

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

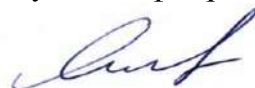
Національний університет кораблебудування імені адмірала Макарова

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.

" ____ " _____ 2026 р.

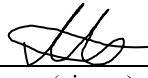
КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

«Розробка WEB-додатку обліку та контролю складських запасів»

Виконав: студент групи 4147БДзст3



(підпис)

Мензатул М.С.

Керівник роботи:

к.ф.-м.н., Доцент

(посада, науковий ступінь вчене звання)



(підпис)

Коваль С.С.

м. Миколаїв – 2026 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 122 "Комп'ютерні науки"
 Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Гайда А.Ю.

" ____ " ____ 2026 р.

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

Студенту Мензатул Марку Сергійовичу
 (прізвище, ім'я, по батькові)

1. Тема роботи: Розробка WEB-додатку обліку та контролю складських запасів
 Керівник роботи Коваль Сергій Станіславович
 Затверджені наказом ректора № 270-уч від "31" березня 2026 року.
2. Термін подання роботи: 22 червня 2026 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) 1. Аналіз предметної області системи обліку товарно-матеріальних цінностей та постановка задачі. 2. Проектування web-застосунку системи обліку товарно-матеріальних цінностей. 3. Розробка web-застосунку системи обліку товарно-матеріальних цінностей. 4. Охорона праці при роботі з інформаційною системою.
5. Перелік презентаційних матеріалів _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Коваль Сергій Станіславович	02.02.2026	23.02.2026
2	Коваль Сергій Станіславович	23.02.2026	20.03.2026
3	Коваль Сергій Станіславович	20.03.2026	24.04.2026
4	Коваль Сергій Станіславович	24.04.2026	01.06.2026

7. Дата видачі завдання 02 лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

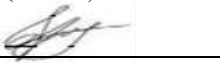
№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі	02.02.2026	
2	Постановка задачі	16.02.2026	
3	Розробка концепції	23.02.2026	
4	Розробка проекту ІС	23.03.2026	
5	Реалізація проекту	27.04.2026	
6	Розробка питань з охорони праці	01.06.2026	
7	Розробка графічних матеріалів	05.06.2026	
8	Подання роботи на перевірку на плагіат	08.06.2026	
9	Подання роботи рецензенту	15.06.2026	
10	Подання роботи на захист	22.06.2026	

Студент


 (підпис)

Мензатул М.С.
 (ПІБ)

Керівник роботи


 (підпис)

Коваль С.С.
 (ПІБ)

АНОТАЦІЯ

Тема дипломної роботи - розробка WEB-додатку обліку та контролю складських запасів.

Проаналізовано тематику та поставлено мету: розробити web-застосунок який автоматизує основні процеси обліку та контролю складських запасів для покращення роботи складу.

Для досягнення поставленої мети були вирішені наступні завдання: опис та аналіз предметної області, визначення недоліків та завдань, проєктування та розробка web-застосунку.

Результат такої роботи - зручний web-застосунок для ефективного обліку та контролю складських запасів.

Робота має 57 сторінки, складається з 4 розділів, містить 8 рисунків, 3 таблицю та 2 додатки. В першому додатку міститься інструкція користувача, а в другому - фрагмент програмного коду.

Ця робота виконана українською мовою.

Ключові слова: системи обліку товарно-матеріальних цінностей (ТМЦ), web-застосунок, система, інформаційна система, ефективність та автоматизація.

ANNOTATION

The topic of the thesis is the development of a web-application for inventory accounting and control.

The subject area was analyzed and the following goal was set: to develop a web-application that automates the core processes of inventory accounting and control to improve warehouse operations.

To achieve this goal, the following tasks were completed: description and analysis of the subject domain, identification of shortcomings and objectives, and the design and development of the web application.

The result of this work is a user-friendly web-application for efficient inventory accounting and control.

The work consists of 57 pages, 4 chapters, 8 figures, 3 tables, and 2 supplements. The first appendix contains the user manual, and the second contains a source code snippet.

This work is written in Ukrainian.

Keywords: inventory accounting systems, web application, system, information system, efficiency and automation.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМИ ОБЛІКУ ТОВАРНО-МАТЕРІАЛЬНИХ ЦІННОСТЕЙ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Опис предметної області	9
1.2 Недоліки роботи системи та шляхи їх подолання.....	11
1.3 Концепція системи.....	13
1.4 Постановка задачі	14
1.5 Висновок до розділу 1	16
РОЗДІЛ 2. ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ СИСТЕМИ ОБЛІКУ ТОВАРНО-МАТЕРІАЛЬНИХ ЦІННОСТЕЙ.....	18
2.1 Загально системні рішення.....	18
2.2 Рішення по інформаційному забезпеченню.....	21
2.2.1 Розробка концептуальної моделі даних	21
2.2.2 Розробка логічної моделі даних	23
2.3 Висновок до розділу 2	25
РОЗДІЛ 3. РОЗРОБКА WEB-ЗАСТОСУНКУ СИСТЕМИ ОБЛІКУ ТОВАРНО-МАТЕРІАЛЬНИХ ЦІННОСТЕЙ.....	27
3.1 Діаграма розгортання.....	27
3.2 Рішення з технічного забезпечення.....	29
3.3 Рішення з програмного забезпечення	30
3.3.1 Розробка бази даних системи	31
3.3.2 Серверна логіка	31
3.3.3 Клієнтська логіка	32
3.3.4 Реалізація інтерфейсу веб-застосунку	33
3.3.5 Модуль синхронізації та збереження даних	34
3.4 Висновок до розділу 3	34
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	36

4.1 Загальні положення охорони праці	36
4.2 Вимоги безпеки при роботі з комп'ютерним обладнанням.....	37
4.3 Заходи щодо зменшення впливу шкідливих виробничих факторів	38
4.4 Висновок до розділу 4	39
ВИСНОВКИ	41
СПИСОК ЛІТЕРАТУРИ.....	43
ДОДАТОК А - ІНСТРУКЦІЯ КОРИСТУВАЧА.....	45
ДОДАТОК Б - ФРАГМЕНТИ ПРОГРАМНОГО КОДУ	47

ВСТУП

Сьогодні технології стрімко розвиваються, і все більше бізнес-процесів переходить у цифровий формат. Це торкнулося й сфери складського обліку - підприємства все частіше відмовляються від паперових журналів та таблиць на користь автоматизованих систем управління запасами. Сучасний бізнес прагне до точності, швидкості та прозорості у роботі з товарно-матеріальними цінностями. З метою підвищити ефективність складських операцій та мінімізувати вплив людського фактора було розроблено web-застосунок для управління складом.

Розроблення такого застосунку є актуальним, адже сьогодні навіть середні підприємства стикаються з труднощами контролю за рухом товарів у режимі реального часу. Це створює потребу в зручних цифрових рішеннях, які автоматизують приймання, відвантаження, інвентаризацію та переміщення товарів. Web-застосунок дозволяє ефективно керувати складськими процесами: вести облік залишків, фіксувати кожну операцію як незмінну транзакцію, формувати миттєві звіти про рух товарів, прибутковість та збитки.

Крім цього, застосунок забезпечує розмежування прав доступу між працівниками, підтримує контроль термінів придатності та легко інтегрується з інтернет-магазинами, CRM-системами, бухгалтерським програмним забезпеченням і сервісами доставки. Таким чином, розробка такого рішення є важливою для підвищення ефективності роботи складу та зниження операційних ризиків.

Мета роботи полягає у розробленні web-застосунку для автоматизації складського обліку.

Завдання: виконати аналіз предметної області та конкурентних рішень, сформулювати функційні вимоги, розробити систему управління товарно-матеріальними цінностями, спроектувати модель даних, реалізувати ключові модулі обліку та адміністрування, інтегрувати та впровадити застосунок.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМИ ОБЛІКУ ТОВАРНО-МАТЕРІАЛЬНИХ ЦІННОСТЕЙ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

Предметна область даного проєкту охоплює процеси управління товарно-матеріальними цінностями (ТМЦ) на підприємствах різних форм власності: від невеликих складських комплексів до розгалужених логістичних мереж. Під товарно-матеріальними цінностями розуміють сукупність матеріальних активів підприємства, що включають сировину, напівфабрикати, готову продукцію, інвентар, інструменти та інші матеріальні об'єкти, що використовуються або реалізуються в процесі господарської діяльності.

Ефективне управління складом є критично важливим елементом операційної діяльності підприємства. Від точності обліку залишків, своєчасності оновлення даних та швидкості пошуку інформації безпосередньо залежить конкурентоспроможність компанії, її здатність виконувати зобов'язання перед клієнтами та мінімізувати операційні витрати.

В умовах сучасного ринку підприємства стикаються з рядом об'єктивних викликів у сфері складського обліку: зростання асортименту товарів, збільшення кількості операцій, розширення мережі складів та необхідність забезпечення актуальності даних у режимі реального часу. Традиційні паперові журнали та таблиці в електронних таблицях вже не відповідають вимогам сучасного бізнесу.

На сьогодні ринок програмного забезпечення для управління складом представлений широким спектром рішень - від простих хмарних додатків до потужних ERP-систем. Проведений аналіз конкурентного середовища (таб. 1.1) включав чотири основні продукти, які є найбільш репрезентативними для різних сегментів ринку.

Zoho Inventory - хмарна система з потужною інтеграцією з провідними маркетплейсами (Amazon, eBay, Shopify) та службами доставки. Забезпечує автоматизацію обробки замовлень, управління декількома складами одночасно та оснащена мобільним додатком зі вбудованим сканером штрих-кодів. Основний

недолік - висока вартість при масштабуванні бізнесу та відсутність україномовного інтерфейсу, що ускладнює впровадження для вітчизняних підприємств.

Sortly спеціалізується на максимальній візуалізації інформації: замість традиційних таблиць використовуються фотокартки товарів. Система відзначається найнижчим порогом входження і є ідеальною для обліку інструментів, обладнання або невеликих складів. Проте функціонал є недостатнім для складних бухгалтерських операцій або виробничих процесів.

Dilovod - вітчизняна хмарна система, що забезпечує повноцінний облік (склад, бухгалтерія, зарплати) відповідно до українських стандартів. Підтримує фіскалізацію та формування детальної звітності. Недолік - перевантажений інтерфейс, що потребує значного часу на навчання персоналу.

Odoo - модульна система з широкими можливостями кастомізації. Дозволяє починати з базового модуля складу та поступово додавати CRM, HR, маркетинг. Має сучасний дизайн та можливість повного налаштування під потреби бізнесу. Суттєвий недолік - висока вартість впровадження та підтримки.

Таблиця 1.1 - Порівняльний аналіз конкурентних систем

Система	Переваги	Недоліки	Цільова аудиторія
Zoho Inventory	Інтеграція з маркетплейсами, мультисклад, мобільний додаток	Висока вартість, лише англomовний інтерфейс	Середній та великий бізнес
Sortly	Простота, візуалізація, низький поріг входу	Обмежений функціонал для складного обліку	Малий бізнес, склади інвентарю
Dilovod	Українські стандарти, фіскалізація, детальна звітність	Складний інтерфейс, потрібне навчання	Підприємства України
Odoo	Модульність, гнучке налаштування, сучасний дизайн	Дороге впровадження та підтримка	Великий бізнес, корпоративний сектор

Проведений аналіз виявив ключову ринкову нішу: відсутність доступної, інтуїтивно зрозумілої системи обліку ТМЦ, яка б відповідала українським

стандартам, мала сучасний UX-дизайн та забезпечувала повний функціонал без надмірних витрат на впровадження. Саме цю прогалину покликаний заповнити розроблюваний програмний продукт. Нище наведено організаційну структуру системи (рис. 1.1).



Рисунок 1.1 - Організаційна структура системи управління ТМЦ.

1.2 Недоліки роботи системи та шляхи їх подолання

Аналіз наявних рішень на ринку програмного забезпечення для управління складом та товарно-матеріальними цінностями виявив системні недоліки, що притаманні переважній більшості існуючих продуктів. Ці недоліки умовно можна розподілити на три ключові групи: технологічні, організаційні та економічні.

Технологічні недоліки:

- Відсутність захисту від від'ємних залишків - система дозволяє списувати товарів більше, ніж їх є на складі, що призводить до критичних помилок обліку.
- Неможливість відстеження повної історії операцій у вигляді незмінних транзакцій - записи можна редагувати або видаляти, що підриває цілісність даних.
- Слабка або відсутня підтримка адресного зберігання - неможливість прив'язати товар до конкретного місця на складі (ряд, стелаж, полиця, комірکا).

- Обмежені можливості інтеграції з зовнішніми системами через відсутність відкритого API або стандартизованих протоколів обміну даними.

- Відсутність механізмів сповіщень при досягненні критичного рівня залишків або закінченні терміну придатності товарів.

Організаційні недоліки:

- Відсутність гнучкої системи ролей та прав доступу - всі користувачі мають однаковий рівень доступу.

- Складний та перевантажений інтерфейс, що вимагає тривалого навчання персоналу та знижує ефективність роботи.

- Відсутність підтримки мультискладовості - неможливість одночасно вести облік на кількох складах з переміщенням між ними.

- Недостатній функціонал звітності - обмежені можливості фільтрації, відсутність аналітичних дашбордів та неможливість експорту даних.

Економічні недоліки:

- Висока вартість ліцензування та впровадження, що є непосильним для малого та середнього бізнесу.

- Відсутність прозорості тарифної політики - приховані платежі за додаткові модулі або збільшення кількості користувачів.

- Залежність від вендора та ризику втрати даних при зміні постачальника програмного забезпечення.

Таблиця 1.2 - Шляхи подолання виявлених недоліків

№	Виявлений недолік	Наслідки	Запропоноване рішення
1	Можливість від'ємних залишків	Критичні помилки обліку	Програмна блокування операцій з перевіркою на рівні БД
2	Редагування транзакцій	Втрата цілісності даних	Незмінні транзакції з повним журналом дій
3	Складний інтерфейс	Низька продуктивність, помилки	UX-орієнтований дизайн з мінімізацією кроків
4	Обмежені ролі доступу	Безпекові ризики, зловживання	Гнучка RBAC-система з 4+ рівнями ролей

Продовження таблиці 1.2

5	Відсутність інтеграцій	Ручне дублювання даних між системами	RESTful API для інтеграції з CRM, інтернет-магазинами, бухгалтерією
6	Відсутність сповіщень	Несвоєчасне реагування на критичні події	Автоматична система нотифікацій (email, push)

1.3 Концепція системи

Концепція розроблюваної інформаційної системи обліку товарно-матеріальних цінностей базується на принципі єдиного цифрового простору, що об'єднує всі аспекти складської діяльності підприємства в одній платформі. Система розробляється як сучасний хмарний веб-застосунок з мобільною адаптацією, що забезпечує доступ до даних у режимі реального часу з будь-якого пристрою та місцезнаходження.

Ключові концептуальні принципи системи:

- Принцип єдиної точки правди - всі дані про залишки, транзакції та довідники зберігаються централізовано та є актуальними для всіх користувачів одночасно.

- Принцип незмінності транзакцій - кожна операція з товарами (прихід, списання, переміщення) фіксується як незмінний запис у базі даних з повною інформацією про автора, дату та час.

- Принцип захисту від людського фактора - система автоматично перевіряє коректність операцій, унеможливорює від'ємні залишки та дублювання артикулів.

- Принцип масштабованості - архітектура системи дозволяє поступово розширювати функціонал від базового обліку до повноцінної ERP-системи.

- Принцип відкритої інтеграції - RESTful API дозволяє підключати зовнішні сервіси: інтернет-магазини, CRM, бухгалтерський софт, служби доставки.

Концептуальна модель системи (рис. 1.2) описує основні сутності предметної області та зв'язки між ними. Система складається з чотирьох ключових функціональних блоків: блоку управління довідниками, блоку операційного обліку, блоку аналітики та звітності, а також блоку адміністрування.

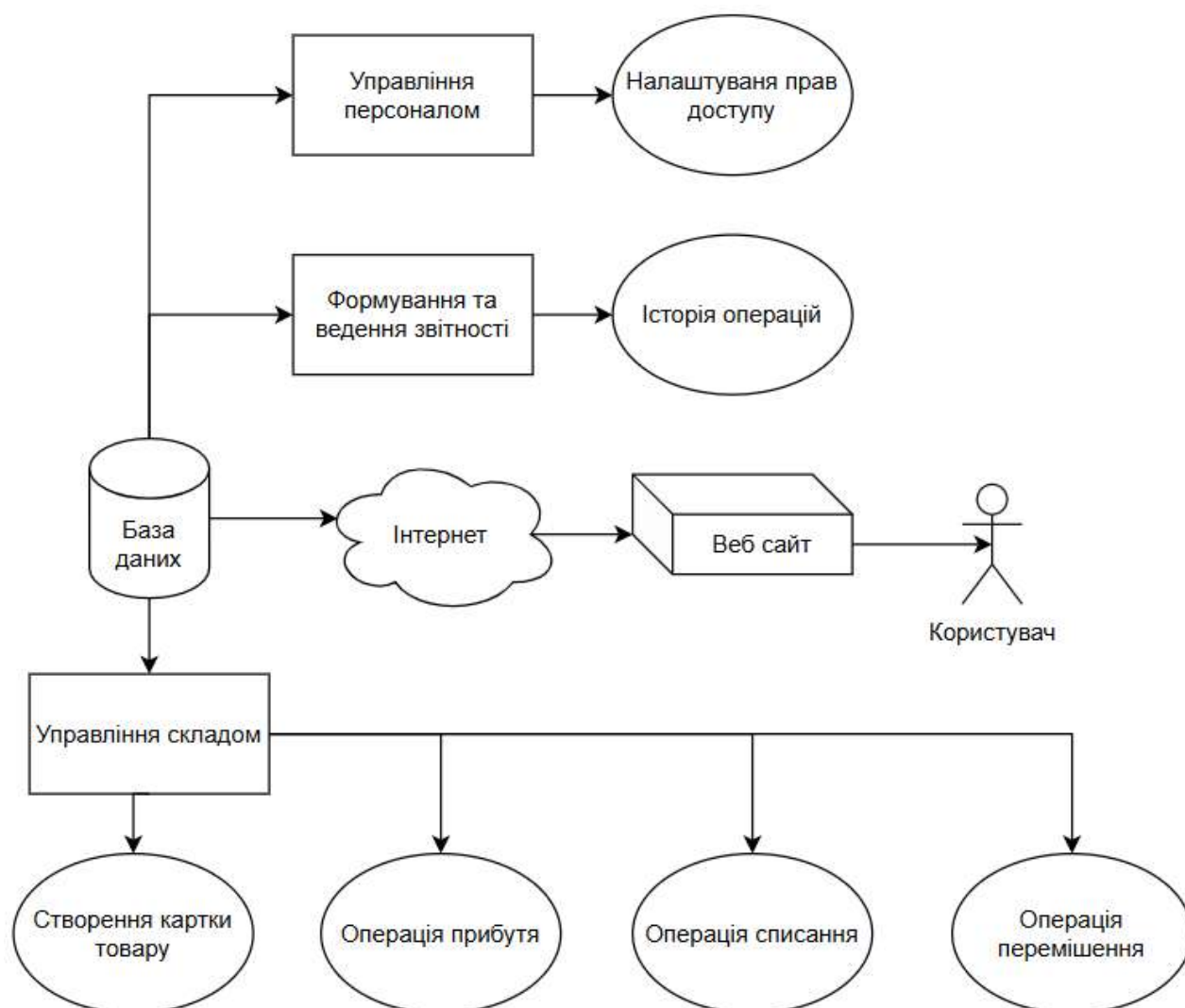


Рисунок 1.2 - Концептуальна модель системи.

Концептуальна модель також включає зовнішні інтерфейси взаємодії: мобільний додаток зі сканером штрих-кодів для операцій на складі, API-шлюз для підключення зовнішніх систем та модуль звітності з можливістю експорту в формати Excel та PDF. Ядром системи є сервіс управління залишками, який в режимі реального часу відслідковує поточний стан усіх товарів та забезпечує захист від некоректних операцій.

1.4 Постановка задачі

На основі проведеного аналізу предметної області та виявлених недоліків існуючих систем сформульовано задачу розробки інформаційної системи обліку товарно-матеріальних цінностей, яка відповідає наступним вимогам.

Мета розробки: Створення єдиного цифрового простору для автоматизації обліку, оптимізації зберігання та забезпечення точного контролю за рухом товарно-матеріальних цінностей у режимі реального часу. Система повинна прискорити процеси приймання, відвантаження, інвентаризації та переміщення товарів, мінімізувати вплив людського фактора, забезпечити контроль термінів придатності, формувати миттєві звіти про рух товарів та легко інтегруватися з зовнішніми сервісами.

Таблиця 1.3 - Функціональні вимоги

№	Назва вимоги	Пріоритет	Опис
FR-01	Створення карток товарів	High	Формування цифрового паспорта товару: назва, артикул, категорія, одиниці виміру
FR-02	Операційний облік (прихід)	High	Фіксація надходження товарів на склад з оновленням залишків та збереженням незмінної транзакції
FR-03	Операційний облік (списання)	High	Фіксація списання товарів з перевіркою достатності залишків та захистом від від'ємного сальдо
FR-04	Операційний облік (переміщення)	High	Переміщення товарів між складами без зміни загальної кількості
FR-05	Моніторинг залишків	High	Відображення актуальних залишків у реальному часі для всіх складів
FR-06	Пошук та фільтрація	High	Швидкий пошук за назвою, артикулом, категорією; фільтрація за параметрами
FR-07	Управління ролями та правами	High	RBAC-система з ролями: Адміністратор, Менеджер, Комірник, Аналітик
FR-08	Робота з даними	High	Імпорт/експорт даних, резервне копіювання, відновлення
FR-09	Звіти та аналітика	Medium	Генерація звітів з можливістю експорту у форматах Excel та PDF
FR-10	Управління налаштуваннями	Medium	Глобальні параметри системи: склади, валюта, довідники
FR-11	Адресне зберігання	Low	Прив'язка товару до місця
FR-12	Система сповіщень	Low	Автоматичні нотифікації про критичні залишки та терміни придатності

Нефункціональні вимоги:

- Продуктивність: час відповіді системи на запит користувача не повинен перевищувати 2 секунди при стандартному навантаженні.
- Надійність: доступність системи не менше 99,5% на місяць; механізми резервного копіювання даних щонайменше раз на добу.
- Безпека: шифрування даних при передачі (TLS 1.3), безпечне зберігання паролів, журналювання всіх дій користувачів.
- Масштабованість: горизонтальне масштабування серверної частини без зупинки роботи системи.
- Юзабіліті: інтерфейс має бути засвоєний новим користувачем без спеціального навчання за не більш ніж 2 години роботи.

1.5 Висновок до розділу 1

У першому розділі проведено комплексний аналіз предметної області інформаційних систем обліку товарно-матеріальних цінностей, що дозволило сформулювати чітке розуміння проблематики та визначити вектор розробки проєкту.

В рамках аналізу предметної області охарактеризовано поняття товарно-матеріальних цінностей, досліджено основні бізнес-процеси складського управління та визначено ключові учасники системи. Розроблено організаційну структуру, що включає п'ять рівнів взаємодії: стратегічний, тактичний, операційний, аналітичний та рівень зовнішніх інтеграцій.

Проведено порівняльний аналіз чотирьох провідних конкурентних рішень: Zoho Inventory, Sortly, Dilovod та Odoo. Аналіз виявив ринкову нішу — відсутність доступної, україномовної системи з сучасним UX, повним функціоналом та відкритим API за прийнятну вартість.

Досліджено недоліки існуючих систем, згруповані у три категорії: технологічні, організаційні та економічні. Для кожного виявленого недоліку запропоновано конкретний шлях його подолання в рамках розроблюваної системи.

Сформульовано концепцію системи на основі п'яти ключових принципів: єдиної точки правди, незмінності транзакцій, захисту від людського фактора,

масштабованості та відкритої інтеграції. Розроблено концептуальну модель, що описує шість основних сутностей системи та зв'язки між ними. Проведено SWOT-аналіз, який підтвердив конкурентоспроможність запропонованого рішення.

На підставі всього вищезазначеного сформульовано постановку задачі: розроблено реєстр із дванадцяти функціональних вимог з пріоритетами та визначено п'ять нефункціональних вимог щодо продуктивності, надійності, безпеки, масштабованості та юзабіліті системи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ СИСТЕМИ ОБЛІКУ ТОВАРНО-МАТЕРІАЛЬНИХ ЦІННОСТЕЙ

2.1 Загально системні рішення

Сформувавши задачі до інформаційної системи обліку товарно-матеріальних цінностей, можна перейти до розробки діаграми варіантів використання. Вона показує, які ролі існують у системі та яким чином вони взаємодіють з нею. Така діаграма допомагає визначити функціональні вимоги до системи - тобто те, що вона може виконати з точки зору користувача, без деталізації внутрішніх процесів або логіки роботи.

Діаграма прицедентів системи обліку ТМЦ включає два основних актори: Адміністратора та Користувача. Розмежування ролей реалізовано відповідно до вимог FR-07 RBAC-системи, визначених у попередньому розділі. Кожна роль має чітко окреслений набір доступних функцій системи (рис. 2.1).

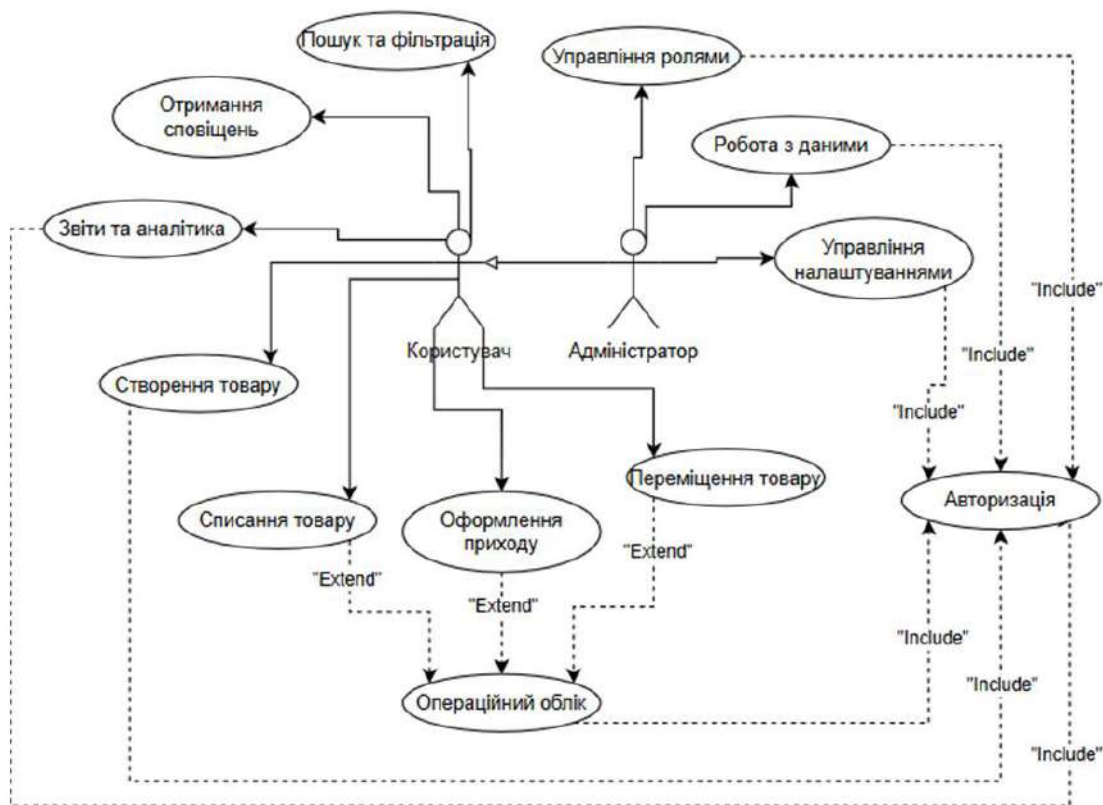


Рисунок 2.1 - Діаграма прецедентів.

Адміністратор є найбільш привілейованою роллю і має доступ до всіх функцій системи. Він керує обліковими записами користувачів, налаштовує глобальні параметри системи, управляє довідниками, формує звіти та має доступ до повного журналу дій усіх користувачів. Адміністратор також виконує резервне копіювання даних та відновлення системи.

Користувач відповідає за операційне управління товарними запасами. Він може створювати та редагувати картки товарів, здійснювати всі типи операцій, формувати звіти та аналітику, а також переглядати поточні залишки в режимі реального часу. Користувач може налаштовувати параметри автоматичних сповіщень про критичні залишки.

На основі діаграми прецедентів розробляються діаграми послідовності, що моделюють логіку окремих сценаріїв. Вони показують обмін повідомленнями між об'єктами системи під час виконання конкретного сценарію (рис 2.2 - 2.3).

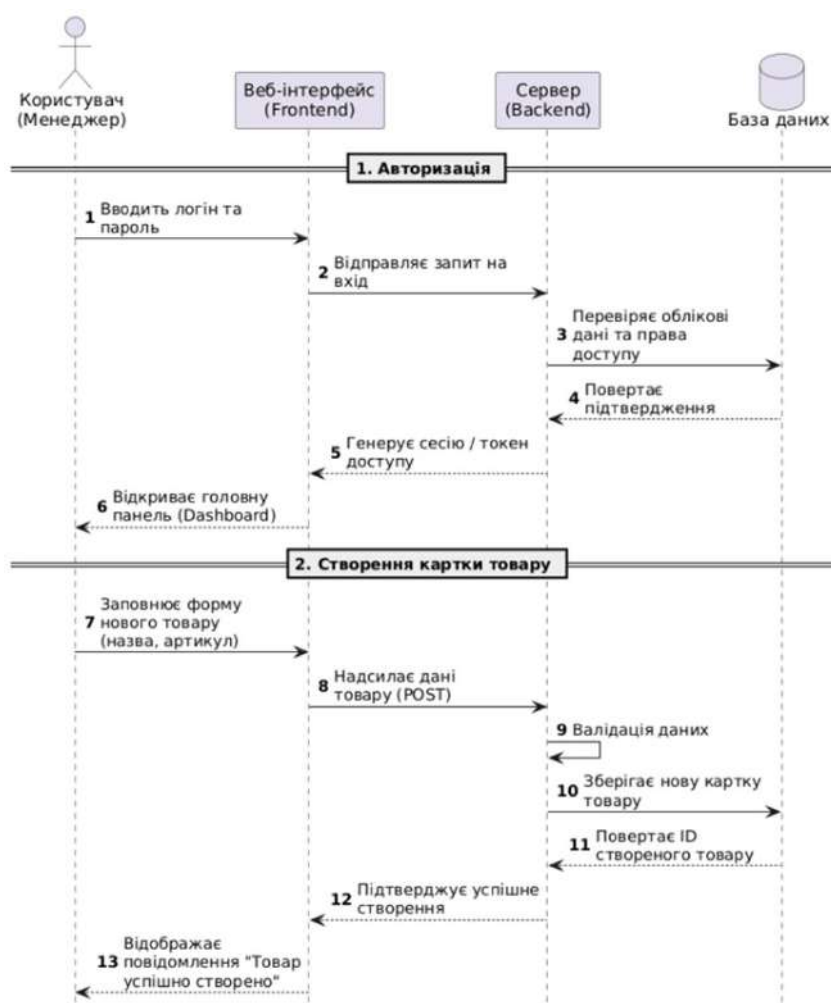


Рисунок 2.2 - Діаграма послідовності (1-2 Етап).

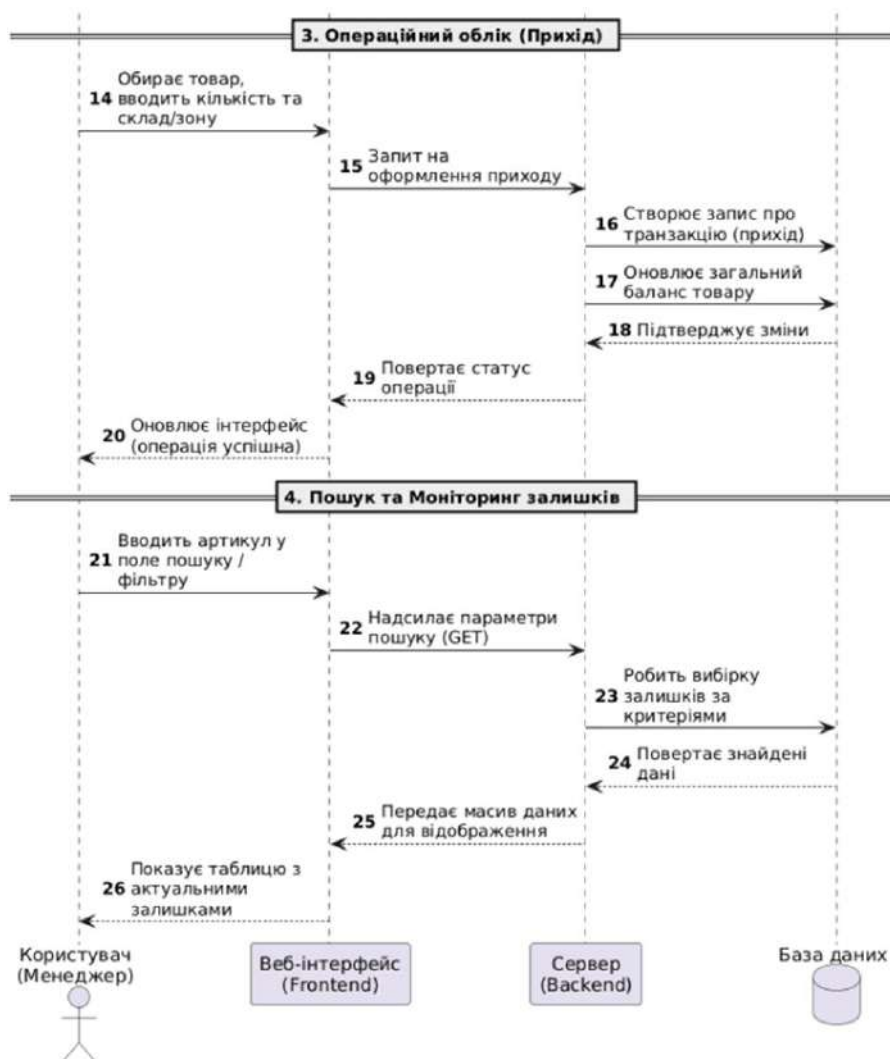


Рисунок 2.3 - Діаграма послідовності (3-4 Етап).

У сценарії «Етап 1: Авторизація» Користувач ініціює операцію через інтерфейс системи вводячи логін та пароль. Веб-інтерфейс відправляє запит на вхід до сервера. Сервер перевіряє коректність введених даних: валідує логін та пароль користувача. При успішній валідації база даних повертає підтвердження права входу серверу, він, в свою чергу, створює токен доступу і передає його до веб-інтерфейсу на якому відривається головна панель.

У сценарії «Етап 2: Створення картки товару» користувач заповнює форму додавання нового товару: назва товару, кількість, склад, постачальник та контакти постачальника. Веб-інтерфейс надсилає дані на сервер де вони проходять валідацію. Якщо перевірка пройшла успішно то в базі даних з'являється новий запис а користувачу виводиться повідомлення про успішне створення нової картки товару.

У сценарії «Етап 3: Операційний облік (прихід)» Користувач ініціює операцію через інтерфейс системи, обираючи товар, його кількість та склад. Система перевіряє коректність введених даних: валідує артикул товару, перевіряє існування складу та коректність кількісних показників. При успішній валідації система створює незмінну транзакцію типу «прихід», оновлює залишки у таблиці та повертає підтвердження операції.

У сценарії «Етап 4: Пошук та моніторинг залишків» Користувач вводить артикул в поле пошуку товару. Веб-інтерфейс надсилає параметри пошуку на сервер, де виконується вибірка за критеріями запиту. Після знаходження дані повертаються до користувача у вигляді таблиці.

Аналіз сценаріїв взаємодії дозволяє перейти до моделювання даних - визначення структури та організації інформації, що зберігається у базі даних системи.

2.2 Рішення по інформаційному забезпеченню

Інформаційне забезпечення системи обліку ТМЦ охоплює сукупність даних, методів їх організації та технологій зберігання, що забезпечують повноцінне функціонування всіх модулів системи. Центральним елементом інформаційного забезпечення є база даних, яка зберігає відомості про товари, склади, транзакції, користувачів та поточні залишки.

Проектування бази даних виконується у три послідовні етапи: розробка концептуальної моделі та логічної моделі. Такий підхід забезпечує послідовне уточнення структури та запобігає виникненню архітектурних помилок на пізніх етапах розробки.

2.2.1 Розробка концептуальної моделі даних

Концептуальна модель даних описує предметну область на найвищому рівні абстракції: визначає ключові сутності системи, їхні атрибути та характер зв'язків між ними, не прив'язуючись до конкретної технології зберігання. На цьому етапі важливо правильно ідентифікувати всі об'єкти предметної області та форми їх взаємодії.

Концептуальна модель даних системи обліку ТМЦ (рис. 2.4) охоплює шість основних сутностей: Товар, Категорія, Одиниця виміру, Склад, Користувач та Транзакція. Кожна сутність відображає окремий аспект предметної області складського обліку.

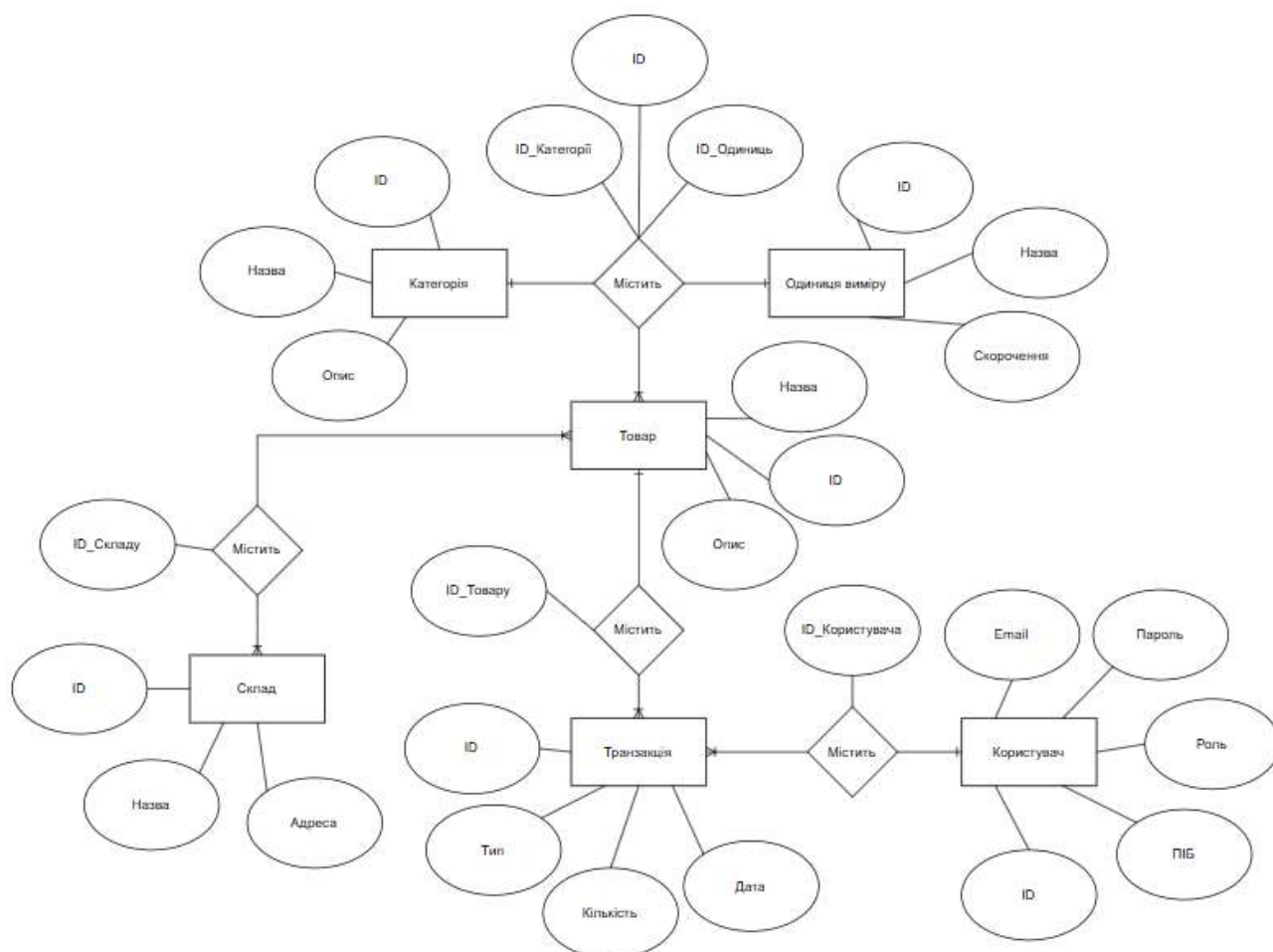


Рисунок 2.4 - Концептуальна модель даних системи обліку ТМЦ.

Сутність Товар є центральною в системі і містить повний цифровий паспорт кожної товарної позиції. Атрибути сутності включають унікальний ідентифікатор (ID), назву товару, текстовий опис, ID пов'язаних атрибутів. Товар пов'язаний з сутностями Категорія, Склад та Одиниця виміру через відношення «багато до одного» або «багато до багатьох».

Сутність Категорія забезпечує ієрархічну класифікацію товарного асортименту. Вона містить ідентифікатор, назву категорії та опис. Одна категорія може включати довільну кількість товарів, тоді як кожен товар належить рівно одній

категорії. Категоризація товарів необхідна для зручної навігації, фільтрації та побудови аналітичних звітів.

Сутність Одиниця виміру зберігає довідникову інформацію про одиниці вимірювання кількості товарів: штуки, кілограми, літри, метри тощо. Атрибути: ідентифікатор, повна назва та скорочення. Одна одиниця виміру може використовуватися для багатьох товарів, забезпечуючи консистентність облікових записів.

Сутність Склад описує місця фізичного зберігання товарів. Кожен склад має унікальний ідентифікатор, назву та адресу. Система підтримує необмежену кількість складів, що відповідає вимозі мультискладовості.

Сутність Користувач представляє облікові записи всіх суб'єктів, що мають доступ до системи. Атрибути включають ідентифікатор, прізвище, ім'я та по батькові, email-адресу (унікальний логін), пароль та роль у системі. Роль визначає набір дозволених дій відповідно до RBAC-моделі.

Сутність Транзакція фіксує кожну операцію з товарами і є незмінним записом, що відповідає принципу цілісності даних. Транзакція включає тип операції (прихід, списання або переміщення), посилання на товар та відповідального користувача, кількість та дату проведення операції. Завдяки незмінності транзакцій система забезпечує повний аудиторський слід усіх змін у запасах.

Зв'язки між сутностями відображають бізнес-правила предметної області. Зв'язок між Категорією та Товаром - один до багатьох: одна категорія об'єднує безліч товарів, але кожен товар належить лише одній категорії. Аналогічний зв'язок між Одиницею виміру та Товаром. Зв'язок між Товаром та Складом та - багато до багатьох: багато товарів може бути пов'язаний з безліччю складів. Зв'язок між Користувачем та Транзакцією - один до багатьох, оскільки один користувач може провести безліч операцій, але кожна транзакція ініційована одним конкретним користувачем.

2.2.2 Розробка логічної моделі даних

Логічна модель даних деталізує концептуальну модель, додаючи конкретні атрибути з типами даних, визначаючи первинні та зовнішні ключі, уточнюючи

кардинальності зв'язків та встановлюючи обмеження цілісності. Логічна модель є технологічно нейтральною, однак вже достатньо деталізованою для подальшого відображення на конкретну СУБД.

Логічна модель даних системи обліку ТМЦ (рис. 2.5) включає шість таблиць: Category, Unit_measurement, Warehouse, Product, User та Transaction.

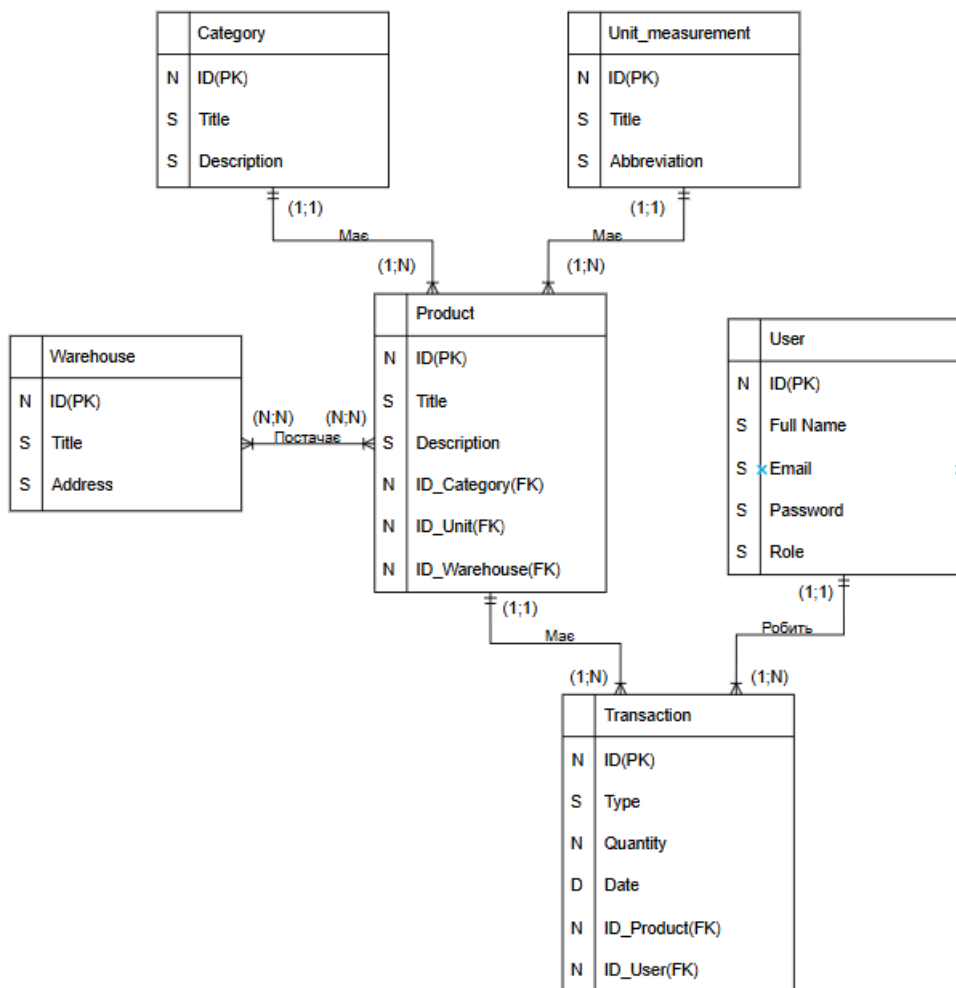


Рисунок 2.5 - Логічна модель даних системи обліку ТМЦ.

Таблиця Category зберігає довідникові дані про категорії товарів. Первинний ключ ID унікально ідентифікує кожну категорію. Поле Title зберігає найменування категорії, яке має бути унікальним у межах системи. Поле Description містить необов'язковий детальний опис категорії. Таблиця є «батьківською» щодо Product через зовнішній ключ ID_Category.

Таблиця Unit_measurement містить довідник одиниць виміру. Первинний ключ ID унікально ідентифікує запис. Поля Title та Abbreviation зберігають повну назву та

аббревіатуру одиниці вимірювання відповідно. Обидва поля мають обмеження, що унеможлиблює дублювання одиниць виміру в довіднику.

Таблиця Product є центральною в моделі. Первинний ключ ID ідентифікує товар. Поле Title зберігає повну назву товару. Поле Description призначене для розгорнутого опису. Зовнішні ключі ID_Category, ID_Unit та ID_Warehouse посиляються на відповідні таблиці-довідники.

Таблиця Warehouse описує місця зберігання. Крім первинного ключа ID, містить поля Title для унікального найменування складу та Address для фізичної адреси. Таблиця є «батьківською» щодо Product.

Таблиця User зберігає облікові записи користувачів системи. Поле email є природним ідентифікатором для автентифікації. Поле Password зберігає хешований пароль - жоден відкритий пароль не зберігається в базі даних. Поле Role визначає рівень доступу. Поле Full_Name фіксує повний ПІБ користувача.

Таблиця Transaction реалізує принцип незмінного журналу операцій. Первинний ключ ID забезпечує унікальність кожного запису. Поле Type визначає тип операції. Поле Quantity зберігає кількість товару, а CHECK-обмеження унеможлиблює запис нульових або від'ємних значень. Поле Date автоматично фіксується системою в момент виконання операції. Зовнішні ключі ID_Product, та ID_User посиляються на відповідні таблиці-довідники.

Обмеження цілісності відіграють ключову роль у коректності даних. Зовнішні ключі між таблицями налаштовані з неможливістю видалити категорію, якщо до неї прив'язані товари, або склад, якщо по ньому є транзакції. Це запобігає «осиротілим» посиланням та забезпечує референційну цілісність даних.

2.3 Висновок до розділу 2

У другому розділі виконано комплексне проектування web-застосунку системи обліку товарно-матеріальних цінностей. Розроблені діаграми та моделі формують технічну основу для подальшої розробки програмного продукту.

В рамках загальносистемних рішень визначено дві ролі користувачів системи: Адміністратор та Користувач. Для кожної ролі описано функціональні можливості

відповідно до RBAC-моделі. Розроблено сценарії взаємодії для ключових операцій: прийому товару, списання та генерації звітів. Визначено логіку перевірки достатності залишків та механізм блокування некоректних операцій.

У частині інформаційного забезпечення розроблено два рівні моделей даних. Концептуальна модель визначила шість ключових сутностей системи: Товар, Категорія, Одиниця виміру, Склад, Користувач та Транзакція. Встановлено зв'язки між ними з визначенням кардинальностей.

Логічна модель даних деталізувала структуру шести таблиць бази даних з визначенням типів даних, обмежень цілісності, первинних та зовнішніх ключів. Особливу увагу приділено таблиці Transaction як незмінному журналу операцій. Встановлено обмеження на поле Quantity для захисту від нульових та від'ємних значень.

Результати проєктування є повною та несуперечливою технічною специфікацією, що забезпечує підґрунтя для реалізації програмного продукту в подальшому.

РОЗДІЛ 3. РОЗРОБКА WEB-ЗАСТОСУНКУ СИСТЕМИ ОБЛІКУ ТОВАРНО-МАТЕРІАЛЬНИХ ЦІННОСТЕЙ

3.1 Діаграма розгортання

На основі діаграм варіантів використання, послідовності та моделей даних, побудованих у попередньому розділі, розроблено діаграму розгортання системи обліку товарно-матеріальних цінностей. Діаграма розгортання - це структурна UML-діаграма, що відображає фізичне розміщення програмних артефактів на вузлах апаратного забезпечення та демонструє комунікаційні зв'язки між цими вузлами.

Діаграма розгортання системи RDDH включає три основні фізичні вузли, між якими налагоджено інформаційний обмін: робоче місце адміністратора складу, робоче місце робітника та сервер додатка з базою даних. Кожен вузол взаємодіє з центральним сервером через мережу Інтернет або локальну мережу підприємства за допомогою протоколу HTTP/HTTPS.

Адміністратор підключається до системи через свій пристрій (персональний комп'ютер або планшет), на якому встановлено веб-браузер. Через веб-інтерфейс він виконує операції стратегічного та тактичного рівня: управління довідниками, перегляд аналітики та звітів, налаштування системи, управління правами доступу користувачів.

Співробітник також підключається до системи через браузер з окремого робочого пристрою. Його інтерфейс оптимізований для швидкого виконання оперативних складських операцій: реєстрації приходу товарів, їх списання та внутрішнього переміщення між складами.

Центральний сервер є основним обчислювальним вузлом системи. На ньому розгорнуто Apache/Nginx вебсервер, PHP-інтерпретатор для обробки серверних скриптів та СУБД MySQL, у якій зберігається вся актуальна інформація. Веб-сервер приймає HTTP-запити від клієнтів, передає їх PHP-скрипту, який зчитує або записує дані до бази даних та повертає відповідь у форматі JSON.

Обмін даними між клієнтськими пристроями та сервером реалізовано за принципом REST: JavaScript-код на стороні браузера формує асинхронні AJAX-

запити методами GET та POST. Завдяки цьому інтерфейс реагує миттєво, без повного перезавантаження сторінки, що забезпечує плавний та зручний користувацький досвід.

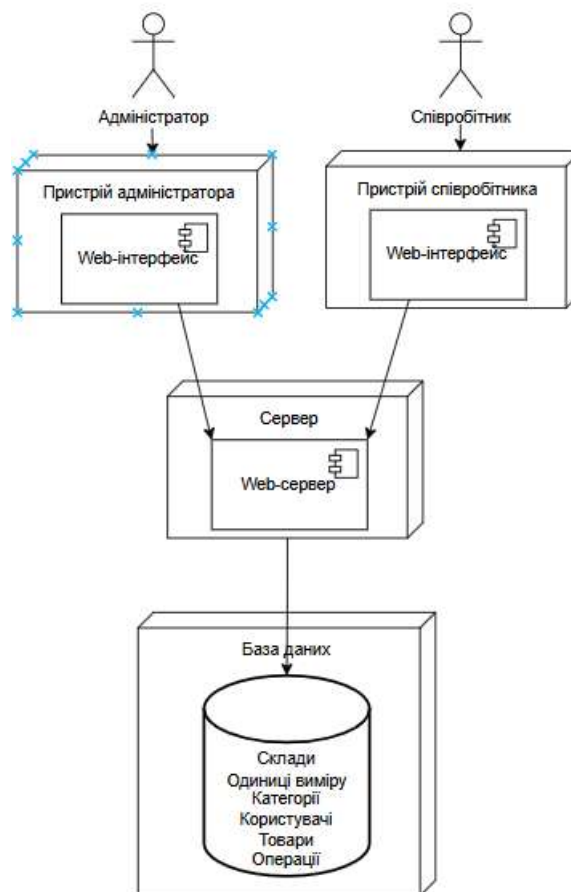


Рисунок 3.1 - Діаграма розгортання системи.

Діаграма розгортання (рис. 3.1) наочно показує, як саме компоненти системи пов'язані між собою та яким чином забезпечується обмін даними у реальному часі. Веб-сервер відіграє роль шлюзу між клієнтами та базою даних, обробляючи всі вхідні запити. База даних є основним сховищем, яке гарантує цілісність та актуальність інформації для всіх одночасно підключених користувачів.

Дана архітектура дозволяє правильно розподілити та налаштувати технічні ресурси для стабільної роботи системи. Вона чітко визначає, які саме пристрої або сервери задіяні у процесі, за які елементи вони відповідають, а також як вони взаємодіють між собою.

3.2 Рішення з технічного забезпечення

На підприємстві для нормальної роботи інформаційної системи рекомендується мати щонайменше два типи пристроїв, кожен з яких виконує свою конкретну функцію. Такий поділ дозволяє розподілити навантаження між пристроями та уникнути збоїв у роботі системи.

Перший пристрій - робоче місце адміністратора. На ньому зосереджено управлінські функції: адміністрування системи, перегляд аналітики, формування звітів, налаштування довідників та управління правами доступу. Другий пристрій - робоче місце співробітника. Він забезпечує швидкий доступ до системи, дозволяє бачити потрібну інформацію про залишки в режимі реального часу та ефективно обслуговувати складські операції.

Мінімальні технічні вимоги до клієнтських пристроїв:

- оперативна пам'ять: не менше 4 ГБ, рекомендується 8 ГБ;
- процесор: двох-ядерний з тактовою частотою не менше 1.6 ГГц;
- накопичувач: SSD або HDD обсягом від 128 ГБ;
- операційна система: Windows 10/11, macOS або будь-який сучасний дистрибутив Linux;
- браузер: Google Chrome, Mozilla Firefox або Microsoft Edge;
- мережа: стабільне з'єднання з Інтернетом або локальною мережею підприємства зі швидкістю не менше 5 Мбіт/с.

Важливо забезпечити стабільне підключення всіх пристроїв до мережі. Без нормального з'єднання веб-застосунк не зможе обмінюватися даними з сервером, що унеможливить виконання будь-яких складських операцій у реальному часі.

Для розгортання серверної частини системи необхідний веб-сервер або хостинг-середовище зі наступними характеристиками:

- оперативна пам'ять: не менше 2 ГБ;
- дисковий простір: не менше 10 ГБ для розміщення файлів системи та бази даних;
- веб-сервер: Apache 2.4+ або Nginx 1.18+;

- PHP: версія 8.0 або новіша з підтримкою розширення PDO (PHP Data Objects);
- СУБД MySQL: версія 8.0 або MariaDB версія 10.5+;
- операційна система сервера: Ubuntu Server 20.04 LTS або CentOS 8+.

Наявність SSD-накопичувача на сервері суттєво підвищує швидкість читання та запису даних, що позитивно впливає на час відповіді системи при виконанні запитів до бази даних.

3.3 Рішення з програмного забезпечення

Для забезпечення стабільної та зручної роботи системи обліку товарно-матеріальних цінностей розроблено програмне рішення, що відповідає сучасним вимогам та органічно інтегрується з визначеним технічним середовищем. Програмне забезпечення системи RDDH реалізовано як веб-застосунок за класичною клієнт-серверною архітектурою.

Клієнтська частина системи побудована на трьох базових технологіях веброзробки: HTML5, CSS3 та JavaScript. Завдяки цим технологіям інтерфейс вибудовувано зручним та інтуїтивно зрозумілим для будь-якого користувача - від адміністратора до рядового комірника.

Серверна частина реалізована на мові програмування PHP 8.0+. Ключовим скриптом є DB.php, який виступає єдиною точкою обміну даними між браузером та базою даних. Скрипт приймає GET-запити для завантаження поточних залишків та журналу операцій, а також POST-запити для синхронізації нових або оновлених даних. Для безпечного підключення до бази даних використовується механізм PDO (PHP Data Objects) з підготовленими запитами, що унеможливорює SQL-ін'єкції.

Для зберігання всієї важливої інформації в системі використовується база даних MySQL, яка є надійною та широко розповсюдженою системою управління реляційними базами даних. MySQL надає зручні інструменти для швидкого пошуку, оновлення та збереження інформації, що робить її важливою частиною веброботи системи. Завдяки цим можливостям система може ефективно обробляти великий обсяг даних та забезпечувати стабільну роботу застосунку.

3.3.1 Розробка бази даних системи

База даних системи розроблена в СУБД MySQL та містить дві основні таблиці: inventory та operations. Таблиця inventory зберігає актуальний реєстр товарів на складі. Таблиця operations є незмінним журналом усіх складських транзакцій.

Ініціалізація бази даних відбувається автоматично при першому зверненні до скрипту DB.php. Якщо база або таблиці ще не існують, вони створюються автоматично. Таблиця inventory призначена для зберігання поточного стану складських залишків. Кожен запис представляє унікальну позицію товару та містить усі необхідні атрибути для ідентифікації та обліку.

Таблиця operations є незмінним журналом усіх виконаних складських транзакцій. Кожна операція фіксується окремим рядком та ніколи не видаляється і не редагується самостійно, що гарантує цілісність аудиторського сліду до моменту генерації звіту.

3.3.2 Серверна логіка

Центральним елементом серверної частини є PHP-скрипт DB.php, який реалізує RESTful-інтерфейс для обміну даними між браузером і базою даних. Скрипт підтримує два HTTP-методи: GET для читання поточного стану системи та POST для збереження оновлених даних.

Підключення до бази даних реалізовано через PDO (PHP Data Objects) - захищений механізм взаємодії з реляційними базами даних. Параметри підключення задаються через змінні конфігурації. Використання PDO::ERRMODE_EXCEPTION налаштовує обробник помилок таким чином, що будь-яка помилка SQL-запиту автоматично генерує виняток, який може бути оброблено на рівні застосунку. Обробник GET-запиту витягує з бази даних усі поточні залишки та журнал операцій, повертаючи їх у форматі JSON.

Запити до БД використовують метод \$pdo->query() для операцій SELECT, що повертають набір даних. Метод fetchAll(PDO::FETCH_ASSOC) перетворює результат у масив асоціативних масивів PHP, а json_encode() серіалізує їх у JSON-рядок для передачі клієнту. Обробник POST-запиту реалізує атомарний механізм синхронізації:

він повністю очищує таблиці та записує актуальний стан даних, що надійшов від клієнта.

Використання підготовлених запитів є критично важливим елементом захисту від SQL-ін'єкцій. Замість прямої вставки даних у SQL-рядок, в шаблон запиту підставляються знаки питання, а фактичні значення передаються окремо через метод `execute()`. СУБД самостійно виконує екранування та перевірку типів, що унеможливорює зловмисне маніпулювання запитами.

3.3.3 Клієнтська логіка

Клієнтська частина системи реалізована у вигляді єдиного JavaScript-файлу `script.js`, який підключається до всіх HTML-сторінок застосунку. Цей файл керує відображенням даних, обробкою форм та синхронізацією стану між браузером та сервером. Архітектура клієнтської логіки базується на принципі єдиного сховища стану.

Завантаження актуальних даних відбувається при відкритті будь-якої сторінки застосунку. Функція `loadFromServer()` виконує асинхронний GET-запит до `DB.php` та оновлює локальний стан. Функція `saveToServer()` виконує синхронізацію поточного стану застосунку з базою даних після кожної операції. Вона формує POST-запит з актуальним вмістом масивів `inventory` та `operations`.

Обробка форм складських операцій реалізована через єдину функцію `handleOperationForm()`, яка визначає тип операції (прихід, списання або переміщення) за значенням прихованого поля `opType` та виконує відповідну бізнес-логіку. Такий підхід дозволив уникнути дублювання коду та забезпечив єдину точку входу для всіх операцій.

При реєстрації приходу система знаходить у масиві `inventory` відповідний товар та збільшує його кількість. Якщо товар з такою назвою не знайдено, операція блокується із відображенням повідомлення про помилку. Аналогічна перевірка виконується при списанні: система перевіряє наявність достатньої кількості товару та забороняє операцію, якщо залишок стане від'ємним. Це ключовий механізм захисту цілісності облікових даних.

При внутрішньому переміщенні система виконує дві пов'язані дії: зменшує кількість товару на складі-джерелі та одночасно збільшує кількість на складі-призначенні. Загальна кількість товару при цьому залишається незмінною, що відповідає принципу збереження матеріальних цінностей.

3.3.4 Реалізація інтерфейсу веб-застосунку

Програмне забезпечення складається з HTML-сторінок, кожна з яких відповідає за конкретну функціональну задачу. Усі сторінки побудовані за єдиним принципом: статичний HTML-каркас завантажується браузером, після чого JavaScript динамічно заповнює сторінку актуальними даними з сервера.

Головна сторінка є відправною точкою системи. Вона виконує функцію дашборду: відображає загальну кількість товарів на складі, кількість позицій з критично низьким залишком та таблицю останніх операцій. Блок швидких дій дозволяє перейти до будь-якої операції одним кліком. Всі дані на головній сторінці завантажуються динамічно при відкритті та автоматично оновлюються після кожної нової операції.

Сторінка `creation.html` призначена для реєстрації нових товарів у системі. Форма містить поля: назва товару, початкова кількість, склад зберігання, категорія та одиниця виміру. При збереженні генерується унікальний UUID-ідентифікатор, і новий запис додається до масиву `inventory` та синхронізується з базою даних через POST-запит до `DB.php`.

Сторінка `incoming.html` містить форму з двома обов'язковими полями: назва товару та кількість. Поле коментаря є необов'язковим. При підтвердженні операції система збільшує кількість обраного товару та додає запис про операцію в журнал.

Сторінка `writeoff.html` структурно схожа на сторінку приходу, але має кілька ключових відмінностей: поле причини списання є обов'язковим для заповнення, а перед виконанням операції система перевіряє наявність достатнього залишку. Якщо кількість до списання перевищує наявний залишок, операція блокується й користувач отримує повідомлення про помилку, що є реалізацією принципу захисту від від'ємних залишків.

Сторінка `transfer.html` є найбільш функціонально насиченою формою. Після вибору товару поле «Зі складу» заповнюється автоматично - система визначає поточне місце зберігання товару. Поле «На склад» має список автодоповнення, що формується з унікальних назв усіх складів системи. Це унеможливорює помилки при введенні назви складу та значно прискорює роботу оператора. При збереженні операції переміщення в таблиці `inventory` оновлюється поле `warehouse` у відповідного товару.

3.3.5 Модуль синхронізації та збереження даних

Архітектура системи реалізує дворівневу модель зберігання даних. На першому рівні дані зберігаються в оперативній пам'яті браузера у вигляді JavaScript-масивів, що забезпечує миттєву реакцію інтерфейсу на дії користувача без мережових затримок. Після кожної операції автоматично викликається функція `saveToServer()`, яка синхронізує актуальний стан із постійним сховищем - базою даних MySQL на сервері.

На другому рівні дані надійно зберігаються у реляційній базі даних MySQL. При кожному відкритті будь-якої сторінки системи виконується запит `loadFromServer()`, який завантажує актуальний стан із сервера. Це гарантує, що всі підключені до системи користувачі завжди бачать однакові та актуальні дані, незалежно від того, з якого пристрою вони увійшли.

3.4 Висновок до розділу 3

У третьому розділі здійснено повний цикл розробки вебзастосунку системи обліку товарно-матеріальних цінностей RDDH: від фізичної архітектури до конкретних програмних рішень.

Розроблено діаграму розгортання, яка демонструє, як саме різні частини інформаційної системи взаємодіють одна з одною. Визначено три фізичні вузли: робоче місце адміністратора/менеджера, робоче місце комірника та центральний сервер з базою даних MySQL. Всі вузли взаємодіють через протокол HTTP/HTTPS за схемою клієнт-сервер.

Визначено рішення з технічного забезпечення. Для клієнтських пристроїв встановлено мінімальні вимоги: оперативна пам'ять від 4 ГБ, сучасний веб-броузер та стабільне підключення до мережі. Для серверної частини визначено вимоги до веб-сервера (Apache/Nginx), PHP-інтерпретатора (версія 8.0+) та СУБД MySQL (версія 8.0+).

Прийняті рішення з технічного та програмного забезпечення відповідають вимогам щодо стабільності, безпеки та зручності використання веб-застосунку. Це важливо для безперебійної роботи системи, адже ці рішення забезпечують надійний доступ до системи як для адміністраторів, так і для операторів складу.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Загальні положення охорони праці

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів і засобів, спрямованих на збереження здоров'я та працездатності людини у процесі трудової діяльності. Дотримання вимог охорони праці є обов'язковим для будь-якого підприємства, яке використовує інформаційні системи в своїй роботі.

У сучасних умовах цифровізації бізнесу більшість співробітників підприємств проводять значну частину робочого часу за комп'ютером. Це стосується й операторів системи обліку товарно-матеріальних цінностей RDDH, які для виконання своїх посадових обов'язків постійно взаємодіють з веб-інтерфейсом системи. Тривала робота з екранними пристроями формує специфічне виробниче середовище з притаманними йому факторами ризику, тому питання безпеки праці набувають особливого значення.

Відповідно до законодавства України, охорона праці сьогодні стосується практично кожної людини, яка працює в офісі, торговому чи промисловому підприємстві. Дотримання вимог охорони праці є необхідним для нормальної роботи будь-якої організації. Без дотримання цих вимог підприємство не може працювати належним чином. При порушенні правил охорони праці власника або іншу відповідальну особу можуть притягнути до адміністративної чи кримінальної відповідальності.

Роботодавець зобов'язаний створити на кожному робочому місці умови праці відповідно до нормативно-правових актів та забезпечити дотримання вимог законодавства щодо прав працівників у галузі охорони праці. З цією метою роботодавець повинен вживати відповідних заходів щодо вдосконалення існуючих і впровадження нових технологій та організації виробництва з урахуванням досягнень науки.

У закладі, де працівники використовують комп'ютери для роботи з системою обліку ТМЦ, роботодавець зобов'язаний надати під підпис інформацію про умови

праці та наявність на робочому місці небезпечних чи шкідливих факторів. Це включає відомості про ті ризики, які ще не було усунуто та які виникають під час роботи з екранними пристроями. Крім того, роботодавець повинен надати інформацію про можливі наслідки впливу цих факторів на здоров'я працівників. Це допоможе співробітнику усвідомлено ставитися до своєї безпеки та дотримуватися відповідних правил безпеки.

Перед початком роботи з комп'ютером роботодавець повинен організувати навчання або перевірити знання працівника з питань охорони праці та безпечного використання екранних пристроїв. Також це потрібно робити, коли обладнання замінюють на нове чи змінюється організація робочого процесу.

4.2 Вимоги безпеки при роботі з комп'ютерним обладнанням

Організація безпечного робочого місця оператора є ключовим елементом охорони праці. Робоче місце, де працівники використовують екранні пристрої для роботи з системою обліку товарно-матеріальних цінностей, має бути зручним, безпечним та не шкодити здоров'ю. Правильно організоване робоче середовище сприяє підвищенню продуктивності праці та мінімізує ризик виникнення професійних захворювань.

Стіл повинен бути досить великим для зручного розміщення монітора, клавіатури, документів або іншого необхідного обладнання. Розміщення монітора на правильній відстані та висоті від очей є критично важливим для запобігання перевтомі зорового апарату. Рекомендована відстань від очей до екрана має становити не менше 50-70 см залежно від розміру монітора. Верхній край екрана повинен знаходитися на рівні очей або трохи нижче.

Крісло має бути міцним, зручним та мати можливість регулювання висоти та куту нахилу спинки. Правильно підібрана висота сидіння забезпечує положення, при якому ступні повністю спираються на підлогу або підставку, а кут між стегнами і тулубом становить не менше 90°. Працівники повинні мати змогу легко рухатися та без труднощів змінювати положення тіла під час роботи. Це сприяє зменшенню

навантаження на опорно-руховий апарат та покращує комфорт під час тривалого перебування на одному місці.

Екран монітора повинен відображати чітке зображення без мерехтіння та відблисків. Яскравість та контрастність повинні легко регулюватися для зниження втоми очей. Монітор повинен мати змогу повертатися та нахилитися для зручного налаштування відповідно до потреб користувача. Рекомендується використання антивідблискового покриття екрана та додаткових фільтрів, що знижують вплив синього спектру випромінювання.

Клавіатура повинна бути окремо від монітора, зручною та з чіткими позначеннями на клавішах. Поверхня столу та клавіатури має бути матовою, щоб не відбивати світло. Це сприяє зменшенню зорового навантаження та підвищує комфорт під час роботи за комп'ютером. При виконанні операцій оператор виконує значну кількість введень даних, тому зручна клавіатура є особливо важливою.

У приміщенні, де здійснюється робота з системою обліку ТМЦ, має бути комфортна температура, свіже повітря та низький рівень шуму. Оптимальна температура повітря в робочому приміщенні має підтримуватися на рівні 20-22°C у холодну пору року та 22-24°C у теплу пору року. Відносна вологість повітря повинна становити 45-65%. Рівень шуму на робочих місцях не повинен перевищувати 50 дБА. Освітлення повинно бути достатнім та рівномірним, природне або штучне, але не повинно сліпити або створювати тіні на екрані.

Перед початком роботи працівники повинні протирати екран та робоче місце від пилу, а після роботи вимикати обладнання відповідно до встановленого порядку. Слід уникати розміщення склянок з рідиною, їжі та сторонніх предметів поблизу клавіатури та системного блоку, оскільки потрапляння рідини або забруднень може призвести до пошкодження обладнання та виникнення аварійних ситуацій.

4.3 Заходи щодо зменшення впливу шкідливих виробничих факторів

Основним шкідливим фактором при роботі з комп'ютером є зорове навантаження. Тривала концентрація зору на екрані під час виконання операцій в системі призводить до втоми очей та може спричинити розвиток комп'ютерного

зорового синдрому. Для запобігання цьому рекомендується застосовувати правило «20-20-20»: кожні 20 хвилин роботи робити паузу на 20 секунд, дивлячись на предмет, розташований на відстані не менше 6 метрів. Також доцільно проводити спеціальну гімнастику для очей, що включає вправи для зняття м'язової напруги.

Статичне навантаження на м'язи шиї, плечей та спини є ще одним значущим фактором ризику при роботі операторів. Тривале перебування в одній позі призводить до накопичення м'язової втоми та може спричинити розвиток захворювань опорно-рухового апарату. Для профілактики рекомендується щогодини робити перерви тривалістю 5-10 хвилин, під час яких виконувати вправи для розминки м'язів шиї, плечей, спини та кистей рук. Правильна постава під час роботи та відповідне ергономічне налаштування робочого місця також суттєво знижують ризик виникнення м'язово-скелетних розладів.

Роботодавець повинен забезпечити внутрішній регламент перерви для відпочинку, щоб знизити навантаження на організм працівників. Регулярні перерви сприяють збереженню працездатності, зменшенню втоми та покращенню самопочуття під час роботи. Відповідно до нормативних вимог, при роботі з відеодисплейними терміналами понад 4 годин на день рекомендується надавати додаткові регламентовані перерви сумарною тривалістю не менше 60 хвилин протягом робочого дня.

4.4 Висновок до розділу 4

Охорона праці є важливою складовою роботи будь-якого підприємства, що використовує інформаційні системи. Правильно організовані умови праці сприяють не лише збереженню здоров'я та працездатності персоналу, але й безпосередньо впливають на ефективність роботи системи та якість виконання операцій обліку.

У розділі розглянуто основні вимоги охорони праці для операторів, адміністраторів та технічного персоналу. Визначено ключові шкідливі виробничі фактори, що впливають на здоров'я працівників: зорове навантаження, статичне навантаження на м'язово-скелетну систему. Для кожного з виявлених факторів запропоновано конкретні профілактичні заходи.

Важливо, щоб кожен працівник мав чітке уявлення про правила безпеки та дотримувався їх під час виконання своїх обов'язків. Для цього проводяться навчання, де працівникам розповідають, як правильно працювати з технікою, як уникнути небезпек та що робити в екстрених ситуаціях. Також необхідно регулярно перевіряти стан обладнання в кімнаті, де розміщено сервер системи та клієнтські пристрої, адже несправне або неналежно працююче обладнання може створити небезпечні умови для працівників та клієнтів.

Дотримання правил безпеки та охорони праці допомагає зберегти здоров'я працівників, підвищити ефективність роботи системи, а також забезпечує стабільне та безпечне функціонування всього підприємства.

ВИСНОВКИ

У результаті виконання роботи було розроблено сучасний web-застосунок для автоматизованого обліку та управління товарно-матеріальними цінностями - система RDDH. Розроблений застосунок вирішує комплекс актуальних проблем, що виникають при традиційному підході до складського обліку, та забезпечує підприємство інструментом для ефективного управління запасами в режимі реального часу.

Завдяки системі RDDH операційний персонал може швидко та зручно виконувати ключові складські операції: реєструвати надходження товарів на склад, здійснювати їх списання, організовувати переміщення між складами. Вся ця інформація автоматично фіксується у базі даних як незмінні транзакції та стає доступною для перегляду в режимі реального часу. Це значно скорочує час на обробку операцій, усуває паперові журнали та мінімізує вплив людського фактора на точність обліку.

В інформаційній системі реалізовано функціонал управління каталогом товарів. Кожен товар отримує цифровий паспорт із усіма необхідними характеристиками. Адміністратор може в будь-який момент оновити інформацію про товар, додати нові позиції або видалити застарілі. Ця можливість забезпечує актуальність асортименту та точність облікових даних, що є критично важливим для коректного управління запасами.

Окремим важливим досягненням є впровадження системи ролей та розмежування прав доступу. Адміністратор системи отримує повний доступ до всіх функцій і налаштувань, тоді як рядовий оператор складу може виконувати лише відведені йому операційні дії. Такий підхід забезпечує захист від несанкціонованих змін у даних та підвищує загальну безпеку системи.

У системі реалізовано потужний механізм захисту від помилок обліку. Програмна блокування від'ємних залишків унеможливорює списання товарів більше, ніж наявно на складі. Принцип незмінності транзакцій гарантує, що жодна проведена операція не може бути таємно видалена або відредагована - це забезпечує повний

аудиторський слід усіх дій в системі. Завдяки цим механізмам підприємство отримує достовірну інформацію про стан запасів, на яку можна беззастережно покладатися при прийнятті управлінських рішень.

Реалізований функціонал аналітики та звітності дозволяє відповідальним особам оперативно отримувати актуальну інформацію про поточні залишки товарів у розрізі окремих складів, відстежувати динаміку руху товарів та формувати звіти для бухгалтерії та керівництва підприємства. Ця функція значно спрощує процес аналізу ефективності складської діяльності та сприяє прийняттю обґрунтованих управлінських рішень.

Система побудована на сучасному стеку технологій: клієнтська частина використовує HTML5, CSS3 та JavaScript для створення інтуїтивного інтерфейсу, серверна частина реалізована на PHP 8.0+ з використанням механізму PDO для безпечного доступу до бази даних, а для зберігання інформації використовується СУБД MySQL версії 8.0+.

Також у роботі було враховано вимоги охорони праці для того, щоб працівники могли безпечно використовувати інформаційну систему. Визначено основні шкідливі виробничі фактори при роботі з комп'ютерним обладнанням та запропоновано конкретні профілактичні заходи. Це зменшує ризики для здоров'я співробітників та забезпечує комфортні умови для їх роботи.

Розроблений web-застосунок є сучасним та ефективним інструментом, який допомагає покращити роботу підприємства в частині управління запасами, сприяє кращій взаємодії між адміністративним та операційним персоналом, а також забезпечує більш ефективний контроль за фінансами, залишками та користувачами системи.

СПИСОК ЛІТЕРАТУРИ

1. Zoho Inventory - Online Inventory Management Software. URL: <https://www.zoho.com/inventory/> (дата звернення: 10.05.2026).
2. Sortly - Inventory Management App. URL: <https://www.sortly.com/> (дата звернення: 10.05.2026).
3. Odoo Inventory - Open Source Warehouse Management. URL: <https://www.odoo.com/app/inventory> (дата звернення: 10.05.2026).
4. What is an Information System? 5 Components, Types, Advantages, Disadvantages. URL: <https://educationleaves.com/what-is-an-information-system-5-components-types-advantages-disadvantages-pdf-included/> (дата звернення: 12.05.2026).
5. Марченко А.В. Проектування інформаційних систем, 2016. - 90 с. URL: https://duikt.edu.ua/uploads/l_144_42481385.pdf (дата звернення: 15.05.2026).
6. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 17.05.2026).
7. Що таке моделювання даних? Типи (концептуальний, логічний, фізичний). URL: <https://www.guru99.com/uk/data-modelling-conceptual-logical.html> (дата звернення: 18.05.2026).
8. MySQL Documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 20.05.2026).
9. PHP Documentation. URL: <https://www.php.net/docs.php> (дата звернення: 20.05.2026).
10. REST API Design: Best Practices and Guidelines. URL: <https://swagger.io/resources/articles/best-practices-in-api-design/> (дата звернення: 22.05.2026).
11. Deployment Diagram Tutorial. URL: <https://online.visual-paradigm.com/diagrams/tutorials/deployment-diagram-tutorial/> (дата звернення: 25.05.2026).
12. RBAC - Role-Based Access Control: What Is It and When to Use It. URL: <https://auth0.com/docs/manage-users/access-control/rbac> (дата звернення: 27.05.2026).

13. Sequence Diagrams. URL: <https://www.maxzosim.com/sequence-diagrams/> (дата звернення: 30.05.2026).

14. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» від 05.07.1994 р. № 80/94-ВР. URL: <https://zakon.rada.gov.ua/laws/show/80/94-вр> (дата звернення: 07.06.2026).

15. Що таке охорона праці? URL: <https://profiteh.ua/shcho-take-okhorona-pratsi/> (дата звернення: 07.06.2026).

16. Inventory Management Systems: Types, Benefits, and Best Practices. URL: <https://www.oracle.com/scm/inventory-management/> (дата звернення: 08.06.2026).

ДОДАТОК А - ІНСТРУКЦІЯ КОРИСТУВАЧА

Для доступу до системи вам необхідно авторизуватися. На сторінці авторизації введіть вашу електронну адресу та пароль, надані адміністратором.

Система підтримує два рівні доступу — «Користувач» та «Адмін». Користувачі з правами Адміністратора мають додатковий доступ до розділу «Адміністрування» для управління довідниками та персоналом. У правому верхньому куті ви завжди можете побачити своє ім'я та поточну роль, а також вийти з акаунта.

Після входу ви потрапляєте на головну сторінку, яка є вашим центром управління. У верхній частині екрана розташовані кнопки для найчастіших операцій: Створити картку, Прихід, Списання, Переміщення.

Ви можете швидко оцінити загальну ситуацію на складі завдяки віджетам, які показують «Всього товарів на складі» та «Товари з критичним залишком». У нижній частині екрана знаходиться таблиця з історією останніх транзакцій.

Перед початком активної роботи з товарами, системі потрібні базові дані. Функції для їх створення знаходяться у меню «Робота з базою»: додайте місця зберігання, вказавши їх назву та адресу; створіть категорії товарів, вказавши назву та короткий опис; заповніть повну назву одиниць виміру та аббревіатуру. Ви можете видаляти записи з довідників, натиснувши на іконку кошика навпроти відповідного рядка.

Розділ «Робота з даними» дозволяє створювати нові позиції та відстежувати їх наявність. Перейдіть у меню «Робота з даними» -> «Створення картки». Введіть назву, початкову кількість, опис товару, а також оберіть категорію, одиницю виміру та склад зберігання зі списків.

На сторінці «Товари та Залишки» відображається весь ваш інвентар. Ви можете шукати товари за назвою чи артикулом за допомогою рядка пошуку. Доступне фільтрування за конкретним складом.

Кожен товар має візуальний статус: «В наявності», «Критичний залишок» або «Немає в наявності».

Усі рухи товарів фіксуються через відповідні меню операцій. Під час будь-якої дії система автоматично підкаже вам товар. «Прихід» додає вказану кількість до існуючого залишку товару. «Списання» зменшує кількість товару. Якщо ви спробуєте списати більше, ніж є на залишку, система видасть попередження. «Переміщення» дозволяє перевести товар з одного складу на інший. Склад відправлення підтягнеться автоматично, вам потрібно лише обрати склад призначення та кількість.

Для операцій приходу та списання ви можете вказати текстову причину операції, яка збережеться в історії.

Система автоматично генерує документацію на основі ваших дій. Перейдіть до розділу «Аналітика та Звіти», щоб створити документ формату PDF. Звіт міститиме всю історію операцій у вигляді таблиці і буде автоматично завантажений на ваш пристрій. За необхідності ви можете повністю видалити історію операцій, натиснувши «Очистити історію». Увага: цю дію неможливо скасувати.

У розділі «Контроль доступу» адміністратори можуть керувати персоналом. Створювати нові акаунти, вказавши ім'я співробітника, його email, пароль та рівень доступу.

Адміністратор може переглядати список усіх зареєстрованих осіб. Він може видаляти неактуальні акаунти.

ДОДАТОК Б - ФРАГМЕНТИ ПРОГРАМНОГО КОДУ

```

const renderSidebar = () => `
  <aside class="sidebar">
    <div class="logo">
      <i class="ph-fill ph-package"></i> RDDH
    </div>
    <div class="menu-section">
      <div class="menu-title">Основне</div>
      <ul class="menu">
        <li><a href="index.html" class="menu-item"><i class="ph ph-squares-four"></i>
Головна</a></li>
        <li><a href="goodAnInven.html" class="menu-item"><i class="ph ph-archive-box"></i>
Товари та Залишки</a></li>
        <li>
          <a href="#" class="menu-item" id="sysMenuToggle">
            <i class="ph ph-gear"></i> Налаштування та інфо
            <i class="ph ph-caret-down" style="margin-left: auto; font-size: 16px;"></i>
          </a>
          <div class="dropdown-container" id="sysDropdown">
            <a href="news.html" class="dropdown-item">Що нового?</a>
            <a href="about.html" class="dropdown-item">Про систему</a>
          </div>
        </li>
      </ul>
    </div>
    <div class="menu-section">
      <div class="menu-title">Робота</div>
      <ul class="menu">
        <li><a href="anAndRep.html" class="menu-item"><i class="ph ph-chart-line-up"></i>
Аналітика та Звіти</a></li>
        <li>
          <a href="#" class="menu-item" id="dataMenuToggle">
            <i class="ph ph-database"></i> Робота з даними
            <i class="ph ph-caret-down" style="margin-left: auto; font-size: 16px;"></i>
          </a>
          <div class="dropdown-container" id="dataDropdown">
            <a href="creation.html" class="dropdown-item">Створення картки</a>
            <a href="incoming.html" class="dropdown-item">Прихід</a>
            <a href="writeoff.html" class="dropdown-item">Списання</a>
            <a href="transfer.html" class="dropdown-item">Переміщення</a>
          </div>
        </li>
        <li><a href="contacts.html" class="menu-item"><i class="ph ph-headset"></i>
Підтримка</a></li>
      </ul>
    </div>
    <div class="menu-section" id="adminSection">
      <div class="menu-title">Адміністрування</div>
      <ul class="admin-menu">

```

```

</li>
  <a href="#" class="menu-item" id="usersMenuToggle">
    <i class="ph ph-users-three"></i> Контроль доступу
    <i class="ph ph-caret-down" style="margin-left: auto; font-size: 16px;"></i>
  </a>
  <div class="dropdown-container" id="usersDropdown">
    <a href="register.html" class="dropdown-item">Реєстрація</a>
    <a href="users.html" class="dropdown-item">Користувачі</a>
  </div>
</li>
</li>
  <a href="#" class="menu-item" id="dbMenuToggle">
    <i class="ph ph-hard-drives"></i> Робота з базою
    <i class="ph ph-caret-down" style="margin-left: auto; font-size: 16px;"></i>
  </a>
  <div class="dropdown-container" id="dbDropdown">
    <a href="category.html" class="dropdown-item">Створення категорії</a>
    <a href="unitMeasur.html" class="dropdown-item">Додати одиницю виміру</a>
    <a href="warehouse.html" class="dropdown-item">Додати склад</a>
  </div>
</li>
</ul>
</div>
</aside>
`;

```

ЛІСТИНГ 1 - Шаблон генерації бокової панелі.

```

if (!currentUser) {
  if (!publicPages.includes(currentPage)) {
    window.location.href = 'login.html';
    return;
  }
} else {
  if (publicPages.includes(currentPage)) {
    window.location.href = 'index.html';
    return;
  }
  const userNameEl = document.getElementById('userName');
  const userRoleTextEl = document.getElementById('userRoleText');
  const userAvatarEl = document.getElementById('userAvatar');
  if (userNameEl) userNameEl.textContent = currentUser.Full_Name;
  if (userRoleTextEl) userRoleTextEl.textContent = currentUser.Role;
  if (userAvatarEl) userAvatarEl.textContent = currentUser.Full_Name.charAt(0).toUpperCase();
  const adminSection = document.getElementById('adminSection');
  if (adminSection) {
    if (currentUser.Role !== 'Адмін') {
      adminSection.style.display = 'none';
    } else {
      adminSection.style.display = 'block';
    }
  }
}

```

```

    }
  }
}

```

Лістинг 2 - Логіка перевірки доступу та перенаправлення.

```

async function syncDB() {
  localStorage.setItem('rddh_inventory', JSON.stringify(inventoryDB));
  localStorage.setItem('rddh_operations', JSON.stringify(operationsDB));
  try {
    await fetch('DB.php', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ inventory: inventoryDB, operations: operationsDB })
    });
  } catch (error) {
    console.error('Помилка синхронізації з БД:', error);
  }
}

```

Лістинг 3 - Синхронізація з базою даних.

```

const applyFilters = () => {
  const searchTerm = inventorySearch.value.toLowerCase();
  const selectedWarehouse = warehouseFilter.value;
  const filteredData = inventoryDB.filter(item => {
    const matchesSearch = item.name.toLowerCase().includes(searchTerm) ||
    item.id.toLowerCase().includes(searchTerm);
    const matchesWarehouse = selectedWarehouse === 'all' || item.warehouse ===
    selectedWarehouse;
    return matchesSearch && matchesWarehouse;
  });
  renderTable(filteredData);
  updateWidgets(filteredData);
};

```

Лістинг 4 - Логіка фільтрації та пошуку товарів.

```

const opt = {
  margin: 10,
  filename: `Звіт_RDDH_${new Date().toLocaleDateString('uk-UA').replace(/\./g,
  '_')}.pdf`,
  image: { type: 'jpeg', quality: 0.98 },
  html2canvas: { scale: 2 },
  jsPDF: { unit: 'mm', format: 'a4', orientation: 'landscape' }
};
html2pdf().set(opt).from(element).save().then(() => {
  btn.innerHTML = '<i class="ph ph-check-circle"></i> Завантажено!';
  btn.style.backgroundColor = 'var(--success)';
  btn.style.color = '#fff';
  setTimeout(() => {

```

```

        btn.innerHTML = originalText;
        btn.style.backgroundColor = "";
        btn.style.color = "";
        btn.disabled = false;
    }, 3000);
});

```

Лістинг 5 - Генерація PDF-звіту.

```

<section class="quick-actions">
  <a href="creation.html" class="action-btn btn-primary">
    <i class="ph ph-plus-circle"></i> Створити картку
  </a>
  <a href="incoming.html" class="action-btn">
    <i class="ph ph-download-simple"></i> Прихід
  </a>
  <a href="writeoff.html" class="action-btn">
    <i class="ph ph-upload-simple"></i> Списання
  </a>
  <a href="transfer.html" class="action-btn">
    <i class="ph ph-arrows-left-right"></i> Переміщення
  </a>
</section>

```

Лістинг 6 - HTML-розмітка кнопок швидких дій.

```

<table>
  <thead>
    <tr>
      <th>Артикул</th>
      <th>Назва товару</th>
      <th>Склад</th>
      <th>Залишок</th>
      <th>Статус</th>
      <th>Дії</th>
    </tr>
  </thead>
  <tbody id="inventoryTableBody">
  </tbody>
</table>

```

Лістинг 7 - Структура таблиці інвентаризації.

```

<div class="widget-card">
  <div class="widget-header" style="align-items: flex-start; margin-bottom: var(--space-1);">
    <div>
      <div class="caption-text">Глобальне оновлення складських операцій</div>
      <h2 style="margin-top: 4px; margin-bottom: var(--space-2);">Запуск модуля
«Операційний облік 2.0»</h2>
    </div>
  </div>

```

```

    <div class="caption-text">11.04.2026</div>
  </div>
  <div style="color: var(--text-main); margin-bottom: var(--space-3);">
    <p style="margin-bottom: var(--space-2);">Тепер доступний повний цикл роботи з
товаром: реєстрація приходу, швидке списання та переміщення між складами.</p>
    <p class="caption-text" style="font-size: 14px; color: var(--text-
main);"><strong>Ключові зміни:</strong> Додано створення карток товарів з розширеними
характеристиками та інтелектуальний моніторинг залишків у реальному часі.</p>
  </div>
  <div style="display: flex; gap: var(--space-1);">
    <span class="tag tag-transfer">#Склад</span>
    <span class="tag tag-out">#High_Priority</span>
  </div>
</div>

```

Лістинг 8 — Розмітка блоку новин

```

<form id="operationForm" class="support-form">
  <input type="hidden" id="opType" value="in">
  <div class="form-group">
    <label class="form-label">Назва товару</label>
    <input type="text" id="opProduct" class="form-control" list="productsList"
autocomplete="off" required>
    <datalist id="productsList"></datalist>
  </div>
  <div class="form-group">
    <label class="form-label">Кількість (шт)</label>
    <input type="number" id="opQuantity" class="form-control" min="1" required>
  </div>
  <div class="form-group">
    <label class="form-label">Коментар до накладної (необов'язково)</label>
    <textarea id="opReason" class="form-control"></textarea>
  </div>
  <button type="submit" class="action-btn btn-primary btn-full-width" style="margin-top:
var(--space-2);">
    <i class="ph ph-check"></i> Зареєструвати прихід
  </button>
</form>

```

Лістинг 9 - Форма реєстрації приходу товару.

```

<div class="contact-layout op-grid-transfer">
  <div class="form-group">
    <label class="form-label">Зі складу</label>
    <input type="text" id="opFrom" class="form-control" readonly required
style="background-color: #f3f4f6; cursor: not-allowed;">
  </div>
  <div class="form-group">
    <label class="form-label">На склад</label>
    <input type="text" id="opTo" class="form-control" list="warehousesList"
autocomplete="off" required>
  </div>

```

```

        <datalist id="warehousesList"></datalist>
    </div>
</div>

```

Лістинг 10 - Блок маршрутизації для внутрішнього переміщення.

```

<form id="operationForm" class="support-form">
    <input type="hidden" id="opType" value="out">
    <div class="form-group">
        <label class="form-label">Назва товару для списання</label>
        <input type="text" id="opProduct" class="form-control" list="productsList"
autocomplete="off" required>
        <datalist id="productsList"></datalist>
    </div>
    <div class="form-group">
        <label class="form-label">Кількість до списання (шт)</label>
        <input type="number" id="opQuantity" class="form-control" min="1" required>
    </div>
    <div class="form-group">
        <label class="form-label">Причина списання</label>
        <textarea id="opReason" class="form-control" required></textarea>
    </div>
    <button type="submit" class="action-btn btn-full-width btn-error-action" style="margin-top:
var(--space-2);">
        <i class="ph ph-trash"></i> Списати зі складу
    </button>
</form>

```

Лістинг 11 - Форма списання товару.

```

<div class="contact-layout" style="margin-top: 0; gap: var(--space-2); grid-template-columns: 1fr 1fr
1fr;">
    <div class="form-group">
        <label class="form-label" for="prodCategory">Категорія</label>
        <select id="prodCategory" class="form-control" required>
            <option value="" disabled selected>Оберіть категорію</option>
        </select>
    </div>
    <div class="form-group">
        <label class="form-label" for="prodUnit">Одиниця виміру</label>
        <select id="prodUnit" class="form-control" required>
            <option value="" disabled selected>Оберіть одиницю</option>
        </select>
    </div>
    <div class="form-group">
        <label class="form-label" for="prodWarehouse">Склад зберігання</label>
        <select id="prodWarehouse" class="form-control" required>
            <option value="" disabled selected>Оберіть склад</option>
        </select>
    </div>
</div>

```

Лістинг 12 - Секція класифікації нової картки товару.

```

<section class="table-section">
  <div class="section-header-actions">
    <h2 class="section-title">Зареєстровані користувачі</h2>
    <a href="register.html">
      <button class="action-btn btn-primary btn-small">
        <i class="ph ph-plus"></i> Додати користувача
      </button>
    </a>
  </div>
  <table>
    <thead>
      <tr>
        <th>ID Користувача</th>
        <th>Повне ім'я</th>
        <th>Електронна пошта</th>
        <th>Рівень доступу</th>
        <th>Дії</th>
      </tr>
    </thead>
    <tbody id="usersTableBody">
      <tr><td colspan="5" style="text-align: center;">Завантаження...</td></tr>
    </tbody>
  </table>
</section>

```

Лістинг 13 - Структура списку користувачів.

```

<form id="supportForm" class="support-form">
  <div class="form-group">
    <label class="form-label">Ім'я користувача</label>
    <input type="text" class="form-control" required>
  </div>
  <div class="form-group">
    <label class="form-label">Email або робочий телефон</label>
    <input type="text" class="form-control" required>
  </div>
  <div class="form-group">
    <label class="form-label">Тема звернення / Проблемний модуль</label>
    <select class="form-control" required>
      <option value="" disabled selected>Оберіть категорію проблеми</option>
      <optgroup label="Базові операції">
        <option value="op-create">Створення карток товарів</option>
        <option value="op-search">Пошук та фільтрація</option>
      </optgroup>
      <optgroup label="Складський облік">
        <option value="stock-ops">Операційний облік</option>
        <option value="stock-monitor">Моніторинг залишків</option>
      </optgroup>
    </select>
  </div>

```

```

    <optgroup label="Адміністрування">
      <option value="admin-roles">Управління ролями та правами</option>
      <option value="admin-data">Робота з даними</option>
    </optgroup>
    <optgroup label="Аналітика та налаштування">
      <option value="analytics-reports">Звіти та аналітика</option>
      <option value="analytics-settings">Управління налаштуваннями</option>
    </optgroup>
  </select>
</div>
<div class="form-group">
  <label class="form-label">Опис проблеми</label>
  <textarea class="form-control" rows="4" required></textarea>
</div>
<div class="form-group">
  <label class="form-label">Прикріпити файл зображення (скріншот)</label>
  <input type="file" class="form-control file-input" accept="image/*">
</div>
<button type="submit" class="action-btn btn-primary" style="width: 100%; justify-
content: center; margin-top: var(--space-2);">
  Відправити запит <i class="ph ph-paper-plane-right"></i>
</button>
</form>

```

Лістинг 14 - Форма зворотного зв'язку служби підтримки.

```

<div class="form-container reg-form-container">
  <form id="registerForm" class="support-form">
    <div class="contact-layout grid-2-col">
      <div class="form-group">
        <label class="form-label" for="regName">Ім'я та Прізвище</label>
        <input type="text" id="regName" class="form-control" required>
      </div>
      <div class="form-group">
        <label class="form-label" for="regEmail">Електронна адреса</label>
        <input type="email" id="regEmail" class="form-control" required>
      </div>
    </div>
    <div class="contact-layout grid-2-col-mt">
      <div class="form-group">
        <label class="form-label" for="regPassword">Пароль</label>
        <input type="password" id="regPassword" class="form-control" required>
      </div>
      <div class="form-group">
        <label class="form-label" for="regRole">Роль</label>
        <select id="regRole" class="form-control" required>
          <option value="Користувач" selected>Користувач</option>
          <option value="Адмін">Адмін</option>
        </select>
      </div>
    </div>
  </form>
</div>

```

```

    <button type="submit" class="action-btn btn-primary btn-submit-centered">
      <i class="ph ph-check-square-offset"></i> Зареєструвати акаунт
    </button>
  </form>
</div>

```

Лістинг 15 - Форма реєстрації та призначення ролей.

```

<form id="createCategoryForm" class="support-form">
  <div class="contact-layout" style="margin-top: 0; gap: var(--space-2); grid-template-
columns: 1 fr;">
    <div class="form-group">
      <label class="form-label">Назва категорії</label>
      <input type="text" id="catTitle" class="form-control" required>
    </div>
    <div class="form-group">
      <label class="form-label">Опис категорії</label>
      <textarea id="catDesc" class="form-control" rows="4"></textarea>
    </div>
  </div>
  <button type="submit" class="action-btn btn-primary" style="width: 100%; justify-content:
center; margin-top: var(--space-2);">
    <i class="ph ph-plus"></i> Додати категорію
  </button>
</form>

```

Лістинг 16 - Форма створення нової категорії товару.

```

<div class="table-section" style="margin-top: var(--space-4); max-width: 800px;">
  <h3 class="section-title">Існуючі категорії</h3>
  <table>
    <thead>
      <tr>
        <th>ID</th>
        <th>Назва</th>
        <th>Опис</th>
        <th style="width: 50px; text-align: center;">Дії</th>
      </tr>
    </thead>
    <tbody id="categoryTableBody">
    </tbody>
  </table>
</div>

```

Лістинг 17 - Таблиця для відображення списку існуючих категорій.

```

<form id="createUnitForm" class="support-form">
  <div class="contact-layout" style="margin-top: 0; gap: var(--space-2); grid-template-
columns: 1 fr 1 fr;">
    <div class="form-group">

```

```

        <label class="form-label">Назва (повна)</label>
        <input type="text" id="unitTitle" class="form-control" required>
    </div>
    <div class="form-group">
        <label class="form-label">Абревіатура</label>
        <input type="text" id="unitAbbr" class="form-control" required>
    </div>
</div>
<button type="submit" class="action-btn btn-primary" style="width: 100%; justify-content:
center; margin-top: var(--space-2);">
    <i class="ph ph-plus"></i> Додати одиницю виміру
</button>
</form>

```

Лістинг 18 - Форма додавання нової одиниці виміру.

```

<form id="createWarehouseForm" class="support-form">
    <div class="contact-layout" style="margin-top: 0; gap: var(--space-2); grid-template-
columns: 1 fr;">
        <div class="form-group">
            <label class="form-label">Назва складу</label>
            <input type="text" id="warehouseTitle" class="form-control" required>
        </div>
        <div class="form-group">
            <label class="form-label">Адреса складу</label>
            <input type="text" id="warehouseAddress" class="form-control" required>
        </div>
    </div>
    <button type="submit" class="action-btn btn-primary" style="width: 100%; justify-content:
center; margin-top: var(--space-2);">
        <i class="ph ph-plus"></i> Додати склад
    </button>
</form>

```

Лістинг 19 - Форма реєстрації нового складу.