

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра комп'ютерних систем та управління

## КОМПЛЕКСНА МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**Розробка автоматизованої системи обліку складських товарів і підтримки торговельної діяльності. Частина 1. Створення бази даних та розробка десктопного програмного забезпечення.**

Виконав: студент 2 курсу, групи 1АКІТР-24м спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та роботехніка

 (шифр і назва спеціальності)  
Павло МИХАЙЛЕНКО  
(ПІБ студента)

Керівник: д.т.н., професор кафедри КСУ  
Марія ЮХИМЧУК  
(науковий ступінь, вчене звання / посада, ПІБ керівника)

« 12 » \_\_\_\_\_ 12 \_\_\_\_\_ 2025 р.

Опонент: к.т.н., доцент кафедри АІТ  
Володимир ГАРМАШ  
(науковий ступінь, вчене звання / посада, ПІБ опонента)

« 12 » \_\_\_\_\_ 12 \_\_\_\_\_ 2025 р.

**Допущено до Захисту**  
Завідувач Кафедри КСУ

д.т.н., проф.  
Вячеслав КОВТУН  
(науковий ступінь, вчене звання)

« 13 » \_\_\_\_\_ 12 \_\_\_\_\_ 2025р.

Вінницький Національний Технічний Університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра комп'ютерних систем управління  
Галузь знань – 17 – Електроніка, автоматизація та електронні комунікації  
Спеціальність – 174 - Автоматизація, комп'ютерно-інтегровані технології та  
робототехніка  
Освітньо-професійна програма – Інтелектуальні комп'ютерні системи

**ЗАТВЕРДЖУЮ**

Зав. кафедри КСУ ВНТУ,  
д.т.н., професор  
  
Вячеслав КОВТУН  
«26» вересня 2025 р.

## **ЗАВДАННЯ**

НА КОМПЛЕКСНУ МАГІСТРЕЦЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ  
студенту групи 1АКІТР-24м Михайленко П. О.

1. Тема магістерської кваліфікаційної роботи: *Розробка автоматизованої системи обліку складських товарів і підтримки торговельної діяльності. Частина 1. Створення бази даних та розробка десктопного програмного забезпечення.*

керівник комплексної магістерської кваліфікаційної роботи д.т.н., Юхимчук М.С.  
затверджені наказом ВНТУ від «24» вересня 2025 року №313

2. Термін подання студентом роботи «10» грудня 2025 року.

3. Вихідні дані до роботи: структура та логіка складського обліку і торговельної діяльності, аналіз бізнес-процесів, вимоги до автоматизації системи обліку складських товарів, потреба у створенні централізованої бази даних та розробці десктопного застосунку для управління товарами, постачанням і продажами, перелік матеріалів до розробки:

1. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. — Addison-Wesley, 2003.
2. Pressman R. Software Engineering: A Practitioner's Approach. — McGraw-Hill, 2019.
3. Litvinenko O. Управління інформаційними системами підприємства. — Київ: Наукова думка, 2015.
4. Kirichenko O. Проектування баз даних: навчальний посібник. — Київ: КНТ, 2018.
5. Сергієнко Т. О. Мобільні додатки: розробка та тестування. — Харків: IT-Books, 2020.

4. Зміст текстової частини: вступ, аналіз предметної області та існуючих програмних продуктів, формування вимог та проектування архітектури системи, проектування та створення бази даних, розробка UML-діаграм варіантів використання та діяльності, розробка програмного забезпечення, тестування роботи системи та усунення помилок, економічна частина, висновки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень). архітектура системи, UML Use-Case діаграми, UML Activity діаграми, ER-модель бази даних, структурна схема бази даних, інтерфейси розробленого десктопного застосунку, приклади таблиць і форм.

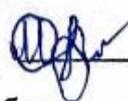
1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	Дата
		Завдання видав	Виконання прийняв
4	Ратушняк О.Г., доцент кафедри економіки підприємства і виробничого менеджменту		

2. Дата видачі завдання "25" вересня 2025 року

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Терміни виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області та об'єкта автоматизації	26.09.2025	04.10.2025	
2	Аналіз існуючих програмних продуктів і систем для автоматизації складського обліку та торгівлі	05.10.2025	14.10.2025	
3	Формування вимог та проектування архітектури систем	15.10.2025	24.10.2025	
4	Проектування та створення бази даних	25.10.2025	04.11.2025	
5	Розробка UML-діаграм варіантів використання та діяльності	05.11.2025	13.11.2025	
6	Розробка програмного забезпечення (десктопний застосунок)	14.11.2025	27.11.2025	
7	Тестування роботи системи, налагодження функцій, усунення помилок	28.11.2025	02.12.2025	
8	Оформлення текстової частини та додатків МКР	03.12.2025	06.12.2025	
9	Розрахунок економічної частини	07.12.2025	09.12.2025	
10	Оформлення МКР та збір підписів супроводжувальних документів	10.12.2025	13.12.2025	
11	Захист МКР	15.12.2025	18.12.2025	

Студент  Павло МИХАЙЛЕНКО

Керівник роботи  Марія ЮХИМЧУК

## АНОТАЦІЯ

УДК: 004.4/.6:658.787

Михайленко П. О. Автоматизована система обліку складських товарів та підтримки торговельної діяльності. Магістерська кваліфікаційна робота зі спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2025. 105 с.

На укр. мові. Бібліогр.: 62 назв; рис.: 44; табл.: 6.

У магістерській кваліфікаційній роботі розроблено та досліджено автоматизовану систему обліку складських товарів і підтримки торговельної діяльності. Метою роботи є підвищення ефективності діяльності торговельних підприємств шляхом автоматизації процесів обліку складських товарів і торговельних операцій, зменшення впливу людського фактору та підвищення точності й оперативності обліку. У роботі виконано аналіз сучасних програмних рішень для автоматизації торгівлі, визначено вимоги до системи, спроектовано архітектуру та базу даних, розроблено UML-діаграми. Реалізовано десктопне програмне забезпечення для обліку товарних запасів, реєстрації постачань і продажів, формування звітності та резервного копіювання даних. Проведено тестування роботи системи та оцінено витрати на її розробку. Ілюстративна частина містить 6 графічних матеріалів.

Ключові слова: автоматизована система, облік товарів, база даних, програмне забезпечення.

## ANNOTATION

UDC: 004.4/.6:658.787

Mykhailenko P. O. Automated system for warehouse inventory accounting and support of trading activities. Master's qualification thesis in specialty 174 – Automation, Computer-Integrated Technologies and Robotics, educational program – Intelligent Computer Systems. Vinnytsia: Vinnytsia National Technical University, 2025. 105 p.

In Ukrainian. References: 62; figures: 44; tables: 6.

The master's qualification thesis is devoted to the development and study of an automated system for warehouse inventory accounting and support of trading activities. The purpose of the work is to increase the efficiency of trading enterprises by automating warehouse inventory accounting and trading operations, reducing the impact of the human factor, and improving the accuracy and efficiency of accounting. The thesis analyzes modern software solutions for trade automation, defines system requirements, and designs the software architecture and database. UML diagrams were developed. A desktop software application was implemented to support inventory accounting, registration of supplies and sales, report generation, and data backup. The system was tested, and the costs of its development were evaluated. The illustrative part contains 6 graphical materials.

Keywords: automated system, inventory accounting, database, software.

## ЗМІСТ

ВСТУП .....	4
1 ДОСЛІДЖЕННЯ ПРАКТИЧНИХ АСПЕКТІВ РОБОТИ АВТОМАТИЗОВАНИХ СИСТЕМ, ЩО ЗАБЕЗПЕЧУЮТЬ ОБЛІК СКЛАДСЬКИХ ТОВАРІВ .....	8
1.1 Аналіз напрямків роботи сучасних автоматизованих систем управління обліку складських товарів та торгівлі.....	8
1.2 Дослідження етапів процесу вибору необхідної автоматизованої системи управління.....	12
1.3 Дослідження переваг для компанії, від використання автоматизованих систем управління .....	15
1.4 Уточнена постановка задачі.....	17
1.5 Висновки до розділу.....	17
2 ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ СКЛАДСЬКИХ ТОВАРІВ ТА ПІДТРИМКИ ТОРГІВЛІ .....	18
2.1 Дослідження функціональних можливостей прикладних програм- аналогів систем управління, що забезпечують облік складських товарів .....	18
2.2 Визначення функціональних можливостей автоматизованої системи, що розробляється .....	33
2.3 Проєктування бази даних системи .....	34
2.4 Розробка UML-діаграм .....	40
2.4.1 Розробка UML-діаграми варіантів використання .....	40
2.4.2 Розробка UML-діаграми компонентів .....	43
2.5 Висновки до розділу .....	44

3 РОЗРОБКА ТА ТЕСТУВАННЯ РОБОТИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ СКЛАДСЬКИХ ТОВАРІВ ТА ПІДТРИМКИ ТОРГІВЛІ .....	45
3.1 Розробка автоматизованої системи в середовищі програмування Microsoft Visual Studio.....	45
3.2 Перевірка правильності роботи автоматизованої системи обліку складських товарів та підтримки торгівлі .....	51
3.3 Висновки до розділу .....	62
4 ЕКОНОМІЧНА ЧАСТИНА .....	62
4.1 Комерційний та технологічний аудит науково-технічної розробки .....	62
4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи .....	66
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором...	71
4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	74
4.5 Висновки до розділу .....	76
ВИСНОВКИ .....	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	78
Додатки .....	84
Додаток А (обов'язковий) протокол перевірки навчальної (кваліфікаційної) роботи .....	85
Додаток Б (обов'язковий) Технічне завдання .....	86
Додаток В (довідковий) Лістинг програми .....	89
Додаток Г (обов'язковий) Ілюстративна частина .....	95

## ВСТУП

**Актуальність.** Системи обліку складських товарів та торгівлі, що створюються для автоматизації процесів підприємницької діяльності складаються з безлічі функцій, що утворюють єдиний надійний комплекс з управління підприємством. Після інтеграції системи в роботу підприємства всі дії фіксуються через програмне забезпечення, а обчислення, відповідно, проводяться без помилок та впливу людського фактору.

Для роботи з готовою системою обліку та торгівлі не потрібно знань бухгалтерського обліку або програмування, оскільки є простий і зрозумілий користувачу інтерфейс, а безкоштовний пробний період дозволить ознайомитися з програмою і її функціоналом, щоб зрозуміти, наскільки саме вона підходить конкретному бізнесу [1].

Автоматизація в роздрібній торгівлі відноситься до будь-якого процесу, який може виконуватися машинами, а не людьми. В галузі електронної комерції є багато речей, які можна автоматизувати, починаючи від управління запасами, закінчуючи маркетинговою діяльністю та бухгалтерською звітністю.

Системи обліку та торгівлі мають істотні переваги в таких напрямках:

### 1. Скорочення витрат

Автоматизація допоможе знизити витрати, в першу чергу на оплату праці, оскільки чим більший об'єм роботи покритий програмною реалізацією, тим менше на нього потрібно використовувати ручної роботи. Не кажучи вже про те, що автоматизація бере на себе всі рутинні повторювані завдання, залишаючи час співробітникам на більш важливу роботу протягом дня.

Існують також певні завдання, які можна повністю автоматизувати, тоді зникне потреба наймати когось для виконання цих обов'язків. Це заощадить заробітну плату, виплати, офісні приміщення та інші суттєві статті витрат [2].

### 2. Підвищення ефективності

Автоматизація може зробити співробітників та бізнес загалом ефективнішими, заощадивши багато часу та енергії. Оскільки робота Інтернет-магазинів завжди відкрита, і в будь-який момент часу хтось із іншого кінця світу

може відвідати веб-сайт та зробити замовлення. Автоматизація може дозволити розміщувати та обробляти замовлення і виконувати їх набагато швидше, ніж якби потрібно було чекати наступного дня для початку робочої зміни співробітника магазину.

### 3. Зменшення помилок людського фактору

Щоразу, коли задіяний людський фактор, будуть виникати помилки, з якими потрібно боротися. Іноді це може бути критичним і може заважати отриманню точного уявлення про стан бізнесу та іншу важливу інформацію. Це особливо актуально, якщо бізнес розвивається, і завжди потрібно бути в курсі кількості запасів, кількості продажів, лояльності клієнтів та багатьох інших даних. Автоматизація бізнес процесів може значно зменшити, або ж і повністю усунути, помилки.

Отже, інвестування в автоматизацію роздрібною торгівлі стає необхідністю для підвищення продуктивності праці співробітників та надання їм можливості приймати розумніші та впевненіші рішення. Це окупиться більшим запасом робочого часу, особливо коли бізнес починає масштабуватися, і постане необхідність опрацьовувати більшу кількість даних, забезпечуючи можливість обслуговувати клієнтів на тому самому рівні, що і тоді, коли компанія була ще стартапом або малим бізнесом [3].

Інноваційні програмні рішення для роздрібною торгівлі, що беруть на себе задачі з обліку складських товарів та торгівлі, можуть допомогти отримати найбільші переваги від автоматизації та підняти бізнес на новий рівень. Підвищуючи ефективність, одночасно знижуючи витрати, компанія отримує перевагу над конкурентами.

**Мета і завдання дослідження.** Метою магістерської кваліфікаційної роботи є підвищення ефективності роботи торгової мережі за рахунок створення автоматизованої системи управління обліком складських товарів та торгівлі, що дозволить автоматизувати рутинні завдання та мінімізувати вплив помилок людського фактору на роботу підприємства, задля зменшення втрат та ефективного розподілу вивільнених робочих ресурсів на інші області роботи компанії.

Для досягнення наведеної мети були поставлені та вирішені наступні задачі:

- 1) проведено аналіз існуючих програмних рішень роботи систем обліку та торгівлі;
- 2) проведено аналіз сучасних технологій реалізації автоматизованих систем управління;
- 3) досліджено практичні аспекти функціонування систем управління обліку складських товарів та торгівлі;
- 4) розроблено десктопне програмне рішення автоматизованої системи управління обліком складських товарів та торгівлі, що функціонує в режимі реального часу;
- 5) проведено дослідження та тестування отриманих результатів.

**Об'єктом дослідження** є процес створення автоматизованої системи управління обліком складських товарів та торгівлі.

**Предметом дослідження** є методи та засоби програмування, що забезпечують функціонування автоматизованої системи управління обліком складських товарів та торгівлі.

**Методи дослідження.** У процесі дослідження застосовувалися: теорія алгоритмів; Agile підхід до розробки програмного забезпечення, методика Scrum, поєднання мови гіпертекстової розмітки HTML, об'єктно-орієнтованої мови програмування JavaScript та каскадних таблиць стилів CSS.

#### **Новизна одержаних результатів.**

1. Встановлено список першочергових функцій, які необхідні для ефективного функціонування десктопного програмного рішення.
2. Розроблено та нормалізовано структуру бази даних, що необхідна для покриття усіх спроектованих функцій роботи автоматизованої системи управління обліком складських товарів та торгівлі.
3. У роботібу розроблено універсальний варіант програмного рішення для роботи автоматизованої системи обліку складських товарів та торгівлі, який містить ряд головних та допоміжних функцій, що дозволить автоматизувати рутинні завдання та мінімізувати вплив помилок людського фактору на роботу підприємства, задля зменшення втрат та

ефективного розподілу вивільнених робочих ресурсів на інші області роботи компанії.

**Практичне значення одержаних результатів** полягає в тому, що на основі отриманих теоретичних положень розроблено десктопний застосунок для реалізації необхідних функціональних можливостей.

**Достовірність теоретичних положень** магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням усіх методів та засобів функціонування інформаційних систем, вживаючи Agile методика при розробці десктопного додатку та безперервною роботою розроблено програмного засобу, а саме автоматизованої системи управління обліком складських товарів та торгівлі.

**Особистий внесок здобувача.** Усі результати отримано автором самостійно.

# 1 ДОСЛІДЖЕННЯ ПРАКТИЧНИХ АСПЕКТІВ РОБОТИ АВТОМАТИЗОВАНИХ СИСТЕМ, ЩО ЗАБЕЗПЕЧУЮТЬ ОБЛІК СКЛАДСЬКИХ ТОВАРІВ

## 1.1 Аналіз напрямків роботи сучасних автоматизованих систем управління обліку складських товарів та торгівлі

В залежності від масштабів та профілю бізнесу, автоматизовані системи обліку складських товарів та торгівлі можуть мати відмінний функціонал, проте існують головні напрямки, які повинні покриватись якісним програмним забезпеченням для найбільшої ефективності їх використання [4]. Схематично напрямки роботи автоматизованих систем обліку товарів та торгівлі зображено на рисунку 1.1.

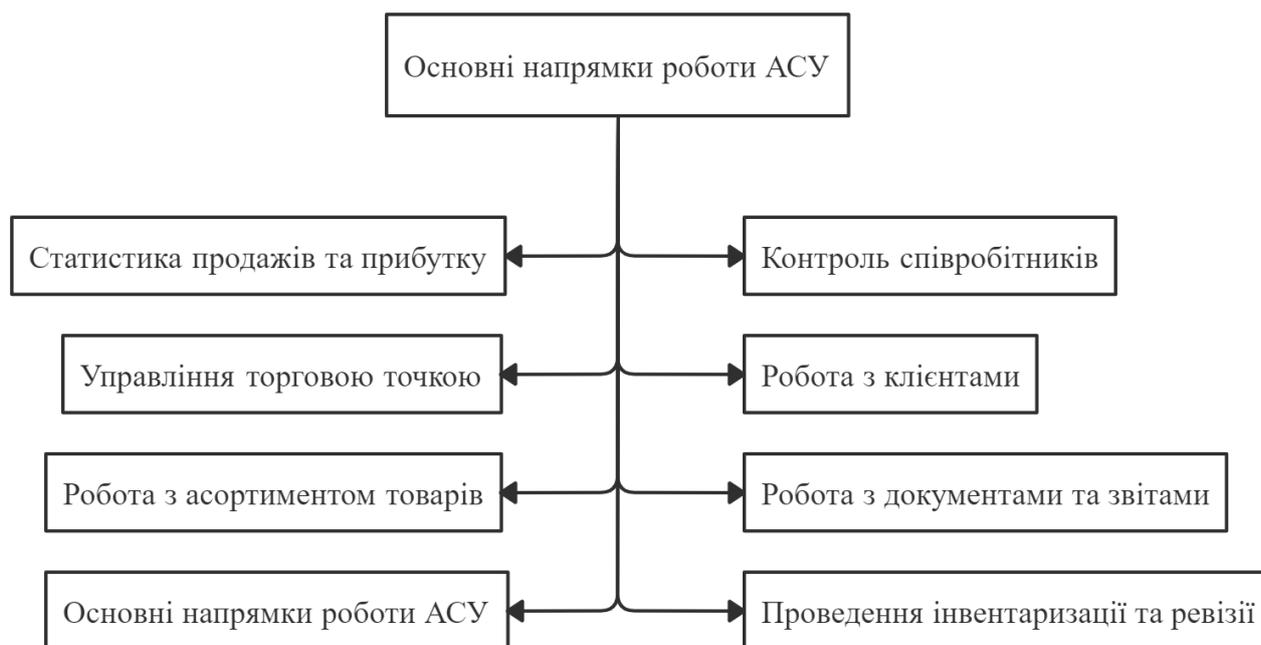


Рисунок 1.1 - Напрямки роботи автоматизованих систем обліку товарів та торгівлі

Кожен із цих напрямків можна розписати детальніше наступним чином.

За допомогою комплексу **статистики продажів і прибутку** можна проаналізувати всі основні показники підприємства, включаючи:

- Звіти з продажу за будь-який період: починаючи від звіту за пару годин до декількох років.

- Продажі за категоріями товарів, дозволяє оцінити, які товари продавалися краще за інших і скільки прибутку принесли.

- Звіти для замовлень допоможуть не забути дозамовити товари в потрібній кількості, враховуючи попит на товар.

- Створення звітів за певними характеристиками товарів, наприклад, швидкості псування та реалізації.

- Звіти по співробітниках - інформація про всі дії вчинені певним співробітником, його діяльність в програмі, кількість продажів і виручки. Завдяки таким звітам також зручно відстежувати ефективність роботи співробітників і нараховувати їм зарплату [5].

**Зручне управління торговою точкою** через інтернет або мобільний додаток допоможе:

- Організувати підтримку будь-якої кількості точок мережі і управління ними. Можна налаштувати переміщення товарів, переглядати залишки на інших точках і більш точно налаштувати логістику.

- Можна керувати торговою точкою з дому, коли у немає часу або бажання їхати на точку особисто. Для власника будуть доступні всі функції для управління. Так само можна звузити коло прав доступу і дати можливість роботи з дому бухгалтеру або іншим співробітникам.

- Зі спрощеним функціоналом є мобільний застосунок, яке дозволить переглядати інформацію і дії на точці, в будь-якому місці, де є інтернет, для синхронізації з точкою.

- Налаштування відправки звітів на email - можна налаштує надсилати звіти щодня, щомісяця, чи щокварталу, за категоріями і іншими показниками.

- Налаштувати оповіщення на пошту при продажі на певну суму або інших діях.

- Можна переглядати статистику в реальному часі.

**Робота з асортиментом товарів** допомагає реалізувати всі основні поведінкові і функціональні особливості для товарних сутностей.

- Асортиментні ряди - використовуються в разі, якщо товар може відрізнитися за кольором, розміром, комплектацією і іншим параметрам;

- Комплекти - з їх допомогою можна сформувати набір;

- Калькуляції - дозволяють скласти список складових замовлення. При продажу складові товари списуються з залишків, враховуючи втрати і на основі витрачених ресурсів формується точна ціна.

- Товари в програму можна додавати заповнюючи картки вручну, але для зручності є можливість завантаження товарів з файлу Excel.

- Пошук товарів по штрих коду, найменуванню, характеристикам.

- Операції повернення і списання. Можна бронювати і резервувати товар.

- Облік продажів за готівку і банківською картою. А також в борг і в розстрочку [6].

**Контроль співробітників** означає, що всі дії та інформація про співробітників заноситься в базу.

- Програма формує журнал роботи персоналу. В системі обліку можуть працювати тільки ті працівники, які внесені в базу адміністратором або власником.

- У системах обліку велика увага приділяється правам доступу, що робить безпечною роботу для користувачів, а власнику дає гарантію, що махінації на його точці проводиться не будуть.

- Всі дії в програмі залишають свій слід: продаж, зміна цін, додавання клієнтів, зміна налаштувань програми, перегляд звітів і багато іншого. Журнал дій може бути доступний тільки власнику або адміністратору.

- Враховується роботу позмінно і якщо співробітник отримує оклад виходячи з відсотка від продажів, система обліку може автоматично підрахувати його зарплату виходячи з особистих показників працівника компанії.

- Фіксування знятої готівки з каси з зазначенням категорії, причини і користувача.

- Статистика продажів по кожному співробітнику - виручка, дохід, кількість продажів, продажу за готівку, картою, зі знижкою і інше [4, 6].

**Робота з клієнтами** передбачає проведення акцій, знижок та програм лояльності, а також забезпечує наступні можливості системи:

- У базу можна вносити контактну інформацію про клієнтів. Створену базу клієнтів можна доповнювати і використовувати в подальшому, якщо потрібно буде перенести її на інший ресурс.

- Для створеного клієнта можна створити дисконтну карту, яку він зможе використовувати при покупках. Це може бути код, який вводиться при продажі вручну або пластикова карта для зручного використання.

- Типи знижок можуть бути трьох типів - фіксована сума, фіксований відсоток або накопичувальна система.

- Якщо здійснюються продажі в борг, то можна буде подивитися боржників сформувавши звіт, в якому буде інформація про клієнта, сума його заборгованості і інші поля.

- Прийом замовлень на тимчасово відсутні товари, які будуть автоматично додані до звітів про дозамовлення.

- Синхронізація клієнтів і бази товарів між точками.

**Робота з документами та звітами** здійснюється завдяки:

- Вбудованому гнучкому редактору шаблонів документів, що дозволяє адаптувати формат документів, під потреби користувача.

- Наявність цінників і етикеток будь-якого формату і розміру. Так само можна редагувати інформацію друкувати на чеку, етикетці і ціннику.

- Формування товарних чеків, накладних та рахунків доступні для друку з потрібною інформацією.

- Акти інвентаризації та списання.

- В цілому всі звіти і операції, що виконуються, можна отримати в електронному вигляді і в подальшому роздрукувати на паперовому носії [7].

**Інвентаризації та ревізії**

- У програмі можна виконувати інвентаризацію. Можна вводити дані вручну або користуватися сканером. Також можна використовувати сканер з накопичувачем, який відзначає товари, записує в пам'ять і після завершення вивантажує дані з пам'яті в програму. Це зручно, якщо, наприклад, є великий склад.

- Інвентаризацію можна припиняти, якщо в цьому є потреба. Всі залишки змінені на час зупинки інвентаризації будуть відображені при поновленні.

- Можливість переглянути журнал недостачі готівки.

- Надходження, списання та повернення товару також документуються і доступні для перегляду.

- Можливість друку актів інвентаризації, а також доступ до історії проведених інвентаризацій.

- База товарів автоматично оновлюється після проведення інвентаризації з урахуванням зафіксованих залишків.

### **Підтримка та інтеграція торгового обладнання**

- Використання сканерів штрих-кодів - дротяних, бездротових, з накопичувачем.

- Підтримка принтерів чеків та етикеток різного розміру та формату.

- Друк документів на офісних принтерах формату А4.

- Фіскальна техніка - касові апарати, фіскальні реєстратори.

- Торговельні ваги, програмовані ваги з друком етикеток.

- Термінали з сенсорним екраном, з декількома екранами.

- Підтримка зчитувачів магнітних карт для персоналу і клієнтів.

Прагнення компаній до автоматизації бізнес-процесів обмежене лише фінансовими можливостями, тому вибір автоматизованої системи управління починається з планування витрат на власне впровадження інноваційних технологій та вибором критеріїв і функціонального призначення програмного рішення [5, 7].

## 1.2 Дослідження етапів процесу вибору необхідної автоматизованої системи управління

Серед сотень програмних рішень для управління складськими товарами та торгівлі керівникам компаній необхідно обрати найкращий варіант, що буде максимально доцільним та ефективним для конкретного бізнесу. Оскільки впровадження автоматизованої системи розглядається як довгострокова перспектива, що, до того ж, вимагає немалих капіталовкладень, то при виборі керуються наступними етапами, що зображені на рисунку 1.2 [8].



Рисунок 1.2 - Етапи процесу вибору найкращого програмного рішення для управління складськими товарами та торгівлі

1. Створення команди по підбору автоматизованої системи управління складськими товарами та торгівлі

Команда проекту контролює весь процес пошуку наявних рішень від постачальників та планує впровадження. Керівник проекту може мати право приймати рішення про купівлю від імені акціонерів компанії.

2. Визначення вимог та розрахунок рентабельності інвестицій

Обираючи системи управління складськими товарами та торгівлі, встановлення мінімальних вимог, які відповідають потребам конкретної компанії, має бути головним пріоритетом. За даними Inbound Logistics, кожна автоматизована система може містити сотні функціональних рішень, починаючи від управління вхідними вантажами і закінчуючи обслуговуванням клієнтів. Проте, на етапі визначення вимог не потрібно поспішати створювати обширний список усіх вимог до майбутньої системи [9].

Одночасно потрібно визначати витрати, пов'язані з кожною зазначеною вимогою. Це допоможе компанії визначити рентабельність інвестицій, і, розглядаючи конкретного постачальника, менеджери проектів можуть зробити точний прогноз, коли рентабельність інвестицій стане позитивною.

3. Пошук постачальників

На даному етапі, зосередившись на одному-двох найважливіших аспектах зі списку вимог функціонування майбутньої системи, необхідно відібрати якомога

ширший список постачальників, чії можливості відповідають доступним критеріям.

#### 4. Оцінка можливостей постачальників

Необхідно оцінити можливості розвитку та вдосконалення кожного постачальника. Кожен з них повинен відповідати мінімальним вимогам у списку вимог компанії, але вимоги можуть бути змінені. Іншими словами, поточні мінімальні потреби можуть не відповідати потребам майбутнього, тому завжди враховують стан розвитку постачальника автоматизованої системи управління складськими товарами та торгівлі.

#### 5. Вибір 6-10 кращих варіантів постачальників

Для першого етапу звуження списку можливих постачальників, краще не сильно скорочувати їх кількість. Адже на наступних етапах існують свої критерії для їх «відсіювання».

#### 6. Подання запитів постачальникам на розгляд конкретної ринкової ситуації

Подання запитів на пропозиції співпраці дозволить тепер уже постачальникам переглянути системи та потреби компанії, розробити модель витрат і, в свою чергу, скласти план впровадження. Це найкращий спосіб більш детально оцінити постачальників [10].

#### 7. Тестування пробних версій автоматизованих систем управління складськими товарами та торгівлі

Зазвичай у постачальників є безкоштовний пробний період користування програмними продуктами, задля оцінки функціональних можливостей та зручності користування програмами. Проте деякі постачальники можуть запропонувати індивідуальні пропозиції, щоб показати, як саме їх індивідуальні рішення можуть покращити діяльність компанії.

#### 8. Вибір 2-3 кращих варіантів постачальників

На основі попереднього етапу, рекомендується визначитись з декількома найкращими варіантами постачальників, які мають схожі між собою програмні продукти, та перейти до наступного етапу.

#### 9. Проведення юридичного огляду

Юристи мають перевірити всі юридичні документи, а також юридичні обов'язки та права, пов'язані з кожним постачальником, та використання програмного рішення конкретного постачальника. Дуже важливо провести юридичний огляд поза командою проекту, оскільки це усуне будь-які упереджені висновки, що можуть плинати на рішення за або проти конкретного постачальника в процесі вибору [11].

#### 10. Ухвалення остаточного рішення

Остаточне рішення в основному базується на результатах юридичного перегляду. Якщо під час перегляду виникає будь-яка юридична проблема, ідеально вибрати іншого постачальника. Деякі юридичні проблеми не обов'язково вимагають відмови від конкретного постачальника, особливо якщо рентабельність інвестицій значно вища.

#### 11. Зворотній зв'язок з постачальником, навчання співробітників компанії

Навіть коли вибір уже зроблено, зв'язок із іншими постачальниками з попереднього етапу обривати не варто. Існує імовірність, що програмний продукт виявиться не зручним чи занадто складним в опануванні для співробітників компанії. В такому випадку будуть вирішуватись проблеми із розірванням контракту, відшкодуванням та іншими нюансами, в той час як співробітники зможуть приступити до вивчення програмного продукту конкурента-постачальника.

Слід зауважити, що етапи процесу носять рекомендаційний характер, можуть пропускатись чи повторюватись відповідно до вимог та можливостей конкретної компанії [12].

### 1.3 Дослідження переваг для компанії, від використання автоматизованих системи управління

Ефективне управління складом за допомогою автоматизованих систем управління - це більше, ніж прості цифри та дані. Це перехідний етап до надійного та індивідуального набору рішень для конкретних робочих процесів компанії. Автоматизована система зв'язує команди підрозділів компанії за межами ваших

складських стін за допомогою ІТ-технологій, допомагаючи зосередитись на ефективних продажах та отриманні прибутку.

Обравши та впровадивши правильну автоматизовану системи управління складськими товарами та торгівлі, компанія отримує наступні переваги [13]:

1. Оптимізовані процеси роботи складу, в тому числі разом із автоматизованими системами управління транспортними поставками.
2. Ефективний розподіл робочої сили – час, зусилля та ресурси ефективно розподіляються між всіма важливими етапами виробничого чи продовольчого процесу. Для кращого досягнення даного показника, доцільно використовувати і системи керування трудовими ресурсами чи управління проектами.
3. Автономність співробітників за рахунок чітко визначених ролей та обов'язків всередині компанії.
4. Покращення відносин з постачальниками товарів та клієнтами. Ефективно керований склад допомагає скоротити терміни доставки та помилки при виконанні замовлень, а це означає, що більше клієнтів будуть задоволені своїми замовленнями, а це, в свою чергу, означає більший обсяг продажів. Ефективність допомагає і з боку постачальника, оскільки за допомогою системи управління налагоджується весь ланцюг поставок, при цьому постачальники можуть зосереджуватись на наступних поставках, поклавшись на автоматизовані етапи прийому товарів.
5. Скорочені операційні витрати, що розуміють під собою ефективне використання та розподіл швидкопсувних запасів, використання простору та робочої сили, - кожен із яких зменшує витрати, не знижуючи якість, та дозволяє інвестувати їх в інші сфери ведення бізнесу.
6. Автоматизований баланс запасів.
7. Інтеграція ERP (Enterprise Resource Planning з англ. Планування ресурсів підприємства).
8. Покращена надійність та безпека здійснення обрахунків та проведення оплати.

9. Прозорість та наочність завдяки сформованим звітам, графікам та зведеним таблицям.
10. Постійне вдосконалення можливостей компанії, в тому числі за рахунок розширення функціональних можливостей автоматизованих систем управління [14].

#### 1.4 Уточнена постановка задачі

Компанії, що перебувають на стадії розгляду вибору автоматизованої системи управління для ведення їх бізнесу, повинні переконатися, що системи, які розглядаються, мають відповідні можливості для рішення потреб компанії.

Правильний вибір повинен відповідати бізнес-моделі компанії, її культурі та стратегії розвитку, забезпечуючи керівникам інновації, функціональність та універсальність, необхідні для зростання та процвітання [15].

Тому, завданням розробки автоматизованої системи в межах магістерської кваліфікаційної роботи є спроектувати таку автоматизовану систему обліку складських товарів та підтримки торгівлі, яка буде містити ряд головних та допоміжних функцій, що дозволить автоматизувати рутинні завдання та мінімізувати вплив помилок людського фактору на роботу підприємства, задля зменшення втрат та ефективного розподілу вивільнених робочих ресурсів на інші області роботи компанії.

#### 1.5 Висновки до розділу

У першому розділі проаналізовано практичні аспекти використання автоматизованих систем обліку складських товарів і торговельної діяльності, визначено їх основні функціональні можливості та переваги впровадження. Розглянуто етапи вибору програмного забезпечення та обґрунтовано доцільність створення власної автоматизованої системи. На основі проведеного аналізу сформульовано вимоги та уточнено постановку задачі подальшої розробки програмного продукту.

## 2 ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ СКЛАДСЬКИХ ТОВАРІВ ТА ПІДТРИМКИ ТОРГІВЛІ

2.1 Дослідження функціональних можливостей прикладних програм-аналогів систем управління, що забезпечують облік складських товарів

Будь-яка підприємницька діяльність пов'язана з фінансовими ризиками і невиправданими втратами. Однак існують такі грошові втрати, які можна легко передбачити і зробити все можливе, щоб виключити їх. Почати можна з актуалізації даних про залишки складських товарів [16].

Роблячи дозамовлення нових партій товарів, підприємці опираються на низку неактуальних даних, зокрема на відсутність точної інформації про кількість, категорії та локалізацію товарів у торговій мережі. Це унеможлиблює прогнозування розвитку компанії, дохід та маржинальність.

А навпаки, при надлишку товарів на торгових точках, інформації про яких немає в обліковій системі, у бухгалтера будуть з'являтися певні залишки. Це все є помилками в обліку, що тягнуть за собою некоректні дані при закритті періоду, додаткові штрафи і пені.

Якщо співробітник компанії знає, що облікова система не відображає реальних залишків на складі, то будь-яку пропажу завжди можна списати на систему. Отже, відсутність коректного обліку негласно може стимулювати злочинство в компанії.

Існують різні способи вирішення цієї проблеми: наведення порядку на торгових точках, забезпечення належних умов зберігання товарів, налаштування коректного обліку терміну придатності, розробка спеціальної мотивації для комірників або проведення регулярної інвентаризації та звірки залишків.

Регулярна інвентаризація дозволяє керівникам отримувати актуальні дані про наявність товарів у компанії та приймати управлінські рішення на основі цієї інформації.

Автоматизація процесу інвентаризації є першим кроком до мінімізації впливу людського фактора та прискорення обліку товарів на складі. Вона передбачає, насамперед, підрахунок залишків із використанням спеціального обладнання,

такого як термінал збору даних (ТЗД), а також швидкий обмін інформацією між обліковою системою та терміналом [17-19].

Впровадження автоматизованого обліку товарів на складі потребує фінансових вкладень у системи управління, проте ці інвестиції швидко окупаються завдяки скороченню витрат робочого часу персоналу.

На ринку програмного забезпечення існує широкий спектр систем управління складом. Такі компанії, як IBM, Microsoft, Oracle та SAP, пропонують продукти або окремі модулі для обліку, аудиту та звітності складських товарів. Найбільш доступними для українських підприємств є програмні продукти 1С:Підприємство, що включають модуль «Управління торгівлею».

Інструменти управління та аналітики в модулі «Управління торгівлею 8.3» дозволяють ефективно організовувати процеси закупівлі, зберігання та відпуску товарів на складах підприємства [20].

Документ «Замовлення постачальнику» зображено на рисунку 2.1 - реєструє домовленості з постачальником про постачання товарів визначеного складу, цін, терміну їх постачання та умови оплати.

**Заказ поставщику Д000-000103 від 01.04.2012 12:00:00**

Провести та закрыти | Провести | Створити на підставі | Печать | Зачет оплаты | Изменить

Номер: Д000-000103 от: 01.04.2012 12:00:00 Статус: К поступлению Приоритет: Средний

Поставщик: Koenig Holding GmbH Организация: Добро

Контрагент: Koenig Holding GmbH Договор: Контракт на поставку бытовой техники

Соглашение: Поставка бытовой техники Склад: Центральный склад

N	Номенклатура поставщ...	Номенклатура	Характеристика	Количество	Упаковка, Ед. изм.	Цена	% руч.	Сумма руч.	Сумма
1	Комбайн МОУЛ...	Комбайн МОУЛ...	характеристик...	50,000	шт	70,00			3 500,00
2	Комбайн кухонн...	Комбайн кухонн...	характеристик...	50,000	шт	70,00			3 500,00

Итоговая сумма (EUR): Заказано: 36 250,00 НДС: Без НДС Отменено: 0,00

Итоговая скидка (наценка): Всего: 0,00 0 % Ручная: 0,00 0 %

Этапы оплаты (1): Аванс: 0,00 0 % Предоплата: 36 250,00 100 % Кредит: 0,00 0 %

Расчеты по заказу (EUR): Долг: 0,00 0 % Оплачено: 36 250,00 100 % Поступило: 36 250,00 100 %

Текущее состояние: **Готов к закрытию**

Рисунок 2.1 – Документ замовлення постачальнику в 1С:Підприємство

Звіт «Стан забезпечення замовлень» - дозволяє проаналізувати стан забезпечення товарів по замовленнях постачальників.

Обробка «Управління товарами» зображено на рисунку 2.2 - допоможе визначити календарну потребу в товарах та замовити їх у постачальників.

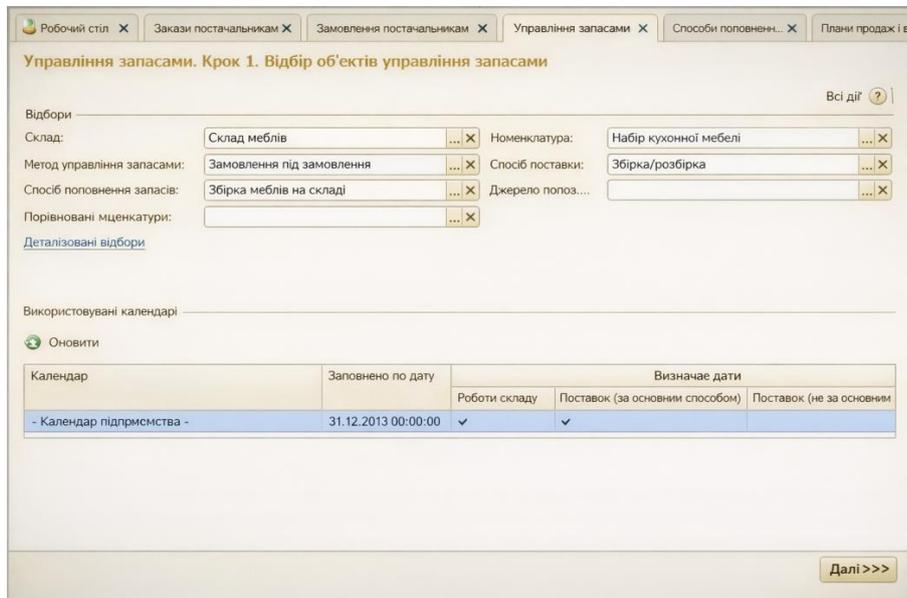


Рисунок 2.2 –Управління товарами в 1С:Підприємство

Програмний продукт «Бухгалтерія 8 для України» призначений для автоматизації бухгалтерського та податкового обліку, а також для формування регламентованої звітності на підприємствах різних сфер діяльності: від оптової та роздрібною торгівлі до виробництва, надання послуг, комісійної та субкомісійної торгівлі.

Система надає широкі можливості бухгалтерським службам і забезпечує керівників інструментами для аналізу діяльності. Вона дозволяє комплексно вирішувати завдання бухгалтерії — від підготовки первинних документів до обліку продажів та інших господарських операцій. При цьому використання програми зосереджене саме на бухгалтерському та податковому обліку; автоматизація інших підрозділів (наприклад, відділу продажів) передбачена через спеціалізовані конфігурації або окремі системи.

«Бухгалтерія 8 для України» повністю відповідає чинному законодавству України. У її склад входить план рахунків, затверджений Наказом Міністерства фінансів від 30 листопада 1999 року № 291 (з урахуванням змін). Структура рахунків, організація аналітичного, валютного та кількісного обліку відповідають

вимогам законодавства щодо ведення бухгалтерського обліку та складання звітності.

Користувач може самостійно створювати додаткові субрахунки й налаштовувати аналітичний облік. При визначенні облікової політики доступно:

вибір схеми оподаткування;

спосіб оцінки запасів і товарів у роздрібній торгівлі;

використання рахунків класів 8 і 9 (разом чи лише одного з них);

налаштування методів виробничого обліку;

встановлення правил нумерації документів тощо [21].

Система підтримує як кількісний, так і кількісно-сумовий складський облік.

У першому випадку оцінка матеріалів та товарів для цілей бухгалтерії не залежить від складу, з якого вони списані. У другому — оцінка ведеться окремо по кожному складу. При необхідності складський облік можна відключити.

У програмі реєструються результати інвентаризацій, які автоматично звіряються з даними системи. На їх основі здійснюється оприбуткування надлишків або списання нестач [20].

Для кожного складу призначається матеріально відповідальна особа. Одночасно може бути закріплений лише один відповідальний співробітник, при цьому історія змін відповідальних зберігається. Інвентаризацію запасів можна проводити як по складу в цілому, так і за конкретною матеріально відповідальною особою.

ABM WMS – це система управління складом, що забезпечує прозорий облік завдяки використанню хмарних технологій. Доступ до неї можливий із будь-якої точки світу через Інтернет [22].

Основні функціональні можливості: Автоматизація складських процесів у ABM WMS дозволяє оптимізувати розміщення товарів із урахуванням їхніх масогабаритних характеристик, застосовувати різні схеми зберігання, скорочувати час на підбір замовлень та проводити інвентаризацію без зупинки роботи складу. Відбір і відвантаження, планування замовлень із контролем партій та термінів придатності відповідно до FIFO / LIFO / FEFO; використання різних технологій складання, пакування, розміщення та навантаження; комплектування

наборів; обробка нестандартних і проблемних ситуацій. Приймання: онлайн-контроль кількості та якості; перепакування, маркування й формування палет; приймання негабаритних вантажів; крос-докінг; ваговий контроль. Зберігання: підтримка різноманітних методів і технологій зберігання, включно з адресним. Топологія складу: автоматичне створення адрес місць зберігання (літерно-цифрові комбінації); використання різних форматів адресних ідентифікаторів; інтеграція з системою адресного зберігання [23].

На рисунках 2.3-2.5 представлено деякі функціональні можливості АВМ WMS.

**Біллінг ресурсів (Основной)**

Сформировать | Выбрать вариант... | Настройки... | Все действия

Период:  Прошлый месяц | Склад:  Склад

Операция над товаром:  Отгрузка упаковки

Параметры: Период: 01.08.2017 - 31.08.2017  
Отбор: Операция над товаром Равно "Отгрузка упаковки" и Склад Равно "Склад"

Ресурс операции Документ	Операция над товаром	Выкл адка с конте йнера	Количество строк	Время начала	Время окончания	Количество единиц	Время на одну строку (мин.)	Время обработк и (мин.)	Кoeffицие нт эффektivно сти	Вес
Ресурс №101			1 308	28.08.2017 17:42:12	29.08.2017 21:09:49	6 986	0,0		258,7	305,753
Движения товаров 1 827 от 28.08.2017 16:46:49	Отгрузка упаковка	Нет	42	28.08.2017 22:55:36	28.08.2017 22:55:39	259	0,0		259,0	9,691
Движения товаров 1 828 от 28.08.2017 16:46:59	Отгрузка упаковка	Нет	173	28.08.2017 20:47:31	28.08.2017 20:47:50	1 096	0,0		1 096,0	29,505
Движения товаров 1 830 от 28.08.2017 16:47:02	Отгрузка упаковка	Нет	23	28.08.2017 18:29:45	28.08.2017 18:29:46	92	0,0		92,0	3,305
Движения товаров 1 833 от 28.08.2017 16:47:09	Отгрузка упаковка	Нет	37	28.08.2017 19:59:35	28.08.2017 19:59:38	139	0,0		139,0	10,517
Движения товаров 1 832 от 28.08.2017 16:47:11	Отгрузка упаковка	Нет	2	28.08.2017 17:42:12	28.08.2017 17:42:12	125	0,0		125,0	1,25
Движения товаров 1 834 от 28.08.2017 16:47:14	Отгрузка упаковка	Нет	86	28.08.2017 19:03:21	28.08.2017 19:03:28	459	0,0		459,0	13,479
Движения товаров 1 851 от 28.08.2017 16:48:38	Отгрузка упаковка	Нет	86	28.08.2017 19:27:11	28.08.2017 19:27:19	617	0,0		617,0	21,11
Движения товаров 1 849 от 28.08.2017 16:48:39	Отгрузка упаковка	Нет	23	28.08.2017 19:40:28	28.08.2017 19:40:30	207	0,0		207,0	10,076
Движения товаров 1 855 от 28.08.2017 16:49:27	Отгрузка упаковка	Нет	22	28.08.2017 23:44:09	28.08.2017 23:44:11	110	0,0		110,0	7,187
Движения товаров 1 865 от 28.08.2017 16:50:48	Отгрузка упаковка	Нет	10	29.08.2017 0:07:56	29.08.2017 0:07:57	21	0,0		21,0	2,889
Движения товаров 1 867 от 28.08.2017 16:50:55	Отгрузка упаковка	Нет	8	29.08.2017 0:50:51	29.08.2017 0:50:51	29	0,0		29,0	2,349
Движения товаров 1 885 от 28.08.2017 16:54:24	Отгрузка упаковка	Нет	38	28.08.2017 19:51:53	28.08.2017 19:51:57	204	0,0		204,0	12,234
Движения товаров 1 896 от 28.08.2017 18:12:46	Отгрузка упаковка	Нет	111	29.08.2017 0:46:21	29.08.2017 0:46:31	233	0,0		233,0	22,115
Движения товаров 1 908 от 28.08.2017 18:13:48	Отгрузка упаковка	Нет	57	28.08.2017 22:31:20	28.08.2017 22:31:23	305	0,0		305,0	14,501
Движения товаров 1 915 от 28.08.2017 18:13:57	Отгрузка упаковка	Нет	32	29.08.2017 1:16:56	29.08.2017 1:16:58	582	0,0		582,0	6,452

Рисунок 2.3 – Білінг ресурсів в АВМ WMS

**ABC Анализ**

Сформировать | Выбрать вариант... | Настройки... | Все действия

Период:  Этот месяц | Объем класса A:  70

Операция над товаром:  Отгрузка отбор | Расчетный Класс ABC:  Не равно | НоменклатураОстатка.КлассОборачиваемостиТовара

Объем класса B:  20

Параметры: Период: 01.09.2017 - 30.09.2017  
Операция над товаром: Отгрузка отбор  
Объем класса B: 20  
Объем класса A: 70

№ п/п	Номенклатура	Текущий класс ABC	Количество операций	Расчетный Класс ABC
2 289	МИДОКАЛМ табл. 150мг №30	A	13	B
2 290	ГЕКСИЖОН супл.ваг.16мг №10	A	13	B
2 291	МЕВЕРИН капс.0,2r №30	A	13	B
2 292	ПРЕДИКАРБ-Д мазь 15 г туба	A	13	B
2 293	ТЕРИДИАН ваг. табл. №10	A	13	B
2 294	ПШИК СПРЕЙ наз Гипертон 2,1% 100мл	A	13	B
2 295	Н-КА ЭВКАЛИПТА фл. 25мл (Житом.ФФ)	A	13	B
2 296	РЕОПОЛИГЛЮСКИН фл. 200 мл	A	13	B
2 297	НОРМОВЕН таб.№60	A	13	B
2 298	СТОМАТОЛИТ р-р 100 мл	A	13	B
2 299	Пр-ва ВИА №12 ультраонки	A	13	B
2 300	ДОПОВЕНЕ гель 20г	A	13	B
2 301	PR ALL. DISCREET Deo Breeze №20	A	13	B
2 302	Злцетка Colgate Classic Plus Medium 1+1	A	13	B
2 303	ЗОЛМИГРЕН табл.№10	A	13	B
2 304	ТАУРИН БИОАКТИВ капс №50 (Злит-Фарм)	A	13	B
2 305	БИСКОДИЛ свечи 0,01 г №10	A	13	B
2 306	ОКСИТОЦИН амл. 1 мл (5 Ед.) №10	A	13	B

Рисунок 2.4 – Аналіз в АВМ WMS

Инвентаризация 1 054 от 07.09.2017 9:30:52

Закрепить и закрыть | Начать инвентаризацию | Закончить инвентаризацию | Печать

Номер: 1 054 | Дата: 07.09.2017 9:30:52 | Склад: Склад | Статус документа: Завершена

Вариант проведения инвентаризации: Корректировать остатки при записи документа

Фильтр товар: | Фильтр категория: | Фильтр ячейка: | Фильтр этаж: 0

Заполнить по фильтру | Комментарий: рома

Ячейка	Ресурс	Артикул	Номенклатура	Учетный остаток	Количество
P.11.4.1.1	Ресурс №168	24287	МАРЛЕВЫЙ ОТПРЕЗ 10м*90см (Белоснежка) тип 17	60	61
P.11.4.2.1	Ресурс №168				
P.11.4.2.2	Ресурс №168				

Серия	Срок годности	Учетный остаток	Количество

Рисунок 2.5 – Процес інвентаризації в АВМ WMS

## Результати впровадження WMS

Підвищення ефективності управління складським комплексом досягається завдяки комплексному підходу, що поєднує автоматизацію процесів, контроль і аналітику бізнес-операцій. Зокрема, використання WMS забезпечує:

- контроль залишків, руху товарів, завантаженості персоналу та складських площ;
- оптимальне розміщення продукції, зонування та раціональне використання потужностей складу;
- збільшення пропускної здатності;
- скорочення витрат на зберігання та зменшення збитків від псування чи неправильного зберігання;
- зниження впливу людського фактора й підвищення якості роботи персоналу;
- оптимізацію переміщення техніки та співробітників складу, мінімізацію простоїв;
- можливість управління територіально розподіленими складами;
- структурованість процесів приймання та відвантаження із дотриманням термінів постачання;
- інструменти для виявлення «вузьких місць», пошуку шляхів оптимізації та аналізу ефективності складської діяльності [24, 25].

- ABM WMS є хмарним рішенням, що функціонує за моделлю SaaS (Software as a Service). Це означає, що користувач не несе додаткових витрат на встановлення серверів чи обслуговування системи. Рішення легко інтегрується з ERP та іншими обліковими системами, а його успішне застосування підтверджене як у великих міжнародних корпораціях, так і в локальних компаніях.

#### Класифікація WMS-систем

- WMS – окремий клас програмних рішень, що тісно інтегрується з ERP-системами.

- WMS початкового рівня – для невеликих складів із обмеженим асортиментом.

- Коробкові рішення – підходять для складів із малим товарообігом.

- Гнучкі адаптовані системи, такі як ABM WMS, призначені для розподільчих центрів та складів із великим асортиментом і значними обсягами товарообігу [26].

EFSOL Автоматизація складу – це дієвий інструмент для оптимізації складських процесів, підвищення рівня керованості та забезпечення повного контролю за обліком товарів і матеріалів у компанії. Рішення дозволяє своєчасно отримувати актуальні дані про залишки на складі, ефективно планувати закупівлі з урахуванням попиту клієнтів і можливостей постачальників, а також знижувати вплив людського фактора, скорочуючи кількість помилок і втрат [27].

#### Результати впровадження EFSOL Автоматизація складу:

- Збільшення швидкості роботи складу і оптимізація площі зберігання.

- Можливість надавати послуги відповідального зберігання - додатковий дохід для компанії.

- Економія трудовитрат за рахунок скорочення ручного обліку та формування документації.

- Зниження пересортування на 99%.

- Систематичний контроль та аналіз залишків, надходження, руху обороту і відвантаження товару.

- Скорочення втрат підприємства: мінімізація злодійства, втрат від прострочених товарів.

- Контроль персоналу, скорочення витрат на його утримання.

- Більш ефективне використання складського обладнання [30].

В основі комплексної автоматизації складу лежить системний підхід до вирішення бізнес-завдань компанії-замовника. Нижче наведені основні етапи автоматизації [28].

#### 1. Підготовка:

- Обстеження, аналіз поточної системи обліку;
- Намальований в графічній формі і описаний процес «продаж»;
- Функціональна архітектура рішення спроектована і узгоджена;
- Сформовано план проекту (по розробці і установці) в розрізі етапів, завдань, результатів, термінів, вартості ctm. Визначено межі проекту;
- Розроблено вимоги і пропозиції до іт інфраструктури.

#### 2. Впровадження:

- Установка і настройка програми;
- Підключення та тестування обладнання.

#### 3. Підтримка:

- Навчання користувачів;
- Обслуговування облікової системи;
- Розвиток системи обліку.

На рисунку 2.6 представлено деякі функціональні можливості ABM WMS.

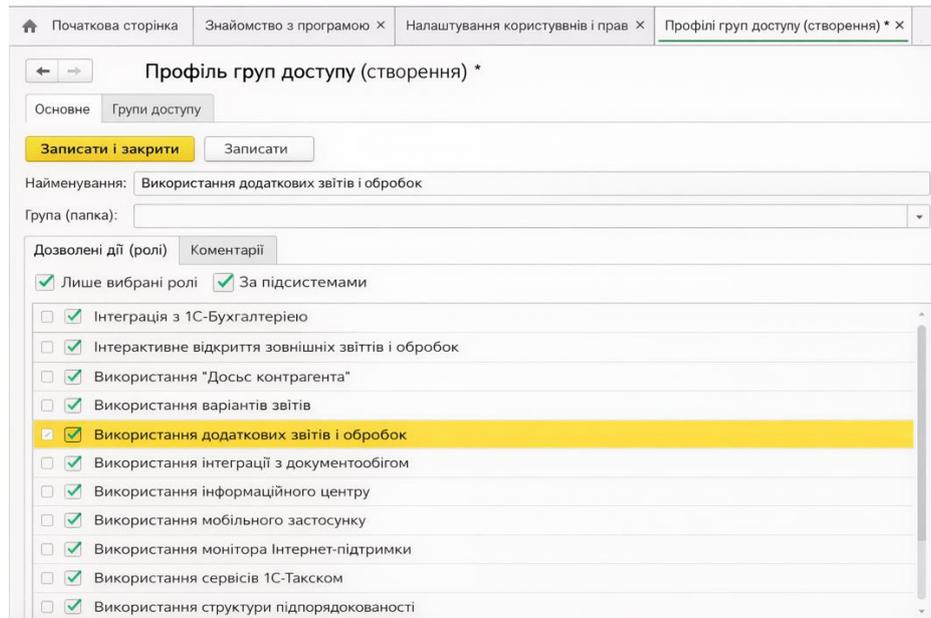


Рисунок 2.6 – Процес вибору групи доступу до товарів в EFSOL Автоматизація складу

Програма автоматизації торгівлі **GBS.MARKET** - просте програмне рішення для автоматизації торгівлі із можливістю ведення звітності та обліку складських товарів.

Переваги GBS.Market, що робить її однією з популярних програм для автоматизації торгівлі для малого бізнесу [29]:

- GBS.Market проста у використанні і не вимагає попереднього навчання співробітників або знання бухгалтерського обліку. Інтерфейс програми інтуїтивно зрозумілий і після установки можна одразу перейти

- и до роботи.

- Доступна вартість ліцензії. Протягом 30 днів після установки є можливість використовувати програму безкоштовно. При бажанні, можна оплатити безстрокову ліцензію.

- Постійні оновлення та доопрацювання.

- Функціональність. Незважаючи на невисоку вартість і простоту, програма готова до роботи в практично будь-якому напрямку роздрібної торгівлі. Підтримка торгового обладнання, віддалений контроль, права доступу і багато іншого є серед функціональних можливостей даного продукту.

○ Робота з асортиментом: додавання товарів вручну або через Excel, ведення обліку продажів за готівку, банківською картою чи в кредит, формування складених товарів та комплектів.

○ Аналітика та звітність: статистика продажів із можливістю формування звітів і діаграм за будь-який період (день, місяць, категорія). Сортування результатів за часом, найменуванням чи вартістю. Доступна функція друку та експорту звітів у Excel.

○ Віддалений контроль: управління необмеженою кількістю магазинів із дому, перегляд статистики в реальному часі, отримання звітів на електронну пошту з обраною періодичністю.

- Підтримка обладнання: інтеграція зі сканерами штрих-кодів, принтерами чеків і етикеток, а також з фіскальними реєстраторами [30–31].

На рисунках 2.8-2.10 представлено деякі функціональні можливості GBS.Market.

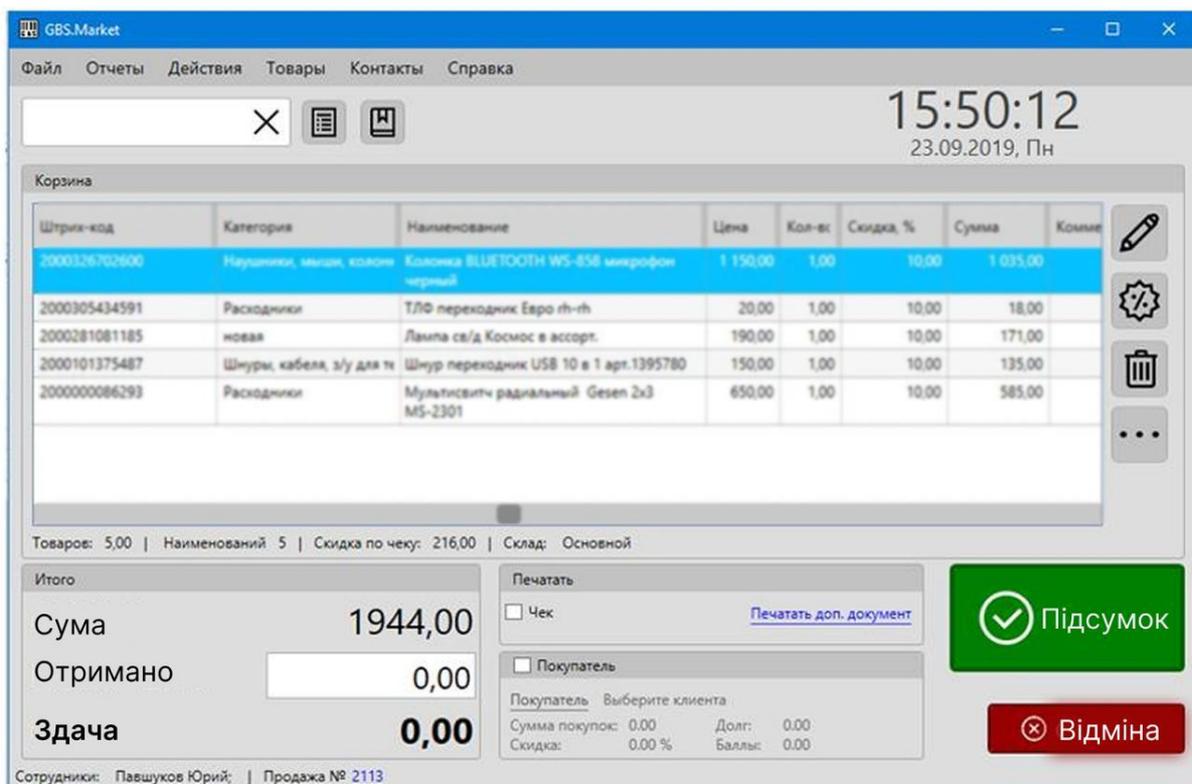


Рисунок 2.8 – Оформлення замовлення в системі GBS.Market

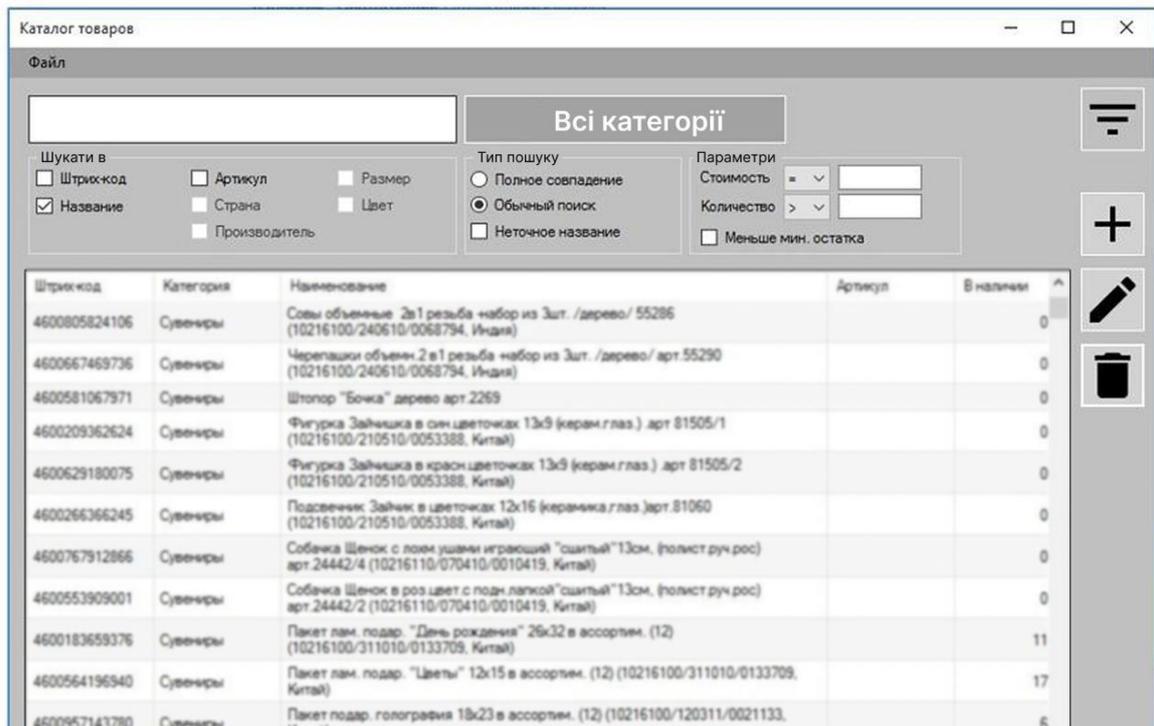


Рисунок 2.9 – Каталог товарів в системі GBS.Market

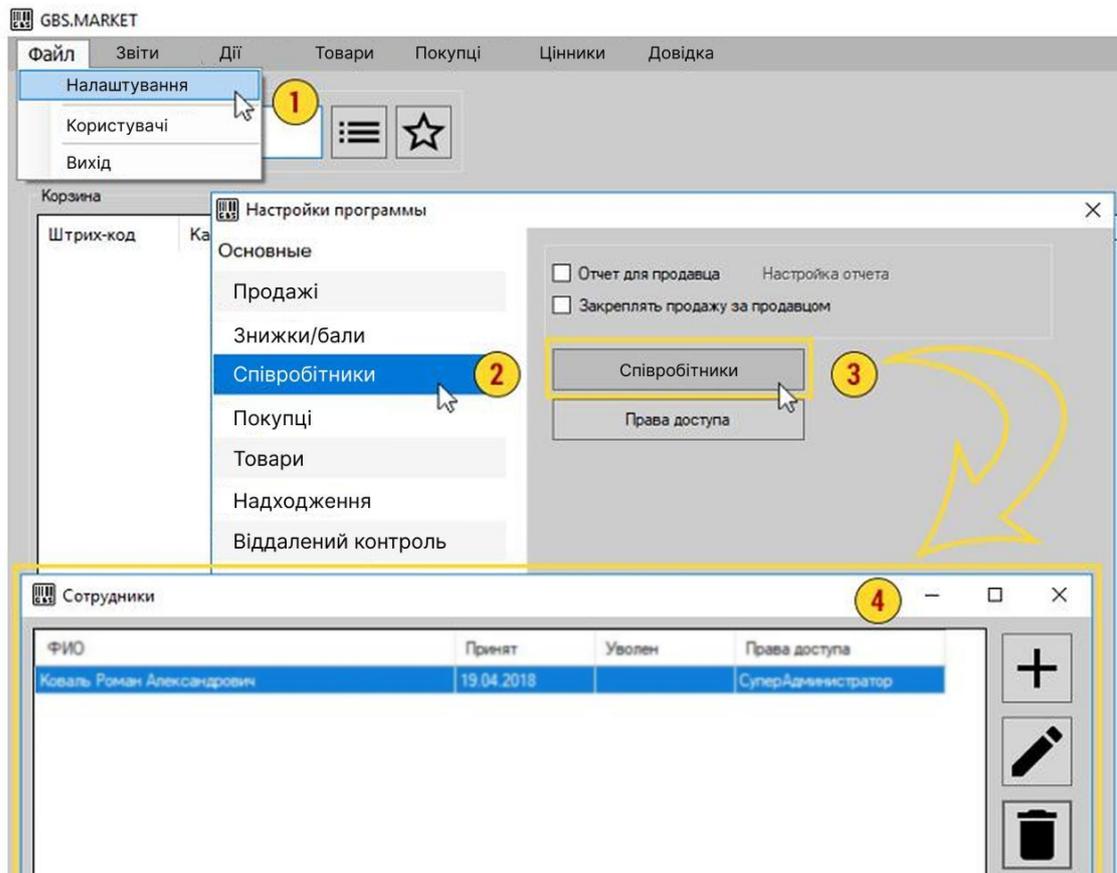


Рисунок 2.10 – Налаштування даних про співробітника компанії в системі GBS.Market

Програма для обліку торгівлі **UNIPRO** підходить для автоматизації таких бізнесів:

- Рітейл: магазини, бутіки, продуктові маркети, гастрономи, роздрібні мережі;
- HoReCa: кафе, бари, ресторани;
- Фаст-фуд: стріт-фуд, бістро, мобільні кафе;
- Склади: роздрібні, оптові, виробничі [32].

Unipro дозволяє автоматизувати наступні робочі місця:

- |                                  |                                |
|----------------------------------|--------------------------------|
| - касир;                         | - офіціант;                    |
| - комірник;                      | - бармен;                      |
| - товарознавець;                 | - адміністратор торгового залу |
| - бухгалтер;                     |                                |
| - директор магазину / ресторану; |                                |
| - власник бізнесу.               |                                |

У програмі Unipro доступні такі документи:

- |                               |                              |
|-------------------------------|------------------------------|
| - замовлення постачальників;  | - видатковий касовий ордер / |
| - прибуткова накладна;        | витратне платіжне            |
| - поворотна накладна;         | доручення;                   |
| - замовлення покупця;         | - накладна на переміщення;   |
| - рахунок-фактура на оплату;  | - внесення залишків;         |
| - видаткова накладна;         | - акт списання;              |
| - повернення від покупця;     | - акт інвентаризації;        |
| - прибутковий касовий ордер / | - екран касира / офіціанта / |
| вхідне платіжне доручення;    | бармена;                     |
|                               | - виробництво / рецепти.     |

Список функціональних можливостей, що доступні в меню програмного продукту, представлений на рисунку 2.11.

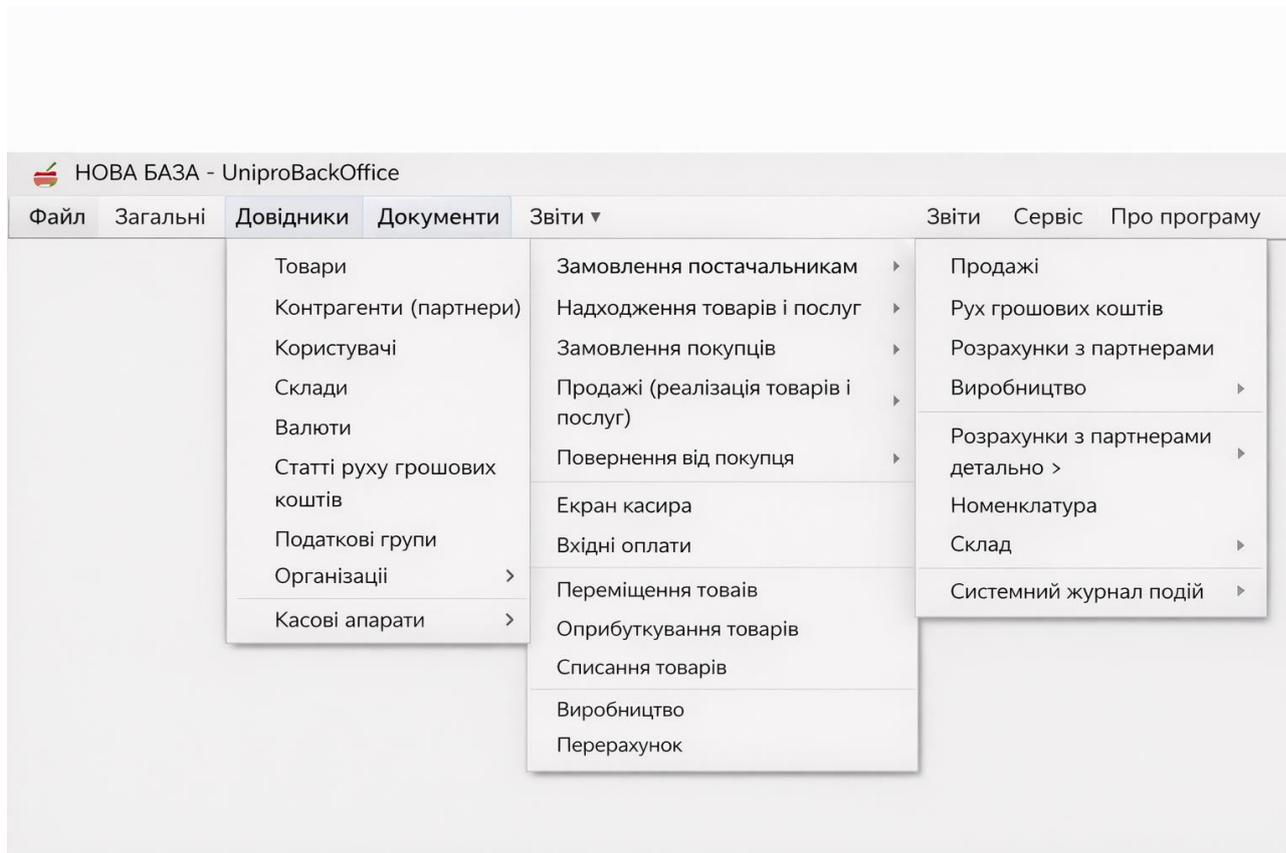


Рисунок 2.11 – Функціональні можливості програма для обліку торгівлі  
UNIPRO

Доступ до системи UNIPRO для кожного користувача здійснюється за допомогою унікального пароля. Це забезпечує надійний захист бази даних від можливих помилок чи зловживань, сприяє узгодженості дій між користувачами, упорядковує та підвищує прозорість бізнес-процесів компанії [33].

Приклад налаштування прав доступу користувачів у системі UNIPRO наведено на рисунку 2.12.

Пользователь	Группа пользователя	Права доступа	Разрешено
1 Admin(пароль: 1)	Полные права	<b>Валюты (UniCurrencies)</b>	<input checked="" type="checkbox"/>
2 Бухгалтер (пароль: 2)	Администратор/Товаровед	Просмотр списка	<input checked="" type="checkbox"/>
3 Кассир1	Кассир/Официант	Создание	<input checked="" type="checkbox"/>
4 Кассир2	Кассир/Официант	Удаление	<input checked="" type="checkbox"/>
5 Кассир3	Кассир/Официант	Изменение	<input checked="" type="checkbox"/>
		<b>Заказы покупателей (UniCustomer0...</b>	<input checked="" type="checkbox"/>
		Просмотр списка	<input checked="" type="checkbox"/>
		Создание	<input checked="" type="checkbox"/>
		Удаление	<input checked="" type="checkbox"/>
		Изменение	<input checked="" type="checkbox"/>
		Печать	<input checked="" type="checkbox"/>
		<b>Кассовые аппараты (UniEKKA)</b>	<input checked="" type="checkbox"/>
		Просмотр списка КА	<input checked="" type="checkbox"/>

Рисунок 2.12 - Налаштування прав доступу користувачів до системи UNIPRO

Програма автоматично веде лог-файл всіх дій користувачів з детальним описом внесених змін. У журналі подій користувач має можливість використовувати різні параметри та фільтри для більш зручної роботи, а також друкувати та експортувати дані в Excel-форматі.

Цей журнал подій дозволяє знайти помилку, зроблену користувачем, знайти не чесного співробітника, а також викоринити бажання персоналу шахраювати і займатися крадіжкою. Дерево зв'язків для кожного документа, що відображається в системі, зображено на рисунку 2.13.

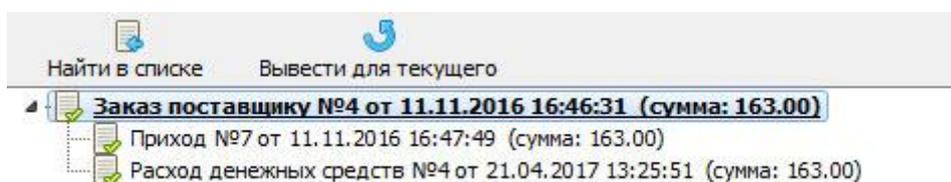


Рисунок 2.13 - Дерево зв'язків для кожного документа

Система UNIPRO підтримує імпорт каталогів товарів, а також документів з EXCEL, тому завантаження цих даних в програму обліку займе максимум 1 хвилину.

Завантаження накладних дозволяє істотно скоротити час введення приходу в базу, а також зменшити кількість помилок, які може зробити оператор при введенні даних [34].

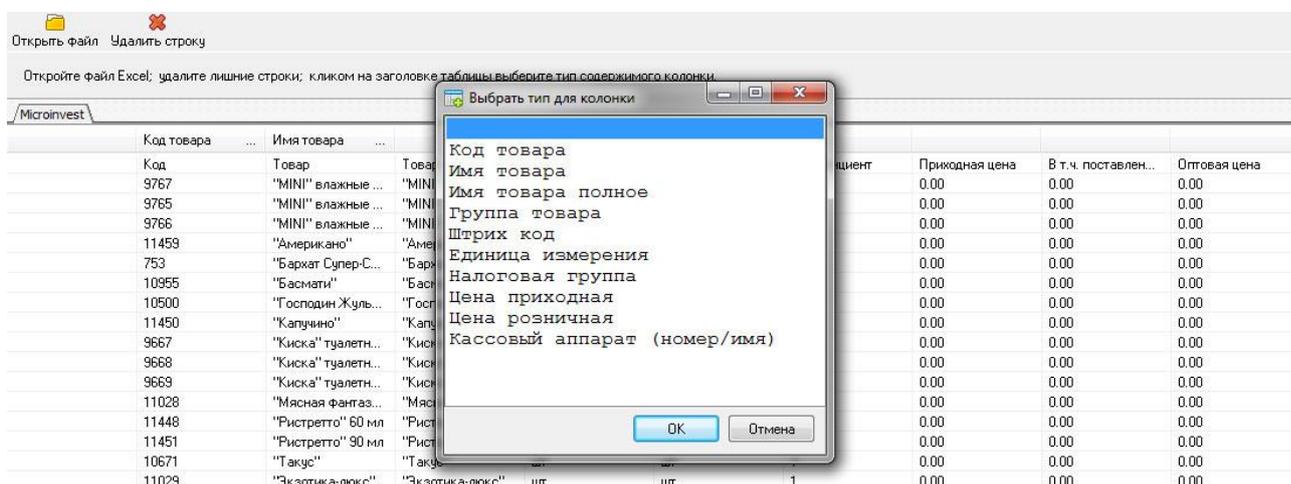


Рисунок 2.14 - Завантаження накладних до система UNIPRO

На рисунку 2.15 зображено екран касира, що керується системою UNIPRO.

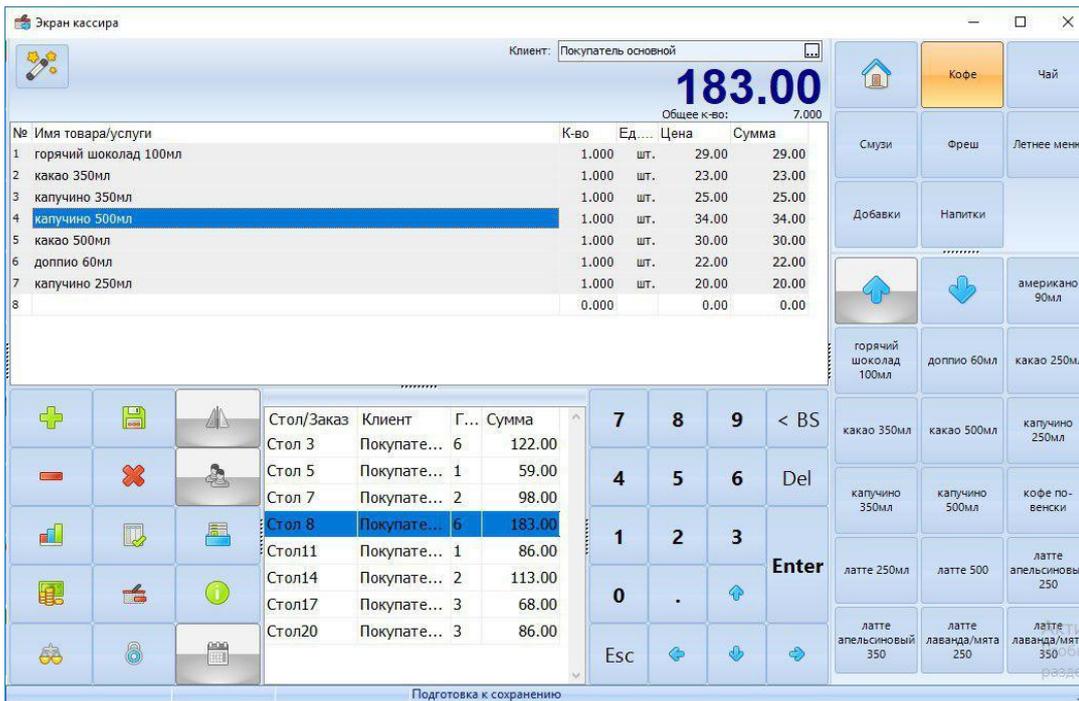


Рисунок 2.15 - Экран касира, що керується системою UNIPRO

На рисунку 2.16 представлено відображення детального звіту з продажів в системі UNIPRO.

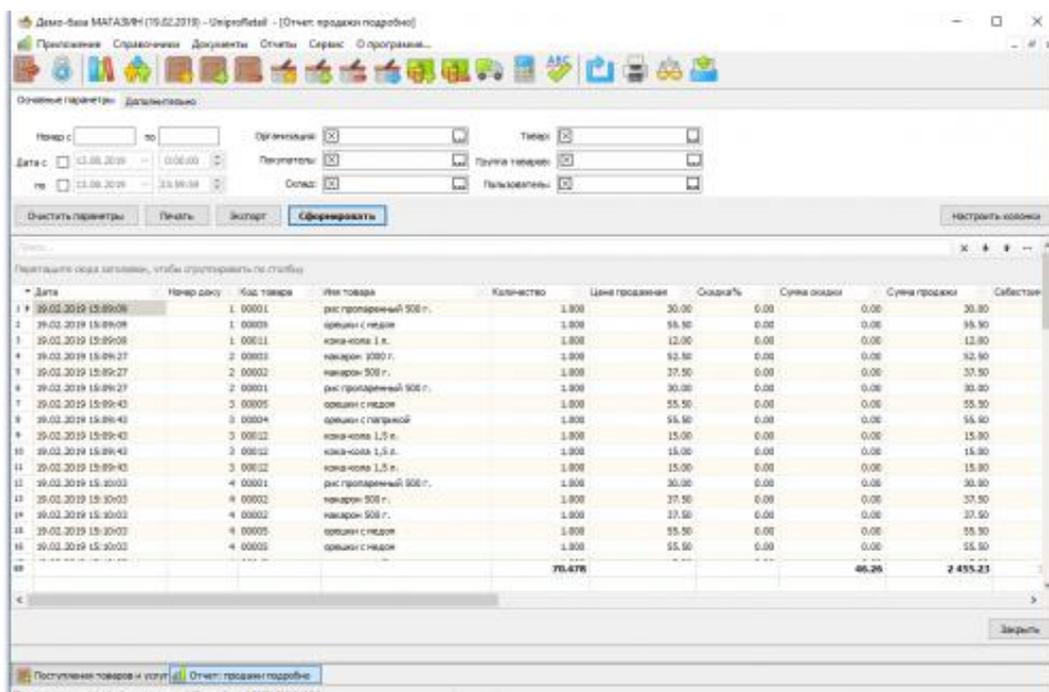


Рисунок 2.16 - Відображення детального звіту з продажів в системі UNIPRO

На рисунку 2.17 представлено вікно заповнення інформації про касовий апарат компанії в системі UNIPRO.

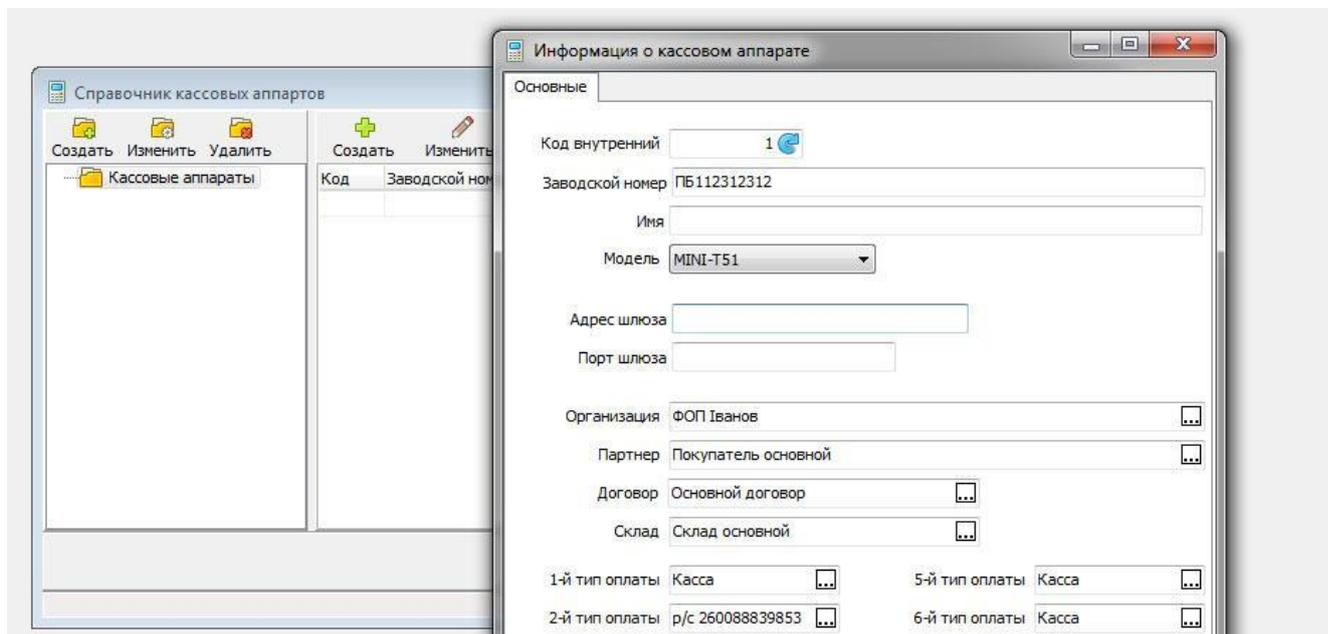


Рисунок 2.17 - Вікно заповнення інформації про касовий апарат компанії в системі UNIPRO

Модель розгортання системи на місці, тобто використання прикладних програм, є все ж розповсюдженим рішенням серед середнього та великого бізнесу, яким важлива стабільність та безпека роботи усієї торгової мережі.

## 2.2 Визначення функціональних можливостей автоматизованої системи, що розробляється

Дослідивши функціональні можливості програм аналогів автоматизованих систем управління складськими товарами та торгівлі, складемо список функцій, які буде реалізовувати програмний комплекс, що розробляється. Система буде призначена для автоматизації діяльності невеликого магазину чи торгової точки та буде здатна:

- реєструвати торгові точки та склади, використовуючи користувацьку топологію мережі;
- вести облік покупців та постачальників;
- реєструвати прихід та відбування товару зі складу;

- відслідковувати залишки товарів;
- реєструвати фінансові операції;
- вести історію фінансових операцій;
- використовувати гнучку систему знижок;
- надавати звіти на основі даних таблиць бази даних;
- виконувати імпорт та експорт даних;
- автоматично виконувати резервне копіювання;
- система повинна бути захищеною та надійною.

Відповідно до цього в рамках комплексної магістерської кваліфікаційної роботи «Розробка автоматизованої системи обліку складських товарів та підтримки торгівлі. Частина 1. Розробка бази даних і десктопного додатку» буде спроектовано та розроблено структуру бази даних системи, UML-діаграми функціонування та безпосередньо саме програмне рішення автоматизованої системи.

### 2.3 Проектування бази даних системи

Процес створення бази даних (БД) повинен враховувати етапи життєвого циклу БД, що представлені на рисунку 2.17, та враховувати дві головні фази – проектування БД та її реалізація [36-38].

Фазу проектування поділяють на наступні етапи:

- визначення стратегії;
- аналіз предметної області застосування;
- концептуальне моделювання;
- логічне й фізичне проектування БД.

Фаза реалізації складається з таких пунктів:

- програмна реалізація;
- документування;
- впровадження.

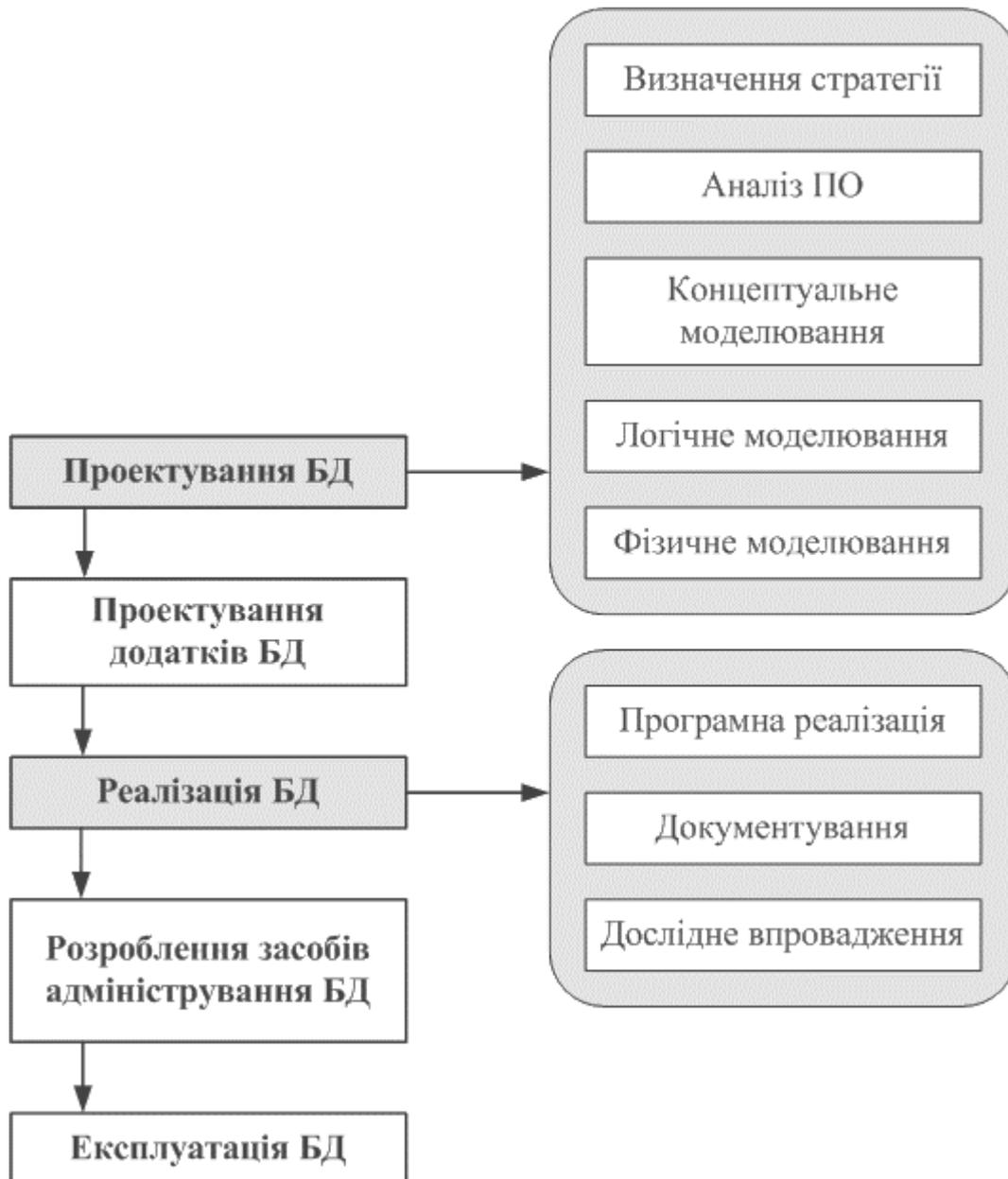


Рисунок 2.17 – Етапи життєвого циклу БД

Визначення предметної області та стратегій використання автоматизованої системи було проаналізовано у розділах вище. Тому можна перейти до етапів концептуального моделювання та проектування БД [39].

На етапі концептуального проектування бази даних системи створюється концептуальна модель даних, виходячи із уявлень користувачів про область застосування програми, на базі чого, розроблюється формальний опис її інформаційного забезпечення.

На рисунку 2.18 зображено ER-модель бази даних на етапі визначення сутностей та зв'язків між ними.

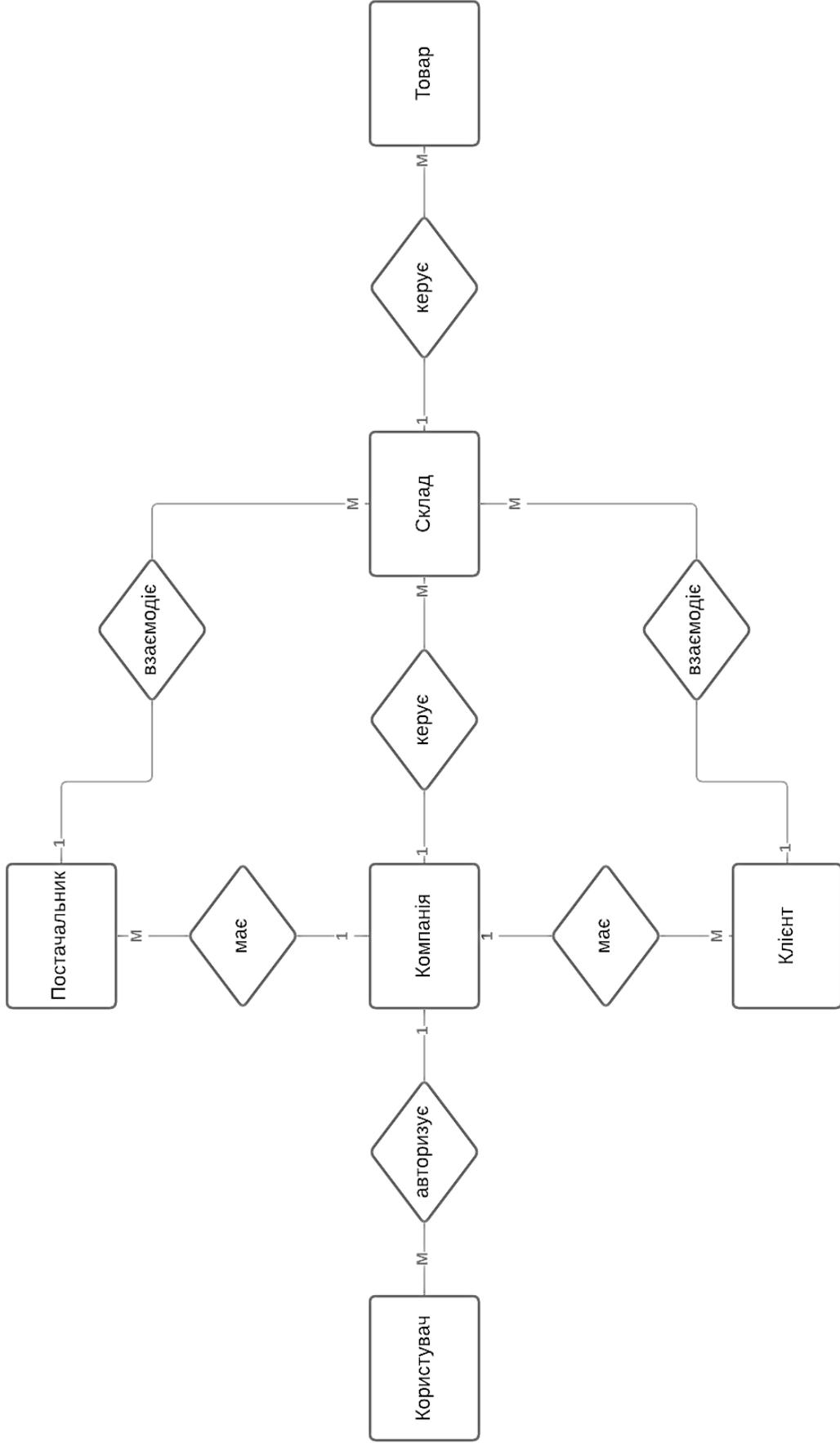


Рисунок 2.18 – ER-модель БД

Визначимо атрибути для кожної із сутностей, інформацію представимо у вигляді зведеної таблиці 2.1.

Таблиця 2.1 – Атрибути сутностей БД

Сутність	Первинний ключ	Атрибути
Торгова точка	Код торгової точки	- Назва торгової точки - Адреса - Телефон
Користувач	Id користувача	- Ім'я користувача - Логін / e-mail - Пароль - Телефон
Клієнт	Id клієнта	- Ім'я клієнта - Телефон - Номер дисконтної карти
Постачальник	Id постачальника	- Ім'я постачальника - Телефон - e-mail - Адреса
Склад	Id складу	- Назва складу - Адреса - Телефон
Товар	Код товару	- Назва товару - Кількість - Характеристики

На рисунку 2.19 представлено структурну схему бази даних, що роз'яснює зв'язки між сутностями та містить перелік атрибутів, необхідних для утворення цих зв'язків.



Сутності представляються у вигляді таблиць бази даних, де атрибути являються полями [40, 41]. Так, для функціонування автоматизованої системи обліку складських товарів та підтримки торгівлі, що розробляється в межах магістерської кваліфікаційної роботи, необхідно створити наступні таблиці в базі даних:

- Company – інформація про компанію, під іменем якої працюють усі магазини та склади мережі.

- User – інформація про адміністраторів чи продавців магазину, в яких є доступ до системи.

- Warehouse – інформація про торгові точки чи склади компанії.

- Order – інформація про замовлення всередині конкретної компанії.

- Delivery – інформація про поставки товарів всередині конкретної компанії.

- Client – інформація про клієнтів компанії.

- Vendor – інформація про постачальників товарів.

- Product – власне інформація про самі товари, що потрапляють на склади чи торгові точки конкретної компанії.

В процесі проектування бази даних було проведено нормалізацію, тобто розділення даних по окремих зв'язаних таблицях, що допоможе уникнути надлишковості даних, дублювання та порушення їх цілісності при редагуванні [42].

Для реалізації розроблюваного програмного забезпечення було обрано систему управління базами даних Firebird. Вона повністю сумісна зі стандартом ANSI у синтаксисі мови SQL та підтримує роботу під різними операційними системами, зокрема Windows, Linux, MacOS, Solaris і низкою Unix-платформ.

Firebird є безкоштовним рішенням, яке не потребує реєстрації чи оплати за технічну підтримку. Завдяки відкритому вихідному коду кожен користувач має можливість створювати на її основі власні некомерційні проекти.

## 2.4 Розробка UML-діаграм

Більшість сучасних методів об'єктно-орієнтованого проєктування засновані на використанні мови UML, що використовується для аналізу предметних областей, існуючих систем, моделювання систем документування об'єктних моделей та проєктування програмного забезпечення [43].

Мова UML дозволяє:

виконувати ієрархічний опис складних систем шляхом поділу їх на пакети;  
формалізувати функціональні вимоги за допомогою діаграм варіантів використання;

деталізувати вимоги через побудову діаграм діяльності та сценаріїв;  
виділяти класи даних і формувати концептуальну модель у вигляді діаграм класів;

визначати класи, що відповідають за користувацький інтерфейс, та створювати схеми навігації між екранами;

описувати взаємодію об'єктів під час виконання системних функцій;  
моделювати поведінку об'єктів за допомогою діаграм діяльності та станів;  
описувати програмні компоненти та їхню взаємодію через інтерфейси;  
представляти фізичну архітектуру системи.

Для опису автоматизованої системи обліку складських товарів і підтримки торгівлі, що розробляється, було створено UML-діаграму варіантів використання та UML-діаграму діяльності [44].

### 2.4.1 Розробка UML-діаграми варіантів використання

UML-діаграма варіантів використання – це письмовий опис того, як користувачі будуть виконувати дії у програмному забезпеченні, що проєктується. З точки зору користувача, діаграма визначає поведінку системи під час реагування на запит. Кожен випадок використання представлений у вигляді послідовності простих кроків, починаючи з мети користувача і закінчуючи етапом, коли ця мета виконана [45].

В діаграму варіантів використання включаються інформацію про:

- те, хто користується програмним засобом;
- те, які дії може виконати користувач;
- мету використання;
- кроки, що проходить користувач для виконання певного завдання;
- те, як програмний засіб повинен реагувати на дії користувача.

Діаграма не покриває інформацію про:

- мову реалізації програмного засобу;
- помилки, що можуть виникати під час використання;
- користувальницькі інтерфейси або екрани [43].

Діаграма випадків використання створюється за допомогою уніфікованої мови моделювання, тому:

- кожен крок (варіант використання) представлений своєю назвою в овалі;
- кожен актор, схематично представлений у вигляді людини з підписаною нижче її роллю в системі;
- кожна дія, позначена лінією між актором та кроком – відношення між ними. Лінії мають різний вигляд та значення в залежності від типу відношення;
- системні межі, позначені прямокутником навколо діаграми варіантів використання [44, 46].

UML-діаграма варіантів використання, що була розроблена для автоматизованої системи обліку складських товарів та підтримки торгівлі зображено на рисунку 2.20.

На діаграмі актор Користувач має п'ять головних варіантів використання, що глибше розписані на діаграмі: Авторизація, Управління списком контрагентів, Управління товарними запасами, Обслуговування клієнтів та управління документами.

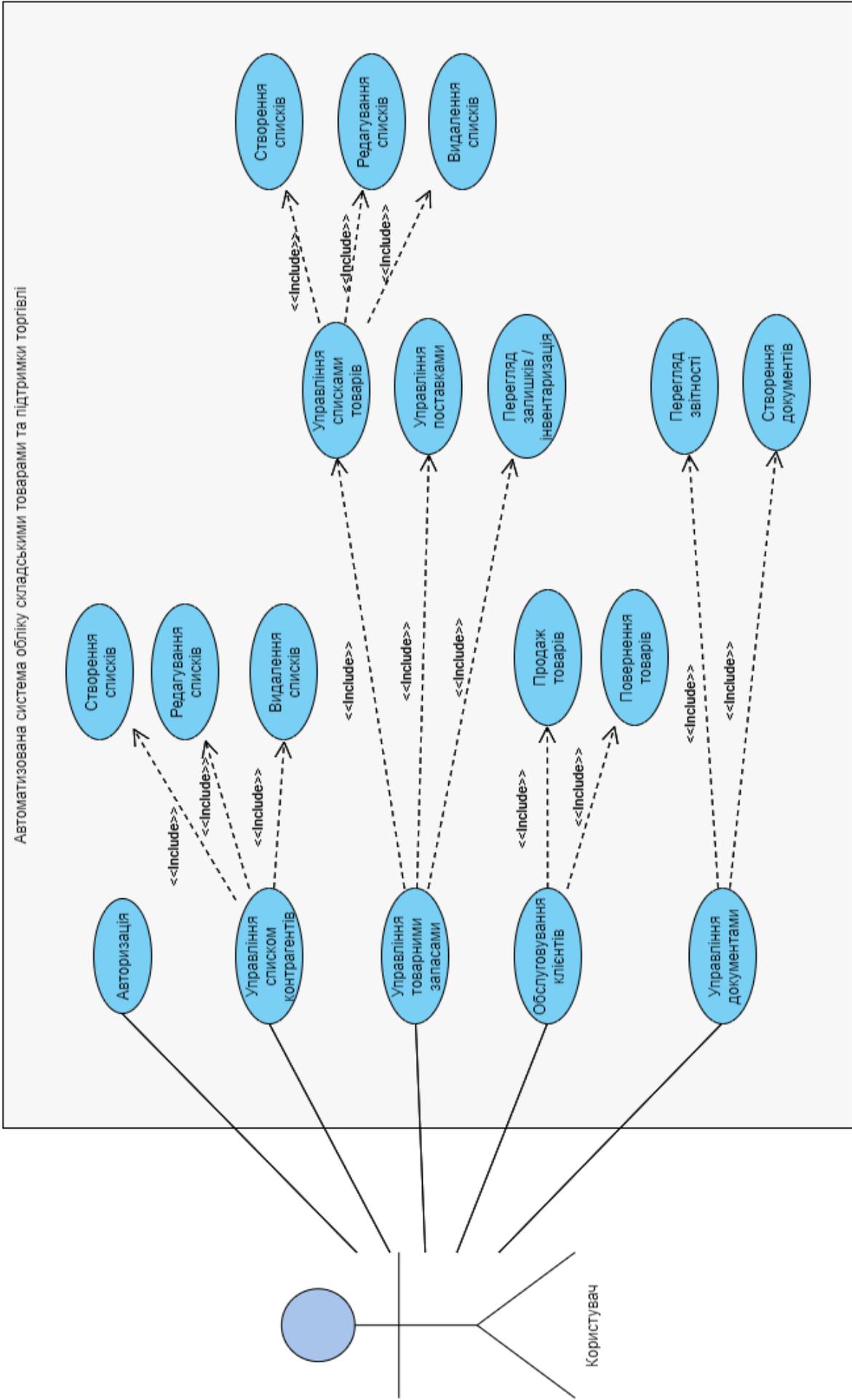


Рисунок 2.20 - UML-діаграма варіантів використання

## 2.4.2 Розробка UML-діаграми діяльності

UML-діаграма діяльності (або ще називають об'єктно-орієнтованою блок-схемою) візуально представляє собою ряд дій або потоку управління в системі, подібно до блок-схеми або схеми потоку даних. Діаграми діяльності часто використовуються при моделюванні бізнес-процесів та робочих процесів. Вони також можуть описати окремі кроки з UML-діаграми варіантів використання [47].

Суть побудови діаграми діяльності зосереджена на передачі певного аспекту динамічної поведінки системи. Діаграми діяльності фіксують динамічні елементи системи.

На рисунку 2.21 зображена UML-діаграма, що представляє діяльність із здійснення покупки товарів покупцем магазину.

Моделювані дії можуть бути послідовними та одночасними. В обох випадках діаграма діяльності матиме початок (початковий стан) і кінець (кінцевий стан).

На діаграмі діяльності вказано п'ять видів діяльності. Після отримання запиту на покупку товарів, продавець перевіряє залишки товару, після чого йде перевірка двох умов [залишки відповідають запрошеній кількості] чи [не відповідають запрошеній кількості]. В залежності від виконання процесу, діяльність переходить до одного з двох описаних шляхів – запит на упаковку товарів з подальшою доставкою товарів чи запит на заміну товарів, відповідно. Після виконання діяльності процес завершується на вузлі завершення.

Створення UML-діаграм є необхідною умовою для циклу розробки бізнес-систем чи програмних продуктів, тому що допомагає уникнути помилок, що виникають при проектуванні, які в свою чергу спричиняють появу нових помилок у наступних фазах розробки системи.

UML-діаграми будуються за допомогою уніфікованої мови моделювання, яка є зрозумілою в колі усієї команди проектувальників та розробників, тому визначені в діаграмах вимоги легко перетворити в комп'ютерну модель [48].

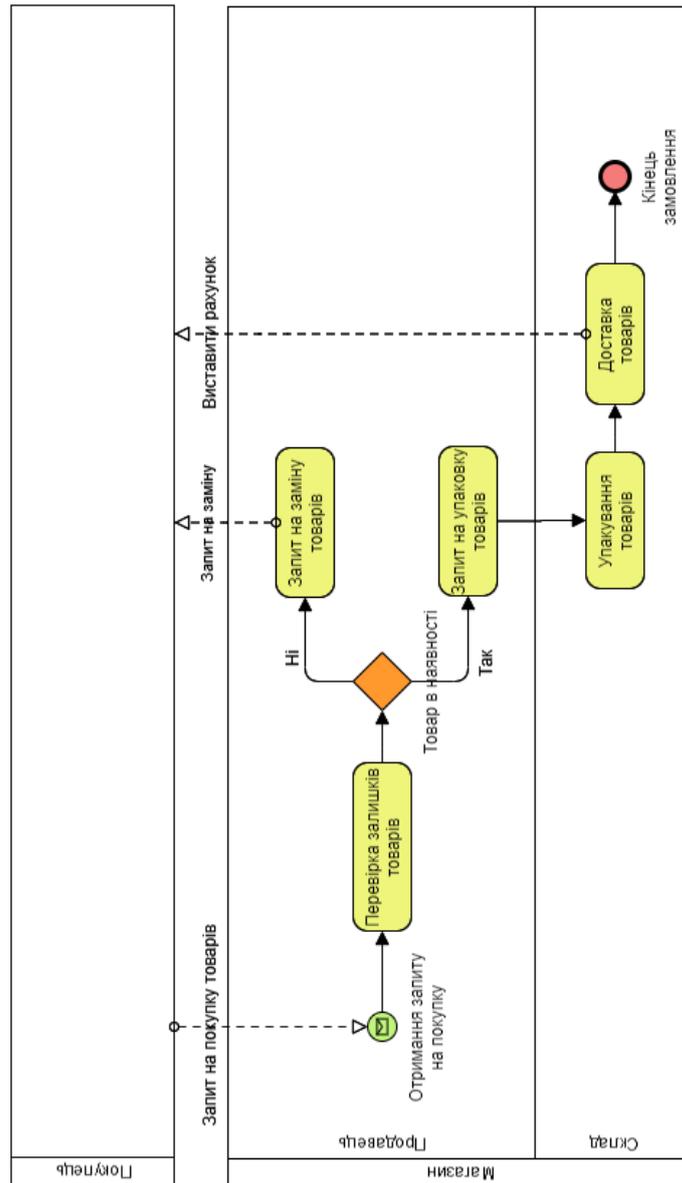


Рисунок 2.21 - UML-діаграма діяльності

## 2.5 Висновки до другого розділу

У другому розділі виконано проектування автоматизованої системи обліку складських товарів і підтримки торговельної діяльності. Розроблено структуру бази даних, UML-діаграми варіантів використання та діяльності, що відображають логіку роботи системи та взаємодію користувача з програмним продуктом. Запропоновані проєктні рішення забезпечують цілісність даних, функціональність системи та створюють основу для її подальшої реалізації й розширення.

## **3 РОЗРОБКА ТА ТЕСТУВАННЯ РОБОТИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ СКЛАДСЬКИХ ТОВАРІВ ТА ПІДТРИМКИ ТОРГІВЛІ**

### 3.1 Розробка автоматизованої системи в середовищі програмування Microsoft Visual Studio.

Для створення автоматизованої системи обліку складських товарів та підтримки торгівлі, що розробляється в межах магістерської кваліфікаційної роботи, було обрано мову програмування C # та середовище розробки додатків під ОС Windows Microsoft Visual Studio (MVS) [49-51].

В C # використовується синтаксис, що сильно нагадує C і наближений до C++ та Java. Для прикладу, C # має підтримку спадковості, поліморфізму, повторного завантаження операторів, статичної типізації. Такі підходи допомагають у вирішенні поширених завдань в сфері розробки масштабних проектів, які зберігають свою гнучкість, розширюваність і масштабованість. C # все ще розвивається, після кожного доповнення в мові з'являються нові функції, переважно досить корисні. Серед них: асинхронна робота, змінне зв'язування, лямбда і т. д.

Якщо мова заходить про C #, часто маються на увазі технології, пов'язані з платформою .NET (WPF, ASP.NET). Те ж саме справедливо в зворотному напрямку, кажучи про .NET, думають про C #. Безумовно, поняття мають жорсткий взаємозв'язок, але не є синонімами. C # - це мова, створена для взаємодії з фреймворком .NET [52].

Ключові риси платформи .NET:

- Працює паралельно з різними мовами. Популярністю користується C #, в чому зобов'язана загальномовному середовищі CLR. Зараз платформа здатна працювати з C #, VB.NET, C ++, F #, але і на цьому список не закінчується, адже вона працює з діалектами, що прив'язані до .NET (на зразок Delphi.NET). Після компіляції коду з будь-якого з перерахованих мов, все інтерпретується в спільну мову CIL - це своєрідний асемблер для .NET. Такий підхід дозволяє використовувати кілька мов для створення модулів програми;

- Кросплатформеність. Дану платформу реально переносити, хоч і є окремі обмеження. Сьогодні актуальна версія фреймворку працює на всіх підтримуваних Windows: від Windows Vista до Windows 10. За рахунок проекту Mono з'явилася можливість розробляти програми під Linux (різні дистрибутиви), Android і iOS;

- Велика бібліотека класів. .NET Framework має єдину, уніфіковану бібліотеку класів, з якою працюють всі підтримувані мови. Бібліотека класів стане в нагоді при створенні будь-яких програм: від блокнота до величезного веб-сайту;

- Маса вбудованих технологій. Середовище CLR в поєднанні з бібліотекою класів - це основа для великого пакета допоміжних технологій. Їх можуть використовувати всі програмісти під час розробки додатків. Як приклад, при взаємодії з базами даних можна використовувати технологію ADO.NET. Під час створення графічних редакторів з численними функціями зручно використовувати WPF. Під час веб-розробки використовують ASP.NET [53].

Варто також згадати про одну особливість фреймворка і мови - автоматичне очищення кешу в пам'яті. Таким чином відпадає необхідність згадувати, і вручну задавати код для вивільнення зайнятої пам'яті. Серед CLR без додаткових команд подбає про збір кешу і його видаленні.

Обране середовище розробки MVS ідеально підходить для розробки проектів різного формату. Microsoft Visual Studio при створенні проекту надає користувачеві обрати серед багатьох типів проектів, що представляє широкі можливості даного середовища для розробки.

При написанні програмного коду, редактор підсвічує синтаксис і виконує форматування коду, що, в свою чергу, пришвидшує написання програми та робить її а рази більш читабельною. Більше того, редактор MVS автоматично завершує деякі структури коду, наприклад, варто почати друкувати оператор вибору switch як редактор сам його завершить.

Для початку роботи із MVS впершу чергу перерахуємо, які компоненти повинні бути встановлені для створення графічного додатку на мові C ++. Їх три, всі відносяться до розділу "Desktop development with C ++":

- VC ++ 2017 v141 toolset (x86, x64);
- Windows 10 SDK (10.0.15063.0) for Desktop C ++ x86 and x64;
- C ++ / CLI support.

Перші два потрібні для створення будь-якої програми на C ++, третій - саме для створення графічної оболонки програми [54]. На рисунку 3.1 показано, де ці компоненти обираються в налаштуваннях.

## Summary

- > Visual Studio core editor
- ✓ Desktop development with C++ \*
- Included
  - ✓ Visual C++ core desktop features
- Optional
  - VC++ 2017 v141 toolset (x86,x64)
  - C++ profiling tools
  - Windows 10 SDK (10.0.15063.0) for Desktop C++ x8...
  - Visual C++ tools for CMake
  - Visual C++ ATL support
  - Windows 8.1 SDK and UCRT SDK
  - Windows XP support for C++
  - MFC and ATL support (x86 and x64)
  - C++/CLI support
  - Clang/C2 (experimental)
  - Modules for Standard Library (experimental)

Рисунок 3.1 – Налаштування компонентів для початку розробки

В Microsoft Visual Studio є два способи зробити додаток з формами, рішення приймається в момент створення нового проекту [55].

Перший з них - використовувати Windows Forms, який реалізує графічний інтерфейс користувача і входить в Microsoft.NET Framework. Даний підхід спрощує доступ до інтерфейсних елементів Microsoft Windows завдяки створенню обгортки для Win32 API в керованому коді. Даний підхід дуже схожий з побудовою консольного додатку, але трохи більш складний тому використовує саме форми.

Другий спосіб заснований на використанні Microsoft Foundation Classes (MFC), бібліотеці, яка бере на себе створення каркасу додатків. На відміну від першого, MFC «з коробки» використовує патерн MVC (Model-View-Cont roller).

В розробці автоматизованої системи обліку складських товарів та підтримки торгівлі, використовувався перший спосіб створенні програмного застосунку [56, 57].

В якості серверу бази даних, в розробці було використано Firebird SQL Server, що був налаштований та підключений через DDEX провайдер. Для того, щоб Firebird Data Source з'явився в Microsoft Visual Studio в якості ресурсу для підвантаження даних, необхідно:

- Встановити DDEX Provider.

- Підготувати reg-файл. Для цього необхідно скачати версію базового reg-файлу, знайти на комп'ютері папку, де зберігаються настройки Visual Studio, в ній знайти папку, ім'я якої починається з 15.0\_. В даному випадку це 15.0\_3ab834cd. Скрін з необхідною папкою зображено на рисунку 3.2.

Имя	Дата изменения	Тип	Размер
10.0	05.11.2015 13:37	Папка с файлами	
11.0	30.09.2015 13:35	Папка с файлами	
14.0	03.05.2017 9:14	Папка с файлами	
15.0_3ab834cd	05.08.2017 21:32	Папка с файлами	
Packages	10.04.2017 10:29	Папка с файлами	
SettingsLogs	08.08.2017 11:59	Папка с файлами	
vshub	30.09.2015 13:37	Папка с файлами	

Рисунок 3.2 – Необхідна папка для заміни reg-файлу

- Відкрити базовий reg-файл в блокноті і зробити заміни

### 15.0\_Config ### на 15.0\_3ab834cd (11 заміні)

### DDEX\_PATH ### на C: \ \ Програмні файли (x86) \ \ FirebirdDDEX (1 аналог)

В результаті файл буде виглядати наступним чином – рисунок 3.3.

```

Windows Registry Editor Version 5.00
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataSources\{2979569E-416D-4DD8-B06B-EBCB70DE7A4E}]
@="Firebird Data Source"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataSources\{2979569E-416D-4DD8-B06B-EBCB70DE7A4E}\Support
ingProviders\{92421248-F044-483A-8237-74C7FBC62971}]
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}]
@=".NET Framework Data Provider for Firebird"
"DisplayName"="Provider_DisplayName, FirebirdSql.VisualStudio.DataTools.Properties.Resources"
"ShortDisplayName"="Provider_ShortDisplayName, FirebirdSql.VisualStudio.DataTools.Properties.Resources"
"Description"="Provider_Description, FirebirdSql.VisualStudio.DataTools.Properties.Resources"
"CodeBase"="C:\Program Files (x86)\FirebirdDDEX\FirebirdSql.VisualStudio.DataTools.dll"
"InvariantName"="FirebirdSql.Data.FirebirdClient"
"Technology"="{77AB9A9D-7889-4ba7-91AC-873F5338F1D2}"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}\Suppo
rtedObjects]
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}\Suppo
rtedObjects\DataConnectionSupport]
@="FirebirdSql.VisualStudio.DataTools.FbDataConnectionSupport"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}\Suppo
rtedObjects\DataConnectionProperties]
@="FirebirdSql.VisualStudio.DataTools.FbDataConnectionProperties"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}\Suppo
rtedObjects\DataConnectionUIControl]
@="FirebirdSql.VisualStudio.DataTools.FbDataConnectionUIControl"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}\Suppo
rtedObjects\DataSourceInformation]
@="FirebirdSql.VisualStudio.DataTools.FbDataSourceInformation"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}\Suppo
rtedObjects\DataObjectSupport]
@="FirebirdSql.VisualStudio.DataTools.FbDataObjectSupport"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\DataProviders\{92421248-F044-483A-8237-74C7FBC62971}\Suppo
rtedObjects\DataViewSupport]
@="FirebirdSql.VisualStudio.DataTools.FbDataViewSupport"
[HKEY_USERS\VS2017PrivateRegistry\Software\Microsoft\VisualStudio\15.0_3ab834cd_Config\Services\{AEF32AEC-2167-4438-81FF-AE6603341536}]
@="{8d9358ba-ccc9-4169-9fd6-a52b8aee2d50}"
"Name"="Firebird Provider Object Factory"

```

Рисунок 3.3 – Змінений reg-файл

- Виправити приватний реєстр. Для його запускуємо Regedit. Далі розділ HK\_USERS. Вибираємо меню Файл -> Завантажити куц. Вибираємо файл privateregistry.bin, який знаходиться в папці налаштувань Visual Studio. Назву куца обов'язково вказуємо як **VS2017PrivateRegistry**. Завантаження куца зображено на рисунку 3.4.

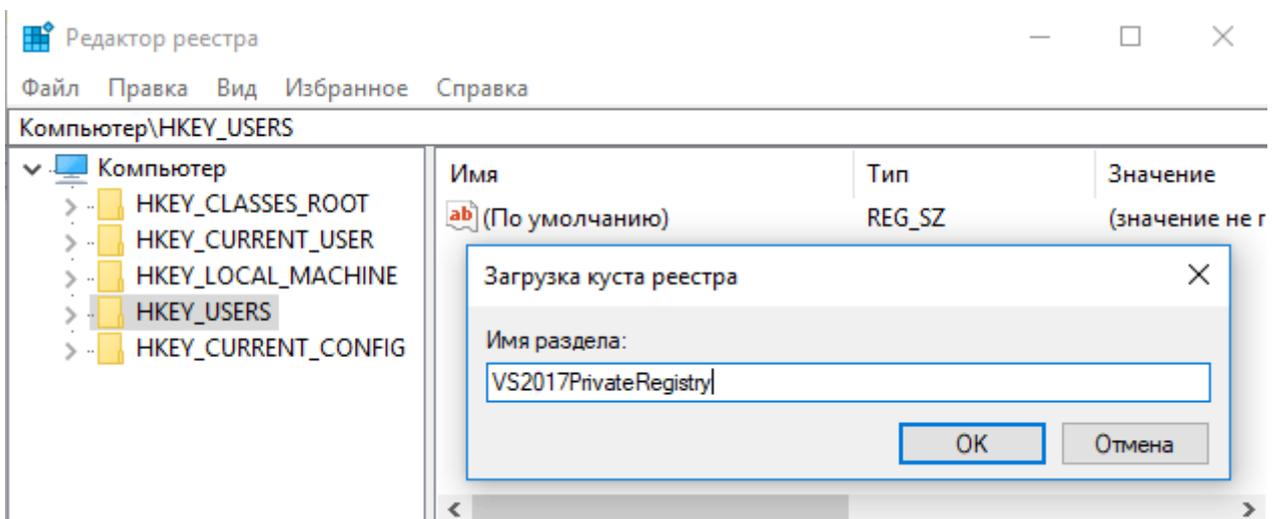


Рисунок 3.4 – Зміни в реєстрі

- Запускаємо на виконання збережений раніше файл реєстру або імпортуємо його через regedit. Розкриваємо розділ HKEY\_USERS. Обираємо щойно створений розділ VS2017PrivateRegistry. Вибираємо меню Файл -> Вивантажити куш.

- Запускаємо Visual Studio і перевіряємо, чи з'явився джерело даних Firebird Data Source в переліку.

Після виконаних дій Firebird Data Source став доступним для підключення і вивантаження даних до системи, що зображено на рисунку 3.5.

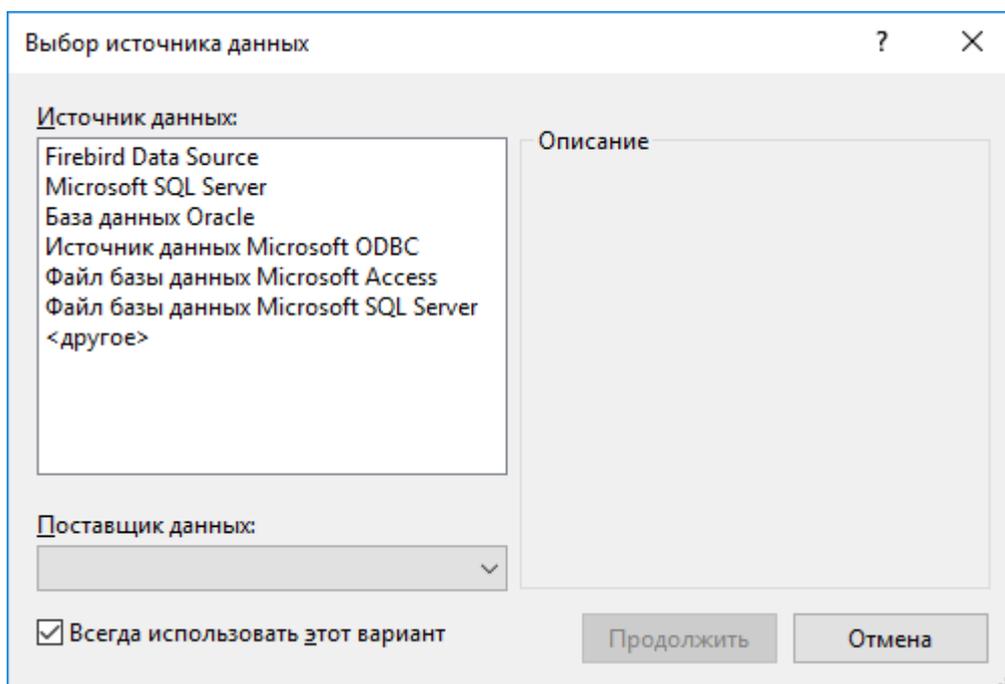


Рисунок 3.5 – Вибір джерела даних в Microsoft Visual Studio

За допомогою елементів управління PrintDocument, PageSetupDialog, PrintDialog, PrintPreviewDialog, що реалізовані в .NET Framework, є можливість забезпечувати друк документів з інформацією із баз даних [58].

Коли викликається команда друку на принтер або попереднього перегляду документа, то в роботу вступають обробники подій класу PrintDocument, за допомогою подій якого можна керувати друком документів.

Перелік можливих подій класу PrintDocument:

- `BeginPrint` – подія, що виникає перед самим початком друку документа. В обробник цієї події вписується код ініціалізації внутрішніх змінних, які керують друком документа;

- `PrintPage` - виникає один раз при друку кожної сторінки.

- `EndPrint` - виникає, після друку документа;

- `QueryPageSetting` - виникає перш ніж кожна сторінка повинна бути надрукована. Перехоплення цієї події використовується у випадках, коли потрібно змінити налаштування конкретної друкованої сторінки [59, 60].

Клас `PageSetupDialog` відповідає за діалогове вікно, яке дозволяє користувачеві змінювати налаштування сторінки для друку (внутрішні поля, орієнтація друку, і т.п.).

`PrintDialog` представляє діалогове вікно, яке дозволяє користувачеві вибирати принтер і інші настройки принтера такі як кількість копій, орієнтація сторінки і т.п .

`PrintPreviewDialog` відповідає за діалогове вікно, яке показує користувачеві попередній перегляд документа в тому вигляді, якому він буде відображений при друку [61].

Лістинг програмного забезпечення представлено в додатку В.

### 3.2 Перевірка правильності роботи автоматизованої системи обліку складських товарів та підтримки торгівлі

Розроблений програмний продукт автоматизованої системи обліку складських товарів та підтримки торгівлі встановлюється із пустою базою даних, тому перше, що потрібно зробити зареєстрованому користувачу це заповнити БД.

До бази даних необхідно занести дані про групи товарів та самі товари, інформацію про покупців та постачальників. Вносити записи можна вручну або імпортувати із файлів з форматом CSV.

CSV це текстовий формат, призначений для представлення табличних даних. Рядки таблиці відповідають рядкам тексту, який містить одне або кілька полів, розділених комами.

Процес заповнення бази даних вручну представлено на рисунках.

Для того, щоб додати чи редагувати групу товарів потрібно перейти у відповідний пункт із меню в лівій частині вікна, далі на пустому полі робочого середовища клікнути правою кнопкою миші та обрати потрібний пункт. Дію зображено на рисунку 3.6.

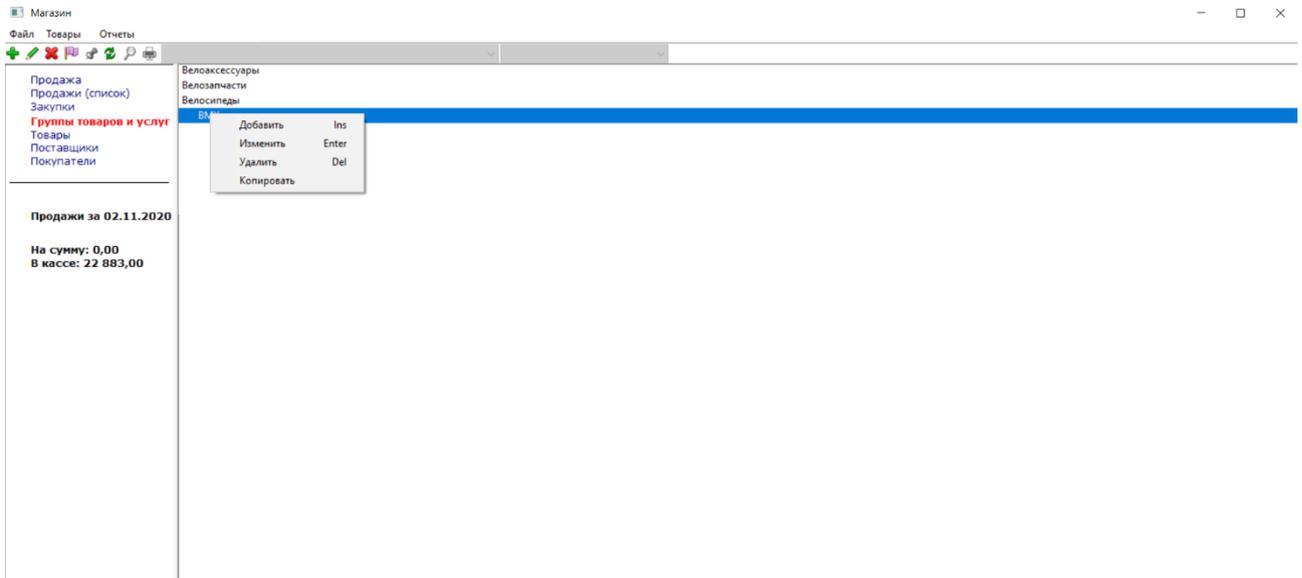


Рисунок 3.6 – Виклик контекстного меню в розділі групи товарі

Група товарів визначається іменем та набором атрибутів чи характеристик для даної групи.

Після створення групи чи груп товарів, можна перейти до створення безпосередньо карточок товарів. Для цього потрібно перейти в пункт меню Товари в лівій частині головного вікна програми. На екрані відобразиться список існуючих товарів в базі даних, що представлено на рисунку 3.7.

Артикул	Наименование	Группа	Цена	Доступно
00012	BMX Stolen CASINO	Велосипеды\BMX	10 585,00 грн	8
2346	BMX Stolen HEIST	Велосипеды\BMX	12 298,00 грн	20
0349BA	Подстаканник	Белоаксессуары...	120,00 грн	1

Рисунок 3.7 – Список створених карточок товарів

Далі на пустому полі робочого середовища клікнути правою кнопкою миші та обрати потрібний пункт – Додати товар. Процес заповнення карточки товару представлено на рисунку 3.8. Інформація про товар складається з:

- Назви товару;
- Приналежності до однієї із груп товарів (опціонально);
- Артикулу – власне це і є поле `product_id`;
- Вартості товару;
- Характеристик, його атрибутів;
- Кількості товарів на балансі складу / магазину;
- Поля `delivery_id`, що визначає, до якої поставки відноситься товар.

Рисунок 3.8 – Заповнення карточки товару

Для заповнення інформації про постачальників, необхідно перейти в однойменний пункт меню та вибрати опцію Додати. Картка постачальника складається із наступних даних:

- Ім'я або назва компанії постачальника;
- Адреса;
- Телефон;
- E-mail.

Вікно із заповненою карткою постачальника представлено на рисунку 3.9.

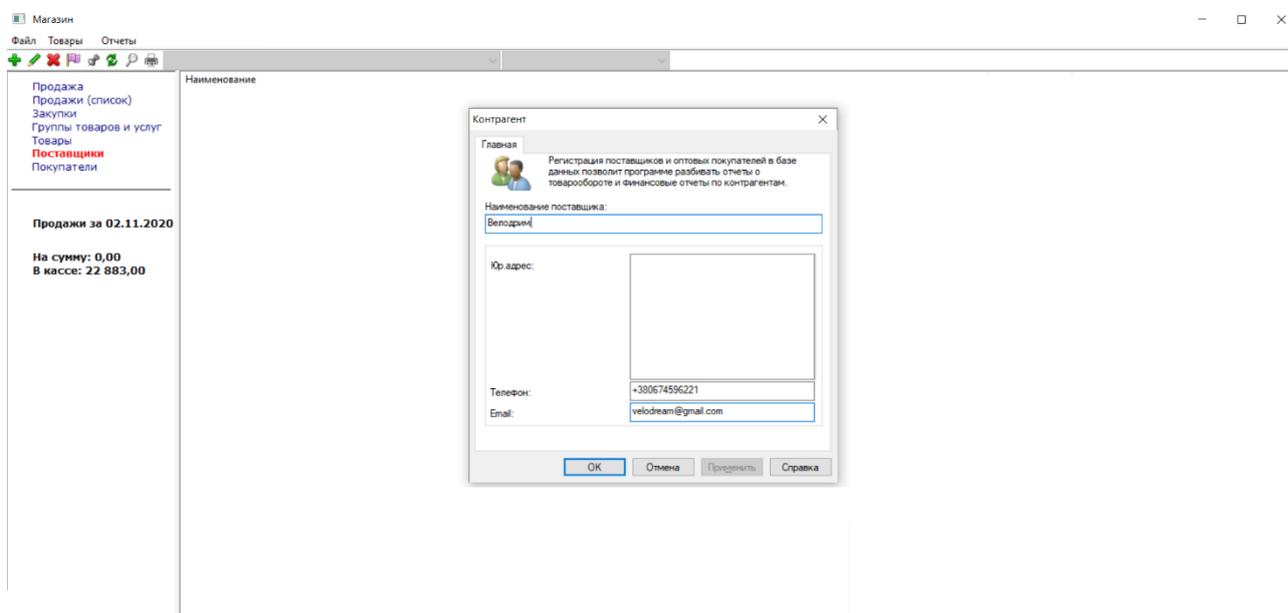


Рисунок 3.9 – Заповнення картки із інформацією про постачальника

Аналогічним способом переходимо до заповнення картки покупця, що містить поля з іменем, номером дисконтної картки – `client_id`, а також номером телефону.

Вікно із заповненою карткою покупця представлено на рисунку 3.10.

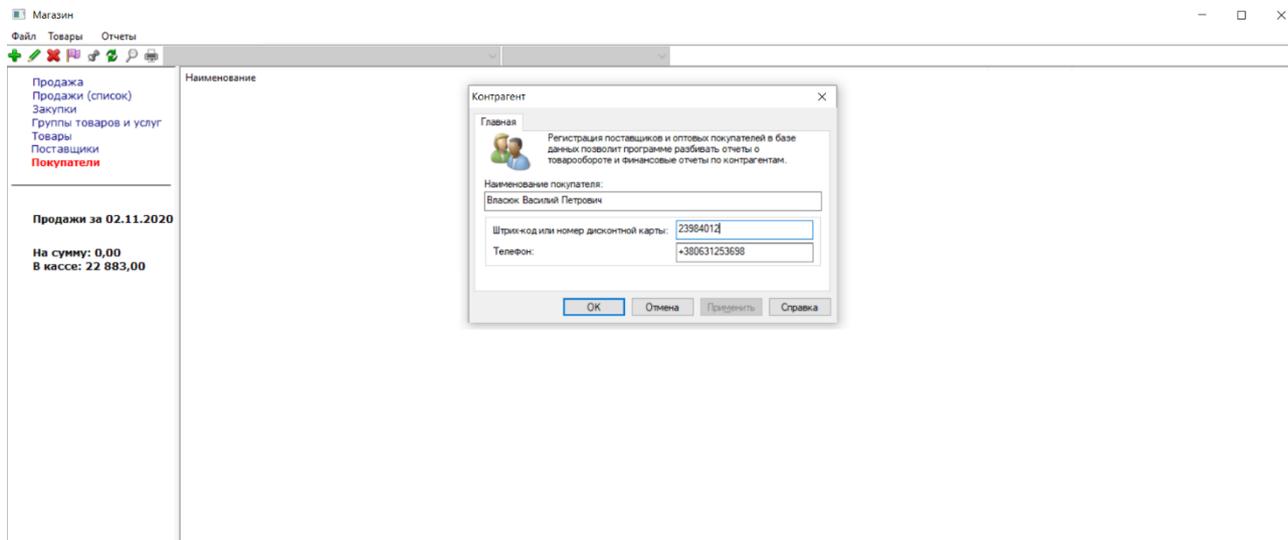


Рисунок 3.10 – Заповнення картки з інформацією про покупця

В поточному розділі Покупці відобразиться список із створеними покупцями, що зображено на рисунку 3.11.

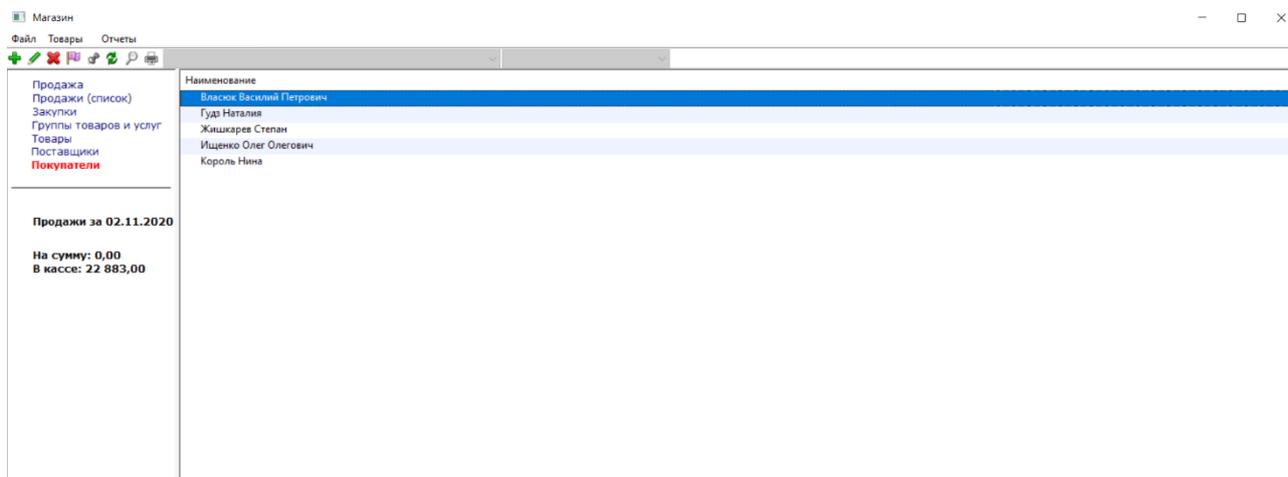


Рисунок 3.11 – Відображення списку існуючих покупців

Коли в базі даних вже міститься інформація про постачальників, можна починати фіксувати закупки в середині автоматизованої системи. Для цього потрібно перейти в пункт меню Закупки та натиснути Додати.

У новому відкритому вікні можна реєструвати прихід товарів, що вже занесені в базу даних системи, або одразу створювати нові картки товарів. Як це працює зображено на рисунках 3.12-3.13.

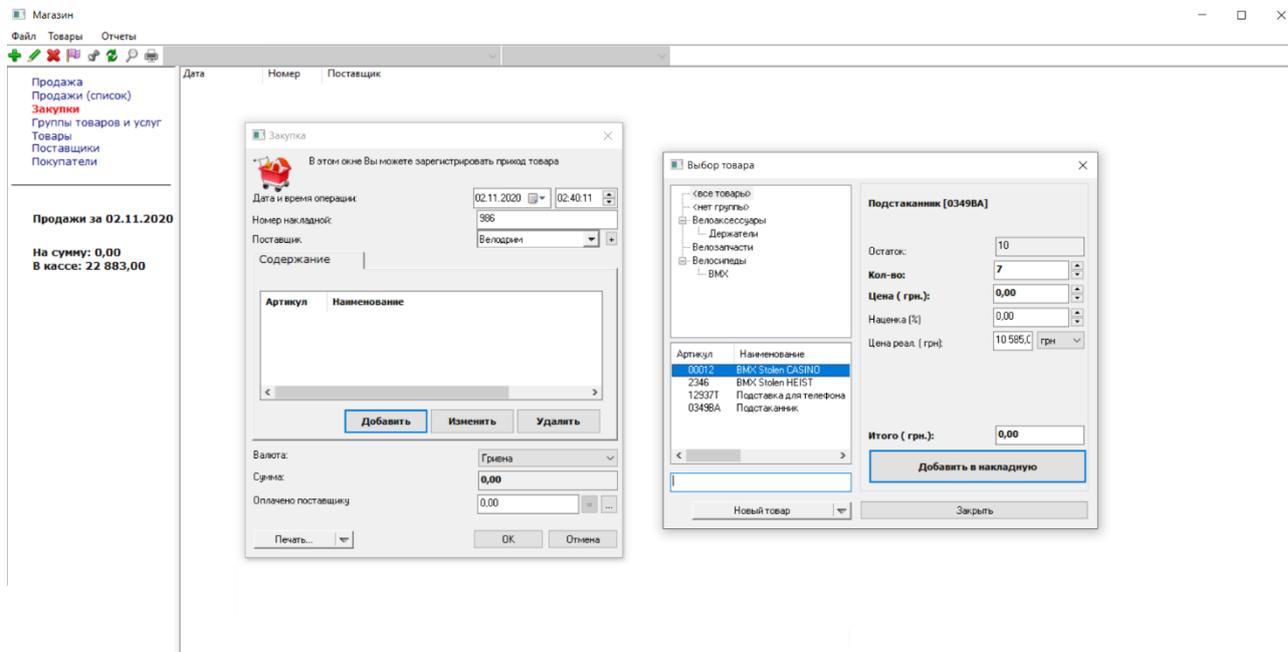


Рисунок 3.12 – Вікно реєстрації закупки та додавання товарів із списку існуючих

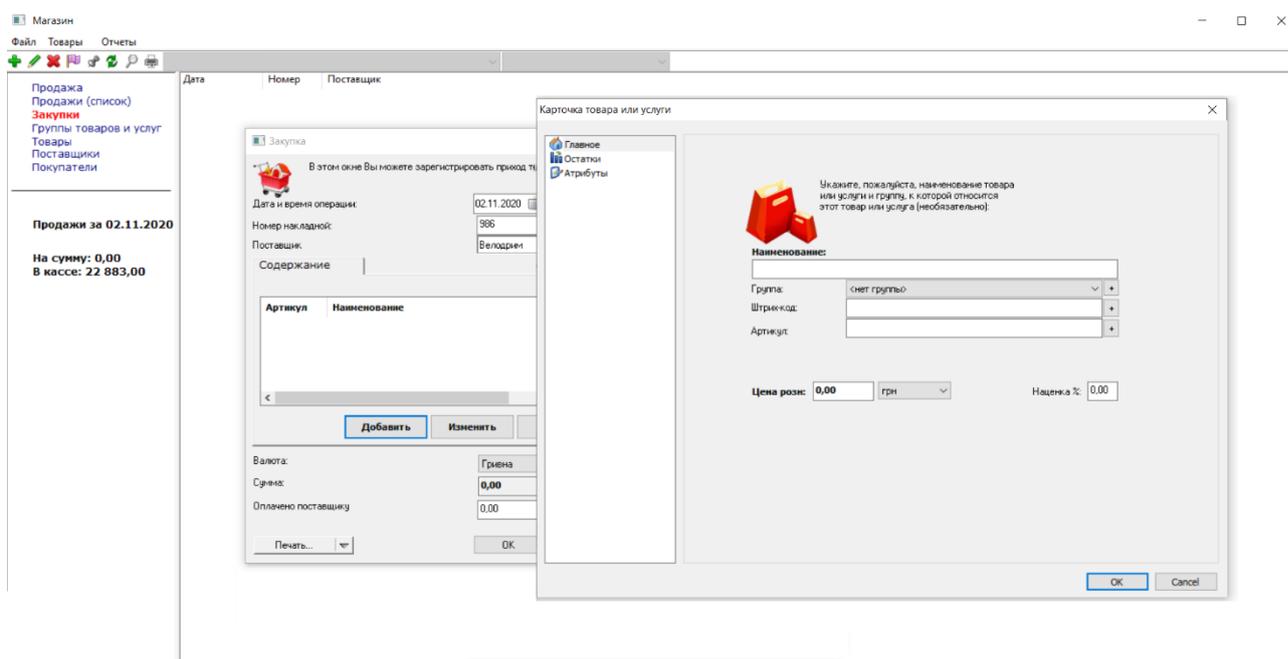


Рисунок 3.13 – Вікно реєстрації закупки та додавання товарів шляхом створення нових карток товарів

Після додавання товарів у вікні реєстрації закупки відображається повна оптова вартість товарів, найменування та артикули товарів, номер накладної закупки – `delivery_id`, ім'я чи назва постачальника, дата та час. Вікно зображено на рисунку 3.14.

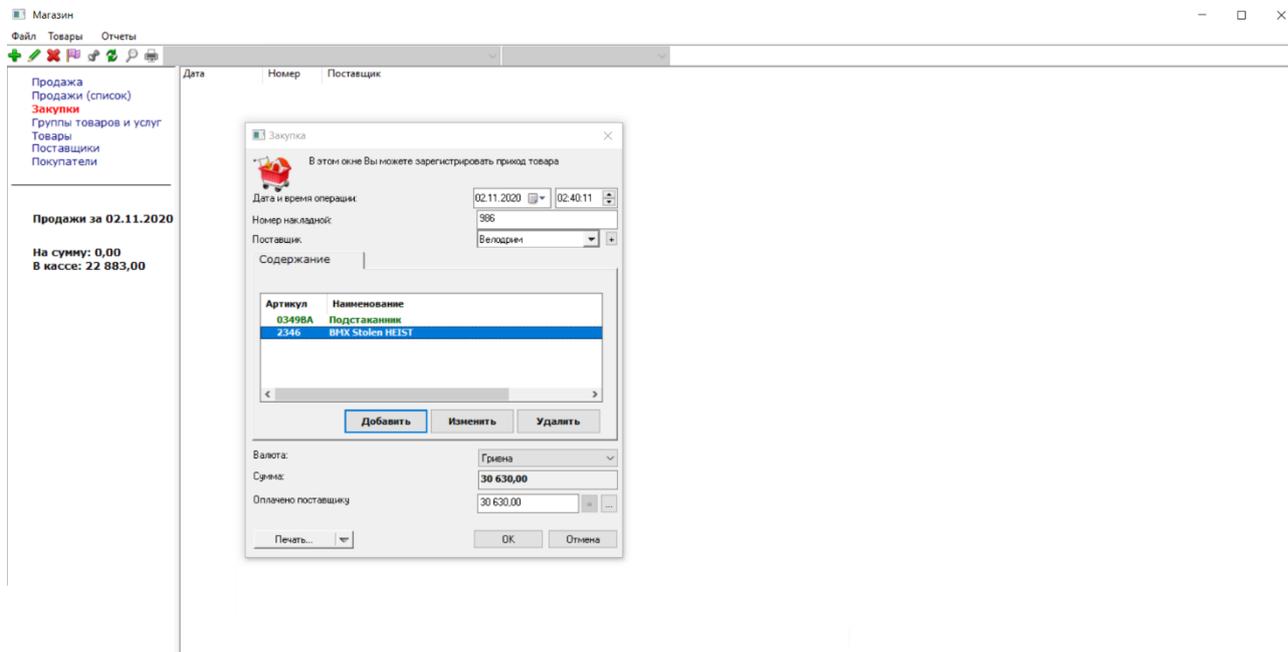


Рисунок 3.14- Вікно реєстрації закупки після додавання товарів

У розділі Закупки тепер будуть відображатись зареєстровані користувачем закупки. Інформацію про них можна переглядати в цьому ж розділі.

На рисунку 3.15 зображено список зареєстрованих в автоматизованій системі закупок.

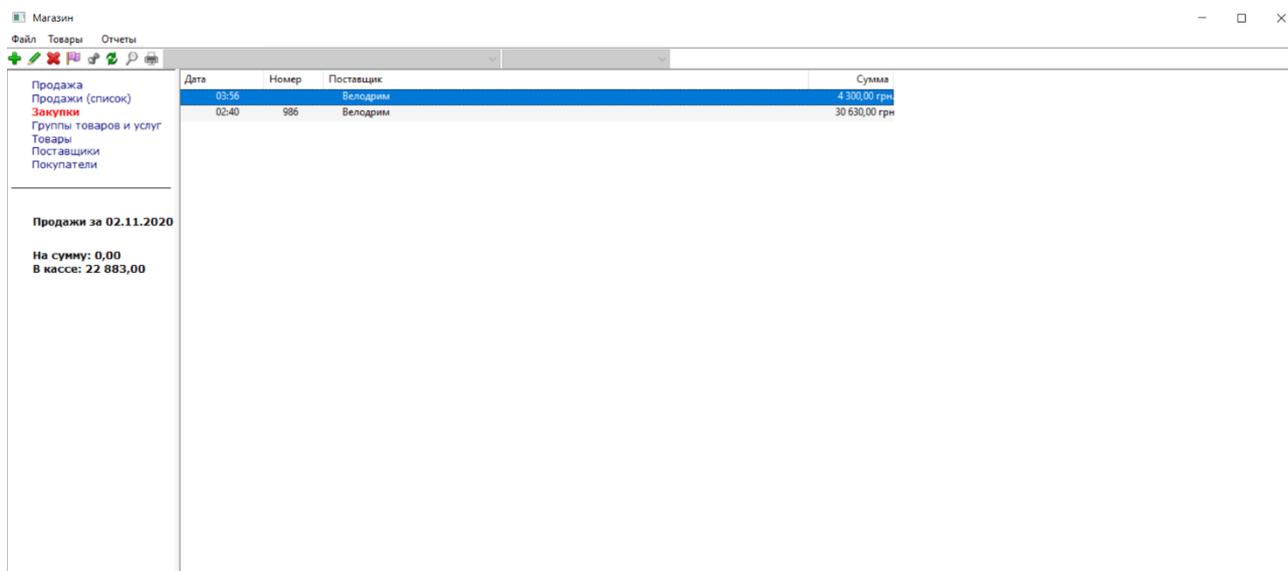


Рисунок 3.15 – Список зареєстрованих в системі закупок

Власне продаж товарів, інформація про які вже міститься в автоматизованій системі, можливий з пункту меню Продажі.

В робочому полі даного розділу потрібно обрати покупця (опціонально), додати до продажу товар, обравши попередньо його із списку наявних в магазині / на складі, та його кількість, потім натиснути на кнопку Оплата або клавішу F12 на клавіатурі. Дію зображено на рисунку 3.16.

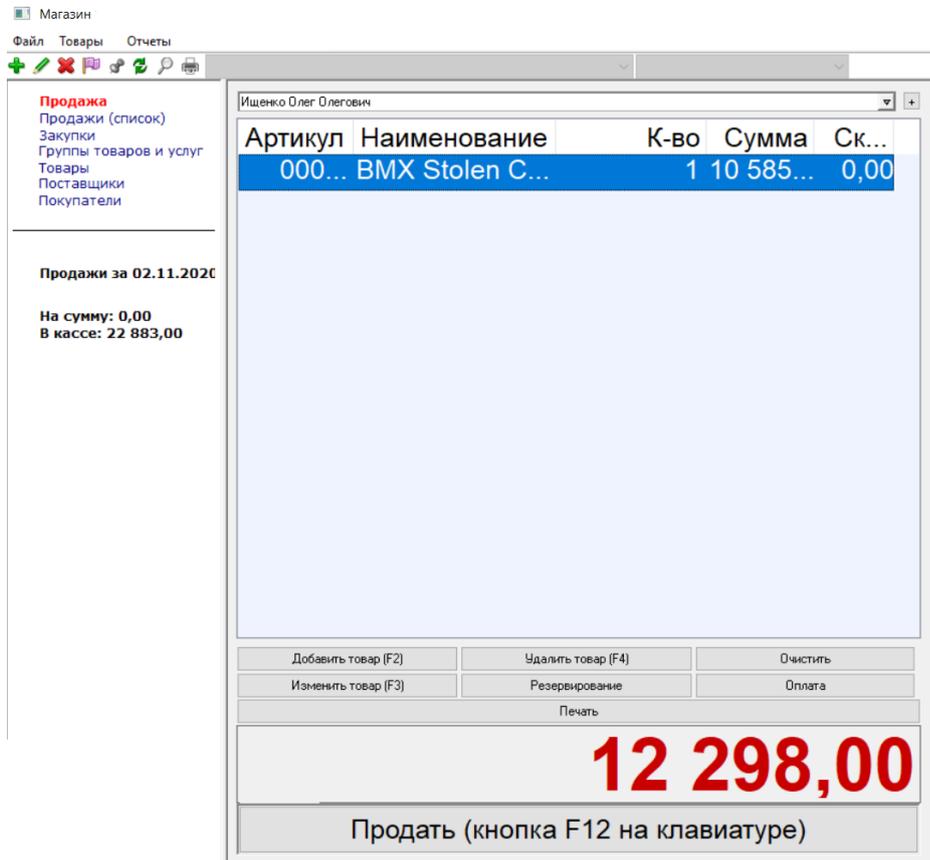


Рисунок 3.16 – Розділ реєстрації продаж

Після оплати покупка відправляється до розділу Продажі (список), де у вигляді, відповідно, списку міститься інформація про зареєстровані покупки, що зображено на рисунку 3.17.

Дата	Товары	Сумма
02:51	[00012] BMX Stolen CASINO	10 585,00 грн.
02:50	[2346] BMX Stolen HEIST	12 298,00 грн.
28.10.2020...	[00012] BMX Stolen CASINO, [2346] BMX Stolen HEIST	22 883,00 грн.

Рисунок 3.17 – Відображення списку зареєстрованих продаж

В головному меню програми є можливість завантаження та друку документів, сформованих на певній інформації із бази даних.

На рисунку 3.18 представлено звіт по продажам обраного Головного складу за вибраний період.

BMX				
Артикул	Наименование	К-во	Сумма	Скидка
00012	BMX Stolen CASINO	2.000	21170.00грн.	0.00 грн.
2346	BMX Stolen HEIST	2.000	24596.00грн.	0.00 грн.
		4.000	45766.00грн.	0.00 грн.
<b>Итого</b>		<b>4.000</b>	<b>45766.00грн.</b>	<b>0.00грн.</b>

Рисунок 3.18 – Звіт по продажам за вибраний звітний період з 03.10.2020 по 02.11.2020

На рисунку 3.19 сформовано звіт із залишками товарів на Головному складі на поточну дату (02.11.2020).

## Залишки товару на складі

02.11.2020

Головний склад

Велоаксесуари\Тримачі						
Арт.	Найменування	К-сть	В цінах реалізації:		В цінах закупівлі:	
			Ціна	Сума	Ціна	Сумма
12937Т	Тримач для телефону	110.000	140,00 грн	15 400,00 грн	100,00 грн	11 000,00 грн
0349ВА	Підстаканник	18.000	120,00 грн	2 160,00 грн	90,00 грн	1 620,00 грн
		<b>128.000</b>	<b>17560,00 грн.</b>		<b>12 620,00 грн.</b>	
Велосипеди\BMX						
Арт.	Найменування	К-сть	В цінах реалізації:		В цінах закупівлі:	
			Ціна	Сума	Ціна	Сумма
00012	BMX Stolen CASINO	7.000	10 585,00 грн	74 095,00 грн	0,00 грн	0,00 грн
2346	BMX Stolen HEIST	22.000	12 298,00 грн	270 556,00 грн	10 000,00 грн	220 000,00 грн
		<b>27.000</b>	<b>344651,00 грн.</b>		<b>220 000,00 грн.</b>	
<b>Разом по звіту:</b>			<b>362 211,00 грн.</b>		<b>232 620,00 грн.</b>	

Рисунок 3.19 – Звіт із залишками товарів на складі

На рисунку 3.20 представлено звіт із доходом, що згруповані по даті, за вибраний період.

### Виручка по дням

з 03.10.2020 по 02.11.2020

Дата	Чеків всього (кріме резерв-ів)	Віднущено товару на сумму	Оплачено		
			Готівкою	Безготівково	Всього
27.10.2020	1	22883.00грн.	22883.00грн.	0,00 грн.	22883.00грн.
02.11.2020	2	22883.00грн.	11475.00грн.	11.408,00 грн.	22883.00грн.
<b>За весь період:</b>	<b>3</b>	<b>45766.00грн.</b>	<b>34358,00 грн.</b>	<b>11408,00 грн.</b>	<b>45766.00 грн.</b>

Рисунок 3.20 – Звіт про виручку по днях за обраний період

На рисунку 3.21 представлено звіт про роботу продавця, що включає інформацію про здійснені продажі та повернення товару.

**Звіт продавця**

з 03.10.2020 по 02.11.2020

<b>Реалізація</b>					
Дата	Артикул	Товар або послуга	К-сть	Знижка	Σ
28.10.20	00012	BMX Stolen CASINO	1.000	0.00	10 585,00 грн.
28.10.20	2346	BMX Stolen HEIST	1.000	0.00	12.298,00 грн.
02.11.20	2346	BMX Stolen HEIST	1.000	0.00	12.298,00 грн.
02.11.20	00012	BMX Stolen CASINO	1.000	0.00	10 585,00 грн.
Разом:					45 766,00 грн.
Сплачено:					45 766,00 грн.
<b>Повернення покупця</b>					
Дата	Артикул	Товар або послуга	К-сть	Знижка	Сума
02.11.20	0349BA	Підстаканник	1.000	0.00	120,00 грн.
Разом:					120,00 грн.
Сплачено:					120,00 грн.
<b>Підсумок:</b>					

Товари на суму з урахуванням знижок: 45 646,00 грн.

Сумарна знижка: 0,00 грн.

Сплачено всього: 45 646,00 грн.

Рисунок 3.21 – Звіт про роботу продавця

**Висновок до розділу 3**

Перевіривши правильність роботи програми, заповнення таблиць бази даних та виводу інформації із них у вікна програми, зроблено висновок про працездатність розробленої автоматизованої системи обліку складських товарів та підтримки торгівлі.

Новизною та перевагою даної автоматизованої системи є її універсальність та одночасно не обтяженість додатковими функціональними можливостями, що відволікають увагу та час користувачів при роботі з програмою.

Дана автоматизована система обліку складських товарів та підтримки торгівлі дозволить автоматизувати рутинні завдання та мінімізувати вплив помилок людського фактору на роботу підприємства, задля зменшення втрат та ефективного розподілу вивільнених ресурсів персоналу на інші області роботи компанії.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Комерційний та технологічний аудит науково-технічної розробки

Комплексна Магістерська кваліфікаційна робота з розробки та дослідження «Розробка автоматизованої системи обліку складських товарів і підтримки торговельної діяльності. Частина 1. Створення бази даних та розробка десктопного програмного забезпечення.» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

Дана магістерська робота присвячена розробці прикладного інструменту для оптимізації процесів у сфері оптової та роздрібною торгівлі. Проектований мобільний застосунок має гнучку модель комерціалізації: він може функціонувати як автономний продукт (за моделлю підписки) або ж інтегруватися як мобільний терміналів у вже існуючі ERP-системи підприємств.

В якості аналогів на ринку можна розглядати наприклад Trini POS, орієнтовна вартість якого складає 350 доларів США на користувача на рік або приблизно 14 800 гривень на рік.

Для проведення комерційного та технологічного аудиту до оцінювання залучають не менше трьох незалежних експертів. Рекомендується оцінювати науково-технічний рівень розробки та її комерційний потенціал за п'ятибальною системою, застосовуючи 12 критерій у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
Ринкові переваги					
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою

7	Активна конкуренція великих компаній	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практика на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	--	---

Усі дані по кожному параметру занесено в таблиці 4.2

Експертами було обрано Юхимчук М. С., Ковтун В. В. та Дубовой В.М., оскільки вони є висококваліфікованими викладачами ВНТУ та водночас керівниками магістерських кваліфікаційних робіт. Їхній фаховий досвід, наукові здобутки та практична компетентність у відповідних галузях забезпечують об'єктивну, професійну й обґрунтовану експертизу результатів дослідження.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Юхимчук М. С.	Ковтун В. В.	Дубовой В.М.
	Бали		
Технічна здійсненність концепції	4	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	3
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	3
Супровідна документація	4	3	3
Сума	45	42	41
Середньоарифметична сума балів	$(45+42+41) / 3 = 42,67$		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 – Рекомендаціями рівня комерційного потенціалу даної

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як свідчать результати оцінювання, комерційний потенціал нового програмного продукту є високим. Це зумовлено підвищенням рівня безпеки ІКС завдяки інтеграції адаптованих методів управління ризиками інформаційної безпеки. Такий підхід дозволяє визначати оптимальні стратегії оцінки ризиків для підприємств у межах функціонування розробленої автоматизованої системи обліку складських товарів і торговельної діяльності

#### 4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Оскільки в межах даної комплексної магістерської кваліфікаційної роботи здійснюється розробка програмного забезпечення, функції проєктування, реалізації та тестування програмного продукту виконуються одним розробником.

Основна заробітна плата розробника, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t \quad (4.1)$$

де  $M$  – місячний посадовий оклад конкретного розробника, грн.;

$T_p$  – число робочих днів у місяці, 23 днів;

$t$  – число днів роботи розробника.

Результати розрахунків наведено в таблиці 4.4.

Таблиця 4.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	20 000,00	1 000,00	5	5 000,00
Програміст	35 000,00	1 750,00	25	43 750,00
UX / UI дизайнер	23 000,00	1 150,00	25	28 750,00
Системний адміністратор	23 000,00	1 150,00	25	28 750,00
Всього				106 250,00

Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 12 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 12 \% / 100 \% \quad (4.2)$$

$$Z_d = (106250,00 \cdot 12 \% / 100 \% ) = 12750,00 \text{ (грн.)}$$

Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (4.3)$$

$$H_z = (106250,00 + 12750,00) \cdot 22 \% / 100 \% = 26180,00 \text{ (грн.)}$$

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T} \cdot \frac{t_{\text{вик}}}{12} \text{ [грн.]} \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{\text{вик}}$  – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на ноутбук балансова вартість якого становить 22000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 2 міс.

$$A_{\text{обл}} = \frac{22000}{2} \times \frac{2}{12} = 1833,33 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5. Так як вартість операційної системи та спеціалізованих ліцензійних нематеріальних ресурсів є меншою за 20 000 грн, такий нематеріальний актив не підлягає амортизації,  $B_{\text{нем.ак.}} = 6100$  грн.

Таблиця 4.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Термін корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Офісне обладнання (меблі)	25 000	4	2	1 041,67

## Продовження таблиці 4.5

Приміщення	1 600 000	20	2	13 333,33
Всього				16 208,33

Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_n, \quad (4.5)$$

де  $V$  – вартість 1 кВт-години електроенергії для 1 класу підприємства з ПДВ в 2025 році для Вінницької області за даними Енера-Вінниця,  $V = 12,69$  грн./кВт;

$P$  – встановлена потужність обладнання, кВт.  $P = 0,3$  кВт;

$\Phi$  – фактична кількість годин роботи обладнання, годин;

$K_n$  – коефіцієнт використання потужності,  $K_n = 0,9$ ;

$$V_e = 0,9 \cdot 0,3 \cdot 8 \cdot 50 \cdot 12,69 = 1370,52 \text{ (грн.)}$$

Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників.

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (4.6)$$

де  $H_{ib}$  – норма нарахування за статтею «Інші витрати».

$$I_e = 106250,00 * 78\% / 100\% = 82875,00 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.7)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 106250,00 * 135\% / 100\% = 143437,50 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 106250,00 + 12750,00 + 26180,00 + 16208,33 + 6100,00 + 1370,52 \\ + 82875,00 + 143437,50 = 395171,35 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \quad (\text{грн}), \quad (5.8)$$

де  $\eta$  – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то  $\eta=0,1$ ; технічного проектування, то  $\eta=0,2$ ; розробки конструкторської документації, то  $\eta=0,3$ ; розробки технологій, то  $\eta=0,4$ ; розробки дослідного зразка, то  $\eta=0,5$ ; розробки промислового зразка, то  $\eta=0,7$ ; впровадження, то  $\eta=0,9$ . Оберемо  $\eta = 0,5$ , так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 395171,35 / 0,5 = 790342,70 \text{ грн.}$$

#### 4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

— абсолютного економічного ефекту (чистого дисконтованого доходу);

— внутрішньої економічної дохідності (внутрішньої норми дохідності);

— терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.9)$$

де  $\pm\Delta\Pi_0$  – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$C_o$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки,  $C_o = C_b \pm \Delta C_o$ ;

$C_b$  – вартість програмного продукту у році до впровадження результатів розробки;

$\Delta N$  – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $\lambda = 0,8333$ .

$p$  – коефіцієнт, який враховує рентабельність продукту;

$\vartheta$  – ставка податку на прибуток, у 2025 році  $\vartheta = 18\%$ .

Припустимо, що при прогнозованій ціні 5600 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 400 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 13000 шт., протягом другого року – на 12000 шт., протягом третього року на 11000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0 \cdot 5500 + (5500 + 500) \cdot 13000) \cdot 0,8333 \cdot 0,26 \cdot (1 - 0,18) = 1385400 \text{ грн.}$$

$$\Delta\Pi_2 = (0 \cdot 5500 + (5500 + 500) \cdot (13000 + 12000)) \cdot 0,8333 \cdot 0,26 \cdot (1 - 0,18) = 26601000$$

грн

$$\Delta\Pi_3 =$$

$$(0 \cdot 5500 + (5500 + 500) \cdot (13000 + 12000 + 11000)) \cdot 0,8333 \cdot 0,26 \cdot (1 - 0,18) = 38257000 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 78712000, 00 грн.

#### 4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків  $III$ , що їх може отримати потенційний інвестор від можливого впровадження та

комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.10)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

$T$  – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;

$t$  – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\begin{aligned} ПП = & (1385400,00/(1+0,1)^1) + (26601000,00/(1+0,1)^2) + (38257000,00/ \\ & / (1+0,1)^3) = 1259454,55 + 21983471,07 + 28744850,49 = 49\,999\,878,90 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} \times ЗВ, \quad (4.11)$$

де  $k_{инв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{инв} = 2 \dots 5$ , але може бути і більшим;

$ЗВ$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 3 \cdot 790342,70 = 2\,371\,028,10 \text{ грн.}$$

Тоді абсолютний економічний ефект  $E_{abc}$  або чистий приведений дохід (*NPV, Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = \text{ПП} - PV, \quad (4.12)$$

$$E_{abc} = 49\,999\,878,90 - 2\,371\,028,10 = 47\,628\,850\,748,8 \text{ грн}$$

Оскільки  $E_{abc} > 0$  то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_e$ . Для цього використаємо формулу:

$$E_e = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.13)$$

$T_{жс}$  – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{1 + 47\,628\,850\,748,8 / 2\,371\,028,10} - 1 = 1,64$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.14)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні  $d = (0,09...0,15)$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05...0,5)$ .

$$\tau_{\min} = 0,14 + 0,05 = 0,19 .$$

Так як  $E_b > \tau_{\min}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_b}, \quad (4.15)$$

$$T_{ок} = 1 / 1,64 = 0,61 \text{ р.}$$

Оскільки  $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,61 роки, то фінансування даної наукової розробки є доцільним.

#### 4.5 Висновки до розділу

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 790 342,70 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,61 роки.

## ВИСНОВКИ

У комплексній магістерській кваліфікаційній роботі вирішено поставлену у вступі науково-практичну задачу підвищення ефективності роботи торговельної мережі шляхом створення автоматизованої системи обліку складських товарів та підтримки торговельної діяльності. На основі аналізу предметної області та бізнес-процесів підтверджено актуальність проблеми, зумовленої відсутністю цілісного інструменту для точного, оперативного та надійного ведення складського та торговельного обліку, а також значним впливом ручних операцій і людського фактору на якість управління.

Проведений огляд сучасних програмних рішень дозволив визначити ключові функціональні вимоги до майбутньої системи та сформувавши архітектуру програмного забезпечення, оптимальну для малого та середнього бізнесу. Розроблена база даних, UML-моделі та структура застосунку забезпечують узгодженість компонентів системи й створюють основу для подальшого розширення функціональності.

Створений десктопний застосунок реалізує автоматизацію основних операцій обліку товарів, ведення документів, управління контрагентами та формування звітності. Результати тестування підтвердили коректність роботи системи: операції заповнення бази даних, обробки інформації та відображення у вікнах програми виконуються стабільно та без збоїв. Це свідчить про працездатність та відповідність програмного продукту поставленим вимогам.

Практична цінність роботи полягає у можливості впровадження розробленої системи на торговельних підприємствах, що дозволить підвищити точність обліку, мінімізувати вплив людських помилок, зменшити операційні втрати та ефективно перерозподіляти робочі ресурси. Подальший розвиток системи може бути спрямований на інтеграцію з онлайн-магазинами, мобільними застосунками, сервісами аналітики та торговим обладнанням. Отримані результати повністю підтверджують досягнення мети магістерської роботи та демонструють ефективність розробленого програмного рішення

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гетьман О. О., Шаповал В. М. Економіка підприємства: навчальний посібник. — 2-ге вид. — Київ: Центр учбової літератури, 2010. — 488 с.
2. Василенко В. А. Теорія і практика розробки управлінських рішень: навчальний посібник. — Київ: Знання, 2002. — 224 с.
3. Шиян А. А. Економічна кібернетика: вступ до моделювання соціальних і економічних систем. — Львів: Магнолія, 2006. — 228 с.
4. Крушельницька О. В., Мельничук Д. П. Управління персоналом: навчальний посібник. — Київ: Кондор, 2003. — 296 с.
5. Пономаренко В. С., Журавльова І. В., Латишева І. Л. Інформаційні системи в управлінні персоналом: навчальний посібник. — Харків: ХНЕУ, 2008. — 336 с.
6. Дубовой В. М. Ідентифікація та моделювання технологічних об'єктів і систем керування: навчальний посібник. — Вінниця: ВНТУ, 2012. — 308 с.
7. Дубовой В. М., Москвіна С. М., Никитенко О. Д. Моделювання процесів і систем керування. — Вінниця: ВНТУ, 2009. — 76 с.
8. Коваленко М. В. Маркетинг у торгівлі: теорія та практика: навчальний посібник. — Київ: Центр учбової літератури, 2018. — 300 с.
9. Пасічник В. О. Управління запасами та логістичні операції: підручник. — Львів: Видавничий центр, 2017. — 320 с.
10. Литвиненко О. М. Управління інформаційними системами підприємства. — Київ: Наукова думка, 2015. — 280 с.
11. Кириченко О. В. Проектування баз даних: навчальний посібник. — Київ: КНТ, 2018. — 248 с.
12. Сергієнко Т. О. Мобільні додатки: розробка та тестування. — Харків: IT-Books, 2020. — 220 с.
13. Кузнєцова М. Г. Забезпечення захищеності інформаційних ресурсів у розподілених системах. — Київ: ІПРИ НАНУ, 2004. — 112 с.
14. Кузьмін О. Є., Мельник О. Г. Основи менеджменту: навчальний посібник. — Львів: Новий Світ, 2005. — 400 с.
15. Бондаренко І. М. Менеджмент логістики: навчальний посібник. — Львів: ВСЛ, 2014. — 280 с.

16. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. — O'Reilly Media, 2017. — 616 p. [https://www.augmentedmind.de/wp-content/uploads/2022/07/Designin-Data-intensive-Applications.pdf?utm\\_source=chatgpt.com](https://www.augmentedmind.de/wp-content/uploads/2022/07/Designin-Data-intensive-Applications.pdf?utm_source=chatgpt.com)
17. Silberschatz A., Korth H. F., Sudarshan S. Database System Concepts. — 7th ed. — McGraw-Hill, 2019. — 1376 p. [https://www.mpgcamb.com/wp-content/uploads/2024/12/Abraham-Silberschatz-Henry-F.-Korth-S.-Sudarshan-Database-System-Concepts-McGraw-Hill-Education-2019.pdf?utm\\_source=chatgpt.com](https://www.mpgcamb.com/wp-content/uploads/2024/12/Abraham-Silberschatz-Henry-F.-Korth-S.-Sudarshan-Database-System-Concepts-McGraw-Hill-Education-2019.pdf?utm_source=chatgpt.com)
18. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. — 3rd ed. — Addison-Wesley, 2003. — 152 p. <https://archive.org/details/umldistilledbrie00fowl>
19. Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2002. — 533 p. <https://archive.org/details/patternsofenterp00fowl>
20. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. — 2nd ed. — Addison-Wesley, 2005. — 512 p. <https://archive.org/details/unifiedmodelingl00booc>
21. Witten I. H., Frank E., Hall M. A. Data Mining: Practical Machine Learning Tools and Techniques. — 3rd ed. — Morgan Kaufmann, 2011. — 664 p. <https://tjzhifei.github.io/links/DM3.pdf>
22. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. — 9th ed. — McGraw-Hill, 2019. — 880 p. <https://thuvien.utm.edu.vn/ktpm/th5.2.%202019-DataScienceforSoftwareEngpreprint.pdf>
23. Стрембіцький П., Юхимчук М., Лесько В., Перепелиця С. (2025). Централізований моніторинг інфраструктури з використанням системи THANOS: перспективи та виклики. Herald of Khmelnytskyi National University. Technical Sciences, 347(1),417-422. <https://doi.org/10.31891/2307-5732-2025-347-57>.
24. Перепелиця С., Юхимчук М., Лесько В. Моделювання кіберфізичних систем управління в умовах негативних зовнішніх факторів / С. Перепелиця, М. Юхимчук, В. Лесько // Measuring and Computing Devices in Technological Processes. — 2025. — № 1. — С. 7–12. — Режим доступу: <https://doi.org/10.31891/2219-9365-2025-81-125>.

25. Юхимчук М., С., Стрембiцький П.П., Перепелиця С.С. Використання віддаленого монiторингу для підвищення якостi та ефективностi у виробничому середовищi. Herald of Khmelnytskyi National University. Technical Sciences, 335(3(1)), 330-334. <https://doi.org/10.31891/2307-5732-2024-335-3-44>
26. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. — 3rd ed. — Wiley, 2013. — 608 p. [https://www.academia.edu/24311950/The Data Warehouse Toolkit](https://www.academia.edu/24311950/The_Data_Warehouse_Toolkit)
27. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. — 12th ed. — Wiley, 2017. — 1200 p. [https://www.academia.edu/123134596/Project management a systems approach to planning scheduling and controlling](https://www.academia.edu/123134596/Project_management_a_systems_approach_to_planning_scheduling_and_controlling)
28. Brynjolfsson E., McAfee A. The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies. — W. W. Norton & Company, 2014. — 304 p. <http://digamo.free.fr/brynmacafee2.pdf>
29. Christopher M. Logistics & Supply Chain Management: Strategy, Operation and Practice. — 5th ed. — Pearson, 2016. — 312 p. [https://archive.org/details/logisticssupplyc0000chri\\_w2t5](https://archive.org/details/logisticssupplyc0000chri_w2t5)
30. Chopra S., Meindl P. Supply Chain Management: Strategy, Planning, and Operation. — 7th ed. — Pearson, 2019. — 544 p. <https://ru.scribd.com/document/901537984/Supply-Chain-Management-Strategy-Planning-And-Operation-7th-Edition-Chopra-Solutions-Manual-pdf-download>
31. Waters D. Global Logistics: New Directions in Supply Chain Management. — Kogan Page, 2019. — 320 p. <https://ftp.idu.ac.id/wp-content/uploads/ebook/ip/LOGISTIK%20GLOBAL/Global%20Logistics%20New%20Directions%20in%20Supply%20Chain%20Management%20074944813X.pdf>
32. Yao F., Zhao W., Forshaw M., Song Y. A Holistic Power Optimization Approach for Microgrid Control Based on Deep Reinforcement Learning. — arXiv:2403.01013, 2024. <https://www.semanticscholar.org/paper/A-Holistic-Power-Optimization-Approach-for-Control-Yao-Zhao/49a63ba6f2818a04e3908dc6b2a2efc639025c72>
33. Wang Z., Yu D., Wu Z. Real-Time Machine-Learning-Based Optimization Using Input Convex Long Short-Term Memory Network. — arXiv:2311.07202, 2023. <https://www.semanticscholar.org/paper/Real-time-machine-learning-based-optimization-using-Wang-Yu/12d525567e4a74cb2389f6419475fb26abef94a8>

34. Ruddick J., Ramirez Camargo L., Putratama M. A., Messagie M., Coosemans T. TreeC: A method to generate interpretable energy management systems using a metaheuristic algorithm. — arXiv:2304.08310, 2023.
35. Cauz M., Bolland A., Wyrsh N., Ballif C. Reinforcement Learning for Efficient Design and Control Co-optimisation of Energy Systems. — arXiv:2406.19825, 2024. [https://www.researchgate.net/publication/383122758\\_Reinforcement\\_Learning\\_for\\_Efficient\\_Design\\_and\\_Control\\_Co-optimisation\\_of\\_Energy\\_Systems](https://www.researchgate.net/publication/383122758_Reinforcement_Learning_for_Efficient_Design_and_Control_Co-optimisation_of_Energy_Systems)
36. Ng A. Machine Learning Yearning. — deeplearning.ai, 2018. — (electronic resource). <https://home-wordpress.deeplearning.ai/wp-content/uploads/2022/03/andrew-ng-machine-learning-yearning.pdf>
37. Armstrong M. Handbook of Human Resource Management Practice. — Kogan Page, 2014. — 680 p. [https://e-uczelnia.uek.krakow.pl/pluginfile.php/604792/mod\\_folder/content/0/Armstrongs%20Handbook%20of%20Human%20Resource%20Management%20Practice\\_1.pdf](https://e-uczelnia.uek.krakow.pl/pluginfile.php/604792/mod_folder/content/0/Armstrongs%20Handbook%20of%20Human%20Resource%20Management%20Practice_1.pdf)
38. Abiteboul S., Buneman P., Suci D. Data on the Web: From Relations to Semistructured Data and XML. — Morgan Kaufmann, 2000. — 384 p. <https://homepages.dcc.ufmg.br/~%E2%80%89laender/material/Data-on-the-Web-Skeleton.pdf>
39. Nguyen T., Simchi-Levi D. Data-Driven Inventory Management in Practice. — MIT Press, 2020. — 198 p. [https://www.researchgate.net/publication/383001602\\_Data-driven\\_inventory\\_management](https://www.researchgate.net/publication/383001602_Data-driven_inventory_management)
40. Lee I., Lee K. The Internet of Things (IoT): Applications to Smart Warehousing. — IEEE Press, 2017. — 140 p. <https://www.csit.carleton.ca/~fyu/Papers/09241736.pdf>
41. Phillips W., Stewart N., Marsicano C. Android Programming: The Big Nerd Ranch Guide. — 4th ed. — Big Nerd Ranch Guides, 2020. — 432 p. <https://lmsspada.kemdiktisaintek.go.id/pluginfile.php/751912/course/section/65846/Android%20Programming%20%20The%20Big%20Nerd%20Ranch%20Guide%20%2081%29.pdf>
42. Deitel P., Deitel H. Android™ for Programmers: An App-Driven Approach. — 3rd ed. — Pearson, 2017. — 768 p. <https://ptgmedia.pearsoncmg.com/images/9780134289366/samplepages/9780134289366.pdf>

43. Skeen J., Greenhalgh D. Kotlin Programming: The Big Nerd Ranch Guide. — Big Nerd Ranch Guides, 2018. — 320 p.  
[https://ptgmedia.pearsoncmg.com/images/9780135161630/samplepages/9780135161630\\_Sample.pdf](https://ptgmedia.pearsoncmg.com/images/9780135161630/samplepages/9780135161630_Sample.pdf)
44. Artaud P.-Y. Mobile Application Development with Kotlin. — Apress, 2019. — 256 p.  
<https://dl.ebooksworld.ir/motoman/Apress.Learn.Kotlin.for.Android.Development..www.EBooksWorld.ir.pdf>
45. Horton S. Beginning Android 11 Development. — Wrox, 2021. — 480 p.  
<https://mituniversityindia.edu.in/academics/ug-programs/school-of-science-and-technology/include/grid/BCA-Syllabus.pdf>
46. Tan H. Cloud-Based Inventory Management: Architecture and Best Practices. — Apress, 2020. — 256 p.  
[https://www.researchgate.net/publication/360087905\\_Development\\_of\\_a\\_Cloud-Based\\_Inventory\\_Management\\_System](https://www.researchgate.net/publication/360087905_Development_of_a_Cloud-Based_Inventory_Management_System)
47. Newman M. Human-Computer Interaction in Mobile Interfaces. — Springer, 2018. — 220 p. [https://www.researchgate.net/publication/262186759\\_Mobile\\_Human-Computer\\_Interaction](https://www.researchgate.net/publication/262186759_Mobile_Human-Computer_Interaction)
48. Shapiro J. Mobile User Experience: Patterns to Make Sense of it All. — Morgan Kaufmann, 2015. — 256 p. <https://www.slideshare.net/slideshow/mobile-user-experience-patterns-to-make-sense-of-it-all-adrian-mendoza/280372442>
49. Bronson K. Supply Chain Metrics and KPIs: Performance Measurement in Warehouse Operations. — Kogan Page, 2017. — 240 p.  
<https://ru.scribd.com/document/91230775/Key-Performance-Indicators-KPIs-Warehousing>
50. Raman A. Barcode and Data Capture Technology in Retail. — Elsevier, 2015. — 198 p.
51. Vollmer T. Point of Sale Systems and Retail Technology. — Routledge, 2019. — 220 p.
52. Alvarez C. Enterprise Resource Planning: Concepts and Practice. — Springer, 2020. — 340 p.
53. Fitzgerald B., Russo N. Agile Methods and Practices in Software Development. — Springer, 2018. — 300 p.

54. Martin R. C. Agile Software Development: Principles, Patterns, and Practices. — Pearson, 2003. — 752 p. <https://www.ndl.ethernet.edu.et/bitstream/123456789/41034/1/82.pdf>
55. ISO 9001:2015 Quality management systems — Requirements. — ISO, 2015. [https://www.nerldc.in/wp-content/uploads/ISO\\_9001\\_2015\\_QMS.pdf](https://www.nerldc.in/wp-content/uploads/ISO_9001_2015_QMS.pdf)
56. ISO 28000:2007 Specification for security management systems for the supply chain. — ISO, 2007. <https://cdn.standards.iteh.ai/samples/44641/fd875afdde654325bbe5c45268a0ab7b/ISO-28000-2007.pdf>
57. ISO 15489-1:2016 Information and documentation — Records management — Part 1: Concepts and principles. — ISO, 2016. <https://static1.squarespace.com/static/5a1c710fbce17620f861bf47/t/5a45d41353450a6f05e9b138/1514525716795/ISO+15489-1-2016.pdf>
58. European Commission. Digital Single Market: e-Procurement and e-Invoicing guidelines. — Publications Office of the EU, 2019. <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019R1780>
59. World Bank. Logistics Performance Index: Global Rankings. — World Bank, 2020. <https://lpi.worldbank.org/>
60. Oracle. Android Developer Documentation — Android Studio and SDK guides. — Oracle/Google, 2024. <https://developer.android.com/develop?hl=ru>
61. Owens M. SQLite Database System: Design and Applications. — Apress, 2018. — 312 p. <https://sqlite.org/books.html>
62. Михайленко Павло, Серeda Катерина. Розробка автоматизованої системи обліку складських товарів та підтримки торгівлі. Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації. «Молодь в науці: дослідження, проблеми, перспективи» <https://conferences.vntu.edu.ua/index.php/mn/mn2026>

## ДОДАТКИ

**ДОДАТОК А**  
**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Назва роботи: Розробка автоматизованої системи обліку складських товарів і підтримки торговельної діяльності. Частина 1. Створення бази даних та розробка десктопного програмного забезпечення.

Тип роботи: магістерська кваліфікаційна робота  
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра КСУ  
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) 11,80 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту.
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

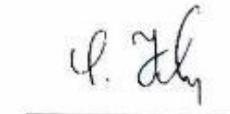
Ковтун В.В., завідувач кафедри КСУ

(прізвище, ініціали, посада)



Ковалюк О.О., доцент кафедри КСУ

(прізвище, ініціали, посада)



(підпис)

Особа, відповідальна за перевірку 

(підпис)

Дубовой В.М.

(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 

(підпис)

Юхимчук М.С., професор кафедри КСУ

(прізвище, ініціали, посада)

Здобувач: 

(підпис)

Михайленко П.О.

(прізвище, ініціали)

ДОДАТОК Б  
(обов'язковий)  
ВНТУ

ЗАТВЕРДЖЕНО

Завідувач кафедри КСУ

д.т.н., проф. Вячеслав КОВТУН

  
\_\_\_\_\_

“ 17 ” ЖОВТНЯ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання комплексної магістерської кваліфікаційної роботи  
Розробка автоматизованої системи обліку складських товарів і підтримки  
торговельної діяльності. Частина 1. Розробка бази даних і десктопного додатку.

08-33.МКР.10.00.000 ТЗ

Керівник роботи:

д.т.н., професор кафедри КСУ

  
Марія ЮХИМЧУК

“ 16 ” ЖОВТНЯ 2025р.

Виконавець:

ст. гр. Павло МИХАЙЛЕНКО

  
“ 16 ” ЖОВТНЯ 2025р.

## 1. Назва та галузь застосування

1.1. Назва – Розробка автоматизованої системи обліку складських товарів і підтримки торговельної діяльності. Частина 1. Розробка бази даних і десктопного додатку.

1.2. Галузь застосування – Комп'ютеризовані системи автоматизації управлінської діяльності.

## 2. Підстава для проведення розробки.

Тема комплексної магістерської кваліфікаційної роботи затверджена наказом по ВНТУ №313 від 24.09.2025р.

## 3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є підвищення ефективності роботи торгової мережі за рахунок створення автоматизованої системи управління обліком складських товарів та торгівлі, що дозволить автоматизувати рутинні завдання та мінімізувати вплив помилок людського фактору на роботу підприємства, задля зменшення втрат та ефективного розподілу вивільнених робочих ресурсів на інші області роботи компанії.

## 4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. — Addison-Wesley, 2003.
2. Pressman R. Software Engineering: A Practitioner's Approach. — McGraw-Hill, 2019.
3. Kleppmann M. Designing Data-Intensive Applications. — O'Reilly Media, 2017.
4. Litvinenko O. Управління інформаційними системами підприємства. — Київ: Наукова думка, 2015.
5. Kirichenko O. Проектування баз даних: навчальний посібник. — Київ: КНТ, 2018.

## 5. Вимоги до розробки.

### 5.1. Перелік головних функцій:

- управління товарними запасами;
- управління документами та звітністю;
- управління списком контрагентів компанії;
- управління підрозділами компанії;
- прийом та відправлення розрахунків;
- здійснення логістики товарів між підрозділами компанії.

## 5.2. Основні технічні вимоги до розробки.

### 5.2.1. Вимоги до програмної платформи:

- WINDOWS 7\8\10.

### 5.2.2. Умови експлуатації системи:

- робота на стандартних ПЕОМ в приміщеннях зі стандартними умовами;
- можливість цілодобового функціонування системи;
- текст програмного забезпечення системи є цілком закритим.

## 6. Стадії та етапи розробки.

### 6.1 Пояснювальна записка:

- Дослідження актуальності поставленої задачі «4» жовтня 2025 р.
- Загальний огляд систем управління обліку складських товарів «12» жовтня 2025 р.  
Аналіз функцій автоматизованих систем управління обліку складських товарів та торгівлі та порівняння програм аналогів «20» жовтня 2025 р.
- Розробка структури програмного забезпечення системи «24» жовтня 2025 р.

### 6.2 Графічні матеріали:

- Розробка UML-діаграм системи «4» листопада 2025 р.
- Тестування програмного забезпечення «27» листопада 2025 р.

## 7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «\_30\_» \_листопада\_ 2025 р.
- 7.2. Атестація проекту здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «\_02\_» \_грудня\_ 2025 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «\_20\_» \_\_грудня\_\_ 2025 р.

## Додаток В (довідковий)

### Лістинг програмного забезпечення

```

#define STRICT
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#include <windowsx.h>

LRESULT CALLBACK WindowProc(HWND hWnd, UINT
uMsg, WPARAM wParam, LPARAM lParam);

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE
hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{
    HWND hWnd;
    MSG msg;
    WNDCLASSEX wndClass;
    wndClass.cbSize=sizeof(wndClass);
    wndClass.style=CS_CLASSDC;
    wndClass.lpfnWndProc=WindowProc;
    wndClass.cbClsExtra=0;
    wndClass.cbWndExtra=0;
    wndClass.hInstance=hInstance;
    wndClass.hIcon=LoadIcon(NULL,
IDI_APPLICATION);
    wndClass.hIconSm=LoadIcon(NULL,
IDI_APPLICATION);
    wndClass.hCursor=NULL;
    wndClass.hbrBackground=GetStockBrush(BLACK_BR
USH);
    wndClass.lpszMenuName=NULL;
    wndClass.lpszClassName="GameClass";

    if(!RegisterClassEx(&wndClass)) return FALSE;
    hWnd=CreateWindowEx(
        WS_EX_TOPMOST,
        "CompanyClass",
        "My company title",
        WS_OVERLAPPEDWINDOW, 0, 0,
        GetSystemMetrics(SM_CXSCREEN),
        GetSystemMetrics(SM_CYSCREEN),
        NULL,NULL,hInstance,NULL);
    if(hWnd==NULL) return(FALSE);
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    ZeroMemory(&msg, sizeof(MSG));

    while(msg.message != WM_QUIT)
    {
        if(PeekMessage(&msg, NULL, 0, 0,
PM_REMOVE))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }

        UnregisterClass("GameClass", hInstance);
        return (msg.wParam);
    }

    // Віконна процедура
    LRESULT CALLBACK WindowProc(HWND hWnd, UINT msg,
WPARAM wParam, LPARAM lParam)
    {
        switch(msg)
        {
            case WM_DESTROY: // У разі отримання цього
повідомлення...
                PostQuitMessage(0); // Повідомляємо
Windows про завершення роботи застосунку
                break;
            // У разі отримання будь-яких інших
повідомлень...
            default:
                return DefWindowProc(hWnd, msg, wParam,
lParam);
        }
        return 0;
    }
    <...>
    wndClass.hCursor = LoadCursor(NULL,
    IDC_ARROW); // Курсор при наведенні на вікно
    wndClass.hbrBackground =
    GetStockBrush(BLACK_BRUSH);
    wndClass.lpszMenuName = NULL;
    wndClass.lpszClassName = "GameClass";
    <...>
    int WINAPI WinMain(
        HINSTANCE hInst, // Ідентифікатор екземпляра
вікна (handle екземпляра)
        HINSTANCE hPrevInst, // Ідентифікатор
попереднього екземпляра вікна
        LPSTR lpzCmdLine, // Командний рядок
        int nCmdShow // Параметр відображення вікна
    );

    int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE
hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
    {
        HWND hWnd; // Дескриптор вікна
        MSG msg; // Структура повідомлення
        WNDCLASSEX wndClass; // Оголошення структури
класу вікна WNDCLASSEX

        wndClass.cbSize = sizeof(wndClass);
        wndClass.style = CS_CLASSDC;
        wndClass.lpfnWndProc = WindowProc; // Віконна
процедура
        wndClass.cbClsExtra = 0;
        wndClass.cbWndExtra = 0;
        wndClass.hInstance = hInstance; // Екземпляр
застосунку
        wndClass.hIcon = LoadIcon(NULL,
IDI_APPLICATION);
        wndClass.hIconSm = LoadIcon(NULL,
IDI_APPLICATION);
        wndClass.hCursor = NULL; // Курсор при
наведенні на вікно
        wndClass.hbrBackground=GetStockBrush(BLACK_BR
USH);
        wndClass.lpszMenuName=NULL;
        // Дескриптор
Головного меню вікна
        wndClass.lpszClassName="GameClass"; // Називаєм
віконний клас.
    }
    /* Прототип структури даних WNDCLASSEX */

```

```

typedef struct _WNDCLASSEX
{
    UINT    cbSize;        // Розмір структури
    UINT    style;        // Стил класу вікна
    WNDPROC lpfnWndProc;  // Вказівник на
віконну процедуру
    int     cbClsExtra;   // Додаткові байти
для класу
    int     cbWndExtra;   // Додаткові
байти для вікна
    HANDLE  hInstance;   // Дескриптор
екземпляра застосунку
    HICON   hIcon;       // Значок вікна
    HCURSOR hCursor;     // Курсор миші
    HBRUSH  hbrBackground; // Фон вікна
    LPCSTR  lpszMenuName; // Назва меню
    LPCSTR  lpszClassName; // Назва класу вікна
    HICON   hIconSm;     // Малий значок вікна
} WNDCLASSEX;
/* Структура класу вікна DirectX-застосунку */
WNDCLASSEX wcex = {
    sizeof(WNDCLASSEX),
    CS_CLASSDC,
    WindowProc,
    0L,
    0L,
    hInstance,
    NULL,
    NULL,
    NULL,
    NULL,
    "GameClass",
    NULL
};
WNDCLASSEX wcex2 = {
    sizeof(WNDCLASSEX),
    CS_HREDRAW | CS_VREDRAW,
    WindowProc,
    0L,
    0L,
    hInstance,
    LoadIcon(NULL, IDI_APPLICATION),
    LoadCursor(NULL, IDC_ARROW),
    (HBRUSH)GetStockObject(BLACK_BRUSH),
    NULL,
    "AppClass",
    NULL
};
/* Прототип функції CreateWindowEx */
HWND CreateWindowEx(
    DWORD dwExStyle, // Додатковий стиль вікна
    LPCSTR lpClassName, // Назва класу вікна (у
нашому випадку GameClass)
    LPCSTR lpWindowName, // Назва вікна
    DWORD dwStyle, // Основний стиль вікна
    int x, int y, // Координати
розташування вікна
    int nWidth, int nHeight, // Ширина та
висота вікна
    HWND hWndParent, // Дескриптор
батьківського вікна (за наявності)
    HMENU hMenu, // Дескриптор меню (за
наявності)
    HINSTANCE hInstance, // Дескриптор екземпляра
застосунку
    LPVOID lpParam // Додаткові параметри
);
// Створення вікна на основі зареєстрованого класу
hWnd = CreateWindowEx(
    WS_EX_TOPMOST, // Додатковий стиль вікна
    "CompanyClass", // Клас вікна
    "My company title", // Текст заголовка (у
верхньому рядку вікна)
    WS_OVERLAPPEDWINDOW, // Стандартне
вікно з кнопками керування
    0, 0, // Координати X та Y
    GetSystemMetrics(SM_CXSCREEN), // Ширина вікна
    GetSystemMetrics(SM_CYSCREEN), // Висота вікна
    NULL, // Дескриптор батьківського
вікна
    NULL, // Дескриптор меню
    hInstance, // Дескриптор екземпляра
застосунку
    NULL // Додаткові дані
);
if (hWnd == NULL) return(FALSE);

/* Прототип функції PeekMessage */
BOOL PeekMessage(
    LPMSG lpMsg, // Вказівник на структуру MSG
    HWND hWnd, // Дескриптор вікна, якому
адресовано повідомлення
    UINT wMsgFilterMin, // Мінімальне значення
фільтра повідомлень
    UINT wMsgFilterMax, // Максимальне значення
фільтра повідомлень
    INT wRemoveMsg // Дія з повідомленням
);
/* Прототип функції SendMessage */
LRESULT SendMessage(
    HWND hWnd, // Дескриптор вікна
    UINT Msg, // Код повідомлення
    WPARAM wParam, // Перший параметр
    LPARAM lParam // Другий параметр
);
LRESULT PostMessage(
    HWND hWnd, // Дескриптор вікна
    UINT Msg, // Код повідомлення
    WPARAM wParam, // Перший параметр
    LPARAM lParam // Другий параметр
);
LRESULT CALLBACK WindowProc(HWND hWnd, UINT msg,
WPARAM wParam, LPARAM lParam)
{
    switch (msg)
    {
        case WM_CREATE:
            if(!SetTimer(hWnd, TIMER_ID,
TIMER_RATE, NULL))
                return (-1L);
            return (0L);
        case WM_TIMER:
            MessageBeep(0xFFFFFFFF);
            return (0L);
        case WM_DESTROY:
            KillTimer(hWnd, TIMER_ID);
            PostQuitMessage(0);
            Return (0L);
        default:
            return DefWindowProc(hWnd, msg, wParam,
lParam);
    }
}
//Видаляємо реєстрацію класу
UnregisterClass("GameClass", hInstance);

// Виходимо з застосунк
return (msg, wParam);

```

модуль Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

namespace Oblik01

```
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the
                'accountDataSet.Account' table. You can move, or remove
                it, as needed.

                this.accountTableAdapter.Fill(this.accountDataSet.Account
                );
                // TODO: This line of code loads data into the
                'accountDataSet.Worker' table. You can move, or remove it,
                as needed.
```

```
this.workerTableAdapter.Fill(this.accountDataSet.Worker); // відображення
    }
}
```

```
private void dataGridView1_Click(object sender,
EventArgs e)
{
```

// створити фільтр, який виведе в dataGridView2 Видалення користувача  
те що потрібно

```
int id;
int index;

index = dataGridView1.CurrentRow.Index;
// взяти значення id = Worker.ID_Worker
id
(int)dataGridView1.Rows[index].Cells[0].Value; // працює
accountBindingSource.Filter = "ID_Worker = " +
id.ToString();
}
```

```
private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
{
```

// створити фільтр, який виведе в dataGridView2  
те що потрібно

```
int id;
int index;

index = dataGridView1.CurrentRow.Index;

// взяти значення id = Worker.ID_Worker
id
(int)dataGridView1.Rows[index].Cells[0].Value; // працює
```

```
accountBindingSource.Filter = "ID_Worker = " +
id.ToString();
}
}
```

Модуль створення користувача

```
private void addToolStripMenuItem_Click(object sender,
EventArgs e)
{
    FormAddWorker f = new FormAddWorker(); // створити
    форму
```

```
if (f.ShowDialog() == DialogResult.OK) // відобразити
форму
{
    // якщо ОК, то додати робітника
    string WName, WPosition, WSex;
```

```
WName = f.textBox1.Text;
WPosition = f.textBox2.Text;
WSex = f.comboBox1.Items[f.comboBox1.SelectedIndex].ToString
();
```

```
// працює
this.workerTableAdapter.Insert(WName, WPosition,
WSex); // вставка
```

```
this.workerTableAdapter.Fill(this.accountDataSet.Worker);
}
}
```

```
private void deleteToolStripMenuItem_Click(object sender,
EventArgs e)
{
```

// Worker => Delete  
FormDelWorker f = new FormDelWorker(); // створити  
форму

```
int id_worker;
string WName, WPosition, WSex;
int index;
```

```
// взяти номер поточної (виділеного) рядка в
dataGridView1
index = dataGridView1.CurrentRow.Index;
```

// заповнити внутрішні змінні з поточного рядка  
dataGridView1

```
id_worker = Convert.ToInt32(dataGridView1[0,
index].Value);
WName = Convert.ToString(dataGridView1[1,
index].Value);
WPosition = Convert.ToString(dataGridView1[2,
index].Value);
WSex = Convert.ToString(dataGridView1[3,
index].Value);
```

```

// сформувати інформаційний рядок
f.label2.Text = WName + " " + WPosition;

if (f.ShowDialog() == DialogResult.OK)
{
    workerTableAdapter.Delete(id_worker, WName, WPosition, WSex); // метод Delete
}

this.workerTableAdapter.Fill(this.accountDataSet.Worker);
}
}

Редагування користувача

private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    // Worker => Edit...
    FormEditWorker f = new FormEditWorker(); // создать форму

    int index;
    string WName, WPosition, WSex;
    int id_worker;

    if (dataGridView1.RowCount <= 1) return;

    // получить позицию ряда в dataGridView1
    index = dataGridView1.CurrentRow.Index;

    if (index == dataGridView1.RowCount - 1) return; //

    // отримати дані рядка
    id_worker = (int)dataGridView1.Rows[index].Cells[0].Value;
    WName = (string)dataGridView1.Rows[index].Cells[1].Value;
    WPosition = (string)dataGridView1.Rows[index].Cells[2].Value;
    WSex = (string)dataGridView1.Rows[index].Cells[3].Value;

    // заповнити поля форми f
    f.textBox1.Text = WName;
    f.textBox2.Text = WPosition;

    if (WSex == "M") f.comboBox1.SelectedIndex = 0;
    else f.comboBox1.SelectedIndex = 1;

    if (f.ShowDialog() == DialogResult.OK) // викликати форму FormEditWorker
    {
        string nWName, nWPosition, nWSex;

        // отримати нові значення з форми
        nWName = f.textBox1.Text;
        nWPosition = f.textBox2.Text;
        nWSex = f.comboBox1.Items[f.comboBox1.SelectedIndex].ToString();

        // зробити зміни в адаптері
        this.workerTableAdapter.Update(nWName, nWPosition, nWSex, id_worker, WName, WPosition, WSex);

        this.workerTableAdapter.Fill(this.accountDataSet.Worker);
    }
}

Модуль створення складу

private void addToolStripMenuItem1_Click(object sender, EventArgs e)
{
    // Inventory => Add...
    FormAddAccount f = new FormAddAccount(); // окно

    if (f.ShowDialog() == DialogResult.OK)
    {
        int Id_Worker;
        int index;
        string ANum, ADate, AInvNum, AObjName;
        short ACount;
        double APrice;

        // взяти значення ID_Worker з таблиці Worker в dataGridView1
        Id_Worker = (int)dataGridView1.Rows[index].Cells[0].Value;

        // взяти значення інших полів з форми FormAddAccount
        ANum = f.textBox1.Text;
        ADate = f.textBox2.Text;
        AInvNum = f.textBox3.Text;
        AObjName = f.textBox4.Text;
        ACount = Convert.ToInt16(f.textBox5.Text);
        APrice = Convert.ToDouble(f.textBox6.Text);

        this.accountTableAdapter.Insert(Id_Worker, ANum, ADate, AInvNum, AObjName, ACount, APrice);

        this.accountTableAdapter.Fill(this.accountDataSet.Account);
    }
}

Видалення складу

private void deleteToolStripMenuItem1_Click(object sender, EventArgs e)
{
    // Inventory => Delete
    FormDelAccount f = new FormDelAccount();
    int id_account, id_worker;
    string ANum, ADate, AInvNum, AObjName;
    short ACount;
    double APrice;
    int index;

    // взяти індекс поточного рядка в dataGridView2
}

```

```

index = dataGridView2.CurrentRow.Index;
// взяти значення полів рядка з номером index
id_account = Convert.ToInt32(dataGridView2[0, index].Value);
id_worker = Convert.ToInt32(dataGridView2[1, index].Value);
ANum = Convert.ToString(dataGridView2[2, index].Value);
ADate = Convert.ToString(dataGridView2[3, index].Value);
AInvNum = Convert.ToString(dataGridView2[4, index].Value);
AObjName = Convert.ToString(dataGridView2[5, index].Value);
ACount = Convert.ToInt16(dataGridView2[6, index].Value);
APrice = Convert.ToDouble(dataGridView2[7, index].Value);

// сформувати інформаційний рядок у вікні
FormDelAccount
f.label2.Text = ANum + " / " + ADate + " / " + AInvNum
+ " / " +
AObjName + " / " + ACount + " / " + APrice;

if (f.ShowDialog() == DialogResult.OK) // вивести вікно
{
    this.accountTableAdapter.Delete(id_account,
id_worker,
ANum, ADate, AInvNum, AObjName, ACount,
APrice); // видалити рядок

this.accountTableAdapter.Fill(this.accountDataSet.Account
); // зафіксувати зміну
}
}

Редагування складу

private void editToolStripMenuItem1_Click(object sender,
EventArgs e)
{
    // Inventory => Edit...
    FormEditAccount f = new FormEditAccount();

    int index;
    int id_account;
    int id_worker;
    string ANum, ADate, AInvNum, AObjName;
    short ACount;
    double APrice;

    if (dataGridView2.RowCount <= 1) return;

    // взяти номер рядка в dataGridView2
    index = dataGridView2.CurrentRow.Index;

    if (index == dataGridView2.RowCount-1) return; //

    // отримати дані рядка
    id_account
(int)dataGridView2.Rows[index].Cells[0].Value;

```

```

id_worker =
(int)dataGridView2.Rows[index].Cells[1].Value;
ANum =
(string)dataGridView2.Rows[index].Cells[2].Value;
ADate =
(string)dataGridView2.Rows[index].Cells[3].Value;
AInvNum =
(string)dataGridView2.Rows[index].Cells[4].Value;
AObjName =
(string)dataGridView2.Rows[index].Cells[5].Value;
ACount =
(short)dataGridView2.Rows[index].Cells[6].Value;
APrice =
(double)dataGridView2.Rows[index].Cells[7].Value;

// заповнити дані рядка в FormEditAccount
f.textBox1.Text = ANum;
f.textBox2.Text = ADate;
f.textBox3.Text = AInvNum;
f.textBox4.Text = AObjName;
f.textBox5.Text = Convert.ToString(ACount);
f.textBox6.Text = Convert.ToString(APrice);

// заповнити поле форми FormEditAccount
if (f.ShowDialog() == DialogResult.OK) // визвати
форму
{
    // нові значення рядка
    string nANum, nADate, nAInvNum, nAObjName;
    short nACount;
    double nAPrice;

    // взяти нові значення
    nANum = f.textBox1.Text;
    nADate = f.textBox2.Text;
    nAInvNum = f.textBox3.Text;
    nAObjName = f.textBox4.Text;
    nACount = Convert.ToInt16(f.textBox5.Text);
    nAPrice = Convert.ToDouble(f.textBox6.Text);

    // оновити дані в рядку
    this.accountTableAdapter.Update(id_worker, nANum, nADate,
nAInvNum, nAObjName, nACount, nAPrice,
id_account, id_worker,
ANum, ADate, AInvNum, AObjName, ACount, APrice);

    this.accountTableAdapter.Fill(this.accountDataSet.Account
);
}

Програмування обробки події PrintPage

// Обробник події PrintPage – тут потрібно програмувати
друк
private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
{
    // Створити шрифт myFont
    Font myFont = new Font("Arial", 14, FontStyle.Regular,
GraphicsUnit.Pixel);

```

```

string curLine; // поточний рядок, що виводиться
// Відступи всередині сторінки
float leftMargin = e.MarginBounds.Left; // лівий відступ у
документі
float topMargin = e.MarginBounds.Top; // верхній відступ
у документі
float yPos = 0; // поточна позиція Y для
виведення рядка

int nPages; // кількість сторінок
int nLines; // максимально можлива кількість рядків на
сторінці
int i; // номер поточного рядка для виведення на
сторінці

// Обчислити максимально можливу кількість рядків на
сторінці
nLines = (int)(e.MarginBounds.Height /
myFont.GetHeight(e.Graphics));

// Обчислити кількість сторінок для друку
nPages = (richTextBox1.Lines.Length - 1) / nLines + 1;

// Цикл друку/виведення однієї сторінки
i = 0;
while ((i < nLines) && (counter <
richTextBox1.Lines.Length))
{
    // Взяти рядок для виведення з richTextBox1
    curLine = richTextBox1.Lines[counter];

    // Обчислити поточну позицію по осі Y
    yPos = topMargin + i * myFont.GetHeight(e.Graphics);

    // Вивести рядок у документ
    e.Graphics.DrawString(curLine, myFont, Brushes.Blue,
leftMargin, yPos, new StringFormat());

    counter++;
    i++;
}

// Якщо весь текст не міститься на 1 сторінку, то
// Необхідно додати додаткову сторінку для друку
e.HasMorePages = false;

if (curPage < nPages)
{
    curPage++;
    e.HasMorePages = true;
}

// Команда Page Setup
private void button1_Click(object sender, EventArgs e)
{
    pageSetupDialog1.ShowDialog(); // відобразити вікно
}

// Команда Print()
private void button2_Click(object sender, EventArgs e)
{
    if (printDialog1.ShowDialog() == DialogResult.OK)
        printDocument1.Print();
}

// Команда PrintPreviewDialog()
private void button3_Click(object sender, EventArgs e)
{
    printPreviewDialog1.ShowDialog();
}

```

Додаток Г  
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

Розробка автоматизованої системи обліку складських товарів і підтримки торговельної діяльності. Частина 1. Створення бази даних та розробка десктопного програмного забезпечення.

Перелік ілюстративних матеріалів:

1. Слайд 1 — Титульний слайд
2. Слайд 2 — Актуальність, мета і завдання дослідження
3. Слайд 3 — Основні напрямки роботи АСУ
4. Слайд 4 — Етапи процесу вибору програмного рішення
5. Слайд 5 — Структурна схема бази даних автоматизованої системи
6. Слайд 6 — Варіанти використання автоматизованої системи обліку
7. Слайд 7 — UML-діаграма діяльності процесу покупки товарів
8. Слайд 8 — Реалізація та інтерфейс розробленого програмного забезпечення
9. Слайд 9 — Реєстрація операцій та формування звітності в системі
10. Слайд 10 — Висновки

Керівник роботи:

д.т.н., професор кафедри КСУ

  
Марія ЮХИМЧУК

Виконавець:

Ст. гр. 1АКІТР-24м

  
ПАВЛО МИХАЙЛЕНКО

# РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ СКЛАДСЬКИХ ТОВАРІВ І ПІДТРИМКИ ТОРГОВЕЛЬНОЇ ДІЯЛЬНОСТІ. ЧАСТИНА 1. СТВОРЕННЯ БАЗИ ДАНИХ ТА РОЗРОБКА ДЕСКОПНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Студент 2 курсу, групи 1АКІТР-24м спеціальності  
174 – Автоматизація та комп'ютерно-інтегровані  
технології та роботехніка Павло МИХАЙЛЕНКО

Керівник: д.т.н., професор кафедри КСУ  
Марія ЮХИМЧУК

## Слайд 1 – Титульний Слайд

- **Актуальність.** Системи обліку складських товарів та торгівлі, що створюються для автоматизації процесів підприємницької діяльності складаються з безлічі функцій, що утворюють єдиний надійний комплекс з управління підприємством. Після інтеграції системи в роботу підприємства всі дії фіксуються через програмне забезпечення, а обчислення, відповідно, проводяться без помилок та впливу людського фактору.
- **Мета і завдання дослідження.** Метою магістерської кваліфікаційної роботи є підвищення ефективності роботи торгової мережі за рахунок створення автоматизованої системи управління обліком складських товарів та торгівлі, що дозволить автоматизувати рутинні завдання та мінімізувати вплив помилок людського фактору на роботу підприємства, задля зменшення втрат та ефективного розподілу вивільнених робочих ресурсів на інші області роботи компанії.
- Для досягнення наведеної мети були поставлені та вирішені наступні задачі:
- проведено аналіз існуючих програмних рішень роботи систем обліку та торгівлі;
- проведено аналіз сучасних технологій реалізації автоматизованих систем управління;
- досліджено практичні аспекти функціонування систем управління обліку складських товарів та торгівлі;
- розроблено десктопне програмне рішення автоматизованої системи управління обліком складських товарів та торгівлі, що функціонує в режимі реального часу;
- проведено дослідження та тестування отриманих результатів.
- **Об'єктом дослідження** є процес створення автоматизованої системи управління обліком складських товарів та торгівлі.
- **Предметом дослідження** є методи та засоби програмування, що забезпечують функціонування автоматизованої системи управління обліком складських товарів та торгівлі.

## Слайд 2 – Актуальність і мета дослідження

В залежності від масштабів та профілю бізнесу, автоматизовані системи обліку складських товарів та торгівлі можуть мати відмінний функціонал, проте існують головні напрямки, які повинні покриватись якісним програмним забезпеченням для найбільшої ефективності їх використання. Схематично напрямки роботи автоматизованих систем обліку товарів та торгівлі зображено на рисунку 1.1.



Рисунок 1.1 - Напрямки роботи автоматизованих систем обліку товарів та торгівлі

### Слайд 3 – Основні напрямки роботи АСУ

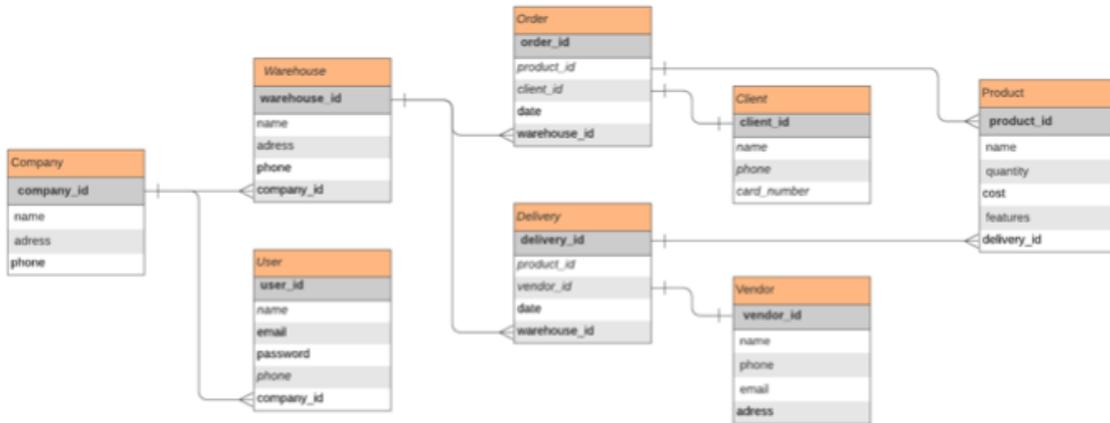
Серед сотень програмних рішень для управління складськими товарами та торгівлі керівникам компаній необхідно обрати найкращий варіант, що буде максимально доцільним та ефективним для конкретного бізнесу. Оскільки впровадження автоматизованої системи розглядається як довгострокова перспектива, що, до того ж, вимагає немалих капіталовкладень, то при виборі керуються наступними етапами, що зображені на рисунку 1.2



Рисунок 1.2 - Етапи процесу вибору найкращого програмного рішення для управління складськими товарами та торгівлі

### Слайд 4 – Етапи процесу вибору програмного рішення

На рисунку представлено структурну схему бази даних, що роз'яснює зв'язки між сутностями та містить перелік атрибутів, необхідних для утворення цих зв'язків



Структурна схема бази даних системи

Слайд 5 - Структурна схема бази даних автоматизованої системи

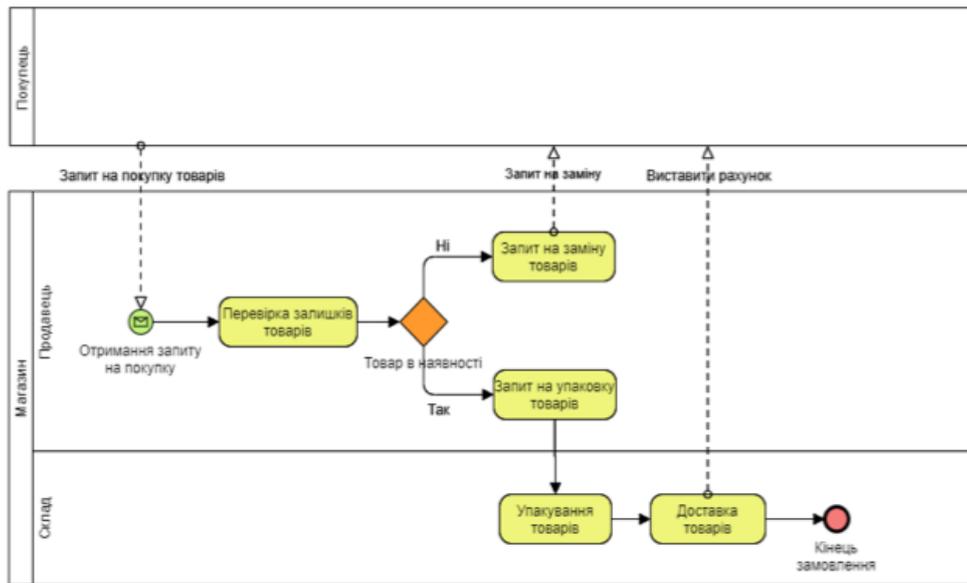
На діаграмі Користувач має п'ять головних варіантів використання, що глибше розписані на діаграмі: Авторизація, Управління списком контрагентів, Управління товарними запасами, Обслуговування клієнтів та управління документами



UML-діаграма варіантів використання

Слайд 6 - Варіанти використання автоматизованої системи обліку

### UML-діаграма, що представляє діяльність із здійснення покупки товарів покупцем магазину



UML-діаграма діяльності

### Слайд 7 - UML-діаграма діяльності процесу покупки товарів

Для розробки системи потрібні ці компоненти програми які обираються в налаштуваннях.

#### Summary

- > Visual Studio core editor
- ✓ Desktop development with C++ \*
  - Included
    - ✓ Visual C++ core desktop features
  - Optional
    - VC++ 2017 v141 toolset (x86,x64)
    - C++ profiling tools
    - Windows 10 SDK (10.0.15063.0) for Desktop C++ x8...
    - Visual C++ tools for CMake
    - Visual C++ ATL support
    - Windows 8.1 SDK and UCRT SDK
    - Windows XP support for C++
    - MFC and ATL support (x86 and x64)
    - C++/CLI support
    - Clang/C2 (experimental)
    - Modules for Standard Library (experimental)

Рисунок 3.1 – Налаштування компонентів для початку розробки

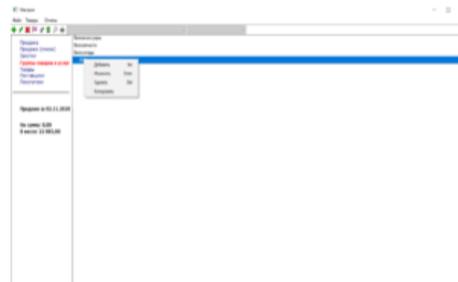


Рисунок 3.6 – Виклик контекстного меню в розділі групи товарів

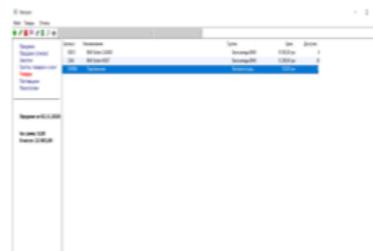


Рисунок 3.7 – Список створених карточок товарів

### Слайд 8 - Реалізація та інтерфейс розробленого програмного забезпечення



Рисунок 3.12 – Вікно реєстрації закупки та додавання товарів із списку існуючих

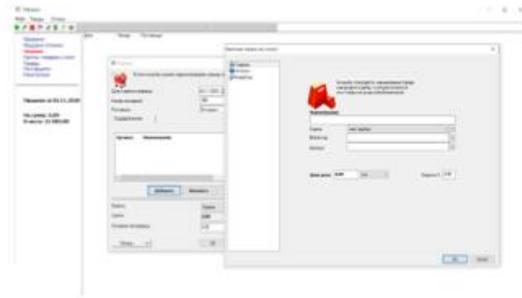


Рисунок 3.13 – Вікно реєстрації закупки та додавання товарів шляхом створення нових карток товарів

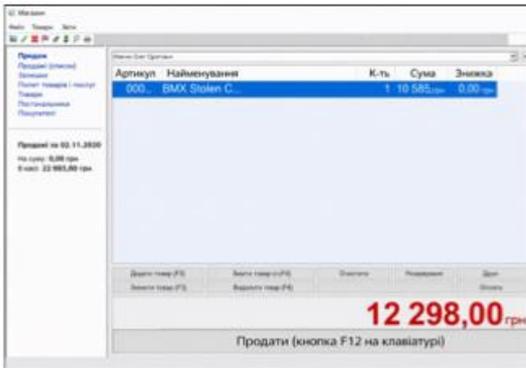


Рисунок 3.16 – Розділ реєстрації продаж

**Звіт з продажів**  
з 03.10.2020 по 02.11.2020  
Головний склад

BMX				
Артикул	Найменування	К-ть	Сума	Знижка
00012	BMX Stolen CASINO	2,000	21 170,00 грн	0,00 грн
2346	BMX Stolen HEEST	2,000	24 596,00 грн	0,00 грн
		4,000	45 766,00 грн	0,00 грн
<b>Разом:</b>		<b>4,000</b>	<b>45 766,00 грн</b>	<b>0,00 грн</b>

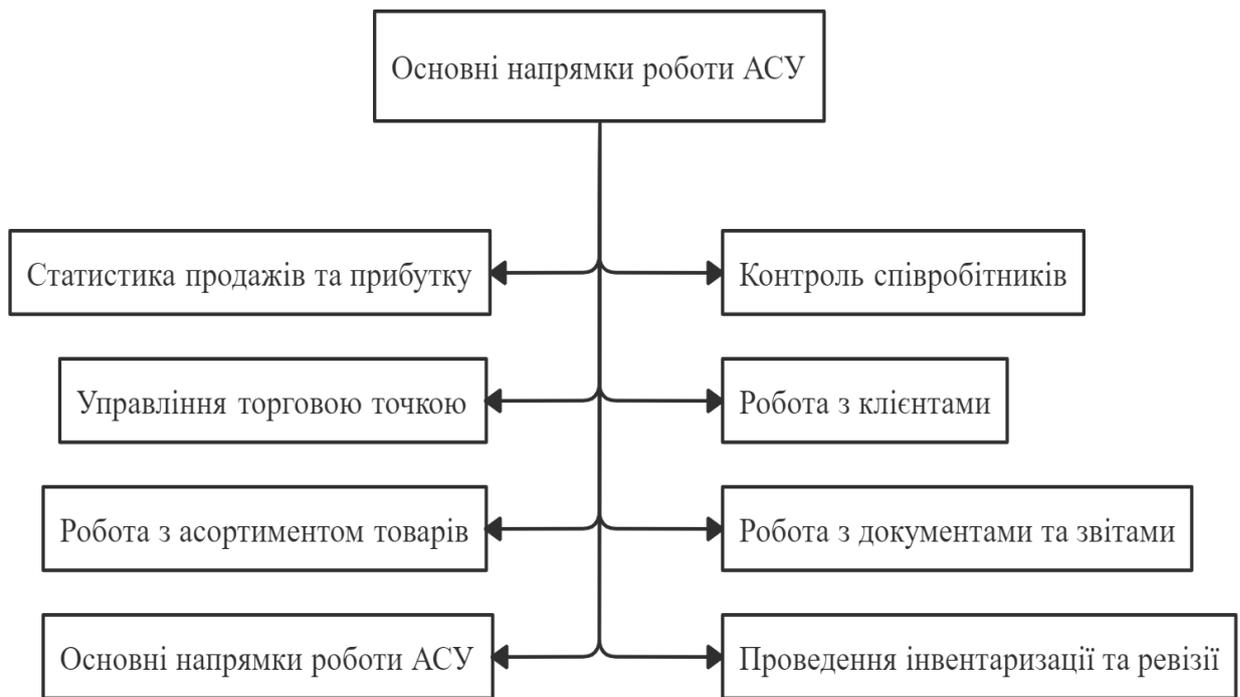
Кількість: 4,000  
Сума: 45 766,00 грн  
Знижка: 0,00 грн

Звіт про виручку по днях за обраний період

## Слайд 9 - Реєстрація операцій та формування звітності в системі

# ВИСНОВКИ

- У комплексній магістерській кваліфікаційній роботі вирішено поставлену у вступі науково-практичну задачу підвищення ефективності роботи торговельної мережі шляхом створення автоматизованої системи обліку складських товарів та підтримки торговельної діяльності. На основі аналізу предметної області та бізнес-процесів підтверджено актуальність проблеми, зумовленої відсутністю цілісного інструменту для точного, оперативного та надійного ведення складського та торговельного обліку, а також значним впливом ручних операцій і людського фактору на якість управління.
- Проведений огляд сучасних програмних рішень дозволив визначити ключові функціональні вимоги до майбутньої системи та сформувати архітектуру програмного забезпечення, оптимальну для малого та середнього бізнесу. Розроблена база даних, UML-моделі та структура застосунку забезпечують узгодженість компонентів системи й створюють основу для подальшого розширення функціональності.
- Створений десктопний застосунок реалізує автоматизацію основних операцій обліку товарів, ведення документів, управління контрагентами та формування звітності. Результати тестування підтвердили коректність роботи системи: операції заповнення бази даних, обробки інформації та відображення у вікнах програми виконуються стабільно та без збоїв. Це свідчить про працездатність та відповідність програмного продукту поставленим вимогам.
- Практична цінність роботи полягає у можливості впровадження розробленої системи на торговельних підприємствах, що дозволить підвищити точність обліку, мінімізувати вплив людських помилок, зменшити операційні втрати та ефективно перерозподіляти робочі ресурси. Подальший розвиток системи може бути спрямований на інтеграцію з онлайн-магазинами, мобільними застосунками, сервісами аналітики та торговим обладнанням.
- Отримані результати повністю підтверджують досягнення мети магістерської роботи та демонструють ефективність розробленого програмного рішення.

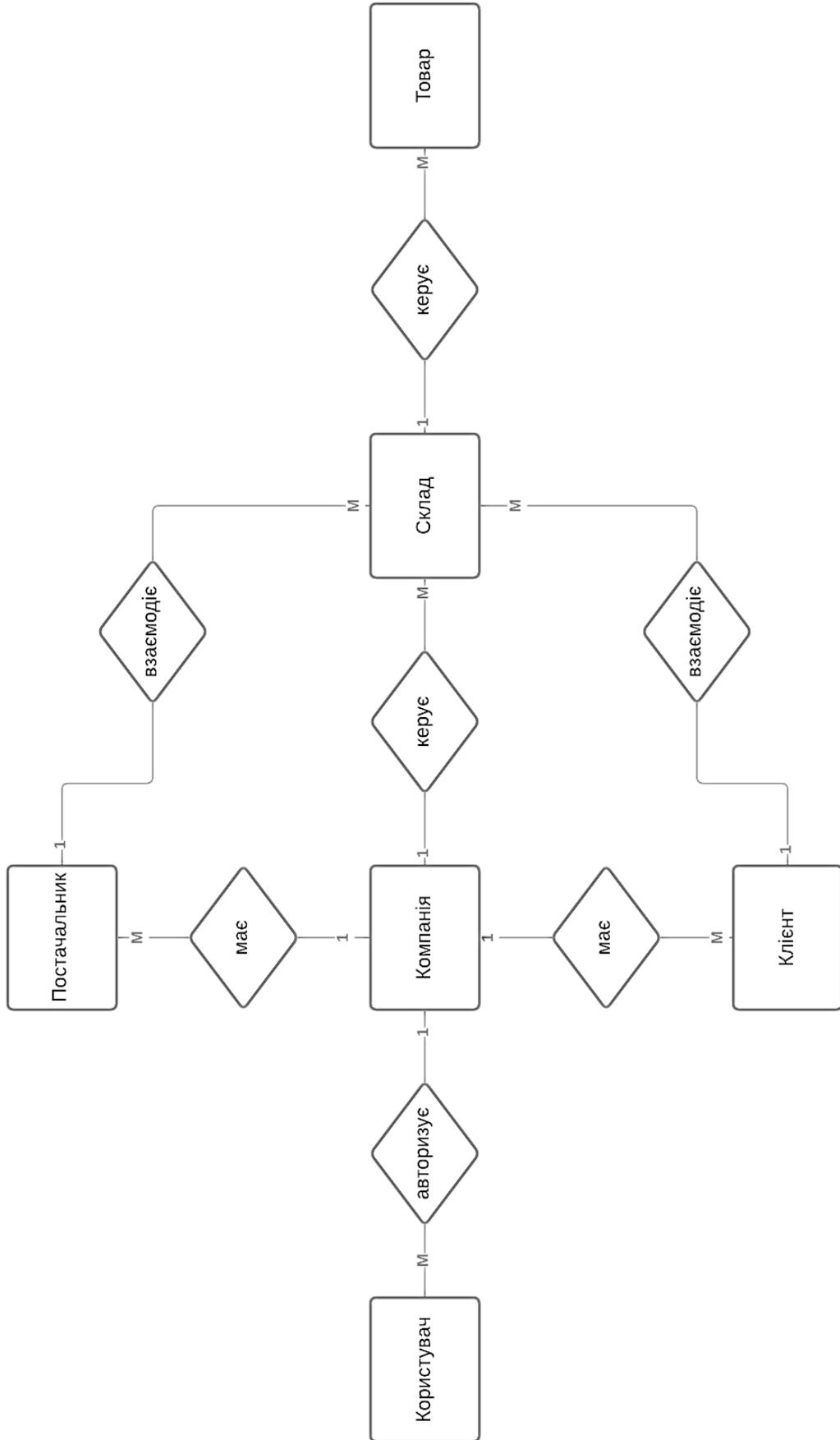


### Напрямки роботи автоматизованих систем обліку товарів та торгівлі

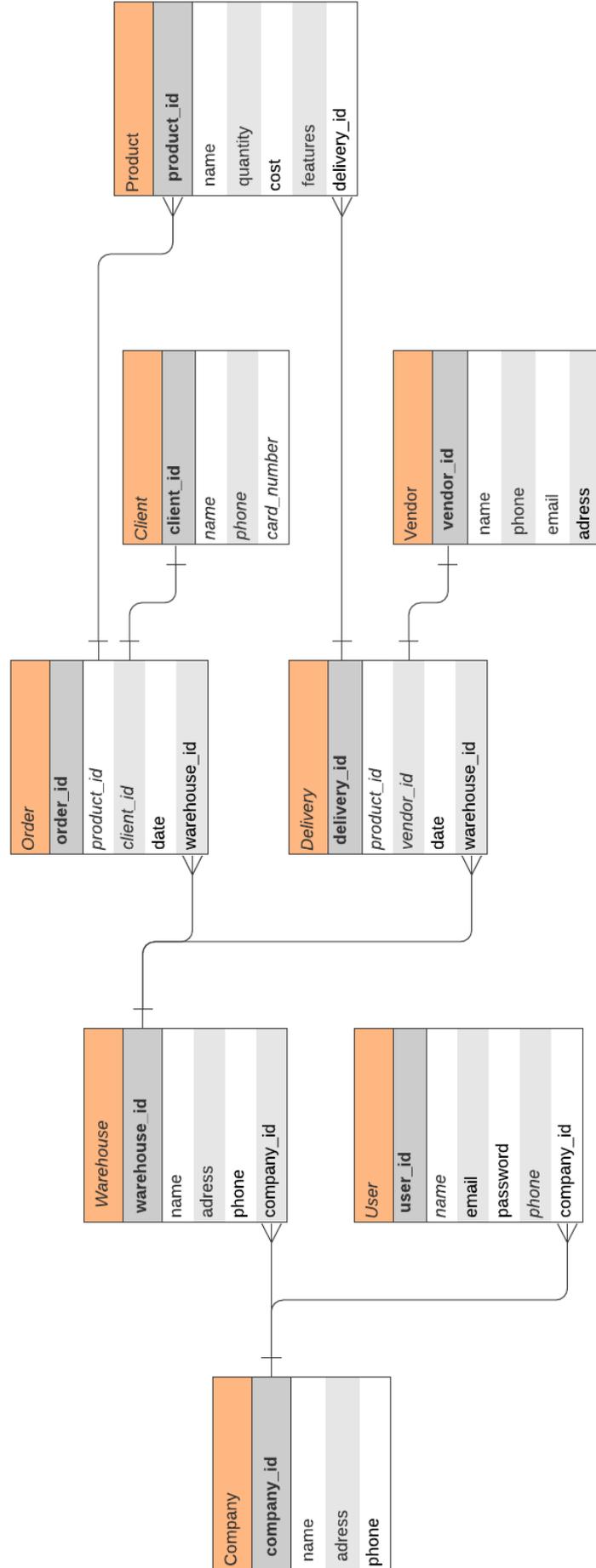


### Етапи процесу вибору найкращого програмного рішення для управління складськими товарами та торгівлі

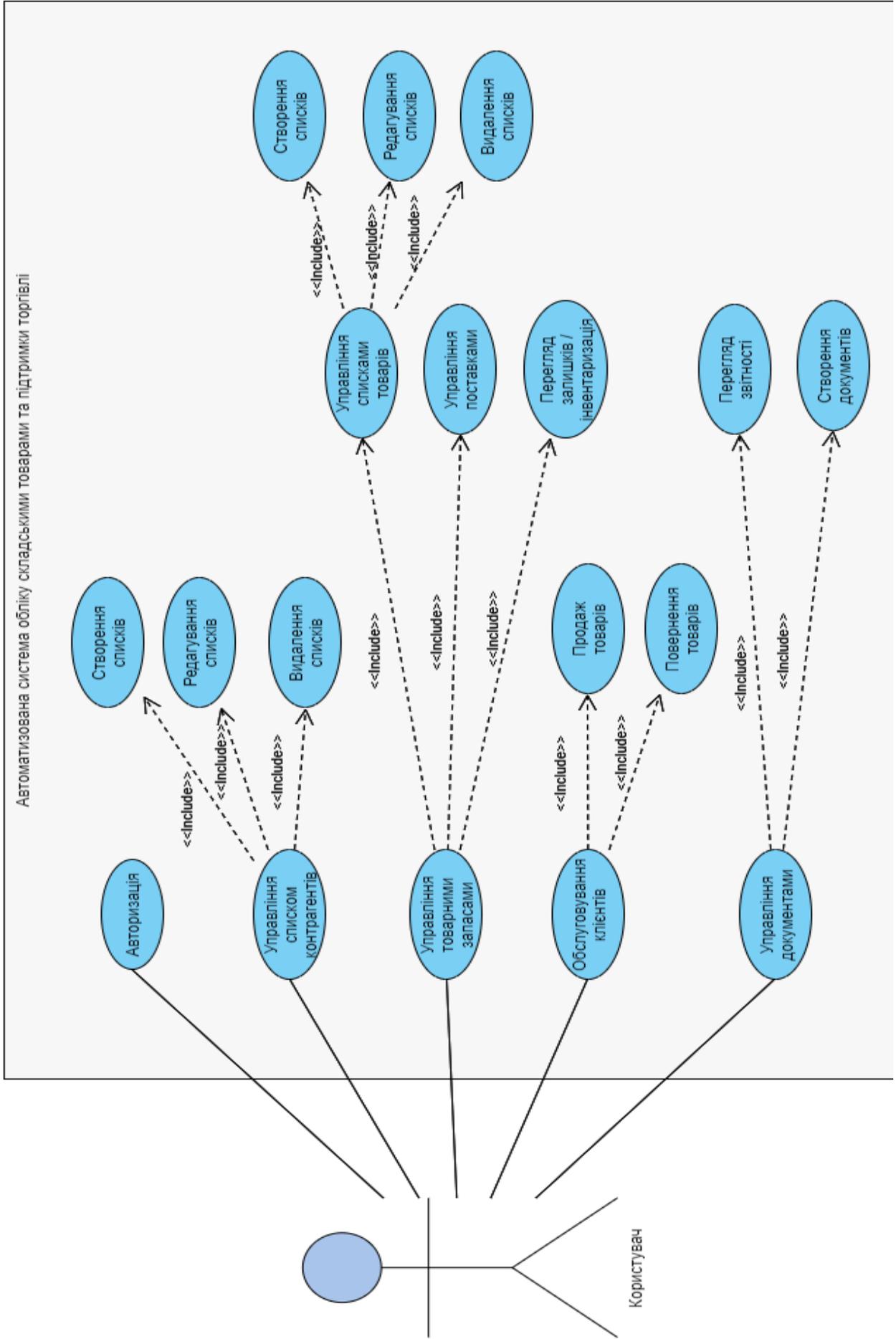
## ER-модель БД



Структурна схема бази даних системи



### UML-діаграма варіантів використання



UML-діаграма діяльності

