

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Web-орієнтована система підтримки діяльності складу

Здобувача студент(ки) групи ІТ.м-22 Сахно Анастасії Федорівни

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_  
(підпис)

Анастасія САХНО  
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник старший викладач кафедри ІТ к.т.н Ольга БОЙКО  
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

\_\_\_\_\_  
(підпис)

**Суми – 2023**

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентіві**

Сахно Анастасії Федорівни

(прізвище, ім'я, по батькові)

**1 Тема кваліфікаційної роботи** Web-орієнтована система підтримки діяльності складу

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

**2 Термін здачі студентом кваліфікаційної роботи** «11» \_\_\_\_\_ грудня \_\_\_\_\_ 2023 р.

**3 Вхідні дані до кваліфікаційної роботи** запит на розробку web-орієнтованої системи

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** аналіз предметної області, постановка задачі та методи дослідження, моделювання та проектування, практична реалізація web-орієнтованої системи.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації)** актуальність, мета кваліфікаційної роботи, задачі дослідження, аналіз аналогів, функціональні вимоги, інструменти реалізації, проектування web-орієнтованої системи, реалізована база даних, практична реалізація, висновки.

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**



## АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Web-орієнтована система підтримки діяльності складу».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 31 найменувань, додатків. Загальний обсяг роботи – 68 сторінок, у тому числі 58 сторінок основного тексту, 3 сторінки списку використаних джерел, 8 сторінок додатків.

Актуальність роботи полягає в спрощенні процесів обліку та доступу до інформації про товари, а також забезпечення ефективного моніторингу та управління запасами.

Мета роботи: розробка web-орієнтованої системи підтримки діяльності складу «SpaceStock». У першому розділі роботи проведено ретельний аналіз обраної предметної області. В даному розділі здійснено огляд актуальних наукових досліджень та публікацій, а також проведено аналіз аналогів та поставлення задачі. У другому розділі сформульовано мету та конкретні задачі дослідження, спрямовані на розробку web-орієнтованої системи. Також в даному розділі проведено відбір засобів для реалізації системи. Третій розділ спрямований на проектування інформаційної системи. Виконано проектування за допомогою контекстної діаграми та декомпозиції, а також створено діаграму варіантів використання, відображаючи функціонал системи. Також спроектовано базу даних. У четвертому розділі було виконано реалізацію системи та детально описано використання розробленої інформаційної системи підтримки діяльності складу. Розглянуті аспекти включають взаємодію з системою зі сторони користувача та адміністратора.

Ключові слова: Автоматизація, склад, адміністратор, працівник складу, Web-орієнтована система, Java, PostgreSQL.

## ЗМІСТ

ВСТУП.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень та публікацій .....	8
1.2 Аналіз існуючих систем.....	9
2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	8
2.1 Постановка задачі.....	<b>1Error! Bookmark not defined.</b>
2.2 Методи дослідження .....	19
3. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ.....	21
3.1 Розроблення структурно-функціональної моделі системи.....	<b>2Error! Bookmark not defined.</b>
3.2 Проектування моделі бази даних .....	<b>Error! Bookmark not defined.</b>
3.3 Моделювання варіантів використання системи .....	25
4. ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	28
4.1 Проектування структури Web-орієнтованої системи .....	28
4.2 Програмна реалізація .....	29
4.3 Використання Web-орієнтованої системи зі сторони працівника складу.....	35
4.4 Використання Web-орієнтованої системи зі сторони адміністратора .....	47
4.5 Використання Web-орієнтованої системи з мобільного пристрою.....	50
ВИСНОВОК .....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	55
ДОДАТОК А .....	58
ДОДАТОК Б.....	67

## ВСТУП

Сучасний підхід до ведення бізнесу, особливо в галузі торгівлі та логістики, знаходиться в умовах глобалізації та постійно зростаючими вимогами споживачів, вимагає швидкості, точності та ефективності у всіх аспектах своєї діяльності. Зокрема, для підприємств, які займаються зберіганням і розподілом товарів, ефективне управління складською діяльністю стає ключовим фактором для забезпечення конкурентоспроможності та задоволення потреб клієнтів [1].

Проблема управління складською діяльністю полягає в необхідності ефективного контролю за запасами, оптимізації простору та ресурсів, врахуванні термінів придатності товарів, підтримці прозорого ланцюжка постачання, та в багатьох інших аспектах [2]. Ці завдання стають все більш складними при зростанні розмірів складу і обсягу товарообігу, а також при взаємодії з різноманітними сторонами, такими як постачальники, клієнти та інші підрозділи підприємства.

Тому, в даному контексті, створення web-орієнтованої системи підтримки діяльності складу, може стати важливим рішенням для вирішення цих проблем. Дана web-орієнтована система забезпечить автоматизацію та оптимізацію процесів управління запасами, контролю над станом товарів на складі, та інших аспектів управління складською діяльністю.

Система обліку для складу, розроблена з використанням сучасних технологій веб-розробки, має за мету сприяти підвищенню продуктивності та зниженню ризиків управління запасами. Ця система повинна стати інструментом для прийняття стратегічних рішень та підвищення ефективності управління складськими ресурсами.

Дана магістерська робота спрямована на розгляд сутності та стану практичної проблеми в галузі управління складською діяльністю та обґрунтування необхідності розробки web-орієнтованої системи для її підтримки. Буде розглянуто сучасні методи та інструменти, що використовуються для управління запасами та досліджено можливість їхнього вдосконалення за допомогою інформаційних

технологій. Також буде розглянуто практичну реалізацію веб-орієнтованої системи для підтримки діяльності складу та проведення аналізу його ефективності.

Дослідження цієї теми має велике значення для підприємств, що оптимізують свою складську діяльність, а також для академічного світу, оскільки це може сприяти розвитку нових методів та підходів у цій галузі.

**Об'єктом дослідження** є підтримка діяльності складу.

**Предметом дослідження** є процес діяльності складу.

**Мета дослідження** полягає у розробці Web-орієнтованої системи підтримки діяльності складу «SpaceStock».

**Практичною цінністю** є автоматизація управління запасами, спрощення процесів обліку та доступу до інформації про товари.

Наступні задачі поставлені для кваліфікаційної роботи магістра:

- Провести огляд останніх досліджень і публікацій.
- Проаналізувати існуючі системи управління складською діяльністю.
- Визначити методи дослідження які будуть використані для подальшої розробки веб-орієнтованої системи.
- Розробити структурно-функціональну модель системи, модель бази даних та діаграму використання веб-орієнтованої системи.
- Розробити функціонал веб-орієнтованої системи підтримки діяльності складу.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень та публікацій

Використання інформаційних технологій швидко зросло в багатьох сферах людського життя. Стрімкий розвиток і зростаюча конкуренція в комп'ютерних і комунікаційних технологіях у світі, що глобалізується, змушують підприємства інтенсивніше використовувати інформаційні технології [3]. Вони використовуються у всіх сферах нашого життя, таких як освіта, охорона здоров'я, зв'язок, транспорт, торгівля та розваги. Інформаційні технології важливі в отриманні, зберіганні та передачі інформації та створенні безпечнішого середовища для компаній, щоб ефективніше контролювати та керувати своїми процесами [4].

Склад – ключовий елемент, спеціально призначений для зберігання та обробки товарів. Можна класифікувати склади за їх функціональністю [5]:

1. Склади для зберігання та розподілу готової продукції – слугують для зберігання готових товарів та їх подальшого розподілу на ринку або доставки клієнтам [6].

2. Склади для зберігання та обробки сировини – використовуються для зберігання та обробки сировини та комплектуючих, які використовуються виробничими підприємствами [7].

3. Транзитні склади – сприяють тимчасовому зберіганню та перевезенню товарів від одного пункту до іншого в рамках логістичного ланцюга [8].

4. Склади для обслуговування послуг – забезпечують необхідним обладнанням та матеріалами для підтримки сервісних або ремонтних робіт.

5. Склади для зберігання харчових продуктів – забезпечують оптимальні умови для зберігання харчової продукції.

6. Експрес-склади – орієнтовані на швидку обробку та доставку замовлень, зазвичай пов'язаних з електронною торгівлею.

7. Резервні склади – передбачені для зберігання додаткових запасів товарів для покриття непередбачуваних ситуацій [9].



Ця класифікація дозволяє краще розуміти різноманітність складів та адаптувати їх функції до конкретних потреб підприємства.

Цільова функція складу полягає в тому, щоб забезпечити нормальну роботу ланцюга поставок, забезпечуючи належний рівень запасів і швидкість обробки замовлень [10].

Розвиток індустрії, а також інформаційна підтримка та автоматизація виробничих процесів створили попит на управління складською діяльністю для підтримки виробничих процесів. Складування є одним із основних процесів у ланцюжку поставок і відповідає за обробку запасів [11]. Ефективне виконання складських операцій часто визначає ефективність реалізованих товарних потоків. Тому інформаційна підтримка та її використання в обміні інформацією між партнерами, особливо про рівні запасів у реальному часі, є критично важливими [12]. Оптимізація процесів на складі повинна позитивно вплинути на конкурентоспроможність підприємства, знизити витрати та покращити обслуговування клієнтів [13]. З цієї точки зору, створення web-орієнтованої системи підтримки діяльності складу є актуальною потребою, яка відповідає сучасним викликам бізнесу.

Однією з основних актуальних проблем управління складською діяльністю є ефективний контроль над запасами. Недостача запасів може призвести до того, що підприємство не зможе задовольнити попит клієнтів вчасно [14]. Це може викликати втрату продажів, невдоволеність клієнтів і навіть втрату клієнтів. З іншого боку, надмір запасів призводить до збільшення витрат на зберігання, обезцінювання товарів і зв'язаних із цим втрат коштів [15]. Ефективне управління запасами означає не лише контроль за їхньою кількістю, але і оптимізацію складських процесів, включаючи замовлення, приймання товарів, розподіл і контроль термінів придатності. Важливо знижувати зв'язані зі складанням та зберіганням запасів витрати [16].

Також важливо враховувати, що технологічний прогрес ніколи не зупиняється, і тому підприємства повинні постійно адаптуватися до нових вимог та можливостей [17]. Відтак, розробка та використання web-орієнтованої системи

підтримки складської діяльності залишається актуальною, оскільки це дозволяє підприємствам залишатися конкурентоздатними, ефективними та готовими до викликів сучасного бізнес-середовища.

Зростаюча конкуренція на ринку також змушує підприємства шукати способи підвищити свою ефективність [18]. Використання web-орієнтованої системи підтримки діяльності складу, який орієнтується саме на зберігання готової продукції, може бути однією з ключових стратегій для досягнення цієї мети.

## 1.2 Аналіз існуючих систем

Автоматизація систем складів виконує ключову роль у сучасному бізнесі через низку важливих переваг. По-перше, це дозволяє ефективно керувати запасами та забезпечити постійну доступність товарів для клієнтів, що сприяє підвищенню клієнтського сервісу та задоволеності клієнтів [19]. Відстеження запасів в реальному часі та автоматизоване оновлення дозволяють уникнути необхідності проводити ручний облік та реагувати на недостачу або надмір запасів.

По-друге, автоматизація систем складів сприяє покращенню продуктивності та ефективності управління складом. Вона дозволяє простіше відстежувати весь життєвий цикл товарів, від поставки до відправлення, а також раціоналізувати процеси, що полегшує взаємодію з постачальниками та клієнтами. Автоматизовані системи також забезпечують збереження інформації та можливість аналізу даних, що сприяє прийняттю обґрунтованих стратегічних рішень та оптимізації складських операцій [20].

Переглянути основний функціонал можна на прикладі вже існуючих зразків. Для прикладу, можна розглянути, web-додатки підтримки складів «TradeGecko» [21] та «ShipFusion» [22] та RightControl [23].

ShipFusion (рис.1.1, рис.1.2) є рішенням для виконання замовлень і служба управління запасами, що володіє своїми складами в Канаді та США. Панель інструментів виглядає інтуїтивно зрозуміло. Є можливість управляти усіма

аспектами власного облікового запису через панель управління.

Функціонал ShipFusion поділяється на наступні категорії.

- Збір деталей – Підбір продуктів для замовлення
- Пакетний збір – Замовлення можуть оброблятися пакетно або окремо.
- Обробка замовлення
- Доставка та оплата
- Звітність (рис.1.3) – перегляд рівнів запасу, вартість доставки і тд.

Із мінусів можна виділити, що на веб-сайті та доволі важко орієнтуватися, якщо потрібні конкретні дані, питання, такі як вартість доставки, вартість товару. Пошуки виводять за межі сайту, і навіть тоді складно знайти потрібну інформацію.

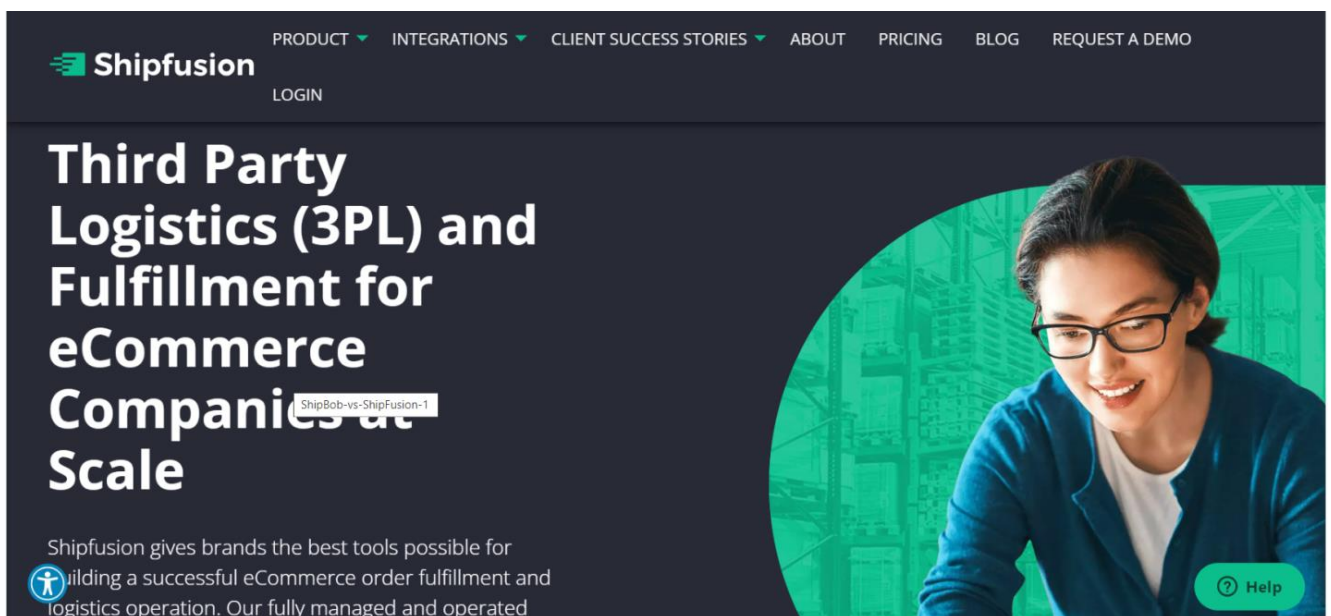


Рисунок 1.1 – Головна сторінка ShipFusion

*Джерело: [13]*

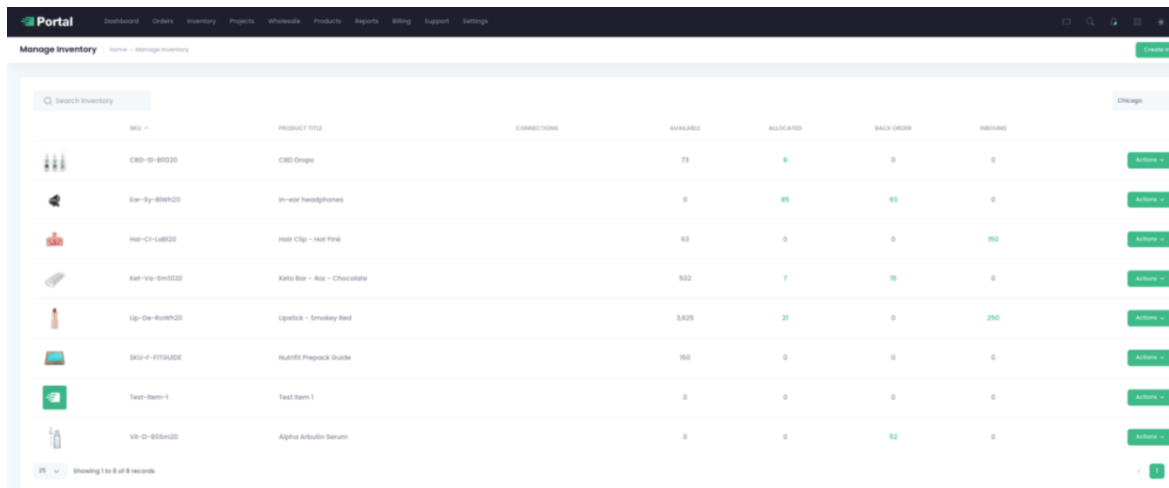
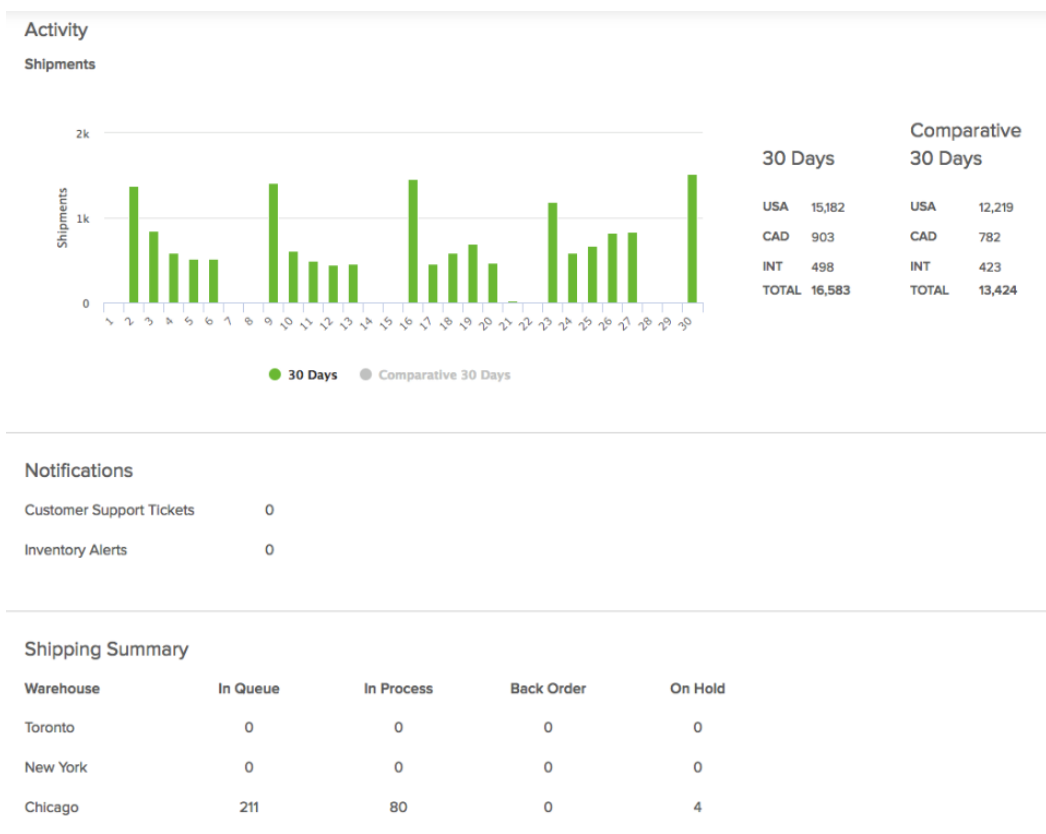


Рисунок 1.2 – Навігація ShipFusion

Джерело: [13]



Notifications

Customer Support Tickets	0
Inventory Alerts	0

Shipping Summary

Warehouse	In Queue	In Process	Back Order	On Hold
Toronto	0	0	0	0
New York	0	0	0	0
Chicago	211	80	0	4

Рисунок 1.3 – Приклад створення звітності ShipFusion

Джерело: [13]

TradeGecko (рис.1.4) є веб-додатком, який відзначається сучасним дизайном та розширеним функціоналом для управління складом. Підходить для компаній, які обробляють великі об'єми запасів. Додаток надає можливості для ефективного керування запасами (рис.1.5), включаючи відстеження товарів, створення звітів та аналітики (рис.1.6, рис.1.7), а також інтеграцію з іншими системами. Один із головних плюсів "TradeGecko" - це його сучасний інтерфейс та можливість налаштувати рівні доступу користувачів для забезпечення безпеки.

Основною проблемою додатку є те, що програмне забезпечення працює не так швидко як повинно. Багато користувачів скаржаться на доволі часті збої.

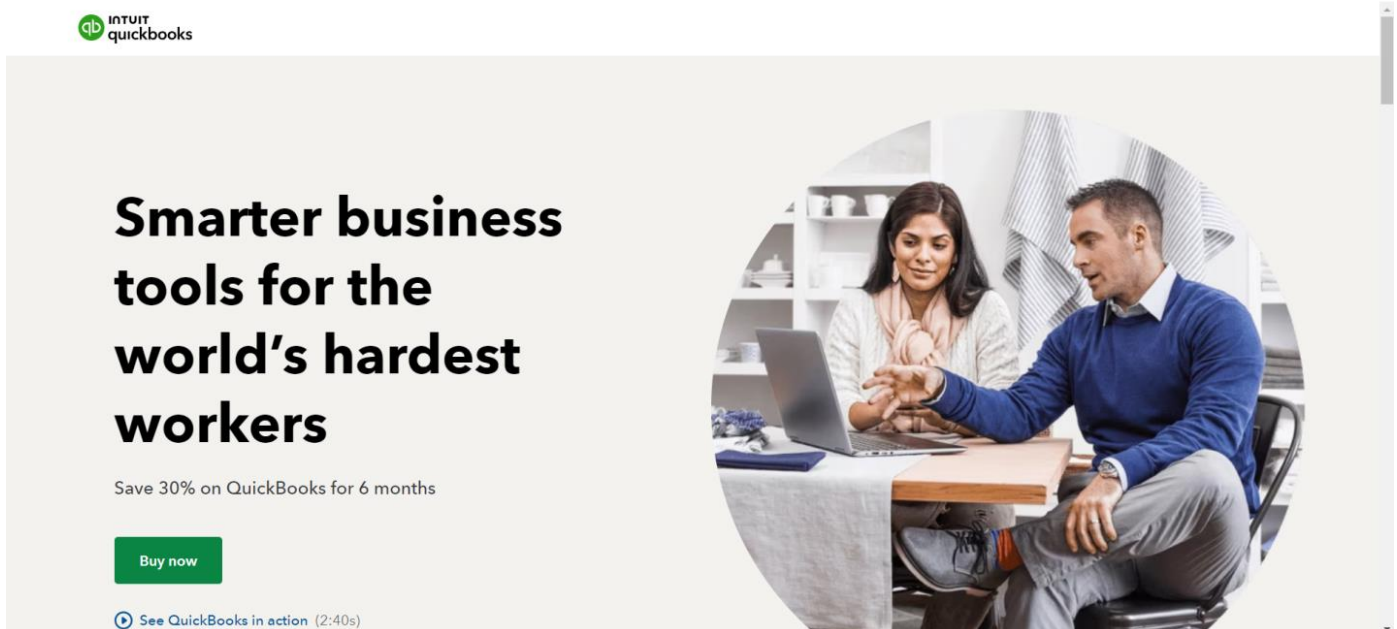


Рисунок 1.4 – Головна сторінка TradeGecko

*Джерело: [14]*

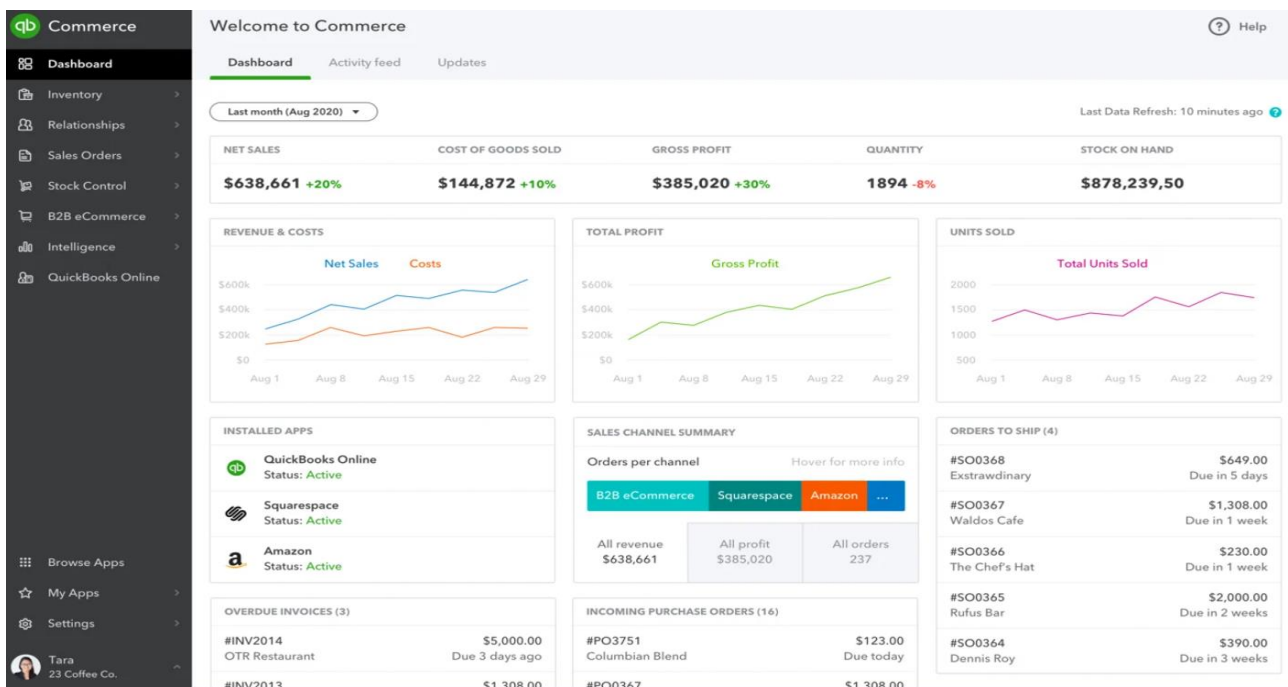


Рисунок 1.5 – Звітність TradeGecko

Джерело: [14]

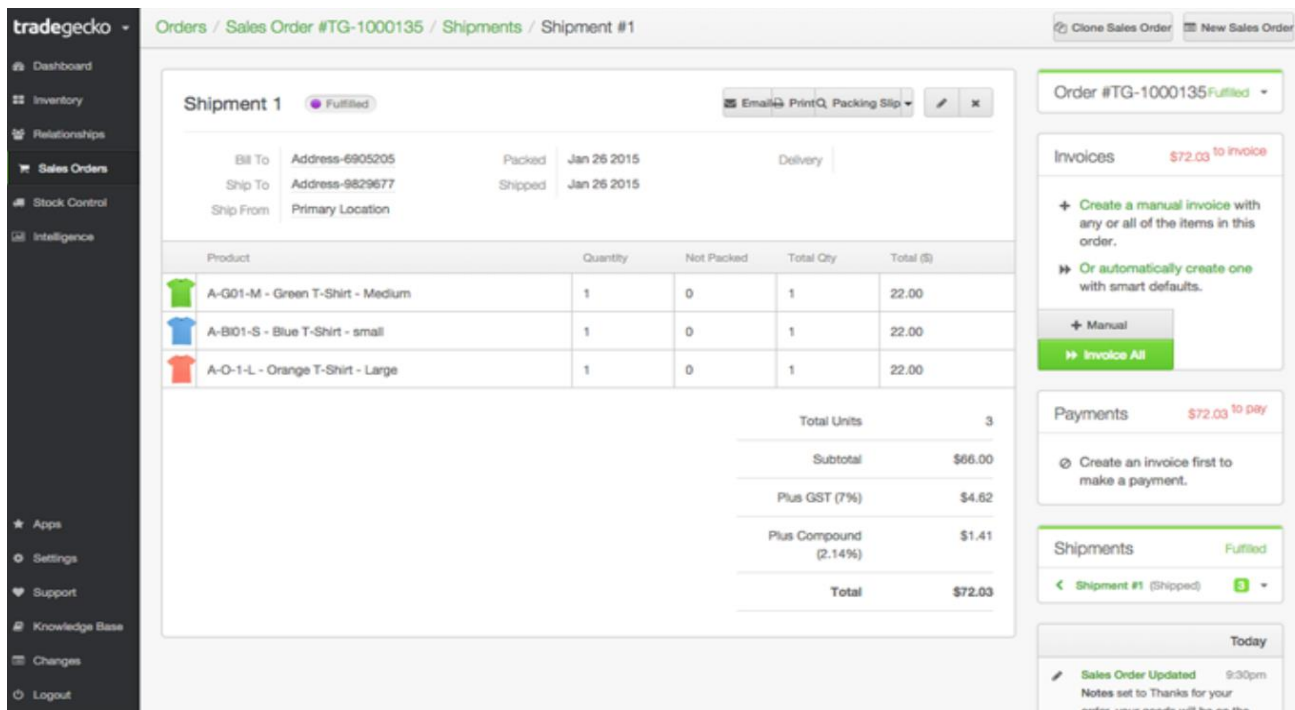


Рисунок 1.6 – Навігація web-додатку TradeGecko

Джерело: [14]

Inventory (12 of 1592)

Q t-shirt

Import Products Update via CSV Export Products New Product

Active All + Filter

<input type="checkbox"/>	Name	Supplier	Product Type	Brand	Available	Status	Last Updated
<input checked="" type="checkbox"/>	Dark Gray T-Shirt	Acme Apparel	Apparel	Acme	150 in 7 variants	Active	Aug 28 2015
<input checked="" type="checkbox"/>	Green T-Shirt	Acme Apparel	Apparel	Acme	-55 in 3 variants	Active	Aug 28 2015
<input checked="" type="checkbox"/>	Light Blue T-Shirt	Acme Apparel	Apparel	Acme	199 in 3 variants	Active	Aug 28 2015
<input checked="" type="checkbox"/>	Light Gray T-Shirt	Acme Apparel	Apparel	Acme	135 in 3 variants	Active	Aug 28 2015
<input type="checkbox"/>	Orange T-Shirt	Acme Apparel	Apparel	Acme	59 in 3 variants	Active	Jan 26 2015
<input type="checkbox"/>	Pink T-Shirt	Acme Apparel	Apparel	Acme	30 in 3 variants	Active	Jan 26 2015
<input type="checkbox"/>	Purple T-Shirt	Acme Apparel	Apparel	Acme	30 in 3 variants	Active	Jan 26 2015
<input type="checkbox"/>	Red T-Shirt	Acme Apparel	Apparel	Acme	150 in 3 variants	Active	Jan 26 2015

Рисунок 1.7 – Управління продуктами на складі web-додатку TradeGecko.

*Джерело: [14]*

RightControl – це система для управління складом та обліку, спрямований на підтримку підприємств у керуванні запасами та оптимізації складських операцій. Web-система надає можливості для ведення обліку товарів, відстеження запасів, створення звітів та ведення фінансового обліку пов'язаних операцій.

Основний функціонал RightControl:

- Управління запасами (рис.1.8, рис.1.9).
- Звітність та аналітика.
- Інтеграція з бухгалтерськими системами.
- Розподіл прав користувачів (рис.1.10).

Однак важливо враховувати, що RightControl може мати свої обмеження у функціоналі та дизайні порівняно з іншими аналогами.

**Booking In**

**Delivery**

Entry Date: 28/05/2015 PO Number: 19 Supplier Name: SUPPLIER User: J

**Model Details**

Category: PC Manufacturer: ACER Model Name: AC1 Unit Cost: 350 Sell price: 0.00

Sub Category: Intel processor Stock Code: IP1 P/N: Fitting:

Description: BLACK

**Item Details**

Report: Condition: Grade: Serial Number: 0 Location: Barcode Number: Extra Fields: 1: 2: 3: 4: 5:

Close Print Barcodes Finish

Рисунок 1.8 - Інтерфейс додатку RightControl

*Джерело: [15]*

File Settings Help

Options  
Label Designer  
Database Connect  
Change Password

Options

logo company financial product user

Category | Sub Category | Stock Grade | Stock Condition | Stock Attributes

Stock Category

Add  
Remove

Close

Рисунок 1.9 - Управління складом додатку RightControl

*Джерело: [15]*



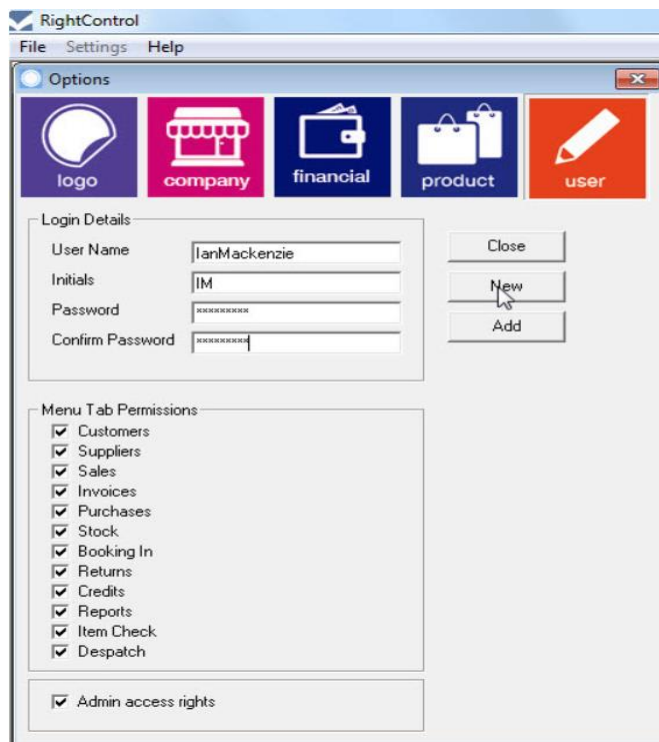


Рисунок 1.10 – Створення юзера з вибором прав доступу

*Джерело: [15]*

Таблиця 1.1 – Порівняльна таблиця характеристик аналогів web-орієнтованої системи

*Джерело: побудовано автором*

Характеристика/ Магазин	ShipFusion	TradeGecko	RightControl	SpaceStock
Зручний інтерфейс	-	+	-	+
Розподіл прав	+	+	+	+
Можливість звітності	+	+	+	+
Ціна	-	-	+	+

Продовження таблиці 1.1

Фільтрація запасів	+	+	-	+
Управління запасами	+	+	+	+
Управління обліковим записом	-	+	-	+
Адаптивність під мобільні пристрої	+	+	-	+

ShipFusion та RightControl отримали оцінку "менше" у аспекті зручного інтерфейсу, що свідчить про меншу зручність користування. Насправді, це важливий аспект для системи обліку на складі, оскільки спрощений та зручний інтерфейс полегшує роботу користувачів.

Усі аналоги отримали оцінку «плюс» у пункті розподіл прав користувачів, що свідчить про їхню здатність налаштовувати рівні доступу користувачів та забезпечувати безпеку даних на складі. Так само усі аналоги надають можливість створювати звітність.

Доступ з мобільного телефону мають додатки «ShipFusion» та «TradeGecko», що є важливим критерієм. Можливість управління запасами присутня у усіх трьох розглянутих аналогів.

Зважаючи на порівняння аналогів було вирішено створити власну Web-орієнтовану системи підтримки діяльності складу, враховуючи важливість кожного критерію, щоб створити власний оптимальний.

## 2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

### 2.1 Постановка задачі

Основною метою роботи є розробка Web-орієнтованої системи підтримки діяльності складу, яка відповідає сучасним вимогам управління складською діяльністю та забезпечує оптимальний контроль над запасами, зручну навігацію та користувацький інтерфейс. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Аналіз існуючих web-систем для управління складською діяльністю для ідентифікації їхніх переваг та недоліків.
2. Розробка web-орієнтованої системи, яка включає в себе модулі для ефективного контролю над запасами, покращеної навігації та користувацького інтерфейсу та можливістю створення звітності.
3. Впровадження розробленої web-системи. Демонстрація роботи web-орієнтованої системи зі сторони різних користувачів.

Для web-орієнтованої системи підтримки діяльності складу необхідно врахувати ряд вимог:

- Web-орієнтована система повинна забезпечувати можливість контролю над запасами, включаючи додавання, редагування та видалення товарів.
- Повинна бути можливість генерування звітів щодо стану запасів.
- Web-орієнтована система повинна бути розроблена з використанням сучасних технологій та мов програмування.
- Користувачі повинні мати можливість легко оволодіти інтерфейсом та використовувати систему без значних труднощів.
- Web-орієнтована система повинна мати зручну навігацію та інтуїтивний інтерфейс, щоб користувачі могли швидко та ефективно виконувати необхідні завдання.

- Система повинна підтримувати можливість розподілу прав доступу для різних користувачів відповідно до їхніх ролей та відповідальностей.
- Адміністратор системи повинен мати можливість налаштовувати рівні доступу користувачів та контролювати їхню роботу.
- Web-орієнтована система повинна бути адаптивною до різних пристроїв.
- Web-орієнтована система повинна надавати можливість сканувати товари або палети, які надійшли на склад, за QR кодом.

Даний розділ постановки задачі становить основу для подальшої розробки web-орієнтованої системи та дозволяє визначити критерії успішності та результативності роботи, які будуть враховані під час реалізації та впровадження системи.

## 2.2 Методи дослідження

Для реалізації Web-орієнтованої системи підтримки діяльності складу обрано монолітну архітектуру. Монолітна архітектура вважається традиційним способом побудови додатків. Монолітний додаток будується як єдине і неподільне ціле. Такий підхід містить інтерфейс користувача на стороні клієнта, програму на стороні сервера та базу даних, а також забезпечує уніфіковану платформу, де всі функції керуються та обслуговуються в одному місці [24].

Вибір технологій які будуть використані в розробці web-орієнтованої системи є надзвичайно важливим. Адже правильний вибір технологічного стеку є ключовим етапом при створенні високоефективних, надійних та інноваційних програмних продуктів.

Для програмної реалізації Web-орієнтованої системи будуть використовуватися HTML, CSS та Java.

Java обрана як основна мова програмування для реалізації веб-додатку. Висока крос-платформеність, широкий вибір бібліотек та активна спільнота роблять її відмінним вибором для проектів .

Використання Bootstrap надає не лише естетично приємний дизайн, але й забезпечує швидкість розробки та адаптивність інтерфейсу. Готові компоненти та гнучка система сіток допоможуть створити зручний та сучасний інструмент для користувачів, що працюють із системою обліку для складу [25].

Spring MVC та Spring Boot обрані для реалізації веб-сервера. Spring MVC забезпечить підтримку паттерну MVC для контролю веб-інтерфейсу, а Spring Boot - швидке налаштування та розгортання веб-додатку [26].

FreeMarker обраний для створення HTML-шаблонів. Простий синтаксис та потужні можливості роблять його зручним інструментом для генерації сторінок, адже FreeMarker легкий у розробці та підтримує динамічний контент на web сторінках [27].

Для системи управління базою даних було обрано PostgreSQL через його високу стабільність та надійність. Забезпечена підтримка транзакцій гарантує точність та відновлення даних в стійкому стані [28]. Гнучкість та можливість розширення PostgreSQL підходять для адаптації бази даних під зростання обсягу операцій та даних на складі [29]. Hibernate буде використовуватися для об'єктно-реляційного відображення, спрощуючи взаємодію з базою даних.

## 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

### 3.1 Розроблення структурно-функціональної моделі системи

Контекстна діаграма інформаційної системи необхідна для графічного зображення взаємодії між функціями у web-орієнтованій системі. Основною метою є подання високорівневої структури системи та визначення її функціональних аспектів, діаграми та їх роль [30].

Блоки на контекстній діаграмі інформаційної системи представляють функції або операції, які виконує система. Кожен блок описує конкретний аспект роботи системи.

Вхідні стрілки вказують на вхідні дані чи ресурси, необхідні для виконання функції. Вихідні стрілки показують результат або вплив функції на інші частини системи.

Управлінські стрілки вказують на те, які елементи керують або контролюють функцію. Механізми представляють технології або інструменти, які використовуються для виконання функції.

Керуючі стрілки показують взаємодію між різними функціями та їх порядок виконання. Вони допомагають визначити послідовність функцій у системі.

На головному блоці розташована основна функція, яку виконує система в цілому. Відсутність вхідних стрілок може вказувати на те, що ця функція є початковою точкою всієї системи [31].

Для web-орієнтованої системи підтримки складу було визначено наступні дані інформаційної системи:

– Вхідні дані: Запит користувача, Персональні дані.

Управління: Нормативні документи, Права доступу.

– Механізми: Користувач, Інформаційна система, Апаратне забезпечення.

– Вихідні дані: Стан запасів, Супроводжуюча накладна.

Контекстна діаграма є важливим інструментом для розуміння структури та функціональності системи на високому рівні, оскільки вона дозволяє аналізувати взаємозв'язки між функціями та елементами системи, що важливо при проектуванні та оптимізації системи [32].

На рисунку 3.1 відображена контекстна діаграма інформаційної системи в нотації IDEF0 web-орієнтованої системи підтримки складу.



Рисунок 3.1 - Контекстна діаграма інформаційної системи в нотації IDEF0

*Джерело: побудовано автором*

Декомпозиція першого рівня IDF0-діаграми включає розгляд більш деталізованих процесів, які необхідно виконати для виконання кожної функції на верхньому рівні (рис.3.2). Було виділено наступні процеси, які вказані у таблиці 3.1.

Таблиця 3.1 – Вхідні та вихідні дані для декомпозиції першого рівня IDFO-діаграми.

*Джерело: побудовано автором*

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Перехід до web-додатку	Персональні дані	Нормативні документи, Відповідність маркуванню, Права користувачів	ІС, Працівник складу, Апаратне забезпечення	Успішна авторизація
Сканування товару за QR кодом	Успішна авторизація			Просканий товар
Додавання в залишок	Просканий товар			Доданий товар в залишок
Пошук / фільтрація товарів на складі	Успішна авторизація, просканий товар			Обраний продукт
Управління товарообігом	Обраний продукт			Отримана та оброблена інформація
Формування документації	Отримана та оброблена інформація			Супроводжувальна документація



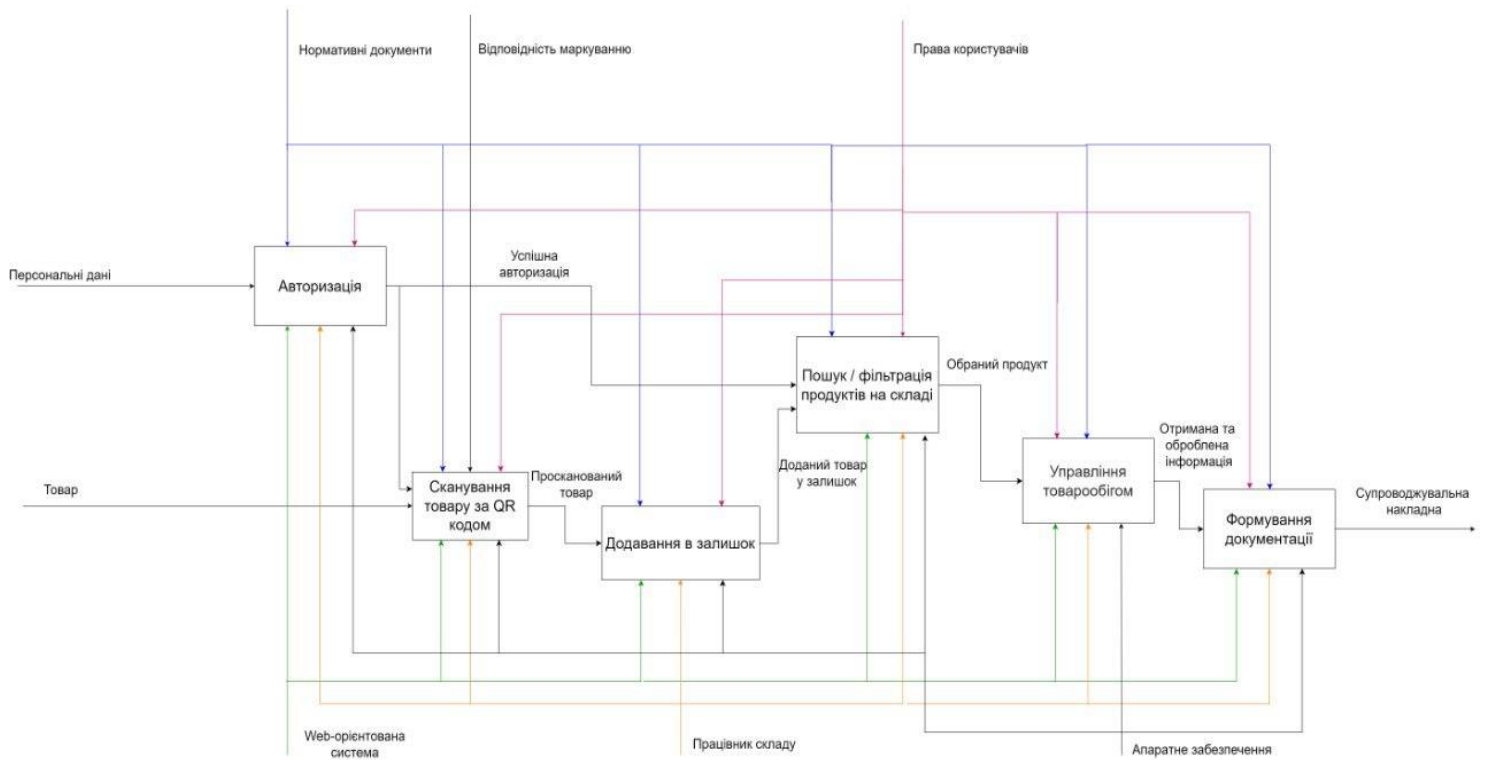


Рисунок 3.2 – Діаграма декомпозиції

*Джерело: побудовано автором*

## 2.2 Проектування моделі бази даних

Проектування моделі бази даних є важливим етапом у розробці інформаційних систем. Цей процес включає в себе створення структури бази даних, визначення таблиць, відносин між ними та визначення атрибутів [33].

В ході проектування були створені наступні сутності:

- **Product** – таблиця містить інформацію про кожний продукт розташований на складі.
- **Stock\_Items** – продукти які знаходяться в стоці.
- **Stock\_item\_type\_dim** – тип продукту що знаходиться на складі. Всього у системі два типи продуктів: Одиничний продукт та Палет продуктів.

- Operations – таблиця містить дані про операції можливі у web-орієнтованій системі.
- Operation\_type\_dim - таблиця містить типи операцій які можливі у системі.
- Categories – таблиця містить інформацію про наявні категорії у системі.
- Users – таблиця має інформацію про користувачів системи. У системі дві ролі. Працівник складу має право виконувати операції з продуктами на складі. Адміністратор має право додавати нових користувачів до системи та виконувати дії, що і працівник складу.

Переглянути фізичну модель бази даних web-орієнтованої системи підтримки складу можна на рисунку 3.3.

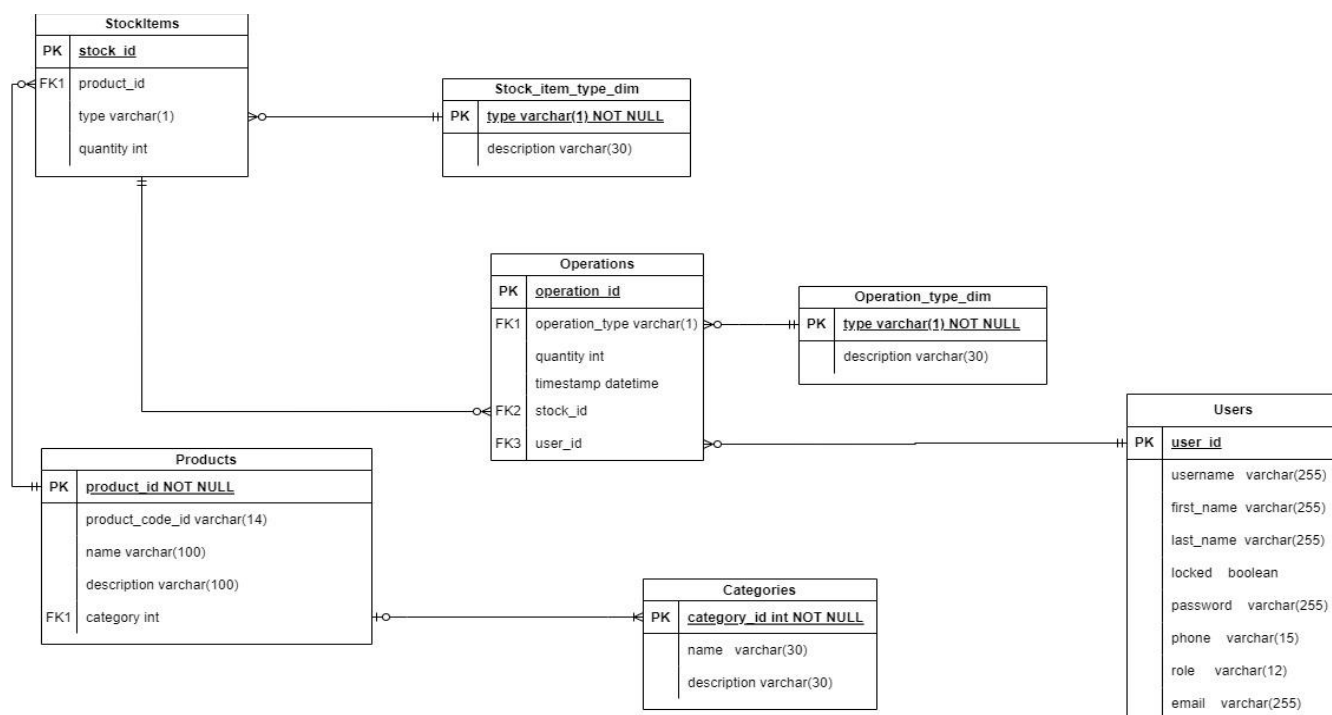


Рисунок 3.3 – Фізична модель бази даних web-орієнтованої системи підтримки складу

*Джерело: побудовано автором*

## 2.3 Моделювання варіантів використання системи

Наступним кроком необхідно було створити діаграму варіантів використання. Діаграма Use Case є графічним інструментом моделювання, який використовується для відображення функціональності системи та взаємодії між системою та її зовнішнім середовищем, таким як користувачі або інші системи [34]. Діаграма варіантів використання допомагає ідентифікувати, класифікувати та візуалізувати різні можливості використання системи.

Для Web-орієнтованої системи підтримки складу виділено наступні варіанти використання:

- Перегляд стану запасів
- Додавання нового товару
- Видалення товару
- Видача накладної
- Авторизація
- Фільтрація товарів
- Додавання категорії товарів
- Видалення категорії товарів
- Додавання нових користувачів у систему.

Акторами Web-орієнтованої системи підтримки складу є працівник складу та Адміністратор. Тільки Адміністратор може додавати нових користувачів у систему.

Діаграма варіантів використання Web-орієнтованої системи підтримки складу зображена на рисунку 3.4.

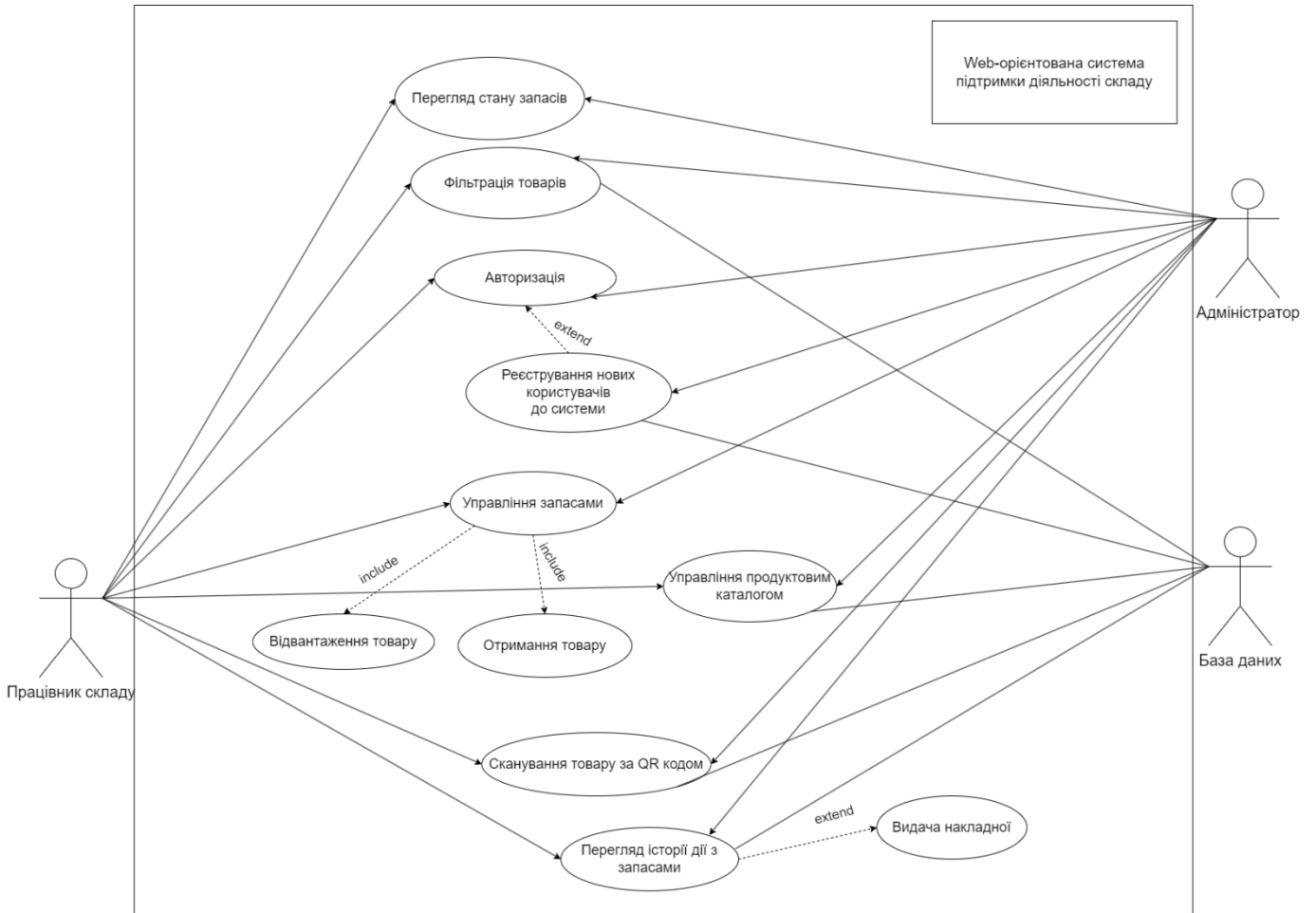


Рисунок 3.4 – Діаграма варіантів використання

*Джерело: побудовано автором*

## 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 4.1 Проектування структури Web-орієнтованої системи

Опис структури (рис.4.1) web-орієнтованої системи є важливим для чіткого розуміння взаємодій між компонентами, що сприяє ефективній розробці та підтримці. Це дозволяє визначити ключові модулі та подальший розвиток системи. Крім того, документація структури є важливим ресурсом для розробників та технічних спеціалістів.

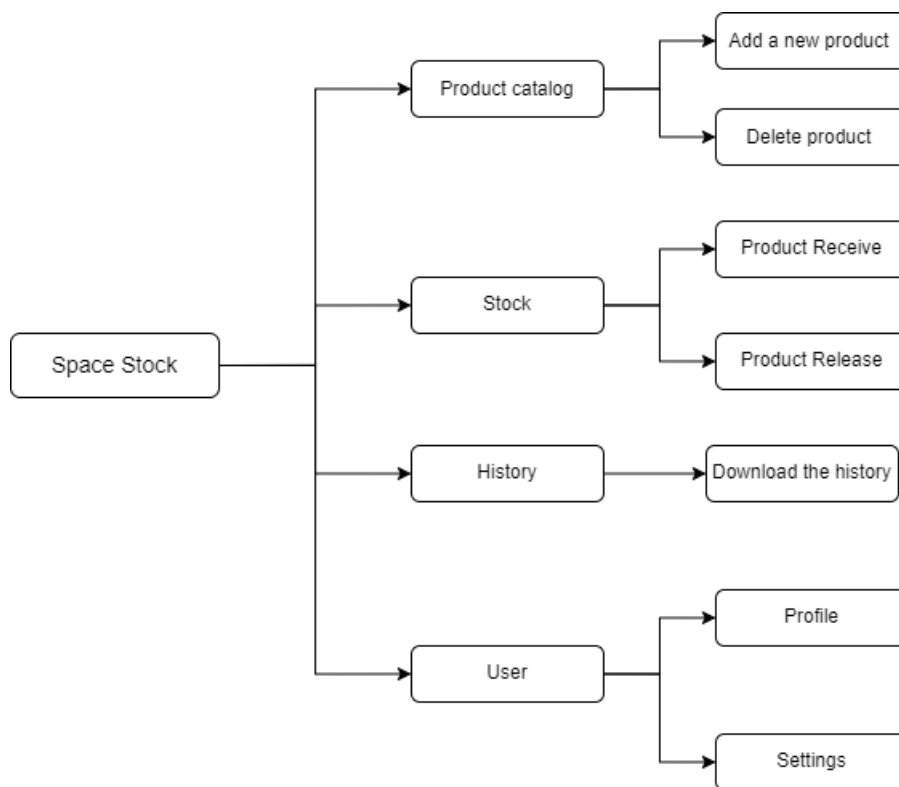


Рисунок 4.1 – Структура Web-орієнтованої системи

*Джерело: побудовано автором*

## 4.2 Програмна реалізація

Програмна реалізація web-орієнтованої системи відбувається у середовищі IntelliJ (рис.4.2)

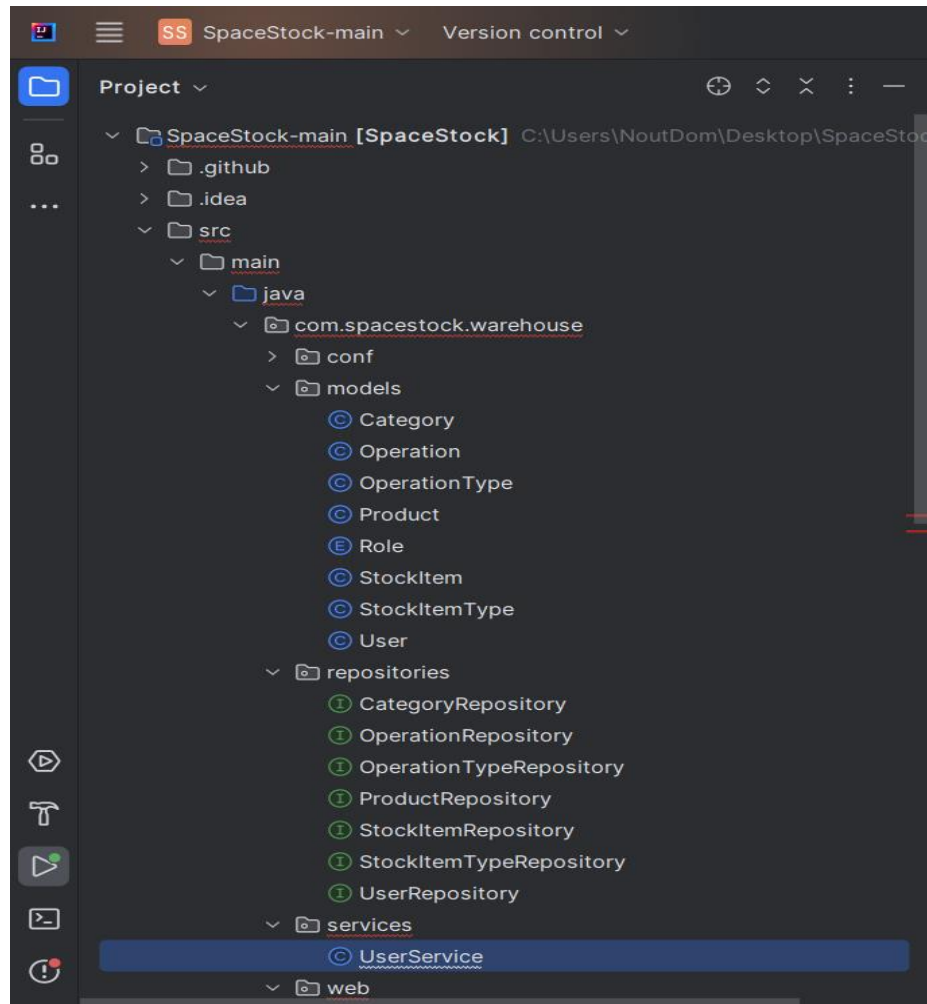


Рисунок 4.2 – Модуль Space Stock

*Джерело: побудовано автором (знімок з екрану)*

Як було зазначено раніше для реалізації системи було обрано фреймворк Spring, а в якості архітектурного підходу – Model View Controller (MVC), тож варто розглянути роботу ключової функції системи, а саме додавання продуктів у залишок (stock), з технічної точки зору.

Додавання продукту в залишок починається зі сторінки з формою Product Receipt (рис.4.3), яка знаходиться за даним шляхом /stock/addToStock і може опціонально містити параметр code.

**Product Receipt**

---

**Product**

Product Code:

**Find Product by Code**

Electronics

Product Name:

Description:

---

**Stock**

Single Product

Quantity:

**Add to Stock**

Рисунок 4.3 – Сторінка додавання в залишок

*Джерело: побудовано автором (знімок з екрану)*

Якщо даний параметр відсутній то користувачу буде запропоновано ввести його і натиснути кнопку Find Product by Code, після чого в разі наявності продукт в каталозі поля буде заповнено автоматично.

Кожен шлях системи має свій мапінг в контролері з відповідним методом POST, GET або іншим. Код метода контролера відповідального за відображення наведено на рисунку 4.4.

```

@GetMapping("/stock/addToStock")
public String addToStock(@AuthenticationPrincipal User user,
                        @RequestParam(required = false) String code,
                        Model model) {
    List<Category> allCategories = categoryRepository.findAll();
    List<StockItemType> allStockItemTypes = stockItemTypeRepository.findAll();
    Product product = productRepository.findByCode(code).orElse( other: null);

    if (product != null ) {
        model.addAttribute( s: "product", product);
    }
    model.addAttribute( s: "categories", allCategories);
    model.addAttribute( s: "stockItemTypes", allStockItemTypes);
    model.addAttribute( s: "code", code);

    return "addToStock";
}

```

Рисунок 4.4 – Код контролера сторінки додавання продукту

*Джерело: побудовано автором (знімок з екрану)*

Варто звернути увагу, що для коректного відображення категорій продуктів та типів одиниць в залишку необхідно додати їх в модель. Якщо в контролер надійшов параметр code, то система знаходить продукт і відповідно передає його в модель, щоб автоматично заповнити поля в шаблоні з формою Freemarker (рис. 4.5).



```

<form action="/stock/addToStock" method="post" name="addToStock"><br>
  <#if product??>
    <input type="hidden" name="category" value="{product.category.id}">
    <input type="text" name="categoryName" class="form-control" value="{product.category.name}" readonly>
  <#else>
    <select name="category" class="form-control" required>
      <#list categories as category>
        <option value="{category.id}">{category.name}</option>
      </#list>
    </select>
  </#if><br>
  <input type="text" name="name" placeholder="Product Name: " class="form-control" required maxlength="255"
    <#if product??> value="{product.name}" readonly </#if>><br>
  <input type="text" name="description" placeholder="Description: " class="form-control"
    <#if product??> value="{product.description}" readonly </#if>><br>
  <#if code??>
    <input type="text" name="code" class="form-control" value="{code}" readonly>
  </#if>
  <hr>
  <h3> Stock</h3>
  <select name="stockItemType" class="form-control" >
    <#list stockItemTypes as stockItemType>
      <option value="{stockItemType.type}">{stockItemType.description}</option>
    </#list>
  </select><br>
  <input type="number" name="quantity" placeholder="Quantity: " class="form-control"><br>
  <button type="submit" class="btn btn-success">Add to Stock</button>
</form>

```

Рисунок 4.5 – Шаблон з формою Freemarker

*Джерело: побудовано автором (знімок з екрану)*

Варто звернути увагу, що значення полів в формі заповнюються лише в тому випадку, коли було передано необхідний атрибут в модель, в даному випадку product.

Після заповнення всіх необхідних полей користувач натискає кнопку Add to Stock і тим самим ініціює виклик метода контролера POST для шляху /stock/addToStock (рис. 4.6, рис.4.7, рис.4.8).

```

@PostMapping("/stock/addToStock")
public String addToStock(
    @AuthenticationPrincipal User user,
    @RequestParam String code,
    @RequestParam String name,
    @RequestParam String description,
    @RequestParam long category,
    @RequestParam String stockItemType,
    @RequestParam long quantity,
    RedirectAttributes redirectAttributes
) {

```

Рисунок 4.6 – Код POST методу контролера додавання в залишок

*Джерело: побудовано автором (знімок з екрану)*

```

Product product = productRepository.findByCode(code).orElse( other: null);

if (product == null) {
    product = new Product();
    product.setName(name);
    product.setDescription(description);
    product.setCategory(categoryRepository.getById(category));
    product.setCode(code);
    productRepository.save(product);
}

StockItem stockItem = stockItemRepository.findByProductCodeAndTypeType(code, stockItemType).orElse( other: null);

if (stockItem == null) {
    stockItem = new StockItem();
    stockItem.setType(stockItemTypeRepository.findById(stockItemType).orElse( other: null));
    stockItem.setProduct(product);
    stockItem.setQuantity(quantity);
    stockItemRepository.save(stockItem);
} else {
    stockItem.setQuantity(stockItem.getQuantity() + quantity);
    stockItemRepository.save(stockItem);
}

Operation operation = new Operation();
operation.setStockItem(stockItem);
operation.setQuantity(quantity);
operation.setTimestamp(Timestamp.valueOf(LocalDate.now()));

```

Рисунок 4.7 – Продовження коду POST методу контролера додавання в залишок

*Джерело: побудовано автором (знімок з екрану)*

```

operation.setOperationType(operationTypeRepository.findById("A").orElse( other: null));
operation.setUser(user);
operationRepository.save(operation);

redirectAttributes.addFlashAttribute( s: "alertSuccess", o: quantity
    + " '" + code + "' " +
    "products added to stock");
return "redirect:/stock";
}

```

Рисунок 4.8 – Продовження коду POST методу контролера додавання в залишок

*Джерело: побудовано автором (знімок з екрану)*

В даному методі є безліч атрибутів, які є відображенням значень в полях форми. Спочатку виконується пошук продукту за вказаним кодом, якщо даний продукт ще не існує в базі, то спочатку створюється новий продукт і зберігається в базі даних. Після роботи з продуктом система перевіряє базу даних на наявність вже існуючого stock item по коду та типу (single product або pallet of products) обраному в формі. Якщо співпадінь не знайдено, то створюється нова одиниця залишку з вказаною кількістю, а в іншому випадку система додає вказану кількість одиниць до існуючого айтему.

Після проведення всіх маніпуляцій з продуктом та сток айтемом система додає необхідний запис в історію операцій з вказанням типу операції A (adding), поточним часом та користувачем, який виконав дану операцію. Наостанок система перенаправляє користувача на сторінку /stock та додає повідомлення про успішність додавання вказаної кількості певного продукту до залишку (рис.3.9, рис.4.10).



24 'PRD009' products added to stock

Рисунок 4.9 – Повідомлення про успішне додавання продукту до залишку

*Джерело: побудовано автором (знімок з екрану)*

```
<#if alertSuccess??>
  <div class="alert alert-success" role="alert">${alertSuccess}</div>
</#if>
```

Рисунок 4.10 – Блок ftl коду для відображення успішності виконання певної операції

*Джерело: побудовано автором (знімок з екрану)*

### 4.3 Використання Web-орієнтованої системи зі сторони працівника складу

При авторизації до системи (рис.4.11 ), працівник складу потрапляє на домашню сторінку (рис. 4.12 ).

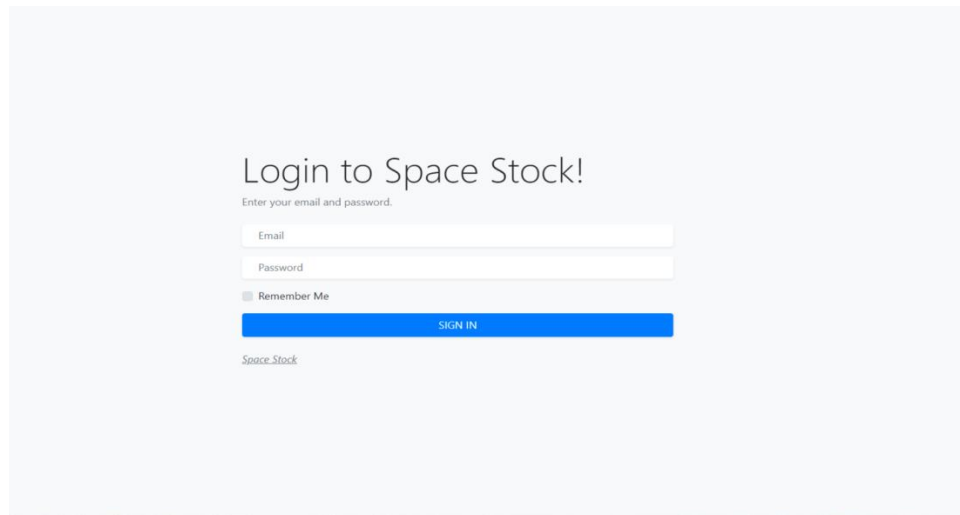


Рисунок 4.11 – Сторінка авторизації до системи

*Джерело: побудовано автором (знімок з екрану)*

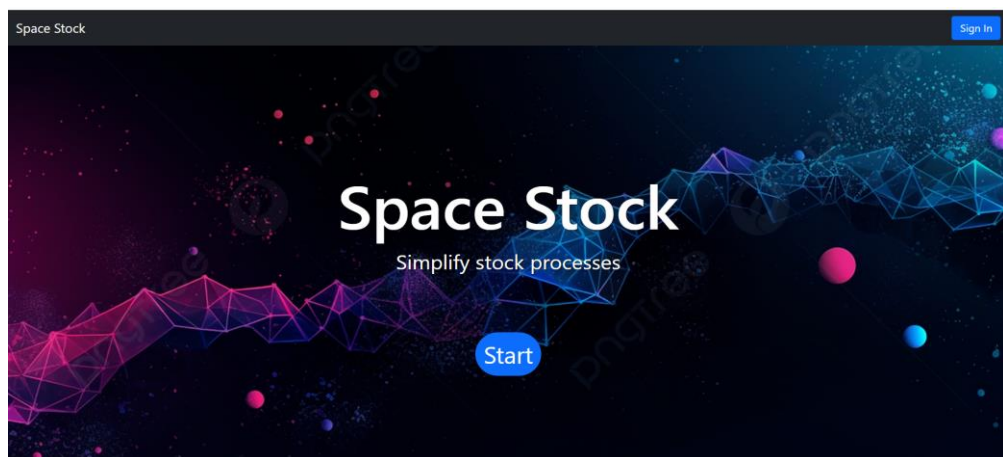
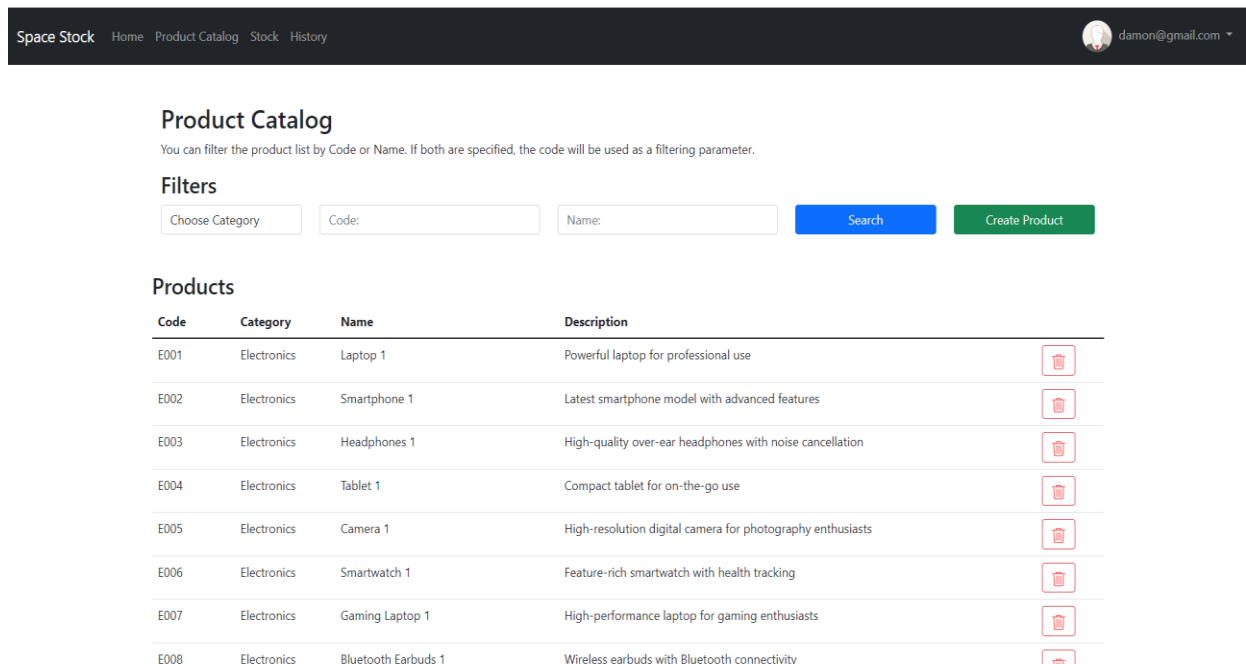


Рисунок 4.12 – Домашня сторінка Web-орієнтованої системи підтримки складу  
Space Stock

*Джерело: побудовано автором (знімок з екрану)*

Працівник складу має можливість перейти до сторінки Продуктовий каталог – «Product Catalog» (рис.4.13). Користувач може відфільтрувати продукти в каталозі за категорією (рис.4.14) , за кодом (рис.4.15) або назвою продукту (рис.4.16). Якщо вказані обидва параметри (код і назва продукту) , то фільтруватися буде за параметром code.



**Space Stock** Home Product Catalog Stock History damon@gmail.com

### Product Catalog

You can filter the product list by Code or Name. If both are specified, the code will be used as a filtering parameter.

**Filters**

Choose Category Code: Name: Search Create Product

#### Products









Code	Category	Name	Description	
E001	Electronics	Laptop 1	Powerful laptop for professional use	
E002	Electronics	Smartphone 1	Latest smartphone model with advanced features	
E003	Electronics	Headphones 1	High-quality over-ear headphones with noise cancellation	
E004	Electronics	Tablet 1	Compact tablet for on-the-go use	
E005	Electronics	Camera 1	High-resolution digital camera for photography enthusiasts	
E006	Electronics	Smartwatch 1	Feature-rich smartwatch with health tracking	
E007	Electronics	Gaming Laptop 1	High-performance laptop for gaming enthusiasts	
E008	Electronics	Bluetooth Earbuds 1	Wireless earbuds with Bluetooth connectivity	

Рисунок 4.13 – Сторінка «Product Catalog»

*Джерело: побудовано автором (знімок з екрану)*

Space Stock Home Product Catalog Stock History damon@gmail.com

## Product Catalog

You can filter the product list by Code or Name. If both are specified, the code will be used as a filtering parameter.

**Filters**

**Products**

Code	Category	Name	Description	
C001	Clothing	T-Shirt 1	Comfortable cotton t-shirt in various colors	
C002	Clothing	Jeans 1	Classic denim jeans for everyday wear	
C003	Clothing	Sneakers 1	Stylish sneakers for casual outings	
C004	Clothing	Dress 1	Elegant dress for special occasions	
C005	Clothing	Jacket 1	Warm jacket for cold weather	
C006	Clothing	Sweater 1	Cozy sweater for colder weather	
C007	Clothing	Casual Shirt 1	Stylish shirt for casual occasions	
C008	Clothing	Summer Dress 1	Light and airy dress for summer days	

Рисунок 4.14 – Фільтрація продуктів за категорією «Clothing»

*Джерело: побудовано автором (знімок з екрану)*

Space Stock Home Product Catalog Stock History damon@gmail.com

## Product Catalog

You can filter the product list by Code or Name. If both are specified, the code will be used as a filtering parameter.

**Filters**

**Products**

Code	Category	Name	Description	
B010	Books	Science Book 1	Educational book exploring scientific concepts	

Рисунок 4.15 – Фільтрація продуктів за кодом

*Джерело: побудовано автором (знімок з екрану)*

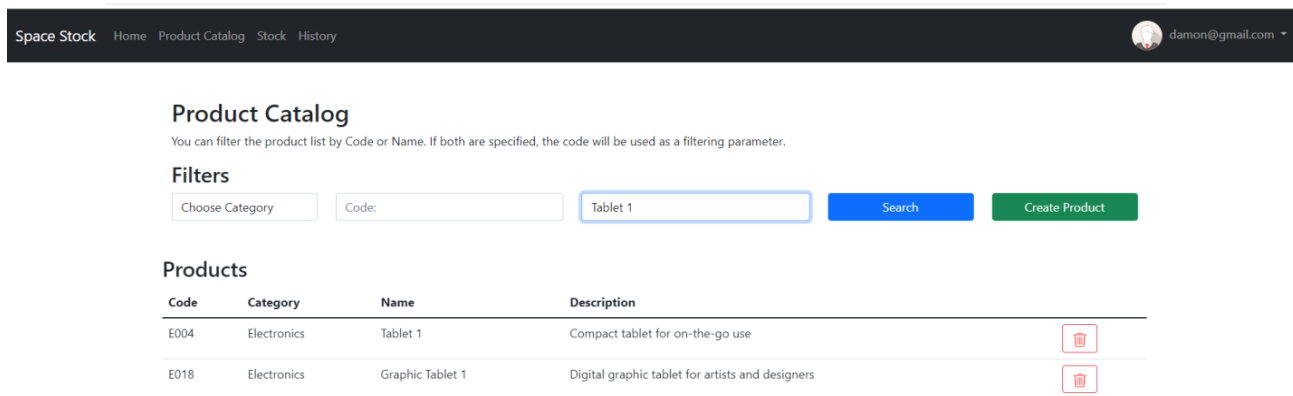


Рисунок 4.16 – Фільтрація за назвою продукту

*Джерело: побудовано автором (знімок з екрану)*

Кнопка Create product – переводить на сторінку створення продукту. Необхідно заповнити 4 поля: Name, Category, Description, Product Code. Переглянути сторінку додавання нового продукту до каталогу можна на рисунку 4.17.

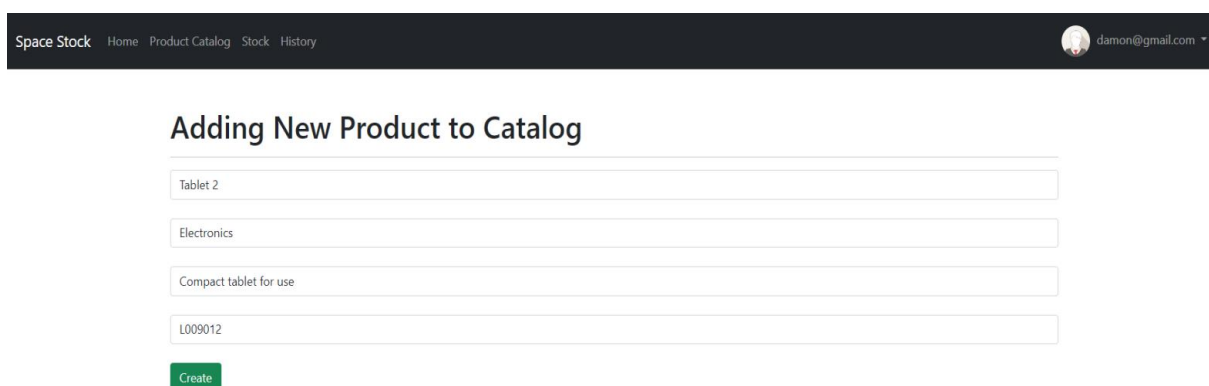


Рисунок 4.17 – Сторінка додавання нового продукту

*Джерело: побудовано автором (знімок з екрану)*

Працівник складу може також видалити продукт (рис.4.18 ), якщо його ще не додано до стоку. Якщо продукт вже додано до стоку, то на екрані з'явиться попереджувальне повідомлення, та не буде можливості видалити продукт (рис.4.19).

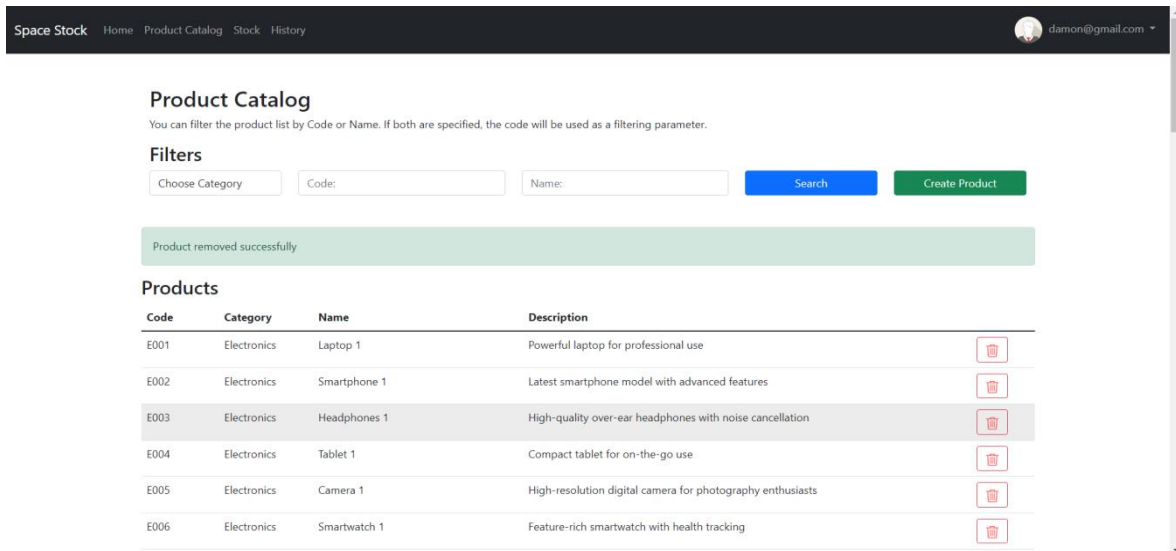


Рисунок 4.18 – Успішне видалення продукту

*Джерело: побудовано автором (знімок з екрану)*

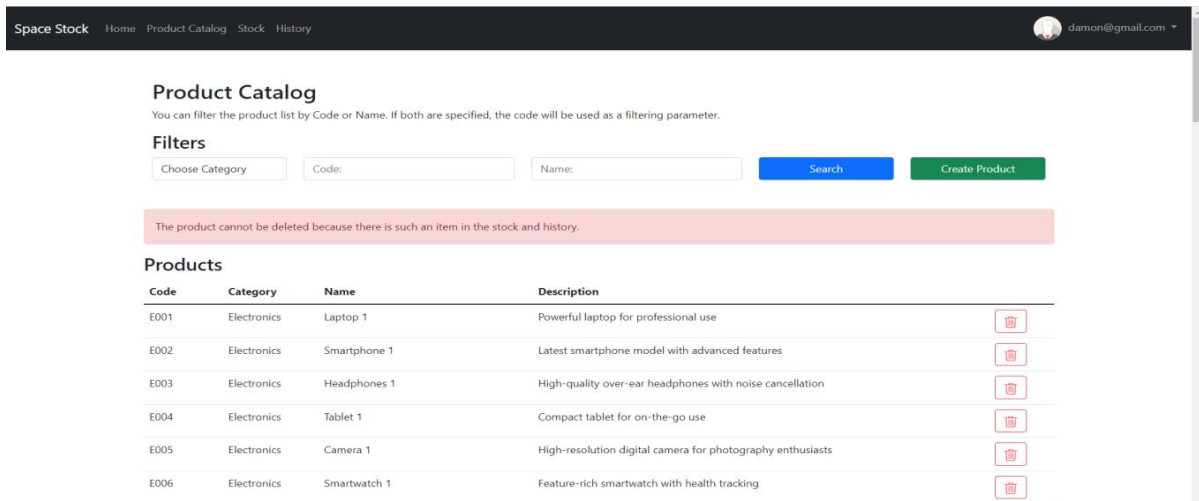


Рисунок 4.19 – Попереджувальне повідомлення, що продукт неможливо видалити, так як він наявний в стоці.

*Джерело: побудовано автором (знімок з екрану)*



На сторінці «Stock» (рис.4.20) розташовані продукти які наявні на даний момент в стоці. Користувач може відфільтрувати продукти, які наявні в стоці по параметрам Product Code та Product Name.

Щоб додати продукт до стоку користувачу потрібно натиснути на кнопку Product Receipt. Після натискання на кнопку – відкривається сторінка додавання продукту до стоку (рис.4.21).

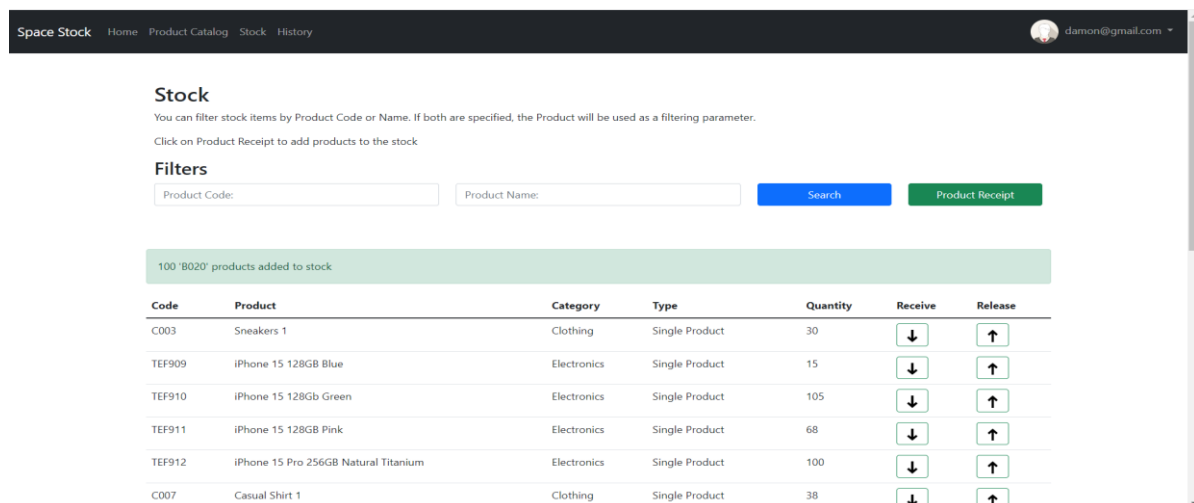


Рисунок 4.20 – Сторінка «Stock»

*Джерело: побудовано автором (знімок з екрану)*

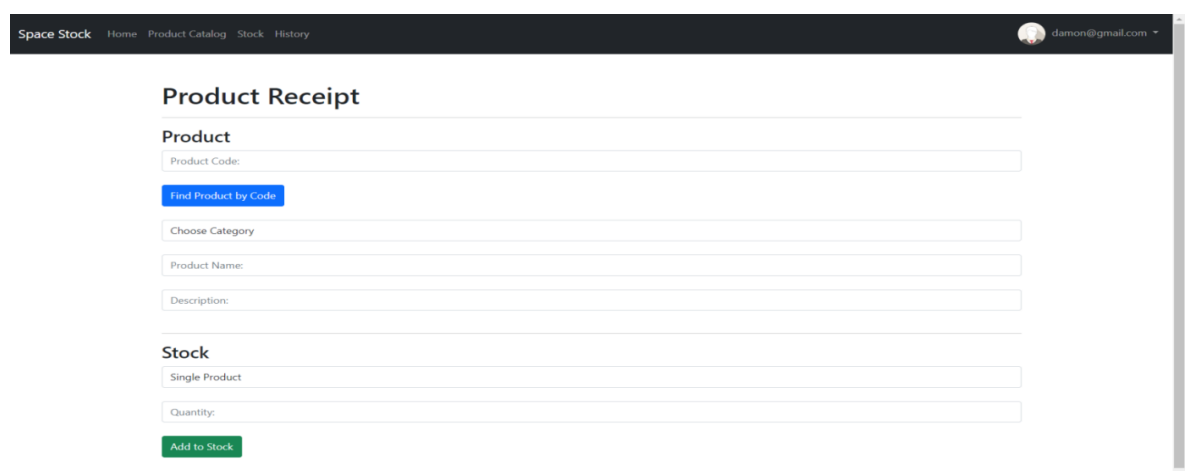
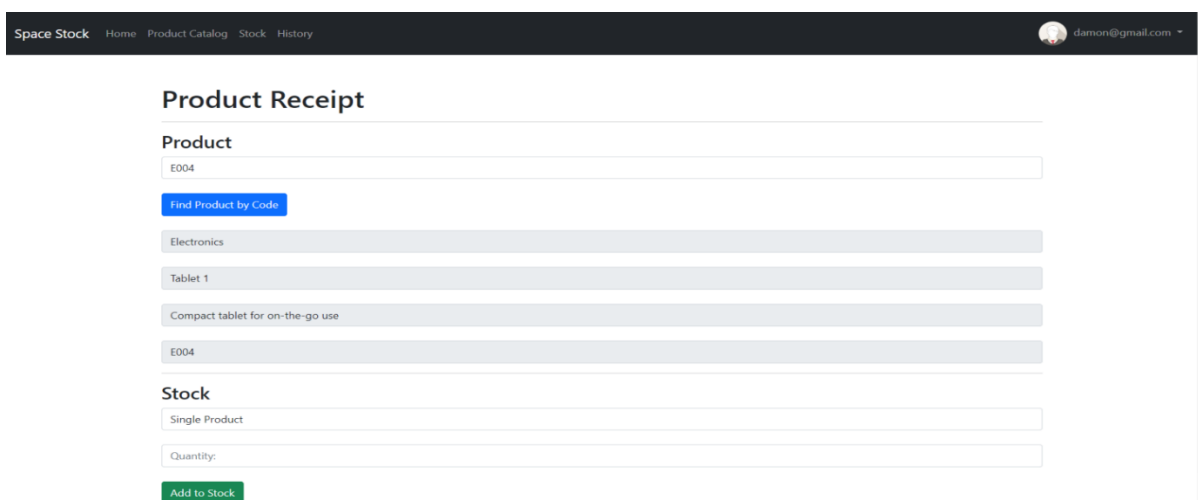


Рисунок 4.21 – Сторінка додавання продукту до стоку

*Джерело: побудовано автором (знімок з екрану)*

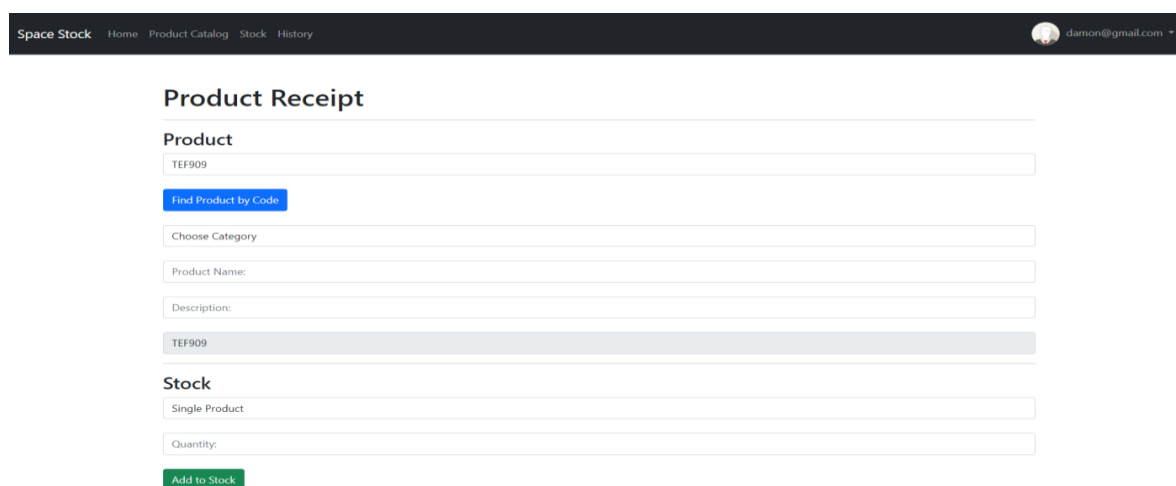
Якщо продукт вже присутній у Product Catalog, то потрібно прописати лише продуктовий код і натиснути кнопку «Find Product by code», такі поля як ім'я продукту, та його опис заповняться автоматично (рис.4.22). Якщо даного продукту немає у продуктовому каталозі, то потрібно вручну заповнити дані, які зберігаються у Product Catalog (рис.4.23).



The screenshot shows a web application interface for 'Space Stock'. The navigation bar includes 'Home', 'Product Catalog', 'Stock', and 'History'. The user is logged in as 'damon@gmail.com'. The main form is titled 'Product Receipt' and is divided into two sections: 'Product' and 'Stock'. In the 'Product' section, the 'Product' field contains 'E004'. A blue button labeled 'Find Product by Code' is visible. Below this, the 'Electronics' category, 'Tablet 1' product name, and 'Compact tablet for on-the-go use' description are automatically populated in greyed-out fields. The 'E004' code is also shown in a separate field. In the 'Stock' section, the 'Single Product' field is empty, and the 'Quantity' field is also empty. A green button labeled 'Add to Stock' is at the bottom of the form.

Рисунок 4.22 – Автоматичне заповнення даних про продукт при присутності у Product Catalog

*Джерело: побудовано автором (знімок з екрану)*



The screenshot shows the same 'Product Receipt' form as in Figure 4.22, but with manual data entry. The 'Product' field contains 'TEF909'. The blue 'Find Product by Code' button is still present. Below it, the 'Choose Category' field is empty. The 'Product Name' field contains 'TEF909' and the 'Description' field contains 'TEF909'. In the 'Stock' section, the 'Single Product' field is empty, and the 'Quantity' field is also empty. The green 'Add to Stock' button is at the bottom of the form.

Рисунок 4.23 – Додавання продукту, якого немає у Product Catalog

*Джерело: побудовано автором (знімок з екрану)*

Наступним кроком необхідно обрати опцію яким чином прийшов цей товар, як Single Product чи як Pallet of Product (рис.4.24) та вказати кількість в якій прийшов товар.



The image shows a web form titled "Stock". It features a dropdown menu with three visible options: "Single Product", "Single Product", and "Pallet of Products". The first "Single Product" option is highlighted in blue. Below the dropdown is a green button labeled "Add to Stock".

Рисунок 4.24 – Вибір опції Single Product або Pallet of Product

*Джерело: побудовано автором (знімок з екрану)*

На рисунку 4.25 можна переглянути всі введені характеристики для додавання нового продукту до стоку та продуктового каталогу. На рисунку 4.26 зображено доданий продукт на сторінці «Stock»



The image shows a web application interface for "Space Stock". The top navigation bar includes "Home", "Product Catalog", "Stock", and "History". A user profile icon for "damon@gmail.com" is in the top right. The main content area is titled "Product Receipt". It contains a "Product" field with the value "B020" and a blue "Find Product by Code" button. Below this is a list of product suggestions: "Books", "Language Learning Book 1", "Guide to learning a new language effectively", and "B020". There is also a "Stock" section with a dropdown menu set to "Single Product" and a quantity field set to "100". A green "Add to Stock" button is at the bottom.

Рисунок 4.25 – Введені характеристики для додавання нового продукту до стоку та продуктового каталогу.

*Джерело: побудовано автором (знімок з екрану)*

**Space Stock** Home Product Catalog Stock History damon@gmail.com

### Stock

You can filter stock items by Product Code or Name. If both are specified, the Product will be used as a filtering parameter.

Click on Product Receipt to add products to the stock

**Filters**

Product Code:  Product Name:  Search Product Receipt

100 'B020' products added to stock

Code	Product	Category	Type	Quantity	Receive	Release
C003	Sneakers 1	Clothing	Single Product	30	↓	↑
TEF909	iPhone 15 128GB Blue	Electronics	Single Product	15	↓	↑
TEF910	iPhone 15 128Gb Green	Electronics	Single Product	105	↓	↑
TEF911	iPhone 15 128GB Pink	Electronics	Single Product	68	↓	↑
TEF912	iPhone 15 Pro 256GB Natural Titanium	Electronics	Single Product	100	↓	↑
C007	Casual Shirt 1	Clothing	Single Product	38	↓	↑

Рисунок 4.26 – Доданий продукт на сторінці «Stock».

*Джерело: побудовано автором (знімок з екрану)*

Вже додані до стоку продукти є можливість отримати більшу кількість (Receive) та випустити зі складу (Release). Якщо до стоку надійшла більша кількість певного продукту то необхідно натиснути на кнопку Receive, яка перенаправить на сторінку додавання продукту. На даній сторінці необхідно ввести скільки саме цього продукту надійшло до стоку.

Якщо якусь кількість певного продукту випустили зі стоку – потрібно натиснути кнопку Release. Дана кнопка перенаправить на сторінку Product Release, де необхідно вказати кількість товарів які випускаються зі стоку (рис. 4.27).

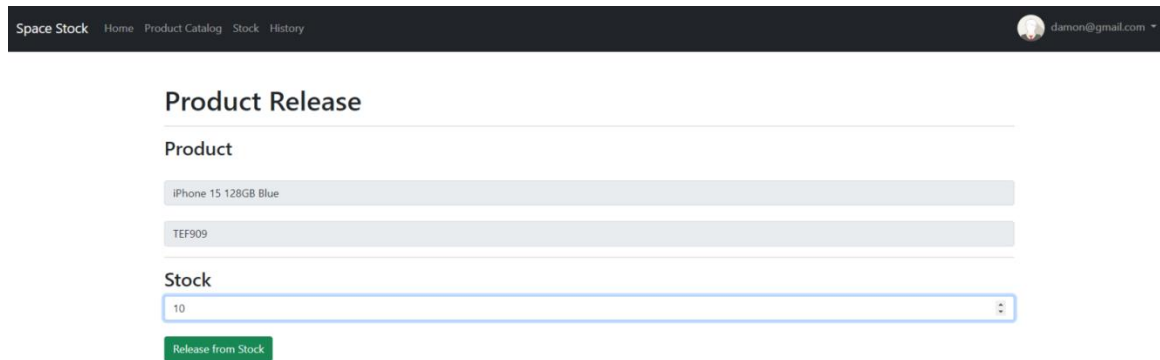


Рисунок 4.27 – Product Release сторінка

*Джерело: побудовано автором (знімок з екрану)*

Всі операції з продуктами у стокі зберігаються на сторінці History (рис. 4.28). При переході на сторінку History користувач може переглянути всю історію дій. Також користувач може по кожній операції відкрити накладну, щоб переглянути які дії були виконані над тим чи іншим продуктом. Накладна зберігається у Word файлі, який пізніше можливо роздрукувати для звітності. Накладна може бути на відвантаження товару (рис.4.29) та надходження (рис.4.30).

Date and Time	Action	Quantity	Product	Code	User	Document
Nov 26, 2023, 3:00:09 PM	Item adding	122	Sneakers 1	C003	damon@gmail.com	
Nov 26, 2023, 3:00:17 PM	Item adding	6	Sneakers 1	C003	damon@gmail.com	
Nov 26, 2023, 3:00:28 PM	Item release	100	Sneakers 1	C003	damon@gmail.com	
Nov 26, 2023, 3:01:28 PM	Item adding	100	sjndkjs	TEST003	damon@gmail.com	
Nov 26, 2023, 5:11:52 PM	Item adding	2	Sneakers 1	C003	damon@gmail.com	
Nov 26, 2023, 5:11:58 PM	Item adding	5	Sneakers 1	C003	damon@gmail.com	
Nov 26, 2023, 6:39:12 PM	Item release	5	Sneakers 1	C003	damon@gmail.com	

Рисунок 4.28 – Сторінка History

*Джерело: побудовано автором (знімок з екрану)*

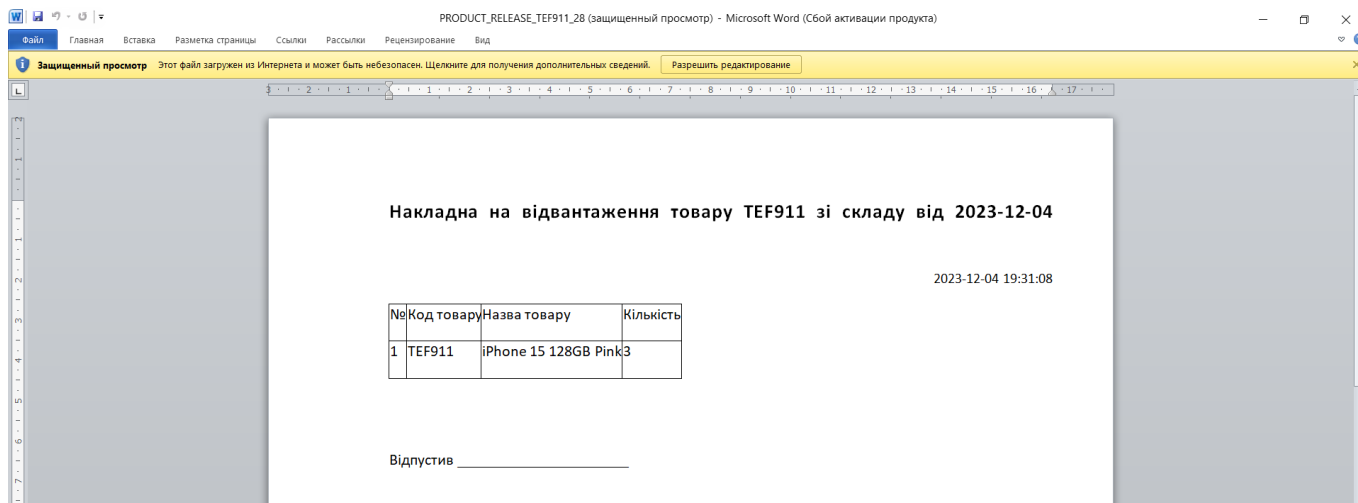


Рисунок 4.29 – Накладна на відвантаження товару

*Джерело: побудовано автором (знімок з екрану)*

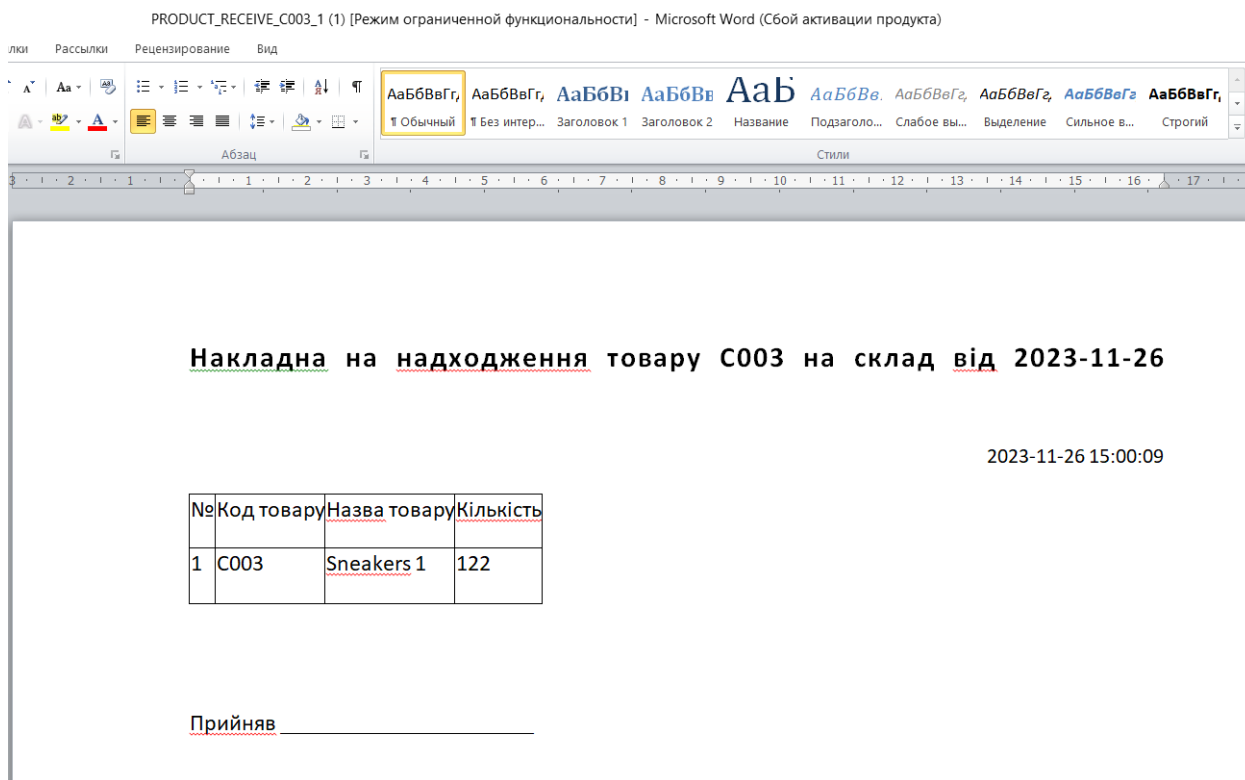


Рисунок 4.30 – Накладна на відвантаження товару

*Джерело: побудовано автором (знімок з екрану)*

## 4.4 Використання Web-орієнтованої системи зі сторони адміністратора

Адміністратору дозволено використовувати ті самі функції що і працівнику складу. Адміністратор може управляти обліковими записами користувачів (працівників складу) в системі. Важливо відзначити, що лише адміністратор має права на додавання нових користувачів, отже єдиний спосіб потрапити до системи – щоб адмін створив обліковий запис.

При авторизації користувача як адміністратора, додається у меню сторінки управління обліковими записами – «Users». Переглянути сторінку «Users» можна на рисунку 4.31.

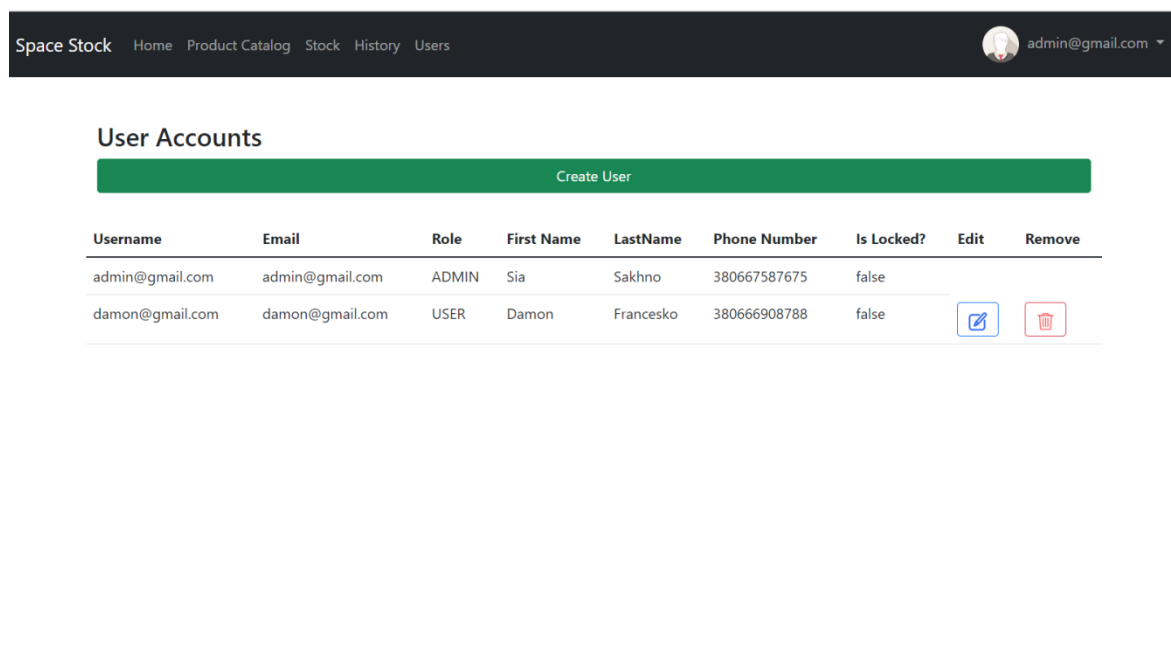


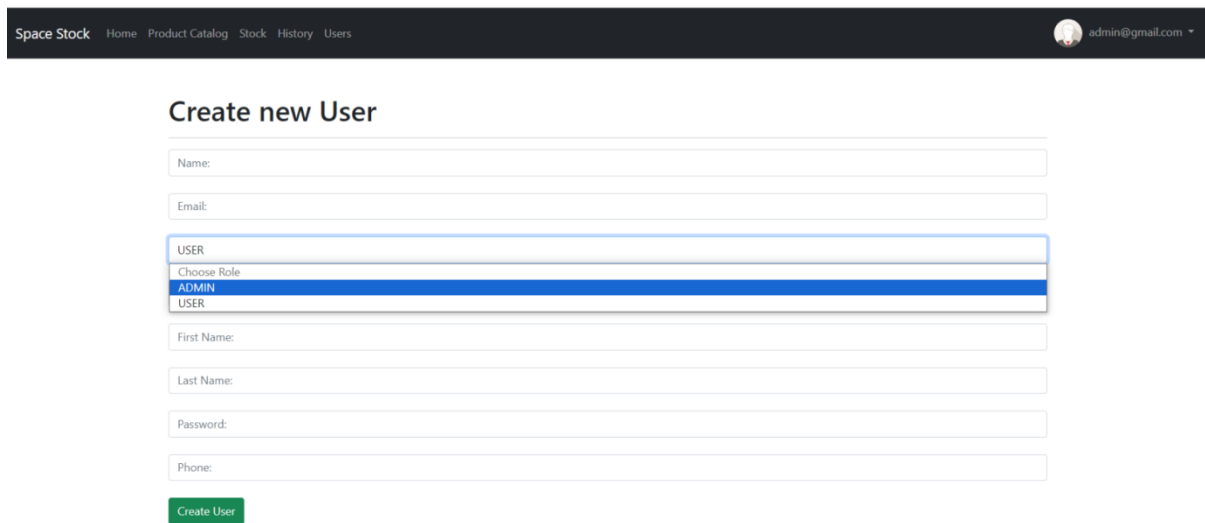
Рисунок 4.31 – сторінка «Users»

*Джерело: побудовано автором (знімок з екрану)*

При натисканні на кнопку «Create user» адміністратор потрапляє на сторінку створення нового користувача (рис.4.32). Сторінка створення нового облікового запису містить наступні поля:

- Name – нікнейм користувача
- Email – електронна пошта працівника складу

- Choose role – роль яка буде присвоєна користувачу: адміністратор (Admin) або працівник складу (User).
- First Name – ім'я користувача.
- Last Name – прізвище користувача.
- Password – пароль для облікового запису, який працівник складу надалі може самостійно змінити.
- Phone – номер телефону користувача.



Space Stock Home Product Catalog Stock History Users admin@gmail.com

### Create new User

Name:

Email:

Choose Role

- USER
- ADMIN
- USER

First Name:

Last Name:

Password:

Phone:

Create User

Рисунок 4.32 – Сторінка створення нового користувача

*Джерело: побудовано автором (знімок з екрану)*

Роль адміністратора також дозволяє редагувати та видаляти вже створені облікові записи. Переглянути сторінку редагування облікового запису можна на рисунку 4.33 .



Space Stock Home Product Catalog Stock History Users admin

### Edit User

Klaus

nicklaus@gmail.com

Nicklaus

Thuis

+380987909089

Save User

Рисунок 4.33 – Сторінка редагування адміністратором облікового запису працівника складу.

*Джерело: побудовано автором (знімок з екрану)*

Після видалення адміністратором облікового запису працівника складу, можна побачити на екрані повідомлення про те, що видалення пройшло успішно (рис.4.34).

Space Stock Home Product Catalog Stock History Users admin@gmail.com

### User Accounts

Create User

User test@gmail.com removed successfully

Username	Email	Role	First Name	LastName	Phone Number	Is Locked?	Edit	Remove
admin@gmail.com	admin@gmail.com	ADMIN	Sia	Sakhno	380667587675	false		
damon@gmail.com	damon@gmail.com	USER	Damon	Francesko	380666908788	false		

Рисунок 4.34 – Видалення адміністратором облікового запису працівника складу.

*Джерело: побудовано автором (знімок з екрану)*

## 4.5 Використання Web-орієнтованої системи з мобільного пристрою

Web-орієнтована система також доступна для використання з мобільного пристрою (рис.4.35). При отриманні товару можливо відсканувати QR код. Після сканування QR коду, користувач переадресовується на сторінку Product Receipt (рис.4.36), для заповнення інформації про продукт та прибувшу кількість. Результат додавання можна переглянути на сторінці Stocks та сторінці History (рис.4.37).

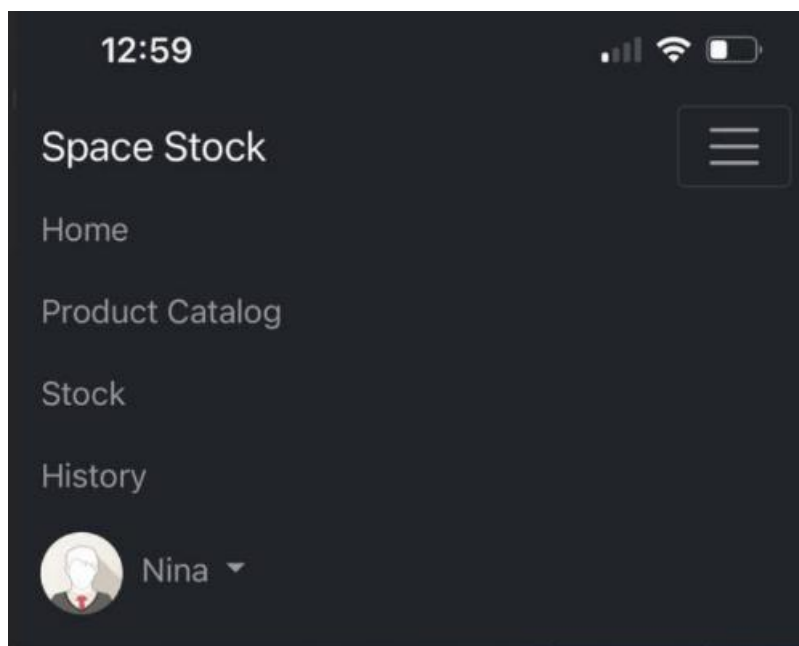


Рисунок 4.35 – Головне меню web-орієнтованої системи з мобільного пристрою.

*Джерело: побудовано автором (знімок з екрану)*

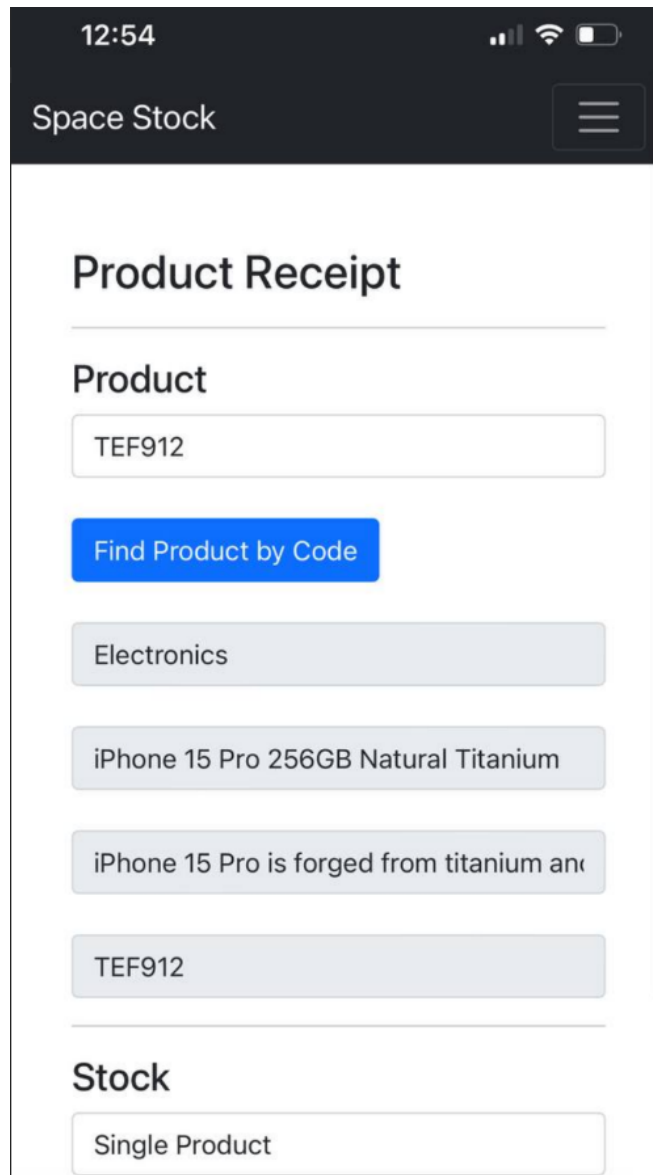





Рисунок 4.36 – Сторінка Product Receipt з мобільного пристрою.

*Джерело: побудовано автором (знімок з екрану)*

13:22   

PM









Nov 26, 2023, 10:42:40 PM	Item adding	3	Poetry Collection 1	B011	damon@gmail.com	
Nov 26, 2023, 10:43:11 PM	Item adding	88	Laptop 1	E001	damon@gmail.com	
Nov 26, 2023, 10:43:44 PM	Item adding	190	Bluetooth Earbuds 1	E008	damon@gmail.com	
Nov 26, 2023, 10:44:25 PM	Item adding	10	4K TV 1	E009	damon@gmail.com	
Nov 26, 2023, 10:44:39 PM	Item adding	59	Fitness Tracker 1	E012	damon@gmail.com	
Nov 26, 2023, 10:46:21 PM	Item adding	110	Wireless Mouse 1	E015	damon@gmail.com	
Nov 26, 2023, 10:46:44 PM	Item adding	55	3D Printer 1	E020	damon@gmail.com	
Nov 26, 2023, 10:46:52 PM	Item release	4	sjndkjs	TEST003	damon@gmail.com	
Nov 26, 2023, 10:50:07 PM	Item adding	100	Language Learning Book 1	B020	damon@gmail.com	
Dec 4, 2023, 7:31:08 PM	Item release	3	iPhone 15 128GB Pink	TEF911	ninel@gmail.com	
Dec 5, 2023, 12:54:56 PM	Item adding	23	iPhone 15 Pro 256GB Natural Titanium	TEF912	ninel@gmail.com	

Рисунок 4.37 – Сторінка History з мобільного пристрою після отримання продукту.

*Джерело: побудовано автором (знімок з екрану)*

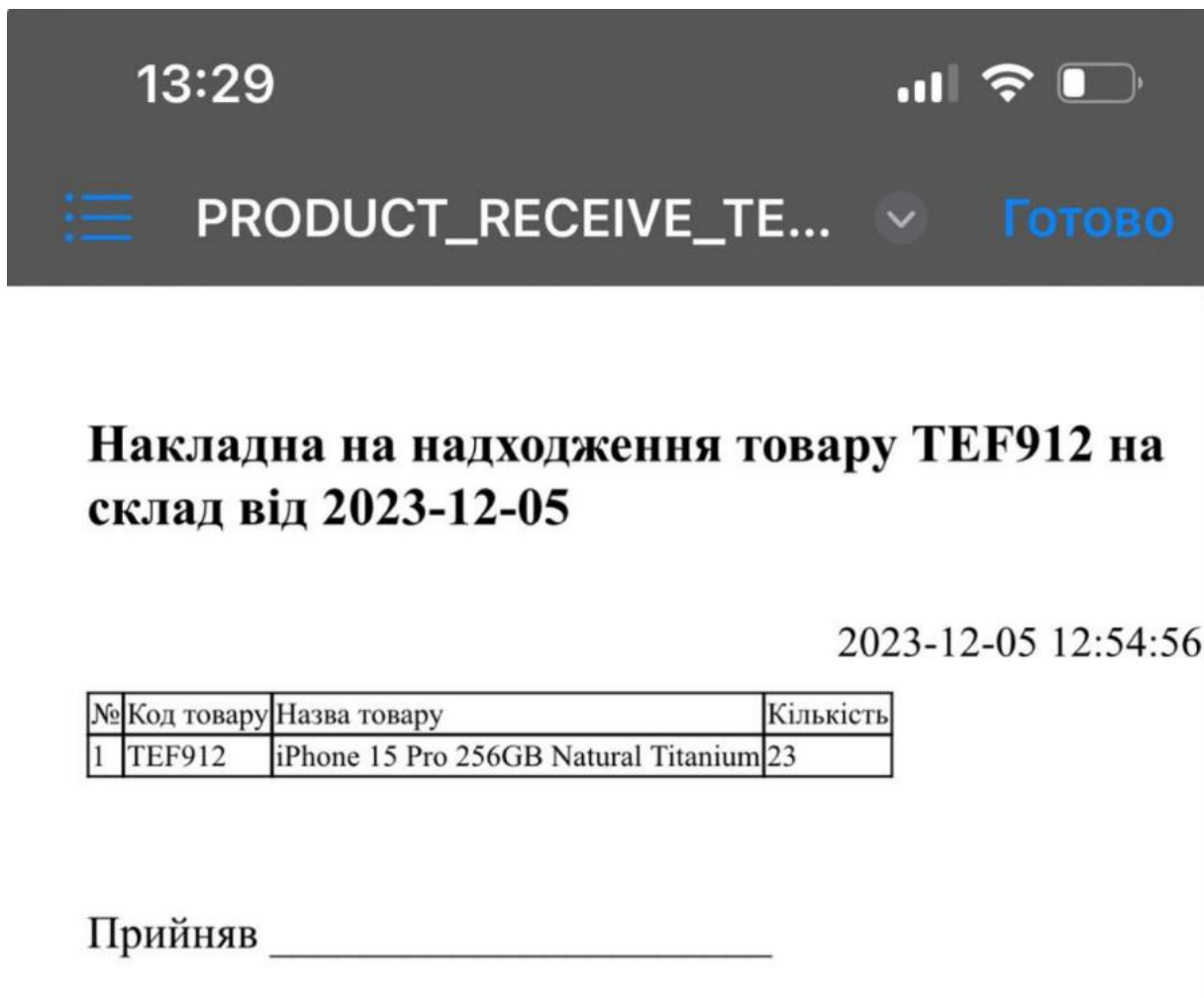


Рисунок 4.38 – Накладна завантажена на мобільний пристрій.

*Джерело: побудовано автором (знімок з екрану)*

## ВИСНОВОК

В ході виконання магістерської роботи було виявлено і проаналізовано актуальні питання управління та підтримки діяльності складів. Огляд літературних джерел та існуючих систем дав можливість визначити ключові аспекти, що впливають на ефективність складського управління.

В ході виконання роботи було використано сучасні методи дослідження та моделювання. Результатом цього стало створення структурно-функціональної моделі системи та проектування відповідної бази даних. Вивчення варіантів використання системи дозволило визначити оптимальні шляхи впровадження розробленої системи в робочий процес. Інструментами реалізації Front-end частини було обрано HTML, CSS, Bootstrap. Для Back-end частини було обрано Java, Spring MVC та Spring Boot. PostgreSQL обрано як систему управління базою даних.

Були реалізовані наступний функціонал: розподіл прав доступу для різних користувачів, адаптивність до різних пристроїв, перегляд інформації у web-системі, управління запасами на складі, фільтрація та сортування запасів для швидкого пошуку у системі, генерування накладних щодо стану запасів, управління особистим профілем у web-системі, надавання можливості сканування товарів які надійшли на склад, за QR кодом.

Отже, розроблена web-орієнтована система підтримки діяльності складу успішно втілила функціональні вимоги, визначені в ході аналізу. Система демонструє гнучкість та адаптивність, забезпечуючи користувачам можливість зручного управління запасами, роботи з каталогом товарів та ведення обліку товарів на складі.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. В. І. Ковшик (2016). Інформаційні технології в контексті управління логістичними витратами промислових підприємств. Режим доступу до ресурсу: <https://core.ac.uk/download/pdf/162868871.pdf>
2. Кириченко А. І (2013). Проблематика застосування інформаційних технологій в управлінні процесами доставки вантажу / А. І. Кириченко // Проблеми транспорту. С. 17–27.
3. Ian J.Lloyd. Information Technology Law. 2019, с.176-179
4. C.Sophie Lee. Modeling the business value of information technology. с.191-210
5. Функціональне призначення та класифікація складів в логістиці. Режим доступу до ресурсу: <https://wareteka.com.ua/uk/blog/klasyfikaciya-skladiv-v-logistyci/>
6. Agnieszka A.Tubis, Junu Rohman. Gunn (2023). Intelligent Warehouse in Industry. Режим доступу до ресурсу: <https://www.mdpi.com/1424-8220/23/8/4105>
7. 10 Types of Warehouses: What Are They and How to Choose the Right One. Режим доступу до ресурсу: <https://www.shipcalm.com/blog/warehouse-types/>
8. Types of Warehouses & Their Functions Режим доступу до ресурсу: <https://dovetail.co.za/types-of-warehouses-their-functions/>
9. René B. M. De Koster, Andrew L. Johnson, Andrew L. Johnson (2017). Warehouse design and management
10. Andriani A, Andry J (2023). Designing a Web-Based Inventory Application at General Steel Supplier Using Extreme Programming Method. CogITo Smart Journal (2023) с. 15-27
11. Warehousing (2023). How to Effectively Manage Trans-Loading Warehousing. Режим доступу до ресурсу: <https://www.18wheelslogistics.com/blog/how-to-effectively-manage-trans-loading-warehousing>
12. Warehousing (2022) How Efficient Warehousing Systems Support Just-in-Time Inventory Management.

13. Ramūnas Palšaitis, Kristina Čižiūnienė, Kristina Vaičiūtė (2017). Improvement of Warehouse Operations Management by Considering Competencies of Human Resources Volume 187, 2017, Pages 604-613.
14. Ma Junhong. Research on Intelligent Warehouse Management System Based on RFID (2020). Xi'an International University.
15. P. Bodnar, Essays on Warehouse Operations (2013). Режим доступа до ресурсу: [http://pure.au.dk/portal/files/56369075/PhD\\_thesis\\_Peter\\_Bodnar.pdf](http://pure.au.dk/portal/files/56369075/PhD_thesis_Peter_Bodnar.pdf).
16. R Fauzan1, M F Shiddiq and N R Raddly (2020). The Designing of Warehouse Management Information System. Citation R Fauzan et al 2020 IOP Conf. Ser.: Mater. Sci. Eng.
17. Putri Shima, Akhmad Syakhroni, ST, M.Eng (2021). Analysis of the layout of the finished goods warehouse using the shared storage method to increase storage effectiveness in PT. NCS Logistic Link. Journal of Applied Science and Technology.
18. Andriani A, Andry J (2023). Designing a Web-Based Inventory Application at General Steel Supplier Using Extreme Programming Method. CogITo Smart Journal, c.15-27.
19. What Is an Automation System: Pros and Cons. Режим доступа до ресурсу: <https://www.rios.ai/post/automation-system-pros-and-cons>.
20. Aldona Jarašūnienė, Kristina Čižiūnienė, Audrius Čereška (2023). Research on Impact of IoT on Warehouse Management. Режим доступа до ресурсу: <https://www.mdpi.com/1424-8220/23/4/2213>.
21. TradeGecko. Режим доступа до ресурсу: <https://www.tradegecko.com/product-tour/inventory-management-system/>.
22. ShipFusion. Режим доступа до ресурсу: <https://www.shipfusion.com/>.
23. RightControl. Режим доступа до ресурсу: <https://www.losoftware.co.uk/>.
24. Marta Beznos (2022). Microservices vs monolith: Which architecture is the best choice for your business. Режим доступа до ресурсу: <https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business>. 24



25. Alexandre Ouellette (2022). What is Bootstrap: A Beginner's Guide Режим доступу до ресурсу: <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/>
26. Craig Walls, Spring in Action 5, Fifth edition : Manning Publications Co, 2019 – 380-498 с.
27. Charles Forsythe (2015). Instant FreeMarker Starter. с 57-60.
28. John C.Worsley. Practical PostgreSQL. с. 30-35.
29. John C.Worsley, Joshua D.Drake (2012). Practical PostgreSQL. с. 25-40.
30. Методологія IDEF0. Режим доступу до ресурсу: [https://stud.com.ua/87184/ekonomika/metodologiya\\_idef0](https://stud.com.ua/87184/ekonomika/metodologiya_idef0).
31. The Complete Guide To Understand IDEF Diagram. Режим доступу до ресурсу: <https://www.edrawmax.com/article/the-complete-guide-to-understand-idef-diagram.html>
32. Understanding IDEF Diagram: An In-depth Look. Режим доступу до ресурсу: <https://boardmix.com/knowledge/idef-diagram/>
33. Coronel, Carlos, Morris, Steven (2019). Database systems : design, implementation and management. с. 51-58.
34. What is Use Case Diagram? Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>.
35. CFI Team (2020). SMART Goals G. Режим доступу до ресурсу: <https://corporatefinanceinstitute.com/resources/management/smart-goal/>.
36. What is a Work Breakdown Structure (WBS). Workbreakdownstructure.Com (2020) с. 1-10.
37. Work Breakdown Structure (WBS). Режим доступу до ресурсу: <https://www.villanovau.com/articles/project-management/work-breakdown-structure/>
38. Organization Breakdown Structure (OBS). Режим доступу до ресурсу: <https://corporatefinanceinstitute.com/resources/management/smart-goal/>.

39. K K Ramachandran, K K Karthick. Gantt Chart: An Important Tool of Management (2019). International Journal of Innovative Technology and Exploring Engineering (IJITEE).
40. Masahiro Shibuya and Xuebin Chen (2021). Production Planning and Management Using Gantt Charts. Journal of Mechanics Engineering and Automation с. 68-76.
41. Brenna Schwartz (2021). The Risk Management Process in Project Management. Режим доступа до ресурсу: <https://www.projectmanager.com/blog/risk-management-process-steps>

## ДОДАТОК А

Метод SMART є способом формулювання цілей, враховуючи п'ять критеріїв: конкретність, вимірюваність, досяжність, важливість та визначеність за термінами. Кожна літера у аббревіатурі SMART представляє окремий критерій, що сприяє ефективності постановки та досягнення цілей: Specific, Measurable, Achievable, Relevant, Time bound [35]. Деталізовані результати методу SMART представлені в таблиці А1.

Таблиця А1 - Деталізований результат методу SMART

Назва	Опис
Specific	Розробити Web-орієнтовану систему підтримки складу з використанням сучасних засобів розробки програмного забезпечення
Measurable	Результатом роботи є Web-орієнтована система підтримки складу
Achievable	Для реалізації Web-орієнтованої системи необхідні знання з HTML, CSS, навички програмування на Java, фреймворку Spring та бази даних PostgreSQL.
Relevant	Розроблена Web-орієнтована система автоматизує складські процеси, сприяючи оптимізації управління запасами та підвищенню ефективності складської діяльності.
Time bound	Розробка Web-орієнтованої системи обмежена у часі – 3 місяці

## Планування змісту структури робіт IT-проекту

Work Breakdown Structure (WBS) - це інструмент планування проекту, що використовується для розбиття робіт на менші, більш керовані елементи. Основною метою є визначення всі компоненти проекту та їхні залежності [36]. Структура розподілу робіт візуально визначає обсяг на керовані фрагменти, які може зрозуміти команда проекту, оскільки кожен рівень структури розподілу робіт забезпечує подальше визначення та деталізацію [37]. Переглянути WBS діаграму можна на рисунку А1

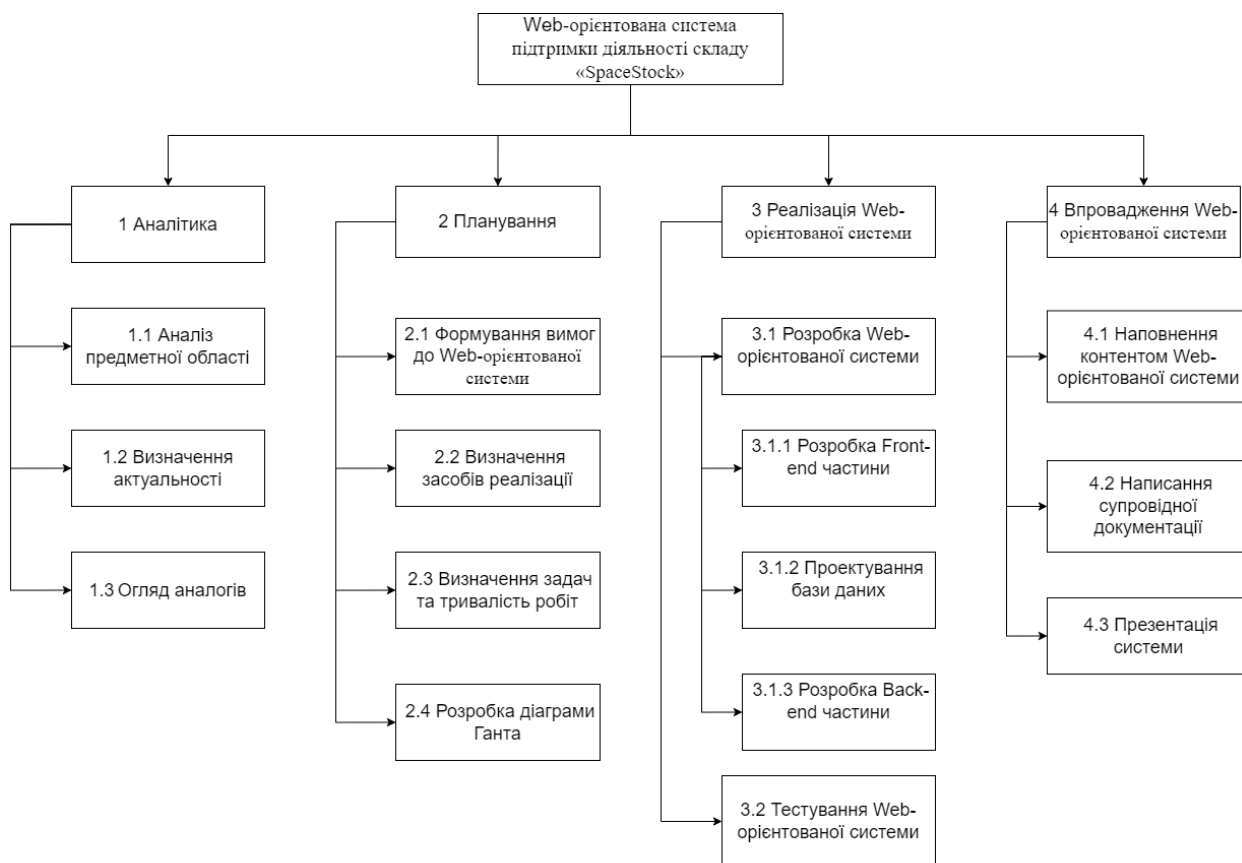


Рисунок А1 – WBS діаграма

*Джерело: побудовано автором*

## Планування структури організації

Організаційна структура організації (Organizational Breakdown Structure - OBS) – інструмент управління, котрий використовується для графічного відображення ієрархії та взаємозв'язків між різними функціональними частинами та підрозділами організації [38]. OBS надає зрозумілу структуру для визначення ролей, та ланок комунікації всередині команди.

Переглянути OBS діаграму можна на рисунку А2

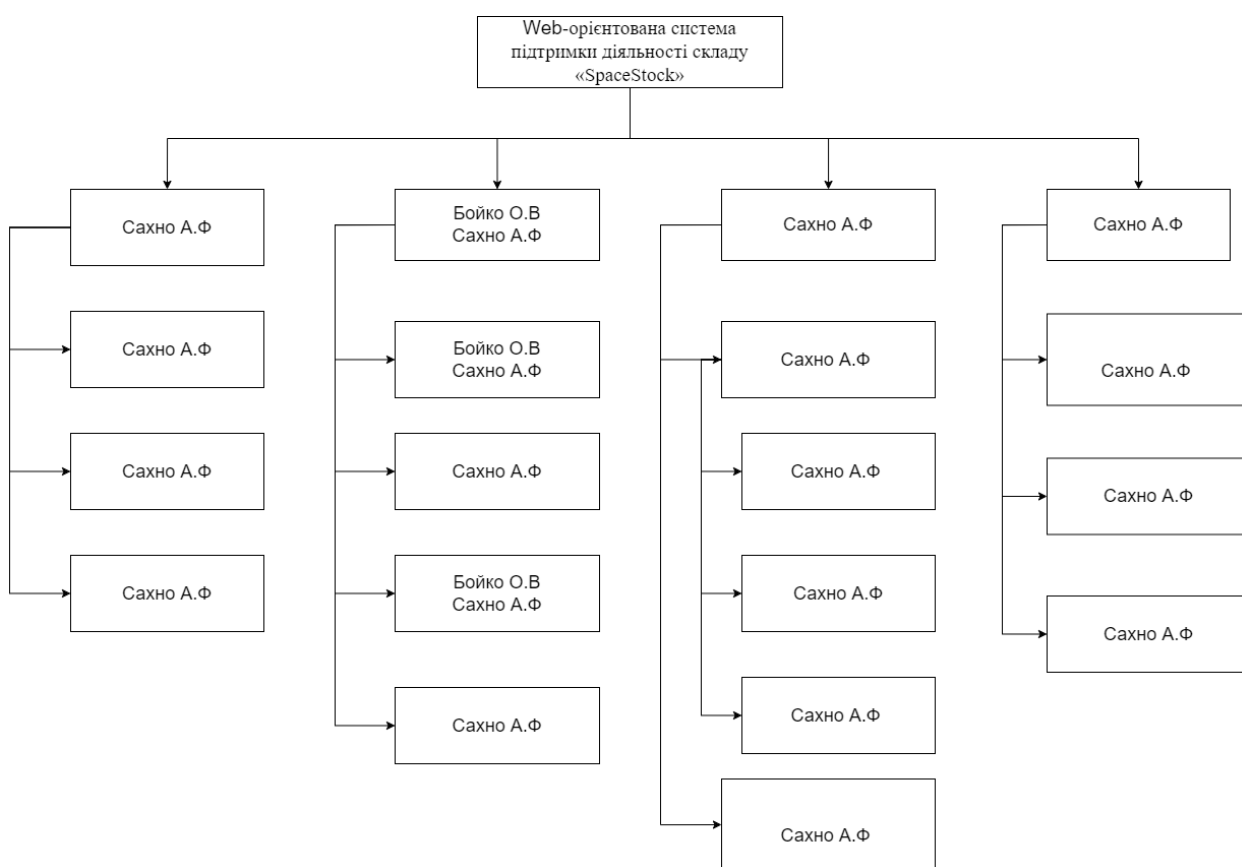


Рисунок А2 – OBS діаграма

*Джерело: побудовано автором*

## Побудова матриці відповідальності

Матриця відповідальності необхідна для чіткого визначення ролей та відповідальностей в рамках:

- Р – бере участь у розробці;
- В – відповідальний виконавець;
- У – особа, що узгоджує вихідний результат

Таблиця А – Матриця відповідальності

№	Задача	Сахно А.Ф	Бойко О.В	Рецензент
1	Аналіз предметної області	В		
2	Визначення актуальності	В		
3	Огляд аналогів	В		
4	Визначення властивостей web-додатку	В	Р	
5	Формування вимог до Web-орієнтованої системи	Р	У	
6	Визначення засобів реалізації	В	У	
7	Визначення задач та тривалість робіт	В	Р	
8	Розробка діаграми Ганта	В	У	

9	Розробка Front-end частини	В	У	
10	Розробка Back-end частини	В	У	
11	Тестування Web-орієнтованої системи	В	У	
12	Наповнення контентом Web-орієнтованої системи	В		
13	Написання супровідної документації	В	У	У
14	Презентація системи	В	У	У

*Джерело: побудовано автором*

### **Побудова календарного графіку**

Побудова календарного графіку за допомогою діаграми Ганта є ефективним інструментом для візуалізації термінів виконання завдань та проектів [39]. Діаграма Ганта дозволяє побудувати часову лінію проекту, розподілити задачі і підзадачі, вказати терміни їх виконання та визначити відношення між ними [40].

Першу частину діаграми Ганта можна переглянути на Рисунку А3

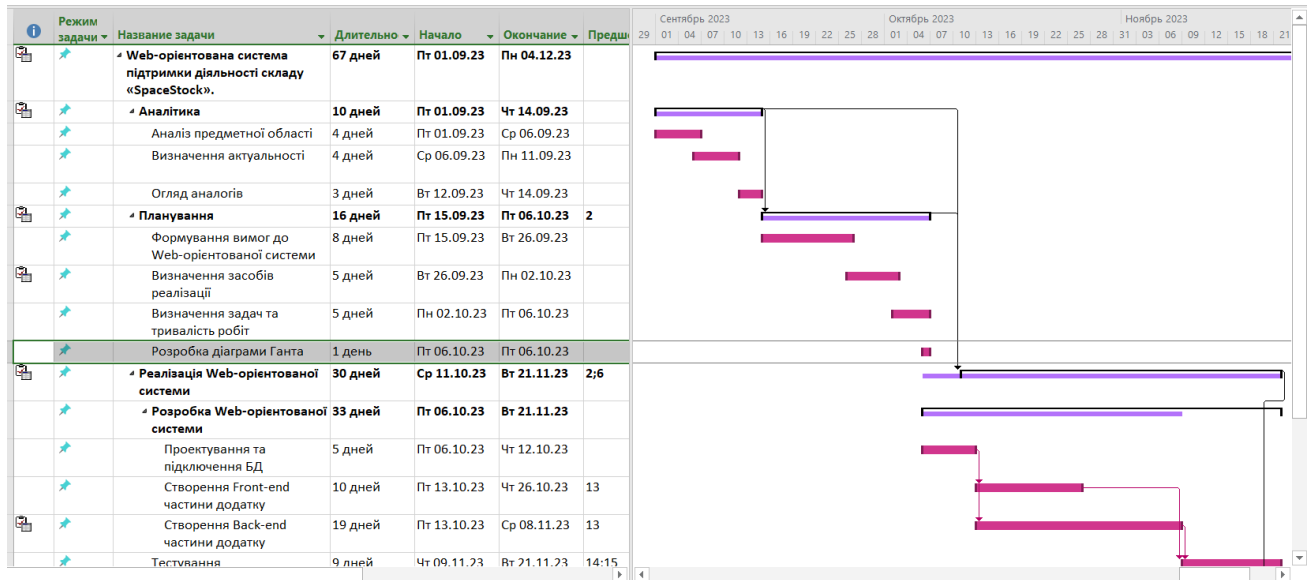


Рисунок А3 – Діаграма Ганта

*Джерело: побудовано автором*

Друга частина діаграми Ганта зображена на рисунку А4

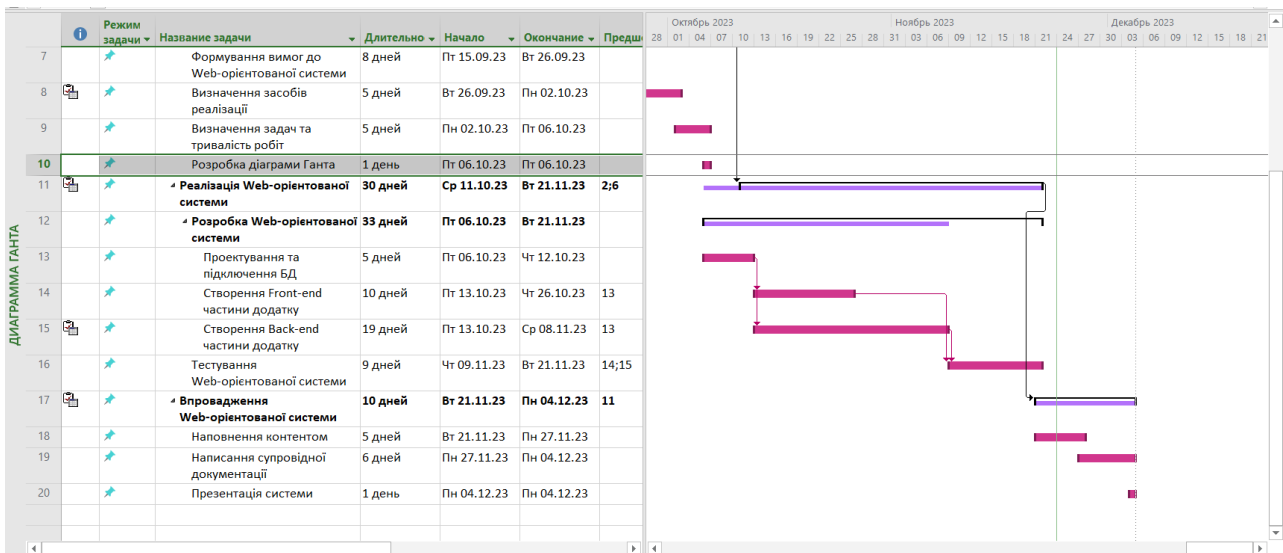


Рисунок А4 – Діаграма Ганта

*Джерело: побудовано автором*



## Управління ризиками проекту

Кожен проект, так само як і будь-яке підприємство, піддається ризикам під час своєї реалізації. Ризик проекту визначається як невизначена подія чи умова, яка може вплинути негативно на цілі проекту [41]. Мета управління ризиками проекту полягає у підвищенні ймовірності виникнення та впливу сприятливих подій і зниженні впливу подій, що можуть виявитися негативними для проекту.

В таблиці А3 вказана шкала оцінювання ризиків

№	Назва	Межі
1	Прийнятні	$1 \leq R \leq 2$
2	Виправдані	$3 \leq R \leq 4$
3	Недопустимі	$6 \leq R \leq 9$

*Джерело: побудовано автором*

У таблиці А.4 вказано можливі ризики:

№	Назва	Опис
1	Відсутність електропостачання	Недоступність розробника вчасно виконати роботу
2	Зміна вимог	Зміна вже на пізньому етапі вимог або термінів роботи
3	Поява альтернативного продукту	Поява альтернативного продукту в Інтернет просторі
4	Технічні ризики	Недостатня ефективність або несправність використовуваних технічних рішень

5	Організаційні ризики	Недостатня підтримка керівництва та відсутність ресурсів для реалізації проекту.
6	Складність у підтримці продукту	Неспроможність замовника самостійно підтримувати продукт

*Джерело: побудовано автором*

В таблиці А5 класифіковано ризики:

Таблиця А5 – Класифікація ризиків

ІД ризику	Ризик	Ймовірність виникнення	Вплив ризику	Ранг ризику
1	Відсутність електропостачання	Висока	Високий	8
2	Зміна вимог продукту	Низька	Високий	8
3	Поява альтернативного продукту	Середня	Середній	4
4	Технічні ризики	Низька	Високий	8
5	Організаційні ризики	Низька	Середній	4
6	Складність у підтримці продукту	Низька	Середній	4

*Джерело: побудовано автором*

В таблиці А6 наведені реакції на ризик з метою звести вплив ризику на мінімум або знизити його вплив.

Таблиця А6 – Реакції на можливі ризики проекту.

ІД ризику	Ризик	Реакція на ризик
1	Відсутність електропостачання	Забезпечити альтернативне джерело енергії та інтернету.
2	Зміна вимог продукту	Постійна комунікація з замовником в процесі розробки продукту.
3	Поява альтернативного продукту	Оптимізація процесу створення проекту шляхом обрання унікальної стратегії.
4	Технічні ризики	Використання вже відомих і перевірених технологій.
5	Організаційні ризики	Ретельне планування проекту, включаючи визначення завдань, ресурсів та графіка виконання, допомагає уникнути непорозумінь та недоліків у роботі команди
6	Складність у підтримці продукту	Провести аналіз та вибрати технологію розробки, яка є зрозумілою та легкою у використанні.

*Джерело: побудовано автором*

## ДОДАТОК Б

### Лістинг програмного коду

#### Лістинг Б.1 – клас сутності бази даних Product

```
PACKAGE COM.SPACESTOCK.WAREHOUSE.MODELS;  
IMPORT JAVAX.PERSISTENCE.*;  
IMPORT LOMBOK.GETTER;  
IMPORT LOMBOK.NOARGSCONSTRUCTOR;  
IMPORT LOMBOK.SETTER;  
  
@ENTITY  
@TABLE (NAME = "PRODUCTS", SCHEMA = "WAREHOUSE")  
@GETTER  
@SETTER  
@NOARGSCONSTRUCTOR  
PUBLIC CLASS PRODUCT {  
    @ID  
    @GENERATEDVALUE (STRATEGY = GENERATIONTYPE.IDENTITY)  
    @COLUMN (NAME = "ID", NULLABLE = FALSE, UNIQUE = TRUE)  
    PRIVATE LONG ID;  
    @COLUMN (NAME = "CODE", NULLABLE = FALSE, UNIQUE = TRUE)  
    PRIVATE STRING CODE;  
    @COLUMN (NAME = "NAME", NULLABLE = FALSE)  
    PRIVATE STRING NAME;  
    @COLUMN (NAME = "DESCRIPTION")  
    PRIVATE STRING DESCRIPTION;  
    @MANYTOONE  
    @JOINCOLUMN (NAME = "CATEGORY_ID")  
    PRIVATE CATEGORY CATEGORY;  
}
```

#### Лістинг Б.2 – клас сутності бази даних Product

```

PACKAGE COM.SPACESTOCK.WAREHOUSE.MODELS;
IMPORT JAVAX.PERSISTENCE.*;
IMPORT LOMBOK.GETTER;
IMPORT LOMBOK.NOARGSCONSTRUCTOR;
IMPORT LOMBOK.SETTER;
IMPORT JAVA.SQL.TIMESTAMP;
@Entity
@Table (NAME = "OPERATIONS", SCHEMA = "WAREHOUSE")
@Getter
@Setter
@NoArgsConstructor
public class OPERATION {
    @Id
    @GeneratedValue (STRATEGY = GenerationType.IDENTITY)
    @Column (NAME = "ID", NULLABLE = FALSE, UNIQUE = TRUE)
    private long ID;
    @ManyToOne
    @JoinColumn (NAME = "STOCK_ITEM_ID")
    private StockItem stockItem;
    @ManyToOne
    @JoinColumn (NAME = "TYPE", REFERENCEDCOLUMNNAME = "TYPE")
    private OPERATIONTYPE operationType;
    @Column (NAME = "QUANTITY")
    private long QUANTITY;
    @Column (NAME = "TIMESTAMP")
    private Timestamp timestamp;
    @ManyToOne
    @JoinColumn (NAME = "USER_ID")
    private User user;
}

```

### Лістинг Б.3 – клас сутності бази даних Product

```

PACKAGE COM.SPACESTOCK.WAREHOUSE.WEB;

```

```

IMPORT COM.SPACESTOCK.WAREHOUSE.MODELS.*;
IMPORT COM.SPACESTOCK.WAREHOUSE.REPOSITORIES.*;
IMPORT ORG.SPRINGFRAMEWORK.BEANS.FACTORY.ANNOTATION.AUTOWIRED;
IMPORT ORG.SPRINGFRAMEWORK.SECURITY.CORE.ANNOTATION.AUTHENTICATIONPRINCIPAL;
IMPORT ORG.SPRINGFRAMEWORK.STEREOTYPE.CONTROLLER;
IMPORT ORG.SPRINGFRAMEWORK.UI.MODEL;
IMPORT ORG.SPRINGFRAMEWORK.WEB.BIND.ANNOTATION.GETMAPPING;
IMPORT ORG.SPRINGFRAMEWORK.WEB.BIND.ANNOTATION.POSTMAPPING;
IMPORT ORG.SPRINGFRAMEWORK.WEB.BIND.ANNOTATION.REQUESTPARAM;
IMPORT ORG.SPRINGFRAMEWORK.WEB.SERVLET.MVC.SUPPORT.REDIRECTATTRIBUTES;
IMPORT JAVA.SQL.TIMESTAMP;
IMPORT JAVA.TIME.LOCALDATETIME;
IMPORT JAVA.UUTIL.LIST;
@Controller
public class StockController {
    @Autowired
    private StockItemRepository stockItemRepository;
    @Autowired
    private ProductRepository productRepository;
    @Autowired
    private CategoryRepository categoryRepository;
    @Autowired
    private StockItemTypeRepository stockItemTypeRepository;
    @Autowired
    private OperationRepository operationRepository;
    @Autowired
    private OperationTypeRepository operationTypeRepository;
    @GetMapping("/stock")
    public String stockItems (@AuthenticationPrincipal User user,
                              @RequestParam(required = false) String code,
                              @RequestParam(required = false) String name,
                              Model model) {
        List<StockItem> stockItems;
        if (code != null && !code.isEmpty()) {

```

```

        STOCKITEMS = STOCKITEMREPOSITORY.FINDBYPRODUCTCODELIKE ("% " + CODE +
"%") .ORELSE (NULL) ;
    } ELSE IF (CODE != NULL && !NAME.ISEMPTY()) {
        STOCKITEMS =
STOCKITEMREPOSITORY.FINDBYPRODUCTNAMELIKE (NAME) .ORELSE (NULL) ;
    } ELSE {
        STOCKITEMS = STOCKITEMREPOSITORY.FINDALL () ;
    }
    MODEL.ADDATTRIBUTE ("STOCKITEMS", STOCKITEMS) ;
    RETURN "STOCKITEMS" ;
}
@GetMapping ("/stock/addtostock")
PUBLIC STRING ADDTOSTOCK (@AUTHENTICATIONPRINCIPAL User USER,
        @REQUESTPARAM (REQUIRED = FALSE) STRING CODE,
        MODEL MODEL) {
    LIST<CATEGORY> ALLCATEGORIES = CATEGORYREPOSITORY.FINDALL () ;
    LIST<STOCKITEMTYPE> ALLSTOCKITEMTYPES = STOCKITEMTYPEREPOSITORY.FINDALL () ;
    PRODUCT PRODUCT = PRODUCTREPOSITORY.FINDBYCODE (CODE) .ORELSE (NULL) ;
    IF (PRODUCT != NULL ) {
        MODEL.ADDATTRIBUTE ("PRODUCT", PRODUCT) ;
    }
    MODEL.ADDATTRIBUTE ("CATEGORIES", ALLCATEGORIES) ;
    MODEL.ADDATTRIBUTE ("STOCKITEMTYPES", ALLSTOCKITEMTYPES) ;
    MODEL.ADDATTRIBUTE ("CODE", CODE) ;
    RETURN "ADDTOSTOCK" ;
}
@PostMapping ("/stock/addtostock")
PUBLIC STRING ADDTOSTOCK (
    @AUTHENTICATIONPRINCIPAL User USER,
    @REQUESTPARAM STRING CODE,
    @REQUESTPARAM STRING NAME,
    @REQUESTPARAM STRING DESCRIPTION,
    @REQUESTPARAM LONG CATEGORY,
    @REQUESTPARAM STRING STOCKITEMTYPE,
    @REQUESTPARAM LONG QUANTITY,
    REDIRECTATTRIBUTES REDIRECTATTRIBUTES
) {
    PRODUCT PRODUCT = PRODUCTREPOSITORY.FINDBYCODE (CODE) .ORELSE (NULL) ;
    IF (PRODUCT == NULL) {

```

```

        PRODUCT = NEW PRODUCT ();
        PRODUCT.SETNAME (NAME);
        PRODUCT.SETDESCRIPTION (DESCRIPTION);
        PRODUCT.SETCATEGORY (CATEGORYREPOSITORY.GETBYID (CATEGORY));
        PRODUCT.SETCODE (CODE);
        PRODUCTREPOSITORY.SAVE (PRODUCT);
    }
    STOCKITEM STOCKITEM =
STOCKITEMREPOSITORY.FINDBYPRODUCTCODEANDTYPE (CODE, STOCKITEMTYPE).ORELSE (NULL);
    IF (STOCKITEM == NULL) {
        STOCKITEM = NEW STOCKITEM ();
        STOCKITEM.SETTYPE (STOCKITEMTYPEREPOSITORY.FINDBYTYPE (STOCKITEMTYPE).ORELSE (NULL));
        STOCKITEM.SETPRODUCT (PRODUCT);
        STOCKITEM.SETQUANTITY (QUANTITY);
        STOCKITEMREPOSITORY.SAVE (STOCKITEM);
    } ELSE {
        STOCKITEM.SETQUANTITY (STOCKITEM.GETQUANTITY () + QUANTITY);
        STOCKITEMREPOSITORY.SAVE (STOCKITEM);
    }
    OPERATION OPERATION = NEW OPERATION ();
    OPERATION.SETSTOCKITEM (STOCKITEM);
    OPERATION.SETQUANTITY (QUANTITY);
    OPERATION.SETTIMESTAMP (TIMESTAMP.VALUEOF (LOCALDATETIME.NOW ()));
    OPERATION.SETOPERATIONTYPE (OPERATIONTYPEREPOSITORY.FINDBYTYPE ("A").ORELSE (NULL));
    OPERATION.SETUSER (USER);
    OPERATIONREPOSITORY.SAVE (OPERATION);
    REDIRECTATTRIBUTES.ADDFLASHATTRIBUTE ("ALERTSUCCESS", QUANTITY
        + " " + CODE + " " +
        "PRODUCTS ADDED TO STOCK");
    RETURN "REDIRECT:/STOCK";
}
@GetMapping ("/STOCK/RELEASEFROMSTOCK")
PUBLIC STRING RELEASEFROMSTOCK (@AUTHENTICATIONPRINCIPAL USER USER,
                                @REQUESTPARAM LONG STOCKITEMID, MODEL MODEL) {
    STOCKITEM STOCKITEM =
STOCKITEMREPOSITORY.FINDBYID (STOCKITEMID).ORELSE (NULL);
    MODEL.ADDATTRIBUTE ("STOCKITEM", STOCKITEM);

```



```

        RETURN "RELEASEFROMSTOCK";
    }
    @PostMapping ("/stock/releasefromstock")
    public String releasefromstock (
        @AuthenticationPrincipal User user,
        @RequestParam Long stockItemId,
        @RequestParam Long quantity,
        Model model,
        RedirectAttributes redirectAttributes
    )
    {
        StockItem stockItem =
        StockItemRepository.findById (stockItemId) .orElse (null) ;
        if (quantity <= stockItem.getQuantity ()) {
            long newQuantity = stockItem.getQuantity () - quantity ;
            stockItem.setQuantity (newQuantity) ;
            StockItemRepository.save (stockItem) ;
            Operation operation = new Operation () ;
            operation.setStockItem (stockItem) ;
            operation.setQuantity (quantity) ;
            operation.setTimestamp (Timestamp.valueOf (LocalDateTime.now ()) ) ;
            Operation.setOperationType (OperationTypeRepository.findByType ("R") .orElse (null) ) ;
            operation.setUser (user) ;
            OperationRepository.save (operation) ;
            redirectAttributes.addFlashAttribute ("alertSuccess",
                quantity + " "
                    + "'" + stockItem.getProduct ().getCode () + "'"
                    + " stock items released" ) ;
            return "redirect:/stock";
        } else {
            redirectAttributes.addFlashAttribute ("alertError", "More product quantity
            indicated than available in stock" ) ;
            model.addAttribute ("product", stockItem.getProduct ()) ;
            model.addAttribute ("stockItemId", stockItemId) ;
            return "redirect:/stock/releasefromstock?stockItemId=" + stockItemId;
        }
    }
}

```