabit Ethernet.
- Джерело безперебійного живлення (ДБЖ) потужністю 1000 ВА для захисту від перебоїв у електропостачанні.
- Монітор, клавіатура, миша.

Кожен із комп'ютерів у магазинах і складах повинен мати наступні характеристики:

- Процесор: Intel i5.
- Оперативна пам'ять 16 ГБ.
- Диск SSD об'ємом 500 ГБ.
- Монітор, клавіатура, миша з ергономічним дизайном.
- Підключення до мережі з доступом до Інтернету через Wi-Fi та Ethernet.

Принтер повинен мати наступні характеристики:

- Формат друку: A4.
- Кількість кольорів: 4.
- Тип з'єднання: USB.

Для забезпечення мережевого доступу до сервера у головному офісі та між магазинами й складами використовується Wi-Fi роутер із такими характеристиками:

- WAN-порт: Ethernet.

- Wi-Fi стандарти: IEEE 802.11 b/g/n/ac/ax.
- Підтримка протоколів: DHCP.
- Частота роботи Wi-Fi: 5 ГГц, 2.4 ГГц.
- Кількість LAN-портів: 4.
- Кількість WAN-портів: 1.

Кабелі для підключення:

- Кабель UTP для з'єднання комп'ютерів із роутером.
- Кабель USB для підключення принтерів.

Висновок

Проведено розробку проекту підсистеми бекенду інформаційної системи мережі магазинів одягу "Південний стиль", що включає загальносистемні рішення, інформаційне, програмне та технічне забезпечення. На основі аналізу предметної області розроблено концептуальну, логічну та фізичну моделі бази даних, що забезпечить структуроване зберігання даних. Вибір технологічного стеку (Java 21, Spring Boot, MySQL, Spring Security) та інструментів (IntelliJ IDEA, JUnit, Postman) гарантує масштабованість, безпеку та швидкість розробки системи. Технічна інфраструктура, що включає серверне обладнання та комп'ютери, відповідає вимогам до стабільної експлуатації системи в умовах реального бізнес-середовища, забезпечуючи безперебійну роботу користувачів у магазинах, на складах та в офісі.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ ПІДСИСТЕМИ БЕКЕНДУ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕРЕЖІ МАГАЗИНІВ ОДЯГУ «ПІВДЕННИЙ СТИЛЬ»

Для ефективного впровадження застосунку розроблено інструкцію для системного адміністратора, де докладно описано процес розгортання та налаштування системи «Південний стиль Бекенд». Інструкцію наведено у Додатку Б.

Бекенд-підсистема розроблена з використанням мови програмування Java 21, фреймворку Spring Boot із вбудованим сервером Tomcat та реляційної бази даних MySQL.

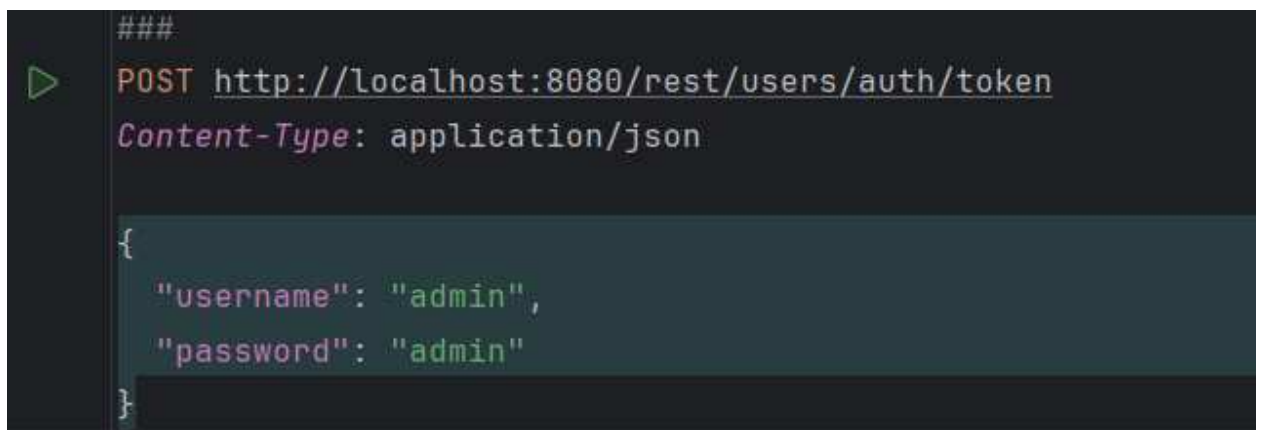
Базу даних реалізовано згідно фізичній моделі з розділу 2

Система поділена на функціональні модулі:

- Управління користувачами;
- Управління запасами;
- Обробка замовлень;
- Відстеження продажів;
- Генерація звітів.

Взаємодія з клієнтами здійснюється через REST API. Розроблено набір RESTful API ендпоінтів, їх повний список наведено у Додатку В.

Для забезпечення безпеки застосовується Spring Security з аутентифікацією на основі JWT. Нижче на рисунках 3.1 та 3.2 наведено успішний запит на аутентифікацію.



```
###
POST http://localhost:8080/rest/users/auth/token
Content-Type: application/json

{
  "username": "admin",
  "password": "admin"
}
```

Рисунок 3. 1 Запит на аутентифікацію

```
POST http://localhost:8080/rest/users/auth/token
Show Request

HTTP/1.1 200
> (Headers) ...Content-Type: application/json...

{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9\{n.eyJzdWIiOiJqb2huZG91IiwiaXhwIjoxNzE2MDE5NTE2fQ.4gEbhPT3Q_TwN2Foh0HUgkg
}
Response file saved.
> 2025-05-21T033113.200.json

Response code: 200; Time: 173ms (173 ms); Content length: 141 bytes (141 B)
```

Рисунок 3. 2 Відповідь на запит аутентифікації

Забезпечено повне покриття коду Unit тестами. На рисунках 3.3 та 3.4 наведено успішне проходження тестів обробника запитів довідника постачальників

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `ProductsTests` and `SupplierControllerTest`.
- Editor:** Displays the `SupplierControllerTest.java` file. The `create()` method is highlighted, which uses `MockMvc` to perform a POST request to `/rest/admin-ui/suppliers` with a JSON body.
- Run Console:** Shows the execution of the `SupplierControllerTest` class. The output indicates that all 7 tests passed within 1 second and 80 milliseconds.
- Test Results:** A detailed view of the `create()` test result is shown, including the mock response status (200), headers, content type (application/json), and the JSON body.

Рисунок 3. 3 Тести до обробника запитів довідника постачальників

Unnamed

information_schema

mysql

performance_schema

pets

phpmyadmin

test

warehousedb

categories

customers

deliveries

employees

orders

order_details

payments

products

sales

sale_details

stores

stores_stock

suppliers

supplies

users

warehouses

warehouse_stock

warehousedb.suppliers: 2 строк (точно)

Далее

#	id	name	phone	email	created_at
1	1	John Doe	0998765434	john@email.com	2025-05-21 02:15:45
2	3	Oleg Petrov	345678989	oleg@mail.com	2025-05-21 00:00:00

Рисунок 3. 4 Поява нового постачальника в базі даних після тестового запиту на його створення

Висновок

Реалізація бекенд-підсистеми для інформаційної системи мережі магазинів "Південний стиль" виконана успішно. Використання сучасних технологій забезпечило надійність, масштабованість і безпеку системи. Тестування підтвердило її відповідність вимогам, а розгортання на сервері гарантує доступність для користувачів.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Розробка бекенд-системи для мережі магазинів "Південний стиль" передбачає інтенсивну інтелектуальну працю програмістів, пов'язану з тривалим використанням комп'ютерної техніки. Відповідно до Закону України "Про охорону праці" та нормативних актів, роботодавець зобов'язаний забезпечити безпечні умови праці, мінімізувати вплив шкідливих факторів та запобігати професійним захворюванням. Основними аспектами охорони праці в даному контексті є:

- Дотримання санітарних норм роботи з екранними пристроями (ДСанПіН 3.3.2.007-98). [16]
- Організація ергономічного робочого місця.
- Забезпечення психоемоційного комфорту та запобігання професійному вигоранню.
- Регулярні медичні огляди працівників (Наказ МОЗ №246 від 21.05.2007).

Шкідливі та небезпечні фактори праці програміста
Праця програміста супроводжується низкою шкідливих факторів, які можуть негативно впливати на здоров'я:

Фізіологічні:

- Навантаження на зір через тривалу роботу з моніторами може викликати синдром комп'ютерного зору, що проявляється втомою очей, головним болем, сухістю та почервонінням.
- Порушення постави та м'язово-скелетні розлади (остеохондроз, синдром зап'ястного каналу).
- Вплив електромагнітного випромінювання від комп'ютерів також становить потенційний ризик.

Психологічні:

- Високий рівень стресу, тиск дедлайнів, соціальна ізоляція та необхідність постійного оновлення знань можуть сприяти хронічному стресу, професійному вигоранню чи депресії.

Організаційні:

- Порушення режиму праці та неналежна організація робочого місця [17]

Ергономічні та санітарні вимоги до робочого місця

Робочі місця програмістів повинні відповідати вимогам:

Простір: Мінімум 4.5 м² на працівника (згідно з ДСанПіН 3.3.2.007-98).

Обладнання:

- Регульований стіл (висота 680-800 мм)
- Крісло має бути регульованим, обертальним, із підтримкою спини та хребта. За потреби передбачається підставка для ніг.
- Монітор з антибликовим покриттям повинен бути розташований перпендикулярно до лінії зору на відстані 600-700 мм від очей.
- Клавіатура з матовою поверхнею, має розміщуватися на відстані 100-300 мм від краю столу з регульованим нахилом від 5 до 15 градусів

Мікроклімат:

- Температура: 22-24°C, вологість 40-60% (ДСН 3.3.6.042-99).
- Вентиляція та відсутність шуму понад 50 дБ (ДСН 3.3.6.037-99).

Освітлення:

Освітлення має створювати належний контраст між екраном та оточуючим середовищем, відповідаючи вимогам ДСанПіН 3.3.2.007-98. Вікна повинні бути обладнані регульованими жалюзі чи шторами, а комп'ютери розміщені так, щоб світло падало на екрани з півночі чи північного сходу. Також важлива відсутність прямих джерел світла, що створюють відблиски. Рівень освітленості має складати 300-500 лк.

Нормативні вимоги до організації праці

Режим роботи:

- Перерви кожні 2 години тривалістю 10-15 хв. для гімнастики очей та розминки (згідно з ДСанПіН 3.3.2.007-98).
- Обмеження безперервної роботи з монітором до 6 годин на день.

Навчання:

- Інструктаж з техніки безпеки під час роботи з електрообладнанням.
- Тренінги з ергономіки та профілактики професійних захворювань.
- Медичні огляди:
- Попередній та періодичні огляди (1 раз на 2 роки) для виявлення порушень зору, опорно-рухового апарату.

Щоденні процедури та заборони

Щоденно перед початком роботи екранні пристрої необхідно очищати від пилу та інших забруднень. Приміщення потребують щоденного вологого прибирання та регулярного провітрювання.

Після завершення роботи пристрої слід відключати від електромережі. У разі аварійної ситуації необхідно негайно відключити пристрій.

Заборони:

- Виконувати технічне обслуговування, ремонт чи налагодження екранних пристроїв на робочому місці під час роботи.
- Відключати захисні пристрої чи вносити несанкціоновані зміни до конструкції пристроїв.
- Працювати з пристроями, що мають нехарактерні сигнали, нестабільне зображення чи інші несправності.
- Розташовувати комп'ютери в підвальних приміщеннях чи поблизу джерел тепла (менше 1 метра).
- Використовувати несправні монітори чи матричні принтери без антивібраційних чохлів.

Покарання за невідповідність

Відповідно до статті 43 Закону України "Про охорону праці", порушення законодавства про охорону праці тягне за собою накладення штрафів. Максимальний розмір штрафу не може перевищувати 5% середньомісячного фонду заробітної плати за попередній рік. Несплата штрафу призводить до нарахування пені у розмірі 120% річних облікової ставки Національного банку України за кожен день прострочення. Сплата штрафу не звільняє від усунення виявлених порушень.

Дотримання вимог охорони праці є необхідною умовою для створення безпечного та продуктивного робочого. Впровадження ергономічних стандартів, регулярних перерв, медичних оглядів та належного навчання дозволяє значно знизити ризик професійних захворювань і підвищити ефективність роботи команди розробників. Ці заходи не лише відповідають законодавчим вимогам, але й сприяють загальному добробуту працівників.

ВИСНОВОК

У процесі виконання дипломного проєкту було розроблено підсистему бекенду інформаційної системи мережі магазинів одягу "Південний стиль", яка оптимізує ключові бізнес-процеси: управління товарними запасами, обробку замовлень, закупівлі та аналітику продажів.

На основі аналізу предметної області сформульовано вимоги до системи та спроектовано базу даних, що охоплює всі необхідні сутності: товари, замовлення, клієнтів, постачальників, склади, продажі та фінансові операції. Система реалізована з використанням сучасних технологій (Java 21, Spring Boot, MySQL), що забезпечує зручну взаємодію для різних категорій користувачів — продавців-консультантів, адміністраторів, бухгалтерів, маркетологів та IT-персоналу — через REST API та веб-інтерфейс.

У межах реалізації передбачено функціонал для автоматизації документообігу, інтеграції з зовнішніми сервісами (платіжні шлюзи, ПРРО), формування звітів, а також інструменти адміністрування користувачів, товарів і замовлень. Результати тестування, проведеного за допомогою JUnit і Postman, показали стабільну роботу системи без помилок, із повною відповідністю функціональним і нефункціональним вимогам.

Розроблена підсистема бекенду має значну практичну цінність для мережі магазинів одягу, сприяючи підвищенню ефективності управління, зменшенню операційних витрат і покращенню якості прийняття управлінських рішень. Система може бути адаптована до потреб інших роздрібних мереж і масштабуватися шляхом додавання нових модулів, наприклад, для аналітики чи інтеграції з мобільними додатками.

Таким чином, поставлена мета дипломного проєкту досягнута, усі завдання виконано, а система готова до практичного впровадження в мережу "Південний стиль" або подальшого розвитку відповідно до потреб бізнесу.

ПЕРЕЛІК ПОСИЛАНЬ

1. REST API - Основи розробки. - [Електронний ресурс] - <https://restfulapi.net/>
2. JSON Web Token (JWT) - Вступ до технології. - [Електронний ресурс] - <https://jwt.io/introduction/>
3. Hibernate ORM - Офіційна документація. - [Електронний ресурс] - <https://hibernate.org/orm/documentation/>
4. Ubuntu Server - Офіційний сайт. - [Електронний ресурс] - <https://ubuntu.com/server>
5. Spring - Офіційний сайт. - [Електронний ресурс] - <https://spring.io/>
- MySQL - Офіційний сайт. - [Електронний ресурс] - <https://www.mysql.com/>
6. Spring Security - Офіційна документація. - [Електронний ресурс] - <https://spring.io/projects/spring-security>
7. JUnit - Офіційна документація. - [Електронний ресурс] - <https://junit.org/junit5/docs/current/user-guide/>
8. Postman - Офіційний сайт. - [Електронний ресурс] - <https://www.postman.com/>
9. IntelliJ IDEA - Офіційний сайт. - [Електронний ресурс] - <https://www.jetbrains.com/idea/>
10. HeidiSQL - Офіційний сайт. - [Електронний ресурс] - <https://www.heidisql.com/>
11. Apache Friends - XAMPP. - [Електронний ресурс] - <https://www.apachefriends.org/ru/index.html>
12. Dilovod - Серверна частина. - [Електронний ресурс] - <https://dilovod.ua/v10/server.php>
13. USAP.ONLINE - Автоматизація бізнес-процесів. - [Електронний ресурс] - <https://my.usap.online/>

14. RemOnline - Управління складом. - [Електронний ресурс] - https://remonline.ua/warehouse-management/?utm_source=google&utm_medium=cpc&utm_campaign=search_inventory-management_ua&gad_source=1&gclid=Cj0KCQjwzYLABhD4ARIsALySuCR0NKMA9BhXvt8BfecbC7BomHdC2aChJQvAb5jhLFMVPIGwDxZgWgAaAtfBEALw_wcB

15. Конноллі Т., Бегг К. Бази даних: проектування, реалізація та супровід. Теорія і практика. - К.: Вільямс, 2004. - 1152 с.

16. ДСТУ 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами ЕОМ. - [Електронний ресурс] - <https://zakon.rada.gov.ua/laws/show/z0930-98>

17. Охорона праці при роботі з комп'ютером. - [Електронний ресурс] - <https://www.sop.com.ua/article/183-ohoron-prats-pri-robot-z-kompyuterom>

ДОДАТОК А

Технічне завдання

1. Вступ

Назва розроблюваного проекту: «Південний стиль Бекенд»

Коротка характеристика використання: Розроблювана підсистема бекенду призначена для автоматизації ключових бізнес-процесів мережі магазинів одягу "Південний стиль", включаючи управління товарними запасами, обробку замовлень, управління клієнтськими даними, закупівлями, складськими операціями та генерацію звітів. Система забезпечує централізований доступ до даних через REST API, інтеграцію з зовнішніми сервісами (платіжні шлюзи, ПРРО, служби доставки) і зручний веб-інтерфейс для різних категорій користувачів.

2. Підстави для розробки

2.1 Підстави для проведення розробки

Розробка здійснюється на підставі завдання на дипломну роботу бакалавра.

2.2 Найменування та умовне позначення теми розробки

Найменування розробки: «Розроблення підсистеми бекенду інформаційної системи мережі магазинів одягу "Південний стиль"».

Умовне позначення: «Південний стиль Бекенд».

3. Призначення розробки

3.1 Функціональне призначення

Система забезпечує автоматизацію бізнес-процесів мережі магазинів, включаючи:

- Управління каталогом товарів (додавання, редагування, видалення, пошук).
- Обробку замовлень (створення, відстеження статусу, інтеграція з касами).
- Управління складськими запасами (облік, переміщення, заявки на поповнення).

- Управління клієнтськими даними (профілі, історія покупок, лояльність).
- Генерацію звітів (продажі, запаси, фінанси).
- Інтеграцію з платіжними шлюзами, службами доставки та ПРРО.

3.2 Експлуатаційне призначення

Система призначена для стабільної роботи в умовах реального використання, забезпечуючи доступ через веб-браузер із комп'ютерів у магазинах, складах і головному офісі.

3.3 Вимоги до надійності

Система має забезпечувати:

Стабільну роботу з мінімальним часом простою (uptime 99.9%).

Цілісність даних через автоматичне резервне копіювання (щоденно).

Захист від несанкціонованого доступу (JWT-аутентифікація, шифрування).

Можливість відновлення після збоїв без втрати даних.

3.4 Вимоги до експлуатації

Система доступна через веб-браузер. Інтерфейс інтуїтивно зрозумілий, не вимагає спеціальних навичок.

3.5 Вимоги до складу та параметрів технічних засобів

Сервер: Intel Xeon E5, 32 ГБ RAM, SSD 500 ГБ, Gigabit Ethernet, ДБЖ 1000 ВА.

Клієнтські комп'ютери: Intel i5, 16 ГБ RAM, SSD 50 ГБ, Wi-Fi/Ethernet, принтер (формат A4, 4 кольори, USB).

Мережа: Wi-Fi роутер (IEEE 802.11 b/g/n/ac/ax, 2.4/5 ГГц, 4 LAN-порти).

3.7 Вимоги до програмної документації

Документація включає:

- Технічне завдання.
- Інструкцію системного адміністратора.
- Повний список ендпоінтів системи

4. Стадії та етапи розробки

4.1 Формування вимог до системи

Аналіз предметної області, визначення функціональних вимог, вимог щодо безпеки та технічних характеристик.

Заплановані терміни: 01.02.2025 – 15.03.2025.

4.2 Розробка концепції

Визначення архітектури, компонентів, принципів роботи, вибір технологій (Java, Spring Boot, MySQL) та інтеграція з зовнішніми сервісами.

Заплановані терміни: 16.03.2025 – 15.04.2025.

4.3 Реалізація проекту

Розробка бази даних, бекенд-сервісів, тестування (JUnit, Postman).

Заплановані терміни: 16.04.2025 – 31.05.2025.

4.4 Охорона праці і техніка безпеки

Розробка розділу щодо безпеки експлуатації системи.

Заплановані терміни: 01.05.2025 – 15.05.2025.

5. Порядок прийому та контролю

Для прийому програмного продукту надається повний комплект документації, опис функціоналу, структура бази даних і результати тестування. Процедура прийому проводиться представником замовника за участю розробника.

У разі виявлення недоліків складається акт із переліком помилок, який підписується обома сторонами та затверджується керівниками. Розробник усуває недоліки протягом одного місяця. Після виправлення проводиться повторна перевірка. Якщо система відповідає вимогам, підписується підсумковий акт прийому. У разі повторних критичних помилок приймання відкладається до їх усунення.

ДОДАТОК Б

Інструкція для системного адміністратора з розгортання та налаштування системи "Південний стиль"

Ця інструкція описує процес розгортання та налаштування підсистеми бекенду інформаційної системи мережі магазинів одягу "Південний стиль" на сервері з Ubuntu Server 22.04 LTS.

Вам необхідно мати:

- JAR-файл додатка (pivdenny-styl.jar), та SQL-файл зі схемою бази даних (schema.sql).
- Права sudo на сервері.
- Стабільне підключення до Інтернету.

Крок 1. Оновіть пакети та встановіть необхідні утиліти:

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y curl unzip
```

Крок 2. Встановіть OpenJDK 17:

```
sudo apt install -y openjdk-17-jdk
java -version
```

Переконайтеся, що вивід показує Java 21.

Крок 3. Встановіть MySQL-сервер:

```
sudo apt install -y mysql-server
sudo mysql_secure_installation
```

Відповідайте на запитання для забезпечення безпеки (рекомендується встановити пароль для root і видалити тестові бази).

Увійдіть до MySQL:

```
sudo mysql
```

Створіть базу даних і користувача:

```
CREATE DATABASE pivdenny_styl;
```

```
CREATE USER 'pivdenny_user'@'localhost' IDENTIFIED BY
'strong_password';
GRANT ALL PRIVILEGES ON pivdenny_styl.* TO
'pivdenny_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Імпортуйте схему бази даних (якщо є файл `schema.sql`):

```
mysql -u pivdenny_user -p pivdenny_styl < /path/to/schema.sql
```

Замініть `/path/to/schema.sql` на актуальний шлях до файлу.

Крок 4. Скомпонуйте JAR-файл у вашому проєкті за допомогою Maven:

```
mvn clean package
```

Створіть директорію на сервері:

```
sudo mkdir /opt/pivdenny-styl
sudo chown $USER:$USER /opt/pivdenny-styl
```

Перенесіть JAR-файл на сервер:

```
scp target/pivdenny-styl.jar user@server:/opt/pivdenny-styl/
```

Замініть `user@server` на ваші дані доступу.

Крок 5. Створіть файл `application.properties` у директорії `/opt/pivdenny-styl`:

```
nano /opt/pivdenny-styl/application.properties
```

Додайте наступні параметри:

```
spring.datasource.url=jdbc:mysql://localhost:3306/pivdenny_styl
spring.datasource.username=pivdenny_user
spring.datasource.password=strong_password
server.port=8080
jwt.secret=your_secret_key_here
jwt.expiration=3600000
# Налаштування для інтеграцій (отримайте ключі від відповідних
сервісів)
payment.liqpay.public_key=your_public_key
payment.liqpay.private_key=your_private_key
delivery.novaposhta.api_key=your_api_key
```

- Замініть `strong_password` на ваш пароль для MySQL.
- Замініть `your_secret_key_here` на безпечний випадковий рядок для JWT.

- Отримайте API-ключі для платіжних шлюзів та служб доставки і вкажіть їх у відповідних полях.

Збережіть файл і закрийте редактор.

Крок 6. Створіть systemd-сервіс для автоматичного запуску додатка:

```
sudo nano /etc/systemd/system/pivdenny-styl.service
```

Додайте наступний вміст:

```
[Unit]
Description=Pivdenny Styl Backend Service
After=network.target

[Service]
User=your_username
WorkingDirectory=/opt/pivdenny-styl
ExecStart=/usr/bin/java -jar pivdenny-styl.jar
Restart=always

[Install]
WantedBy=multi-user.target
```

- Замініть `your_username` на ім'я користувача, який запускатиме сервіс (наприклад, `ubuntu`).
- Збережіть файл і закрийте редактор.

Активуйте сервіс:

```
sudo systemctl daemon-reload
sudo systemctl enable pivdenny-styl.service
sudo systemctl start pivdenny-styl.service
```

Перевірте статус сервісу:

```
sudo systemctl status pivdenny-styl.service
```

При успішному запуску ви побачите активний статус сервісу та логи запуску.

Крок 7. Налаштування фаєрвола

Дозвольте вхідні підключення на порт 8080:

```
sudo ufw allow 8080/tcp
sudo ufw enable
```

Перевірте статус фаєрвола:

```
sudo ufw status
```

Крок 8. Перевірка розгортання

З клієнтського комп'ютера відкрийте веб-браузер і перейдіть за адресою:

```
http://<server_ip>:8080
```

Замініть <server_ip> на IP-адресу або доменне ім'я сервера. Переконайтеся, що веб-інтерфейс або API відповідає коректно.

Крок 9. Налаштування резервного копіювання

Створіть скрипт для резервного копіювання бази даних:

```
sudo nano /usr/local/bin/backup_db.sh
```

Додайте вміст:

```
#!/bin/bash
```

```
mysqldump -u pivdenny_user -pstrong_password pivdenny_styl >  
/backups/pivdenny_styl_$(date +%F).sql
```

- Замініть strong_password на ваш пароль.
- Збережіть і закрийте файл.

Зробіть скрипт виконуваним:

```
sudo chmod +x /usr/local/bin/backup_db.sh
```

Створіть директорію для резервних копій:

```
sudo mkdir /backups
```

```
sudo chown $USER:$USER /backups
```

Налаштуйте cron для щоденного виконання:

```
crontab -e
```

Додайте рядок для запуску щодня о 2:00:

```
0 2 * * * /usr/local/bin/backup_db.sh
```

Крок 10. Моніторинг логів

Переглядайте логи додатка за допомогою journalctl:

```
journalctl -u pivdenny-styl.service
```

Для перегляду логів у реальному часі:

```
journalctl -u pivdenny-styl.service -f
```

Додаткові рекомендації

- Регулярно оновлюйте сервер (`sudo apt update && sudo apt upgrade`).
- Перевіряйте логи на наявність помилок і забезпечте фізичний захист сервера (обмежений доступ до серверної кімнати).

ДОДАТОК В

Повний перелік ендпоінтів

Таблиця В.1

Ендпоінти на обробку запитів щодо користувачів

Назва функції	Відносна URL	HTTP метод	Опис функції
register_user	/auth/register	POST	Реєстрація нового користувача (створення облікового запису).
login_user	/auth/login	POST	Авторизація користувача (вхід до системи, отримання JWT токена).
logout_user	/auth/logout	POST	Вихід користувача із системи (завершення сеансу).
get_all_users	/users	GET	Отримання списку усіх користувачів.
get_user_by_id	/users/{id}	GET	Отримання інформації про конкретного користувача (за ID).
create_user	/users	POST	Створення нового користувача (адміністратором).
update_user	/users/{id}	PUT	Оновлення даних користувача (роль, пароль тощо).
delete_user	/users/{id}	DELETE	Видалення користувача.

Таблиця В.2

Ендпоінти на обробку запитів щодо співробітників

Назва функції	Відносна URL	HTTP метод	Опис функції
get_all_employees	/employees	GET	Отримання списку усіх співробітників.
get_employee_by_id	/employees/{id}	GET	Отримання інформації про конкретного співробітника.
create_employee	/employees	POST	Створення нового співробітника.
update_employee	/employees/{id}	PUT	Оновлення даних співробітника.
delete_employee	/employees/{id}	DELETE	Видалення співробітника.

Таблиця В.3

Ендпоінти на обробку запитів щодо клієнтів

Назва функції	Відносна URL	HTTP метод	Опис функції
get_all_customers	/customers	GET	Отримання списку усіх клієнтів.
get_customer_by_id	/customers/{id}	GET	Отримання інформації про конкретного клієнта.
create_customer	/customers	POST	Створення нового клієнта.
update_customer	/customers/{id}	PUT	Оновлення даних клієнта (контактна інформація, бонуси тощо).
delete_customer	/customers/{id}	DELETE	Видалення клієнта.

Ендпоінти на обробку запитів щодо товарів

Назва функції	Відносна URL	HTTP метод	Опис функції
get_all_categories	/categories	GET	Отримання списку усіх категорій товарів.
get_category_by_id	/categories/{id}	GET	Отримання інформації про конкретну категорію.
create_category	/categories	POST	Створення нової категорії товарів.
update_category	/categories/{id}	PUT	Оновлення даних категорії товарів.
delete_category	/categories/{id}	DELETE	Видалення категорії товарів.
get_all_products	/products	GET	Отримання списку усіх товарів.
get_product_by_id	/products/{id}	GET	Отримання інформації про конкретний товар.
create_product	/products	POST	Створення нового товару.
update_product	/products/{id}	PUT	Оновлення даних товару.
delete_product	/products/{id}	DELETE	Видалення товару.

Ендпоїнти на обробку запитів щодо замовлень та продажів

Назва функції	Відносна URL	HTTP метод	Опис функції
get_all_orders	/orders	GET	Отримання списку усіх замовлень.
get_order_by_id	/orders/{id}	GET	Отримання деталей конкретного замовлення.
create_order	/orders	POST	Створення нового замовлення (оформлення продажу/онлайн-замовлення).
update_order	/orders/{id}	PUT	Оновлення замовлення (статусу, деталей тощо).
delete_order	/orders/{id}	DELETE	Видалення (скасування) замовлення.
get_order_items	/orders/{id}/items	GET	Отримання списку позицій (товарів) у конкретному замовленні.
add_order_item	/orders/{id}/items	POST	Додавання нової позиції товару до замовлення.
update_order_item	/orders/{id}/items/{itemId}	PUT	Оновлення позиції замовлення (зміна кількості товару).
remove_order_item	/orders/{id}/items/{itemId}	DELETE	Видалення позиції (товару) із замовлення.

get_all_sales	/sales	GET	Отримання списку усіх продажів (чеків).
get_sale_by_id	/sales/{id}	GET	Отримання деталей конкретного продажу (чека).
create_sale	/sales	POST	Створення нового запису про продаж (оформлення продажу в магазині).
update_sale	/sales/{id}	PUT	Оновлення даних продажу.
delete_sale	/sales/{id}	DELETE	Видалення запису про продаж.
get_sale_items	/sales/{id}/items	GET	Отримання списку позицій у конкретному продажу.

Ендпоїнти на обробку запитів щодо платежів та доставки

Назва функції	Відносна URL	HTTP метод	Опис функції
get_all_payments	/payments	GET	Отримання списку усіх платежів.
get_payment_by_id	/payments/{id}	GET	Отримання інформації про конкретний платіж.
create_payment	/payments	POST	Створення нового запису про платіж (оплату замовлення).
update_payment	/payments/{id}	PUT	Оновлення даних платежу.
delete_payment	/payments/{id}	DELETE	Видалення запису про платіж.
get_all_deliveries	/deliveries	GET	Отримання списку усіх доставок замовлень.
get_delivery_by_id	/deliveries/{id}	GET	Отримання інформації про конкретну доставку.
create_delivery	/deliveries	POST	Створення нового запису доставки для замовлення.
update_delivery	/deliveries/{id}	PUT	Оновлення даних доставки.
delete_delivery	/deliveries/{id}	DELETE	Видалення запису про доставку.

Таблиця В.7

Ендпоінти на обробку запитів щодо постачання та запасів

Назва функції	Відносна URL	HTTP метод	Опис функції
get_all_suppliers	/suppliers	GET	Отримання списку усіх постачальників.
get_supplier_by_id	/suppliers/{id}	GET	Отримання інформації про конкретного постачальника.
create_supplier	/suppliers	POST	Додавання нового постачальника.
update_supplier	/suppliers/{id}	PUT	Оновлення даних постачальника.
delete_supplier	/suppliers/{id}	DELETE	Видалення постачальника.
get_all_supplies	/supplies	GET	Отримання списку усіх поставок.
get_supply_by_id	/supplies/{id}	GET	Отримання інформації про конкретну поставку.
create_supply	/supplies	POST	Створення нового

			запису про поставку.
update_supply	/supplies/{id}	PUT	Оновленн я даних поставки.
delete_supply	/supplies/{id}	DEL ETE	Видаленн я запису про поставку.
get_all_warehouses	/warehouses	GET	Отриманн я списку усіх складів.
get_warehouse_by_id	/warehouses/{id}	GET	Отриманн я інформаці ї про конкретни й склад.
create_warehouse	/warehouses	POST	Створення нового складу.
update_warehouse	/warehouses/{id}	PUT	Оновленн я даних складу.
delete_warehouse	/warehouses/{id}	DEL ETE	Видаленн я складу.
get_warehouse_stock	/warehouses/{id}/stock	GET	Отриманн я списку товарів на складі.
get_product_stock_in_warehouse	/warehouses/{id}/stock/{productId}	GET	Залишок конкретно го товару на складі.
update_warehouse_stock	/warehouses/{id}/stock/{productId}	PUT	Оновленн я залишку товару на складі.
get_all_stores	/stores	GET	Отриманн я списку

			усіх магазинів.
get_store_by_id	/stores/{id}	GET	Отримання інформації про конкретний магазин.
create_store	/stores	POST	Додавання нового магазину.
update_store	/stores/{id}	PUT	Оновлення даних магазину.
delete_store	/stores/{id}	DELETE	Видалення магазину.
get_store_stock	/stores/{id}/stock	GET	Отримання залишків товару в магазині.
get_product_stock_in_store	/stores/{id}/stock/{productId}	GET	Залишок конкретного товару в магазині.
update_store_stock	/stores/{id}/stock/{productId}	PUT	Оновлення залишку товару в магазині.
transfer_between_warehouses	/warehouses/{fromId}/transfer_to_warehouse/{toId}	POST	Переміщення товарів між складами.
transfer_to_store	/warehouses/{id}/transfer_to_store/{storeId}	POST	Переміщення товару зі складу в магазин.

Таблиця В.8

Ендпоінти на обробку запитів на формування звітів

Назва функції	Відносна URL	HTTP метод	Опис функції
get_orders_report	/reports/orders	GET	Отримання звіту щодо замовлень.
get_sales_report	/reports/sales	GET	Отримання звіту щодо продажів.
get_category_sales_report	/reports/category_sales	GET	Звіт щодо продажів певної категорії.
get_supplies_report	/reports/supplies	GET	Звіт щодо поставок товарів.
get_financial_report	/reports/financial	GET	Фінансовий звіт (доходи/витрати).
get_tax_report	/reports/tax	GET	Дані для податкової декларації.