

**Державний торговельно-економічний університет**

**Кафедра комп'ютерних наук та інформаційних систем**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему:

**«Розробка комп'ютерної системи обліку оптової закупівлі  
товарів та їх роздрібного збуту торговельним  
підприємством»**

Студента 5 курсу, 2з групи,  
спеціальності  
126 «Інформаційні системи та  
технології»

\_\_\_\_\_

*підпис керівника*

Бесарабець  
Ігор  
Вікторович

Науковий керівник  
Кандидат технічних наук,  
доцент

\_\_\_\_\_

*підпис керівника*

Демідов Павло  
Георгійович

Гарант освітньої програми  
PhD з інформаційних технологій,  
старший викладач

\_\_\_\_\_

*підпис керівника*

Тищенко Ігор  
Анатолійович

**Київ 2025**

# Державний торговельно-економічний університет

Факультет інформаційних технологій  
Кафедра комп'ютерних наук та інформаційних систем  
Спеціальність 126 «Інформаційні системи та технології»  
Освітня програма «Інформаційні системи та технології»

Зав. кафедри \_\_\_\_\_ **Затверджую**  
Пурський О.І.  
«5» березня 2024р.

## Завдання на кваліфікаційну роботу студенту

**Бесарабець Ігор Вікторович**

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

«Розробка комп'ютерної системи обліку оптової закупівлі товарів та їх роздрібного збуту торговельним підприємством»

Затверджена наказом ректора від «01» березня 2024 р. № 730

2. Строк здачі студентом закінченої роботи 11 лютого 2025 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробка комп'ютерної системи обліку товарів для автоматизації процесів оптової закупівлі та роздрібного збуту, що дозволить підвищити ефективність управління товарними запасами на торговельному підприємстві.

Об'єкт дослідження: процеси розробки комп'ютерної системи з обліку товарів на торговельному підприємстві, яке охоплює оптову закупівлю та роздрібний збут.

Предмет дослідження: методи та засоби автоматизації обліку товарів з використанням комп'ютерних технологій.

4. Перелік графічного матеріалу

---

---

---

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Демідов П. Г.	05.03.2024 р.	05.03.2024 р.
2	Демідов П. Г.	05.03.2024 р.	05.03.2024 р.
3	Демідов П. Г.	05.03.2024 р.	05.03.2024 р.

6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)  
ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ОБЛІКУ ТОВАРІВ У ТОРГІВЛІ

1.1. Основи обліку оптових закупівель товарів

1.2. Характеристика процесу роздрібного збуту товарів

1.3. Значення комп'ютерних систем в управлінні обліком товарів

1.4. Вимоги до автоматизованих систем обліку в торгівлі

РОЗДІЛ 2. АНАЛІЗ І СТРУКТУРА КОМП'ЮТЕРНОЇ СИСТЕМИ ОБЛІКУ  
ТОВАРІВ

2.1. Функціональні вимоги до системи обліку оптових закупівель і роздрібного збуту

2.2. Архітектура комп'ютерної системи обліку товарів

2.3. Розробка бази даних для обліку товарів

РОЗДІЛ 3. РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОБЛІКУ ОПТОВОЇ  
ЗАКУПІВЛІ ТОВАРІВ ТА ЇХ РОЗДРІБНОГО ЗБУТУ

3.1. Інтерфейс користувача комп'ютерної системи обліку товарів

3.2. Алгоритми обробки даних у системі обліку товарів

3.3. Тестування та впровадження системи обліку товарів

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

## 7. Календарний план виконання роботи

№ Пор.	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми кваліфікаційної роботи</i>	10.02.2024	10.02.2024
2	<i>Розробка та затвердження завдання на кваліфікаційну роботу</i>	05.03.2024	05.03.2024
3	<i>Вступ</i>	03.04.2024	03.04.2024
4	<i>РОЗДІЛ 1. Теоретичні аспекти оцінки конкурентоспроможності підприємства</i>	12.06.2024	12.06.2024
5	<i>РОЗДІЛ 2. Математичні моделі оцінки та управління конкурентоспроможністю підприємства</i>	02.09.2024	02.09.2024
6	<i>РОЗДІЛ 3. Інформаційна технологія оцінки конкурентоспроможності підприємств електронної торгівлі</i>	01.11.2024	01.11.2024
7	<i>Висновки</i>	02.12.2024	02.12.2024
8	<i>Здача кваліфікаційної роботи на кафедрі науковому керівнику</i>	31.01.2025	31.01.2025
9	<i>Попередній захист кваліфікаційної роботи</i>	04.02.2025	04.02.2025
10	<i>Виправлення зауважень, зовнішнє рецензування кваліфікаційної роботи</i>	07.02.2025	07.02.2025
12	<i>Представлення готової зшитої кваліфікаційної роботи на кафедрі</i>	11.02.2025	11.02.2025
13	<i>Публічний захист кваліфікаційної роботи</i>	За розкладом роботи ЕК	

Дата видачі завдання «05.03.2024 р.»

9. Керівник кваліфікаційної роботи

Демідов П.Г.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Тищенко І.А.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент

Бесарабець І.В.

(прізвище, ініціали, підпис)

## 12. Відгук керівника кваліфікаційної роботи

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Керівник кваліфікаційної роботи

\_\_\_\_\_ *(підпис, дата)*

## 13. Висновок про кваліфікаційну роботу

Кваліфікаційна робота студента

Бесарабець І.В.

*(прізвище, ініціали)*

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми

Тищенко І.А.

*(підпис, прізвище, ініціали)*

Завідувач кафедри

Пурський О.І.

*(підпис, прізвище, ініціали)*

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

## АНОТАЦІЯ

Дана робота присвячена розробці комп'ютерної системи для автоматизації процесів обліку оптової закупівлі товарів та їх роздрібного збуту на торговельному підприємстві. У роботі розглянуто теоретичні аспекти обліку товарів, проведено аналіз існуючих програмних рішень, визначено функціональні вимоги до системи та побудовано її архітектуру. Описано створення бази даних, інтерфейсу користувача, а також алгоритмів обробки даних для забезпечення ефективної роботи системи. Проведено тестування розробленого програмного забезпечення та запропоновано рекомендації щодо його подальшого вдосконалення та інтеграції.

У результаті роботи створено систему, яка дозволяє автоматизувати облік товарних запасів, оптимізувати обробку даних про закупівлі та продажі, а також підвищити точність та швидкість виконання операцій. Система забезпечує зручність використання для користувачів завдяки інтуїтивному інтерфейсу та широкому функціоналу для управління товарними запасами.

Ключові слова: автоматизація обліку, управління запасами, база даних, торговельне підприємство.

## **ABSTRACT**

This work is devoted to the development of a computer system for automating the processes of accounting for the wholesale purchase of goods and their retail sale at a trading company. The paper examines theoretical aspects of accounting for goods, analyzes existing software solutions, defines functional requirements for the system, and constructs its architecture. The creation of a database, a user interface, as well as data processing algorithms to ensure the efficient operation of the system is described. The developed software was tested and recommendations for its further improvement and integration were offered.

As a result of the work, a system was created that allows you to automate inventory accounting, optimize the processing of data on purchases and sales, and also increase the accuracy and speed of operations. The system provides ease of use for users thanks to an intuitive interface and extensive functionality for managing inventory.

Keywords: accounting automation, inventory management, database, trading company.

## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ОБЛІКУ ТОВАРІВ У ТОРГІВЛІ .....	12
1.1. Основи обліку оптових закупівель товарів .....	12
1.2. Характеристика процесу роздрібного збуту товарів.....	20
1.3. Значення комп'ютерних систем в управлінні обліком товарів .....	25
1.4. Вимоги до автоматизованих систем обліку в торгівлі.....	29
РОЗДІЛ 2. АНАЛІЗ І СТРУКТУРА КОМП'ЮТЕРНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ.....	31
2.1. Функціональні вимоги до системи обліку оптових закупівель і роздрібного збуту.....	31
2.2. Архітектура комп'ютерної системи обліку товарів .....	32
2.3. Розробка бази даних для обліку товарів .....	38
РОЗДІЛ 3. РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОБЛІКУ ОПТОВОЇ ЗАКУПІВЛІ ТОВАРІВ ТА ЇХ РОЗДРІБНОГО ЗБУТУ.....	44
3.1. Інтерфейс користувача комп'ютерної системи обліку товарів .....	44
3.2. Алгоритми обробки даних у системі обліку товарів.....	49
3.3. Тестування та впровадження системи обліку товарів.....	52
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТОК А.....	64

## ВСТУП

У сучасних умовах розвитку інформаційних технологій і високої конкуренції на ринку торговельні підприємства стикаються з необхідністю оптимізації своїх процесів для досягнення високої продуктивності та ефективності. Одним з ключових аспектів, що впливає на успішність компанії, є управління оптовими закупівлями товарів та їх подальшою роздрібною реалізацією. Цей процес потребує ефективного обліку і контролю за запасами, витратами та продажами, а також чіткої організації роботи із постачальниками й покупцями. Розробка комп'ютерної системи для автоматизації цього процесу стає не лише доцільною, але й необхідною умовою для досягнення стабільних результатів та розвитку бізнесу.

Комп'ютерна система обліку оптової закупівлі товарів та їх роздрібного збуту дозволяє значно поліпшити контроль за ланцюгом постачання, мінімізувати ризики надмірних або недостатніх запасів та забезпечити прозорість усього процесу реалізації товарів [1]. Завдяки автоматизованій системі обліку, підприємство може оперативнo відстежувати залишки на складах, аналізувати попит, планувати замовлення і формувати прогнози на основі реальних даних. Це дає можливість торговельним компаніям швидше реагувати на зміни попиту і гнучко адаптувати свої стратегії збуту [2]. Впровадження такої комп'ютерної системи також сприяє оптимізації документообігу, полегшує роботу з бухгалтерською звітністю, скорочує кількість помилок при обробці замовлень та зменшує ризик людського фактору. Окрім цього, система забезпечує доступ до даних у режимі реального часу, що дозволяє керівництву приймати обґрунтовані управлінські рішення, знижувати витрати та покращувати загальну конкурентоспроможність підприємства.

**Актуальність дослідження:** у сучасних умовах розвитку торгівлі та зростання конкурентного тиску ефективне управління товарними запасами і процесом обліку набуває ключового значення для торговельних підприємств. Традиційні методи обліку часто не забезпечують необхідної швидкості обробки інформації та надійності даних. Впровадження комп'ютерних систем для обліку

оптових закупівель і роздрібного збуту дозволяє автоматизувати облік, мінімізувати помилки, підвищити продуктивність та покращити контроль за товарними залишками. Тому розробка і впровадження сучасної комп'ютерної системи обліку є актуальним кроком для досягнення стратегічних цілей підприємства.

**Мета дослідження:** розробка комп'ютерної системи обліку товарів для автоматизації процесів оптової закупівлі та роздрібного збуту, що дозволить підвищити ефективність управління товарними запасами на торговельному підприємстві.

**Завдання дослідження:**

1. Вивчити теоретичні аспекти обліку товарів у торгівлі та значення комп'ютерних систем в управлінні обліком.
2. Визначити функціональні вимоги та архітектуру комп'ютерної системи обліку товарів.
3. Розробити базу даних для обліку товарів.
4. Створити інтерфейс користувача та алгоритми обробки даних у системі обліку.
5. Провести тестування комп'ютерної системи та описати можливості її впровадження.

**Об'єкт дослідження:** процеси розробки комп'ютерної системи з обліку товарів на торговельному підприємстві, яке охоплює оптову закупівлю та роздрібний збут.

**Предмет дослідження:** методи та засоби автоматизації обліку товарів з використанням комп'ютерних технологій.

**Методи дослідження:** *аналіз літератури та нормативних документів* - для визначення теоретичних основ обліку товарів у торгівлі. *Проектування баз даних* - для створення структурованої системи зберігання та обробки даних. *Метод структурного моделювання* - для розробки архітектури та функціональних вимог комп'ютерної системи. *Метод тестування* - для оцінки якості роботи системи, її продуктивності та відповідності вимогам.

**Практичне значення** розробленої системи полягає у створенні автоматизованої комп'ютерної системи для обліку оптової закупівлі товарів та їх роздрібного збуту, яка забезпечує точність, швидкість та зручність обробки даних. Вона дозволяє ефективно управляти запасами, вести детальний облік надходжень і продажів, контролювати взаємодію з постачальниками та оптимізувати комерційні процеси торгівельного підприємства. Це сприятиме зниженню операційних витрат, підвищенню продуктивності роботи персоналу та забезпеченню конкурентоспроможності підприємства на ринку.

**Структура та обсяг кваліфікаційної роботи.** До змісту роботи входять: вступ, три розділи, висновки по роботі, перелік використаних літературних джерел та додатки. Обсяг роботи складає 60 сторінок. Робота має 28 рисунків, 11 таблиць та 30 посилань на використані літературні джерела.

# РОЗДІЛ 1

## ТЕОРЕТИЧНІ АСПЕКТИ ОБЛІКУ ТОВАРІВ У ТОРГІВЛІ

### 1.1. Основи обліку оптових закупівель товарів

Облік оптових закупівель є однією з ключових складових ефективного управління торговельним підприємством. Оптові закупівлі створюють основу для формування товарних запасів, від яких залежить можливість своєчасного задоволення попиту покупців, безперервність процесів збуту та фінансова стабільність компанії [3]. Ефективний облік оптових закупівель дозволяє підприємству не лише контролювати витрати, пов'язані з поповненням запасів, а й оптимізувати логістичні процеси, знижуючи витрати на зберігання, транспортування і обробку товарів.

На сучасному ринку облік оптових закупівель набуває особливої актуальності через потребу швидкого реагування на зміну попиту, коливання цін на товари та умови, що диктують конкуренцію. Правильна організація обліку закупівель допомагає підтримувати необхідний рівень товарних запасів, уникати проблем із дефіцитом чи надлишками продукції, а також оптимально планувати фінансові потоки [4]. Це стає можливим завдяки застосуванню сучасних інформаційних технологій та систем обліку, які забезпечують автоматизацію процесів і надають точні дані в реальному часі.

Процес обліку оптових закупівель забезпечує кілька ключових завдань [6]:

- контроль за кількістю та якістю закупленої продукції: дозволяє підприємству уникати нестачі і надлишків, що може призвести до втрат;
- оптимізація витрат: допомагає управляти закупівельними витратами, що знижує собівартість товарів;
- підтримка безперервності бізнес-процесів: завдяки налагодженому обліку запобігає перебоям у постачанні.

Організація обліку оптових закупівель вимагає наявності певної структури та процедури. Перш за все, важливо встановити стандарти обліку, що

відповідають законодавству та враховують специфіку діяльності підприємства. Стандартизація процесу обліку включає [7]:

- ведення первинної документації: договори з постачальниками, накладні, акти приймання товарів та рахунки-фактури;
- оцінка вартості закупівель: формування вартості товарів з урахуванням витрат на транспортування, страхування, митні збори та інші додаткові витрати;
- ведення аналітичного обліку: детальна інформація про кожну партію товарів, що дозволяє відстежувати витрати по кожній одиниці продукції.

Облік оптових закупівель товарів базується на первинній документації, яка слугує юридичною основою для фіксації кожної закупівлі. Основними документами є [8]:

- договір з постачальником: включає умови постачання, ціни, терміни та штрафні санкції за порушення умов;
- накладна: документ, що підтверджує факт передачі товару від постачальника до покупця;
- акт приймання товару: використовується для підтвердження кількості та якості поставленого товару, що особливо важливо при оптових закупівлях.

Ці документи забезпечують чіткий контроль за процесом закупівель, а також дають змогу перевірити виконання умов контракту, виявити дефекти чи розбіжності в кількості продукції.

Сучасний облік оптових закупівель дедалі частіше базується на автоматизованих інформаційних системах. Автоматизація обліку дає численні переваги:

- прискорення обробки інформації: зменшується час на введення та обробку даних, завдяки чому забезпечується швидший доступ до інформації;
- зменшення помилок: автоматичні системи дозволяють уникати помилок, характерних для ручного обліку;
- реальний час та аналітика: автоматизована система дозволяє отримувати інформацію про наявність товарів на складі, планувати поповнення запасів та оптимізувати обсяги замовлень.

Найбільш поширеними системами для автоматизації обліку оптових закупівель є спеціалізовані програмні комплекси, такі як 1С:Підприємство, SAP ERP та Microsoft Dynamics. Кожна з цих систем забезпечує інструменти для обліку товарів, контролю залишків, формування звітів та аналізу ефективності закупівель.

Вибір методу обліку і оцінки товарів є важливим аспектом обліку оптових закупівель. Основні методи включають:

- метод середньозваженої вартості: оцінка товарів на основі середньої вартості кожної одиниці товару;
- метод FIFO (перший прийшов - перший вийшов): облік товарів вважається таким, що перші закуплені товари першими ж і продаються [9].

Метою обліку оптових закупівель є створення прозорої системи контролю за закупівельними операціями, що дозволяє підприємству раціонально використовувати фінансові ресурси, забезпечуючи постійне відстеження товарних залишків і витрат. Зокрема, облік закупівель дозволяє досягати наступних завдань [10]:

- оцінка вартості товарних запасів та витрат на їх придбання;
- оптимізація закупівельних процесів з урахуванням попиту і пропозиції;
- забезпечення своєчасної та точної інформації для ухвалення управлінських рішень;
- запобігання можливим помилкам, пов'язаним із недооцінкою чи переоцінкою товарних запасів;
- формування аналітичних даних для оцінки ефективності закупівельної діяльності підприємства.

З огляду на цілі обліку, важливо дотримуватись загальноприйнятих стандартів та нормативних вимог, які регламентують процеси обліку, документообіг і звітність на підприємстві.

Сучасні комп'ютерні системи дозволяють значно поліпшити процес обліку оптових закупівель завдяки автоматизації рутинних операцій, що знижує ймовірність людських помилок і підвищує ефективність облікових процесів.

Інформаційні системи для обліку закупівель забезпечують можливість оперативного доступу до інформації про запаси на складах, поточні залишки, суми витрат на закупівлі та фінансові зобов'язання перед постачальниками.

Інтеграція комп'ютерної системи дозволяє в реальному часі [11]:

- відстежувати поставки, запаси і продажі товарів;
- планувати нові закупівлі, базуючись на даних про попит і наявність товарів на складах;
- оптимізувати фінансові витрати, що виникають при закупівлі та зберіганні товарів;
- забезпечувати аналітику витрат і доходів, що допомагає формувати стратегії закупівель.

До переваг автоматизованого обліку оптових закупівель належать [12]:

- швидкість та точність обліку - скорочується час на обробку даних і усувається ймовірність помилок через людський фактор;
- ефективне управління запасами - автоматизована система дозволяє відстежувати запаси на складі, уникати їхнього надлишку або дефіциту;
- аналітика та прогнозування - комп'ютерні системи надають доступ до детальних звітів і аналітичних даних, що допомагають приймати обґрунтовані управлінські рішення;
- зменшення операційних витрат - автоматизація процесів допомагає скоротити витрати на персонал та оптимізувати використання ресурсів.

Недоліками є:

- висока вартість впровадження - закупівля, налаштування і підтримка комп'ютерної системи потребують значних фінансових вкладень;
- потреба у навчанні персоналу - співробітники повинні пройти навчання, щоб освоїти нову систему, що може займати час і викликати початкові труднощі;
- технічні ризики - можливі технічні збої, які можуть призвести до втрати даних чи тимчасового призупинення роботи.

Облік оптових закупівель товарів є однією з ключових складових діяльності сучасного торговельного підприємства, що забезпечує ефективність і стабільність

його роботи на ринку. Даний аспект обліку сприяє не лише контролю фінансових потоків і управлінню запасами, а й оптимізації витрат, що пов'язані з постачанням і зберіганням товарів. Зі зростанням обсягу товарообігу та розширенням асортименту особливо важливим стає впровадження автоматизованих систем, що значно покращують точність і швидкість обробки даних. Це дозволяє підприємствам своєчасно відстежувати залишки на складах, контролювати ціни, аналізувати закупівельні витрати та прогнозувати майбутні потреби.

Сучасні технології для обліку оптових закупівель товарів включають в себе різноманітні програмні продукти, що забезпечують автоматизацію процесів закупівель, зберігання та обробку даних, а також інтеграцію з іншими елементами облікової системи підприємства.

Основні характеристики сучасних технологій:

- автоматизація обробки даних. Використання спеціалізованих програм для автоматизації введення даних дозволяє скоротити час на обробку інформації та знизити кількість помилок. Це забезпечує точний контроль над закупівлями і зменшує людський фактор;
- інтеграція з іншими системами [13]. Багато сучасних програм для обліку закупівель мають можливість інтеграції з іншими бізнес-системами підприємства, такими як бухгалтерський облік, склади та логістика, що дозволяє досягти комплексного управління процесами;
- гнучкість і масштабованість. Сучасні системи обліку можуть бути налаштовані під специфічні потреби підприємства і забезпечують можливість масштабування, що особливо важливо для великих підприємств, що здійснюють закупівлі на різних етапах ланцюга постачання;
- оперативний доступ до даних. Завдяки використанню баз даних і хмарних технологій, сучасні програми дозволяють здійснювати оперативний доступ до даних в реальному часі, що полегшує прийняття рішень.

*Програми для обліку оптових закупівель товарів*

Існує ряд спеціалізованих програмних продуктів для обліку закупівель, які дозволяють підприємствам автоматизувати цей процес, спрощувати управлінські

завдання та знижувати витрати. Ось кілька прикладів найбільш популярних програм:

### *1С:Підприємство*

1С:Підприємство - це універсальна програмна система, що дозволяє автоматизувати більшість бізнес-процесів, включаючи облік оптових закупівель [14]. Це одна з найпоширеніших систем в Україні та країнах СНД.

Особливості програми:

- можливість інтеграції з іншими модулями (склади, бухгалтерія, продажі);
- ведення обліку товарів, в тому числі облік закупівель і постачальників;
- автоматичний розрахунок цін закупівель, наявність функцій для ведення рахунків-фактур і актів приймання [15];
- модуль управління запасами, що дозволяє планувати закупівлі та контролювати рівень товарних запасів [16].

У табл. 1.1 приведено основні функціональні можливості програмного забезпечення 1С:Підприємство.

Таблиця 1.1

### Функціональні можливості 1С:Підприємство

Функція	Опис
Облік закупівель	Автоматичний облік всіх етапів закупівель (платежі, накладні, акти).
Підтримка багатьох складських ланок	Можливість ведення обліку для декількох складів і торгових точок.
Управління постачальниками	Формування та контроль контрактів, угод з постачальниками.
Інтеграція з бухгалтерією	Автоматичний обмін даними з програмою бухгалтерії для коректного фінансового обліку.
Аналітика і звітність	Формування звітів по закупівлях, витратах і прогнозах.

Отже, 1С:Підприємство є ефективним інструментом для автоматизації бізнес-процесів, зокрема обліку закупівель, управління запасами та

постачальниками. Система забезпечує інтеграцію з іншими модулями, такими як бухгалтерія та продажі, а також формує аналітичні звіти для підтримки прийняття управлінських рішень. Завдяки гнучкості налаштувань і широким функціональним можливостям, 1С:Підприємство залишається популярним рішенням для малого та середнього бізнесу в Україні та СНД.

### *SAP Business One*

SAP Business One - це комплексна система для автоматизації управлінських процесів малого та середнього бізнесу, включаючи закупівлі товарів [17]. Ця система допомагає компаніям вести облік поставок, відстежувати витрати та планувати майбутні закупівлі.

Особливості програми:

- інтуїтивно зрозумілий інтерфейс, який дозволяє швидко освоїти програму;
- можливість формування замовлень на закупівлю, облік витрат і прогнозування потреб [18];
- інтеграція з іншими модулями, такими як склад і фінанси [19];
- використання в хмарному варіанті для малого та середнього бізнесу [20].

Табл. 1.2 висвітлює основні функціональні можливості програмного забезпечення SAP Business One.

Таблиця 1.2

### Функціональні можливості SAP Business One

<b>Функція</b>	<b>Опис</b>
Управління закупівлями	Планування і відстеження всіх етапів закупівель.
Звітність по витратах і закупках	Формування детальних звітів про закупівельні операції та витрати.
Управління ланцюгами поставок	Ведення обліку поставок і забезпечення ланцюгів постачання.
Автоматизація обліку	Автоматичне створення рахунків та контролю за платежами.
Управління фінансами	Підключення до бухгалтерії та фінансового моніторингу.

Отже, SAP Business One є універсальним рішенням для автоматизації управлінських процесів малого та середнього бізнесу, зокрема в галузі закупівель і фінансів. Система дозволяє ефективно планувати закупівлі, відстежувати витрати та формувати звіти для аналізу діяльності. Завдяки можливості інтеграції з іншими модулями та підтримці хмарних технологій, SAP Business One забезпечує гнучкість і зручність у використанні для бізнесу різного масштабу.

### *TradeSoft*

TradeSoft - це система для автоматизації оптової торгівлі, що спеціалізується на управлінні закупівлями, поставками і запасами товарів [21].

Особливості програми:

- простота в налаштуванні та використанні [22];
- облік товарів на складах, автоматичне відображення вартості закупівель;
- функції для формування звітів та аналітики[23];
- можливість інтеграції з електронною поштою для управління замовленнями [24].

У табл. 1.3 приводиться основні функціональні можливості програмного рішення TradeSoft.

Таблиця 1.3

### Функціональні можливості TradeSoft

<b>Функція</b>	<b>Опис</b>
Облік товарів	Ведення обліку товарів на складах, а також контролю залишків.
Аналіз витрат	Формування звітів по витратах на закупівлю товарів та їх транспортування.
Інтерфейс замовлень	Управління замовленнями від постачальників через електронну пошту та систему.
Звітність і аналітика	Генерація звітів по закупівлях, витратах і постачальниках.

Отже, TradeSoft є ефективним програмним рішенням для автоматизації процесів оптової торгівлі, зокрема управління закупівлями, поставками та обліком

товарних запасів. Система забезпечує зручний інтерфейс, інтеграцію з електронною поштою для управління замовленнями та функції для аналітики та звітності. Завдяки простоті використання і широким можливостям налаштування, TradeSoft є надійним інструментом для оптимізації бізнес-процесів у сфері оптової торгівлі.

Сучасні комп'ютерні програми для обліку оптових закупівель товарів дозволяють значно підвищити ефективність і точність управління закупівлями, знижуючи витрати і зменшуючи ймовірність помилок. Вибір конкретної програми залежить від розміру підприємства, його потреб та специфіки діяльності. Всі згадані системи - 1С:Підприємство, SAP Business One, TradeSoft, Microsoft Dynamics NAV - є популярними і ефективними інструментами для автоматизації обліку, що дозволяють полегшити роботу з закупівлями і забезпечити належний контроль на

У підсумку, впровадження автоматизованих систем обліку оптових закупівель товарів відкриває значні перспективи для розвитку бізнесу. Вони не лише підвищують конкурентоспроможність підприємства на ринку, але й дозволяють ефективніше використовувати ресурси, що, в свою чергу, сприяє зростанню прибутковості та фінансової стійкості. Використання таких систем у поєднанні з регулярним аналізом закупівельних даних допомагає оптимізувати логістичні та управлінські процеси, що є вагомим перевагою в умовах швидко змінного бізнес-середовища.

## **1.2. Характеристика процесу роздрібного збуту товарів**

Процес роздрібного збуту товарів є важливою складовою економічної діяльності підприємств, спрямованою на задоволення споживчого попиту. Він охоплює комплекс операцій, які здійснюються для доведення товарів від виробника до кінцевого споживача, забезпечуючи ефективну комунікацію між учасниками ринку. В умовах сучасної конкуренції та глобалізації бізнесу роздрібний збут став ключовим елементом формування конкурентних переваг

підприємств, зокрема через адаптацію до змінних запитів споживачів, використання інноваційних технологій та оптимізацію логістичних процесів.

У роздрібному збуті поєднуються різноманітні маркетингові та управлінські функції: від дослідження ринку і вибору асортименту до організації ефективної системи продажу і післяпродажного обслуговування клієнтів [25]. Такий підхід дозволяє забезпечити оптимальний рівень сервісу та утримувати лояльність споживачів, що є важливим для стабільного розвитку компанії.

Особливості роздрібного збуту полягають також у використанні різноманітних каналів дистрибуції, серед яких виділяють традиційні магазини, онлайн-платформи, мобільні застосунки та навіть соціальні мережі. Це вимагає від підприємств ефективної взаємодії з партнерами та адаптації своїх бізнес-процесів до нових умов.

Роздрібний збут товарів є невід'ємною складовою економічної системи та життєво важливим процесом для компаній, що прагнуть ефективно задовольняти потреби кінцевих споживачів. Він не лише виконує функцію передачі товарів від виробника до споживача, а й формує цінності бренду, впливає на сприйняття продукції та рівень задоволеності клієнтів. Здатність правильно організувати роздрібний збут має суттєвий вплив на фінансові результати підприємства, забезпечуючи стабільні продажі, розширення ринку та зростання лояльності покупців.

Роздрібний збут товарів відіграє ключову роль у забезпеченні економічної стабільності та розвитку компаній, орієнтованих на кінцевого споживача. Різноманітність каналів збуту, таких як традиційні офлайн-магазини, цифрові платформи, соціальні мережі та мобільні застосунки, дозволяє бізнесу охоплювати широку аудиторію та задовольняти різні споживчі потреби. Правильний вибір та інтеграція цих каналів сприяють підвищенню лояльності клієнтів, розширенню ринкових позицій і покращенню фінансових результатів.

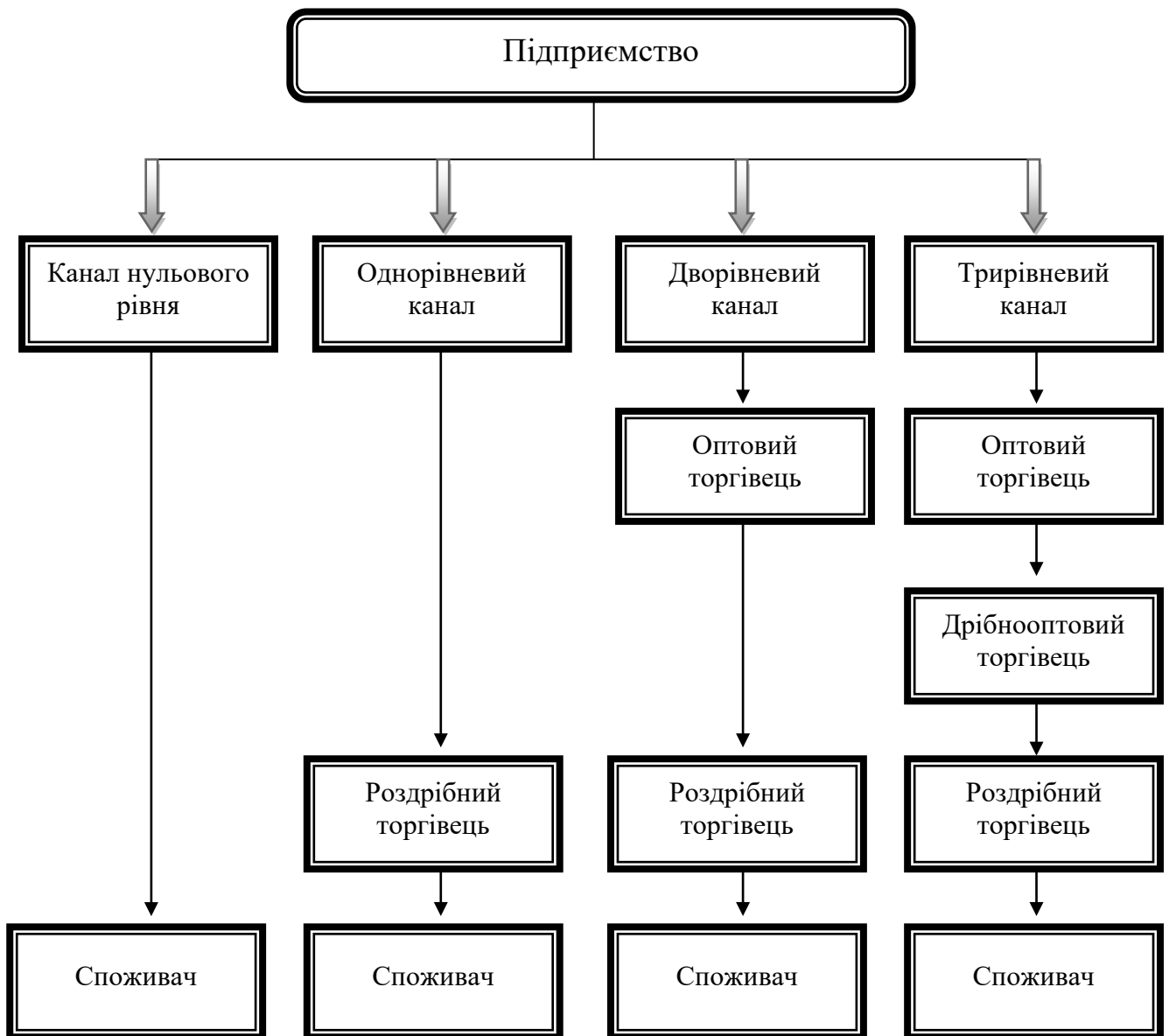


Рис. 1.1. Схема системи розподілу роздрібно збути товарів

Роздрібний збут, як канал розподілу, дозволяє розміщувати продукцію у місцях найбільшого попиту, враховуючи зручність для клієнтів та їхні потреби. Це сприяє оптимальному використанню ресурсів, створюючи баланс між виробництвом та споживанням. Завдяки цьому каналу споживачі мають доступ до необхідних товарів, тоді як виробники отримують відгуки та дізнаються про ринкові тенденції, що є цінним інструментом для планування подальшої діяльності. Роздрібний збут є також джерелом численних робочих місць, що сприяє економічному розвитку, забезпечуючи прибутки не тільки компаніям, але й державі у вигляді податкових надходжень.

## 1. Етапи процесу роздрібного збуту

Процес роздрібного збуту охоплює декілька важливих етапів, які забезпечують ефективну реалізацію товарів та задоволення потреб клієнтів [26]:

- вибір асортименту товарів. На першому етапі визначається асортимент, який буде представлений споживачам. Важливо, щоб він відповідав вимогам ринку, враховував популярні тенденції, сезонність та вподобання цільової аудиторії. На цьому етапі проводиться аналіз попиту та пропозиції, враховується конкурентне середовище, а також економічні фактори, що впливають на купівельну спроможність.

- формування цінової політики. Роздрібна ціна є важливим чинником, що визначає попит на товар. При її формуванні враховуються виробничі та логістичні витрати, ціни конкурентів, цінність продукту для споживача та можливості надання знижок або акцій. Правильна цінова політика сприяє підвищенню конкурентоспроможності товарів та залученню нових покупців.

- організація презентації продукції. Для успішного збуту велике значення має презентація товарів. Вона включає не лише оформлення місця продажу, але й викладку продукції, освітлення, оформлення вітрин, навігаційні елементи тощо. Чітка та приваблива презентація допомагає привернути увагу клієнтів, спонукає до покупки та створює позитивне враження про бренд.

- управління запасами та постачанням. Безперервна наявність товарів є важливим чинником, що впливає на рівень задоволеності покупців. Недостатній обсяг товарів або надлишок запасів може призвести до фінансових втрат. Управління запасами включає моніторинг рівня товарів на складі, планування закупівель, прогнозування попиту, а також взаємодію з постачальниками.

- обслуговування клієнтів. Рівень сервісу, що надається клієнтам, безпосередньо впливає на їхнє задоволення та лояльність. Сюди входять консультації, допомога у виборі товару, обробка запитів, надання знижок та акцій, а також робота з відгуками клієнтів. Висока якість обслуговування сприяє створенню позитивного іміджу компанії та стимулює повторні покупки.

## 2. Технологічні зміни в роздрібному збуті

Цифровізація та розвиток технологій значно вплинули на процес роздрібного збуту товарів. Серед головних тенденцій можна виділити:

Електронна комерція. Онлайн-платформи та інтернет-магазини стали важливим каналом збуту, надаючи споживачам зручність вибору, порівняння цін і оформлення покупки безпосередньо з дому. Електронна комерція дозволяє розширити ринок збуту, охоплюючи віддалені регіони, знижує витрати на оренду приміщень і забезпечує швидкий зворотний зв'язок від споживачів.

Соціальні мережі. Інстаграм, Фейсбук, ТікТок та інші соціальні мережі активно використовуються для просування товарів, взаємодії з аудиторією та збору зворотного зв'язку. Це дозволяє не тільки підвищити впізнаваність бренду, але й залучити нових покупців.

Використання аналітики та штучного інтелекту. Аналітичні інструменти дозволяють вивчати поведінку покупців, прогнозувати попит, автоматизувати управління запасами та оптимізувати асортимент. Штучний інтелект використовується для персоналізації пропозицій, автоматизації обслуговування клієнтів через чат-боти та створення рекомендацій на основі аналізу попередніх покупок.

## 3. Взаємодія з клієнтами та управління лояльністю

Роздрібний збут не обмежується одноразовою продажем товару; він також передбачає формування тривалих відносин з клієнтами. Важливу роль відіграє управління лояльністю, яке включає програми лояльності, надання бонусів, системи знижок та персоналізовані пропозиції. Такий підхід допомагає не лише збільшити обсяги продажів, але й підвищити довіру до бренду, що особливо важливо в умовах високої конкуренції.

## 4. Виклики та перспективи розвитку роздрібного збуту

Сучасний ринок роздрібної торгівлі стикається з численними викликами, серед яких особливо виразними є посилення конкуренції, зміни у споживчій поведінці, стрімкий розвиток технологій та економічні коливання. У табл. 1.4 наведено основні етапи процесу організації роздрібного збуту товарів.

## Етапи процесу роздрібного збуту товарів

Етап збуту	Основні дії	Цілі	Очікувані результати
1. Вибір асортименту	Аналіз попиту, визначення популярних товарів	Формування оптимального асортименту товарів	Відповідність потребам ринку, задоволення клієнтів
2. Формування цінової політики	Встановлення ціни на основі витрат та аналізу конкурентів	Оптимізація ціноутворення для залучення клієнтів	Конкурентоспроможні ціни, привабливість товарів
3. Презентація продукції	Оформлення торгових площ, викладка товарів	Створення привабливого вигляду товарів для залучення клієнтів	Позитивний імідж бренду, зростання обсягу продажів
4. Управління запасами	Моніторинг залишків	Забезпечення постійної наявності товарів	Мінімізація витрат, оптимізація оборотності запасів
5. Обслуговування	Надавання консультацій	Підвищення задоволеності клієнтів	Зростання повторних покупок, позитивні відгуки
6. Аналіз зворотного зв'язку	Збір та аналіз відгуків клієнтів	Виявлення переваг і недоліків товарів та сервісу	Підвищення якості обслуговування, адаптація під потреби клієнтів

Загалом, процес роздрібного збуту товарів спрямований на створення оптимальної комбінації продукту, ціни, доступності та обслуговування для формування конкурентних переваг і лояльності клієнтів. Підприємства, які успішно інтегрують новітні цифрові технології та підтримують високу якість обслуговування, мають більші шанси не лише залучати нових клієнтів, а й утримувати постійних, що, врешті-решт, сприяє сталому розвитку та зростанню бізнесу.

### 1.3. Значення комп'ютерних систем в управлінні обліком товарів

Сучасний бізнес усе більше залежить від технологій, і однією з ключових сфер, де комп'ютерні системи відіграють вирішальну роль, є управління обліком

товарів. Автоматизація обліку товарів дозволяє підприємствам ефективніше контролювати свої запаси, оптимізувати процеси постачання, зберігання та продажу продукції. Комп'ютерні системи не лише допомагають здійснювати щоденний контроль за наявністю товарів, але й забезпечують більш глибокий аналіз даних, виявлення закономірностей попиту та прогнозування потреб у товарах на майбутнє.

Комп'ютерні системи в управлінні обліком товарів забезпечують кілька важливих функцій, таких як збір і обробка даних у режимі реального часу, миттєве оновлення інформації про запаси, автоматичне формування звітності та аналіз операцій. Такі можливості дозволяють керівництву оперативно приймати обґрунтовані рішення, знижуючи ризики неправильного управління запасами та витратами. Використання сучасного програмного забезпечення допомагає скоротити кількість помилок, які можуть виникати при ручному введенні даних, і спрощує координацію між різними відділами підприємства.

Важливе значення має й інтеграція комп'ютерних систем обліку товарів з іншими бізнес-процесами, такими як продажі, постачання, фінансовий облік і управління клієнтськими відносинами. Такі інтегровані системи дозволяють створювати єдину інформаційну платформу, що полегшує доступ до даних, покращує прозорість операцій та сприяє зростанню ефективності роботи компанії. Це, у свою чергу, підвищує конкурентоспроможність підприємства, забезпечуючи швидшу реакцію на ринкові зміни та адаптацію до нових умов.

Крім того, сучасні системи обліку на основі комп'ютерних технологій, таких як RFID, штрихкоди, IoT і хмарні платформи, відкривають додаткові можливості для підприємств. Вони дозволяють впроваджувати новітні методи контролю за товарами та вдосконалювати логістичні процеси. Хмарні рішення, наприклад, дають змогу керувати запасами з будь-якої точки світу, а інтеграція технології штучного інтелекту допомагає прогнозувати попит і оптимізувати запаси.

Комп'ютерні системи управління обліком товарів (Inventory Management Systems, IMS) створені для автоматизації, зберігання та обробки даних про запаси [27]. Основна мета таких систем – забезпечити ефективне та точне управління

товарами, уникнути втрат через дефіцит чи надлишок запасів, та оптимізувати витрати підприємства. У сучасних умовах IT-інфраструктура та програмне забезпечення, як-от ERP (Enterprise Resource Planning), виконують ключову роль у координації та контролі всіх етапів логістичного циклу – від закупівлі та зберігання до продажу товарів.

Комп'ютеризовані системи обліку, зазвичай, складаються з трьох основних компонентів:

1. **Фізичні компоненти (апаратне забезпечення)** – комп'ютери, сервери, сканери штрих-кодів, RFID-зчитувачі та принтери.
2. **Програмні компоненти** – системи управління базами даних (DBMS), спеціалізовані програми для обліку та звітності.
3. **Комунікаційні технології** – мережі та канали зв'язку, через які відбувається передача даних у реальному часі.

Комп'ютеризовані системи використовуються на всіх етапах управління запасами: від автоматичного введення даних до аналізу операцій за допомогою спеціалізованого ПЗ, зокрема алгоритмів штучного інтелекту (AI) та машинного навчання (ML), що підвищує точність прогнозування попиту.

Бази даних є основою будь-якої системи обліку, зберігаючи та структуровано організуючи дані про товари. У базах даних міститься інформація про кількість наявних товарів, місце їх зберігання, вартість та терміни зберігання. Найпоширеніші системи управління базами даних (СУБД), такі як PostgreSQL, MySQL, Oracle, MongoDB, використовуються для ефективного обліку товарів завдяки своїм можливостям обробки великих обсягів даних та забезпеченню швидкого доступу до інформації.

Типова структура бази даних для обліку товарів включає кілька таблиць:

- **товари.** Інформація про найменування товарів, їх коди, штрих-коди або RFID-теги, опис;
- **запаси.** Кількість товарів у різних точках зберігання, склади, дати постачань;

- **операції.** Інформація про переміщення товарів, дати продажів, дані про замовлення та поставки;
- **аналітика.** Підрахунок середньої оборотності запасів, прогнозування попиту тощо.

Важливою складовою таких баз даних є індексація та оптимізація запитів для швидкого доступу до інформації, що особливо важливо для великих підприємств із значними обсягами даних. Крім того, бази даних можна інтегрувати з іншими системами, наприклад, з бухгалтерським або фінансовим обліком, що дозволяє отримувати зведену інформацію про витрати та доходи в реальному часі.

Ідентифікація товарів є важливою функцією комп'ютерних систем обліку, і для цього широко використовуються штрих-коди та технологія RFID (Radio-Frequency Identification).

**Штрих-коди** дозволяють прискорити процес сканування товарів під час інвентаризації та продажу. Кожен товар має унікальний код, що зчитується сканером, а дані автоматично передаються в базу даних. Це значно знижує ризик помилок, характерних для ручного введення даних.

**RFID-технології** є більш сучасним та ефективним засобом ідентифікації, особливо для великих складів. RFID-чіпи не потребують прямої видимості, як у випадку штрих-кодів, і можуть зчитуватися з відстані. Це дає можливість автоматизувати процес обліку: під час проходження товарів через спеціальні RFID-зчитувачі, система автоматично оновлює кількість товарів у базі даних, що прискорює процес інвентаризації.

ERP-системи є багатофункціональними платформами, що дозволяють централізовано управляти всіма аспектами бізнесу, включаючи облік товарів. Такі системи, як SAP, Microsoft Dynamics, Oracle ERP, забезпечують єдиний доступ до даних у різних підрозділах підприємства, що підвищує прозорість і точність операцій. За допомогою ERP, дані про запаси автоматично оновлюються при кожному переміщенні товару, що забезпечує точність обліку.

#### **1.4. Вимоги до автоматизованих систем обліку в торгівлі**

Для автоматизованих систем обліку в торгівлі ставляться вимоги, що забезпечують точність і оперативність управління складськими запасами. Метою є створення інформаційної системи, яка обробляє дані про товари, постачальників, замовлення та формує аналітичну інформацію для підтримки прийняття управлінських рішень.

Основні задачі для реалізації системи:

1. Обробка вхідних даних. Розробка механізмів введення даних про товари, що надходять на склад, включаючи:

- назви товарів, їх кількість і характеристики.
- дані про постачальників: назва, контактні дані, умови поставок.
- інформацію про замовлення клієнтів: деталі замовлень, терміни виконання.
- дані про списання продукції (причини, обсяги).
- автоматизація перевірки точності введених даних і їхнього збереження.

2. Формування вихідної інформації. Реалізація модулів для створення аналітичних звітів, включаючи:

- стан складських запасів: поточні залишки, категоризація продукції;
- звіти про закупівлі, продажі та списання товарів: динаміка руху товарів, аналіз закупівельної активності;
- попередження про критичні рівні запасів: прогнозування необхідності замовлення додаткових товарів;
- розробка інтерфейсу для швидкого доступу до звітів і візуалізації даних.

На основі аналізу, рис. 1.2 демонструє ключові компоненти системи та взаємозв'язки між ними. Ця модель охоплює всі етапи процесу – від введення даних до формування звітів і прогнозів.

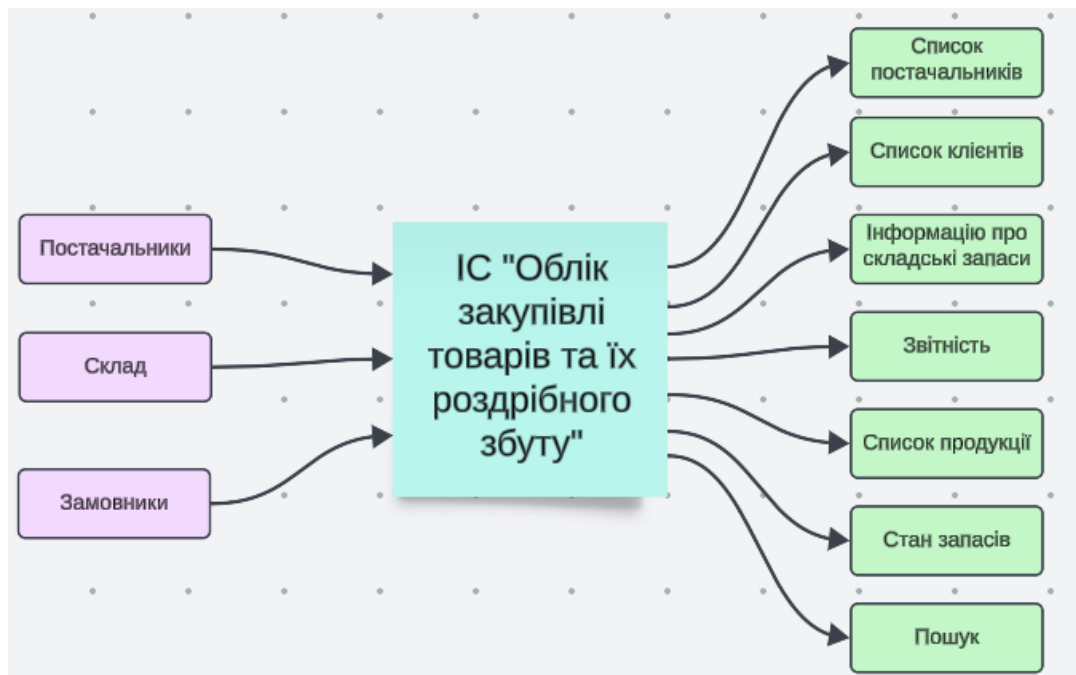


Рис. 1.2. Інформаційна модель поставленої задачі

Дотримання зазначених вимог дозволить створити інформаційну систему для управління складськими запасами, яка забезпечить підприємство інструментами для прийняття обґрунтованих управлінських рішень.

## РОЗДІЛ 2

### АНАЛІЗ І СТРУКТУРА КОМП'ЮТЕРНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ

#### 2.1. Функціональні вимоги до системи обліку оптових закупівель і роздрібного збуту

Система обліку оптових закупівель і роздрібного збуту має забезпечувати автоматизацію основних бізнес-процесів, пов'язаних із закупівлею, зберіганням, обліком і реалізацією товарів. Її функціонал спрямований на підтримку управління товарними запасами, підвищення ефективності операцій та надання інструментів для аналітики.

Ключові функціональні вимоги до системи включають:

- обробку даних про надходження товарів від постачальників та їх реалізацію кінцевим споживачам;
- ведення обліку складських запасів у режимі реального часу;
- автоматизацію процесів формування замовлень, рахунків-фактур і звітів.

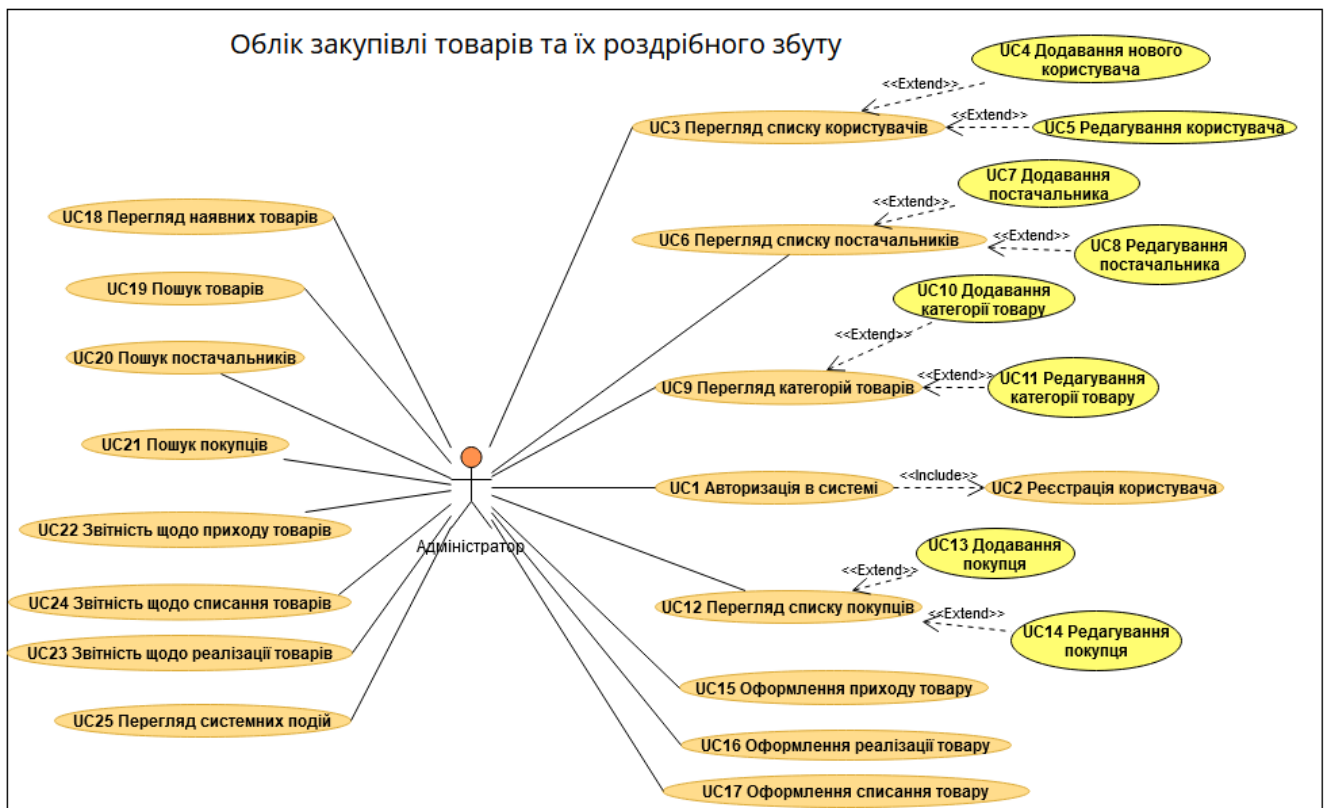


Рис. 2.1. Діаграма варіантів використання системи для ролі «адміністратор»

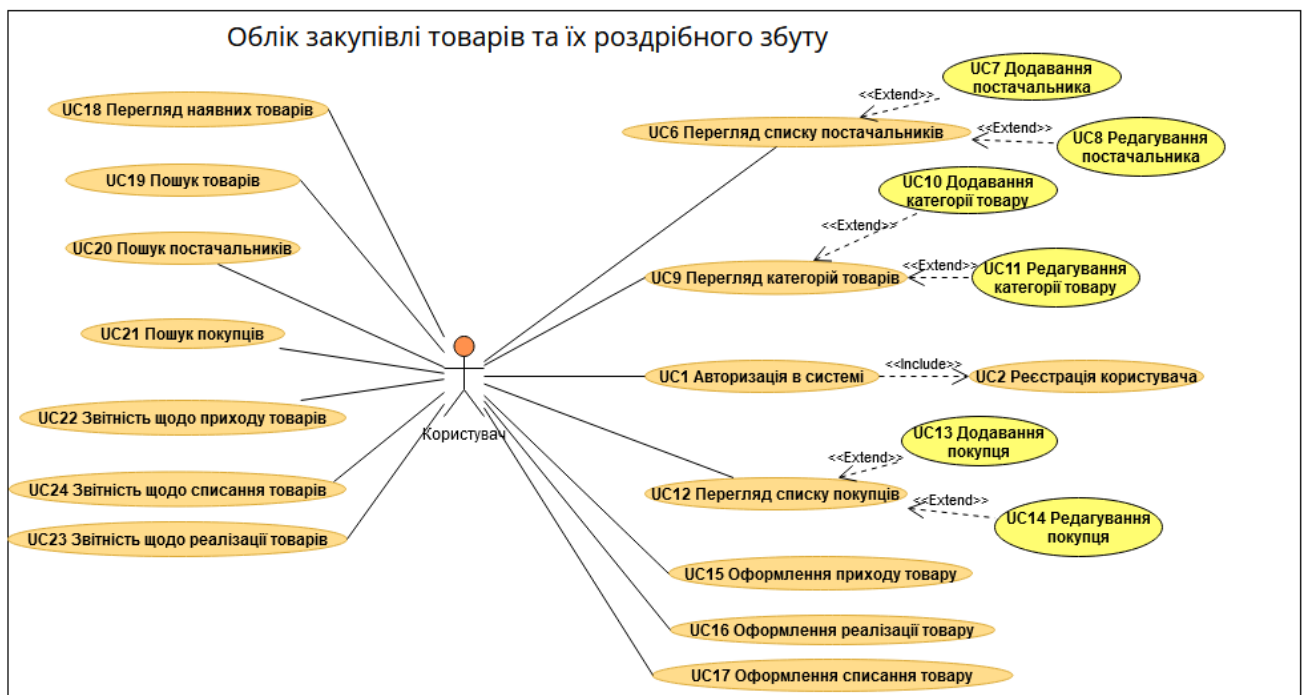


Рис. 2.2. Діаграма варіантів використання системи для ролі «користувач»

Для уточнення особливостей функціонування системи та визначення її можливостей розроблено варіанти використання [28], які демонструють типові сценарії роботи з нею. У додатку 3 наведено опис основних варіантів використання системи, які охоплюють ключові задачі, що вирішуються автоматизовано.

Для розуміння функціональних вимог до системи розроблено діаграми варіантів використання, яка ілюструє основні сценарії роботи для ролі «адміністратор». На рис. 2.1-2.2 наведено діаграми варіантів використання системи для ролі «адміністратор» та «користувач».

Діаграми дозволяють візуально представити взаємодію користувача із системою, підкреслюючи можливість виконання ключових бізнес-процесів. Завдяки цьому можна оптимізувати сценарії роботи та забезпечити простоту використання системи.

## 2.2. Архітектура комп'ютерної системи обліку товарів

Архітектура комп'ютерної системи обліку товарів визначає її структурну організацію, компоненти та способи взаємодії між ними. Для розробки системи було обрано трьохрівневу архітектуру [29], яка включає рівні презентації, логіки

та даних. Такий підхід забезпечує розподіл функцій між компонентами, що сприяє покращенню продуктивності, зручності обслуговування та масштабованості системи. Трьохрівнева архітектура дозволяє незалежно оновлювати інтерфейс користувача, бізнес-логіку та базу даних, що робить систему більш гнучкою до змін. Окрім того, вона спрощує інтеграцію з іншими підсистемами та забезпечує чіткий поділ відповідальності між розробниками різних рівнів системи [30]. Така архітектура є оптимальним вибором для автоматизованих систем обліку, які потребують роботи в реальному часі та підтримують складні бізнес-процеси.

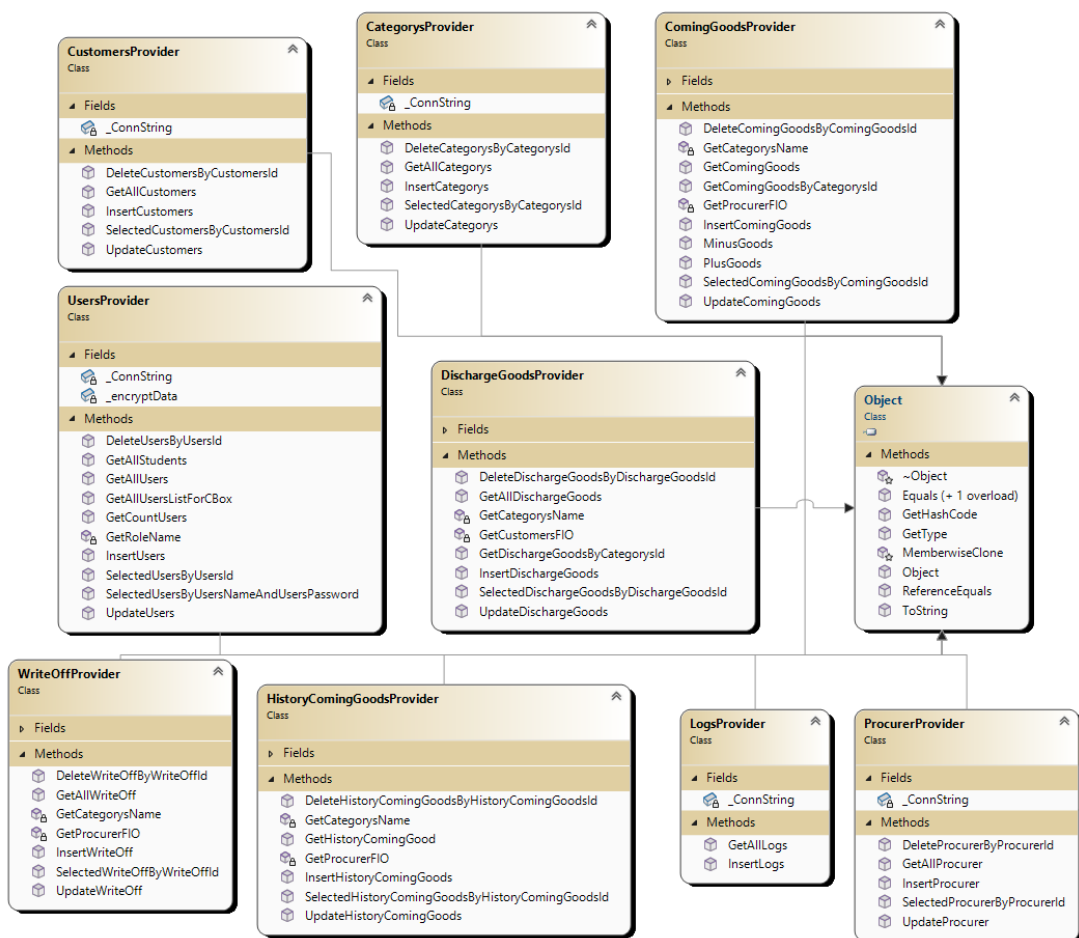


Рис. 2.3. Діаграма класів рівня даних

Для забезпечення ефективної роботи системи обліку оптових закупівель і роздрібного збуту був розроблений рівень доступу до даних, який відповідає за взаємодію з базою даних. Цей рівень забезпечує операції із зчитування, запису, оновлення та видалення даних, а також гарантує цілісність і актуальність

інформації. На рис. 2.3 наведена діаграма класів, що описує структуру та функціональність рівня доступу до даних розробленої системи.

Даний рівень даних складається з таких класів:

- CategoriesProvider відповідає за управління даними про категорії продукції. Він реалізує функціонал для отримання, оновлення, видалення та вибору категорій з бази даних, що забезпечує актуальність інформації про категорії продукції;
- CustomersProvider забезпечує роботу з даними про клієнтів. Він включає методи для отримання, додавання, редагування та видалення інформації про покупців, що дозволяє підтримувати актуальні дані про клієнтську базу;
- UsersProvider призначений для управління інформацією про користувачів системи. Клас реалізує методи для додавання, оновлення, видалення та вибору даних про користувачів, включаючи перевірку ідентифікаційних даних;
- ComingGoodsProvider відповідає за роботу з даними про надходження товарів. Він включає функції для додавання, оновлення, видалення та отримання інформації про поставки, що сприяє контролю за рухом товарів;
- DischargeGoodsProvider відповідає за управління даними про реалізацію товарів. Він забезпечує функціонал для обліку продажів, їх оновлення, видалення та перегляду, що дозволяє контролювати процес реалізації;
- WriteOffProvider забезпечує облік списаних товарів. Клас включає методи для додавання, видалення, оновлення та отримання даних про списання товарів, що важливо для ведення точного обліку складських запасів;
- HistoryComingGoodsProvider відповідає за зберігання історії надходжень товарів. Він реалізує методи для отримання інформації про попередні поставки, що сприяє аналітиці та моніторингу операцій;
- LogsProvider призначений для реєстрації системних подій. Він включає методи для додавання та перегляду логів, що дозволяє відстежувати активність у системі;

– ProcurerProvider забезпечує управління інформацією про постачальників. Він включає методи для додавання, оновлення, отримання та видалення даних про постачальників, що полегшує управління цим аспектом діяльності.

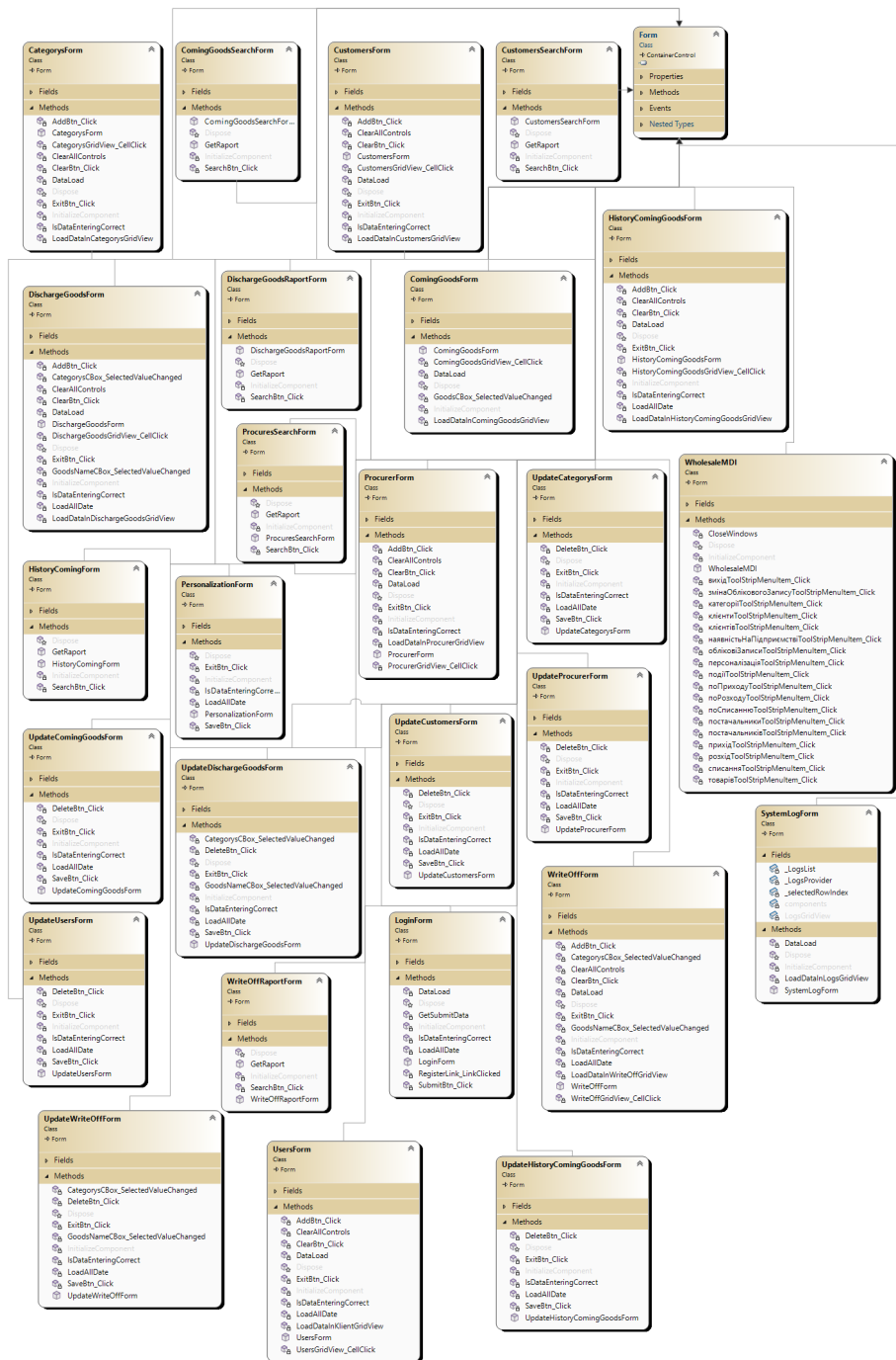


Рис. 2.4. Діаграма класів рівня користувацького інтерфейсу

Дана структура дозволяє розподілити функціональність між класами, забезпечуючи модульність і гнучкість у роботі системи.

Рівень користувацького інтерфейсу розробленої системи відповідає за взаємодію між користувачем та функціональністю системи. Він включає форми та компоненти, що забезпечують введення, відображення й обробку даних, необхідних для роботи з обліком закупівель і роздрібного збуту товарів. На рис. 2.4 представлена діаграма класів, що ілюструє структуру й організацію компонентів рівня користувацького інтерфейсу.

Даний рівень складається із наступних класів:

- CategoriesForm відповідає за управління категоріями продукції. Цей клас забезпечує відображення списку категорій, редагування їх даних, а також додавання нових категорій;
- CustomersForm реалізує інтерфейс для роботи з клієнтами. Він дозволяє переглядати інформацію про покупців, додавати нових клієнтів, редагувати наявні записи й видаляти їх за необхідності;
- ComingGoodsForm забезпечує введення даних про надходження товарів. Цей клас дозволяє користувачеві вказувати деталі нових поставок, переглядати історію надходжень і здійснювати редагування;
- DischargeGoodsForm реалізує інтерфейс для обліку реалізації товарів. Він забезпечує додавання записів про продажі, редагування даних і перегляд операцій, пов'язаних із реалізацією;
- WriteOffGoodsForm відповідає за облік списання товарів. Цей клас дозволяє користувачеві створювати записи про списання, редагувати їх і переглядати історію списаних товарів;
- LogsForm забезпечує відображення журналу системних подій. Він дозволяє адміністраторам переглядати деталі активності користувачів і роботи системи.
- HistoryComingGoodsForm дозволяє переглядати історію надходжень товарів із зазначенням дати, постачальника та характеристик поставки, що сприяє аналітиці та контролю;

– ProcurersForm забезпечує управління постачальниками. Цей клас дозволяє переглядати інформацію про постачальників, редагувати її, а також додавати нових постачальників.

Кожен клас на рівні користувацького інтерфейсу пов'язаний із відповідними класами рівня доступу до даних, забезпечуючи інтеграцію між користувачем і бізнес-логікою системи. Така структура дозволяє ефективно розподілити функціональність і спростити взаємодію із системою.

Рівень бізнес-логіки розробленої системи відповідає за обробку даних і реалізацію основних алгоритмів, що забезпечують виконання бізнес-процесів, пов'язаних із закупівлею, зберіганням і реалізацією товарів. На рисунку 2.5 наведена діаграма класів, яка описує структуру й функціональність рівня бізнес-логіки, демонструючи взаємодію між класами, що реалізують логіку роботи системи.

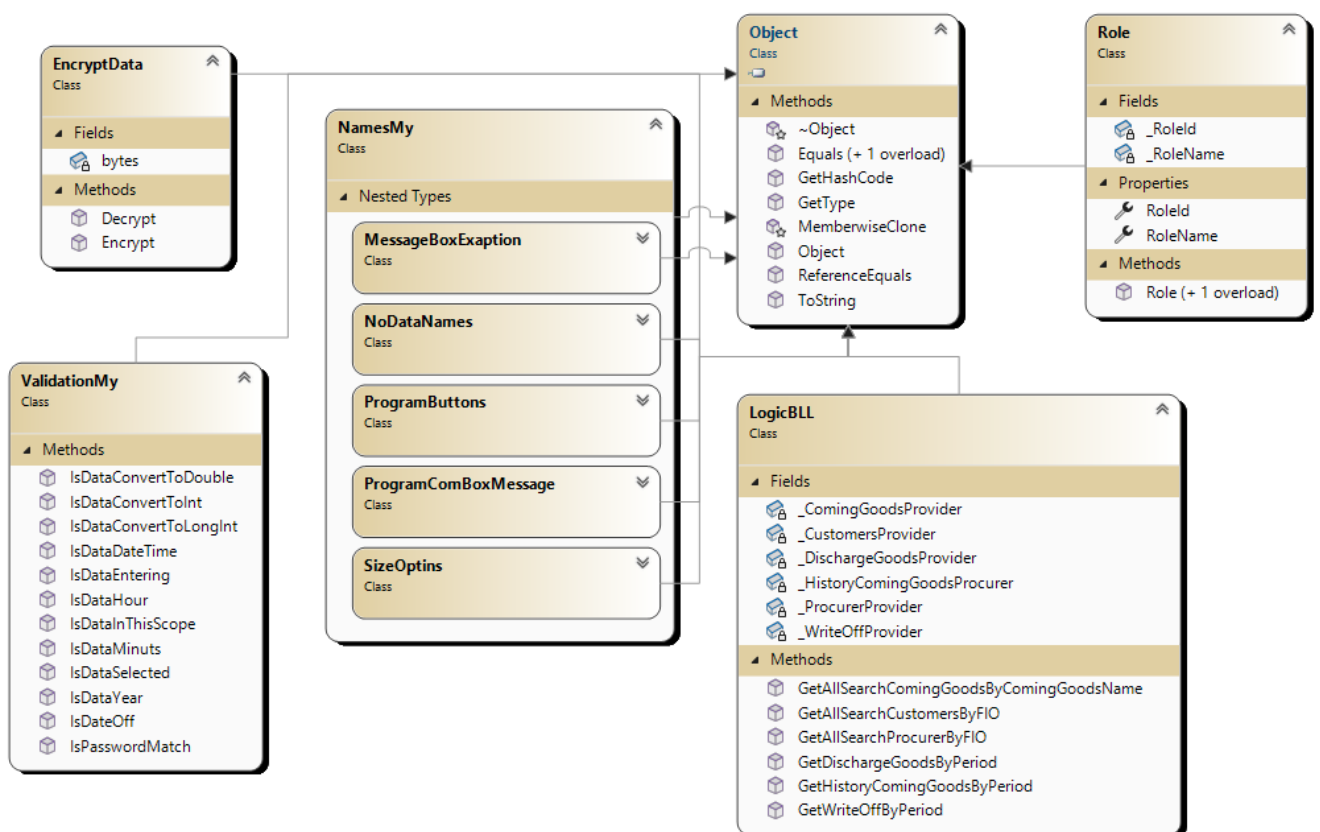


Рис. 2.5. Діаграма класів рівня бізнес-логіки

Даний рівень складається з наступних класів:

- LogicBLL є центральним класом бізнес-логіки, що забезпечує взаємодію з рівнем доступу до даних. Він містить методи для пошуку інформації про товари, постачальників, клієнтів і операції за різні періоди, а також агрегує функціонал, необхідний для обробки даних;
- ValidationMy відповідає за перевірку введених даних. Цей клас реалізує методи для перевірки типу, формату й відповідності даних встановленим критеріям, забезпечуючи точність і коректність інформації в системі;
- EncryptData забезпечує функціонал для шифрування та розшифрування даних. Це дозволяє гарантувати безпеку конфіденційної інформації, такої як паролі чи інші критично важливі дані;
- Role відповідає за управління ролями користувачів у системі. Він містить дані про назву ролі та методи для визначення її атрибутів, що сприяє реалізації контролю доступу до функціональності системи;
- NamesMy включає вкладені типи, що використовуються для налаштувань і обробки даних у системі. Серед вкладених типів виділяються класи для повідомлень, кнопок програм та інших параметрів, що забезпечують зручність налаштування.

Ця структура рівня бізнес-логіки дозволяє об'єднати всі ключові процеси системи, забезпечуючи їх узгодженість, захищеність і відповідність вимогам до обробки даних.

### **2.3. Розробка бази даних для обліку товарів**

База даних є основним компонентом системи обліку товарів, яка забезпечує централізоване зберігання та управління всіма даними, необхідними для функціонування системи. Для забезпечення ефективності й надійності роботи системи була розроблена реляційна база даних, модель якої наведено на рис 2.6.

Кожна таблиця бази даних відповідає за зберігання певного типу даних, що дозволяє розділяти функціональність, спрощуючи управління даними та забезпечуючи їх цілісність. Усі таблиці мають чітко визначені зв'язки між собою,

що дозволяє виконувати комплексні запити для формування звітності та забезпечення підтримки бізнес-процесів.

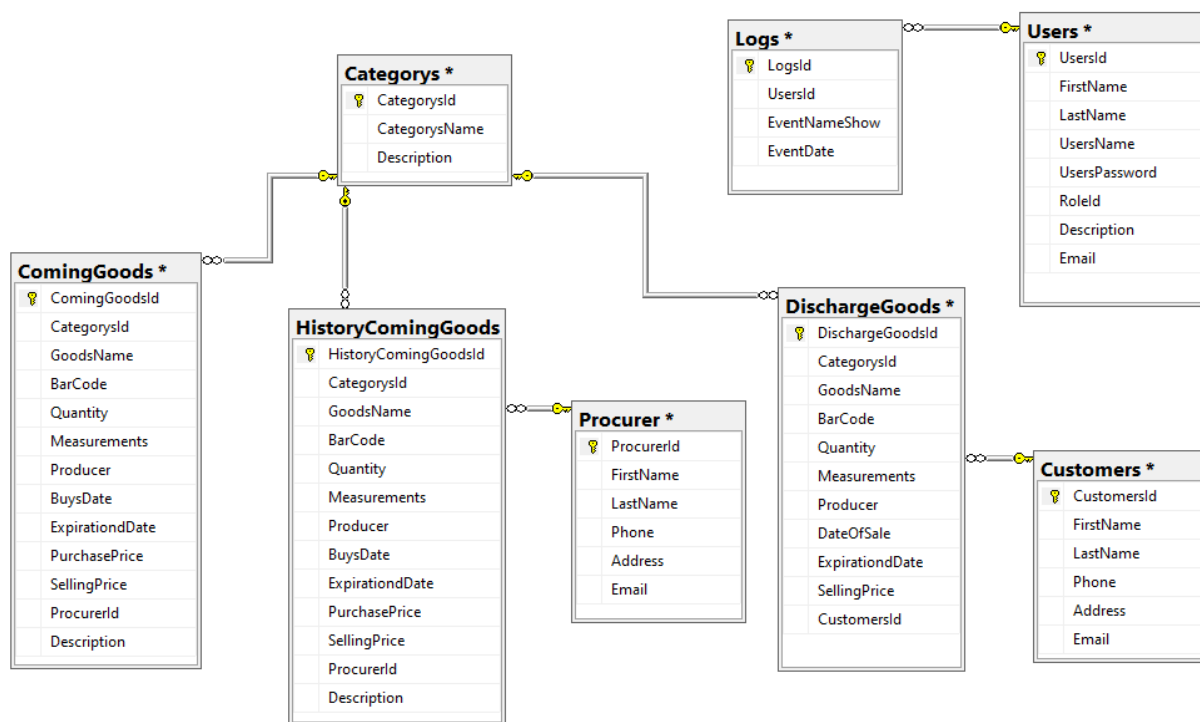


Рис. 2.6. Фізична модель бази даних

Таблиця «ComingGoods» призначена для зберігання даних про товари, що надходять на склад, і забезпечує облік їх характеристик, кількості, ціни та інших параметрів (табл. 2.2).

Таблиця 2.2

### Структура таблиці «ComingGoods»

Назва поля	Тип даних	ПК	ЗК	Опис поля
ComingGoodsId	INT	+	-	Ідентифікатор товару, первинний ключ
CategoriesId	INT	-	+	Ідентифікатор категорії, зв'язок із таблицею «Categories»
GoodsName	NVARCHAR (200)	-	-	Назва товару
BarCode	NVARCHAR (200)	-	-	Штрих-код товару
Quantity	INT	-	-	Кількість одиниць товару
Measurements	NVARCHAR (200)	-	-	Одиниця вимірювання
Producer	NVARCHAR (200)	-	-	Виробник товару
BuysDate	DATETIME	-	-	Дата закупівлі

ExpirationDate	DATETIME	-	-	Термін придатності
PurchasePrice	FLOAT (53)	-	-	Закупівельна ціна
SellingPrice	FLOAT (53)	-	-	Роздрібна ціна
ProcurerId	INT	-	+	Ідентифікатор постачальника
Description	NVARCHAR (2000)	-	-	Опис товару

Таблиця «Categorys» відповідає за зберігання інформації про категорії товарів (табл. 2.3). У цій таблиці зберігається унікальний ідентифікатор категорії, її назва та опис, що дозволяє забезпечити структурування даних про категорії та полегшує пошук необхідної інформації.

Таблиця 2.3

## Структура таблиці «Categorys»

Назва поля	Тип даних	ПК	ЗК	Опис поля
CategorysId	INT	+	-	Ідентифікатор категорії, первинний ключ
CategorysName	NVARCHAR (200)	-	-	Назва категорії, що дозволяє визначити тип продукції
Description	NVARCHAR (2000)	-	-	Опис категорії для деталізації її характеристик

Таблиця «HistoryComingGoods» використовується для зберігання історичних даних про надходження товарів (табл. 2.4). У ній фіксуються всі операції з поставок товарів, включаючи деталі про назву товару, кількість, одиниці вимірювання, постачальника, дати закупівлі й терміни придатності.

Таблиця 2.4

## Структура таблиці «HistoryComingGoods»

Назва поля	Тип даних	ПК	ЗК	Опис поля
HistoryComingId	INT	+	-	Ідентифікатор запису історії надходження, первинний ключ

Продовження табл. 2.4

CategorysId	INT	-	+	Ідентифікатор категорії товару, зв'язок із таблицею «Categorys»
GoodsName	NVARCHAR (200)	-	-	Назва товару
BarCode	NVARCHAR (200)	-	-	Штрих-код товару
Quantity	INT	-	-	Кількість одиниць товару
Measurements	NVARCHAR (200)	-	-	Одиниця вимірювання
Producer	NVARCHAR (200)	-	-	Виробник товару
BuysDate	DATETIME	-	-	Дата закупівлі
ExpirationdDate	NVARCHAR (200)	-	-	Термін придатності
PurchasePrice	FLOAT (53)	-	-	Закупівельна ціна
SellingPrice	FLOAT (53)	-	-	Роздрібна ціна
ProcurerId	INT	-	+	Ідентифікатор постачальника, зв'язок із таблицею «Procurers»
Description	NVARCHAR (200)	-	-	Опис товару

Таблиця «Customers» використовується для зберігання даних про клієнтів системи (табл. 2.5). У ній фіксується унікальний ідентифікатор клієнта, його особисті дані, контактна інформація та адреса. Це дозволяє вести облік клієнтської бази, що є важливим для аналізу продажів, управління замовленнями та підтримання зв'язків із клієнтами.

Таблиця 2.5

Структура таблиці «Customers»

Назва поля	Тип даних	ПК	ЗК	Опис поля
CustomersId	INT	+	-	Ідентифікатор клієнта, первинний ключ
FirstName	NVARCHAR (50)	-	-	Ім'я клієнта
LastName	NVARCHAR (50)	-	-	Прізвище клієнта
Phone	NVARCHAR (20)	-	-	Номер телефону
Address	NVARCHAR (2000)	-	-	Адреса клієнта
Email	NVARCHAR (150)	-	-	Адреса електронної пошти

Таблиця «Procurer» використовується для зберігання даних про постачальників, які співпрацюють із системою (табл. 2.6). У ній зберігаються такі відомості, як ідентифікатор постачальника, його ім'я, прізвище, контактні дані, адреса та електронна пошта. Ця інформація дозволяє підтримувати облік і контроль взаємодії з постачальниками, забезпечуючи управління закупівлями.

Таблиця 2.6

#### Структура таблиці «Procurer»

Назва поля	Тип даних	ПК	ЗК	Опис поля
ProcurerId	INT	+	-	Ідентифікатор постачальника
FirstName	NVARCHAR (50)	-	-	Ім'я постачальника
LastName	NVARCHAR (50)	-	-	Прізвище постачальника
Phone	NVARCHAR (20)	-	-	Номер телефону постачальника
Address	NVARCHAR (2000)	-	-	Адреса постачальника
Email	NVARCHAR (150)	-	-	Електронна пошта постачальника

Таблиця «Users» призначена для зберігання даних про користувачів системи (табл. 2.7). У ній фіксуються унікальний ідентифікатор користувача, його ім'я, прізвище, логін, пароль, роль у системі, опис, а також електронна пошта. Ця таблиця забезпечує управління доступом до системи, враховуючи різні ролі користувачів, і дозволяє ідентифікувати кожного користувача.

Таблиця 2.7

#### Структура таблиці «Users»

Назва поля	Тип даних	ПК	ЗК	Опис поля
UserId	INT	+	-	Ідентифікатор користувача
FirstName	NVARCHAR (50)	-	-	Ім'я користувача
LastName	NVARCHAR (50)	-	-	Прізвище користувача
UserName	NVARCHAR (50)	-	-	Логін користувача
UsersPassword	NVARCHAR (50)	-	-	Пароль користувача
RoleId	INT	-	+	Ідентифікатор ролі
Description	NVARCHAR (1000)	-	-	Опис користувача або його ролі
Email	NVARCHAR (150)	-	-	Електронна пошта користувача

Таблиця «Logs» призначена для реєстрації подій у системі (табл. 2.8). У ній зберігаються дані про активність користувачів, включаючи ідентифікатор користувача, назву події та дату її виконання. Це дозволяє контролювати роботу системи, відслідковувати дії користувачів і забезпечувати безпеку даних.

Таблиця 2.8

Структура таблиці «Logs»

Назва поля	Тип даних	ПК	ЗК	Опис поля
LogsId	INT	+	-	Ідентифікатор запису журналу, первинний ключ
UsersId	INT	-	+	Ідентифікатор користувача, зв'язок із таблицею «Users»
EventNameShow	NVARCHAR (MAX)	-	-	Назва події, що відображає виконану операцію
EventDate	DATETIME	-	-	Дата та час події

Таблиця «DischargeGoods» призначена для зберігання даних про реалізовані товари (додаток 4). У ній фіксуються всі операції з продажу товарів, включаючи унікальний ідентифікатор запису, назву товару, штрих-код, кількість, дату продажу, роздрібну ціну, а також ідентифікатори категорій товарів і клієнтів.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОБЛІКУ ОПТОВОЇ ЗАКУПІВЛІ ТОВАРІВ ТА ЇХ РОЗДРІБНОГО ЗБУТУ

#### 3.1. Інтерфейс користувача комп'ютерної системи обліку товарів

Інтерфейс користувача забезпечує зручну взаємодію користувачів із системою. При його розробці основна увага була зосереджена на простоті, інтуїтивності та функціональності, що дозволяє зменшити час навчання користувачів і підвищити продуктивність роботи. Однією з таких форм є форма «Прихід», яка була розроблена для внесення даних про надходження товарів на склад (рис. 3.1). Ця форма містить поля для введення інформації про категорію товару, його назву, штрих-код, кількість, одиницю вимірювання, виробника, дату постачання, термін придатності, ціни та постачальника. Додатково передбачене поле для опису товару, що дозволяє внести додаткову інформацію за потреби. Форма також оснащена кнопками для додавання запису, очищення даних і виходу з форми, забезпечуючи зручність у роботі.

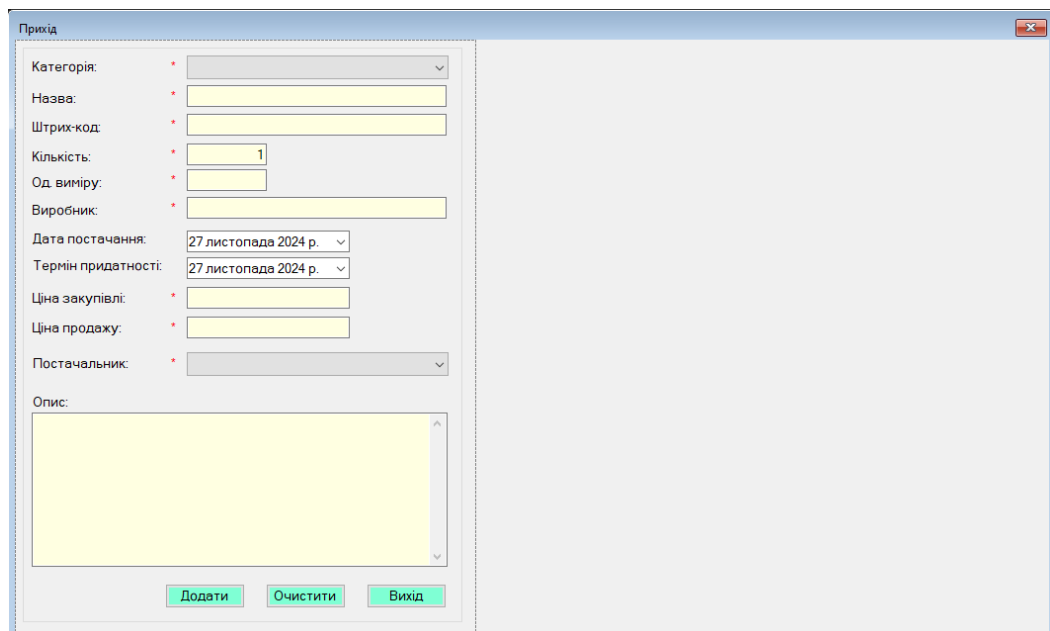


Рис. 3.1. Розробка форми обліку приходу продукції

Метод, код якого зображено на рис. 3.2 прив'язаний до події натискання кнопки, відповідає за додавання нового запису про надходження товару в базу

даних. Спочатку перевіряється правильність введених даних за допомогою методу `IsDataEnteringCorrect`.

```
1 reference
private void AddBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        _ComingGoodsProvider.InsertComingGoods(Convert.ToInt32(CategorysCBox.SelectedValue),
            GoodsNameTBox.Text, BarCodeTBox.Text,
            Convert.ToInt32(QuantityTBox.Text), MeasurementsTBox.Text,
            ProducerTBox.Text, BuysDateDTP.Value, ExpirationDateDTP.Value,
            Convert.ToDouble(PurchasePriceTBox.Text), Convert.ToDouble(SellingPriceTBox.Text),
            Convert.ToInt32(ProcurerCBox.SelectedValue), DescriptionTBox.Text);

        DataLoad();
        ClearAllControls();
    }
}
```

Рис. 3.2. Додавання інформації про надходження товару

Якщо всі дані коректні, викликається метод `InsertComingGoods` з об'єкта `_ComingGoodsProvider`, який додає новий запис у базу даних, використовуючи значення, отримані з полів форми. Дані включають категорію товару, назву, штрих-код, кількість, одиницю вимірювання, виробника, дату постачання, термін придатності, закупівельну й роздрібну ціни, ідентифікатор постачальника та опис. Після успішного додавання даних викликається метод `DataLoad` для оновлення інформації на формі, а також `ClearAllControls` для очищення полів вводу, щоб підготувати форму до внесення нового запису.

На рис. 3.3 зображено метод, який призначений для додавання нового запису про надходження товарів у таблицю бази даних `ComingGoods`. У ньому формується SQL-запит, який виконує вставку даних у відповідні поля таблиці. Запит містить параметри, які пізніше будуть заповнені переданими аргументами методу, такими як ідентифікатор категорії, назва товару, штрих-код, кількість, одиниця вимірювання, виробник, дата закупівлі, термін придатності, ціни закупівлі та продажу, ідентифікатор постачальника та опис.

Далі відкривається з'єднання з базою даних за допомогою `SqlConnection`, у яке передається підключення, збережене в полі `_ConnString`. Через об'єкт `SqlCommand` переданий SQL-запит пов'язується із цим з'єднанням. Потім кожен параметр запиту заповнюється значеннями, що надаються під час виклику методу.

Після заповнення всіх параметрів виконується команда `ExecuteNonQuery()`, яка вносить дані в базу, після чого з'єднання автоматично закривається завдяки використанню конструкції `using`. Такий підхід забезпечує безпечне та ефективне виконання операцій із базою даних, зменшуючи ризик витоку ресурсів або виникнення помилок.

```

1 reference
public void InsertComingGoods(int CategoriesId, string GoodsName, string BarCode, int Quantity, string Measurements, string Producer,
DateTime BuysDate, DateTime ExpirationndDate, double PurchasePrice, double SellingPrice, int ProcurerId, string Description) {

    string SqlString = "INSERT INTO ComingGoods (CategoriesId, GoodsName, BarCode, Quantity, Measurements, Producer, " +
        "BuysDate, ExpirationndDate, PurchasePrice, SellingPrice, ProcurerId, Description" +
        ") Values(@CategoriesId, @GoodsName, @BarCode, @Quantity, @Measurements," +
        " @Producer, @BuysDate, @ExpirationndDate, @PurchasePrice, @SellingPrice, @ProcurerId, @Description)";

    using (SqlConnection conn = new SqlConnection(_ConnString)) {
        using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@CategoriesId", CategoriesId);
            cmd.Parameters.AddWithValue("@GoodsName", GoodsName);
            cmd.Parameters.AddWithValue("@BarCode", BarCode);
            cmd.Parameters.AddWithValue("@Quantity", Quantity);
            cmd.Parameters.AddWithValue("@Measurements", Measurements);
            cmd.Parameters.AddWithValue("@Producer", Producer);
            cmd.Parameters.AddWithValue("@BuysDate", BuysDate);
            cmd.Parameters.AddWithValue("@ExpirationndDate", ExpirationndDate);
            cmd.Parameters.AddWithValue("@PurchasePrice", PurchasePrice);
            cmd.Parameters.AddWithValue("@SellingPrice", SellingPrice);
            cmd.Parameters.AddWithValue("@ProcurerId", ProcurerId);
            cmd.Parameters.AddWithValue("@Description", Description);
            conn.Open();
            cmd.ExecuteNonQuery();
        }
    }
}

```

Рис. 3.3. Код методу «InsertComingGoods»

Метод на рис. 3.4 відповідає за обробку події кліку на клітинку в таблиці `HistoryComingGoodsGridView`.

```

private void HistoryComingGoodsGridView_CellClick(object sender, DataGridViewCellEventArgs e) {
    if (e.RowIndex >= 0 &&
        HistoryComingGoodsGridView[0, e.RowIndex].Value.ToString() != _HistoryComingGoodsList[0].Message) {
        _selectedRowIndex = e.RowIndex;
        UpdateHistoryComingGoodsForm updateHistoryComingGoodsForm =
            new UpdateHistoryComingGoodsForm(Convert.ToInt32(HistoryComingGoodsGridView[0, e.RowIndex].Value.ToString()));
        updateHistoryComingGoodsForm.ShowDialog();
        DataLoad();
    }
}

```

Рис. 3.4. Код методу «InsertComingGoods»

На початку перевіряється, чи обрана рядкова індексація є валідною (тобто чи рядок має індекс, що дорівнює або перевищує нуль) і чи значення в першій колонці обраного рядка не збігається з попередньо визначеним повідомленням у списку `_HistoryComingGoodsList`. У разі задоволення цих умов у змінну `_selectedRowIndex` записується індекс обраного рядка.

Після цього створюється новий екземпляр форми UpdateHistoryComingGoodsForm, у яку передається ідентифікатор запису, отриманий із першої комірки обраного рядка. Виклик методу ShowDialog() дозволяє відкрити нову форму для оновлення даних. Завершується робота методу викликом DataLoad для оновлення даних у таблиці після завершення операцій із внесення змін. Цей підхід забезпечує інтерактивність і зручність редагування записів безпосередньо з інтерфейсу користувача.

Метод «SaveBtn\_Click» виконує функцію збереження змін у записі про товар, викликаний натисканням кнопки «Зберегти» (рис. 3.5). Спочатку перевіряється коректність введених користувачем даних за допомогою методу IsDataEnteringCorrect. Якщо дані відповідають усім вимогам, викликається метод UpdateComingGoods об'єкта \_ComingGoodsProvider, який оновлює існуючий запис у базі даних.

```
private void SaveBtn_Click(object sender, EventArgs e) {  
    if (IsDataEnteringCorrect()) {  
        _ComingGoodsProvider.UpdateComingGoods(Convert.ToInt32(CategorysCBox.SelectedValue),  
        GoodsNameTBox.Text, BarCodeTBox.Text,  
        Convert.ToInt32(QuantityTBox.Text), MeasurementsTBox.Text,  
        ProducerTBox.Text, BuysDateDTP.Value, ExpirationDateDTP.Value,  
        Convert.ToDouble(PurchasePriceTBox.Text), Convert.ToDouble(SellingPriceTBox.Text),  
        Convert.ToInt32(ProcurerCBox.SelectedValue),  
        DescriptionTBox.Text, _ComingGoodsId);  
        this.Close();  
    }  
}
```

Рис. 3.5. Код методу «SaveBtn\_Click»

Для оновлення використовуються значення, отримані з елементів управління форми, включаючи вибрану категорію, назву товару, штрих-код, кількість, одиницю вимірювання, виробника, дати постачання й терміну придатності, а також закупівельну й роздрібну ціни, ідентифікатор постачальника та опис. Разом із цими даними передається унікальний ідентифікатор товару \_ComingGoodsId, який дозволяє знайти потрібний запис у базі.

Після успішного виконання оновлення форма закривається за допомогою методу this.Close(), сигналізуючи про завершення процесу редагування. Такий підхід забезпечує зручність і надійність під час оновлення даних у базі через інтерфейс користувача.

Для забезпечення зручного обліку постачальників у системі була розроблена спеціалізована форма, яка дозволяє користувачам легко додавати, редагувати та переглядати інформацію про постачальників. Її дизайн орієнтований на інтуїтивно зрозумілу взаємодію, що спрощує введення даних та управління базою постачальників. Форма включає необхідні поля для заповнення ключових параметрів, таких як ім'я, прізвище, контактні дані, електронна пошта та адреса.

Форма обліку постачальників, представлена на рис. 3.6, містить усі необхідні елементи для введення інформації про постачальників. Вона має текстові поля для введення прізвища, імені, номера телефону, електронної пошти та адреси постачальника.

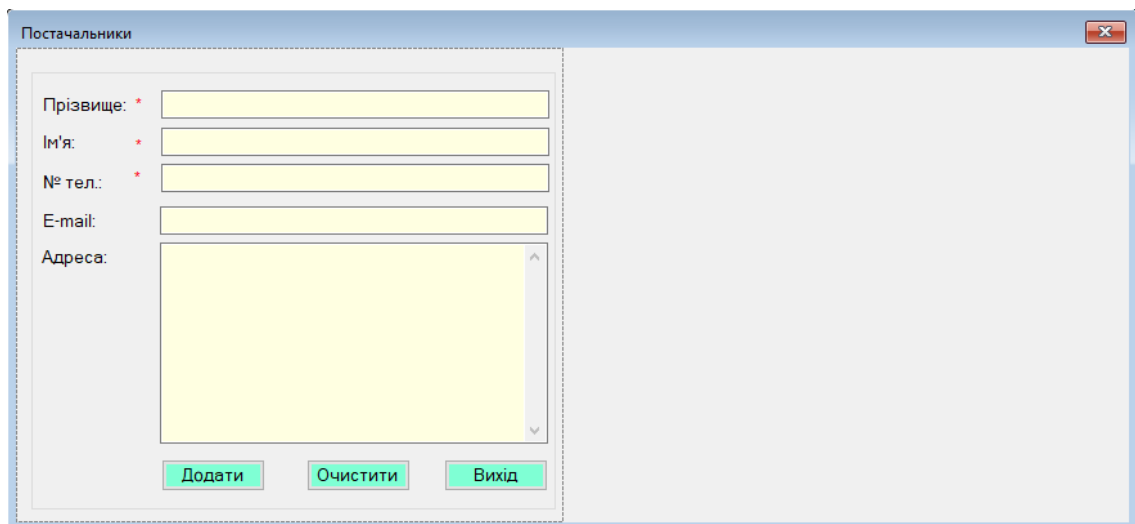
The image shows a screenshot of a Windows-style application window titled "Постачальники" (Suppliers). The window contains a form with several input fields: "Прізвище:" (Surname), "Ім'я:" (Name), "№ тел.:" (Phone number), "E-mail:", and "Адреса:" (Address). Each field is highlighted in yellow. Below the fields are three buttons: "Додати" (Add), "Очистити" (Clear), and "Вихід" (Exit). The window has a standard Windows title bar with a close button in the top right corner.

Рис. 3.6. Розробка форми обліку постачальників підприємства

У нижній частині форми розташовані кнопки для додавання нового запису, очищення введених даних і виходу з форми. Такий підхід забезпечує простоту використання та мінімізує помилки при введенні даних, що сприяє підвищенню ефективності обліку.

Метод на рис. 3.7 виконує функцію оновлення даних у таблиці постачальників `ProcurerGridView` та забезпечує коректне відображення й взаємодію з даними після змін. На початку зберігається індекс першого видимого рядка таблиці, якщо його значення більше нуля, щоб після оновлення повернути користувача до тієї ж позиції. Потім викликається метод отримання всіх записів

про постачальників із бази даних через об'єкт `_ProcurerProvider`, і отриманий список передається для відображення у таблиці через функцію `LoadDataInProcurerGridView`.

```
private void DataLoad() {
    int firstRowIndex = 0;
    if (ProcurerGridView.FirstDisplayedScrollingRowIndex > 0) {
        firstRowIndex = ProcurerGridView.FirstDisplayedScrollingRowIndex;
    }
    try {
        _ProcurerList = _ProcurerProvider.GetAllProcurer();
        LoadDataInProcurerGridView(_ProcurerList);
        if (_selectedRowIndex == ProcurerGridView.Rows.Count) {
            _selectedRowIndex = ProcurerGridView.Rows.Count - 1;
        }
        if (_selectedRowIndex >= 0) {
            ProcurerGridView.FirstDisplayedScrollingRowIndex = firstRowIndex;
            ProcurerGridView.Rows[_selectedRowIndex].Selected = true;
        }
    } catch { }
}
```

Рис. 3.7. Завантаження даних постачальників

Після завантаження даних перевіряється, чи обраний рядок індексу `_selectedRowIndex` не виходить за межі кількості рядків у таблиці. Якщо це так, коректується індекс, щоб він відповідав останньому доступному рядку. Якщо вибраний рядок валідний, таблиця оновлює свій вид, зберігаючи попереднє положення прокручування, і встановлює вибір на обраному рядку. Такий підхід гарантує зручність роботи користувача, зберігаючи його поточний контекст навіть після оновлення даних.

### 3.2. Алгоритми обробки даних у системі обліку товарів

Алгоритми обробки даних відіграють ключову роль у забезпеченні коректності, безпеки та швидкості функціонування системи обліку товарів. Вони охоплюють процеси перевірки введених даних, їх передачі до бази даних, аналізу отриманих результатів та виведення відповідних повідомлень для користувачів. Одним із таких алгоритмів є процес авторизації, який забезпечує контроль доступу до системи та дозволяє працювати лише зареєстрованим користувачам.

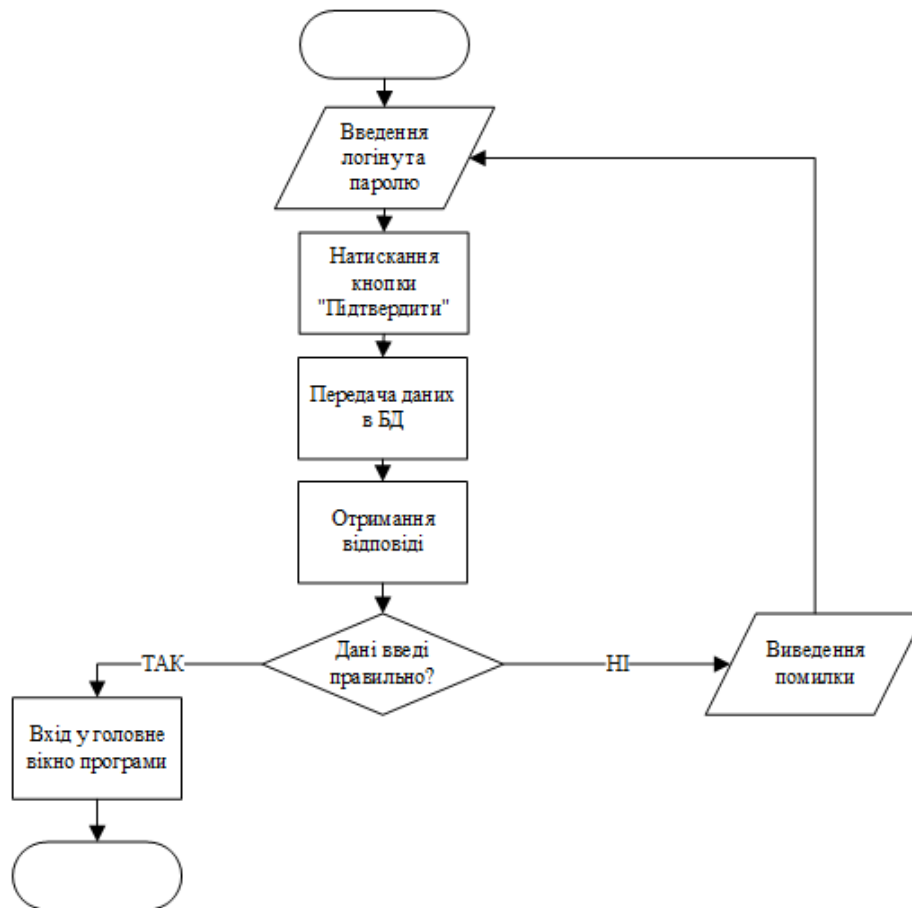


Рис. 3.8. Схема алгоритму проведення автоматизації користувача

Процес починається з введення логіну та паролю, після чого користувач натискає кнопку підтвердження. Дані передаються в базу для перевірки їх коректності, і на основі відповіді система вирішує, чи дозволити вхід у головне вікно програми, чи вивести повідомлення про помилку введення. Алгоритм забезпечує безпечний та інтуїтивний процес авторизації, мінімізуючи ризик несанкціонованого доступу.

На рис. 3.9 зображено алгоритм процесу додавання інформації про надходження нового товару в систему починається з введення користувачем усіх необхідних даних про товар, таких як назва, категорія, штрих-код, кількість, одиниці вимірювання, виробник, дата постачання, ціна закупівлі та продажу. Після введення даних користувач натискає кнопку «Додати», яка ініціює перевірку коректності введеної інформації. Алгоритм перевіряє всі поля на наявність помилок чи порожніх значень, і якщо виявляються невідповідності,

користувач отримує повідомлення про помилку та може виправити введені дані.

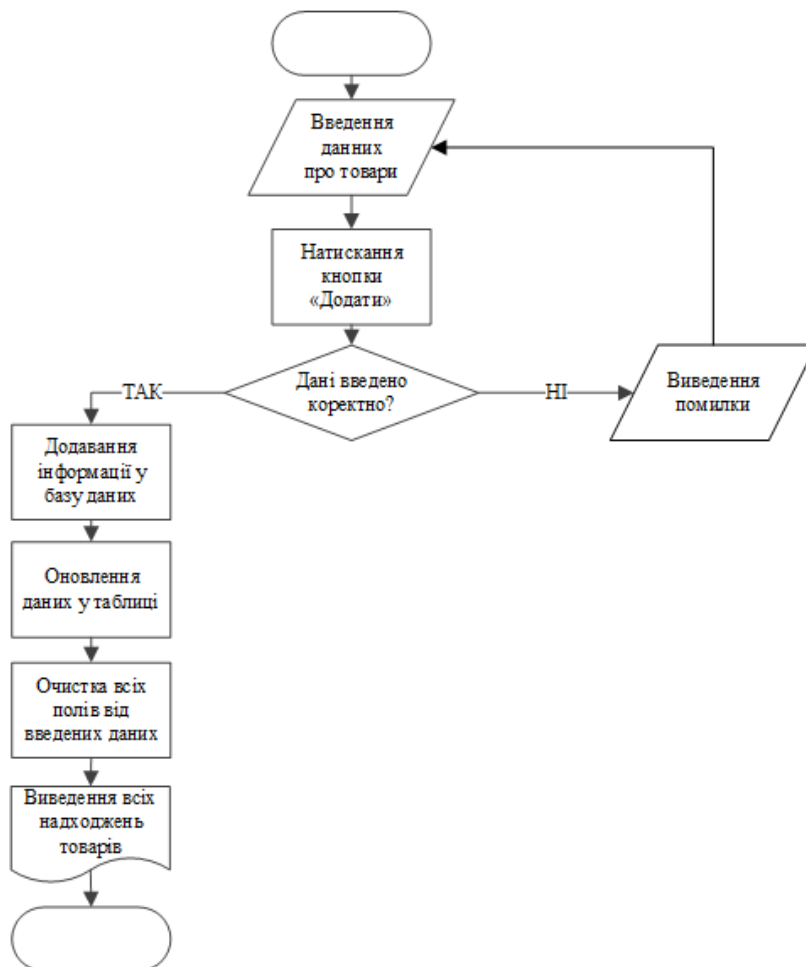


Рис. 3.9. Схема алгоритму додавання приходу нового товару

Якщо дані введено коректно, система додає запис до бази даних, де інформація про надходження товару фіксується для подальшого обліку та аналізу. Після цього таблиця на інтерфейсі оновлюється, щоб користувач міг одразу побачити внесений запис і впевнитися у його правильності. Для зручності роботи всі поля форми очищаються автоматично після успішного додавання запису, щоб користувач міг швидко ввести нові дані без додаткових маніпуляцій.

Наприкінці алгоритму формується загальна інформація про всі надходження товарів, яка доступна для перегляду користувачем. Такий підхід забезпечує зручність і швидкість роботи з системою, мінімізуючи кількість помилок і спрощуючи процес управління складськими запасами.

Алгоритм пошуку інформації про товар у системі починається з вибору користувачем пункту меню «Товари», який відкриває відповідний інтерфейс для

введення критеріїв пошуку (рис. 3.10). Користувач вводить параметри, за якими система має здійснити пошук, наприклад, назву товару, штрих-код, категорію або інші характеристики. Після цього система перевіряє правильність введених даних, щоб уникнути помилок або некоректних запитів.

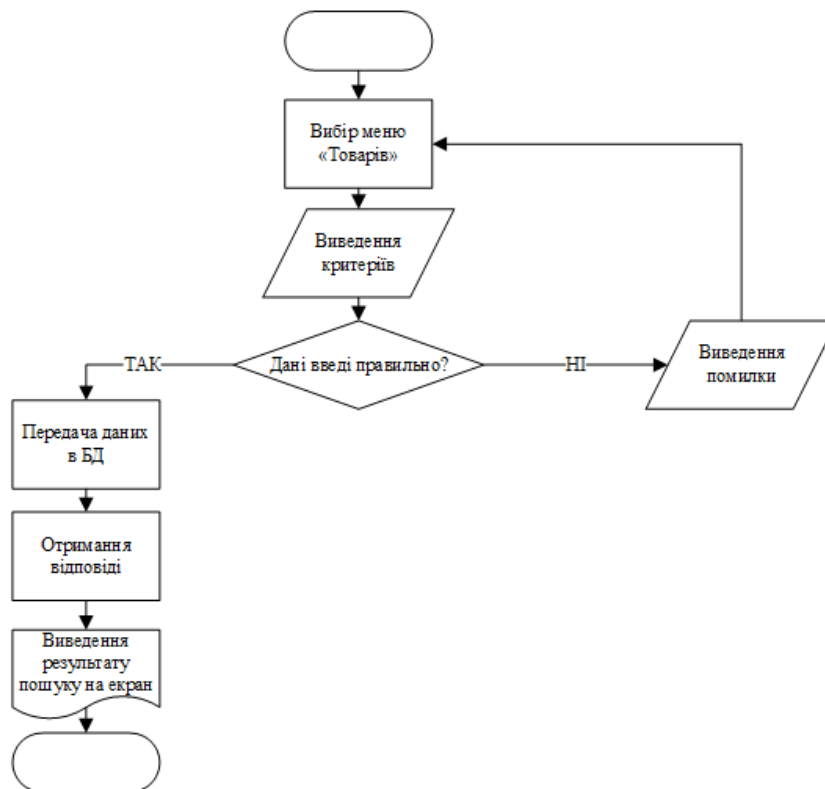


Рис. 3.10. Схема алгоритму пошуку інформації про товар

У разі виявлення помилок система інформує користувача про некоректність введення за допомогою відповідного повідомлення, що дозволяє виправити помилки й повторити процес. Якщо дані введені правильно, вони передаються до бази даних, де здійснюється обробка запиту. Результатом цього етапу є отримання відповіді від бази даних із потрібною інформацією про товар.

### 3.3. Тестування та впровадження системи обліку товарів

Процес тестування є важливим етапом розробки системи обліку товарів, який забезпечує перевірку її працездатності, коректності функціонування та відповідності заданим вимогам. Метою тестування є виявлення потенційних помилок у роботі системи та їх усунення ще до впровадження. Особлива увага

приділяється критично важливим функціям, таким як облік надходжень, реалізації товарів та інтеграція з базою даних, щоб гарантувати їх стабільність і надійність.

Після завершення розробки було проведено повне тестування системи, результати якого наведені на рис. 3.11.

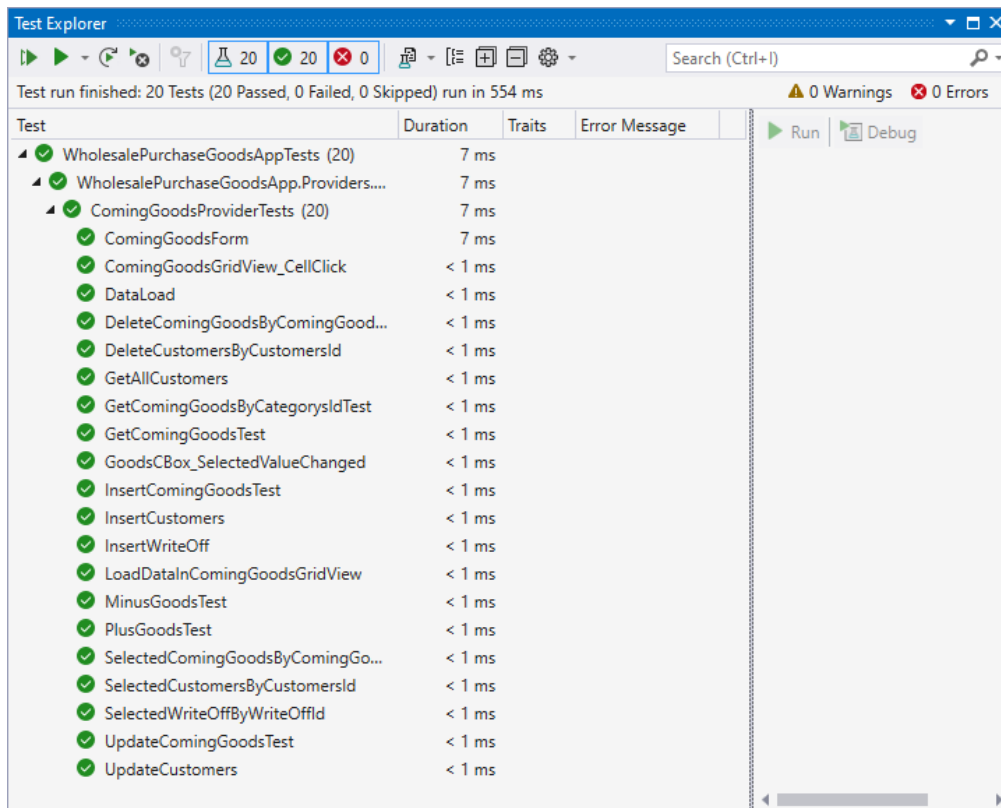


Рис. 3.11. Результати проведеного тестування

Для перевірки використовувався інструмент Test Explorer, який включає 20 тестів для різних функціональних компонентів системи. Усі тести завершилися успішно, не було виявлено жодних помилок чи попереджень, що свідчить про правильність реалізації функціональності та готовність системи до впровадження.

Форма авторизації є важливим компонентом системи, що забезпечує доступ користувачів до її функціоналу на основі перевірки введених облікових даних. Вона реалізує механізм контролю доступу, дозволяючи працювати з системою лише зареєстрованим користувачам. На формі, представленій на рис. 3.12, користувачеві пропонується ввести свій логін і пароль у відповідні поля. Для підтвердження введених даних передбачена кнопка «Підтвердити», натискання на яку ініціює перевірку облікових даних у базі даних.

Логин \*

Пароль \*

[Реєстрація](#)

Рис. 3.12. Авторизація користувача

Форма «Прихід», що є ключовою складовою системи обліку товарів, поєднує інтерфейс для введення нових даних про надходження та інтерактивну таблицю, яка відображає всі записи про наявні товари (рис. 3.13).

№	Категорія	Назва	Штрих-код	Кількість	Од. виміру	Виробник	Ціна закупівлі	Ціна п
1	Побутова техніка	Холодильник Sam...	1239321893421	91	кг	Луцко, Скоропадс...	87,312675...	
2	Побутова техніка	Пральна машина LG	2635602113605	90	шт	Латанський - Євпак	374,64835...	
3	Побутова техніка	Пилосос Philips	4647213897490	70	кг	Тарасюк - Євенко	983,50938...	
4	Продукти харчува...	Рис Басмат	8051931595839	68	м	Сторожук, Гайдук...	69,249831...	
5	Продукти харчува...	Консервованій го...	7398199541323	47	шт	Зленко, Корнійчук...	513,31428...	
6	Продукти харчува...	Мінеральна вода ...	2348002505344	55	шт	Гаман - Лугова	372,27617...	
7	Одяг та взуття	Куртка зимова	7429075094708	89	кг	Коров'як Збут	324,27262...	
8	Одяг та взуття	Кросівки Adidas	2858550225255	67	л	Карпук - Васильк...	367,24283...	
9	Одяг та взуття	Футболка Nike	8066622290207	44	л	Гарай Трейд	777,19434...	
10	Електроніка	Смартфон iPhone 14	8226340888104	14	кг	Андрухович, Емец...	537,37934...	
11	Електроніка	Ноутбук Dell Inspiron	8366393376873	64	кг	Бандера Пром	703,60213...	
12	Електроніка	Планшет Samsung...	1741063534130	42	шт	Трясун - Скиба	593,07269...	
13	Канцелярія та офі...	Блокнот А5	2000692294693	14	м	Трясун - Кулинич	341,92374...	
14	Канцелярія та офі...	Гелева ручка Parker	2920333496739	59	шт	Павленко - Гарун	452,76559...	
15	Канцелярія та офі...	Принтер HP LaserJet	9061756519902	61	шт	Ганич, Петик and ...	812,82764...	

Рис. 3.13. Надходження товару

Форма «Розхід» у системі обліку товарів дозволяє здійснювати облік реалізації продукції, інтегруючи функції введення даних про продаж та перегляду вже виконаних операцій (рис. 3.14). У лівій частині форми розміщено поля для вибору категорії товару, назви, штрих-коду, кількості, одиниці вимірювання, виробника, ціни продажу, дати продажу, терміну придатності та покупця. Для зручності користувач бачить підказки, наприклад, кількість доступних одиниць товару на складі, що спрощує контроль залишків. У разі помилкового введення даних можна скористатися кнопкою «Очистити», яка швидко очищує всі поля.

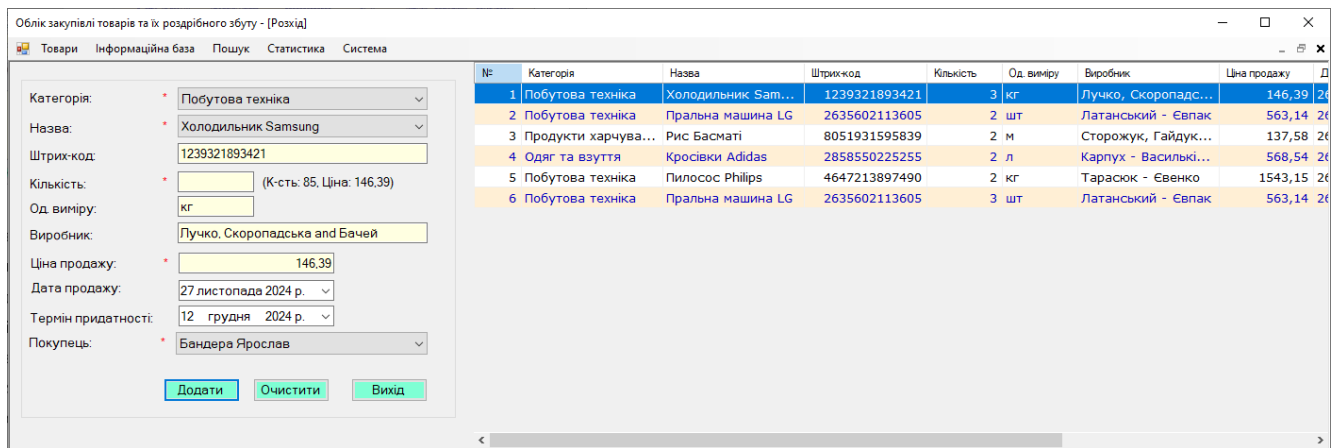


Рис. 3.14. Розхід товару

Форма «Категорії» у системі обліку товарів забезпечує управління списком категорій, які використовуються для класифікації товарів (рис. 3.15). У лівій частині форми розташовані поля для введення назви категорії та її опису. Поле для назви є обов'язковим, про що сигналізує спеціальна позначка, а текстове поле для опису дозволяє деталізувати призначення чи особливості кожної категорії. Для зручності взаємодії передбачені кнопки «Додати», «Очистити» та «Вихід», які спрощують виконання базових операцій.

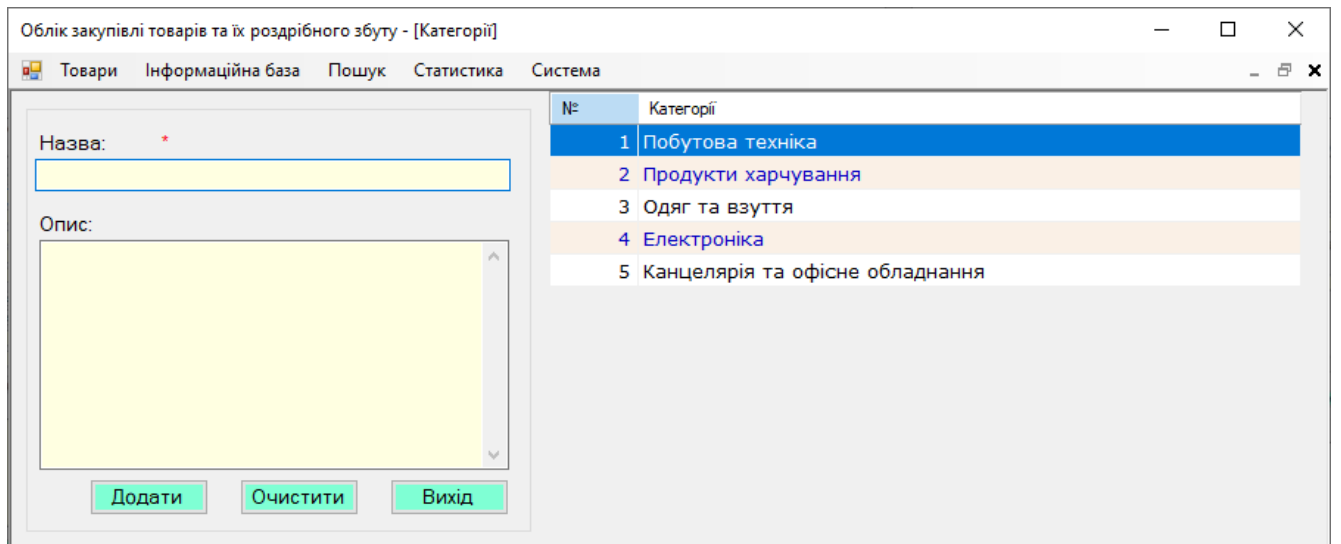


Рис. 3.15. Категорії товару

Форма «Клієнти» у системі обліку товарів призначена для управління інформацією про клієнтів підприємства (рис. 3.16). У лівій частині форми знаходяться поля для введення основних даних про клієнта, таких як прізвище, ім'я, номер телефону, електронна пошта та адреса.

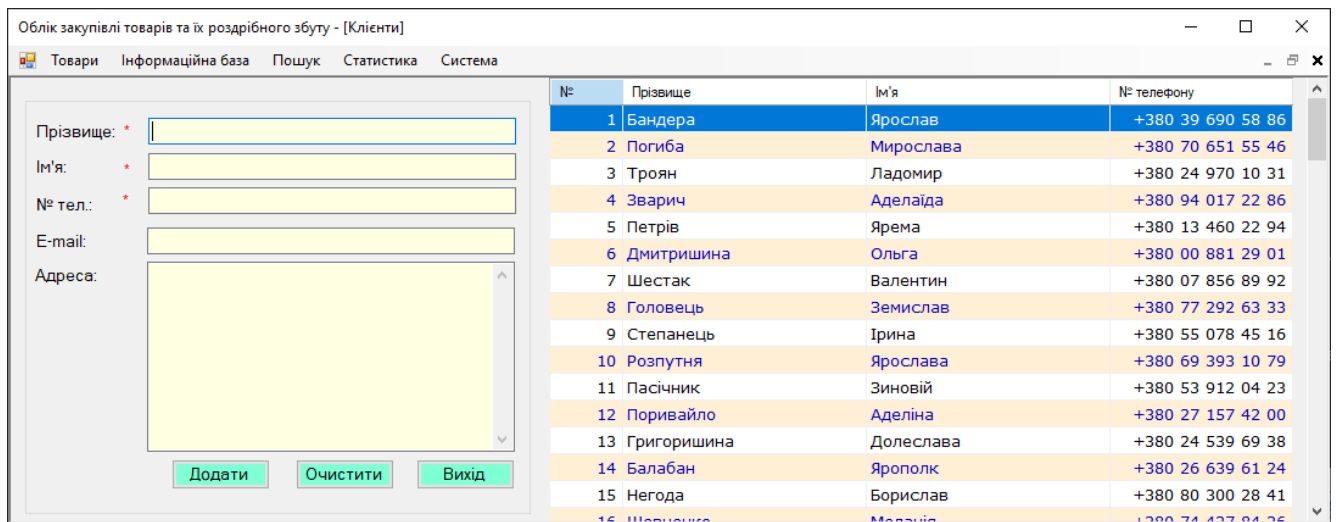


Рис. 3.16. Клієнти підприємства

Форма «Постачальники» призначена для ведення бази даних про постачальників, з якими співпрацює підприємство (рис. 3.17). Ліва частина форми містить текстові поля для введення основної інформації про постачальника, зокрема прізвища, імені, номера телефону, електронної пошти та адреси. Поля для прізвища, імені та номера телефону позначені як обов'язкові, що сприяє точності й повноті введених даних. Також передбачені кнопки «Додати», «Очистити» та «Вихід», які забезпечують зручність взаємодії з формою.

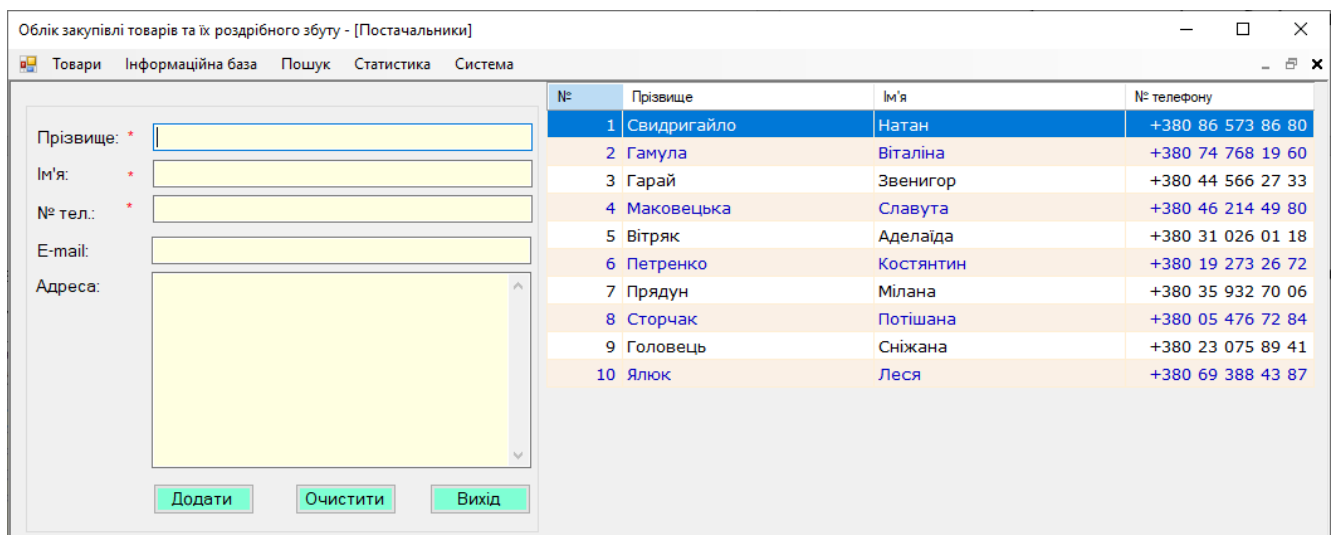


Рис. 3.17. Постачальники товарів

Форма «Пошук по товарах» у системі обліку реалізована для швидкого знаходження інформації про наявні товари на складі (рис. 3.18). Інтерфейс форми включає поле для введення назви товару та кнопку «Пошук», що активує процес

фільтрації даних у базі. Ця функція дозволяє користувачеві оперативно отримати доступ до необхідної інформації, не переглядаючи вручну великі списки записів.

№	Товар	Кількість	Од. виміру	Ціна пр.	Постачальник
1	Холодильник Samsung	85	кг	146,39	Сторчак Потішана

Рис. 3.18. Пошук товарів, які реалізуються на підприємстві

Результати пошуку відображаються у вигляді таблиці в нижній частині форми, де вказано найменування товару, його кількість, одиницю вимірювання, ціну продажу та постачальника.

Форма «По приходу товару» створена для генерації звітів про закупівлі товарів за вказаний період, забезпечуючи користувачам можливість детального аналізу поставок (рис. 3.19). У верхній частині інтерфейсу розміщені поля для вибору початкової та кінцевої дати, які визначають часовий проміжок звіту, а також кнопка «Формувати», що ініціює створення звіту.

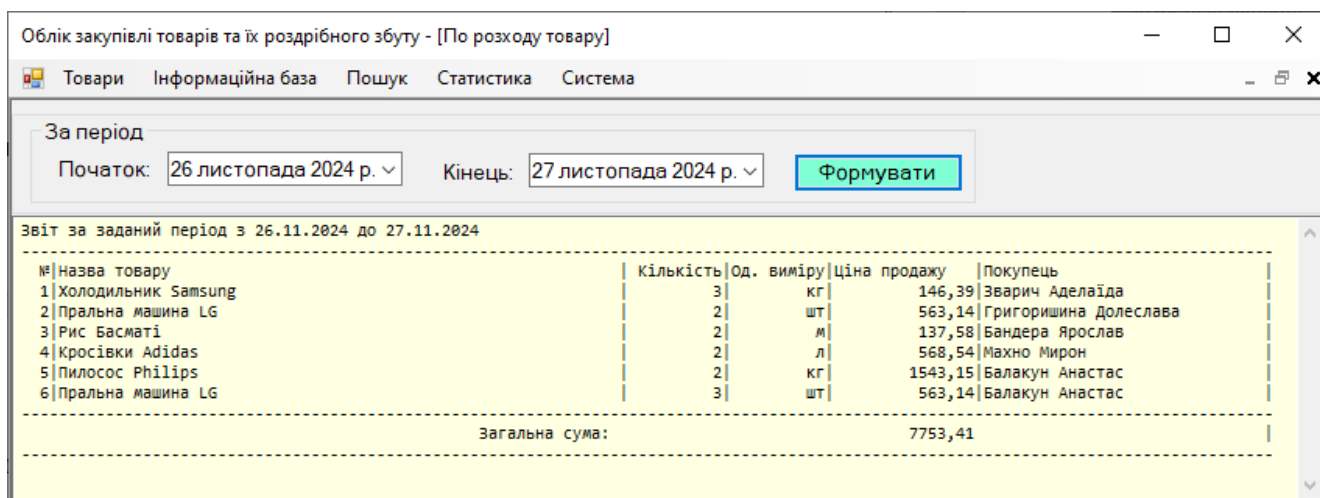
Звіт за заданий період з 01.01.2024 до 27.11.2024

№	назва товару	Кількість	Од. виміру	ціна закупівлі	Постачальник
1	Холодильник Samsung	91	кг	87,31	Сторчак Потішана
2	Пральна машина LG	90	шт	374,65	Гарай Звенигор
4	Рис Басматі	68	м	69,25	Маковецька Славута
5	Консервованний горошок	47	шт	513,31	Ялюк Леся
8	Кросівки Adidas	67	л	367,24	Свидригайло Натан
9	Футболка Nike	44	л	777,19	Гамула Віталіна
10	Смартфон iPhone 14	14	кг	537,38	Свидригайло Натан
11	Ноутбук Dell Inspiron	64	кг	703,60	Гарай Звенигор
12	Планшет Samsung Galaxy Tab	42	шт	593,07	Гамула Віталіна
13	Блокнот A5	14	м	341,92	Вітряк Аделаїда
14	Гелева ручка Parker	59	шт	452,77	Ялюк Леся
Загальна сума:				238263,39	

Рис. 3.19. Звітність по приходу товару

Форма «По розходу товару» дозволяє користувачам отримувати звіти про реалізацію товарів за обраний період, забезпечуючи детальний аналіз продажів. У верхній частині форми розташовані елементи для вибору дати початку та кінця

періоду, що дозволяє адаптувати звітність до конкретних часових рамок. Кнопка «Формувати» активує генерацію звіту, яка відображається в табличному форматі, забезпечуючи зрозуміле й структуроване представлення даних.



№	Назва товару	Кількість	Од. виміру	Ціна продажу	Покупець
1	Холодильник Samsung	3	кг	146,39	Зварич Аделаїда
2	Пральна машина LG	2	шт	563,14	Григорішина Долеслава
3	Рис Басматі	2	л	137,58	Бандера Ярослав
4	Кросівки Adidas	2	л	568,54	Махно Мирон
5	Пилосос Philips	2	кг	1543,15	Балакун Анастас
6	Пральна машина LG	3	шт	563,14	Балакун Анастас
Загальна сума:				7753,41	

Рис. 3.20. Звітність по реалізації товару

У таблиці звіту вказано назву товару, кількість реалізованих одиниць, одиницю вимірювання, ціну продажу та ім'я покупця. Також наприкінці таблиці автоматично розраховується загальна сума продажів за вибраний період, що дозволяє оцінити прибуток за заданий час. Такий інструмент є ефективним для аналізу комерційної діяльності, оскільки надає вичерпну інформацію про кожну транзакцію, а також про загальні результати. Інтуїтивний дизайн форми та автоматизація розрахунків сприяють швидкому доступу до аналітичних даних і підтримці прийняття обґрунтованих рішень щодо продажів.

## ВИСНОВКИ

У рамках виконання кваліфікаційної роботи були проведені теоретичні та практичні дослідження, спрямовані на розробку комп'ютерної системи для обліку оптової закупівлі товарів і їх роздрібного збуту. Метою роботи було створення програмного забезпечення для автоматизації обліку, що дозволяє покращити управління товарними запасами на торговельному підприємстві. У процесі виконання роботи отримані наступні результати:

1. Проведено аналіз теоретичних аспектів обліку товарів у торгівлі, що включав вивчення процесів оптової закупівлі та роздрібного збуту, а також значення автоматизації у цих процесах. Аналіз сучасних програмних рішень, таких як 1С:Підприємство, SAP Business One і TradeSoft, дозволив визначити основні підходи до автоматизації обліку та сформулювати вимоги до системи.

2. Визначено функціональні вимоги та архітектуру системи, яка базується на трьохрівневій моделі. Це дозволило чітко розділити відповідальність між рівнями доступу до даних, бізнес-логіки та користувацького інтерфейсу, забезпечивши зручність експлуатації та масштабованість системи.

3. Розроблено базу даних для обліку товарів, яка охоплює основні сутності, такі як товари, постачальники, клієнти та операції з надходження й реалізації. Побудована фізична модель бази даних дозволила забезпечити її цілісність, узгодженість і ефективність зберігання інформації.

4. Створено інтерфейс користувача, який включає функції для введення, перегляду та редагування даних. Інтерфейс забезпечує зручну взаємодію з системою, включаючи алгоритми авторизації користувачів, облік надходжень і розходу товарів, а також функції пошуку і формування звітів. Це дозволило врахувати всі ключові бізнес-процеси підприємства.

5. Проведено модульне тестування системи, яке підтвердило правильність її роботи та відповідність функціональності заданим вимогам. Система пройшла всі перевірки без помилок, що свідчить про її надійність і готовність до впровадження.

У майбутньому дослідження можуть бути зосереджені на інтеграції розробленої системи з іншими інформаційними продуктами, такими як CRM або ERP-системи, що дозволить забезпечити більш комплексний підхід до управління бізнес-процесами підприємства. Це відкриє нові можливості для автоматизації взаємодії між різними підрозділами компанії та підвищить загальну ефективність роботи. Окрім цього, перспективним напрямком є розширення функціональності системи, зокрема, шляхом впровадження модулів прогнозування залишків товарів, аналітичних інструментів для аналізу продажів та оцінки ефективності процесів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Павлов Р. А. Організаційно-економічний механізм антикризового регулювання банківського сектору національної економіки: теоретичні аспекти та вдосконалення. Схід, 2007. № 2. С. 39-41.
2. Апопій В.В., Міщук І.П., Ребицький В.М., Рудницький С.І. Організація торгівлі: підруч. 3-тє вид. Київ, 2009. 632 с.
3. Кудирко О.М. Удосконалення контролю товарних запасів. Україна. Ефективна економіка. 2018. № 4. 185 с.
4. Любезна І.В. Облік і внутрішньогосподарський контроль матеріальних запасів: дис. канд. екон. наук: 08.00.09 «Бухгалтерський облік, аналіз та аудит»; Тернопільський нац. економ. ун-т. Тернопіль, 2009. 226 с.
5. Небава М.І., Адлер О.О., Лесько О.Й. Економіка та організація виробничої діяльності підприємства. Ч. 1. Економіка підприємства: навчальний посібник. ВНТУ, 2011. 117 с.
6. Тарасова Т.О. Обліково-аналітичне забезпечення управління сталим розвитком підприємств торгівлі: дис. д-ра екон. наук: 08.00.09 «Бухгалтерський облік, аналіз та аудит»; Харківський держ.ун-т харч.та торгівлі. Харків, 2017. 492 с.
7. Рожелюк, В. М. Стандартизація та гармонізація обліку в контексті міжнародних та національних стандартів. 2011. 5 с.
8. Тарасова, Т. О., Янчев, А. В., Волошан, І. Г., Тарасова, Т. А., & Волошан, И. Г. Первинний облік товарних операцій підприємств роздрібної торгівлі: монографія. 2015. 240 с.
9. Нестеренко, Є. Є. Методи обліку товарно-матеріальних запасів: вітчизняний та зарубіжний досвід. Управління розвитком, (17), 2013. 58-60.
10. Бунда, О. М., & Войналович, Н. М. Особливості застосування методів обліку виробничих запасів. Проблеми інноваційно-інвестиційного розвитку. Серія: Економіка та менеджмент. №12, 2017. 103-112 с.

11. Кучерук, О. А. Інтеграція комп'ютерних методів у традиційну систему організації самостійної роботи учнів з української мови. Вісник Житомирського державного університету імені Івана Франка. №57, 2011. 72-75 с.
12. Бурак, І. О. Переваги автоматизованого обліку розрахунків на підприємствах торгівлі. Економіка. Управління. Інновації. Серія: Економічні науки. №1. 2015. 8 с.
13. Кемарська, Л. Г., & Пугачова, І. С. Проблеми і перспективи впровадження комп'ютерної форми обліку і звітності бюджетними установами. Вісник Хмельницького національного університету. Економічні науки Вип. 2 №1, 2016. 87-92 с.
14. Савченко, Т. В., & Каюха, С. В. Порівняльна характеристика програмних продуктів «1С: Підприємство: Малий бізнес» і «ONEBOX». сучасному менеджменті», 2020. 267 с.
15. Михайлишин, С. П. Інформаційно-аналітична система обігу пакетованого насіння дочірнього підприємства “Світ насіння” товариства з обмеженою відповідальністю “Тернопільсортнасілляовоч”. 2017. 6с.
16. Костюнік, О. В., & Маламуж, А. А. Автоматизація основних засобів. Науковий вісник Херсонського державного університету. Сер.: Економічні науки. №2, 2017. Вип. 26 с. 128-130.
17. SAP Business One Professional. URL: <https://www.pcmag.com/reviews/sap-business-one-professional> (дата звернення 27.11.2024).
18. SAP Business One. URL: <https://www.softwareadvice.com/accounting/sap-business-one-profile/reviews/> (дата звернення 27.11.2024).
19. Reviews of SAP Business. URL: <https://www.capterra.com/p/214667/SAP-Business-One/reviews/> (дата звернення 27.11.2024).
20. About SAP Business. URL: <https://www.softwareadvice.com/accounting/sap-business-one-profile/> (дата звернення 27.11.2024).
21. Top Rated TradeSoft. URL: <https://www.g2.com/products/tradesoft/reviews> (дата звернення 27.11.2024).

22. What is TradeSoft? URL: <https://www.capterra.com/p/155369/TradeSoft/> (дата звернення 27.11.2024).
23. About TradeSoft. URL: <https://sourceforge.net/software/product/TradeSoft/> (дата звернення 27.11.2024).
24. TradeSoft Description. URL: <https://slashdot.org/software/p/TradeSoft/> (дата звернення 27.11.2024).
25. Ткаченко, Н. А., Демченко, В. А., Червоненко, Н. М., Заричная, Т. П., Ткаченко, Н. О., Демченко, В. О., ... & Зарічна, Т. П. Маркетинг у фармацевції та парфумерно-косметичній галузі:[навчально-методичний посібник]. 2016. 318 с.
26. Ключник, А. В., & Білозерцева, І. О. Теоретичний аналіз поняття «збут» та його функцій. Науковий вісник Херсонського державного університету. Серія: Економічні науки. 2014. с. 188-191.
27. Madamidola, O. A., Daramola, O. A., Akintola, K. G., & Adeboje, O. T. A Review of existing inventory management systems. International Journal of Research in Engineering and Science (IJRES), 2024. Vol. 12 No9. pp. 40-50.
28. Ganesh, R., & Prabu, G. Determination of internet banking usage and purpose with explanation of data flow diagram and use case diagram. International Journal of Management and Humanities. 2020. Vol. 4 No7, pp. 52-58.
29. Гладун А.В. Трьохрівнева архітектура побудови веб-систем. Вісник Національного університету "Львівська політехніка". 2018. – 315 с.
30. Liu, C., Song, Y., Li, R., Ma, W., Hao, J. L., & Qiang, G. Three-level modular grid system for sustainable construction of industrialized residential buildings: A case study in China. Journal of Cleaner Production, 2023. – 395 p.

## Лістинги програмного коду

ЛІСТИНГ 1. Код класу «HistoryComingGoodsProvider»

```

using WholesalePurchaseGoodsApp.AppCode;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WholesalePurchaseGoodsApp.Providers {
    class HistoryComingGoodsProvider {
        private string _ConnString =
System.Configuration.ConfigurationSettings.AppSettings["CONNECT"];
        private CategorysProvider _CategorysProvider = new CategorysProvider();
        private List<Categorys> _CategorysList = new List<Categorys>();
        private ProcurerProvider _ProcurerProvider = new ProcurerProvider();
        private List<Procurer> _ProcurerList = new List<Procurer>();

        public void InsertHistoryComingGoods(int CategorysId, string GoodsName, string BarCode, int
Quantity, string Measurements, string Producer,
        DateTime BuysDate, DateTime ExpirationdDate, double PurchasePrice, double SellingPrice, int
ProcurerId, string Description) {

            string SqlString = "INSERT INTO HistoryComingGoods (CategorysId, GoodsName, BarCode,
Quantity, Measurements, Producer, " +
                "BuysDate, ExpirationdDate, PurchasePrice, SellingPrice, ProcurerId, Description" +
                ") Values(@CategorysId, @GoodsName, @BarCode, @Quantity, @Measurements, @Producer,
@BuysDate, @ExpirationdDate, @PurchasePrice, @SellingPrice, @ProcurerId, @Description)";

            using (SqlConnection conn = new SqlConnection(_ConnString)) {
                using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@CategorysId", CategorysId);
                    cmd.Parameters.AddWithValue("@GoodsName", GoodsName);
                    cmd.Parameters.AddWithValue("@BarCode", BarCode);
                    cmd.Parameters.AddWithValue("@Quantity", Quantity);
                    cmd.Parameters.AddWithValue("@Measurements", Measurements);
                    cmd.Parameters.AddWithValue("@Producer", Producer);
                    cmd.Parameters.AddWithValue("@BuysDate", BuysDate);
                    cmd.Parameters.AddWithValue("@ExpirationdDate", ExpirationdDate);
                    cmd.Parameters.AddWithValue("@PurchasePrice", PurchasePrice);
                    cmd.Parameters.AddWithValue("@SellingPrice", SellingPrice);
                    cmd.Parameters.AddWithValue("@ProcurerId", ProcurerId);
                    cmd.Parameters.AddWithValue("@Description", Description);
                    conn.Open();
                    cmd.ExecuteNonQuery();
                }
            }
        }
    }
}

```

```

    }
}

public List<HistoryComingGoods> GetHistoryComingGood() {
    _ProcurerList = _ProcurerProvider.GetAllProcurer();
    _CategorysList = _CategorysProvider.GetAllCategorys();
    int i = 0;
    string SqlString = "SELECT * FROM HistoryComingGoods";

    List<HistoryComingGoods> listHistoryComingGoods = new List<HistoryComingGoods>();
    using (SqlConnection conn = new SqlConnection(_ConnString)) {
        using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
            conn.Open();
            using (SqlDataReader reader = cmd.ExecuteReader()) {
                while (reader.Read()) {
                    HistoryComingGoods oneHistoryComingGoods = new HistoryComingGoods();
                    oneHistoryComingGoods.Number = ++i;
                    oneHistoryComingGoods.HistoryComingGoodsId =
Convert.ToInt32(reader["HistoryComingGoodsId"]);
                    oneHistoryComingGoods.CategorysId = Convert.ToInt32(reader["CategorysId"]);
                    oneHistoryComingGoods.GoodsName = reader["GoodsName"].ToString();
                    oneHistoryComingGoods.BarCode = reader["BarCode"].ToString();
                    oneHistoryComingGoods.Quantity = Convert.ToInt32(reader["Quantity"]);
                    oneHistoryComingGoods.Measurements = reader["Measurements"].ToString();
                    oneHistoryComingGoods.Producer = reader["Producer"].ToString();
                    oneHistoryComingGoods.BuysDate = Convert.ToDateTime(reader["BuysDate"]);
                    oneHistoryComingGoods.ExpirationdDate =
Convert.ToDateTime(reader["ExpirationdDate"]);
                    oneHistoryComingGoods.PurchasePrice = Convert.ToDouble(reader["PurchasePrice"]);
                    oneHistoryComingGoods.SellingPrice = Convert.ToDouble(reader["SellingPrice"]);
                    oneHistoryComingGoods.ProcurerId = Convert.ToInt32(reader["ProcurerId"]);
                    oneHistoryComingGoods.Description = reader["Description"].ToString();
                    listHistoryComingGoods.Add(oneHistoryComingGoods);
                }
            }
        }
    }

    if (listHistoryComingGoods.Count == 0) {
        HistoryComingGoods noHistoryComingGoods = new HistoryComingGoods();
        noHistoryComingGoods.HistoryComingGoodsId = 0;
        noHistoryComingGoods.Message = NamesMy.NoDataNames.NoDataInHistoryComingGoods;
        listHistoryComingGoods.Add(noHistoryComingGoods);
    } else {
        for (int j = 0; j < listHistoryComingGoods.Count; j++) {
            listHistoryComingGoods[j].ProcurerFIO =
GetProcurerFIO(listHistoryComingGoods[j].ProcurerId, _ProcurerList);
            listHistoryComingGoods[j].CategorysName =
GetCategorysName(listHistoryComingGoods[j].CategorysId, _CategorysList);
        }
    }
    return listHistoryComingGoods;
}

```

```

}

private string GetProcurerFIO(int ProcurerId, List<Procurer> ProcurerList) {
    for (int i = 0; i < ProcurerList.Count; i++) {
        if (ProcurerId == ProcurerList[i].ProcurerId) {
            return ProcurerList[i].FIO;
        }
    }
    return "";
}

private string GetCategorysName(int CategorysId, List<Categorys> CategorysList) {
    for (int i = 0; i < CategorysList.Count; i++) {
        if (CategorysId == CategorysList[i].CategorysId) {
            return CategorysList[i].CategorysName;
        }
    }
    return "";
}

public HistoryComingGoods SelectedHistoryComingGoodsByHistoryComingGoodsId(int
HistoryComingGoodsId) {
    string SqlString = "SELECT * FROM HistoryComingGoods WHERE HistoryComingGoodsId =
@HistoryComingGoodsId";

    HistoryComingGoods oneHistoryComingGoods = new HistoryComingGoods();
    using (SqlConnection conn = new SqlConnection(_ ConnString)) {
        using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
            cmd.Parameters.AddWithValue("@HistoryComingGoodsId", HistoryComingGoodsId);
            conn.Open();
            using (SqlDataReader reader = cmd.ExecuteReader()) {
                while (reader.Read()) {
                    oneHistoryComingGoods.HistoryComingGoodsId =
Convert.ToInt32(reader["HistoryComingGoodsId"]);
                    oneHistoryComingGoods.CategorysId = Convert.ToInt32(reader["CategorysId"]);
                    oneHistoryComingGoods.GoodsName = reader["GoodsName"].ToString();
                    oneHistoryComingGoods.BarCode = reader["BarCode"].ToString();
                    oneHistoryComingGoods.Quantity = Convert.ToInt32(reader["Quantity"]);
                    oneHistoryComingGoods.Measurements = reader["Measurements"].ToString();
                    oneHistoryComingGoods.Producer = reader["Producer"].ToString();
                    oneHistoryComingGoods.BuysDate = Convert.ToDateTime(reader["BuysDate"]);
                    oneHistoryComingGoods.ExpirationdDate =
Convert.ToDateTime(reader["ExpirationdDate"]);
                    oneHistoryComingGoods.PurchasePrice = Convert.ToDouble(reader["PurchasePrice"]);
                    oneHistoryComingGoods.SellingPrice = Convert.ToDouble(reader["SellingPrice"]);
                    oneHistoryComingGoods.ProcurerId = Convert.ToInt32(reader["ProcurerId"]);
                    oneHistoryComingGoods.Description = reader["Description"].ToString();
                }
            }
        }
    }
    return oneHistoryComingGoods;
}

```

```

public void UpdateHistoryComingGoods(int CategorysId, string GoodsName, string BarCode, int
Quantity, string Measurements, string Producer,
    DateTime BuysDate, DateTime ExpirationdDate, double PurchasePrice, double SellingPrice, int
ProcurerId, string Description, int HistoryComingGoodsId) {

```

```

    string SqlString = "UPDATE HistoryComingGoods SET CategorysId=@CategorysId,
GoodsName=@GoodsName, BarCode=@BarCode, Quantity=@Quantity,
Measurements=@Measurements, Producer=@Producer," +
    "BuysDate=@BuysDate, ExpirationdDate=@ExpirationdDate, PurchasePrice=@PurchasePrice,
SellingPrice=@SellingPrice, ProcurerId=@ProcurerId, Description=@Description " +
    "WHERE HistoryComingGoodsId=@HistoryComingGoodsId";

```

```

using (SqlConnection conn = new SqlConnection(_ConnString)) {
    using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@CategorysId", CategorysId);
        cmd.Parameters.AddWithValue("@GoodsName", GoodsName);
        cmd.Parameters.AddWithValue("@BarCode", BarCode);
        cmd.Parameters.AddWithValue("@Quantity", Quantity);
        cmd.Parameters.AddWithValue("@Measurements", Measurements);
        cmd.Parameters.AddWithValue("@Producer", Producer);
        cmd.Parameters.AddWithValue("@BuysDate", BuysDate);
        cmd.Parameters.AddWithValue("@ExpirationdDate", ExpirationdDate);
        cmd.Parameters.AddWithValue("@PurchasePrice", PurchasePrice);
        cmd.Parameters.AddWithValue("@SellingPrice", SellingPrice);
        cmd.Parameters.AddWithValue("@ProcurerId", ProcurerId);
        cmd.Parameters.AddWithValue("@Description", Description);
        cmd.Parameters.AddWithValue("@HistoryComingGoodsId", HistoryComingGoodsId);
        conn.Open();
        cmd.ExecuteNonQuery();
    }
}
}

```

```

public void DeleteHistoryComingGoodsByHistoryComingGoodsId(int HistoryComingGoodsId) {
    string SqlString = "DELETE FROM HistoryComingGoods WHERE
HistoryComingGoodsId=@HistoryComingGoodsId";
    using (SqlConnection conn = new SqlConnection(_ConnString)) {
        using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
            cmd.Parameters.AddWithValue("@HistoryComingGoodsId", HistoryComingGoodsId);
            conn.Open();
            cmd.ExecuteNonQuery();
        }
    }
}
}
}
}

```

```

public class HistoryComingGoods {

```

```

private int _Number;
private int _HistoryComingGoodsId;
private int _CategorysId;
private string _GoodsName;
private string _BarCode;
private int _Quantity;
private string _Measurements;
private string _Producer;
private DateTime _BuysDate;
private DateTime _ExpirationdDate;
private double _PurchasePrice;
private double _SellingPrice;
private int _ProcurerId;
private string _Description;
private string _ProcurerFIO;
private string _CategorysName;
private string _Message;

public HistoryComingGoods() {
    _Number = 0;
    _HistoryComingGoodsId = 0;
    _CategorysId = 0;
    _GoodsName = String.Empty;
    _BarCode = String.Empty;
    _Quantity = 0;
    _Measurements = String.Empty;
    _Producer = String.Empty;
    _BuysDate = new DateTime();
    _ExpirationdDate = new DateTime();
    _PurchasePrice = 0.0;
    _SellingPrice = 0.0;
    _ProcurerId = 0;
    _Description = String.Empty;
    _ProcurerFIO = String.Empty;
    _CategorysName = String.Empty;
    _Message = String.Empty;
}

public int Number {
    set { _Number = value; }
    get { return _Number; }
}

public int HistoryComingGoodsId {
    set { _HistoryComingGoodsId = value; }
    get { return _HistoryComingGoodsId; }
}

public int CategorysId {
    set { _CategorysId = value; }
    get { return _CategorysId; }
}

public string GoodsName {
    set { _GoodsName = value; }

```

```

    get { return _GoodsName; }
}
public string BarCode {
    set { _BarCode = value; }
    get { return _BarCode; }
}
public int Quantity {
    set { _Quantity = value; }
    get { return _Quantity; }
}
public string Measurements {
    set { _Measurements = value; }
    get { return _Measurements; }
}
public string Producer {
    set { _Producer = value; }
    get { return _Producer; }
}
public DateTime BuysDate {
    set { _BuysDate = value; }
    get { return _BuysDate; }
}
public DateTime ExpirationdDate {
    set { _ExpirationdDate = value; }
    get { return _ExpirationdDate; }
}
public double PurchasePrice {
    set { _PurchasePrice = value; }
    get { return _PurchasePrice; }
}
public double SellingPrice {
    set { _SellingPrice = value; }
    get { return _SellingPrice; }
}
public string Description {
    set { _Description = value; }
    get { return _Description; }
}
public string ProcurerFIO {
    set { _ProcurerFIO = value; }
    get { return _ProcurerFIO; }
}
public string CategorysName {
    set { _CategorysName = value; }
    get { return _CategorysName; }
}
public int ProcurerId {
    set { _ProcurerId = value; }
    get { return _ProcurerId; }
}
public string Message {
    set { _Message = value; }

```

```

    get { return _Message; }
}
}

```

Лістинг 2. Код класу «GenData»

```

using Bogus;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WholesalePurchaseGoodsApp.Providers {
    internal class GenData {
        private string _ConnString =
            System.Configuration.ConfigurationSettings.AppSettings["CONNECT"];

        // Метод для заповнення таблиці Customers випадковими даними
        public void SeedCustomers(int numberOfRecords) {
            // Налаштування Faker для генерації даних українською мовою
            var faker = new Faker<Customers>("uk")
                .RuleFor(o => o.FirstName, f => f.Name.FirstName())
                .RuleFor(o => o.LastName, f => f.Name.LastName())
                .RuleFor(o => o.Phone, f => f.Phone.PhoneNumber("+380 ## ### ## ##"))
                .RuleFor(o => o.Address, f => f.Address.FullAddress())
                .RuleFor(o => o.Email, f => f.Internet.Email());

            // Створення записів
            for (int i = 0; i < numberOfRecords; i++) {
                var customer = faker.Generate();
                InsertCustomer(customer.FirstName, customer.LastName, customer.Phone, customer.Address,
                    customer.Email);
            }
        }

        // Метод для вставки одного запису в базу даних
        private void InsertCustomer(string FirstName, string LastName, string Phone, string Address, string
            Email) {
            using (SqlConnection connection = new SqlConnection(_ConnString)) {
                string query = "INSERT INTO Customers (FirstName, LastName, Phone, Address, Email) " +
                    "VALUES (@FirstName, @LastName, @Phone, @Address, @Email)";
                using (SqlCommand command = new SqlCommand(query, connection)) {
                    command.Parameters.AddWithValue("@FirstName", FirstName);
                    command.Parameters.AddWithValue("@LastName", LastName);
                    command.Parameters.AddWithValue("@Phone", Phone);
                    command.Parameters.AddWithValue("@Address", Address);
                    command.Parameters.AddWithValue("@Email", Email);

                    connection.Open();
                }
            }
        }
    }
}

```

```

        command.ExecuteNonQuery();
    }
}

// Метод для заповнення таблиці Procurer випадковими даними
public void SeedProcurers(int numberOfRecords) {
    // Налаштування Faker для генерації даних українською мовою
    var faker = new Faker<Procurer>("uk")
        .RuleFor(p => p.FirstName, f => f.Name.FirstName())
        .RuleFor(p => p.LastName, f => f.Name.LastName())
        .RuleFor(p => p.Phone, f => f.Phone.PhoneNumber("+380 ## ### ## ##"))
        .RuleFor(p => p.Address, f => f.Address.FullAddress())
        .RuleFor(p => p.Email, f => f.Internet.Email());

    // Створення записів
    for (int i = 0; i < numberOfRecords; i++) {
        var procurer = faker.Generate();
        InsertProcurer(procurer.FirstName, procurer.LastName, procurer.Phone, procurer.Address,
procurer.Email);
    }
}

// Метод для вставки одного запису в базу даних
private void InsertProcurer(string FirstName, string LastName, string Phone, string Address, string
Email) {
    using (SqlConnection connection = new SqlConnection(_ ConnString)) {
        string query = "INSERT INTO Procurer (FirstName, LastName, Phone, Address, Email) " +
            "VALUES (@FirstName, @LastName, @Phone, @Address, @Email)";
        using (SqlCommand command = new SqlCommand(query, connection)) {
            command.Parameters.AddWithValue("@FirstName", FirstName);
            command.Parameters.AddWithValue("@LastName", LastName);
            command.Parameters.AddWithValue("@Phone", Phone);
            command.Parameters.AddWithValue("@Address", Address);
            command.Parameters.AddWithValue("@Email", Email);

            connection.Open();
            command.ExecuteNonQuery();
        }
    }
}

public void AddDefaultCategories() {
    // Масив категорій із назвами та описами
    var categories = new[]
    {
        new { CategoryName = "Побутова техніка", Description = "Техніка для дому, включаючи
холодильники, пральні машини, пилососи тощо." },
        new { CategoryName = "Продукти харчування", Description = "Основні продукти
харчування: крупи, консерви, напої, солодощі." },
    }
}

```

```

    new { CategoryName = "Одяг та взуття", Description = "Одяг, взуття та аксесуари для чоловіків, жінок і дітей." },
    new { CategoryName = "Електроніка", Description = "Смартфони, ноутбуки, планшети, аксесуари та комплектуючі." },
    new { CategoryName = "Канцелярія та офісне обладнання", Description = "Папір, ручки, офісна техніка та матеріали." }
};

// Додавання кожної категорії в базу даних
foreach (var category in categories) {
    InsertCategory(category.CategoryName, category.Description);
}

// Метод для вставки однієї категорії в базу даних
private void InsertCategory(string CategoryName, string Description) {
    using (SqlConnection connection = new SqlConnection(_ConnString)) {
        string query = "INSERT INTO Categorys (CategorysName, Description) " +
            "VALUES (@CategorysName, @Description)";
        using (SqlCommand command = new SqlCommand(query, connection)) {
            command.Parameters.AddWithValue("@CategorysName", CategoryName);
            command.Parameters.AddWithValue("@Description", Description);

            connection.Open();
            command.ExecuteNonQuery();
        }
    }
}

public void PopulateGoods(List<Categorys> categories, List<Procurer> procurers) {
    var faker = new Faker("uk");

    // Реальні товари для кожної категорії
    var categoryGoods = new Dictionary<int, List<string>>
    {
        { 1, new List<string> { "Холодильник Samsung", "Пральна машина LG", "Пилосос Philips" } }, // Побутова техніка
        { 2, new List<string> { "Рис Басматі", "Консервований горошок", "Мінеральна вода Боржомі" } }, // Продукти харчування
        { 3, new List<string> { "Куртка зимова", "Кросівки Adidas", "Футболка Nike" } }, // Одяг та взуття
        { 4, new List<string> { "Смартфон iPhone 14", "Ноутбук Dell Inspiron", "Планшет Samsung Galaxy Tab" } }, // Електроніка
        { 5, new List<string> { "Блокнот A5", "Гелева ручка Parker", "Принтер HP LaserJet" } } // Канцелярія та офісне обладнання
    };

    foreach (var category in categories) {
        if (!categoryGoods.ContainsKey(category.CategorysId))
            continue;
    }
}

```

```

var goodsList = categoryGoods[category.CategoryId];

foreach (var goodsName in goodsList) {
    // Генеруємо додаткові дані для товару
    string barCode = faker.Commerce.Ean13();
    int quantity = faker.Random.Int(1, 100);
    string measurements = faker.PickRandom(new[] { "шт", "кг", "л", "м" });
    string producer = faker.Company.CompanyName();
    DateTime buysDate = faker.Date.Past(1); // Дата покупки до року назад
    DateTime expirationDate = faker.Date.Future(1, buysDate); // Термін придатності
    double purchasePrice = faker.Random.Double(10, 1000);
    double sellingPrice = Math.Round(purchasePrice * faker.Random.Double(1.2, 2.0), 2);
    int procurerId = faker.PickRandom(procurers).ProcurerId;
    string description = faker.Lorem.Sentence();

    // Вставляємо записи в обидві таблиці
    InsertHistoryComingGoods(
        category.CategoryId,
        goodsName,
        barCode,
        quantity,
        measurements,
        producer,
        buysDate,
        expirationDate,
        purchasePrice,
        sellingPrice,
        procurerId,
        description
    );

    InsertComingGoods(
        category.CategoryId,
        goodsName,
        barCode,
        quantity,
        measurements,
        producer,
        buysDate,
        expirationDate,
        purchasePrice,
        sellingPrice,
        procurerId,
        description
    );
}
}
}

```

```

private void InsertComingGoods(int CategorysId, string GoodsName, string BarCode, int Quantity,
string Measurements, string Producer,
DateTime BuysDate, DateTime ExpirationdDate, double PurchasePrice, double SellingPrice, int
ProcurerId, string Description) {

```

```

    string SqlString = "INSERT INTO ComingGoods (CategorysId, GoodsName, BarCode, Quantity,
Measurements, Producer, " +
        "BuysDate, ExpirationdDate, PurchasePrice, SellingPrice, ProcurerId, Description" +
        ") Values(@CategorysId, @GoodsName, @BarCode, @Quantity, @Measurements, @Producer,
@BuysDate, @ExpirationdDate, @PurchasePrice, @SellingPrice, @ProcurerId, @Description)";

```

```

using (SqlConnection conn = new SqlConnection(_ ConnString)) {
    using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@CategorysId", CategorysId);
        cmd.Parameters.AddWithValue("@GoodsName", GoodsName);
        cmd.Parameters.AddWithValue("@BarCode", BarCode);
        cmd.Parameters.AddWithValue("@Quantity", Quantity);
        cmd.Parameters.AddWithValue("@Measurements", Measurements);
        cmd.Parameters.AddWithValue("@Producer", Producer);
        cmd.Parameters.AddWithValue("@BuysDate", BuysDate);
        cmd.Parameters.AddWithValue("@ExpirationdDate", ExpirationdDate);
        cmd.Parameters.AddWithValue("@PurchasePrice", PurchasePrice);
        cmd.Parameters.AddWithValue("@SellingPrice", SellingPrice);
        cmd.Parameters.AddWithValue("@ProcurerId", ProcurerId);
        cmd.Parameters.AddWithValue("@Description", Description);
        conn.Open();
        cmd.ExecuteNonQuery();
    }
}

```

```

private void InsertHistoryComingGoods(int CategorysId, string GoodsName, string BarCode, int
Quantity, string Measurements, string Producer,
DateTime BuysDate, DateTime ExpirationdDate, double PurchasePrice, double SellingPrice, int
ProcurerId, string Description) {

```

```

    string SqlString = "INSERT INTO HistoryComingGoods (CategorysId, GoodsName, BarCode,
Quantity, Measurements, Producer, " +
        "BuysDate, ExpirationdDate, PurchasePrice, SellingPrice, ProcurerId, Description" +
        ") Values(@CategorysId, @GoodsName, @BarCode, @Quantity, @Measurements, @Producer,
@BuysDate, @ExpirationdDate, @PurchasePrice, @SellingPrice, @ProcurerId, @Description)";

```

```

using (SqlConnection conn = new SqlConnection(_ ConnString)) {
    using (SqlCommand cmd = new SqlCommand(SqlString, conn)) {
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@CategorysId", CategorysId);
        cmd.Parameters.AddWithValue("@GoodsName", GoodsName);
        cmd.Parameters.AddWithValue("@BarCode", BarCode);
        cmd.Parameters.AddWithValue("@Quantity", Quantity);
        cmd.Parameters.AddWithValue("@Measurements", Measurements);
        cmd.Parameters.AddWithValue("@Producer", Producer);
    }
}

```

```

cmd.Parameters.AddWithValue("@BuysDate", BuysDate);
cmd.Parameters.AddWithValue("@ExpirationdDate", ExpirationdDate);
cmd.Parameters.AddWithValue("@PurchasePrice", PurchasePrice);
cmd.Parameters.AddWithValue("@SellingPrice", SellingPrice);
cmd.Parameters.AddWithValue("@ProcurerId", ProcurerId);
cmd.Parameters.AddWithValue("@Description", Description);
conn.Open();
cmd.ExecuteNonQuery();
}
}
}

```

```

}
}

```

Лістинг 3. Код класу «ProcuresSearchForm»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using WholesalePurchaseGoodsApp.AppCode;

namespace WholesalePurchaseGoodsApp.Forms.Srch {
    public partial class ProcuresSearchForm : Form {
        LogicBLL _LogicBLL = new LogicBLL();
        List<Procurer> _ProcurerList = new List<Procurer>();

        public ProcuresSearchForm() {
            InitializeComponent();
        }

        private void SearchBtn_Click(object sender, EventArgs e) {
            _ProcurerList = _LogicBLL.GetAllSearchProcurerByFIO(FIOTBox.Text);
            GetRaport(_ProcurerList);
        }

        public void GetRaport(List<Procurer> ProcurerList) {
            RaportTBox.Text = "Результати пошуку:\r\n";
            for (int i = 0; i < 167; i++) {
                RaportTBox.Text += "-";
            }
            RaportTBox.Text += "\r\n";
        }
    }
}

```

```

        RaportTBox.Text += String.Format("{0,3}|{1, -60}|{2, -50}|{3, -50}|r\n", "№", "П.И.Б", "Номер
телефону", "Ел. адрес");
        for (int i = 0; i < 167; i++) {
            RaportTBox.Text += "-";
        }
        RaportTBox.Text += "r\n";
        for (int i = 0; i < ProcurerList.Count; i++) {
            string raportString = String.Format("{0,3}|{1, -60}|{2, -50}|{3, -50}|r\n",
            ProcurerList[i].Number,
            ProcurerList[i].FIO,
            ProcurerList[i].Phone,
            ProcurerList[i].Email);
            RaportTBox.Text += raportString;
        }
    }
}
}
}
}

```

## Порівняння каналів роздрібного збуту

Канал збуту	Основні характеристики	Переваги	Недоліки	Приклади
Традиційні канали	Офлайн-продаж товарів у фізичних магазинах	Безпосередня взаємодія з клієнтом, можливість оцінити продукт особисто	Витрати на оренду приміщень, персонал, обмежена географія збуту	Супермаркети, бутики, роздрібні точки
Цифрові канали (онлайн)	Продаж товарів через інтернет-платформи та додатки	Широка географія, зручність покупки для клієнтів, мінімізація витрат	Відсутність фізичного контакту з товаром, необхідність доставки	Інтернет-магазини, соціальні мережі
Соціальні мережі	Продаж через платформи соціальних мереж (Instagram, Facebook, TikTok)	Взаємодія з молодіжною аудиторією, використання впливу блогерів	Обмежена аналітика продажів, залежність від політик соцмереж	Реклама та магазини у Facebook та Instagram
Мобільні застосунки	Спеціалізовані програми для зручного замовлення товарів	Персоналізація, простота у використанні, лояльність клієнтів	Витрати на розробку додатка, підтримка й оновлення	Glovo, Rozetka, Amazon

## Опис варіантів використання системи

№	Назва	Опис
1	2	3
UC1	Авторизація в системі	Забезпечує користувачеві можливість пройти авторизацію для доступу до функціоналу системи
UC2	Реєстрація користувача	Дозволяє провести реєстрацію нового користувача у системі
UC3	Перегляд списку користувачів	Дозволяє адміністратору переглядати перелік усіх зареєстрованих користувачів системи
UC4	Додавання нового користувача	Надає адміністратору можливість створити обліковий запис нового користувача
UC5	Редагування користувача	Дозволяє адміністратору змінювати інформацію про конкретного користувача
UC6	Перегляд списку постачальників	Забезпечує доступ до переліку постачальників із можливістю перегляду деталей про них
UC7	Додавання постачальника	Дозволяє користувачеві додати нового постачальника до бази даних системи.
UC8	Редагування постачальника	Надає можливість оновлювати інформацію про постачальників.
UC9	Перегляд категорій товарів	Забезпечує доступ до переліку категорій товарів, доступних у системі.
UC10	Додавання категорії товару	Дозволяє користувачеві додати нову категорію товару до системи.
UC11	Редагування категорії товару	Надає можливість змінювати дані про існуючу категорію товару.
UC12	Перегляд списку покупців	Дозволяє переглядати перелік покупців із деталями про кожного.
UC13	Додавання покупця	Забезпечує можливість додати нового покупця до системи.
UC14	Редагування покупця	Надає можливість оновлювати дані про покупців.
UC15	Оформлення приходу товару	Дозволяє здійснити облік надходження товару від постачальників.
UC16	Оформлення реалізації товару	Забезпечує облік продажу товарів роздрібним покупцям.

## Продовження табл

UC17	Оформлення списання товару	Дозволяє зафіксувати списання товарів, які втратили товарний вигляд або стали непридатними.
UC18	Перегляд наявних товарів	Надає користувачеві можливість переглядати актуальний стан складських запасів.
UC19	Пошук товарів	Дозволяє шукати товари за заданими параметрами, такими як назва, код чи категорія.
UC20	Пошук постачальників	Забезпечує пошук інформації про постачальників за заданими критеріями.
UC21	Пошук покупців	Надає можливість шукати інформацію про покупців у базі даних.
UC22	Звітність щодо приходу товарів	Забезпечує генерацію звітів про надходження товарів за певний період.
UC23	Звітність щодо реалізації товарів	Надає можливість створювати звіти про обсяг реалізованих товарів.
UC24	Звітність щодо списання товарів	Дозволяє формувати звіти про списання товарів із вказанням причин.
UC25	Перегляд системних подій	Надає адміністратору доступ до журналу системних подій для моніторингу дій користувачів і системних процесів.

## Структура таблиці «DischargeGoods»

Назва поля	Тип даних	ПК	ЗК	Опис поля
DischargeGoodsId	INT	+	-	Ідентифікатор запису про реалізацію товару, первинний ключ
CategorysId	INT	-	+	Ідентифікатор категорії товару
GoodsName	NVARCHAR (200)	-	-	Назва товару
BarCode	NVARCHAR (200)	-	-	Штрих-код товару
Quantity	INT	-	-	Кількість проданих одиниць товару
Measurements	NVARCHAR (200)	-	-	Одиниця вимірювання
Producer	NVARCHAR (200)	-	-	Виробник товару
DateOfSale	DATETIME	-	-	Дата продажу товару
ExpirationdDate	DATETIME	-	-	Термін придатності товару
SellingPrice	FLOAT (53)	-	-	Роздрібна ціна товару
CustomersId	INT	-	+	Ідентифікатор покупця, зв'язок із таблицею «Customers»