

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ
Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»
Директор інституту(декан факультету)
_____ Андрій ФОРСЮК _____
(підпис) (ім'я та прізвище)
«2» червня 2025р.

«До захисту допущено»
Завідувач кафедри
_____ Сергій ГРИБКОВ _____
(підпис) (ім'я та прізвище)
«2» червня 2025р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»
(код та назва спеціальності)
освітньо-професійної програми Інформаційні системи та штучний інтелект

на тему: «Розроблення інформаційної системи для складського обліку мережі магазинів “Смак Кореї”»

Виконав: здобувач 4 курсу, групи КН-4-4

_____ Мазаєва Юлія Андріївна _____
(підпис)

Керівник Харкянен Олена Валеріївна _____
(прізвище , ім'я та по батькові повністю) (підпис)

Консультанти _____
(ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

Рецензент _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незарядженої допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ – 2025 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма Інформаційні системи та штучний інтелект

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних технологій, штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ

“ 28 ” квітня 2025 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Мазасва Юлія Андріївна

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення інформаційної системи для складського обліку мережі магазинів “Смак Кореї”»

керівник роботи Харкянен Олена Валеріївна, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 28 квітня 2025 року № 254-кв

2. Строк подання здобувачем роботи 30.05.2025 р.

3. Вихідні дані до роботи

1. Інформація про мережу магазинів «Смак Кореї»

2. Організаційна структура мережі магазинів «Смак Кореї»

3. Штатно-посадові інструкції, облікові документи.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження предметної області та постановка задачі

2. Технічне завдання

3. Розроблення програмного продукту.

5. Перелік графічного матеріалу

1. Організаційна структура мережі магазинів «Смак Кореї».

2. Моделі бізнес процесів мережі магазинів «Смак Кореї».

3. Логічна структура бази даних.

4. Фізична структура бази даних.

5. Структура бази даних в СУБД MS SQL Server.

6. Скріншоти інтерфейсу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Харкянен О.В., доцент НУХТ	28.04.2025	12.05.2025
2	Харкянен О.В., доцент НУХТ	28.04.2025	19.05.2025
3	Харкянен О.В., доцент НУХТ	28.04.2025	26.05.2026

7. Дата видачі завдання 28.04.2025**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області та постановка завдання на проектування	28.04.2025- 29.04.2025	Виконано
2	Проектування бази даних	29.04.2025- 10.05.2025	Виконано
3	Розробка інформаційної системи	11.05.2025- 20.05.2025	Виконано
4	Тестування інформаційної системи	21.05.2025- 25.05.2025	Виконано
5	Оформлення пояснювальної записки та створення презентації	25.05.2025- 26.05.2025	Виконано

Здобувач

(підпис)

Юлія МАЗАСВА

(прізвище та ініціали)

Керівник роботи

(підпис)

Олена ХАРКЯНЕН

(прізвище та ініціали)

АНОТАЦІЯ

У кваліфікаційній роботі розглянута організація діяльності магазину роздрібної торгівлі магазину корейської продукції, зокрема зосереджено увагу на оптимізації внутрішньої структури підприємства для забезпечення ефективної роботи. Основну увагу було приділено роботі та функції відділів, серед яких важливу роль займає автоматизований складський відділ, який є центром управління товарами, їх залишками та зберіганням продукції, взаємодіє з логістичним, фінансовим та відділом продажів.

Розроблено інформаційну систему обліку надходжень та видачі товару, місця їх зберігання, та кількість однакових товарів різного терміну придатності.

Для реалізації інформаційної системи були використані сучасні засоби розробки. Моделювання структури бази даних виконано в середовищі AllFusion ERwin Data Modeler, що дозволило створити логічну та фізичну модель системи з чітко визначеними зв'язками між сутностями. Для реалізації самої бази даних використано Microsoft SQL Server, а адміністрування та керування базою здійснювалося через SQL Server Management Studio (SSMS). Інтерфейс користувача було реалізовано з використанням технології Windows Forms на мові програмування C#, що забезпечує зручну взаємодію з користувачем та стабільну роботу застосунку в середовищі Windows.

Дипломна робота складається з 67 сторінок, 8 таблиць, 23 ілюстрацій та 32 використаних джерел.

Ключові слова: МАГАЗИН КОРЕЙСЬКОЇ ПРОДУКЦІЇ, СКЛАДСЬКИЙ ВІДДІЛ, ОБЛІК ТОВАРУ, РОЗРОБКА ДОДАТКУ.

SUMMARY

The qualification thesis examined the organization of a retail store specializing in Korean products, with a focus on optimizing the internal structure of the enterprise to ensure efficient operation. Particular attention was paid to the work and functions of the departments, among which the automated warehouse department plays an important role. It serves as the center for managing goods, their inventory, and product storage, and interacts with the logistics, finance, and sales departments.

An information system was developed to record the receipt and release of goods, their storage locations, and the quantity of identical products with different expiration dates.

Modern development tools were used to implement the information system. The database structure was modeled using AllFusion ERwin Data Modeler, which allowed the creation of a logical and physical model of the system with clearly defined relationships between entities. The database itself was implemented using Microsoft SQL Server, and database management and administration were carried out via SQL Server Management Studio (SSMS). The user interface was developed using Windows Forms technology in the C# programming language, which ensures convenient user interaction and stable application performance in the Windows environment.

The thesis consists of 67 pages, 8 tables, 23 illustrations, and 32 sources.

Keywords: KOREAN PRODUCTS STORE, WAREHOUSE DEPARTMENT, INVENTORY MANAGEMENT, APPLICATION DEVELOPMENT.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1. Загальна характеристика магазинів корейської продукції «Смак Кореї».....	9
1.2. Організаційна структура магазинів корейських продуктів «Смак Кореї», роль та взаємодія підрозділів	10
1.3. Аналіз нинішнього стану комп'ютеризації магазинів «Смак Кореї».....	15
1.4. Моделювання та аналіз існуючих бізнес-процесів магазину «Смак Кореї»	17
1.5 . Огляд існуючих рішень	20
1.6. Техніко-економічне обґрунтування впровадження програмного забезпечення.....	22
1.7. Обґрунтування доцільності проектування й розроблення інформаційної системи для складського відділу магазину корейської продукції «Смак Кореї»...	26
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ.....	27
2.1. Загальні положення.....	27
2.2. Призначення і цілі створення системи.....	27
2.3. Характеристика об'єкта автоматизації	28
2.4. Вимоги до системи.....	28
2.5. Склад і зміст робіт по створенню системи	36
2.6. Порядок контролю і приймання системи	38
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію	38
2.8. Вимоги до документації	39
2.9. Джерела розробки	40
РОЗДІЛ 3. РОЗРОБЛЕННЯ ПРОГРАМНОГО ПРОДУКТУ	41

3.1.Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту	41
3.2.Проектування та створення бази даних	42
3.3.Реалізація функцій системи	46
3.4.Інструкція користувача.....	49
3.5.Тестування програмного продукту	59
ВИСНОВОК.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТКИ.....	67
Додаток А. Схема організаційної структури мережі магазинів «Смак Кореї»	67
Додаток Б. Модель бізнес-процесів діяльності мережі магазинів «Смак Кореї» ..	68
Додаток В. Логічна модель бази даних.....	70
Додаток Г. Фізична модель бази даних.....	71
Додаток Д. Модель бази даних в СУБД MS SQL Server.....	72
Додаток Е. Фрагменти програмного коду	73

ВСТУП

У сучасному світі відбувається швидкий розвиток власних малих бізнесів самих незвичних напрямків, тому бути конкурентноспроможним та відповідати сучасним вимогам є необхідною складовою для успішного бізнесу. Для підтримання статусу надійного та ефективного постачальника, яким мережа магазинів Смак Кореї і є, потрібно встигати за новими тенденціями. Саме тому створення інформаційної системи для складського обліку товарів є доцільним рішенням, адже саме це сприятиме спрощенню роботи всієї установи та підвищить ефективність роботи.

Метою даної кваліфікаційної роботи є розроблення повноцінної складської системи для автоматизації процесів управління запасами з використанням технологій C# та Windows Forms та СУБД Microsoft SQL Server для серверної частини. Основний акцент було зроблено на автоматизацію завантаження, фільтрацію та відображення товару, запасів та замовлень.

У межах роботи здійснено аналіз складського обліку, сформульовано вимоги та мету системи, підготовлено технічне завдання, спроектовано архітектуру програмного забезпечення, реалізовано ключовий функціонал і проведено тестування.

Результатом проєктування і розробки є складська система, готова до впровадження в роботу складу та відповідає актуальним вимогам щодо ефективності, надійності та зручності у використанні. Система надає зручний інтерфейс для робітників складу, обробку замовлень та поставок і ефективно управління запасами.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальна характеристика магазинів корейської продукції «Смак Кореї»

1.1.1. Історія створення

Ідея створення магазину «Смак Кореї» вперше з'явилася з приїздом етнічних жильців Південної Кореї в Україну. Так як в Україні майже відсутня культура інших країн, то рішення про створення даного типу магазину була вигідною та цікавою для українського ринку [5].

Організувавши стабільні та фінансово вигідні поставки, за допомогою прямих зв'язків з постачальниками, спочатку було створено ТОВ «ВІДУС», який і досі є основним доходом для власників. При отриманні статусу надійного та швидкого постачальника та отримавши початковий бюджет, було створено мережу з магазинів «Смак Кореї» і для зручності розроблено інтернет-магазин [6].

1.1.2. Мета та основні цілі магазину

Як відомо, багато людей не мають достатньо сміливості або зацікавленості щоб пробувати продукцію інших країн, тому однією із головних завдань магазину «Смак Кореї» буде створення привабливої картинки на рекламах для просування продукції та створення зацікавленості у потенційних покупців.

Також важливими аспектами роботи магазину буде організація постійних поставок та поповнення товарів актуальними та трендовими товарами. Адже, основна передбачувана клієнтура буде молодь, яка регулярно проводить багато часу у соціальних мережах та готова спробувати багато чого цікавого, якщо воно стало популярним.

Звідси можна сформувати основну мету магазину – забезпечити українців якісною продукцією перевірених брендів, сприяючи розширенню їх кругозору та розвинути зацікавленість до країни виробника – Південної Кореї.

Основні цілі діяльності магазину:

1. Просування іноземної продукції на українському ринку та заохочення населення до нових вражень.
2. Забезпечення стабільних поставок та оновлення асортименту відповідно до світових трендів.
3. Формування довіри та лояльності клієнтів через якісний сервіс, чесність та ефективність у комунікації.

1.2. Організаційна структура магазинів корейських продуктів «Смак Кореї», роль та взаємодія підрозділів

Для злагодженої роботи магазину було прийняте рішення створити 5 відділів, кожен з яких виконує свої обов'язки та грає важливу роль у загальному функціонуванні бізнес-процесів. У таблиці 1.1 наведено функції відділів магазину «Смак Кореї» та їх взаємодії.

Таблиця 1.1. Функції відділів магазину та їх взаємодія

Назва відділу	Основні функції відділу	Взаємодія з іншими відділами
1. Фінансовий відділ	<ul style="list-style-type: none"> • Ведення бухгалтерського обліку, контроль доходів та витрат. • Формування фінансової звітності. • Розрахунок заробітних плат, витрат на податки, оренду та логістику. 	<ul style="list-style-type: none"> • Отримання даних про продажі з відділу продажів. • Отримання документів на поставку (ТТН та квитанції) з відділу логістики. • Узгодження бюджету на реклами з маркетинговим відділом.
2. Відділ продажів	<ul style="list-style-type: none"> • Обслуговування клієнтів у магазинах. • Консультація покупців. • Ведення обліку продажів. 	<ul style="list-style-type: none"> • Отримання товару, при прибутті з складського відділу у магазин. • Передання даних про попит та побажань клієнтів у відділ логістики та маркетингу.

Назва відділу	Основні функції відділу	Взаємодія з іншими відділами
		<ul style="list-style-type: none"> • Передання звітів про виручку у фінансовий відділ.
3. Відділ логістики	<ul style="list-style-type: none"> • Вибір найвигіднішого постачальника. • Узгодження термінів поставок та заключення контрактів з постачальниками. • Організування транспортування. • Аналіз обсягу запасів. 	<ul style="list-style-type: none"> • Передання інформації про прибуття нового товару у складський відділ. • Отримання звітів про залишки зі складу.
4. Маркетинговий відділ	<ul style="list-style-type: none"> • Просування продукції магазину (у соціальних мережах або месенджерах). • Створення візуально приємного оформлення у магазині. • Аналіз трендів та комунікація з аудиторією. 	<ul style="list-style-type: none"> • Узгодження витрат на промо із фінансовим відділом. • Повідомлення логістичного відділу про наявність акцій для достатньої кількості товару.
5. Складський відділ	<ul style="list-style-type: none"> • Приймання товару з таможні та доставка його на склад. • Зберігання продукції згідно до відповідних умов. • Ведення обліку залишків товару. • Надійне пакування та доставка продукції у магазини. 	<ul style="list-style-type: none"> • Отримання поставок від логістичного відділу. • Передання звітів про залишки у фінансовий відділ.

1.2.1. Загальна схема організаційної структури

Для чіткої та ефективної роботи магазину корейських продуктів було створено 5 відділів та розподілено обов'язки між ними, згідно з потребами. Організаційна структура побудована таким чином, щоб забезпечити якісне обслуговування

клієнтів, ефективно управляти персоналом і фінансами та рівномірно розподілити обов'язки між працівниками (див. рисунок 1.1)

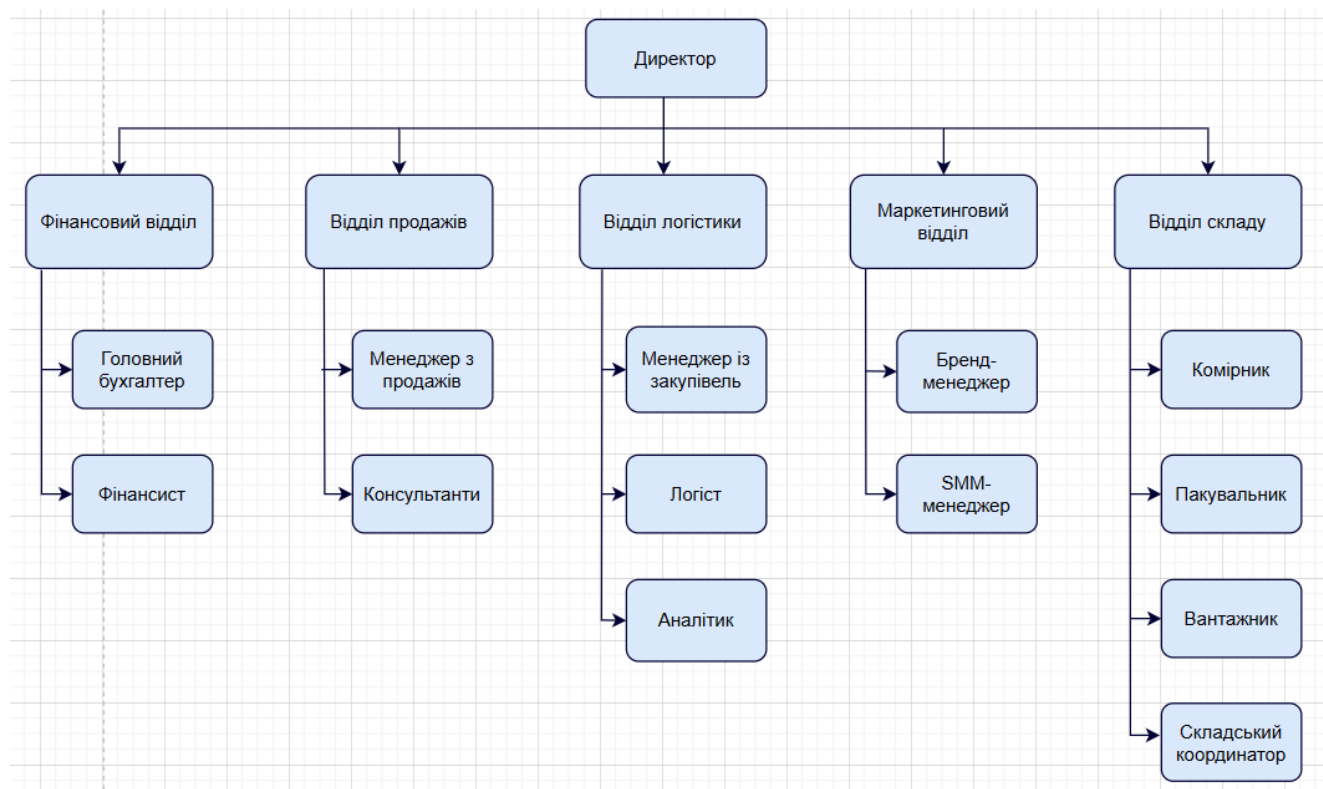


Рисунок 1.1 – Організаційна структура магазинів Смак Кореї

Директор – приймає участь у розвитку бізнесу, прийнятті стратегічно важливих рішень та контролює роботу всіх підрозділів, отримуючи від них всіх звітності.

Відділ логістики – займається вибором постачальників, укладенням контрактів з ними та організацією доставки до складу.

Фінансовий відділ – контролює всі фінансові операції магазину, веде облік надходжень та витрат; дає згоду на витрати іншим відділам, узгоджуючи це з директором.

Відділ продажів – забезпечує обслуговування клієнтів у магазині, створює надійні та довірливі стосунки зі споживачами.

Маркетинговий відділ – просуває продукцію та ідею магазину, створенням візуально привабливого оформлення магазину, реклами та соціальних мереж; завжди має зворотній зв'язок з клієнтами.

Складський відділ – приймає товари, визначає їх місце на складі, згідно з вимогами самих продуктів, та зберігає їх там; контролює залишки товарів, надійно упакує їх та відправляє у магазини, згідно їх потреб.

1.2.2. Структура складського відділу

Злагоджена робота складського відділу є дуже важливою складовою у загальній структурі роботи підприємства. Приймання, облік, зберігання та видача товару для мережі з двох магазинів є основними завданнями відділу, тому для покращення та пришвидшення роботи було впроваджено автоматизовану систему, яка дозволяє відслідковувати залишки, терміни придатності та розміщення продукції на складі.

Коли один із магазинів подає електронну заявку на отримання певної кількості товарів через систему, вона автоматично обробляється у програмі, після чого комірник формує замовлення, знаходить потрібні позиції за адресним зберіганням. Координатор складу уважно перевіряє терміни придатності, формує накладну та оновлює залишки товарів у додатку. Після чого, з допомогою вантажника та пакувальника, продукція надійно упакується та покидає складське приміщення.

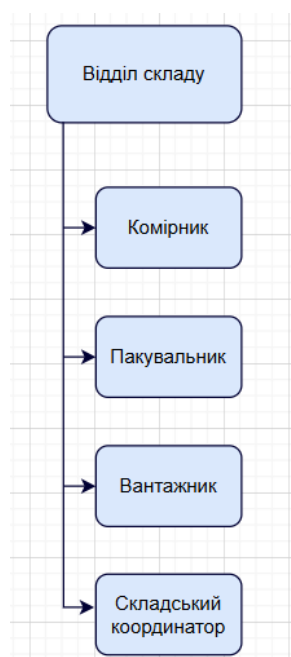


Рисунок 1.2 – Схема складського відділу

Автоматизація даних процесів значно підвищує швидкість обробки заявок, знижує ризики помилок при підборі певного товару, покращує контроль за запасами та забезпечує оперативне постачання продукції до кожного магазину, згідно до їх потреб.

Робота кожного з працівників є злагодженою та чітко розподіленою між собою для правильної координації роботи всієї системи. Їх обов'язки показано у таблиці 1.2.

Таблиця 1.2. Посадові обов'язки робітників складу

Працівник	Посадові обов'язки	Взаємодія з іншими відділами
Комірник	<ul style="list-style-type: none"> • Приймання товарів на склад, перевірка супровідних документів. • Проведення планових інвентаризацій. • Контроль умов зберігання продуктів. • Вибір оптимального місця для розташування товару. 	<ul style="list-style-type: none"> • Отримання документів про надходження від логістичного відділу. • Надання звітів про запаси до відділу продажів.
Складський координатор	<ul style="list-style-type: none"> • Приймання заявок від магазинів через автоматизовану систему. • Розподіл роботи між робітниками складського відділу. • Контроль термінів формування відправлень. 	<ul style="list-style-type: none"> • Приймання замовлення з відділу продажів. • Координація графіку відправлень з логістичним відділом.
Пакувальник	<ul style="list-style-type: none"> • Перевірка товару на відповідність замовленню. • Упаковка продукції згідно вимогам безпеки та потребам товару. • Нанесення маркування для ідентифікації товару. 	<ul style="list-style-type: none"> • Контакткування з відділом продажів у разі виникненні питань чи проблем з термінами чи компонентами замовлення.

Працівник	Посадові обов'язки	Взаємодія з іншими відділами
Вантажник	<ul style="list-style-type: none"> • Завантаження та розвантаження товарів. • Переміщення товарів по складу або торговому залу, згідно з інструкціями. 	<ul style="list-style-type: none"> • Робота під керівництвом складського координатора при погрузці товарів на склад. • Робота під керівництвом адміністратора магазину про погрузці товару в торгову залу.

Для чіткої та ефективної роботи даного відділу важливим моментом є налагоджена система обміну документами з іншими відділами, яка описана у таблиці 1.3.

Таблиця 1.3. Інформаційна взаємодія складського відділу

Підрозділ	Одержані документи	Надані документи
Складський відділ	<ul style="list-style-type: none"> • Повідомлення про надходження товару. • Документи на вантаж (ТТН, сертифікати якості). • Заявки від магазинів на видачу товару. 	<ul style="list-style-type: none"> • Підписана заява на прийнятий з митниці товар. • Звіти про дефіцит або пошкодження. • Звіти про залишки товару до фінансового, логістичного та торгового відділу.

1.3. Аналіз нинішнього стану комп'ютеризації магазинів «Смак Кореї»

Для оцінки рівня комп'ютеризації мережі магазинів «Смак Кореї» було проведено аналіз діяльності ключових відділів, що дозволяє виявити недоліки в автоматизації бізнес-процесів та визначити потенційні напрями для їх оптимізації.

1. Фінансовий відділ

У фінансовому відділі використовується базове бухгалтерське програмне забезпечення для ведення податкової звітності, обліку витрат та прибутків,

нарахування заробітної плати. Але система не є інтегрованою з внутрішнім обліком документів магазину, тому дані у звіти вносяться та передаються вручну. Через це можуть виникнути затримки у формуванні звітів або помилки при їх створенні.

2. Відділ продажів

У фізичних магазинах продажі ведуться через базові касові системи, які під'єднані зі складом. Інформація про попит, залишки та відгуки клієнтів передається у звітах, таблицях та у соціальних мережах між працівниками. Внутрішній облік залишків продукції у торговому залі ведеться вручну, тому формування звітів відбувається із затримкою.

3. Відділ логістики

Діяльність відділу логістики відбувається в основному в Excel-документах, де знаходиться інформація про постачальників, терміни поставок та вартість замовленої продукції. Комунікація з постачальниками відбувається персонально через соціальні мережі. Всі ТТН та електронні квитанції зберігаються в одному місці на робочих комп'ютерах, а інформація про надходження товару передається вручну, що може спричинити неточності у термінах поставок.

4. Маркетинговий відділ

Відділ використовує самі базові та популярні соціальні мережі та інструменти для створення реклами, оскільки вони і є найефективнішими на даний момент (Canva, Google Docs, Instagram, Twitter). Недоліком відділу є відсутність єдиного ефективного інструменту для аналізу результатів маркетингових кампаній та зв'язку з відділом продажів для точного аналізу ефективності актуальних акцій.

5. Складський відділ

Вже є автоматизованим, оскільки є одним із самих ключових ланок у роботі бізнес-процесів. Облік товарів, їх точне місце знаходження та швидкий доступ до інформації про кожен продукт допомагає облегшити роботу не тільки працівникам складського відділу, а і усього магазину. Автоматизація роботи складського відділу допомагає не тільки прискорити роботу, а і не допустити елементарних помилок у відслідковуванні залишків та наявності продукції.

Отже, за результатами аналізу можна помітити що процеси основних п'ятьох відділів мають низький рівень комп'ютеризації та автоматизації. У майже всіх відділах комунікація, і між собою, і між клієнтами відбувається вручну або з використанням елементарних та простих програм. Відсутність єдиної централізованої системи може іноді спричинити затримки у прийнятті рішень або створенні звітів, втрату важливої інформації та знизити загальну ефективність роботи магазину.

1.4. Моделювання та аналіз існуючих бізнес-процесів магазину «Смак Кореї»

1.4.1. Модель бізнес-процесів діяльності складського відділу магазину «Смак Кореї»

Опишемо бізнес-процеси діяльності складського відділу магазину «Смак Кореї» у нотації BPMN рівня AS-IS та TO-BE.

На основі аналізу поточного стану складських процесів (AS-IS) мережі магазинів «Смак Кореї», було створено модель AS-IS, яка відображає неструктуризовані процеси з високим ризиком помилок, та модель TO-BE, яка у свою чергу передбачає автоматизовані процеси з чіткими етапами: завантаження, фільтрація, оновлення та редагування запасів. Рівень AS-IS має базову автоматизацію (ручне введення даних, відсутність інтеграції), а TO-BE забезпечує високий рівень автоматизації з інтеграцією бази даних, реальним часом відображення змін і валідацією даних, що оптимізує управління складом.

На даному етапі діяльність складського відділу магазину «Смак Кореї», а саме забезпечення власних магазинів товаром та контроль залишків, відбувається вручну або через базові електронні таблиці.

При отриманні заявки на товар (у вигляді Google-таблиці), який перевіряється складським координатором, який звіряє запит з наявними залишками, та вручну формує комплектувальний лист і передає про відсутність товарів (якщо така ситуація виникає).

Перевірка товару за накладними або ТТН робить вручну комірником, який і відмічає у роздрукованому списку наявність всіх позицій та їх кількість.

Координатор щотижнево проводить інвентаризацію вручну, змінює кількість у загальній таблиці залишків, а на окремі аркуші заносить інформацію що стосується термінів придатності. При недостатній кількості запасів координатор вручну зв'язується з логістичним відділом та складає список потрібних товарів.

Пакування товару відбувається вручну пакувальником, з урахуванням характеристик про хрупкість товару та додається маркування («Магазин 1», «Акційна позиція» тощо)

Після пакування складський координатор вручну створює вручну транспортний лист, який підписується комірником та передається водієві.

Недоліки даної моделі:

- дублювання даних у таблицях;
- високий ризик помилок при ручному введенні;
- відсутність автоматичного оновлення інформації між відділами та робітниками;
- довгий пошук потрібної інформації.

Модель бізнес-процесів діяльності складського відділу магазину «Смак Кореї» у нотації BPMN рівня AS-IS наведено на рисунку 1.3.

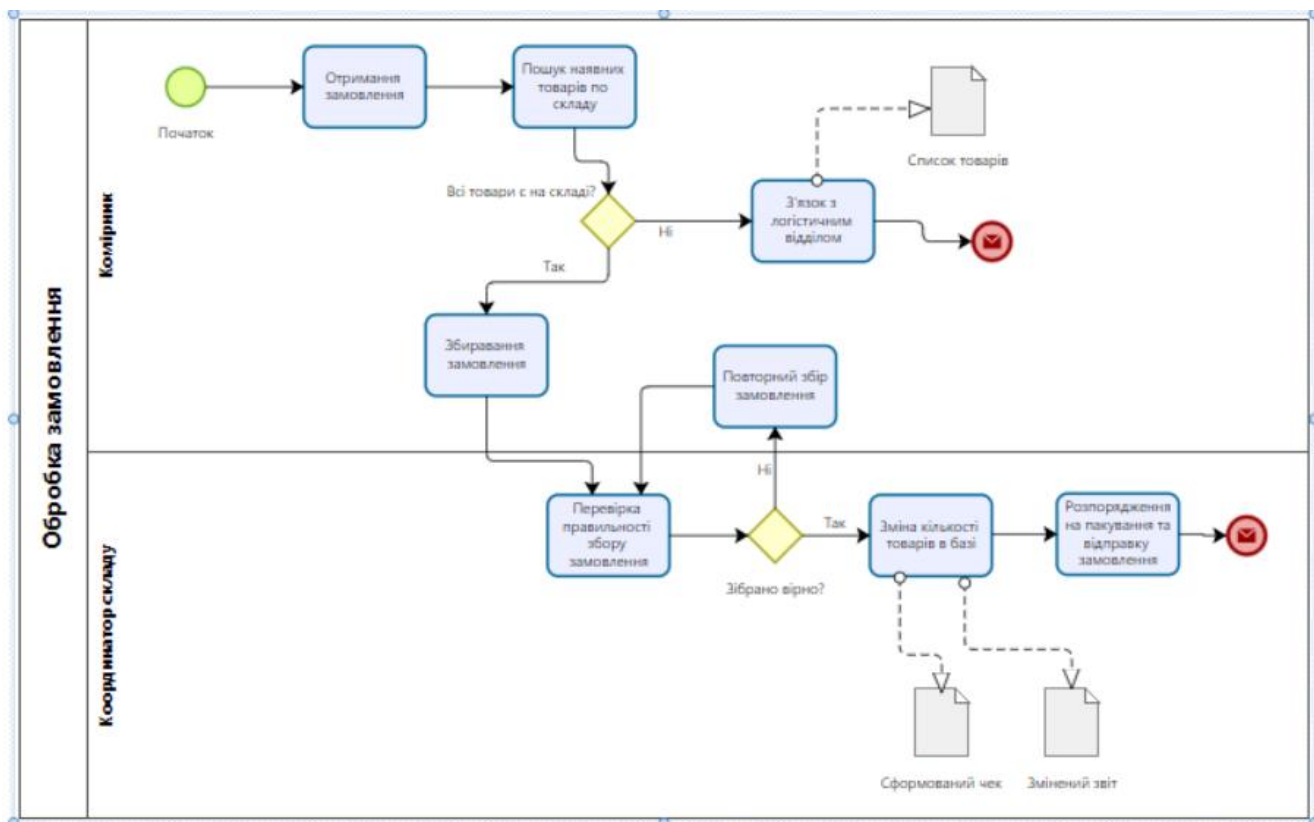


Рисунок 1.3 – Модель бізнес-процесів у нотації BPMN рівня AS-IS

1.4.2. Пропозиції щодо вдосконалення

Створення інформаційної системи для складського відділу дозволить не тільки пришвидшити роботу, а і зменшити кількість можливих помилок. Створена система має список важливих для даного відділу функцій, які покращать ефективність роботи.

1. Автоматичне оновлення залишків товару.
2. Перегляд всієї інформації про товар в одному вікні.
3. Швидкий та зручний пошук певного продукту.
4. Перелік кількості товару різних термінів придатності.

Модель бізнес-процесів діяльності складського відділу магазину «Смак Кореї» у нотації BPMN рівня TO-BE наведена на рисунку 1.4.

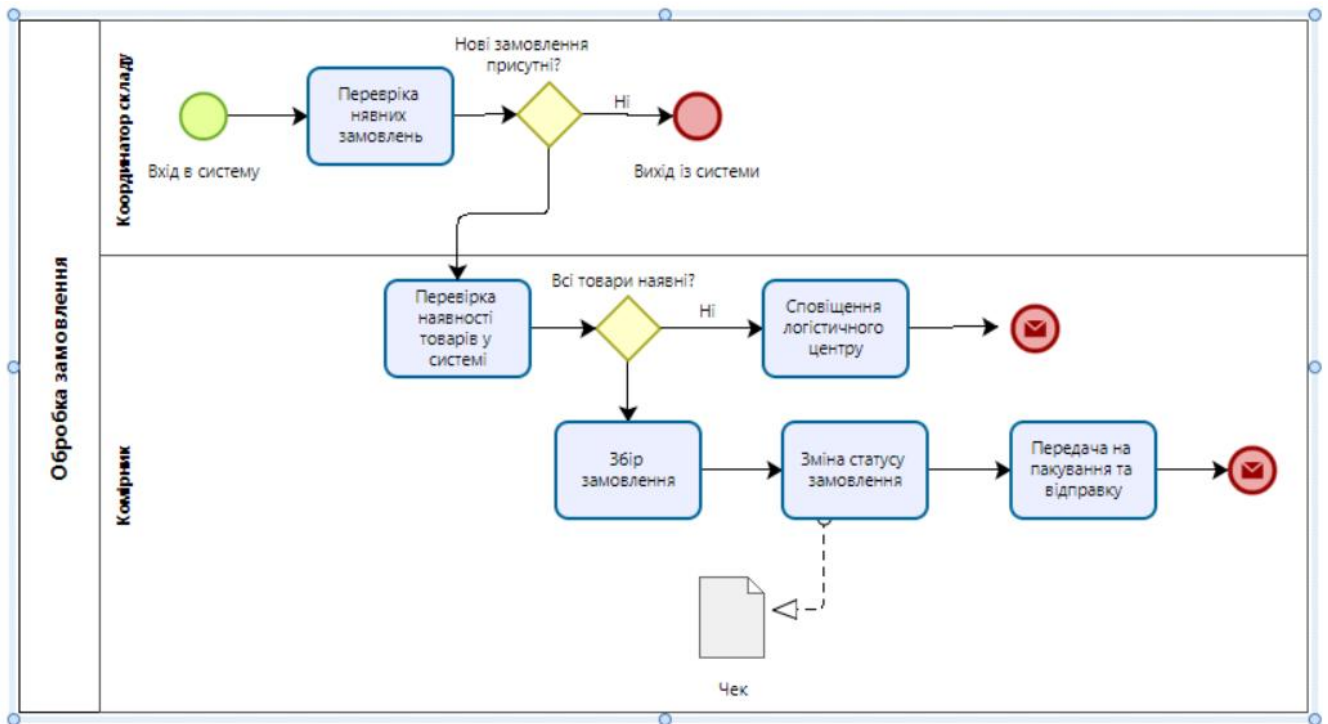


Рисунок 1.4 – Модель бізнес-процесів у нотації BPMN рівня TO-BE

1.5. Огляд існуючих рішень

Для вирішення актуальних проблем відділу та створенні дієвої інформаційної системи було проаналізовано декілька систем подібних магазинів. У цьому підрозділі розглянуто аналоги популярних магазинів, оскільки вони задовольняють вимоги магазину «Смак Кореї».

1.5.1. Інформаційна система магазину «Eva»

Eva – найбільша мережа магазинів косметичних засобів в Україні, яка спеціалізується на продажі не тільки імпортової продукції, а і виробленні та розповсюдженні власних брендів [7].

Система: Власна ERP та WMS IT-Enterprise.

Особливості:

- автоматизоване управління запасами;
- контроль залишків;
- облік термінів придатності;
- звіти та аналізи по кожному із складів.

1.5.2. Інформаційна система магазину «Lan-Ukraine»

Lan-Ukraine – дистриб'ютор продуктів харчування, що забезпечує поставки, як до роздрібних невеликих магазинів, так і закладів харчування по всій Україні.

Система: Qguar WMS Pro [8].

Особливості:

- автоматизовані всі етапи складської логістики;
- гнучка та зручна система контролю залишків і строків придатності;
- зручний модуль управління персоналом.

1.5.3. Інформаційна система магазину «FOZZY C&C»

FOZZY C&C – мережа гуртово-роздрібних магазинів, які спеціалізуються на продажі продуктів харчування, побутової хімії та господарських товарів.

Система: WMS на основі Microsoft Dynamics NAV [9].

Особливості:

- автоматичне формування та контроль замовлень з урахуванням залишків на складі;
- оптимізація маршрутів комплектування товарів.

Проаналізувавши складські інформаційні системи відомих магазинів, було зроблено висновок, що більшість з них вже впровадили ефективні та дійові інформаційні системи для автоматизації роботи складу, що дозволяє їм конкурувати між собою та оперативно реагувати на потреби фізичних магазинів, що дозволяє розширювати бізнес.

1.5.4. Порівняння систем-аналогів

Для оцінки конкурентних рішень у сфері складського управління було проведено порівняння характеристик програмного забезпечення, яке використовують мережі магазинів Eva, Lan-Ukraine та FOZZY C&C. Аналіз охоплює ключові аспекти автоматизації, інтеграції та функціональності систем, що забезпечують ефективне управління запасами. Порівняння наведено у таблиці 1.4.

Таблиця 1.4. Порівняння систем конкурентів

Характеристика	Eva	Lan-Ukraine	FOZZY C&C
Використовуване ПЗ	1С:Підприємство + WMS (власна розробка)	Qguar WMS Pro	WMS на основі Microsoft Dynamics NAV
Автоматичне оновлення залишків у реальному часі	Є	Є	Є
Відстеження термінів придатності	Є, для косметичних товарів обов'язкове	Є, з підтримкою партий та термінів	Є, з управлінням температурними зонами
Облік партій	Є	Є	Є
Мобільне робоче місце для складу	Є	Є	Є
Інтеграція з логістикою	Повна	Повна	Повна
Можливість формування звітів	Є	Є	Є
Підтримка декількох складів	Є	Є	Є
Рівень автоматизації	Високий	Високий	Високий

1.6. Техніко-економічне обґрунтування впровадження програмного забезпечення

Для підтвердження економічної ефективності розробки та впровадження інформаційної системи у складський відділ магазину корейської продукції «Смак Кореї» було виконано розрахунок ключових показників доцільності реалізації проекту [10].

Основні етапи проекту:

1. Аналіз вимог

- Трудомісткість (ТМ): 6 люд.-днів
- Чисельність працюючих (ЧП): 2 особи
- Тривалість (ТР): 3 днів
- Витрати: 1,200 грн (зарплата персоналу) + 400 грн (консультації) = 1,600 грн

2. Проектування

- Трудомісткість (ТМ): 8 люд.-днів
- Чисельність працюючих (ЧП): 2 особи
- Тривалість (ТР): 4 днів
- Витрати: 1,600 грн (зарплата персоналу) + 600 грн (програмне забезпечення для проектування) = 2,200 грн

3. Розробка та налаштування системи

- Трудомісткість (ТМ): 16 люд.-днів
- Чисельність працюючих (ЧП): 2 осіб
- Тривалість (ТР): 8 днів
- Витрати: 10,000 грн (зарплата розробників) + 3,000 грн (програмне забезпечення) = 13,000 грн

4. Тестування

- Трудомісткість (ТМ): 9 люд.-днів
- Чисельність працюючих (ЧП): 3 особи
- Тривалість (ТР): 3 днів
- Витрати: 5,000 грн (зарплата тестувальників) = \$5,000 грн

5. Впровадження

- Трудомісткість (ТМ): 8 люд.-днів
- Чисельність працюючих (ЧП): 2 особи
- Тривалість (ТР): 4 днів
- Витрати: 2,500 грн (зарплата) + 1,500 грн (сервери та налаштування) = 4,000 грн

6. Навчання персоналу

- Трудомісткість (ТМ): 8 люд.-днів
- Чисельність працюючих (ЧП): 2 особи
- Тривалість (ТР): 4 днів
- Витрати: 1,800 грн (тренінги та навчальні матеріали) = 1,800 грн

7. Підтримка та супровід

- Трудомісткість (ТМ): 21 люд.-днів
- Чисельність працюючих (ЧП): 3 особи
- Тривалість (ТР): 7 днів
- Витрати: 10,000 грн (підтримка та обслуговування системи) = \$10,000

Загальні витрати = 1,600 грн + 2,200 грн + 13,000 грн + 5,000 грн + 4,000 грн + 1,800 грн + 10,000 грн = 37,600 грн

Очікуваний економічний ефект:

1. Зниження операційних витрат: завдяки автоматизації процесів управління замовленнями та підтримки клієнтів очікується зниження витрат на обслуговування клієнтів на 15%, що призведе до економії 7,500 грн на рік.

2. Збільшення обсягів продажу: автоматизована система дозволить обробляти до 30% більше запитів та замовлень, що призведе до збільшення доходів на 12,000 грн на рік.

3. Скорочення часу на обробку замовлень: завдяки оптимізації процесів обробки та виконання замовлень скорочення часу на обробку кожного запиту дозволить заощадити 4,000 грн на рік.

Загальний економічний ефект:

7,500 грн (зниження витрат) + 12,000 грн (збільшення доходів) + 4,000 грн (економія на часі) = 23,500 грн на рік.

Варіанти залучення додаткових коштів для бізнесу:

- Гранти та субсидії для малого бізнесу.

Від державних або приватних організацій можуть бути доступні гранти або субсидії для малих підприємств, особливо якщо ваш бізнес працює в інноваційних або соціально значущих сферах.

- Кредити від постачальників та партнерів.

Деякі постачальники можуть запропонувати вам фінансування або розстрочку на придбання товарів. Це дозволяє зменшити початкові витрати на закупку товару для подальшого продажу.

Окупність проекту:

Термін окупності проекту розраховується за формулою:

$$\text{Термін окупності} = \frac{\text{Витрати на проєкт}}{\text{Річний економічний ефект}} = \frac{37,600}{23,500} \approx 1,60 \text{ роки}$$

Отже, термін окупності складає приблизно 1,60 роки.

Звідси можемо зробити висновок, що проєкт автоматизації магазину корейської продукції є економічно вигідним. Очікуваний річний економічний ефект складає 23,500 грн, що дозволяє окупити всі витрати за 1,60 року. Впровадження автоматизованої системи для управління замовленнями та обслуговування клієнтів дозволить значно підвищити ефективність роботи

магазину, оптимізувати процеси замовлень і обробки запитів, а також збільшити дохід на 12,000 грн на рік.

1.7. Обґрунтування доцільності проєктування й розроблення інформаційної системи для складського відділу магазину корейської продукції «Смак Кореї»

Проєктування інформаційної системи для складського відділу магазинів «Смак Кореї» є доцільним та необхідним рішенням для підвищення ефективності та злагодженої роботи. Впровадження даної системи дозволить автоматизувати рутинні процеси, які виконувалися вручну, зменшити кількість повторів та помилок при внесенні інформації та забезпечити відстеження даних про товари у реальному часі. Це позитивно вплине на швидкість обробки заявок та швидкість обслуговування магазинів.

Додатково, така інформаційна система забезпечить централізоване зберігання даних, можливість ефективного контролю залишків, автоматичне формування звітів та легку інтеграцію з іншими відділами та працівниками. Завдяки цьому складський відділ зможе уникати надлишків або дефіциту товару, оптимізувати простір на складі та контролювати умови зберігання продукції [30].

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Загальні положення

2.1.1. Найменування системи

«Розроблення інформаційної системи для складського обліку товарів мережі магазинів Смак Кореї»

2.1.2. Результати робіт

Результати робіт із створення системи оформлюються згідно з вимогами ДСТУ на відповідні етапи розроблення. Порядок оформлення і передачі результатів визначається змістом і календарним планом виконання робіт.

2.1.3. Зміни та уточнення

У випадку необхідності на наступних стадіях зі створення системи окремі положення можуть уточнюватися і розвиватися за погодженням із замовником.

2.2. Призначення і цілі створення системи

2.2.1. Призначення системи

Система призначення для автоматизації складських процесів у мережі магазинів Смак Кореї через окрему інформаційну систему. Дана система надає можливості швидкого та зручного доступу до інформації про товар, його кількість, терміни придатності та постачальників. Також було додано функції обліку товарів, при яких кількість відповідно збільшується або зменшується при постачанні чи продажу.

2.2.2. Цілі створення системи

Основна ціль створення системи це зручне та швидке користування складом з метою зменшення ймовірних помилок, повторень та витрат часу і людських ресурсів на роботу, що позитивно вплине на загальну роботу всіх бізнес процесів системи.

Цілі створення:

- оптимізація підрахунків товарів на складі;
- стеження за термінами придатності;
- зменшення помилок та відхилень від постановлених термінів;

- відстеження нестачі товару;
- розстановка товару згідно рекомендованих умов зберігання.

2.3. Характеристика об'єкта автоматизації

2.3.1. Короткі відомості про об'єкт автоматизації

Об'єкт автоматизації це складський відділ мережі магазинів за допомогою інформаційної системи. Мережа магазинів Смак Кореї, як локально відомий постачальник товарів із Південної Кореї, повинна мати злагоджену роботу на складі для подальшого існування та підтримці свого іміджу, як надійних партнерів.

Основні функції, які потребують автоматизації:

- перегляд списку всіх наявних товарів та їх характеристик;
- фільтрування за всіма можливими наявними характеристиками (секціями де вони зберігаються, категоріями, ціною та назвою);
- перегляд списку товарів з певним термінами придатності, по обраним датам;
- перегляд нових замовлень, можливість одразу їх виконати та перегляд історії замовлень;
- відслідковування товарів, вказуючи їх мінімальну та максимальну кількість.

2.4. Вимоги до системи

2.4.1. Вимоги до системи в цілому

2.4.1.1. Вимоги до структури і функціонування системи

Архітектура системи

Система повинна маю клієнт-серверну архітектуру з використанням таких технологій:

- десктопний додаток (клієнт) – Visual Studio C# WinForms;
- серверна частина – SQL server;
- база даних – SQL Server Management Studio.

Діагностування функціонування системи

Система надає діагностику помилок та збоїв користувачу за допомогою короткого сповіщення. Без виправлення даної помилки подальша робота з системою не можлива.

Можливості розвитку і модернізації

Додаток має функції редагування наявних функцій секцій, кількості товару та адаптації змін вимог бізнесу без необхідності повного переписування коду.

Режими функціонування

Функціонування системи забезпечує повне функціонування без наявності підключення до інтернету.

2.4.1.2. Вимоги до користувачів системи

Категорії користувачів.

Інформаційна система передбачає наявність двох користувачів:

1. Комірник – має не повний доступ до системи. Він може переглядати товари, їх характеристики та фільтрувати їх за потреби. Має можливість переглядати замовлення, збирати їх та позначати як виконані та фільтрувати товари за термінами придатності та цінами.

2. Координатор складу – має повний доступ до системи. Окрім всіх функцій які доступні для комірника, отримує дозвіл редагувати ціну та кількість наявних у системі товарів, як особа що відповідає за склад та координує діяльність інших робітників складського відділу.

Вимоги до кваліфікації користувачів

Для ефективного користування додатком користувачі мають:

- мати гарну пам'ять на товари та їх характеристики, щоб ефективно використовувати фільтрацію;

- розуміти принцип дії системи;

- пам'ятати про випадки, в яких система може видати збій чи помилку та вміти їх виправити.

Показники призначення

Характеристики якості системи

Система повинна забезпечувати:

1. Швидкість завантаження головної сторінки після входу в акаунт користувача – не більше 3 секунд.

2. Пошук товару методом введення назви без додаткових кнопок – не більше 0,30 секунди.

3. Швидкість переходу між вкладками додатку – не більше 0,5 секунди.

Показники надійності

Основні показники надійності системи:

- ймовірність безвідмовної роботи – не менше 0,97;
- максимальний час простою системи – не більше 3 годин на місяць.

Забезпечення надійності

Для забезпечення надійності необхідно передбачити

- автоматичне резервне копіювання даних не менше 60 днів;
- моніторинг працездатності системи в режимі реального часу;
- механізм перевірки цілісності даних при їх передачі.

2.4.1.3. Вимоги до безпеки

Для забезпечення безпеки при експлуатації системи потрібно дотримуватись вимог ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011 [27, 26, 22].

2.4.1.4. Вимоги з ергономіки та технічної естетики

Загальні ергономічні і естетичні вимоги до системи повинні відповідати держстандартам ДСТУ 8604:2015, ДСТУ 7298:2013. Інтерфейс ПК додатку повинен відповідати сучасним тенденціям UI/UX дизайну та загальноприйнятим принципам побудови інтерфейсів для настільних комп'ютерів та ноутбуків.

2.4.1.5. Вимоги до експлуатації

Види обслуговування

Види обслуговування системи визначаються у відповідності з ДСТУ EN 13306:2019. Загальні вимоги з експлуатації, технічного обслуговування і ремонту повинні відповідати ДСТУ 3576-97 [25].

Вимоги до технічних засобів

Для коректної роботи інформаційної системи рекомендується мати такі необхідні технічна засоби:

- комп'ютер з операційною системою Windows не нижче версії 10;

- оперативна пам'ять: не менше 4 ГБ;
- вільний простір на жорсткому диску: не менше 500 МБ;
- сервер бази даних Microsoft SQL Server 2016 або новіше.

2.4.1.6. Вимоги до захисту інформації від несанкціонованого доступу

Для надійності збереження інформації необхідно застосувати такі засоби захисту як:

- автентифікація користувачів за допомогою логіна та пароля;
- шифрування персональних даних у базі даних;
- розмежування прав доступу до функцій системи.

2.4.1.7. Вимоги щодо збереження інформації при аваріях

Засоби резервного збереження

Для збереження даних необхідно передбачити:

- щотижневе повне резервне копіювання даних;
- зберігання резервних копій протягом не менше 3 місяців;
- автоматичне відновлення даних після збоїв системи.

Розміщення резервних копій

Резервні копії повинні зберігатися на окремих ПК, доступ до яких має директор системи з різних пристроїв для забезпечення максимальної надійності.

2.4.1.8. Вимоги щодо захисту від впливу зовнішніх діянь

Захист від перепадів напруги

Всі технічні засоби інформаційної системи мають бути захищені від перепадів напруги в електромережі шляхом використання джерел безперебійного живлення з функцією стабілізації напруги

Захист від надзвичайних ситуацій

Приміщення складу має бути обладнано автоматичною системою пожежної сигналізації та системою пожежогасіння. Інформаційна система повинна бути інтегрована з системою оповіщення про надзвичайні ситуації.

2.4.1.9. Вимоги до патентної частини

Під час створення системи патентні дослідження не проводяться. Усі використані бібліотеки та компоненти повинні мати відкриту ліцензію, що дозволяє їх комерційне використання.

2.4.1.10. Вимоги щодо стандартизації і уніфікації

У системі кодування інформації необхідно використовувати UTF-8. Дати, час та інші метрики повинні відповідати міжнародним стандартам.

2.4.2. Вимоги до функцій системи

2.4.2.1. Перелік функцій із зазначенням вхідної та вихідної інформації

Перелік функцій із зазначенням вхідної та вихідної інформації наведено в таблиці 2.1.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

Найменування функції	Вхідна інформація	Вихідна інформація
Автентифікація працівника складу	Логін та пароль працівника	Створений обліковий запис із відповідним рівнем доступу
Приймання товару на склад	Найменування, кількість та термін придатності товару	Оновлені залишки на складі та розміщення продукції згідно умов зберігання
Розміщення товарів на складі	Дані про товар, його характеристики	Призначене місце зберігання (секція)
Пошук товарів на складі	Пошуковий запит (найменування, місце зберігання, виробник, категорія)	Інформація про товар (його місцезнаходження, кількість та термін)
Комплектація замовлень	Номер замовлення, перелік товарів, статус замовлення	Сформоване замовлення, оновлений статус замовлення
Контроль термінів придатності	Параметри перевірки (фільтрація за проміжком дати)	Звіт про товари з критичним терміном придатності

2.4.3. Вимоги до видів забезпечення

2.4.3.1. Вимоги до математичного забезпечення

Система не вимагає спеціального математичного забезпечення для реалізації покладених на неї функцій. Достатньо можливостей обраних технологій.

2.4.3.2. Вимоги до інформаційного забезпечення

Організація інформаційного забезпечення

Інформаційне забезпечення системи повинно містити дані, достатні для виконання всіх покладених на систему функцій. Основні сутності:

- користувачі;
- товари;
- категорії товарів;
- замовлення;
- статуси замовлень;
- виробники товару;
- секції для збегірання.

Захист даних

Слід передбачити захист даних від руйнування при аваріях і порушеннях у енергоживленні системи — використання резервних копій БД.

2.4.3.3. Вимоги до лінгвістичного забезпечення

Мови розроблення

Для розроблення програмних засобів повинні використовуватися:

- SQL/NoSQL для роботи з базою даних;
- C# для роботи з UI.

Організація діалогу

Організація діалогу користувача з системою має будуватися на інтуїтивно зрозумілому інтерфейсі з використанням стандартних компонентів десктопних додатків (кнопки, меню, форми введення тощо).

2.4.3.4. Вимоги до програмного забезпечення

Загальносистемне програмне забезпечення

До загальносистемного забезпечення відноситься:

- операційна система пристроїв складу — Windows 10/11;
- система управління базами даних (СУБД) — реляційна СУБД, SQL SSMS.

Загальні вимоги до системного ПЗ

- мінімальні вимоги до ресурсів технічних засобів;
- максимальна швидкодія;
- надійність та безпека;
- повне задоволення потреб функціональних завдань системи.

Вимоги до технологій розробки

- автоматизація процесів: CI/CD з використанням GitHub Actions або Jenkins;
- база даних: SQL SSMS.

Вимоги до СУБД

- ефективне управління необхідним обсягом і структурою даних;
- швидкість виконання запитів;
- надійність зберігання даних.

Програмні засоби взаємодії з користувачем

- інтуїтивно зрозумілий інтерфейс;
- контроль введення даних з відображенням повідомлень про помилки;
- адаптивний дизайн для різних розмірів екранів.

Вимоги до розробки спеціального ПЗ

- можливість розширення функціоналу;
- оптимізація для роботи на пристроях з різною продуктивністю;
- використання сучасних патернів проектування.

2.4.3.5. Вимоги до технічного забезпечення

Вимоги до обладнання наведено у таблиці 2.2.

Таблиця 2.2. Вимоги до технічного забезпечення системи

	Основні характеристики
Серверна частина	<p>CPU: не менше 4 ядер, частота не менше 2.5 GHz</p> <p>RAM: не менше 8 GB</p> <p>HDD/SSD: не менше 100 GB (рекомендовано SSD; можливість RAID)</p> <p>Підключення до мережі: 1 Gbps Ethernet</p> <p>ОС: Linux Ubuntu 20.04 LTS або новіша</p>
Клієнтська частина	<p>CPU: не менше 2 ядер, частота не менше 2.0 GHz</p> <p>RAM: не менше 2 GB</p> <p>HDD/SSD: не менше 100 GB</p> <p>Монітор: від 15"</p> <p>Пристрої вводу: клавіатура, миша (USB або Bluetooth)</p> <p>ОС: Windows 10 або Linux (дистрибутив за потребою)</p>
Додаткові пристрої (за потреби)	<p>Сканери штрихкодів (1D/2D) — для ідентифікації товару</p> <p>Мобільні пристрої (планшети, смартфони) — для роботи з мобільною версією системи</p> <p>Принтери етикеток та документів — для маркування продукції та друку супровідних документів</p> <p>Мережеве обладнання — маршрутизатори, точки доступу Wi-Fi для покриття складу</p>

Обмін інформацією

Засоби обчислювальної техніки повинні забезпечувати обмін інформації в обсягах, зазначених у вимогах до інформаційного забезпечення.

2.4.3.6. Вимоги до метрологічного забезпечення

Інформаційна система складського відділу може інтегруватися з окремими вимірювальними приладами, що використовуються для забезпечення точного обліку товарів та контролю умов зберігання. До такого обладнання належать:

- електронні ваги — для зважування товарів при прийманні та інвентаризації;

- сканери габаритів – для автоматичного визначення розмірів пакування товару.

2.4.3.7. Вимоги до організаційного забезпечення

Стандарти організаційного забезпечення

Організаційне забезпечення системи розробляється відповідно до вимог державного стандарту по АСУП.

Кадрове забезпечення

При впровадженні системи не передбачається збільшення штатної чисельності підприємства. Для роботи з системою має бути призначений відповідальний співробітник.

Вимоги до функціонування системи

- наказом керівника визначається список співробітників, які мають доступ до системи;
- контроль і прийняття рішень при аварійних ситуаціях здійснює відповідальний за систему;
- регулярне навчання співробітників роботі з системою;
- дотримання правил безпеки та конфіденційності інформації.

2.5. Склад і зміст робіт по створенню системи

2.5.1. Стадії створення системи і терміни виконання робіт

Стадії створення інформаційної системи та чіткі терміни робіт зазначені у таблиці 2.3.

Таблиця 2.3. Найменування робіт при створенні системи

Етапи проекту	Початок	Тривалість	Затримка	Кінець
Аналіз вимог	28.04.2025	6	0	30.04.2025
Проектування	01.05.2025	4	+1	04.05.2025
Розробка	05.05.2025	8	+2	12.05.2025
Тестування	13.05.2025	3	0	15.05.2025
Впровадження	16.05.2025	4	0	19.05.2025
Навчання персоналу	20.05.2025	4	-1	23.05.2025
Підтримка та супровід	24.05.2025	7	-2	30.05.2025

2.5.2. Діаграма Ганта

Проект зазнав незначних затримок, загалом зміщених на 4 дні, через зміни в етапах. Проте ці затримки не мали критичного впливу на загальний термін завершення проекту, який зберігся в межах прийнятних строків. Рекомендується посилити контроль за етапами розробки та тестування, а також уважніше планувати терміни для навчання персоналу, щоб уникнути подібних затримок у майбутньому.

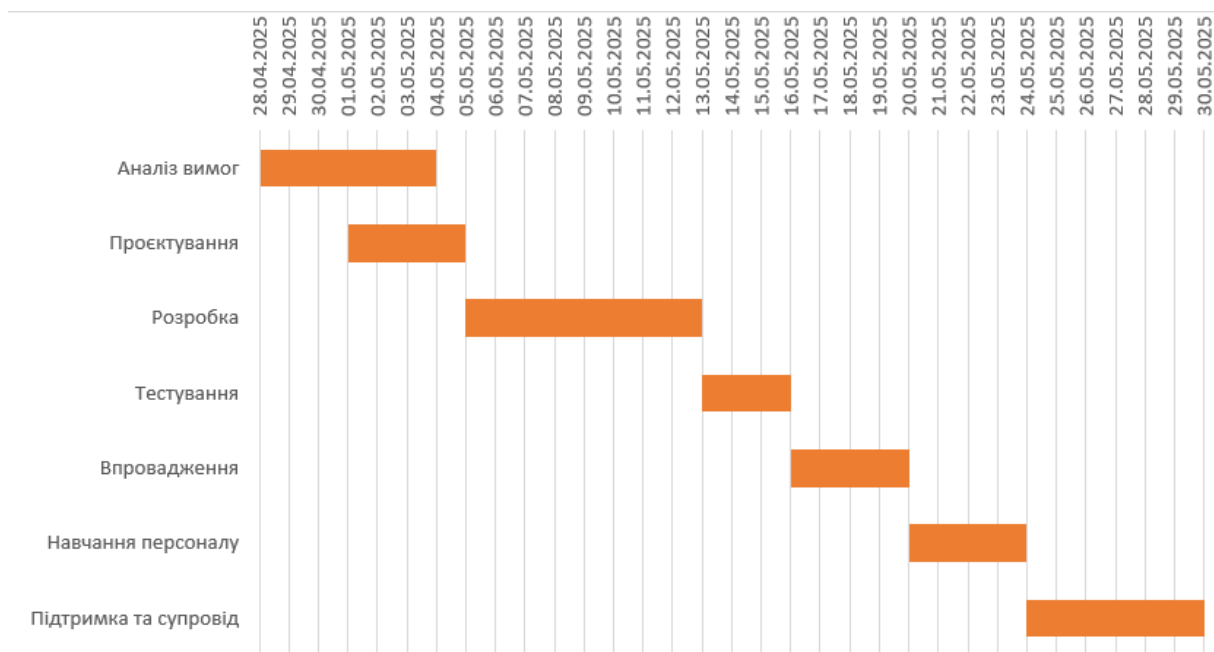


Рисунок 2.1 – Діаграма Ганта

2.6. Порядок контролю і приймання системи

2.6.1. Загальні умови приймання

Система вводиться у діючих магазинах корейської продукції Смак Кореї. При введенні в дію система повинна пройти приймальні випробування згідно з ДСТУ 3974-2000 [21].

2.6.2. Випробування системи

Випробування для визначення працездатності і рішення про можливість приймання системи в дослідну експлуатацію проводять працівники складу, а саме комірник та координатор складу, оскільки само вони будуть користуватися цією системою та повинні зрозуміти чи є вона для них зручною та ефективною.

2.6.3. Дослідна експлуатація

Здача в дослідну експлуатацію здійснюється на основі технічного завдання та інструкції користувача. За результатами дослідної експлуатації формується перелік доробок і рекомендовані строки їх виконання.

2.6.4. Оформлення результатів

Введення в дію системи оформлюється актом здачі-прийому, який підписується директором, після узгодження з робітниками складу, та розробником системи.

2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

Для успішного впровадження інформаційної системи для складського відділу мережі магазинів «Смак Кореї» було проведено такі підготовчі роботи:

- проведення укомплектування технічних засобів згідно з вимогами до технічного забезпечення;
- організція навчання персоналу роботі з адміністративною панеллю та основними функціями системи;
- призначення відповідальних осіб за підтримку та експлуатацію системи;
- забезпечення інтеграції із внутрішніми системами обліку та управління товарами;

- організація тестової групи користувачів для бета-тестування додатку;
- проведення дослідної експлуатації і введення системи в дію на основі результатів тестування.

2.8. Вимоги до документації

2.8.1. Перелік документації

На систему розробляється комплекс документації у складі:

1. Технічне завдання на розробку мобільного додатку
2. Технічний проект, що включає:
 - пояснювальну записку;
 - опис архітектури системи;
 - модель даних;
 - керівництво користувачів.
3. Програма та методика випробувань
4. Акт приймально-здавальних випробувань
5. Експлуатаційна документація:
 - інструкція з установки інформаційної системи;
 - інструкція з налаштування.

2.8.2. Вимоги до оформлення документації

Документація на систему розробляється у відповідності з вимогами Державних стандартів:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
- ДСТУ 3974-2000 Система розроблення та поставлення продукції на виробництво. Правила виконання дослідно-конструкторських робіт [21];
- ДСТУ ISO/IEC 25051:2016 Інженерія програмного забезпечення. Вимоги до якості готового до використання програмного продукту (RUSP) і інструкції щодо тестування;
- ДСТУ ISO/IEC/IEEE 26511:2016 Системна та програмна інженерія. Вимоги до менеджерів інформації користувача;

- ДСТУ ISO/IEC/IEEE 26512:2016 Системна та програмна інженерія. Вимоги до укладачів інформації користувача.

2.8.3. Порядок внесення змін до документації

Внесення змін до документації здійснюється у відповідності з ДСТУ ГОСТ 2.503:2013 Єдина система конструкторської документації. Правила внесення змін.

2.9. Джерела розробки

2.9.1. Нормативні документи

При розробленні технічного завдання на систему використано наступні документи:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
- ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво [2];
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом [4];
- ДСТУ ISO/IEC 27001:2015 Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. Вимоги [3];
- ДСТУ ISO/IEC TR 29110-5-1-2:2015 Інженерія програмного забезпечення. Профілі життєвого циклу для малих організацій;
- ДСТУ ISO/IEC/IEEE 29119-1:2015 Тестування програмного забезпечення. Частина 1. Поняття і визначення;
- ДСТУ ISO 9241-210:2019 Ергономіка взаємодії людина-система. Частина 210. Людиноцентричне проектування інтерактивних систем.

РОЗДІЛ 3. РОЗРОБЛЕННЯ ПРОГРАМНОГО ПРОДУКТУ

3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту

Для розробки інформаційної системи складського відділу мережі магазинів «Смак Кореї» було обрано такий технологічний стек, який міг би бути доступним, не складним у розумінні та ефективним для вирішення проблем відділу.

Засоби розробки інформаційної системи:

1. Visual Studio з C# та WinForms – це основне середовище, яке використовувалося для розробки та фреймворку, щоб створити зручний десктопний додаток. WinForms забезпечує швидке створення графічного інтерфейсу за допомогою зручного drag-and-drop редактора [17].

2. СУБД SQL Server – це серверна частина для обробки запитів та управління даними. Це реляційна система управління базами даних, яка має високу продуктивність та надійність [18].

3. SQL Server Management Studio – це інструмент для адміністрування бази даних, швидкого та зручного створення таблиць і запитів та управління SQL Server [19].

Для інтерфейсу та функціональності було обрано:

1. WinForms компоненти, які використовують стандартні компоненти WinForms, такі як кнопки, текстові поля, таблиці, для створення зрозумілого та функціонального інтерфейсу [20].

2. Власні елементи керування для забезпечення єдиного стилю та адаптації під бажання та зручність користувача. Також були розроблені кастомні елементи, такі як спеціалізовані форми для введення даних.

Управління даними та взаємодія між сервером та клієнтом:

1. ADO.NET – використовується для взаємодії між десктопним додатком та SQL Server та забезпечує простий доступ до даних через підключення до бази, обробку результатів та виконання SQL-запитів.

2. LINQ to SQL – застосовується для спрощення роботи з даними, дозволяє писати запити до бази без проблем та помилок у стилі C#, що робить код більш читабельним та зменшує ймовірність помилок.

Обґрунтування вибору технологій:

1. Visual Studio з C# та WinForms було обрано через:

- простоту створення десктопних додатків для Windows;
- велику кількість вбудованих інструментів у Visual Studio;
- підтримку платформи Microsoft.

2. SQL Server було обрано через:

- високу продуктивність при роботі з великою кількістю даних;
- надійність та безпеку, включаючи важливі для нас документи резервного копіювання;
- можливість масштабування для майбутнього розширення системи та її покращення.

3. SQL Server Management Studio було обрано через:

- інтуїтивно зрозумітий інтерфейс для адміністрування бази даних;
- можливість створення та оптимізації SQL-запитів;
- безкоштовні дрібні деталі у складі SQL.

3.2. Проєктування та створення бази даних

Логічна модель бази даних, наведена у Додатку В, яка розроблена для збереження інформації щодо складських операцій та обробки замовлень з використанням засобу ERwin Data Modeler. Логічна модель бази даних включає дев'ять сутностей: `supplies` (поставки), `supply_details` (деталі поставок), `manufacturers` (виробники), `products` (продукти), `categories` (категорії продуктів), `sections` (секції складу), `stock` (запаси), `orders` (замовлення) та `order_details` (деталі замовлень). Кожна сутність пов'язана через зовнішні ключі, що забезпечують цілісність даних: наприклад, `supply_details` пов'язана з `supplies` і `products`, а `stock` — з `products` і `sections`.

Фізична модель бази даних, наведена у Додатку Г, і складеться з: supplies, supply_details, manufacturers, categories, products, stock, sections, orders, order_details, що забезпечують детальне представлення кожного етапу руху товару — від постачання до виконання замовлення.

Таблиці supplies та supply_details відповідають за облік постачань і містять інформацію про дати, постачальників, ціни, кількість і секції зберігання. Таблиця products пов'язана з категоріями (categories) та виробниками (manufacturers) і містить опис товарів. Таблиця stock відображає поточний стан складу: у якій секції зберігається кожен товар, у якій кількості та до якого терміну придатності. Таблиця sections фіксує параметри зон складу, зокрема температурні умови. Модуль замовлень представлений таблицями orders і order_details, що містять інформацію про склад і статус кожного замовлення, а також його наповнення.

При проектуванні структури бази даних враховані принципи нормалізації даних: кожна таблиця приведена до 3NF форми, використовуються зовнішні ключі для забезпечення цілісності даних, кожна таблиця має чітке призначення, а структура дозволяє ефективно керувати як внутрішніми процесами складу, так і зовнішніми замовленнями.

Фізична модель бази даних була згенерована до СУБД Microsoft SQL Server, яка забезпечує зручне зберігання та обробку даних для додатку складського відділу. Вона має таблиці для категорій, такий як локшина, снеки або соуси, також наявні таблиці про виробників, секцію зберігання продукту, замовлень та залишків товару. Розроблена структура бази даних дозволяє ефективно зберігати інформацію для упарвління логістичними процесами магазину.

Секції зберігання – «сухий склад» та «прохолодний склад» поділені за максимально можливими температурами зберігання для продуктів. Замовлення мають статус «нове» та показують список товарів які туди входять, а зміна статусу замовлення на «виконане» призводить до перерахунку залишків товару.

Структура бази даних згенерована в Microsoft SQL Server наведена у Додатку Д і складається з наведених таблиць:

1. Categories (Категорії) – зберігає інформацію про категорії товарів:

- category_id – унікальний ідентифікатор категорії в списках;
- name – назва категорії;

Використовується для класифікації продуктів.

2. Manufacturers (Виробники) – зберігає інформацію про виробників товарів:

- manufacturer_id – унікальний ідентифікатор виробника;
- name – назва виробника;

Використовується для зв'язку з продуктами та постачальниками.

3. Products (Продукти) – зберігає інформацію про товари в каталозі:

- product_id – унікальний ідентифікатор продукту;
- name – назва продукту;
- description – опис продукту;
- category_id – ідентифікатор категорії (зв'язок із таблицею categories);
- manufacturer_id – ідентифікатор виробника (зв'язок із таблицею manufacturers);
- max_storage_temperature – максимальна температура зберігання;

Використовується для управління асортименту товарів.

4. Supplies (Поставки) – зберігає інформацію про поставки товарів:

- supply_id – унікальний ідентифікатор поставки;
- manufacturer_id – ідентифікатор виробника (зв'язок із таблицею manufacturers);
- creation_date – дата створення поставки;
- status – статус поставки;
- received_date – дата отримання поставки;

Використовується для відстеження поставок від виробників.

5. Supply_Details (Деталі поставки) – зберігає деталі кожної поставки:

- supply_detail_id – унікальний ідентифікатор деталі поставки;
- supply_id – ідентифікатор поставки (зв'язок із таблицею supplies);
- product_id – ідентифікатор продукту (зв'язок із таблицею products);
- quantity – кількість товару;
- price – ціна товару;
- manufacturing_date – дата виробництва;
- expiration_date – дата закінчення терміну придатності;
- section_id – ідентифікатор секції (зв'язок із таблицею sections) ;

Використовується для деталізації вмісту поставок.

6. Sections (Секції) – зберігає інформацію про секції складу:

- section_id – унікальний ідентифікатор секції;
- section_name – назва секції;
- temperature – температура зберігання;

Використовується для організації зберігання товарів на складі.

7. Stock (Запаси) – зберігає інформацію про наявність товарів на складі:

- stock_id – унікальний ідентифікатор запасу;
- product_id – ідентифікатор продукту (зв'язок із таблицею products);
- section_id – ідентифікатор секції (зв'язок із таблицею sections);
- quantity – кількість товару на складі;
- price – ціна товару;
- manufacturing_date – дата виробництва;
- expiration_date – дата закінчення терміну придатності;

Використовується для відстеження залишків продукції.

8. Orders (Замовлення) – зберігає інформацію про оформлені замовлення:

- order_id – унікальний ідентифікатор замовлення;
- creation_date – дата створення замовлення;
- status – статус замовлення;

- `completion_date` – дата завершення замовлення;
- `destination` – пункт призначення;

Використовується для управління процесами замовлень.

9. `Order_Details` (Деталі замовлення) – зберігає деталі кожного замовлення:

- `detail_id` – унікальний ідентифікатор деталі замовлення;
- `order_id` – ідентифікатор замовлення (зв'язок із таблицею `orders`);
- `product_id` – ідентифікатор продукту (зв'язок із таблицею `products`);
- `quantity` – кількість товару;
- `price` – ціна товару;

Використовується для детального опису товару та їх характеристик.

3.3.Реалізація функцій системи

3.3.1. Авторизація користувачів

Для забезпечення персоналізованої роботи з додатком реалізовано функцію авторизації. Цей функціонал у формі `loginForm`, яка надає наступні можливості:

1. Автентифікація користувачів – метод обробки події `buttonLogin_Click` перевіряє введені дані (логін і пароль) та порівнює їх із даними, збереженими у файлі `user.txt`:

- `buttonLogin_Click` – зчитує логін і пароль із текстових полів `loginBox` і `passBox`, перевіряє їх на відповідність записам у файлі, після чого перенаправляє користувача на форму `adminForm` (для адміністратора) або `MainMenu` (для звичайного користувача).

- **Перевірки:** перевіряє, чи заповнені поля, чи мають вони щонайменше 3 символи, а також чи існує користувач і чи правильний пароль.

2. Вихід із програми – метод обробки події `closeLogin_Click` забезпечує завершення роботи додатку:

- `closeLogin_Click` – викликає `Application.Exit()` для закриття програми.

3. Інтерактивність інтерфейсу – методи обробки подій забезпечують додаткову функціональність:

- `eye_MouseHover` та `eye_MouseLeave` – дозволяють показувати або приховувати пароль у полі `passBox`, змінюючи іконку ока та параметр `UseSystemPasswordChar`.
 - `minimize_Click` – згортає форму до панелі завдань через `WindowState = FormWindowState.Minimized`.
 - `panell1_MouseMove` та `label1_MouseMove` – дозволяють переміщувати форму, утримуючи мишкою панель або заголовок.
4. Збереження стану – авторизація зберігається у файлі `user.txt`, звідки зчитуються дані для перевірки. У разі успішного входу додаток визначає тип користувача (адміністратор або звичайний) і відкриває відповідну форму.

3.3.2. Завантаження, фільтрація та відображення товарів

Цей функціонал реалізовано у формі `adminForm`, а методи `LoadProducts` та пов'язані обробники подій дозволяють переглядати каталог товарів із фільтрацією:

- `LoadProducts` – виконує динамічний SQL-запит із параметрами для фільтрації за назвою (`nameFilter.Text`), категоріями (`categoryCheckBoxes`), виробниками (`manufacturerCheckBoxes`) та секціями (`sectionCheckBoxes`), відображаючи результати в `dataGrid` із колонками: `product_id`, `product_name`, `description`, `category`, `manufacturer`, `max_storage_temperature`;

- `nameFilter_TextChanged` – оновлює список товарів при зміні тексту в фільтрі назви.

Програмний код завантаження, фільтрації та відображення товару наведено у [Додатку Е1](#).

3.3.3. Завантаження, фільтрація та відображення замовлень

1. Завантаження, фільтрація та відображення замовлень – методи `LoadOrders`, `LoadOrderDetails` та пов'язані обробники подій забезпечують керування замовленнями:
 - `LoadOrders` – фільтрує замовлення за номером (`orderIdFilter.Text`), датою створення (`creationDateFrom`, `creationDateTo`), статусом (`statusFilter`) та

сумою (totalAmountFrom, totalAmountTo), розраховуючи загальну суму (total_amount), і відображає дані в dataGridOrders;

- LoadOrderDetails – завантажує деталі вибраного замовлення в dataGridOrderDetails із колонками: detail_id, product_name, quantity, price;
- orderIdFilter_TextChanged, creationDateFrom_ValueChanged, creationDateTo_ValueChanged, statusFilter_SelectedIndexChanged, totalAmountFrom_TextChanged, totalAmountTo_TextChanged – оновлюють список замовлень при зміні фільтрів.

2. Виконання замовлення – метод ExecuteOrder обробляє виконання замовлення:

- Перевіряє наявність достатньої кількості товару в запасах, оновлює статус замовлення на "Виконано", дату завершення та зменшує кількість у запасах через транзакцію, викликаний подією executeButton_Click.

Програмний код операцій із замовленнями: Додаток E2.

3.3.4. Завантаження, фільтрація та відображення залишків

Завантаження, фільтрація та відображення запасів – методи LoadStock та пов'язані обробники подій забезпечують перегляд залишків на складі:

- LoadStock – фільтрує за назвою (stockNameFilter.Text), категоріями (stockCategoryCheckBoxes), виробниками (stockManufacturerCheckBoxes), секціями (stockSectionCheckBoxes), ціною (priceFrom, priceTo), терміном придатності (expirationFrom, expirationTo) та кількістю (quantityFrom, quantityTo), розраховуючи резервовані (reserved) та доступні (available) товари, і відображає дані в dataGridStock;
- stockNameFilter_TextChanged, priceFrom_TextChanged, priceTo_TextChanged, expirationFrom_ValueChanged, expirationTo_ValueChanged, quantityFrom_TextChanged, quantityTo_TextChanged – оновлюють список запасів при зміні фільтрів із валідацією введених значень (наприклад, ціна в межах minPrice і maxPrice).

Програмний код реалізації цих функцій наведено у Додатку E3.

3.3.5. Редагування та видалення запасів

Редагування та видалення запасів – методи `editButton_Click`, `deleteButton_Click`, `dataGridStockEdit_SelectionChanged`, `saveButton_Click` та `refreshButton_Click` дозволяють керувати запасами:

- `dataGridStockEdit_SelectionChanged` – заповнює поля `editPrice` і `editQuantity` при виборі рядка в `dataGridStockEdit`;
- `editButton_Click` – оновлює ціну та кількість у вибраному записі після перевірки (значення від 0 до 10000);
- `deleteButton_Click` – видаляє вибраний рядок із `dataGridStockEdit`;
- `saveButton_Click` – зберігає зміни в базі даних через транзакцію, оновлюючи таблицю `stock`;
- `refreshButton_Click` – перезавантажує дані для редагування через `LoadStockEdit`.

Програмний код даних функцій наведено у [Додатку Е4](#).

3.4. Інструкція користувача

Після запуску додатку з'являється вікно входу, яке зображено на рисунку 3.1, де потрібно ввести свої дані для входу. Важлива примітка: координатор складу та комірник мають різні аккаунти та їх доступ до редагування даних відрізняється обмеженістю.

- На екрані вводу введіть свій логін та пароль у відповідні поля.
- Натисніть кнопку «Увійти».

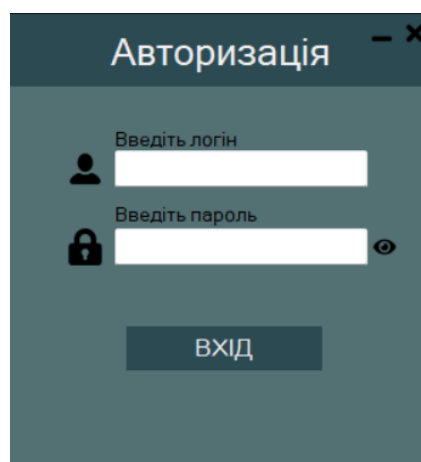


Рисунок 3.1 – Вікно входу

- Головне меню «Всі продукти»

Після успішної авторизації на екрані відображається головне вікно програми, яке зображено на рисунку 3.2, для відображення списку товарів і їх характеристик, та здійснення їх пошуку та фільтрації.

Назва:	ID	Назва товару	Опис	Категорія	Виробник	Макс. температура
	1	Samyang Hot Chicken R...	Гостра локшина швидкого пр...	Локшина	Samyang ...	25,00
	2	Samyang Hot Chicken R...	Гостра локшина з сирним сма...	Локшина	Samyang ...	25,00
	3	Samyang Buldak 2x Spicy	Дуже гостра локшина швидко...	Локшина	Samyang ...	25,00
	4	Nongshim Shin Ramyun	Класична гостра локшина шв...	Локшина	Nongshim	25,00
	5	Nongshim Neoguri Seafo...	Локшина зі смаком морепрод...	Локшина	Nongshim	25,00
	6	Nongshim Chapaggetti	Локшина з соусом чорної сої	Локшина	Nongshim	25,00
	7	Ottogi Jin Ramen Mild	Локшина з м'яким смаком	Локшина	Ottogi	25,00
	8	Ottogi Cheese Ramen	Локшина з ніжним сирним соу...	Локшина	Ottogi	25,00
	9	Lotte Milkis Original	Газований молочний напій	Hanoi	Lotte Conf...	15,00
	10	Lotte Chilsung Cider	Газований напій зі смаком ли...	Hanoi	Lotte Conf...	15,00
	11	Lotte Sac Sac Grape	Напій зі смаком винограду	Hanoi	Lotte Conf...	15,00
	12	Ottogi Bacchus Energy ...	Енергетичний напій	Hanoi	Ottogi	15,00
	13	Lotte Pepero Original	Шоколадні палички з печивом	Снеки	Lotte Conf...	25,00
	14	Lotte Pepero Almond	Палички з мигдалем і шокола...	Снеки	Lotte Conf...	25,00
	15	Lotte Choco Pie	Шоколадне тістечко з зефіром	Снеки	Lotte Conf...	25,00
	16	Nongshim Shrimp Crack...	Креветкові чіпси	Снеки	Nongshim	25,00
	17	Nongshim Onion Rings	Цибулеві кільця	Снеки	Nongshim	25,00
	18	Samyang Buldak Hot Sa...	Гострий соус для локшини	Соуси	Samyang ...	25,00
	19	Samyang Carbonara Sa...	Соус зі смаком карбонара	Соуси	Samyang ...	25,00
	20	Ottogi Gochujang Paste	Паста гострого перцю	Соуси	Ottogi	25,00
	21	Ottogi Soy Sauce	Класичний соевий соус	Соуси	Ottogi	25,00

Рисунок 3.2. – Вкладка «Всі продукти»

- Одразу на головній сторінці можна переглянути найменування, опис, категорію, виробника та рекомендовану температуру зберігання.

- Для пошуку товару на назвою треба почати вводити найменування товару у поле «Назва», тоді у списку товарів будуть відображатися лише товари з введеним найменуванням. Результат пошуку зображений на рисунку 3.3.

Назва:	ID	Назва товару	Опис	Категорія	Виробник	Макс. температура
Pe	11	Lotte Sac Sac Grape	Напій зі смаком винограду	Hanoi	Lotte Conf...	15,00
	13	Lotte Pepero Original	Шоколадні палички з печивом	Снеки	Lotte Conf...	25,00
	14	Lotte Pepero Almond	Палички з мигдалем і шокола...	Снеки	Lotte Conf...	25,00

Рисунок 3.3 – Пошук товару за назвою

• Здійснюється фільтрація по одному або декільком виробникам, просто поставивши галочку біля одного або декількох найменувань. У списку товарів будуть відображені лише відфільтровані одиниці. Результат фільтрації показаний на рисунку 3.4.

ID	Назва товару	Опис	Категорія	Виробник	Макс. температура
1	Samyang Hot Chicken R...	Гостра локшина швидкого пр...	Локшина	Samyang ...	25,00
2	Samyang Hot Chicken R...	Гостра локшина з сирним сма...	Локшина	Samyang ...	25,00
3	Samyang Buldak 2x Spicy	Дуже гостра локшина швидко...	Локшина	Samyang ...	25,00
9	Lotte Milkis Original	Газований молочний напій	Напої	Lotte Conf...	15,00
10	Lotte Chilsung Cider	Газований напій зі смаком ли...	Напої	Lotte Conf...	15,00
11	Lotte Sac Sac Grape	Напій зі смаком винограду	Напої	Lotte Conf...	15,00
13	Lotte Pepero Original	Шоколадні палички з печивом	Снеки	Lotte Conf...	25,00
14	Lotte Pepero Almond	Палички з мигдалем і шокола...	Снеки	Lotte Conf...	25,00
15	Lotte Choco Pie	Шоколадне тістечко з зефіром	Снеки	Lotte Conf...	25,00
18	Samyang Buldak Hot Sa...	Гострий соус для локшини	Соуси	Samyang ...	25,00
19	Samyang Carbonara Sa...	Соус зі смаком карбонара	Соуси	Samyang ...	25,00

Рисунок 3.4 – Фільтрація за виробником

• Фільтрації товарів за умовами зберігання, обираючи з трьох можливих варіантів. Результат фільтрації показаний на рисунку 3.5.

ID	Назва товару	Опис	Категорія	Виробник	Макс. температура
1	Samyang Hot Chicken R...	Гостра локшина швидкого пр...	Локшина	Samyang ...	25,00
2	Samyang Hot Chicken R...	Гостра локшина з сирним сма...	Локшина	Samyang ...	25,00
3	Samyang Buldak 2x Spicy	Дуже гостра локшина швидко...	Локшина	Samyang ...	25,00
4	Nongshim Shin Ramyun	Класична гостра локшина шв...	Локшина	Nongshim	25,00
5	Nongshim Neoguri Seafo...	Локшина зі смаком морепрод...	Локшина	Nongshim	25,00
6	Nongshim Chapaghetti	Локшина з соусом чорної сої	Локшина	Nongshim	25,00
7	Ottogi Jin Ramen Mild	Локшина з м'яким смаком	Локшина	Ottogi	25,00
8	Ottogi Cheese Ramen	Локшина з ніжним сирним соу...	Локшина	Ottogi	25,00
9	Lotte Milkis Original	Газований молочний напій	Напої	Lotte Conf...	15,00
10	Lotte Chilsung Cider	Газований напій зі смаком ли...	Напої	Lotte Conf...	15,00
11	Lotte Sac Sac Grape	Напій зі смаком винограду	Напої	Lotte Conf...	15,00
12	Ottogi Bacchus Energy ...	Енергетичний напій	Напої	Ottogi	15,00
13	Lotte Pepero Original	Шоколадні палички з печивом	Снеки	Lotte Conf...	25,00
14	Lotte Pepero Almond	Палички з мигдалем і шокола...	Снеки	Lotte Conf...	25,00
15	Lotte Choco Pie	Шоколадне тістечко з зефіром	Снеки	Lotte Conf...	25,00
16	Nongshim Shrimp Crack...	Креветкові чіпси	Снеки	Nongshim	25,00
17	Nongshim Onion Rings	Цибулеві кільця	Снеки	Nongshim	25,00
18	Samyang Buldak Hot Sa...	Гострий соус для локшини	Соуси	Samyang ...	25,00
19	Samyang Carbonara Sa...	Соус зі смаком карбонара	Соуси	Samyang ...	25,00
20	Ottogi Gochujang Paste	Паста гострого перцю	Соуси	Ottogi	25,00
21	Ottogi Soy Sauce	Класичний соевий соус	Соуси	Ottogi	25,00

Рисунок 3.5 – Фільтрація за умовами зберігання

• На вкладці «Залишки на складі», можна побачити найменування товарів, що знаходяться на складі, їх характеристики, яких нема в головному меню. Вигляд вікна зображений на рисунку 3.6.

ID залишку	Назва товару	Категорія	Виробник	Секція	Термін придатності	Кількість	Ціна	Резерв	Доступні
9	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.01.2026 0:...	100	45,00	2	98
10	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.04.2026 0:...	50	40,00	2	48
11	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.02.2026 0:...	80	45,00	2	78
12	Samyang Buldak 2x Spicy	Локшина	Samyang ...	Сухий скл...	01.03.2026 0:...	60	50,00	0	60
13	Nongshim Shin Ramyun	Локшина	Nongshim	Сухий скл...	15.01.2026 0:...	120	42,00	3	117
14	Nongshim Shin Ramyun	Локшина	Nongshim	Сухий скл...	15.04.2026 0:...	70	38,00	3	67
15	Nongshim Neoguri Seafo...	Локшина	Nongshim	Сухий скл...	15.02.2026 0:...	90	48,00	1	89
16	Nongshim Chapaggetti	Локшина	Nongshim	Сухий скл...	15.03.2026 0:...	100	40,00	0	100
17	Ottogi Jin Ramen Mild	Локшина	Ottogi	Сухий скл...	20.01.2026 0:...	110	40,00	2	108
18	Ottogi Cheese Ramen	Локшина	Ottogi	Сухий скл...	20.02.2026 0:...	85	45,00	0	85
19	Lotte Milkis Original	Напої	Lotte Conf...	Прохолод...	01.09.2025 0:...	200	30,00	3	197
20	Lotte Chilsung Cider	Напої	Lotte Conf...	Прохолод...	10.09.2025 0:...	150	28,00	2	148
21	Lotte Sac Sac Grape	Напої	Lotte Conf...	Прохолод...	15.09.2025 0:...	180	32,00	3	177
22	Ottogi Bacchus Energy ...	Напої	Ottogi	Прохолод...	20.09.2025 0:...	140	35,00	0	140
23	Lotte Pepero Original	Снеки	Lotte Conf...	Сухий скл...	01.02.2026 0:...	90	60,00	1	89
24	Lotte Pepero Almond	Снеки	Lotte Conf...	Сухий скл...	10.02.2026 0:...	70	65,00	0	70
25	Lotte Choco Pie	Снеки	Lotte Conf...	Сухий скл...	15.02.2026 0:...	120	70,00	1	119
26	Nongshim Shrimp Crack...	Снеки	Nongshim	Сухий скл...	01.03.2026 0:...	100	55,00	2	98

Рисунок 3.6 – Вікно «Залишки на складі»

• Категорію товарів, наявних на складі, їх виробника, ціну, секцію де вони знаходяться та термін придатності. «Кількість товару» - це загальна кількість товару на складі, «Резерв» показує скільки товарів входить до замовлень клієнтів, «Доступні» - фактична кількість товарів яка буде після виконання замовлення.

• Фільтрувати товари, як і у головному меню, можна за виробниками, секціями та категоріями, а також присутній пошук за назвою в окремому полі.

• Ввівши мінімальну та максимальну ціну в поля «Ціна», «Від» та «До», товари будуть відповідно відфільтровуватися згідно введеним даним, як зображено на рисунку 3.7.

Система управління складом (режим адміна)

Всі продукти | Залишки на складі | **Замовлення** | **Замовлення постачальникам** | Редагування залишків

Назва: Категорії: Локшина Наної Снеки Соуси Виробники: Samyang Foods Nongshim Ціна: Від: До: Секція: Сухий склад Прохолодний склад Холодильник Склад середньої темпе Термін придатності: Від: До: Кількість: Від: До:

ID залишку	Назва товару	Категорія	Виробник	Секція	Термін придатності	Кількість	Ціна	Резерв	Доступні
9	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.01.2026 0:...	100	45,00	2	98
10	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.04.2026 0:...	50	40,00	2	48
11	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.02.2026 0:...	80	45,00	2	78
12	Samyang Buldak 2x Spicy	Локшина	Samyang ...	Сухий скл...	01.03.2026 0:...	60	50,00	0	60
13	Nongshim Shin Ramyun	Локшина	Nongshim	Сухий скл...	15.01.2026 0:...	120	42,00	3	117
14	Nongshim Shin Ramyun	Локшина	Nongshim	Сухий скл...	15.04.2026 0:...	70	38,00	3	67
15	Nongshim Neoguri Seafo...	Локшина	Nongshim	Сухий скл...	15.02.2026 0:...	90	48,00	1	89
16	Nongshim Chapaggetti	Локшина	Nongshim	Сухий скл...	15.03.2026 0:...	100	40,00	0	100
17	Ottogi Jin Ramen Mild	Локшина	Ottogi	Сухий скл...	20.01.2026 0:...	108	40,00	0	108
18	Ottogi Cheese Ramen	Локшина	Ottogi	Сухий скл...	20.02.2026 0:...	85	45,00	0	85
19	Lotte Milkis Original	Наної	Lotte Conf...	Прохолод...	01.09.2025 0:...	200	30,00	3	197
20	Lotte Chilsung Cider	Наної	Lotte Conf...	Прохолод...	10.09.2025 0:...	150	28,00	2	148
21	Lotte Sac Sac Grape	Наної	Lotte Conf...	Прохолод...	15.09.2025 0:...	180	32,00	3	177
22	Ottogi Bacchus Energy ...	Наної	Ottogi	Прохолод...	20.09.2025 0:...	140	35,00	0	140
23	Lotte Pepero Original	Снеки	Lotte Conf...	Сухий скл...	01.02.2026 0:...	90	60,00	1	89
24	Lotte Pepero Almond	Снеки	Lotte Conf...	Сухий скл...	10.02.2026 0:...	70	65,00	0	70
25	Lotte Choco Pie	Снеки	Lotte Conf...	Сухий скл...	15.02.2026 0:...	120	70,00	1	119
26	Nongshim Shrimp Crack...	Снеки	Nongshim	Сухий скл...	01.03.2026 0:...	100	55,00	2	98

Рисунок 3.7 – Фільтрація по ціні

• Для зручного пошуку товарів з критичним терміном придатності було додано поле «Термін придатності», де можна відфільтрувати товари за датами «Від», наприклад, початку місяця та «До» його кінця. Результат фільтрації зображено на рисунку 3.8.

Система управління складом (режим адміна)

Всі продукти | Залишки на складі | **Замовлення** | **Замовлення постачальникам** | Редагування залишків

Назва: Категорії: Локшина Наної Снеки Соуси Виробники: Samyang Foods Nongshim Ціна: Від: До: Секція: Сухий склад Прохолодний склад Холодильник Склад середньої темпе Термін придатності: Від: До: Кількість: Від: До:

ID залишку	Назва товару	Категорія	Виробник	Секція	Термін придатності	Кількість	Ціна	Резерв	Доступні
19	Lotte Milkis Original	Наної	Lotte Conf...	Прохолод...	01.09.2025 0:...	200	30,00	3	197
20	Lotte Chilsung Cider	Наної	Lotte Conf...	Прохолод...	10.09.2025 0:...	150	28,00	2	148
21	Lotte Sac Sac Grape	Наної	Lotte Conf...	Прохолод...	15.09.2025 0:...	180	32,00	3	177
22	Ottogi Bacchus Energy ...	Наної	Ottogi	Прохолод...	20.09.2025 0:...	140	35,00	0	140

Рисунок 3.8 – Фільтрація по терміну придатності

- Вкладка «Замовлення»

У даній вкладці знаходиться інформація про нові та виконані замовлення. Вигляд вкладки зображено на рисунку 3.9.

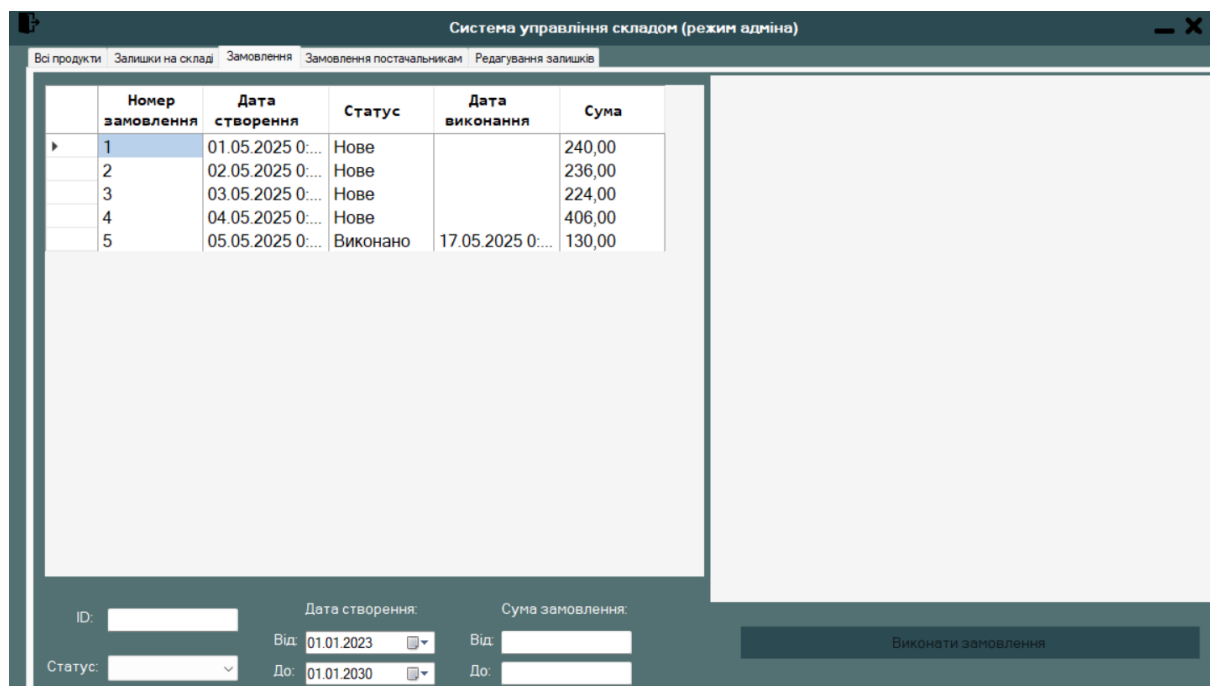


Рисунок 3.9 – Вікно «Замовлення»

• У лівому вікні можна переглянути наявні замовлення магазину, а обравши одне з них, інформація про склад замовлення з'являється у правому вікні, так як зображено на рисунку 3.10.

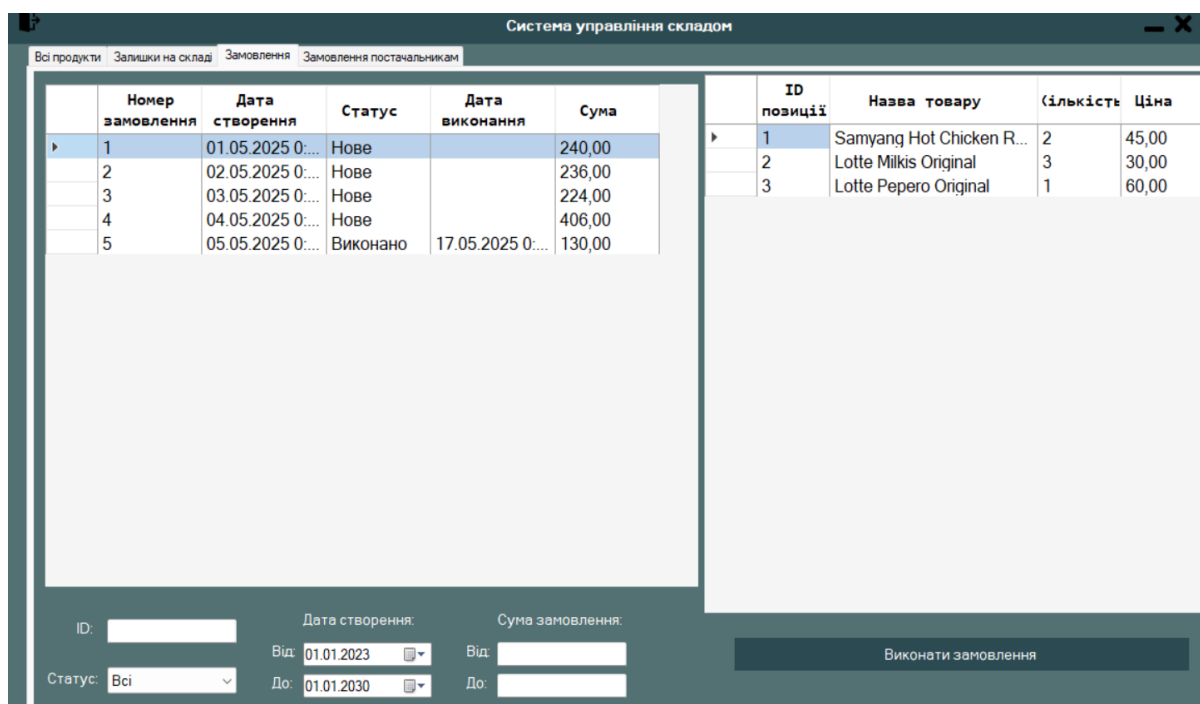
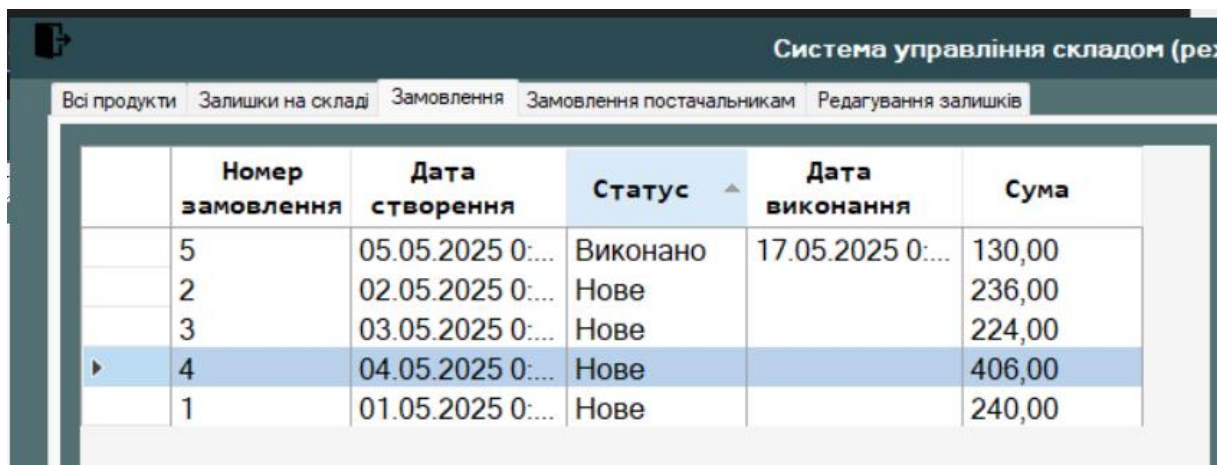


Рисунок 3.10 – Інформація про склад замовлення

- Натиснувши кнопку «Виконати замовлення» у правому вікні, його статус у лівому вікні зміниться, а залишки товарів у базі зменшаться, відповідно до складових замовлення.

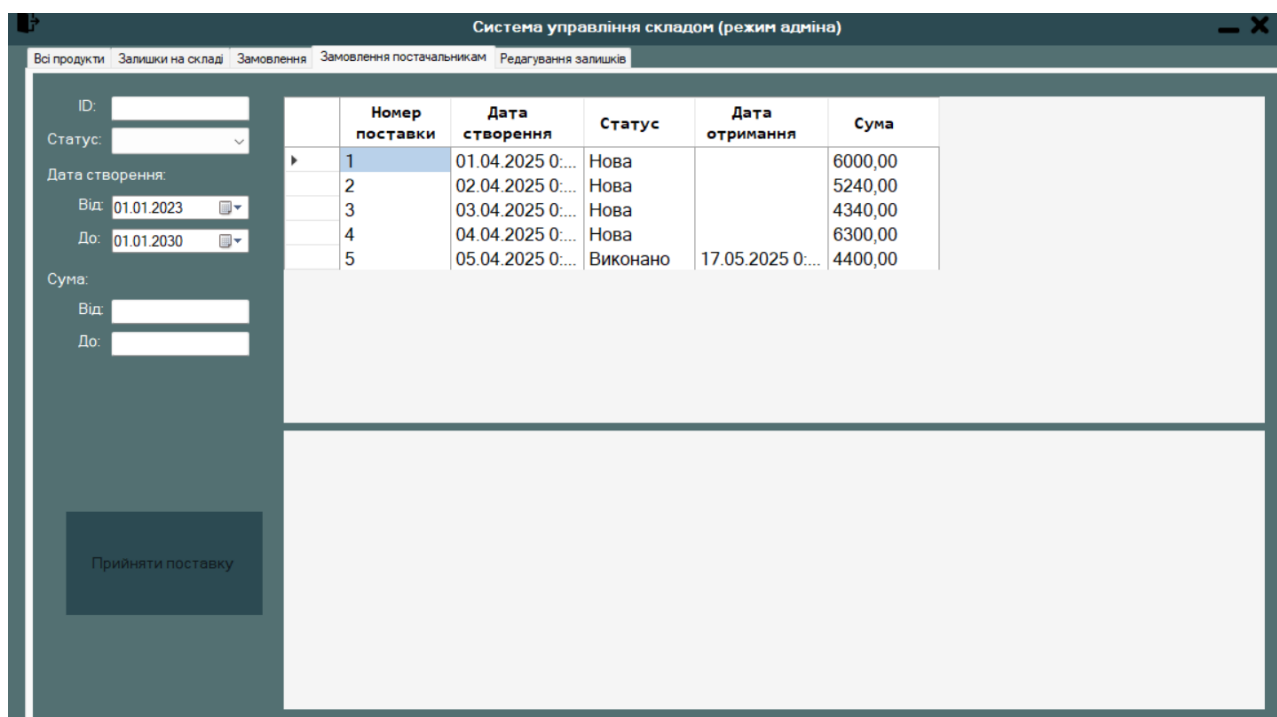
- Фільтрація замовлення за статусом (нове чи виконане), завдяки нижньому полю, в якому є функція пошуку по даті створення чи сумі замовлення, зображена на рисунку 3.11.



Номер замовлення	Дата створення	Статус	Дата виконання	Сума
5	05.05.2025 0:...	Виконано	17.05.2025 0:...	130,00
2	02.05.2025 0:...	Нове		236,00
3	03.05.2025 0:...	Нове		224,00
4	04.05.2025 0:...	Нове		406,00
1	01.05.2025 0:...	Нове		240,00

Рисунок 3.11 – Сортуння за статусом

- Вкладка «Замовлення постачальникам» містить інформацію про всі поставки які прибули до магазину. Вміст вкладки зображений на рисунку 3.12.



Номер поставки	Дата створення	Статус	Дата отримання	Сума
1	01.04.2025 0:...	Нова		6000,00
2	02.04.2025 0:...	Нова		5240,00
3	03.04.2025 0:...	Нова		4340,00
4	04.04.2025 0:...	Нова		6300,00
5	05.04.2025 0:...	Виконано	17.05.2025 0:...	4400,00

Рисунок 3.12 – Вкладка «Замовлення постачальника»

•Верхнє поле містить список всіх поставок, і прийнятих до магазину, і тих що прибули і не були ще оброблені. Дата створення замовлення постачальником, статус та дата виконання і сума замовлення – характеристики поставок, які вказані у верхньому полі.

•При обранні однієї нової поставки можна побачити зміни у нижньому полі – це інформація про складники поставки: назва, кількість, ціна, термін придатності, дата виготовлення і рекомендації щодо зберігання. Перегляд всієї інформації відбувається як на рисунку 3.13.

Номер поставки	Дата створення	Статус	Дата отримання	Сума
1	01.04.2025 0:...	Нова		6000,00
2	02.04.2025 0:...	Нова		5240,00
3	03.04.2025 0:...	Нова		4340,00
4	04.04.2025 0:...	Нова		6300,00
5	05.04.2025 0:...	Виконано	17.05.2025 0:...	4400,00

ID позиції	Назва товару	Кількість	Ціна	Дата виготовлення	Термін придатності	Секція
1	Samyang Hot Chicken R...	100	40,00	01.04.2025 0:...	01.04.2026 0:...	Сухий скл...
2	Samyang Hot Chicken R...	50	40,00	01.04.2025 0:...	01.04.2026 0:...	Сухий скл...

Рисунок 3.13 – Інформація про склад поставки

- Натиснувши кнопку «Прийняти поставку» статус поставки у верхньому полі зміниться на «Виконано» та з'явиться дата прийняття поставки. Також автоматично оновляться залишки на складі відповідно до кількості товарів, які прибули.
- За допомогою кнопок зліва, відбувається фільтрація замовлення за статусом виконання, датою створення та сумою. Результат фільтрації зображено на рисунку 3.14.

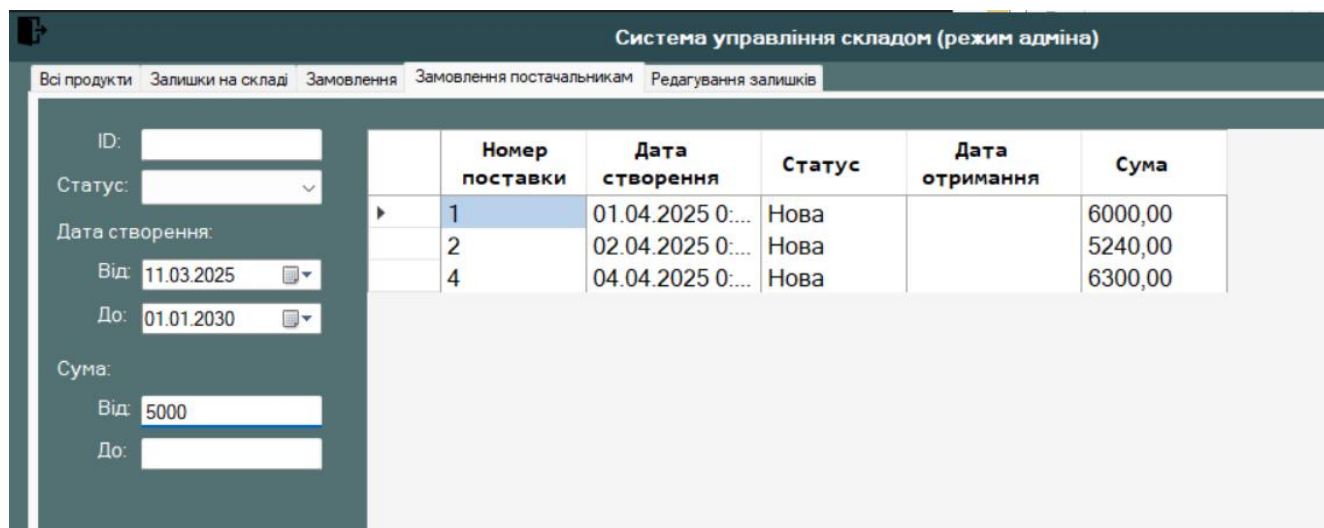


Рисунок 3.14 – Фільтрація поставок за датою та сумою

- Вікно «Редагування залишків» – це унікальна вкладка, до якої має доступ лише координатор складу, зайшовши до свого профілю, її вигляд зображено на рисунку 3.15. Таке рішення було узгоджене з директором магазину, оскільки координатор є довіреним лицем, що несе відповідальність за склад та правильне відстеження товарів.

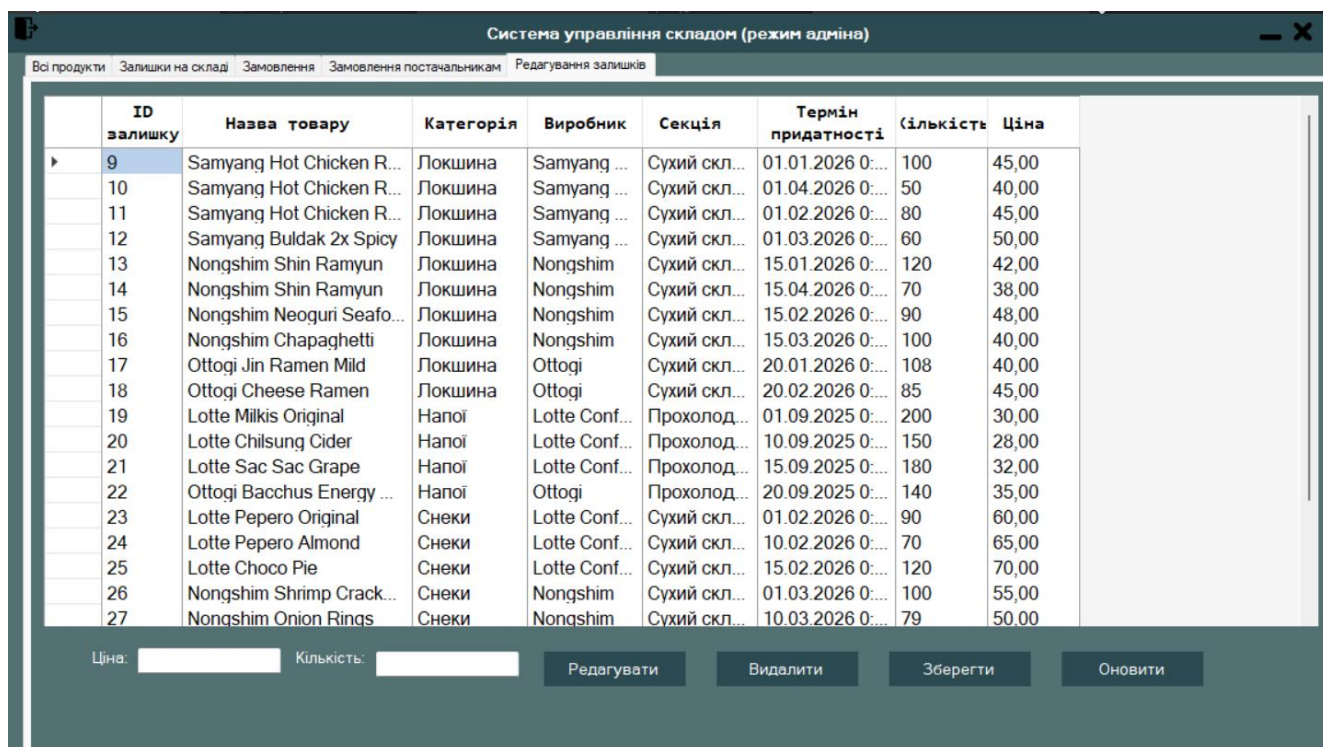


Рисунок 3.15 – Вікно «Редагування залишків»

- Вікно містить загальну інформацію про всі наявні у системі товари: назву, категорію, виробника, секцію зберігання, термін придатності, кількість та ціну.
- Обравши потрібний для редагування товар, можна змінити його ціну та кількість, ввівши потрібні дані у поля «Ціна» та «Кількість» та натиснувши кнопку «Редагувати». На рисунку 3.16 показано як це відбувається. Обов'язково потрібно не забути зберегти відредаговані данні у системі за допомогою окремої кнопки «Зберегти».

Система управління складом (режим адміна)

Всі продукти Залишки на складі Замовлення Замовлення постачальникам Редагування залишків

ID залишку	Назва товару	Категорія	Виробник	Секція	Термін придатності	Кількість	Ціна
9	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.01.2026 0...	100	45,00
10	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.04.2026 0...	50	40,00
11	Samyang Hot Chicken R...	Локшина	Samyang ...	Сухий скл...	01.02.2026 0...	80	45,00
12	Samyang Buldak 2x Spicy	Локшина	Samyang ...	Сухий скл...	01.03.2026 0...	60	50,00
13	Nongshim Shin Ramyun	Локшина	Nongshim	Сухий скл...	15.01.2026 0...	120	42,00
14	Nongshim Shin Ramyun	Локшина	Nongshim	Сухий скл...	15.04.2026 0...	70	38,00
15	Nongshim Neoguri Seafo...	Локшина	Nongshim	Сухий скл...	15.02.2026 0...	90	48,00
16	Nongshim Chapaggetti	Локшина	Nongshim	Сухий скл...	15.03.2026 0...	100	40,00
17	Ottogi Jin Ramen Mild	Локшина	Ottogi	Сухий скл...	20.01.2026 0...	108	40,00
18	Ottogi Cheese Ramen	Локшина	Ottogi	Сухий скл...	20.02.2026 0...	85	45,00
19	Lotte Milkis Original	Напої	Lotte Conf...	Прохолод...	01.09.2025 0...	200	30,00
20	Lotte Chilsung Cider	Напої	Lotte Conf...	Прохолод...	10.09.2025 0...	150	28,00
21	Lotte Sac Sac Grape	Напої	Lotte Conf...	Прохолод...	15.09.2025 0...	180	32,00
22	Ottogi Bacchus Energy ...	Напої	Ottogi	Прохолод...	20.09.2025 0...	140	35,00
23	Lotte Pepero Original	Снеки	Lotte Conf...	Сухий скл...	01.02.2026 0...	90	60,00
24	Lotte Pepero Almond	Снеки	Lotte Conf...	Сухий скл...	10.02.2026 0...	70	65,00
25	Lotte Choco Pie	Снеки	Lotte Conf...	Сухий скл...	15.02.2026 0...	120	70,00
26	Nongshim Shrimp Crack...	Снеки	Nongshim	Сухий скл...	01.03.2026 0...	100	55,00
27	Nongshim Onion Rings	Снеки	Nongshim	Сухий скл...	10.03.2026 0...	79	50,00

Ціна: 50,00 Кількість: 60

Редагувати Видалити Зберегти Оновити

Рисунок 3.16 – Редагування товару

- Натиснувши кнопку «Оновити», внесені у додаток дані скинуться до початкових, що є корисною функцією та швидким рішенням при неправильному підрахунку.
- «Видалити» - це кнопка яка видаляє всю інформацію про товар, і він більше не буде знаходитися у базі чи додатку. Це корисна функція для випадків, якщо координатор складу точно впевнений що даний товар вже не буде постачатися.
- Кнопка «Вийти»

На всіх вкладках згори зліва є працююча кнопка, яка дозволяє повернутися до меню авторизації. Вона зображена на рисунку 3.17.



Рисунок 3.17 – Кнопка «Вийти»

- «Згорнути» та «Закрити» додаток

Також для більшої зручності в правому верхньому кутку було створено дві кнопки, які зображені на рисунку 3.18.

- «Згорнути» щоб згорнути додаток до панелі, не закриваючи його.
- «Закрити» щоб швидко закрити програму, а не повертатися до меню авторизації.

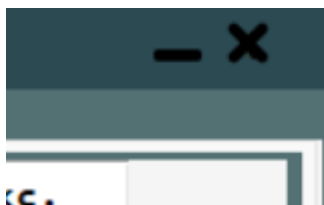


Рисунок 3.18 – Кнопки «Згорнути» та «Закрити»

3.5. Тестування програмного продукту

3.5.1. Таблиця результатів тестування

В таблиці 3.1 наведено результати конкретних тестів, які було проведено для перевірки функцій створеної інформаційної системи.

Таблиця 3.1 Результати тестування функцій додатку

№	Пріоритет	Перевірка	Очікуваний результат	Windows 10	Windows 8
1	високий	Завантаження списку товарів	Успішне завантаження та відображення товарів у dataGrid	Passed	Passed
2	високий	Фільтрація товарів за категорією	Товари фільтруються коректно за вибраною категорією	Passed	Passed

№	Пріоритет	Перевірка	Очікуваний результат	Windows 10	Windows 8
3	середній	Завантаження запасів із розрахунком резерву	Запаси відображаються коректними значеннями reserved та available	Passed	Passed
4	середній	Фільтрація запасів за ціною (priceFrom/priceTo)	Запаси фільтруються в межах заданого діапазону цін	Passed	Passed
5	високий	Завантаження та відображення замовлень	Замовлення завантажуються та відображаються в dataGridOrders	Passed	Passed
6	високий	Виконання замовлення (ExecuteOrder)	Статус замовлення змінюється на «Виконано»	Passed	Passed
7	середній	Редагування запасів (editButton_Click)	Ціна та кількість оновлюється в таблиці після редагування	Passed	Passed
8	середній	Видалення запасів (deleteButton_Click)	Вибраний запис видаляється з dataGridStockEdit	Passed	Passed
9	високий	Збереження змін запасів у базі даних (saveButton_Click)	Зміни коректно зберігаються в таблиці stock	Passed	Passed
10	низький	Обробка порожніх значень при фільтрації	Система коректно обробляє порожні поля	Passed	Passed

Тестування складської інформаційної системи на платформах Windows 11 та Windows 8 показало успішну працездатність ключових функцій. Система продемонструвала стабільність при обробці даних, коректну валідацію введення та сумісність із обома операційними системами.

ВИСНОВОК

Аналіз діяльності складського відділу мережі магазинів «Смак Кореї» показав значні недоліки в управлінні запасами через відсутність автоматизації. Ручні методи обробки даних знижують продуктивність діяльності відділу, ускладнюють швидкий доступ до інформації про запаси та обмежують можливості компанії відповідати сучасним стандартам ритейлу, де швидкість і зручність роботи з даними є ключовими для ефективності. Це не лише зменшує продуктивність, а й послаблює конкурентоспроможність компанії на ринку.

Проведений аналіз конкурентів підтверджує, що магазини-аналоги вже використовують централізовані системи з ефективними функціями фільтрації, редагування та обробки даних, що забезпечує їм перевагу в оперативності діяльності. На основі цього аналізу було проведено розрахунки економічної доцільності впровадження автоматизованої складської системи для магазину «Смаку Кореї». Загальна вартість розробки та впровадження інформаційної системи становить 37600 грн., а очікуваний економічний ефект — 23500 грн. на рік, що формується за рахунок: зниження операційних витрат на 15% (7500 грн.) завдяки автоматизації процесів; підвищення ефективності обробки замовлень на 30% (12000 грн.) через швидше виконання та зменшення помилок; скорочення витрат часу на обробку даних на 25% (4000 грн.) завдяки оптимізації редагування запасів. Порівняно з комерційними рішеннями вартістю від 50000 грн. до 70000 грн., власна розробка є значно вигіднішою, з терміном окупності лише 1,6 року.

На основі аналізу поточного стану складських процесів магазину (рівень AS IS) було розроблено оптимізовану модель бізнес-процесів (рівень TO BE), яка усуває виявлені недоліки. Ця модель включає: стандартизовані процеси завантаження запасів, виконання замовлень і редагування даних; оперативне завантаження та фільтрацію запасів; автоматичне оновлення запасів під час виконання замовлень; редагування та видалення даних із валідацією; інтеграцію з базою даних для відображення змін у реальному часі. Реалізація цих функцій оптимізує управління складом, знижує навантаження на персонал і підвищує точність обліку.

Для реалізації проєкту обрано технологічний стек на основі C#, Windows Forms і Microsoft SQL Server. Даний вибір забезпечує високу продуктивність і стабільність на десктопних платформах, зручний графічний інтерфейс, ефективну роботу з великими обсягами даних через SQL-запити та можливість подальшого розширення функціоналу. Завдяки цьому система вирішує ключові проблеми компанії: автоматизована фільтрація підвищує ефективність управління запасами; оптимізація процесів виконання замовлень і отримання товарів зменшує витрати часу; зручний доступ до актуальної інформації про товари забезпечує точність обліку; зниження операційних витрат сприяє фінансовій ефективності.

Загалом, розробка та впровадження інформаційної системи для складського обліку є стратегічно важливим рішенням для «Смаку Кореї». Воно дозволяє не лише відповідати сучасним вимогам до автоматизації складських процесів, а й створює міцну основу для підвищення конкурентоспроможності компанії та її подальшого зростання на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Інформаційні системи та штучний інтелект» денної форми здобуття освіти [Електрон. ресурс] / уклад. С. В. Грибков, Н. В. Ліманська, М. П. Костіков. – К.: НУХТ, 2025. – 43 с.
2. ДСТУ 3973-2000. СИСТЕМА РОЗРОБЛЕННЯ ТА ПОСТАВЛЕННЯ ПРОДУКЦІЇ НА ВИРОБНИЦТВО. Чинний від 27.11.2000. Київ : ДЕРЖСТАНДАРТ УКРАЇНИ, 2001. 43 с. URL: <https://www.pdau.edu.ua/sites/default/files/node/2805/dstu39732000.pdf> (дата звернення: 15.05.2025)
3. ДСТУ ISO/IEC 27001:2015. МЕТОДИ ЗАХИСТУ СИСТЕМИ УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ. На заміну ДСТУ ISO/IEC 27001:2010 ; чинний від 01.01.2017. Київ : ДП «УкрНДНЦ», 2016. 28 с. URL: https://www.assistem.kiev.ua/doc/dstu_ISO-IEC_27001_2015.pdf (дата звернення: 15.05.2025).
4. ДСТУ Б В.2.5-82:2016. ЕЛЕКТРОБЕЗПЕКА В БУДІВЛЯХ І СПОРУДАХ Вимоги до захисних заходів від ураження електричним струмом. На заміну ДБН В.2.5-27-2006 ; чинний від 01.04.2017. Київ : ДП "УкрНДНЦ", 2016. 110 с. URL: <http://www.tsatu.edu.ua/ettp/wp-content/uploads/sites/25/dstu-b-v.2.5-82-2016-elektrobezpeka-v-budivljah-i-sporudah-1.pdf> (дата звернення: 15.05.2025).
5. Смак Кореї. Офіційний сайт [Електронний ресурс]. – Режим доступу: <https://www.smak-korea.com.ua/> – Дата звернення: 19.05.2025.
6. Смак Кореї. Про нас [Електронний ресурс]. – Режим доступу: <https://www.smak-korea.com.ua/about/> – Дата звернення: 19.05.2025.
7. EVA. Офіційний сайт [Електронний ресурс]. – Режим доступу: <https://eva.ua/> – Дата звернення: 19.05.2025.
8. Лан-Україна. Офіційний сайт [Електронний ресурс]. – Режим доступу: <https://www.lanukr.com/> – Дата звернення: 19.05.2025.

9. FOZZY C&C. Офіційний сайт [Електронний ресурс]. – Режим доступу: https://fozzy.ua/ua/retail_chains/fozzy/ – Дата звернення: 19.05.2025.
10. Про бухгалтерський облік та фінансову звітність в Україні : Закон України від 16.07.1999 № 996-XIV. – Відомості Верховної Ради України. – 1999. – № 46–47. – Ст. 391.
11. ДСТУ ISO/IEC/IEEE 26512:2018. ІНЖЕНЕРІЯ СИСТЕМ І ПРОГРАМНИХ ЗАСОБІВ. На заміну - ; чинний від 18.12.2018. Київ : Не є офіційним виданням. 20 с. (дата звернення: 15.05.2025).
12. Автоматизація складу: порівняння систем [Електронний ресурс]. – Режим доступу: <https://www.logistics-ukraine.com/wms-soft/> – Дата звернення: 19.05.2025.
13. Системи автоматизації складу: огляд рішень для бізнесу [Електронний ресурс]. – Режим доступу: <https://www.prosale.ua/articles/sistemy-avtomatizatsii-sklada> – Дата звернення: 19.05.2025.
14. WMS-системи для малого бізнесу: огляд популярних платформ [Електронний ресурс]. – Режим доступу: <https://it-enterprise.com/blog/wms-sistemy-dlya-biznesu> – Дата звернення: 19.05.2025.
15. Автоматизація складської логістики: програмні продукти та сервіси [Електронний ресурс]. – Режим доступу: <https://logist.today/uk/blog/wms-solutions-for-logistics/> – Дата звернення: 19.05.2025.
16. Warehouse Management System [Електронний ресурс]. – Режим доступу: <https://github.com/ArtemSharaievskyi/Warehouse-Management-System> – Дата звернення: 19.05.2025.
17. C# Warehouse Management System [Електронний ресурс]. – Режим доступу: <https://github.com/Guapi12138/C-WMS> – Дата звернення: 19.05.2025.
18. Мова SQL. Створення та обробка баз даних / В.М. Боярчук, Л.М. Боярчук
19. Посібник присвячений основам роботи з реляційними базами даних та мовою SQL, що ідеально підходить для практичної реалізації складської системи. [Електронний ресурс]. – Режим доступу:

<https://elib.nakkkim.edu.ua/bitstream/123456789/3791/1/SQL.pdf> – Дата звернення: 20.05.2025.

20. Програмування мовою C# у середовищі .NET Framework / С.О. Семеріков, В.М. Ткачук
Видання охоплює створення Windows-додатків мовою C#, включаючи роботу з базами даних, WinForms та приклади практичного застосування. [Електронний ресурс]. – Режим доступу: https://lib.iitta.gov.ua/726463/1/Csharp_2020.pdf – Дата звернення: 20.05.2025.
21. ДСТУ 3974-2000. СИСТЕМА РОЗРОБЛЕННЯ ТА ПОСТАВЛЕННЯ ПРОДУКЦІЇ НА ВИРОБНИЦТВО. На заміну - ; чинний від 27.11.2000. Київ : ДЕРЖСТАНДАРТ УКРАЇНИ, 2001. 48 с. (дата звернення: 15.05.2025).
22. ДСТУ 7237:2011. СИСТЕМА СТАНДАРТІВ БЕЗПЕКИ ПРАЦІ ЕЛЕКТРОБЕЗПЕКА. На заміну на заміну ДСТУ ІЕС 60050-604:2004 ; чинний від 02.02.2011. Київ : ДЕРЖСТАНДАРТ УКРАЇНИ, 2011. 12 с. (дата звернення: 15.05.2025).
23. ДСТУ 7238:2011. СИСТЕМА СТАНДАРТІВ БЕЗПЕКИ ПРАЦІ. Чинний від 02.02.2011. Київ : ДЕРЖСТАНДАРТ УКРАЇНИ, 2011. 9 с. (дата звернення: 15.05.2025).
24. ДСТУ EN 12464-1:2016. Світло та освітлення. Освітлення робочих місць. Частина 1. Внутрішні робочі місця.. Чинний від 01.12.2017. Київ : ДП «УкрНДНЦ», 2018. 47 с. (дата звернення: 15.05.2025).
25. ДСТУ EN 13306:2019. ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ. Терміни та визначення понять. Чинний від 01.10.2007. Київ : ДЕРЖСПОЖИВСТАНДАРТ УКРАЇНИ, 2008. 34 с. (дата звернення: 15.05.2025).
26. ДСТУ ISO 6309:2007. ПРОТИПОЖЕЖНИЙ ЗАХИСТ. Знаки безпеки Форма та колір. На заміну ГОСТ 12.4.026–76 ; чинний від 30.03.2007. Київ : ДЕРЖСПОЖИВСТАНДАРТ УКРАЇНИ, 2008. 12 с. URL: https://www.ksv.biz.ua/GOST/DSTY_ALL/DSTY1/dsty_iso_6309-2007.pdf (дата звернення: 15.05.2025).

27. ДСТУ 2293:2014. ОХОРОНА ПРАЦІ. Терміни та визначення основних понять. На заміну ДСТУ 2293-99 ; чинний від 01.05.2015. Київ : МІНЕКОНОМПРОЗВИТКУ УКРАЇНИ, 2015. 18 с. (дата звернення: 15.05.2025).
28. ДСТУ 3008:2015. ЗВІТИ У СФЕРІ НАУКИ І ТЕХНІКИ. На заміну ДСТУ 3008-95 ; чинний від 22.06.2015. Київ : ДП «УкрНДНЦ», 2016. 26 с. URL: https://science.kname.edu.ua/images/dok/derzhstandart_3008_2015.pdf (дата звернення: 15.05.2025).
29. ДСТУ EN ISO 9241-210:2022. Ергономіка взаємодії людина-система.. На заміну На заміну ДСТУ EN ISO 9241-210:2019 ; чинний від 28.12.2022. Київ : Не є офіційним виданням. 21 с. (дата звернення: 15.05.2025).
30. WMS Логістика. Управління складом | BGS Solutions [Електронний ресурс]. – Режим доступу: <https://bgs.com.ua/wms-logistika-upravlinnya-skladom> – Дата звернення: 19.05.2025.
31. WMS система управління складом | Автоматизація складу [Електронний ресурс]. – Режим доступу: <https://ilogist.com.ua/ukr/wms.html> – Дата звернення: 19.05.2025.
32. LSC WMS – система керування складом [Електронний ресурс]. – Режим доступу: <https://lsccenter.com.ua/ua/logistics-optimization/lsc-wms> – Дата звернення: 19.05.2025.

ДОДАТКИ

Додаток А. Схема організаційної структури мережі магазинів «Смак Кореї»

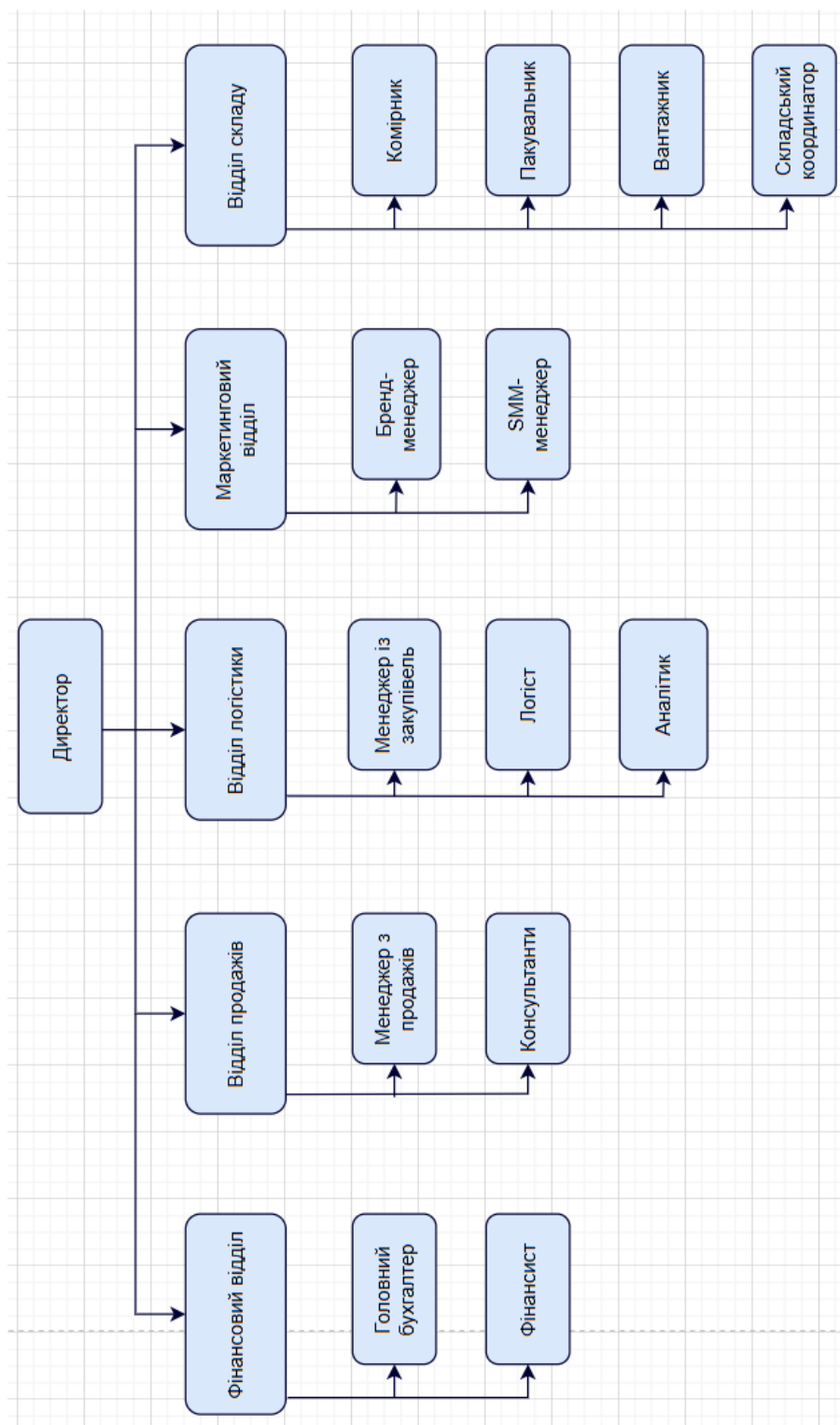


Рисунок А.1 – Структура підприємства

Додаток Б. Модель бізнес-процесів діяльності мережі магазинів «Смак Кореї»

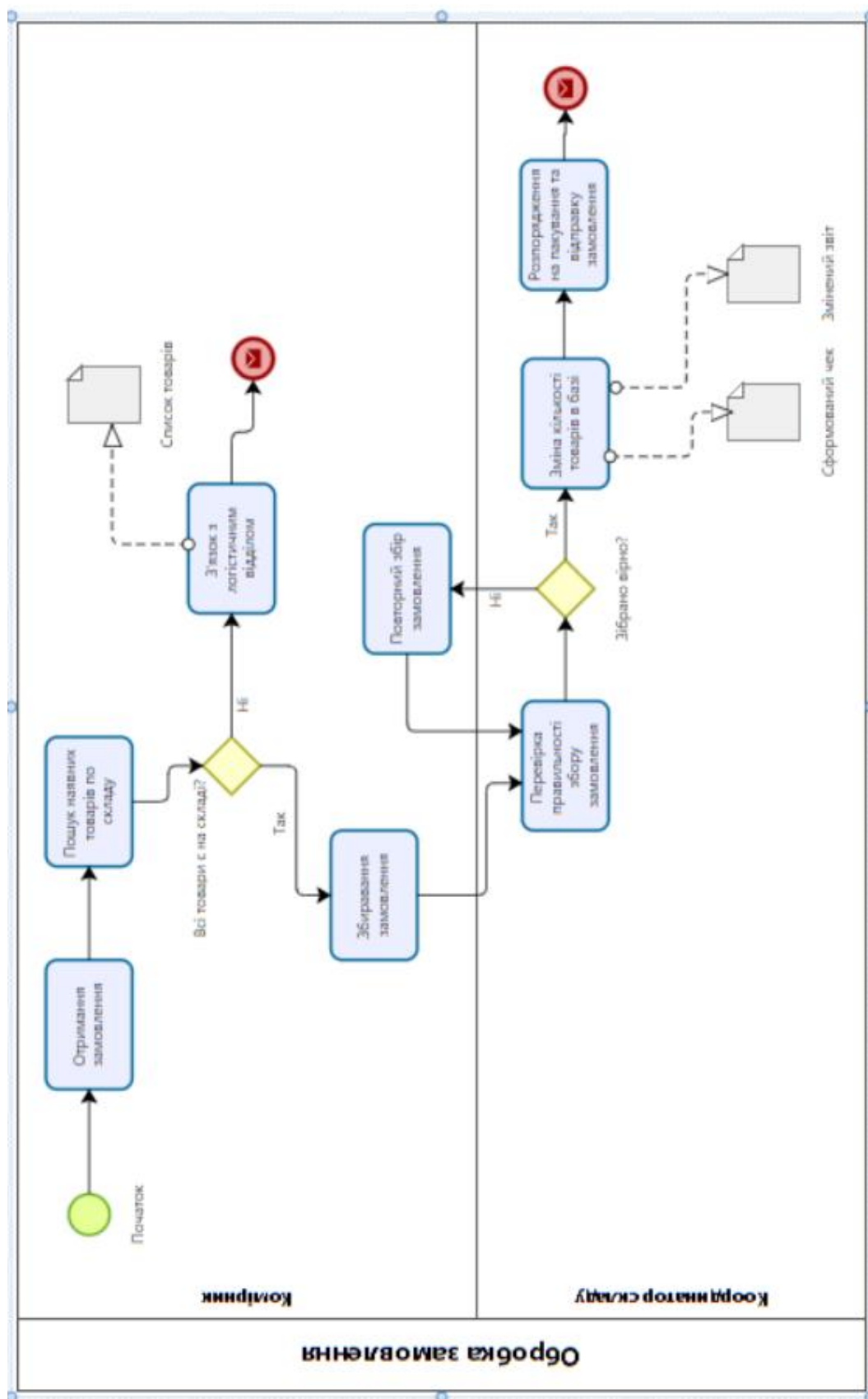


Рисунок Б.1 – Модель бізнес-процесів у нотації BPMN рівня AS-IS

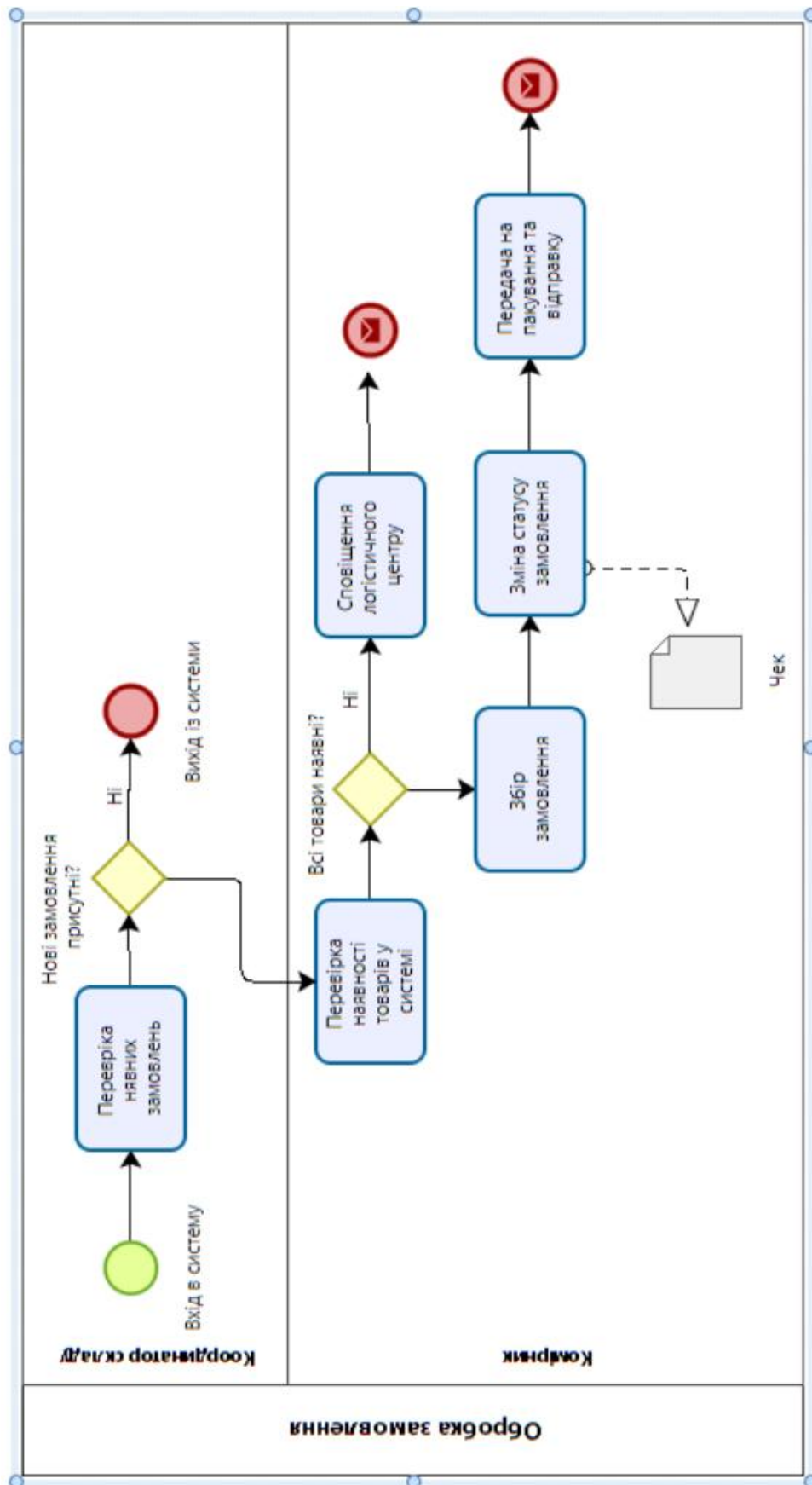


Рисунок Б.2 – Модель бізнес-процесів у нотації BPMN рівня TO-BE

Додаток В. Логічна модель бази даних

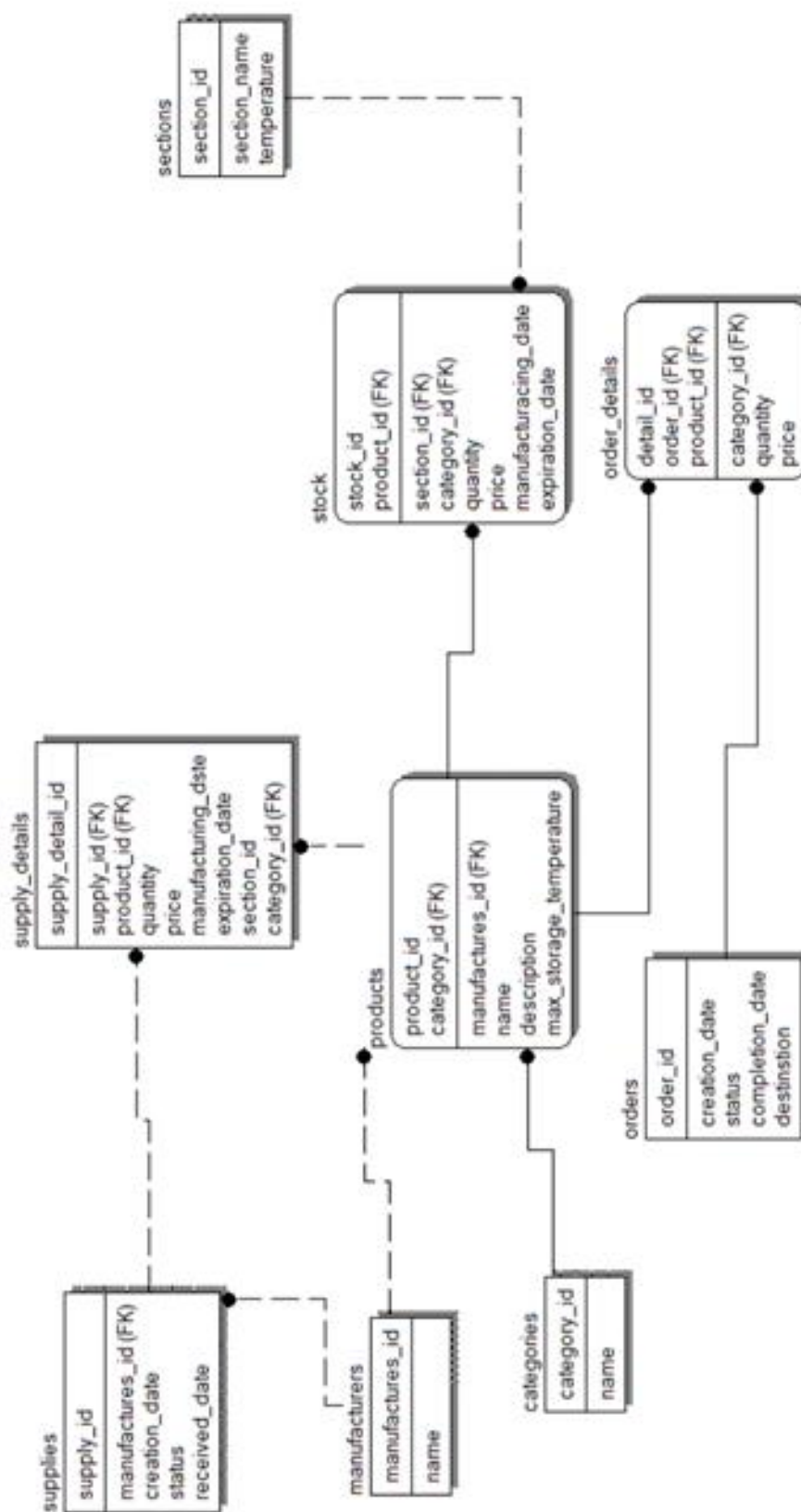


Рисунок В.1 – Логічна модель бази даних

Додаток Г. Фізична модель бази даних

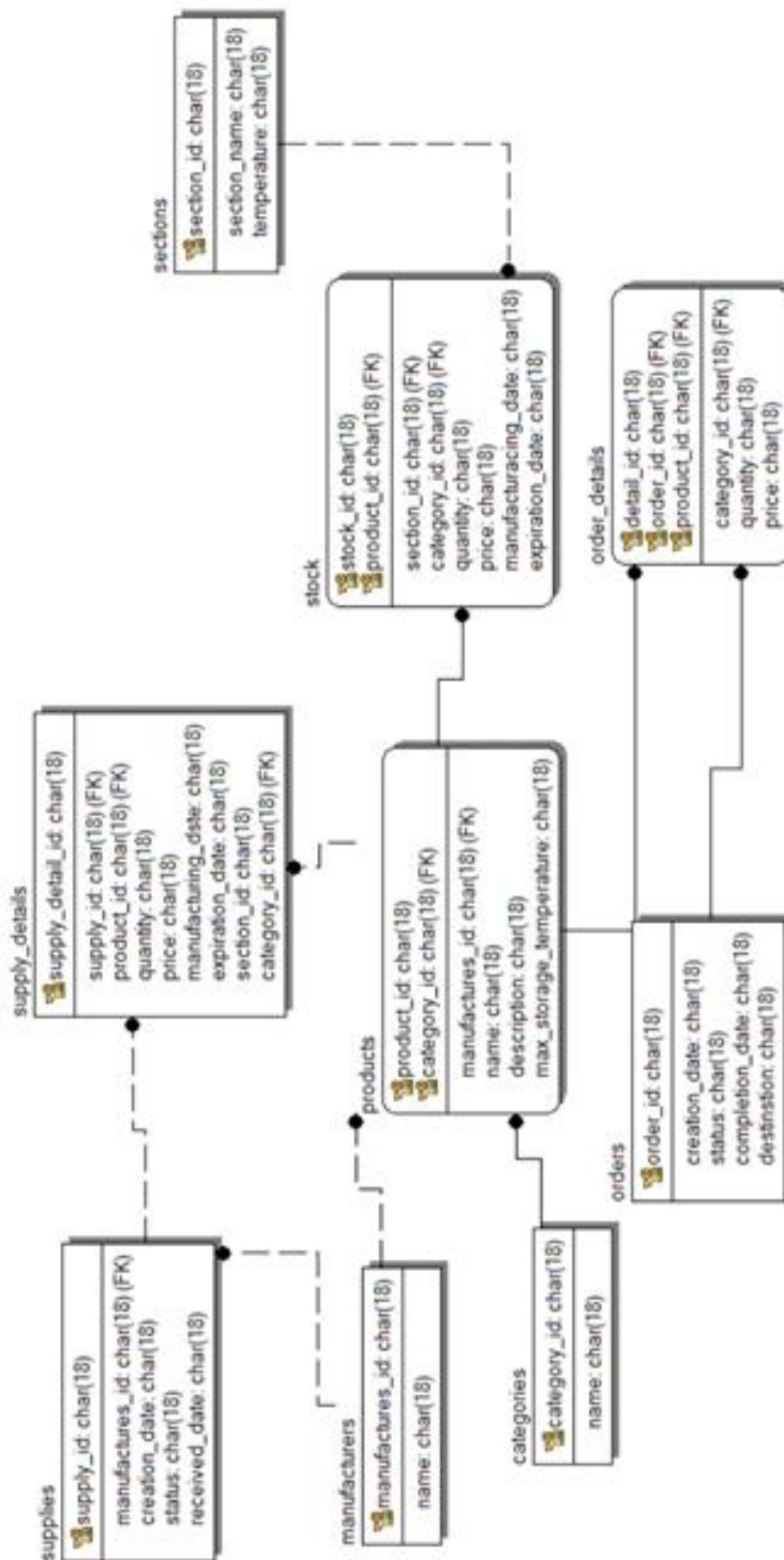


Рисунок Г.1 – Фізична модель бази даних

Додаток Д. Модель бази даних в СУБД MS SQL Server

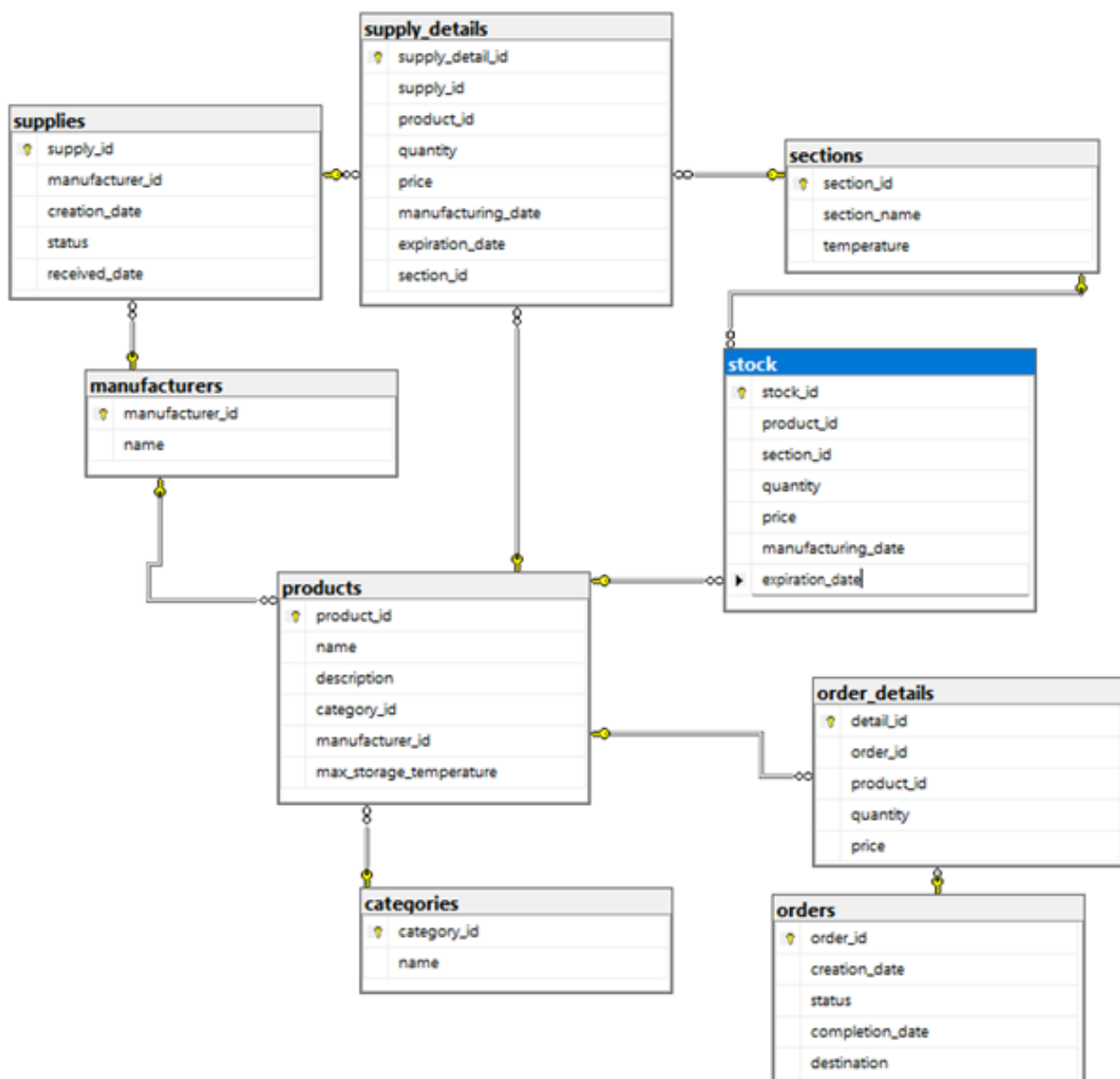


Рисунок Д.1 - Модель бази даних в СУБД MS SQL Server

Е.1. Програмний код завантаження, фільтрації та відображення товару

```

private void LoadProducts()
{
    try
    {
        connection.Open();
        string query = @"
            SELECT p.product_id, p.name AS product_name, p.description, c.name
AS category, m.name AS manufacturer, p.max_storage_temperature
            FROM products p
            JOIN categories c ON p.category_id = c.category_id
            JOIN manufacturers m ON p.manufacturer_id = m.manufacturer_id
            WHERE 1=1";

        List<string> conditions = new List<string>();
        List<SqlParameter> parameters = new List<SqlParameter>();
        if (!string.IsNullOrEmpty(nameFilter.Text))
        {
            conditions.Add("p.name LIKE @name");
            parameters.Add(new SqlParameter("@name", "%" + nameFilter.Text +
"%"));
        }
        var selectedCategories = categoryCheckBoxes.Where(cb =>
cb.Checked).Select(cb => cb.Text).ToList();
        if (selectedCategories.Any())
        {
            conditions.Add("c.name IN (" + string.Join(",",
selectedCategories.Select((_, i) => "@cat" + i)) + ")");
            for (int i = 0; i < selectedCategories.Count; i++)

```

```

        {
            parameters.Add(new SqlParameter("@cat" + i, selectedCategories[i]));
        }
    }

    var selectedManufacturers = manufacturerCheckBoxes.Where(cb =>
cb.Checked).Select(cb => cb.Text).ToList();
    if (selectedManufacturers.Any())
    {
        conditions.Add("m.name IN (" + string.Join(", ",
selectedManufacturers.Select((_, i) => "@man" + i)) + ")");
        for (int i = 0; i < selectedManufacturers.Count; i++)
        {
            parameters.Add(new SqlParameter("@man" + i,
selectedManufacturers[i]));
        }
    }

    var selectedSections = sectionCheckBoxes.Where(cb =>
cb.Checked).Select(cb => cb.Text).ToList();
    if (selectedSections.Any())
    {
        string sectionQuery = "SELECT section_name, temperature FROM
sections WHERE section_name IN (" +
            string.Join(", ", selectedSections.Select((_, i) => "@sec" + i)) + ")";
        SqlCommand sectionCmd = new SqlCommand(sectionQuery, connection);
        for (int i = 0; i < selectedSections.Count; i++)
        {
            sectionCmd.Parameters.AddWithValue("@sec" + i,
selectedSections[i]);
        }
        List<double> sectionTemperatures = new List<double>();
    }

```

```

using (SqlDataReader reader = sectionCmd.ExecuteReader())
{
    while (reader.Read())
    {
        sectionTemperatures.Add(Convert.ToDouble(reader["temperature"]));
    }
}
if (sectionTemperatures.Any())
{
    double minSectionTemperature = sectionTemperatures.Min();
    conditions.Add("p.max_storage_temperature >= @minSectionTemp");
    parameters.Add(new SqlParameter("@minSectionTemp",
minSectionTemperature));
}
}
if (conditions.Any())
{
    query += " AND " + string.Join(" AND ", conditions);
}
SqlCommand cmd = new SqlCommand(query, connection);
cmd.Parameters.AddRange(parameters.ToArray());
SqlDataAdapter adapter = new SqlDataAdapter(cmd);
DataTable dt = new DataTable();
adapter.Fill(dt);
dataGridView.Columns.Clear();
dataGridView.Columns.Add("product_id", "ID");
dataGridView.Columns.Add("product_name", "Назва товару");
dataGridView.Columns.Add("description", "Опис");
dataGridView.Columns.Add("category", "Категорія");
dataGridView.Columns.Add("manufacturer", "Виробник");

```

```
dataGrid.Columns.Add("max_storage_temperature", "Макс. температура");
dataGrid.Rows.Clear();
foreach (DataRow row in dt.Rows)
{
    dataGrid.Rows.Add(row["product_id"],          row["product_name"],
row["description"],          row["category"],          row["manufacturer"],
row["max_storage_temperature"]);
}
dataGrid.Columns["product_id"].Width = 50;
dataGrid.Columns["product_name"].Width = 200;
dataGrid.Columns["description"].Width = 250;
dataGrid.Columns["category"].Width = 100;
dataGrid.Columns["manufacturer"].Width = 100;
dataGrid.Columns["max_storage_temperature"].Width = 100;
connection.Close();
}
catch (Exception ex)
{
    MessageBox.Show($"Помилка завантаження товарів: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
```

Е.2. Програмний код завантаження, фільтрація, відображення та виконання замовлень

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
namespace kursova
{
    public partial class adminForm : Form
    {
        private readonly string connectionString = "Server=DESKTOP-
OLO366B\\SQLEXPRESS;Database=smakkorea;Integrated Security=True;";
        private SqlConnection connection;
        public adminForm()
        {
            InitializeComponent();
            connection = new SqlConnection(connectionString);
            LoadOrders();
        }
        private void LoadOrders()
        {
            try
            {
                connection.Open();
                string query = @"
                SELECT o.order_id, o.creation_date, o.status, o.completion_date,
                (SELECT SUM(od.quantity * od.price)
                FROM order_details od
                WHERE od.order_id = o.order_id) AS total_amount
```

```

FROM orders o
WHERE 1=1";
List<string> conditions = new List<string>();
List<SqlParameter> parameters = new List<SqlParameter>();
if (!string.IsNullOrEmpty(orderIdFilter.Text) &&
int.TryParse(orderIdFilter.Text, out int orderId))
{
    conditions.Add("o.order_id = @orderId");
    parameters.Add(new SqlParameter("@orderId", orderId));
}
if (creationDateFrom.Value != null && creationDateFrom.Value !=
DateTime.MinValue)
{
    conditions.Add("o.creation_date >= @creationFrom");
    parameters.Add(new SqlParameter("@creationFrom",
creationDateFrom.Value));
}
if (creationDateTo.Value != null && creationDateTo.Value !=
DateTime.MinValue)
{
    conditions.Add("o.creation_date <= @creationTo");
    parameters.Add(new SqlParameter("@creationTo",
creationDateTo.Value));
}

if (statusFilter.SelectedIndex > 0)
{
    conditions.Add("o.status = @status");
    parameters.Add(new SqlParameter("@status",
statusFilter.SelectedItem.ToString()));
}

```

```

    }
    if (!string.IsNullOrWhiteSpace(totalAmountFrom.Text) &&
double.TryParse(totalAmountFrom.Text, out double amountFrom))
    {
        conditions.Add("(SELECT SUM(od.quantity * od.price) FROM
order_details od WHERE od.order_id = o.order_id) >= @amountFrom");
        parameters.Add(new SqlParameter("@amountFrom", amountFrom));
    }
    if (!string.IsNullOrWhiteSpace(totalAmountTo.Text) &&
double.TryParse(totalAmountTo.Text, out double amountTo))
    {
        conditions.Add("(SELECT SUM(od.quantity * od.price) FROM
order_details od WHERE od.order_id = o.order_id) <= @amountTo");
        parameters.Add(new SqlParameter("@amountTo", amountTo));
    }
    if (conditions.Any())
    {
        query += " AND " + string.Join(" AND ", conditions);
    }
    SqlCommand cmd = new SqlCommand(query, connection);
    cmd.Parameters.AddRange(parameters.ToArray());

    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    adapter.Fill(dt);
    dataGridOrders.Columns.Clear();
    dataGridOrders.Columns.Add("order_id", "Номер замовлення");
    dataGridOrders.Columns.Add("creation_date", "Дата створення");
    dataGridOrders.Columns.Add("status", "Статус");
    dataGridOrders.Columns.Add("completion_date", "Дата виконання");

```

```

dataGridOrders.Columns.Add("total_amount", "Сума");
dataGridOrders.Rows.Clear();
foreach (DataRow row in dt.Rows)
{
    dataGridOrders.Rows.Add(row["order_id"],          row["creation_date"],
row["status"],
    row["completion_date"], row["total_amount"]);
}
dataGridOrders.Columns["order_id"].Width = 100;
dataGridOrders.Columns["creation_date"].Width = 120;
dataGridOrders.Columns["status"].Width = 100;
dataGridOrders.Columns["completion_date"].Width = 120;
dataGridOrders.Columns["total_amount"].Width = 100;
if (dataGridOrders.SelectedRows.Count > 0)
{
LoadOrderDetails((int)dataGridOrders.SelectedRows[0].Cells["order_id"].Value);
}
UpdateExecuteButtonState();
connection.Close();
}
catch (Exception ex)
{
    MessageBox.Show($"Помилка завантаження замовлень: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void LoadOrderDetails(int orderId)
{
    try
    {

```

```
connection.Open();
string query = @"
    SELECT od.detail_id, p.name AS product_name, od.quantity, od.price
    FROM order_details od
    JOIN products p ON od.product_id = p.product_id
    WHERE od.order_id = @orderId";
SqlCommand cmd = new SqlCommand(query, connection);
cmd.Parameters.AddWithValue("@orderId", orderId);
SqlDataAdapter adapter = new SqlDataAdapter(cmd);
DataTable dt = new DataTable();
adapter.Fill(dt);
dataGridViewOrderDetails.Columns.Clear();
dataGridViewOrderDetails.Columns.Add("detail_id", "ID позиції");
dataGridViewOrderDetails.Columns.Add("product_name", "Назва товару");
dataGridViewOrderDetails.Columns.Add("quantity", "Кількість");
dataGridViewOrderDetails.Columns.Add("price", "Ціна");
dataGridViewOrderDetails.Rows.Clear();
foreach (DataRow row in dt.Rows)
{
    dataGridViewOrderDetails.Rows.Add(row["detail_id"], row["product_name"],
row["quantity"], row["price"]);
}
dataGridViewOrderDetails.Columns["detail_id"].Width = 70;
dataGridViewOrderDetails.Columns["product_name"].Width = 200;
dataGridViewOrderDetails.Columns["quantity"].Width = 80;
dataGridViewOrderDetails.Columns["price"].Width = 80;
connection.Close();
}
catch (Exception ex)
{
```

```

        MessageBox.Show($"Помилка завантаження деталей замовлення:
{ex.Message}", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
private void UpdateExecuteButtonState()
{
    if (dataGridOrders.SelectedRows.Count > 0)
    {
        string status =
dataGridOrders.SelectedRows[0].Cells["status"].Value.ToString();
        executeButton.Enabled = status == "Нове";
    }
    else
    {
        executeButton.Enabled = false;
    }
}
private void ExecuteOrder()
{
    if (dataGridOrders.SelectedRows.Count == 0) return;
    int orderId = (int)dataGridOrders.SelectedRows[0].Cells["order_id"].Value;
    string status =
dataGridOrders.SelectedRows[0].Cells["status"].Value.ToString();
    if (status != "Нове")
    {
        MessageBox.Show("Замовлення вже виконано або не може бути
виконано.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    try

```

```

{
    connection.Open();
    SqlTransaction transaction = connection.BeginTransaction();
    string checkStockQuery = @"
        SELECT od.product_id, od.quantity, s.quantity AS stock_quantity
        FROM order_details od
        JOIN stock s ON od.product_id = s.product_id
        WHERE od.order_id = @orderId AND s.quantity < od.quantity";
    SqlCommand checkCmd = new SqlCommand(checkStockQuery, connection,
transaction);
    checkCmd.Parameters.AddWithValue("@orderId", orderId);
    SqlDataReader reader = checkCmd.ExecuteReader();
    if (reader.HasRows)
    {
        reader.Close();
        transaction.Rollback();
        connection.Close();
        MessageBox.Show("Недостатньо товару на складі для виконання
замовлення.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    reader.Close();
    string updateOrderQuery = @"
        UPDATE orders
        SET status = @status, completion_date = @completionDate
        WHERE order_id = @orderId";
    SqlCommand updateCmd = new SqlCommand(updateOrderQuery,
connection, transaction);
    updateCmd.Parameters.AddWithValue("@status", "Виконано");
    updateCmd.Parameters.AddWithValue("@completionDate", DateTime.Now);

```

```

updateCmd.Parameters.AddWithValue("@orderId", orderId);
updateCmd.ExecuteNonQuery();
string updateStockQuery = @"
    UPDATE s
    SET s.quantity = s.quantity - od.quantity
    FROM stock s
    JOIN order_details od ON s.product_id = od.product_id
    WHERE od.order_id = @orderId";
SqlCommand stockCmd = new SqlCommand(updateStockQuery, connection,
transaction);

stockCmd.Parameters.AddWithValue("@orderId", orderId);
stockCmd.ExecuteNonQuery();
transaction.Commit();
connection.Close();

MessageBox.Show("Замовлення виконано успішно. Час виконання: 11:51
PM EEST, 16 травня 2025 року.", "Успіх", MessageBoxButtons.OK,
MessageBoxIcon.Information);

LoadOrders();
LoadStock();
}
catch (Exception ex)
{
    connection.Close();
    MessageBox.Show($"Помилка виконання замовлення: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void orderIdFilter_TextChanged(object sender, EventArgs e)
{
    LoadOrders();
}

```

```
}  
private void creationDateFrom_ValueChanged(object sender, EventArgs e)  
{  
    LoadOrders();  
}  
private void creationDateTo_ValueChanged(object sender, EventArgs e)  
{  
    LoadOrders();  
}  
private void statusFilter_SelectedIndexChanged(object sender, EventArgs e)  
{  
    LoadOrders();  
}  
private void totalAmountFrom_TextChanged(object sender, EventArgs e)  
{  
    LoadOrders();  
}  
private void totalAmountTo_TextChanged(object sender, EventArgs e)  
{  
    LoadOrders();  
}  
private void dataGridOrders_SelectionChanged(object sender, EventArgs e)  
{  
    if (dataGridOrders.SelectedRows.Count > 0)  
    {  
        int orderId = (int)dataGridOrders.SelectedRows[0].Cells["order_id"].Value;  
        LoadOrderDetails(orderId);  
    }  
    UpdateExecuteButtonState();  
}
```

```

private void executeButton_Click(object sender, EventArgs e)
{
    ExecuteOrder();
}
}
}

```

Е.3. Програмний код функцій завантаження, фільтрації та відображення запасів

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
namespace kursova
{
    public partial class adminForm : Form
    {
        private readonly string connectionString = "Server=DESKTOP-
OLO366B\\SQLEXPRESS;Database=smakkorea;Integrated Security=True;";
        private SqlConnection connection;
        private List<CheckBox> stockCategoryCheckBoxes = new List<CheckBox>();
        private List<CheckBox> stockManufacturerCheckBoxes = new List<CheckBox>();
        private List<CheckBox> stockSectionCheckBoxes = new List<CheckBox>();
        private double minPrice = 0;
        private double maxPrice = double.MaxValue;
        public adminForm()
        {
            InitializeComponent();

```

```
connection = new SqlConnection(connectionString);
LoadPriceRange();
LoadCategoriesForCheckBoxes();
LoadManufacturersForCheckBoxes();
LoadSectionsForCheckBoxes();
LoadStock();
}

private void LoadPriceRange()
{
    try
    {
        connection.Open();
        string query = "SELECT MIN(price), MAX(price) FROM stock";
        SqlCommand cmd = new SqlCommand(query, connection);
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            minPrice = reader[0] != DBNull.Value ? Convert.ToDouble(reader[0]) : 0;
            maxPrice = reader[1] != DBNull.Value ? Convert.ToDateTime(reader[1]) :
double.MaxValue;
        }
        reader.Close();
        connection.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка завантаження діапазону цін: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

```
}  
private void LoadCategoriesForCheckBoxes()  
{  
    try  
    {  
        connection.Open();  
        string query = "SELECT name FROM categories";  
        SqlCommand cmd = new SqlCommand(query, connection);  
        SqlDataReader reader = cmd.ExecuteReader();  
        stockCategoryCheckBoxPanel.Controls.Clear();  
        stockCategoryCheckBoxes.Clear();  
        while (reader.Read())  
        {  
            string categoryName = reader["name"].ToString();  
            CheckBox checkBox = new CheckBox  
            {  
                Text = categoryName,  
                AutoSize = true,  
                Font = new Font("Microsoft Sans Serif", 9.75F),  
                ForeColor = Color.White  
            };  
            checkBox.CheckedChanged += (s, e) => LoadStock();  
            stockCategoryCheckBoxPanel.Controls.Add(checkBox);  
            stockCategoryCheckBoxes.Add(checkBox);  
        }  
        reader.Close();  
        connection.Close();  
    }  
    catch (Exception ex)  
    {
```

```
        MessageBox.Show($"Помилка завантаження категорій: {ex.Message}",
        "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
private void LoadManufacturersForCheckBoxes()
{
    try
    {
        connection.Open();
        string query = "SELECT name FROM manufacturers";
        SqlCommand cmd = new SqlCommand(query, connection);
        SqlDataReader reader = cmd.ExecuteReader();
        stockManufacturerCheckBoxPanel.Controls.Clear();
        stockManufacturerCheckBoxes.Clear();
        while (reader.Read())
        {
            string manufacturerName = reader["name"].ToString();
            CheckBox checkBox = new CheckBox
            {
                Text = manufacturerName,
                AutoSize = true,
                Font = new Font("Microsoft Sans Serif", 9.75F),
                ForeColor = Color.White
            };
            checkBox.CheckedChanged += (s, e) => LoadStock();
            stockManufacturerCheckBoxPanel.Controls.Add(checkBox);
            stockManufacturerCheckBoxes.Add(checkBox);
        }
        reader.Close();
        connection.Close();
    }
}
```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка завантаження виробників: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

private void LoadSectionsForCheckBoxes()
{
    try
    {
        connection.Open();
        string query = "SELECT section_name FROM sections";
        SqlCommand cmd = new SqlCommand(query, connection);
        SqlDataReader reader = cmd.ExecuteReader();
        stockSectionCheckBoxPanel.Controls.Clear();
        stockSectionCheckBoxes.Clear();
        while (reader.Read())
        {
            string sectionName = reader["section_name"].ToString();
            CheckBox checkBox = new CheckBox
            {
                Text = sectionName,
                AutoSize = true,
                Font = new Font("Microsoft Sans Serif", 9.75F),
                ForeColor = Color.White
            };
            checkBox.CheckedChanged += (s, e) => LoadStock();
            stockSectionCheckBoxPanel.Controls.Add(checkBox);
        }
    }
}

```

```

        stockSectionCheckBoxes.Add(checkBox);
    }
    reader.Close();
    connection.Close();
}
catch (Exception ex)
{
    MessageBox.Show($"Помилка завантаження секцій: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void LoadStock()
{
    try
    {
        connection.Open();
        string query = @"
            SELECT s.stock_id, p.name AS product_name, c.name AS category,
m.name AS manufacturer,
                sec.section_name, s.expiration_date, s.quantity, s.price,
                (SELECT ISNULL(SUM(od.quantity), 0)
                FROM order_details od
                JOIN orders o ON od.order_id = o.order_id
                WHERE o.status = 'Нове' AND od.product_id = s.product_id) AS
reserved,
                (s.quantity - (SELECT ISNULL(SUM(od.quantity), 0)
                FROM order_details od
                JOIN orders o ON od.order_id = o.order_id
                WHERE o.status = 'Нове' AND od.product_id = s.product_id)) AS
available

```

```

FROM stock s
JOIN products p ON s.product_id = p.product_id
JOIN categories c ON p.category_id = c.category_id
JOIN manufacturers m ON p.manufacturer_id = m.manufacturer_id
JOIN sections sec ON s.section_id = sec.section_id
WHERE 1=1";

```

```

List<string> conditions = new List<string>();
List<SqlParameter> parameters = new List<SqlParameter>();
if (!string.IsNullOrEmpty(stockNameFilter.Text))
{
    conditions.Add("p.name LIKE @name");
    parameters.Add(new SqlParameter("@name", "%" + stockNameFilter.Text
+ "%"));
}
var selectedCategories = stockCategoryCheckBoxes.Where(cb =>
cb.Checked).Select(cb => cb.Text).ToList();
if (selectedCategories.Any())
{
    conditions.Add("c.name IN (" + string.Join(", ",
selectedCategories.Select((_, i) => "@cat" + i)) + ")");
    for (int i = 0; i < selectedCategories.Count; i++)
    {
        parameters.Add(new SqlParameter("@cat" + i, selectedCategories[i]));
    }
}
var selectedManufacturers = stockManufacturerCheckBoxes.Where(cb =>
cb.Checked).Select(cb => cb.Text).ToList();
if (selectedManufacturers.Any())
{

```

```

        conditions.Add("m.name IN (" + string.Join(", ",
selectedManufacturers.Select((_, i) => "@man" + i)) + ")");
        for (int i = 0; i < selectedManufacturers.Count; i++)
        {
            parameters.Add(new SqlParameter("@man" + i,
selectedManufacturers[i]));
        }
    }
    var selectedSections = stockSectionCheckBoxes.Where(cb =>
cb.Checked).Select(cb => cb.Text).ToList();
    if (selectedSections.Any())
    {
        conditions.Add("sec.section_name IN (" + string.Join(", ",
selectedSections.Select((_, i) => "@sec" + i)) + ")");
        for (int i = 0; i < selectedSections.Count; i++)
        {
            parameters.Add(new SqlParameter("@sec" + i, selectedSections[i]));
        }
    }
    if (!string.IsNullOrEmpty(priceFrom.Text) &&
double.TryParse(priceFrom.Text, out double priceFromValue))
    {
        conditions.Add("s.price >= @priceFrom");
        parameters.Add(new SqlParameter("@priceFrom", priceFromValue));
    }
    if (!string.IsNullOrEmpty(priceTo.Text) &&
double.TryParse(priceTo.Text, out double priceToValue))
    {
        conditions.Add("s.price <= @priceTo");
        parameters.Add(new SqlParameter("@priceTo", priceToValue));
    }
}

```

```

    }
    if (expirationFrom.Value != null && expirationFrom.Value !=
DateTime.MinValue)
    {
        conditions.Add("s.expiration_date >= @expFrom");
        parameters.Add(new SqlParameter("@expFrom", expirationFrom.Value));
    }
    if (expirationTo.Value != null && expirationTo.Value != DateTime.MinValue)
    {
        conditions.Add("s.expiration_date <= @expTo");
        parameters.Add(new SqlParameter("@expTo", expirationTo.Value));
    }
    if (!string.IsNullOrEmpty(quantityFrom.Text) &&
int.TryParse(quantityFrom.Text, out int qtyFrom))
    {
        conditions.Add("s.quantity >= @qtyFrom");
        parameters.Add(new SqlParameter("@qtyFrom", qtyFrom));
    }
    if (!string.IsNullOrEmpty(quantityTo.Text) &&
int.TryParse(quantityTo.Text, out int qtyTo))
    {
        conditions.Add("s.quantity <= @qtyTo");
        parameters.Add(new SqlParameter("@qtyTo", qtyTo));
    }
    if (conditions.Any())
    {
        query += " AND " + string.Join(" AND ", conditions);
    }
    SqlCommand cmd = new SqlCommand(query, connection);
    cmd.Parameters.AddRange(parameters.ToArray());

```

```
SqlDataAdapter adapter = new SqlDataAdapter(cmd);
DataTable dt = new DataTable();
adapter.Fill(dt);
dataGridStock.Columns.Clear();
dataGridStock.Columns.Add("stock_id", "ID залишку");
dataGridStock.Columns.Add("product_name", "Назва товару");
dataGridStock.Columns.Add("category", "Категорія");
dataGridStock.Columns.Add("manufacturer", "Виробник");
dataGridStock.Columns.Add("section_name", "Секція");
dataGridStock.Columns.Add("expiration_date", "Термін придатності");
dataGridStock.Columns.Add("quantity", "Кількість");
dataGridStock.Columns.Add("price", "Ціна");
dataGridStock.Columns.Add("reserved", "Резерв");
dataGridStock.Columns.Add("available", "Доступні");
dataGridStock.Rows.Clear();
foreach (DataRow row in dt.Rows)
{
    dataGridStock.Rows.Add(row["stock_id"],          row["product_name"],
row["category"], row["manufacturer"],
    row["section_name"],    row["expiration_date"],    row["quantity"],
row["price"],
    row["reserved"], row["available"]);
}
dataGridStock.Columns["stock_id"].Width = 70;
dataGridStock.Columns["product_name"].Width = 200;
dataGridStock.Columns["category"].Width = 100;
dataGridStock.Columns["manufacturer"].Width = 100;
dataGridStock.Columns["section_name"].Width = 100;
dataGridStock.Columns["expiration_date"].Width = 120;
dataGridStock.Columns["quantity"].Width = 80;
```

```
dataGridStock.Columns["price"].Width = 80;
dataGridStock.Columns["reserved"].Width = 80;
dataGridStock.Columns["available"].Width = 80;
connection.Close();
}
catch (Exception ex)
{
    MessageBox.Show($"Помилка завантаження залишків: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void stockNameFilter_TextChanged(object sender, EventArgs e)
{
    LoadStock();
}
private void priceFrom_TextChanged(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(priceFrom.Text))
    {
        LoadStock();
        return;
    }
    if (!double.TryParse(priceFrom.Text, out double value) || value < 0)
    {
        priceFrom.Text = "0";
    }
    else if (value < minPrice)
    {
        priceFrom.Text = minPrice.ToString();
    }
}
```

```
else if (value > maxPrice)
{
    priceFrom.Text = maxPrice.ToString();
}
LoadStock();
}
private void priceTo_TextChanged(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(priceTo.Text))
    {
        LoadStock();
        return;
    }
    if (!double.TryParse(priceTo.Text, out double value) || value < 0)
    {
        priceTo.Text = "0";
    }
    else if (value < minPrice)
    {
        priceTo.Text = minPrice.ToString();
    }
    else if (value > maxPrice)
    {
        priceTo.Text = maxPrice.ToString();
    }
    LoadStock();
}
private void expirationFrom_ValueChanged(object sender, EventArgs e)
{
    LoadStock();
}
```

```
}  
private void expirationTo_ValueChanged(object sender, EventArgs e)  
{  
    LoadStock();  
}  
private void quantityFrom_TextChanged(object sender, EventArgs e)  
{  
    if (string.IsNullOrEmpty(quantityFrom.Text))  
    {  
        LoadStock();  
        return;  
    }  
    if (!int.TryParse(quantityFrom.Text, out int value) || value < 0)  
    {  
        quantityFrom.Text = "0";  
    }  
    LoadStock();  
}  
private void quantityTo_TextChanged(object sender, EventArgs e)  
{  
    if (string.IsNullOrEmpty(quantityTo.Text))  
    {  
        LoadStock();  
        return;  
    }  
    if (!int.TryParse(quantityTo.Text, out int value) || value < 0)  
    {  
        quantityTo.Text = "0";  
    }  
    LoadStock();  
}
```

```

    }
}
}

```

Е.4. Програмний код для редагування та видалення запасів

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
namespace kursova
{
    public partial class adminForm : Form
    {
        private readonly string connectionString = "Server=DESKTOP-
OLO366B\\SQLEXPRESS;Database=smakkorea;Integrated Security=True;";
        private SqlConnection connection;
        private DataTable stockEditDataTable;
        public adminForm()
        {
            InitializeComponent();
            connection = new SqlConnection(connectionString);
            LoadStockEdit();
        }
        private void LoadStockEdit()
        {
            try
            {
                connection.Open();
                string query = @"

```

```

SELECT s.stock_id, p.name AS product_name, c.name AS category,
m.name AS manufacturer,
        sec.section_name, s.expiration_date, s.quantity, s.price
FROM stock s
JOIN products p ON s.product_id = p.product_id
JOIN categories c ON p.category_id = c.category_id
JOIN manufacturers m ON p.manufacturer_id = m.manufacturer_id
JOIN sections sec ON s.section_id = sec.section_id";
SqlCommand cmd = new SqlCommand(query, connection);
SqlDataAdapter adapter = new SqlDataAdapter(cmd);
stockEditDataTable = new DataTable();
adapter.Fill(stockEditDataTable);
dataGridViewStockEdit.Columns.Clear();
dataGridViewStockEdit.Columns.Add("stock_id", "ID залишку");
dataGridViewStockEdit.Columns.Add("product_name", "Назва товару");
dataGridViewStockEdit.Columns.Add("category", "Категорія");
dataGridViewStockEdit.Columns.Add("manufacturer", "Виробник");
dataGridViewStockEdit.Columns.Add("section_name", "Секція");
dataGridViewStockEdit.Columns.Add("expiration_date", "Термін придатності");
dataGridViewStockEdit.Columns.Add("quantity", "Кількість");
dataGridViewStockEdit.Columns.Add("price", "Ціна");
dataGridViewStockEdit.Rows.Clear();
foreach (DataRow row in stockEditDataTable.Rows)
{
    dataGridViewStockEdit.Rows.Add(row["stock_id"], row["product_name"],
row["category"], row["manufacturer"],
        row["section_name"], row["expiration_date"], row["quantity"],
row["price"]);
}
dataGridViewStockEdit.Columns["stock_id"].Width = 70;

```

```
dataGridStockEdit.Columns["product_name"].Width = 200;
dataGridStockEdit.Columns["category"].Width = 100;
dataGridStockEdit.Columns["manufacturer"].Width = 100;
dataGridStockEdit.Columns["section_name"].Width = 100;
dataGridStockEdit.Columns["expiration_date"].Width = 120;
dataGridStockEdit.Columns["quantity"].Width = 80;
dataGridStockEdit.Columns["price"].Width = 80;
connection.Close();
}
catch (Exception ex)
{
    MessageBox.Show($"Помилка завантаження залишків для редагування:
{ex.Message}", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void dataGridStockEdit_SelectionChanged(object sender, EventArgs e)
{
    if (dataGridStockEdit.SelectedRows.Count > 0)
    {
        DataGridViewRow selectedRow = dataGridStockEdit.SelectedRows[0];
        editPrice.Text = selectedRow.Cells["price"].Value.ToString();
        editQuantity.Text = selectedRow.Cells["quantity"].Value.ToString();
    }
    else
    {
        editPrice.Text = "";
        editQuantity.Text = "";
    }
}
private void editButton_Click(object sender, EventArgs e)
```

```
{
    if (dataGridViewStockEdit.SelectedRows.Count == 0)
    {
        MessageBox.Show("Виберіть запис для редагування.", "Помилка",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    if (!double.TryParse(editPrice.Text, out double price) || price < 0 || price >
10000)
    {
        MessageBox.Show("Ціна повинна бути числом від 0 до 10000.",
        "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    if (!int.TryParse(editQuantity.Text, out int quantity) || quantity < 0 || quantity >
10000)
    {
        MessageBox.Show("Кількість повинна бути цілим числом від 0 до
10000.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    DataGridViewRow selectedRow = dataGridViewStockEdit.SelectedRows[0];
    int stockId = (int)selectedRow.Cells["stock_id"].Value;
    foreach (DataRow row in stockEditDataTable.Rows)
    {
        if ((int)row["stock_id"] == stockId)
        {
            row["price"] = price;
            row["quantity"] = quantity;
            break;
        }
    }
}
```

```

    }
}
selectedRow.Cells["price"].Value = price;
selectedRow.Cells["quantity"].Value = quantity;
MessageBox.Show("Дані успішно відредаговано в таблиці. Час: 12:06 AM
EEST, 17 травня 2025 року.", "Успіх", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
private void deleteButton_Click(object sender, EventArgs e)
{
    if (dataGridStockEdit.SelectedRows.Count == 0)
    {
        MessageBox.Show("Виберіть запис для видалення.", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    DataGridViewRow selectedRow = dataGridStockEdit.SelectedRows[0];
    int stockId = (int)selectedRow.Cells["stock_id"].Value;
    foreach (DataRow row in stockEditDataTable.Rows)
    {
        if ((int)row["stock_id"] == stockId)
        {
            row.Delete();
            break;
        }
    }
    dataGridStockEdit.Rows.Remove(selectedRow);
    MessageBox.Show("Запис успішно видалено з таблиці. Час: 12:06 AM
EEST, 17 травня 2025 року.", "Успіх", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}

```

```
}  
private void saveButton_Click(object sender, EventArgs e)  
{  
    try  
    {  
        connection.Open();  
        SqlTransaction transaction = connection.BeginTransaction();  
        string updateQuery = @"  
            UPDATE stock  
            SET quantity = @quantity, price = @price  
            WHERE stock_id = @stockId";  
        string deleteQuery = @"  
            DELETE FROM stock  
            WHERE stock_id = @stockId";  
        foreach (DataRow row in stockEditDataTable.Rows)  
        {  
            if (row.RowState == DataRowState.Modified)  
            {  
                SqlCommand updateCmd = new SqlCommand(updateQuery,  
connection, transaction);  
                updateCmd.Parameters.AddWithValue("@stockId", row["stock_id"]);  
                updateCmd.Parameters.AddWithValue("@quantity", row["quantity"]);  
                updateCmd.Parameters.AddWithValue("@price", row["price"]);  
                updateCmd.ExecuteNonQuery();  
            }  
            else if (row.RowState == DataRowState.Deleted)  
            {  
                SqlCommand deleteCmd = new SqlCommand(deleteQuery, connection,  
transaction);
```

```

        deleteCmd.Parameters.AddWithValue("@stockId", row["stock_id",
DataRowVersion.Original]);
        deleteCmd.ExecuteNonQuery();
    }
}
transaction.Commit();
connection.Close();
stockEditDataTable.AcceptChanges();
MessageBox.Show("Зміни успішно збережено в базі даних. Час: 12:06
AM EEST, 17 травня 2025 року.", "Успіх", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
catch (Exception ex)
{
    if (connection.State == ConnectionState.Open)
    {
        connection.Close();
    }
    MessageBox.Show($"Помилка збереження змін: {ex.Message}",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void refreshButton_Click(object sender, EventArgs e)
{
    LoadStockEdit();
    MessageBox.Show("Дані оновлено з бази даних. Час: 12:06 AM EEST, 17
травня 2025 року.", "Успіх", MessageBoxButtons.OK, MessageBoxIcon.Information);
} } }

```