

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ УПРАВЛІННЯ ЗАПАСАМИ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
INVENTORY MANAGEMENT TOOLS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-133м
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

СТАРИХ Д.В.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЛПАКОВА Т.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент ІЛ'ЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

СТАРИХ Данила Володимировича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів управління запасами. Research and Software Implementation of Inventory Management Tools

керівник проєкту (роботи) к.т.н., доцент, КОЛПАКОВА Тетяна Олексіївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “07” жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЛПАКОВА Т.О., доцент		
Нормоконтроль	КАЛІНІНА М.В., асистент		

7. Дата видачі завдання “9” вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

_____ Данило СТАРИХ
(підпис) (Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Тетяна КОЛПАКОВА
(підпис) (Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
94 с., 3 табл., 37 рис., 3 дод., 23 джерела.

C#, VISUAL STUDIO, ЗАПАСИ, МОВА ПРОГРАМУВАННЯ,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СЕРЕДОВИЩЕ РОЗРОБКИ, СКЛАД.

Об'єкт дослідження – процес розробки програмного забезпечення для управління запасами.

Предмет дослідження – програмні засоби для підтримки процесу управління запасами.

Мета роботи – дослідження та розробка програмного забезпечення для управління запасами.

Матеріали, методи та технічні засоби: мова програмування C#, середовище розробки Visual Studio.

Результати. Розроблено програмне забезпечення для управління запасами за допомогою мови програмування C# та середовища розробки Visual Studio.

Практична цінність роботи полягає у розробці програмного забезпечення, що забезпечує підтримку процесу управління запасами.

Висновки. Виконано проєктування програмного забезпечення для управління запасами. Розроблено програмне забезпечення для управління запасами. Здійснено тестування розробленого програмного забезпечення для управління запасами.

Галузь використання – управління складськими запасами.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 94 pages, 3 tables, 37 figures, 3 appendixes, 23 sources.

C#, VISUAL STUDIO, RESOURCES, PROGRAMMING LANGUAGE, SOFTWARE, DEVELOPMENT ENVIRONMENT, WAREHOUSE.

The object of research is the process of developing software for inventory management.

The subject of the research is software for supporting the inventory management process.

The purpose of this work is research and development of inventory management software.

Materials, methods and technical tools: C# programming language, Visual Studio development environment.

Results. The software tools to support the inventory management process using the C# programming language and the Visual Studio development environment have been developed.

The practical value of the work lies in the development of software that provides support for the inventory management process.

Conclusions. The software for inventory management has been designed. The software for inventory management has been developed. The developed software for inventory management process has been tested.

Scope of use – management of warehouse stocks.

ЗМІСТ

С.

Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	11
1.1 Програмне забезпечення для управління запасами	11
1.2 Аналіз програмного забезпечення для управління запасами	22
1.3 Висновки за розділом 1	34
2 Матеріали і методи	36
2.1 Вибір мови програмування	36
2.2 Вибір середовища розробки для створення програмного забезпечення для управління запасами	38
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для управління запасами	40
2.4 Висновки за розділом 2	42
3 Опис програми	43
3.1 Структура програмного забезпечення для управління запасами	43
3.2 Функціонування програмного забезпечення для управління запасами	45
3.3 Використання генетичних алгоритмів для управління запасами	46
3.4 Проектування інтерфейсу програмного забезпечення для управління запасами	50
3.5 Висновки за розділом 3	55
4 Експлуатація, тестування та експериментальне дослідження програми	56
4.1 Призначення й умови застосування програми	56
4.2 Характеристики програми для управління запасами	57
4.3 Інструкція по експлуатації програми	58
4.3.1 Звернення до програми	58
4.3.2 Вхідні й вихідні дані	58
4.3.3 Повідомлення	58
4.4 Виконання програмного забезпечення для управління запасами	58

4.5 Тестування програмного забезпечення для управління запасами.....	64
4.6 Висновки за розділом 4	65
Висновки	66
Перелік джерел посилань	69
Додаток А Технічне завдання	72
Додаток Б Фрагмент тексту програми	76
Додаток В Слайди презентації	87

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ERP – Enterprise Resource Planning;

JIT – Just in Time;

IMS – Inventory Management Software;

MRP – Materials Requirement Planning;

БД – база даних;

ПЗ – програмне забезпечення;

ПЗУЗ – програмного забезпечення для управління запасами;

УЗ – управління запасами.

ВСТУП

Актуальність програмного забезпечення для управління запасами (Inventory Management Software, IMS) зумовлена постійними змінами в бізнес-середовищі, особливо в умовах глобалізації, електронної комерції та зростаючої потреби в автоматизації операцій [1]-[3].

Програмні засоби управління запасами допомагають автоматизувати процеси замовлення, зберігання та контролю продукції, зменшуючи людські помилки та підвищуючи ефективність [4]-[6].

Зростання кількості онлайн магазинів та багатоканальної торгівлі вимагає інтегрованих програмних рішень для управління запасами, що можуть відслідковувати наявність товару в реальному часі, синхронізувати дані з кількома платформами продажу та забезпечувати безперебійне обслуговування клієнтів. Деякі програмні засоби управління запасами дозволяють точно прогнозувати потреби в запасах, уникати надмірних або недостатніх запасів, що скорочує витрати на зберігання та запобігає втратам через недостачу товару [1]-[6].

Програмне забезпечення для управління запасами часто має функції аналітики та прогнозування, що дозволяють приймати ефективні рішення, керовані даними. Це допомагає підприємствам ефективно управляти ланцюгами постачання та прогнозувати майбутні потреби [1]-[6].

Багато рішень для управління запасами інтегруються з іншими системами управління бізнесом, такими як системи управління ресурсами підприємства (Enterprise Resource Planning, ERP), що сприяє кращій координації між різними бізнес-процесами, зокрема виробництвом, логістикою та бухгалтерією [1]-[6].

Проте деякі програми для управління запасами можуть бути досить дорогими, особливо для малого та середнього бізнесу. Деякі програмні засоби для управління запасами потребують складного налаштування та адаптації під конкретні бізнес-процеси. Інші програмні системи для управління запасами

можуть не бути достатньо гнучкими для масштабування, що у деяких випадках може вимагати оновлення або повної заміни програмного забезпечення, що супроводжується додатковими витратами та зусиллями працівників підприємства. Крім того, деякі програми для управління запасами можуть не враховувати унікальні вимоги або специфіку конкретного бізнесу [1]-[6].

Тому актуальною є розробка програмного забезпечення для управління запасами. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання дослідження та розробка програмного забезпечення для управління запасами.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для управління запасами;
- розробити модель взаємодії користувача з програмним забезпеченням для управління запасами;
- здійснити проєктування програмного забезпечення для управління запасами;
- створити програмне забезпечення для управління запасами;
- виконати тестування розробленого програмного забезпечення для управління запасами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для управління запасами

Управління запасами допомагає компаніям приймати обґрунтовані рішення про те, що і в яких кількостях варто зберігати. Відстежується весь цикл запасів, від придбання до продажу. Цей процес забезпечує своєчасну ідентифікацію наявних запасів та відповідну реакцію на зміни, щоб завжди мати достатню кількість товарів для виконання замовлень клієнтів і вчасно попереджати про можливі нестачі [1]-[8].

Продаж запасів приносить дохід, але до моменту продажу запаси, навіть якщо вони враховані на балансі, заморожують кошти. Внаслідок цього існують різні методи оптимізації витрат у процесі управління запасами та збереження ліквідності [2]-[4].

Одним із ключових показників ефективності управління запасами є їх оборотність. Це показник того, як часто товари продаються за визначений період часу. Продавець завжди прагне мати оптимальну кількість запасів для продажу. Проблеми з управлінням запасами можуть призвести до накопичення невикористаних або непроданих товарів, що створює додаткові витрати [3]-[6].

Основна мета управління запасами полягає в тому, щоб зрозуміти рівень запасів і їх розміщення на складах. Програмне забезпечення для цього процесу контролює рух продукції від постачальника через виробництво до кінцевого споживача. На етапі зберігання на складі відслідковуються всі операції, пов'язані з прийомом товарів, їх комплектуванням, пакуванням та відправкою [1]-[8].

Існують різні підходи до управління запасами, деякі з них використовують математичні моделі та аналітику для оптимізації процесів, інші базуються на певних процедурах. Незалежно від методу, всі вони спрямовані на підвищення точності. Вибір конкретного підходу залежить від характеру запасів і потреб компанії [1]-[8].

Проаналізуємо підходи до розв'язання задачі управління запасами. ABC-аналіз допомагає визначити найпопулярніші та найменш затребувані види товарів. Пакетне відстеження дозволяє групувати подібні товари для контролю термінів придатності та управління дефектними одиницями. Метод масових поставок орієнтований на роботу з матеріалами, що доставляються у великих об'ємах, не розпаковуючи їх [1]-[8].

При консигнаційному управлінні бізнес не оплачує товар постачальнику, поки він не буде проданий, і право власності залишається у постачальника до моменту продажу. Інший метод дозволяє переміщувати товари безпосередньо від постачальника до кінцевого отримувача, мінімізуючи зберігання на складі [1]-[8].

Прогнозування попиту допомагає передбачити майбутню потребу в товарі. Дропшипінг передбачає доставку товарів постачальником безпосередньо клієнту, минаючи склад компанії. Формула економічного обсягу замовлення оптимізує обсяг замовлень для зменшення витрат на зберігання [1]-[8].

Інші методи з використанням спеціалізованого програмного забезпечення визначають послідовність продажу запасів: у першому випадку продаються товари, які надійшли раніше, а у другому — новіші, що може бути вигідним при зростанні цін. Спеціальна методика допомагає підтримувати мінімальний рівень запасів і поповнювати їх лише за необхідності [1]-[8].

Економічне виробництво спрямоване на зменшення відходів у процесі виробництва, а планування потреб у матеріалах за допомогою спеціалізованого програмного забезпечення допомагає координувати запаси та планувати виробничі процеси. Мінімальна кількість замовлення дозволяє компанії замовляти лише необхідний мінімум товарів для зниження виробничих витрат [1]-[8].

Формула точки пере упорядкування визначає мінімальний рівень запасів, при якому потрібно робити нове замовлення. Постійне управління запасами дає змогу відслідковувати їх використання в режимі реального часу.

Страховий запас забезпечує додатковий резерв товарів на випадок затримок із постачанням [1]-[8]. Деякі підходи допомагають виявляти і усувати відходи в процесі управління запасами, підвищуючи ефективність і точність [1]-[8].

Планування попиту є важливою складовою управління запасами, яка допомагає передбачити обсяг і час потреби в товарах. Після прогнозування попиту відбувається контроль руху товарів від постачальників через виробничі етапи до виконання замовлень клієнтів [1]-[8].

Володіння знаннями про формули управління запасами та використання відповідного програмного забезпечення є вирішальним для покращення рівня запасів [1]-[8].

Ефективне управління запасами за допомогою відповідного програмного забезпечення значно впливає на фінансові показники компанії. Існує низка ключових показників для оцінки успіху інвентаризації в різних організаціях. Важливо виявити, які показники найкраще демонструють ефективність вашого бізнесу [1]-[8].

Контроль запасів є важливим елементом управління запасами в компанії, де ланцюг поставок забезпечує переміщення товарів від виробника до споживача. Управління запасами спрямоване на моніторинг наявності товарів і контроль їх переміщення в конкретних місцях [1]-[8].

Контроль товарів є невід'ємною частиною управління запасами, що відповідає за ефективний рух товарів на складах. Оптимізація запасів — це процес підвищення ефективності використання товарів, що сприяє зменшенню витрат на їх утримання та зберігання [1]-[8].

Управління запасами охоплює організацію та моніторинг руху товарів від моменту їх надходження на склад. Процес обробки замовлень передбачає отримання та внесення замовлень від клієнтів. У багатьох випадках ці функції об'єднуються в одному програмному забезпеченні [1]-[8].

Управління запасами є ключовим аспектом забезпечення стійкості бізнесу. Під час отримання замовлень запаси можуть бути закріплені за конкретними замовленнями, після чого їхній статус змінюється на більш

високий рівень відповідно до цього замовлення. Завдяки інтеграції системи управління замовленнями з складською системою можна визначити, з якого складу потрібно виконати замовлення, а також об'єднати різні товари для оптимізації відправлень [1]-[8].

Управління ланцюгом постачань охоплює координацію відносин з постачальниками та контроль за потоком товарів як до компанії, так і в межах її операцій. Управління запасами зосереджується на управлінні основними активами компанії або її підрозділів [1]-[8].

Управління запасами відіграє ключову роль у контролі за ланцюгом поставок. Процес управління складом охоплює моніторинг руху товарів під час їхнього прийому, зберігання та відвантаження. Ланцюг постачання включає такі етапи, як прогнозування попиту, закупівлі, виробництво, контроль якості, виконання замовлень, облік запасів і обслуговування клієнтів, і всі ці процеси потребують чіткого контролю [1]-[8].

Управління запасами сприяє ефективному контролю за запасами. Система складу організовує зберігання товарів, а управління складом забезпечує моніторинг рівня запасів та аналіз тенденцій на різних складах або в масштабах всієї компанії [1]-[8].

Основою високої продуктивності складу є ретельно спланована та організована структура його функціонування. Розміщення кожного товару у визначеному місці на складі допомагає уникнути помилок персоналу та підвищує ефективність роботи. Однак всі ці процеси будуть ефективними лише за наявності належної складської документації [1]-[8].

Логістика полягає в управлінні потоками товарів як на складі, так і в рамках договірних угод і доставки. Управління запасами підтримує необхідний рівень товарів та контролює їхнє розташування у торгових точках. Управління запасами є важливим елементом у процесі логістики компаній. Логістика і зберігання взаємопов'язані, оскільки логістичні операції залежать від належного управління складом для ефективної роботи. Сучасні логістичні

системи включають як зберігання, так і обробку товарів для забезпечення плавного функціонування процесів [1]-[8].

Планування ресурсів підприємства (Enterprise Resource Planning, ERP) є програмною платформою, яка охоплює різні бізнес-функції, зокрема фінансовий облік, закупівлі, відповідність вимогам і керування ланцюгом поставок. Управління складом є частиною ERP-системи, яка забезпечує повний контроль над рівнем запасів, товарами в дорозі та їхнім поточним станом, надаючи в реальному часі загальну картину для всієї компанії [1]-[8].

Управління запасами сприяє оптимізації процесів замовлень. ERP-системи дозволяють компаніям отримувати точні дані про наявні запаси, що допомагає оперативно оновлювати їх. ERP ефективно керує інформацією, полегшуючи управління запасами й спрощуючи роботу з ними [1]-[8].

Управління запасами є ключовим елементом ланцюга поставок, хоча термінологія залишається незмінною. Ланцюг поставок контролює переміщення матеріалів від джерела сировини до кінцевого розподілу. Управління запасами охоплює процеси пошуку, організації, зберігання та моніторингу запасів для отримання необхідної інформації [1]-[8].

Для будь-якого бізнесу, що займається продажем товарів, оптимізація запасів є критичною умовою виживання. Якщо немає достатньо продуктів для продажу чи виконання замовлень, бізнес втрачає можливість отримати дохід. Часто спостерігається тенденція до замовлення надлишкових товарів, що може дозволити швидкий заробіток, але надмірні запаси призводять до втрат через застарілі чи непридатні для продажу товари [1]-[8].

Зменшення рівня запасів часто корелює зі збільшенням виручки та доходу. Якщо процес обробки товарів на складі займає значну частину часу, варто переглянути підходи до управління складом. Ефективне управління складом із товарними запасами за допомогою відповідного програмного забезпечення не лише мінімізує втрати, а й підвищує продуктивність [1]-[8].

Проаналізуємо етапи створення програмного забезпечення управління запасами, що включає процеси, засоби контролю та інструменти, необхідні для ефективного ведення бізнесу з управління запасами на складі [1]-[8].

На початковому етапі необхідно обрати способи доставки та зберігання продукції. Метод організації та зберігання товарів впливатиме на управління запасами. Якщо запаси зберігаються на власному сайті, всі процеси управління здійснюватимуться внутрішньо. У випадку використання віддалених центрів виконання або складів, а також за потреби в зовнішніх постачальниках, слід інтегрувати програмні засоби контролю запасів у їхні системи [1]-[8].

Далі відбувається визначення способу збору даних про запаси. Моніторинг даних про запаси є ключовим для ефективного управління, незалежно від того, зберігаються вони на власних складах, обслуговуються партнером чи здійснюється дропшипінг. Програмні інструменти системи управління запасами є вирішальними для цього. Дані про запаси можуть включати: номер товару та постачальника, кількість запасів на складі, розташування товару для зберігання чи демонстрації, інформацію про постачальника (мінімальний розмір замовлення, час виконання тощо), собівартість товару та оптові ціни, роздрібні ціни та ін. [1]-[8].

Для невеликої кількості товарів можна використовувати таблиці. Але для малого бізнесу більш економічно ефективними є інтегровані програмні системи управління запасами [1]-[8]. Ці системи спрощують введення замовлень, моніторинг запасів, інформацію про постачальників та інвентаризацію в одній платформі. Вони також інтегруються з терміналами для оплати замовлень, онлайн продажами та іншими каналами для оновлення даних про запаси в реальному часі [1]-[8].

Після цього виконується розробки системи кодування запасів, що є ключовою для швидкої ідентифікації та моніторингу товарів у довгострокових процесах. У позначеннях товарів зазвичай використовують комбінацію букв і

цифр, що дозволяє швидко отримувати основну інформацію про продукт, що зберігається на складі [1]-[8].

Далі виконується організація зон зберігання. Маючи спеціально визначені місця для кожного елемента, весь інвентар працює швидше і ефективніше. Перш ніж організувати запаси для бізнесу або магазину, важливо визначити методи зберігання та місця, такі як полиці, стелажі та ящики, де кожен товар буде розміщено у відповідному місці [1]-[8].

На наступному етапі, після впровадження програмної системи управління запасами виконується прогнозування для замовлення запасів. На цьому етапі визначається, скільки товарів потрібно для задоволення майбутніх потреб. Це враховує різні фактори, такі як темпи продажів, заплановані рекламні кампанії, ринкові тенденції, погодні умови та розвиток діяльності підприємства [1]-[8].

Метою прогнозування рівня запасів є забезпечення наявності достатньої кількості запасів для виконання очікуваних продажів протягом визначеного періоду. Розуміння швидкості обігу запасів є критично важливим для систем закупівель і управління запасами, які оптимально функціонують за підтримки інструментів прогнозування, інтегрованих у процес оформлення замовлень [1]-[8].

Також важливо враховувати терміни виконання замовлень клієнтів під час планування. Надійні постачальники, які швидше виконують замовлення, дозволяють зберігати менше запасів і розміщувати більше замовлень, що позитивно впливає на грошовий потік. Якщо постачання або закупівлі відбуваються повільно, це може призвести до накопичення коштів в інвентарі через зниження купівельної спроможності [1]-[8].

Швидкість доставки є важливим аспектом ефективного управління запасами. Товари, які не пройшли перевірку, неправильно зберігаються або відправлені на благодійність, не можуть бути продані чи відправлені. Тому важливо, щоб доступність запасів стала пріоритетом у стратегії управління ними за допомогою спеціалізованого програмного забезпечення [1]-[8].

Точний облік запасів за допомогою спеціалізованого програмного забезпечення також має велике значення, оскільки помилки можуть негативно вплинути на дані про якість продукту і призвести до надмірних замовлень, помилкових повернень та збуту неактуальних товарів. Усе це позначається на загальній ефективності бізнесу [1]-[8].

Оптимальним підходом є проведення інвентаризації у відповідності з замовленнями на купівлю, відкриваючи всі коробки та контейнери для перевірки правильності. Не варто покладатися лише на етикетки упаковок і коробок, оскільки помилки можуть виникати на будь-якому етапі [1]-[8].

Після отримання товарів їх слід одразу розмістити в призначених для цього місцях. Якщо ж є дорогі чи тимчасові предмети, їх можна позначити в інвентарі для зручності знаходження в майбутньому [1]-[8].

Використання методів «останній прийшов, перший вийшов» або «перший прийшов, перший вийшов» є корисним при розміщенні нових товарів на складі. Зазвичай рекомендується другий метод, який передбачає розміщення нових товарів позаду старих, що дозволяє першими продавати ті, які мають більший термін зберігання. Це особливо важливо для швидкопсувних товарів та продуктів з обмеженим терміном придатності, таких як фарба [1]-[8].

Багато компаній, які працюють на основі інвентаризації, щорічно проводять аудит своїх запасів з метою оподаткування. Це передбачає порівняння фактичної кількості товарів на складі з даними про кількість наявних товарів, що зберігаються у базі даних відповідного програмного забезпечення. Виявити та кількісно оцінити відмінності між річними звітами можна, оскільки помилки можуть накопичуватися протягом року [1]-[8].

Весь інвентар можна поділити на частини, відповідно до графіку чергування. Кількість циклів може варіюватися в залежності від постачальника, типу продукту, розташування складу або інших факторів, що впливають на операції. Регулярна інвентаризація декількох товарів допомагає

виявити випадкові помилки в обліку запасів, замовленнях, зберіганні або крадіжках [1]-[8].

Важливо за допомогою відповідного програмного забезпечення ретельно контролювати запаси, покращувати грошові потоки, виявляти крадіжки та інші втрати, а також максимізувати цінність [1]-[8].

На ринку доступні різноманітні програмні інструменти для управління запасами, деякі з яких безкоштовні. Більшість цих програмних систем пропонують всі необхідні функції для обліку запасів. Крім того, ці програмні засоби поєднують маркетингові канали та джерела виконання, що сприяє розвитку бізнесу [1]-[8].

Управління запасами охоплює організацію, зберігання, використання та реалізацію товарів компанії. Це включає обробку сировини, матеріалів та готової продукції, а також їх зберігання, використання та їх облік за допомогою відповідного програмного забезпечення [1]-[8].

Різні види промисловості та продажу покладаються на програмні засоби управління запасами для оптимізації виробничих процесів і зменшення обсягів запасів. Раніше цей процес здійснювався вручну, але тепер він виконується за допомогою складських програмних систем, інтегрованих з логістичним ланцюгом постачання, завдяки таким технологічним рішенням, як системи планування ресурсів підприємства і програмне забезпечення для управління складом. Ці інструменти надають дані в реальному часі, що дозволяє компаніям визначати, планувати і вдосконалювати управління запасами [1]-[8].

Переваги ефективного управління запасами за допомогою відповідних програмних засобів численні. Найголовніше, що воно забезпечує наявність товарів у потрібний час, що є критично важливим для виконання замовлень клієнтів і підтримки стабільних обсягів продажів, особливо під час запуску нових продуктів [1]-[8].

Ефективне управління запасами може поліпшити обіг товарів, свіжість продукції і зменшити витрати, що є особливо важливим у сферах з коротким життєвим циклом продуктів, таких як мода або високі технології [1]-[8].

Крім того, ефективне управління запасами сприяє зменшенню витрат, оскільки знижує потребу в надлишкових запасах, які можуть виснажувати ресурси і капітал, якщо їх не контролювати належним чином [1]-[8].

Проведення фізичної інвентаризації або підрахунку запасів є критично важливим, оскільки дозволяє перевірити кількість та якість наявних товарів. Цей основний процес забезпечує інформацію для фінансової звітності, прогнозування запасів і планування ланцюга постачання [1]-[8].

Ефективна програмна система обліку запасів є ключовою для підтримки цілісності даних, що стосується всіх аспектів управління, включаючи задоволення потреб клієнтів. Цей процес особливо важливий наприкінці звітних періодів, оскільки дозволяє відобразити реальну вартість активів компанії в звітності [1]-[8].

Розглянемо технології та підходи, які використовуються для ефективної обробки товарів та матеріалів на складі, та реалізуються у відповідному програмному забезпеченні для обліку та управління запасами [1]-[8].

Підхід до управління запасами точно вчасно (Just in Time Management, JIT) передбачає доставку запасів лише в момент їх необхідності під час виробничого процесу, що допомагає зменшити витрати на запаси. Особливо в автомобільній промисловості JIT сприяє зниженню рівня запасів і відповідних витрат, що призводить до більш ефективних ланцюгів постачання [1]-[8].

Програмні засоби планування матеріальних потреб (Materials Requirement Planning MRP) розраховують необхідну кількість матеріалів і витратних матеріалів для виробництва продуктів. Цей підхід є особливо корисним у виробництві, оскільки забезпечує доступність сировини без прив'язування надмірного капіталу [1]-[8].

Формула економічної кількості замовлення використовується для визначення оптимальної кількості замовлення, що мінімізує витрати на запаси, включаючи витрати на зберігання. Цей підрахунок є важливим для підтримання балансу в управлінні запасами в різних галузях [1]-[8].

Інвентаризація денних продажів демонструє, наскільки швидко компанія конвертує свої запаси у продажі. Нижчий рівень цього показнику свідчить про успішність продажу товарів, що є важливим критерієм для оцінки якості запасів компанії [1]-[8].

Незважаючи на свою значущість, управління запасами стикається з викликами, які можуть перевантажити ланцюг поставок і вплинути на ефективність операцій [1]-[8].

Однією з поширених проблем є зайве накопичення запасів, яке може створювати фінансовий тягар і призводити до застарілих товарів. З іншого боку, недостатні запаси можуть спричинити їх дефіцит через затримки у виробництві та незадоволеність клієнтів. Тому для усунення зазначених недоліків доцільно використовувати спеціалізоване програмне забезпечення для управління запасами [1]-[8].

Відзначено, що управління запасами допомагає компаніям приймати обґрунтовані рішення про те, що і в яких кількостях варто зберігати. Цей процес забезпечує своєчасну ідентифікацію наявних запасів та відповідну реакцію на зміни, щоб завжди мати достатню кількість товарів для виконання замовлень клієнтів і вчасно попереджати про можливі нестачі. Існують різні підходи до управління запасами, деякі з них використовують математичні моделі та аналітику для оптимізації процесів, інші базуються на певних процедурах. Проаналізовано підходи до розв'язання задачі управління запасами. Проаналізовано етапи створення програмного забезпечення управління запасами. Відзначено, що важливим є прогнозування рівня запасів з метою забезпечення наявності достатньої кількості запасів для виконання очікуваних продажів протягом визначеного періоду. Показано, що однією з поширених проблем управління складом є зайве накопичення запасів, яке

може створювати фінансовий тягар і призводити до застарілих товарів, з іншого боку, недостатні запаси можуть спричинити їх дефіцит через затримки у виробництві та незадоволеність клієнтів. Тому для усунення зазначених недоліків доцільно використовувати спеціалізоване програмне забезпечення для управління запасами.

1.2 Аналіз програмного забезпечення для управління запасами

Проаналізуємо програмне забезпечення для управління запасами [9]-[15].

Програмне забезпечення Ordoro (рис. 1.1) автоматично відстежує запаси на різних платформах продажу та надсилає сповіщення, коли їх рівень знижується. Крім того, система пропонує знижені тарифи на доставку завдяки партнерствам з основними перевізниками [9]-[13].

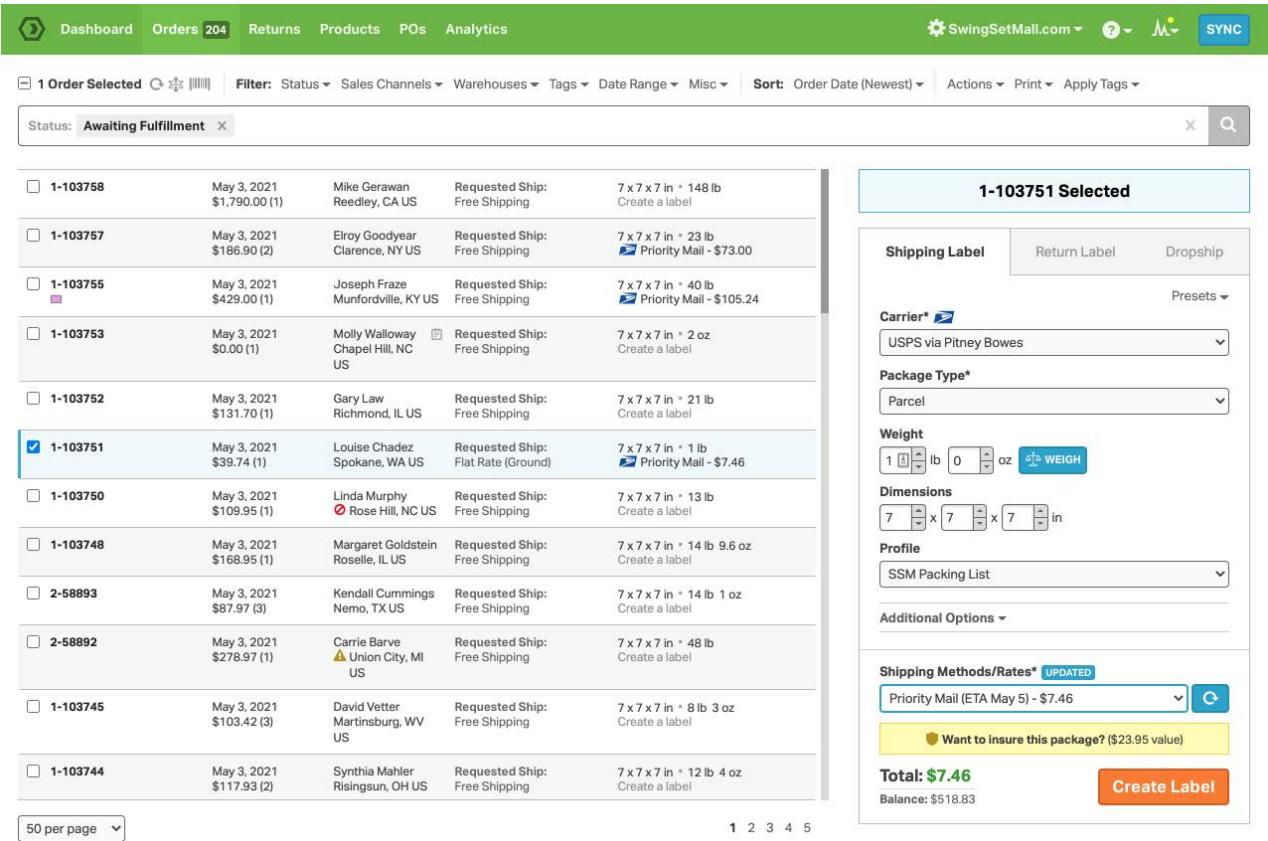


Рисунок 1.1 – Програмне забезпечення Ordoro [13]

Управління запасами в Ordoro є надійним й гнучким. Програмне забезпечення Ordoro дозволяє створювати продукти з індивідуальними описами та зображеннями для різних каналів продажу, а також об'єднувати кілька артикулів в одну одиницю для продажу. За допомогою штрих-кодів можна легко відслідковувати та коригувати рівень запасів. Ordoro також підтримує функції, орієнтовані на електронну комерцію, що дозволяє компаніям автоматично передавати замовлення клієнтів постачальникам [9]-[13].

Варто відзначити, що Ordoro має високу масштабованість та інтегрується з такими платформами, як Amazon, eBay, Shopify і BigCommerce, а також з бухгалтерськими програмами на кшталт QuickBooks [9]-[13].

Це рішення ідеально підходить для компаній, орієнтованих на електронну комерцію та багатоканальні продажі, забезпечуючи ефективне управління запасами, координацію ланцюга постачання та безперебійну обробку замовлень. Ordoro синхронізує дані про запаси та статуси доставки в реальному часі на різних платформах, дозволяючи ефективно керувати роздрібною торгівлею [9]-[13].

Унікальною особливістю Ordoro є автоматизація процесу закупівель: система автоматично направляє замовлення на склади та постачальникам, а також створює замовлення на купівлю в один клік на основі прогнозування продажів або встановлених порогів повторного замовлення [9]-[13].

Програмне забезпечення Ordoro значно спрощує багатоканальну торгівлю. Завдяки цьому програмному забезпеченню можна створювати основні продукти у своїй системі інвентаризації та редагувати описи, зображення для всіх або вибраних каналів продажу [9]-[13].

Ordoro також пропонує зручні інструменти для керування постачальниками, що робить процес електронної комерції надзвичайно простим. Наприклад, можна призначати постачальників для окремих товарів у каталозі. Кожен раз, коли клієнт замовляє такі товари, Ordoro автоматично надсилає замовлення безпосередньо постачальнику, мінімізуючи участь

посередника, оптимізуючи ланцюг постачання та забезпечуючи швидку доставку товарів клієнтам [9]-[13].

Ordoro також підтримує нестандартні стратегії управління запасами. Наприклад, ви можете перепродавати свої товари стратегічно або обмежувати кількість доступних товарів для клієнтів, створюючи ілюзію дефіциту, що стимулює покупців швидше завершити покупку. Це допомагає збільшити прибуток і уникнути накопичення зайвих дорогих запасів [9]-[13].

Основними характеристиками програмного забезпечення Ordoro є такі [9]-[13]:

- засоби управління багатоканальним продажем: програмне забезпечення Ordoro дозволяє синхронізувати та керувати запасами через різні канали продажу. Це полегшує процес оновлення описів та зображень товарів у всіх каналах одночасно [9]-[13];

- автоматизація управління запасами: програмне забезпечення Ordoro автоматично відстежує рівень запасів у реальному часі та надсилає сповіщення, коли запаси зменшуються, запобігаючи дефіциту товарів. Вона також підтримує використання штрих-кодів для спрощення інвентаризації та коригування рівнів запасів [9]-[13];

- інтеграція з логістичними партнерами: програмне забезпечення Ordoro має угоди з провідними перевізниками, що дозволяє користувачам отримувати знижені тарифи на доставку [9]-[13];

- підтримка електронної комерції: програмне забезпечення Ordoro автоматично надсилає замовлення постачальникам, які займаються електронною комерцією, мінімізуючи участь продавця. Це дозволяє продавцям швидко виконувати замовлення, покращуючи ефективність ланцюга постачання [9]-[13];

- масштабованість: програмне забезпечення Ordoro підходить як для невеликих підприємств, так і для великих компаній, що працюють на ринку електронної комерції. Програмне забезпечення легко інтегрується з

бухгалтерськими програмами, як-от QuickBooks, для ефективного фінансового управління [9]-[13];

- автоматизація закупівель: програмне забезпечення Ordoro надає можливість автоматично генерувати замовлення на купівлю на основі прогнозів продажів або низьких рівнів запасів, що оптимізує процес поповнення товарів і зменшує ризик дефіциту [9]-[13];

- аналітика та звітування: програмне забезпечення Ordoro надає детальні аналітичні дані щодо запасів, продажів і ефективності виконання замовлень, допомагаючи бізнесу краще планувати запаси і поліпшувати фінансову звітність [9]-[13];

- гнучке управління запасами: програмне забезпечення Ordoro підтримує методи управління запасами, таких як "останній прийшов, перший вийшов" та "перший прийшов, перший вийшов", що дозволяє оптимально управляти продукцією з коротким терміном придатності [9]-[13].

Серед недоліків програмного забезпечення Ordoro можна відзначити високу вартість для малого бізнесу та обмеженість безкоштовної версії, що робить обмеженим його використання для невеликих підприємств. Крім того, не зважаючи на те, що програмне забезпечення Ordoro має багато корисних функцій, його інтерфейс може бути дещо складним для нових користувачів, тому вивчення всіх можливостей та функціоналу може зайняти час. Хоча Ordoro інтегрується з багатьма великими платформами, кількість підтримуваних інтеграцій все ще обмежена [9]-[13].

Програмне забезпечення inFlow Inventory (рис. 1.2) надає потужні функції для відстеження запасів, закупівель і виконання замовлень, що робить його відповідним рішенням як для роздрібних продавців, так і виробників. Він підтримує детальний контроль за сировиною, готовими виробами, штрих-кодами, управління кількома складами, електронною комерцією і автоматизацію закупівель [9]-[12], [14].

inFlow Inventory - Premium Edition

Search: Order #, Inventory Status, Payment Status, Customer, Show Cancelled, Refresh

Customer: Serious Deals For You, Sales Rep, Contact: Debra Ignatieff, Phone: 310-555-4932, Address: P.O. Box 2103, Mountain View, CA, USA, 94043

Order # SO-000004, Order Date 9/30/2016, Status Unfulfilled, Uninvoice

Item	Description	Quantity	Unit Price	Discount	Tax	Sub-Total
00303A	Army Combat Helmet	10 set	\$8.99	0 %	Taxable	\$89.90
PM-BW01	24" x 750' Sheet of 3/16" Bubble Wrap	5	\$5.00	0 %	Taxable	\$25.00
PM-SB03	16" X 14" X 10" Shipping Boxes	1	\$0.50	0 %	Taxable	\$0.50
Catalog06	2006 Product Catalog	1	\$0.00	0 %	Taxable	\$0.00

Non-Customer Costs: These are extra costs associated with the sale but not charged to the customer, e.g. payment processing fees.

Sub-Total: \$115.40, State Tax: \$9.52, Total: \$124.92, Paid: \$0.00, Balance: \$124.92

Buttons: Fulfill, Mark Paid, SALES, Fulfill, Return, Restock

Finished loading Sales Order - Order #SO-000004, Current User: Default User

Рисунок 1.2 – Програмне забезпечення inFlow Inventory [14]

Для компаній з обмеженими фінансовими можливостями програмне забезпечення InFlow Inventory є ефективним рішенням для управління запасами. Це програмне забезпечення пропонує безкоштовну версію з обмеженими функціями для управління запасами. Крім того, користувачі можуть скористатися розширеними функціями за допомогою платних планів inFlow [9]-[12], [14].

InFlow відзначається простотою у використанні, оскільки більшість його функцій для відстеження запасів автоматизовані. Програмне забезпечення також пропонує зручний мобільний додаток, що дозволяє легко сканувати штрих-коди і дає можливість торговим агентам та менеджерам з інвентаризації оперативно переглядати рівень запасів [9]-[12], [14].

Проте варто зазначити, що inFlow Inventory є доступним за ціною, оскільки його можливості масштабування обмежені в порівнянні з іншими

платформами. Наприклад, хоча система значно спрощує процеси управління запасами, вона обмежує кількість замовлень, які можна обробляти на місяць. InFlow Inventory пропонує зручні засоби налаштування, що дає можливість компаніям налаштовувати систему управління запасами відповідно до своїх індивідуальних потреб. Додатково, завдяки мобільному додатку, користувачі можуть зручно керувати запасами та виконувати операції [9]-[12], [14].

Головна відмінність inFlow полягає в широкому наборі функцій для оптової торгівлі. Це включає віртуальні виставкові зали, котирування та рахунки, маршрутизацію замовлень для торгових агентів, а також спеціалізовані робочі процеси для виробництва, що робить його ідеальним вибором для компаній у цих сегментах [9]-[12], [14].

Програмне забезпечення inFlow Inventory має такі особливості [9]-[12], [14]:

- зручні засоби налаштування: користувачі програмного забезпечення inFlow можуть адаптувати систему управління запасами під свої унікальні потреби завдяки гнучким полям, що можуть налаштовуватися [9]-[12], [14];

- мобільний застосунок: програмне забезпечення inFlow доступне для мобільних платформ iOS та Android, що дозволяє управляти запасами на ходу, зокрема здійснювати сканування штрих-кодів і відстеження рівнів запасів у реальному часі [9]-[12], [14];

- масштабованість: програмне забезпечення inFlow підтримує обробку до десяти тисяч замовлень на продаж щомісяця, що робить її відповідною для організацій середніх масштабів [9]-[12], [14];

- інтеграція зі штрих-кодами: inFlow підтримує розширені можливості управління запасами через використання штрих-кодів, що спрощує інвентаризацію та облік товарів [9]-[12], [14];

- управління декількома складами: програма inFlow дозволяє управляти запасами на різних складах, що робить її зручною для компаній з кількома місцями розташування [9]-[12], [14];

– автоматизація закупівель: підтримка автоматизованих процесів закупівель забезпечує зручне управління поповненням запасів [9]-[12], [14];

– функції для оптової торгівлі: програмне забезпечення inFlow пропонує віртуальні виставкові зали, котирування і оптове виставлення рахунків, що робить цю програму зручною для організацій, орієнтованих на оптовий ринок [9]-[12], [14];

– електронна комерція: програмне забезпечення підтримує автоматизоване відправлення замовлень постачальникам товарів електронної комерції, що спрощує процес виконання замовлень [9]-[12], [14];

– відстеження компонентів та готової продукції: inFlow дозволяє відстежувати сировину, компоненти та готові вироби, що підходить для виробничих компаній [9]-[12], [14];

– можливість використання в хмарі: після припинення підтримки локальної версії inFlow, програма буде доступна виключно у хмарній версії, що забезпечує актуальність оновлень та технічну підтримку [9]-[12], [14].

Серед недоліків програмного забезпечення inFlow Inventory можна відзначити можна відзначити обмежену масштабованість, оскільки inFlow підтримує до десяти тисяч замовлень на місяць, це може бути недостатньо для великих компаній з великою кількістю транзакцій. Для більш масштабних компаній можуть знадобитися системи, що підтримують більші обсяги. Крім того, не зважаючи на те, що програмне забезпечення inFlow Inventory має багато корисних функцій, його інтерфейс може бути дещо складним для нових користувачів, тому вивчення всіх можливостей та функціоналу може зайняти час. Хоча inFlow Inventory інтегрується з багатьма великими платформами, кількість підтримуваних інтеграцій все ще обмежена [9]-[12], [14].

Програмне забезпечення Cin7 (рис. 1.3) спрощує управління запасами, виробництвом, інтеграцією продажів, звітністю та автоматизацією на корпоративному рівні для компаній будь-якого масштабу. Завдяки багатомодульній системі, програмне забезпечення Cin7 об'єднує всі процеси управління запасами в одному місці [9]-[12], [15].

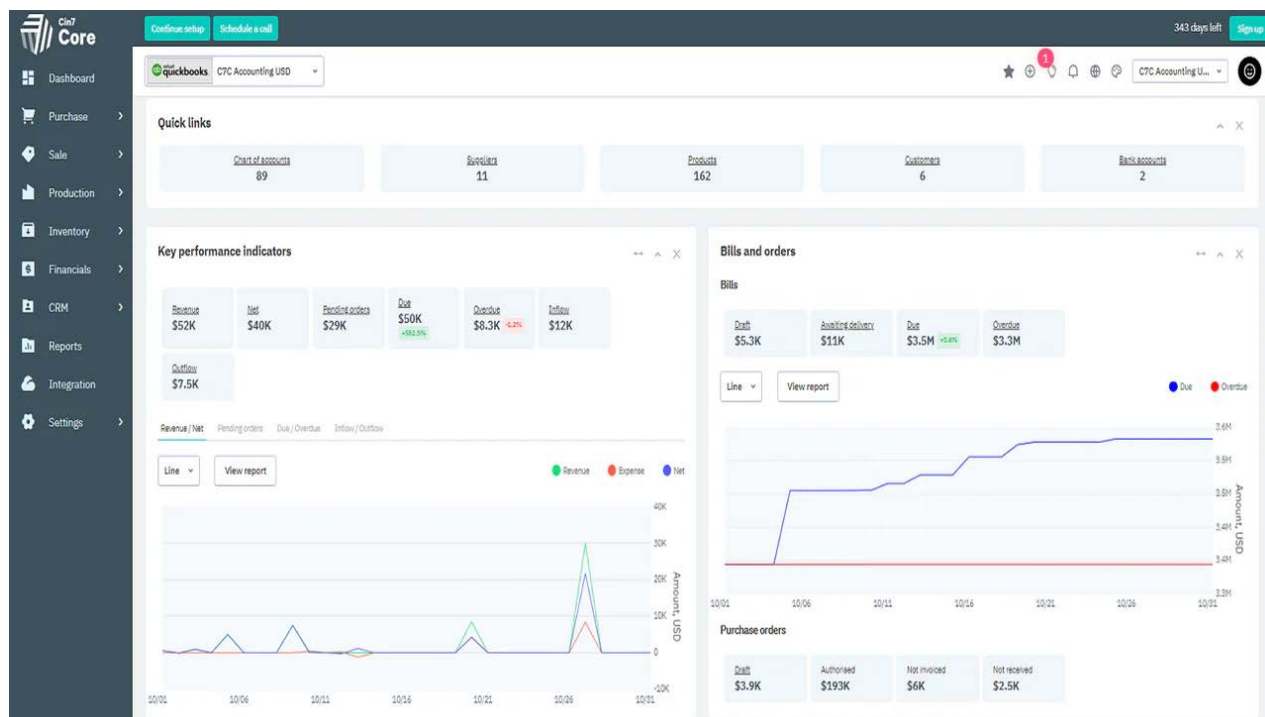


Рисунок 1.3 – Програмне забезпечення Cin7 [15]

Програмне забезпечення Cin7 забезпечує повну підтримку продажів по всіх каналах і підтримує інтеграцію з онлайн магазинами та платформами доставки, що дозволяє розширювати охоплення клієнтів та збільшувати продажі. Це універсальне рішення є економічно вигідною альтернативою складним програмним системам [9]-[12], [15].

Cin7 Core підтримує бухгалтерське програмне забезпечення, як-от QuickBooks, і інтегрується з основними платформами електронної комерції та службами доставки, такими як Square, Amazon, Fulfillment by Amazon, Shopify, WooCommerce, Shipstation, Vend, Floship тощо [9]-[12], [15].

Для малого та середнього бізнесу Cin7 Core забезпечує повну синхронізацію управління запасами та бухгалтерського обліку. Система дозволяє відстежувати рівні запасів і витрати на сировину, компоненти та готову продукцію, а також контролювати замовлення з різних каналів, зокрема онлайн продажів. Завдяки потужним функціям виробництва та відстеження компонентів, Cin7 Core автоматично узгоджує рух запасів з бухгалтерськими даними, мінімізуючи розбіжності [9]-[12], [15].

Програмне забезпечення Cin7 створено для компаній, що прагнуть ефективно працювати в багатоканальних продажах, забезпечуючи інструменти для управління складами та логістикою. Система поєднує інтеграцію, автоматизацію та відстеження запасів для комплексного управління замовленнями, інвентаризацією та доставкою, забезпечуючи безперебійний робочий процес [9]-[12], [15].

Програмне забезпечення Cin7 може використовуватися для задоволення потреб оптовиків, роздрібних торговців та компаній електронної комерції, інтегрується з великою кількістю платформ, включаючи популярні сайти електронної комерції, бухгалтерські програми, тощо. Це забезпечує плавну синхронізацію запасів на всіх каналах продажів, що допомагає уникнути перепродажу або нестачі товарів [9]-[12], [15].

Можливості відстеження запасів у реальному часі та детальні звіти, які пропонує Cin7 Core, надають повний контроль над запасами, продажами та операціями ланцюга постачання. Програмне забезпечення також підтримує відстеження партій і серійних номерів, що робить його ідеальним вибором для компаній зі складними вимогами до управління запасами. Хоча початкове налаштування може бути трохи складним, Cin7 пропонує пакети адаптації, які можуть спростити процес, але вони доступні за додаткову плату [9]-[12], [15].

Cin7 відрізняється тим, що надає вбудовані модулі, які входять у стандартний план. Окрім повністю інтегрованої системи для оплати замовлень, програма включає функції електронної комерції, модулі для виробництва та навіть платіжний портал. Все це доповнюється стандартними функціями, такими як сканування штрих-кодів, відстеження запасів і виконання замовлень, які користувачі очікують від сучасних рішень для інвентаризації [9]-[12], [15].

Cin7 також забезпечує просунуте прогнозування попиту і підтримку багатоканальних продажів з оновленням даних кожні п'ять хвилин, що гарантує актуальність інформації та ефективне управління запасами з мінімальними зусиллями [9]-[12], [15].

Програмне забезпечення Cin7 має такі особливості [9]-[12], [15]:

- інтеграція з платформами: програмне забезпечення Cin7 Core інтегрується з великою кількістю платформ, включаючи популярні сайти електронної комерції та бухгалтерське програмне забезпечення, що забезпечує безперебійну синхронізацію даних [9]-[12], [15];

- відстеження запасів у реальному часі: програмне забезпечення пропонує можливості моніторингу запасів в реальному часі, що дозволяє користувачам мати актуальну інформацію про наявність товарів [9]-[12], [15];

- докладні звіти: програмне забезпечення Cin7 генерує детальні звіти про запаси, продажі та операції в ланцюзі постачання, що допомагає підприємствам приймати обґрунтовані рішення [9]-[12], [15];

- підтримка відстеження партій і серійних номерів: програмне забезпечення Cin7 дозволяє відстежувати товари за партіями та серійними номерами, що є корисним для компаній зі складними потребами в управлінні запасами [9]-[12], [15];

- модулі для електронної комерції: програмне забезпечення Cin7 включає вбудовані функції електронної комерції B2B, які підтримують різноманітні платформи та можливості продажу [9]-[12], [15];

- виробничі можливості: програмне забезпечення Cin7 має функції для управління виробництвом, включаючи відстеження компонентів і автоматизовані процеси [9]-[12], [15];

- система оплати товарів: програмне забезпечення Cin7 постачає повністю інтегровану систему точок продажу, що дозволяє обробляти транзакції безпосередньо в програмі [9]-[12], [15];

- прогнозування попиту: програмне забезпечення Cin7 надає розширене прогнозування попиту, що допомагає підприємствам планувати свої запаси відповідно до ринкових тенденцій [9]-[12], [15];

- багатоканальна підтримка: програмне забезпечення Cin7 підтримує синхронізацію між різними каналами продажів, що дозволяє користувачам ефективно управляти продажами через різні платформи [9]-[12], [15];

– автоматизація процесів: програмне забезпечення Cin7 автоматизує багато рутинних завдань, що зменшує навантаження на працівників і підвищує загальну ефективність роботи [9]-[12], [15].

Серед недоліків програмного забезпечення Cin7 можна відзначити можна відзначити обмежену масштабованість. Для більш масштабних компаній можуть знадобитися системи, що підтримують більші обсяги. Крім того, не зважаючи на те, що програмне забезпечення Cin7 має багато корисних функцій, його інтерфейс може бути дещо складним для нових користувачів, тому вивчення всіх можливостей та функціоналу може зайняти час. Хоча Cin7 інтегрується з багатьма великими платформами, кількість підтримуваних інтеграцій все ще обмежена [9]-[12], [15].

Порівняльну характеристику програмного забезпечення для управління запасами наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для управління запасами

Критерій порівняння	Ordoro [13]	inFlow Inventory [14]	Cin7 [15]
Зручність налаштування	–	+	+–
Автоматизація закупівель	+–	+	+
Засоби управління багатоканальним продажем	+	+	+–
Аналітика та звітування	+–	+–	+
Масштабованість	+–	+–	–
Інтеграція з програмним забезпеченням провідних перевізників	+	–	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для управління запасами. Проте деякі програмні засоби для управління запасами можуть бути досить дорогими, особливо для малого та середнього бізнесу. Деякі програмні засоби для управління запасами потребують складного налаштування та адаптації під конкретні бізнес-процеси. Інші програмні системи для управління запасами можуть не бути достатньо гнучкими для масштабування, що у деяких випадках може вимагати оновлення або повної заміни програмного забезпечення, що супроводжується додатковими витратами та зусиллями працівників підприємства. Крім того, деякі програми для з управління запасами можуть не враховувати унікальні вимоги або специфіку конкретного бізнесу.

Тому актуальною є розробка програмного забезпечення для управління запасами.

При розробці програмного забезпечення для управління запасами необхідно забезпечити такі функціональні вимоги:

- підтримка можливості управління запасами різними методами;
- підтримка можливості завдання значень параметрів (можливість дефіциту, розподіл попиту та ін.) підходу до управління запасами;
- можливість табличного подання розрахункових значень запасів товарів на складі;
- можливість динамічного керування запасами (визначення рівня запасів товарів для різних періодів);
- подання інформації про програмне забезпечення для управління запасами;
- підтримка можливості роботи з файлами даних.

1.3 Висновки за розділом 1

Відзначено, що управління запасами допомагає компаніям приймати обґрунтовані рішення про те, що і в яких кількостях варто зберігати. Цей процес забезпечує своєчасну ідентифікацію наявних запасів та відповідну реакцію на зміни, щоб завжди мати достатню кількість товарів для виконання замовлень клієнтів і вчасно попереджати про можливі нестачі. Існують різні підходи до управління запасами, деякі з них використовують математичні моделі та аналітику для оптимізації процесів, інші базуються на певних процедурах. Проаналізовано підходи до розв'язання задачі управління запасами. Проаналізовано етапи створення програмного забезпечення управління запасами. Відзначено, що важливим є прогнозування рівня запасів з метою забезпечення наявності достатньої кількості запасів для виконання очікуваних продажів протягом визначеного періоду. Показано, що однією з поширених проблем управління складом є зайве накопичення запасів, яке може створювати фінансовий тягар і призводити до застарілих товарів, з іншого боку, недостатні запаси можуть спричинити їх дефіцит через затримки у виробництві та незадоволеність клієнтів. Тому для усунення зазначених недоліків доцільно використовувати спеціалізоване програмне забезпечення для управління запасами.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для управління запасами. Відзначено, що деякі програмні засоби для управління запасами можуть бути досить дорогими, особливо для малого та середнього бізнесу. Деякі програмні засоби для управління запасами потребують складного налаштування та адаптації під конкретні бізнес-процеси. Інші програмні системи для управління запасами можуть не бути достатньо гнучкими для масштабування, що у деяких випадках може вимагати оновлення або повної заміни програмного забезпечення, що супроводжується додатковими витратами та зусиллями працівників підприємства. Крім того, деякі програми для з управління

запасами можуть не враховувати унікальні вимоги або специфіку конкретного бізнесу. Тому актуальною є розробка програмного забезпечення для управління запасами.

Сформульовано функціональні вимоги до програмного забезпечення для управління запасами.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для управління запасами було використано мову програмування C# [16], [17].

C# – це об'єктно-орієнтована мова програмування, створена для платформи .NET. Її синтаксис схожий на C++ і Java. C# підтримує статичні змінні, поліморфізм, перевантаження методів, функції класів, властивості, події, об'єкти, обробку винятків та парсинг XML. На відміну від C++, Delphi, Module і Smalltalk, C# уникає деяких складних прикладів програмування, таких як використання множинної спадковості, що створювало проблеми в цих мовах [16], [17].

C# можна вважати компактнішою мовою порівняно з Java. Java, розроблена Sun Microsystems для поліпшення можливостей Інтернету, ґрунтувалася на C++, але спростила деякі небезпечні аспекти, такі як безкінечні цикли [16], [17].

У контексті розподілених обчислень виникла концепція віртуальної машини, яка виступає посередником між програмним кодом і апаратними інструкціями комп'ютера, що дозволяє коду бути незалежним від конкретної платформи [16], [17].

Однією з переваг C# є можливість легкого інтегрування з кодом, написаним іншими мовами, що корисно для масштабних проєктів. Коли інші мови використовуються в межах платформи .NET, відбувається спільне використання системи типів даних. На сьогоднішній день C# є основною мовою програмування Microsoft, яка максимально використовує нові можливості платформи .NET. Хоча підтримуються й інші мови, .NET-додатки можуть мати певні уразливості [16], [17].

Синтаксис C# є відносно простим і зрозумілим, що полегшує розробку, супровід і підтримку програмного забезпечення для управління запасами. C# має велику кількість бібліотек і фреймворків для вирішення різних завдань,

що прискорює процес розробки та знижує вартість проєкту програмного забезпечення для управління запасами [16], [17].

Завдяки .NET Core, C# можна використовувати для розробки кросплатформних програм, які працюватимуть на Windows, Linux або macOS. Це особливо важливо для підприємств, що використовують різні операційні системи [16], [17].

Обґрунтування вибору мови програмування для створення програмного забезпечення для управління запасами наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для створення програмного забезпечення для управління запасами

Критерій порівняння мов програмування	Мова програмування		
	C#	C++	Ruby
Зручність роботи з інтерфейсом користувача	+	–	+–
Простота у використанні	+	+–	+
Висока продуктивність	+	+	–
Зручність для створення ПЗ для управління запасами	+	–	–
Масштабованість	+	+–	+

Отже, для реалізації програмного забезпечення для управління запасами обрано мову програмування C#, яка має високу продуктивність завдяки ефективній роботі з пам'яттю, а також дозволяє створювати зручні та інтерактивні користувацькі інтерфейси, що підвищує зручність роботи для співробітників, які керують запасами. Крім того, програми, написані на мові C#, можуть легко масштабуватися, щоб обробляти збільшені обсяги даних без втрати продуктивності, що є важливим для великих підприємств з великою кількістю товарів та постачальників.

2.2 Вибір середовища розробки для створення програмного забезпечення для управління запасами

В якості середовища розробки для створення програмного забезпечення для управління запасами було обрано середовище Visual Studio [18], [19].

Середовище розробки Visual Studio – це набір продуктів від Microsoft, що включає інтегроване середовище розробки (IDE) та широкий спектр інструментів [18], [19]. Він надає можливості для створення настільних та графічних додатків, таких як Windows Forms, а також вебсайтів, вебзастосунків і сервісів з вебпідтримкою. Visual Studio підтримує як вихідний, так і керований код для різних платформ. Серед підтримуваних платформ — Microsoft Windows, Windows Mobile, Windows Phone, а також Windows CE. Інструменти працюють із такими технологіями, як .NET Framework, .NET Compact Framework, Microsoft Silverlight і UWP [18], [19].

Visual Studio є основним інструментом для розробки на мові C# і платформі .NET, що дозволяє максимально ефективно використовувати можливості цієї мови. Всі необхідні інструменти для роботи з C#, включаючи інтегровані фреймворки та бібліотеки, доступні прямо з Visual Studio. Завдяки вбудованим засобам автозавершення коду (IntelliSense), налагоджувальним інструментам та багатьом іншим допоміжним функціям, розробка програм на C# стає швидшою та менш схильною до помилок [18], [19].

Visual Studio надає ефективні інструменти налагодження, включаючи підтримку покрокового виконання коду, перевірки значень змінних у реальному часі та моніторинг ресурсів. Це дозволяє швидко виявляти і виправляти помилки в коді [18], [19].

Visual Studio надає можливість розробляти програми для Windows і мобільних пристроїв, а також веб-додатки за допомогою ASP.NET Core. Останні версії Visual Studio включають підтримку .NET Framework із

додатковими покращеннями для кросплатформених рішень, особливо для вебзастосунків і бізнес-аналітики [18], [19].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для управління запасами наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для управління запасами

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	Atom	Xcode
Глибока інтеграція з C# та .NET	+	–	–
Підтримка кросплатформних рішень	+	+	+–
Засоби для налагодження і тестування	+	+–	+
Інтегровані фреймворки та бібліотеки для мови C#	+	–	–
Зручність інструментів для створення інтерфейсу користувача	+	+	+–

Таким чином, для створення програмного забезпечення для управління запасами обрано середовище розробки Visual Studio, яке є основним інструментом для розробки програм на мові C# і платформі .NET, що дозволяє максимально ефективно використовувати можливості цієї мови. Всі необхідні інструменти для роботи з C#, включаючи інтегровані фреймворки та бібліотеки, доступні безпосередньо з середовища Visual Studio. Крім того, Visual Studio надає ефективні інструменти налагодження, включаючи

підтримку покрокового виконання коду, перевірки значень змінних у реальному часі та моніторинг ресурсів.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для управління запасами

З метою можливості розробки програмного забезпечення для управління запасами створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для управління запасами враховуємо те, що користувачі програми мають можливості управління запасами різними методами, завдання значень параметрів (можливість дефіциту, розподіл попиту та ін.) підходу до управління запасами, табличного подання розрахункових значень запасів товарів на складі, динамічного керування запасами (визначення рівня запасів товарів для різних періодів), отримання інформації про програмне забезпечення для управління запасами, роботи з файлами даних.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для управління запасами можна описати таким чином:

- користувач програмного забезпечення для управління запасами запускає програму;
- після запуску програми користувачу відображається головна форма програми;
- користувач ознайомлюється з довідкою про програму;
- користувач вибирає метод управління запасами;
- користувач вводить значення параметрів підходу до управління запасами та натискає відповідну кнопку для вирішення задачі;
- програма вирішує задачу управління запасами та надає користувачу відповідь у вигляді табличного подання розрахункових значень запасів товарів на складі;

– користувач завершує роботу з програмним забезпеченням для управління запасами.

Враховуючи сценарії роботи користувачів при роботі з програмою, модель взаємодії користувача з програмним забезпеченням для управління запасами подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.

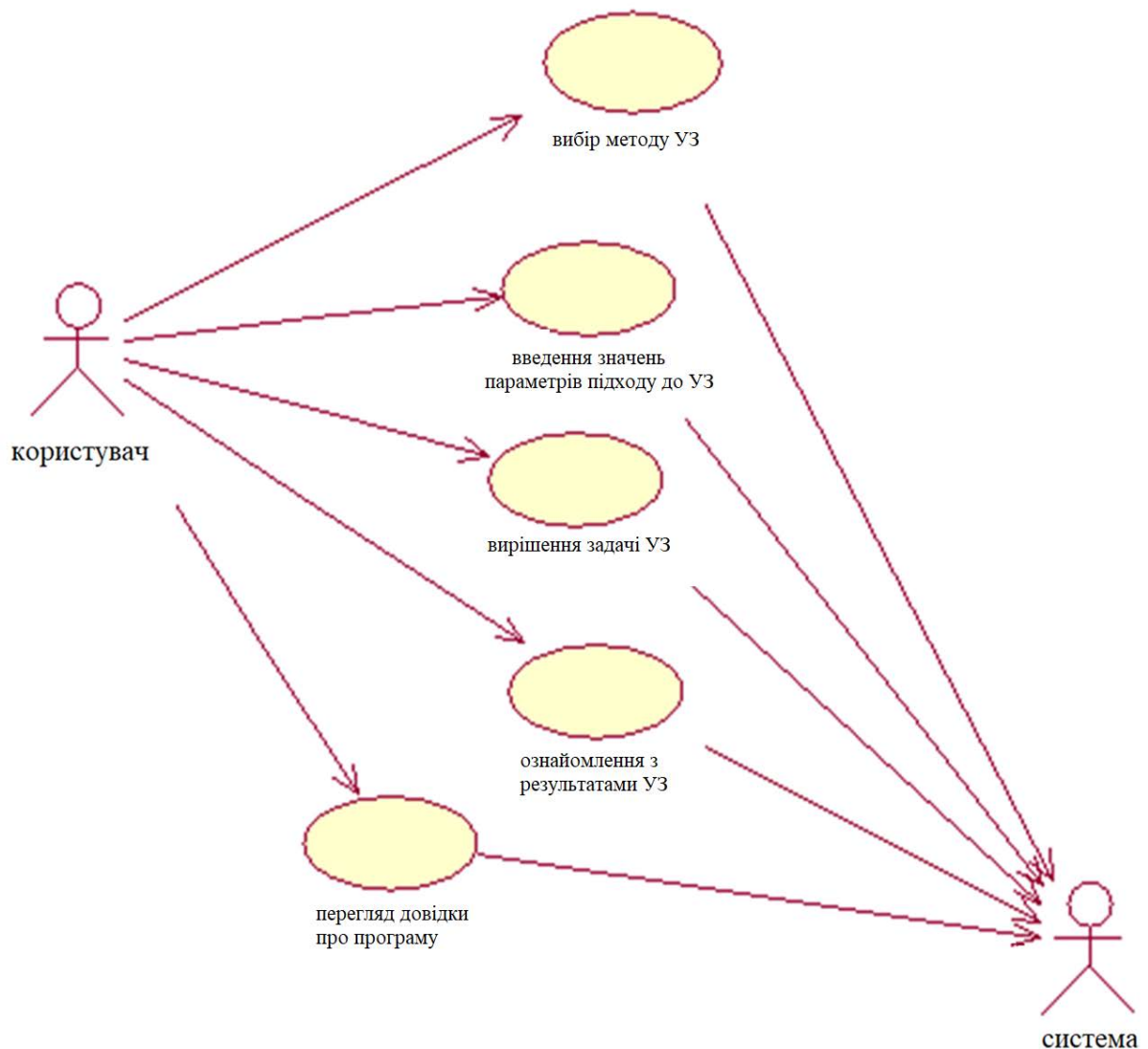


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для управління запасами

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для управління запасами, яка подана у вигляді діаграми

прецедентів, та дозволяє розроблювати програмне забезпечення для управління запасами.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для управління запасами обрано мову програмування C#, яка має високу продуктивність завдяки ефективній роботі з пам'яттю, а також дозволяє створювати зручні та інтерактивні користувацькі інтерфейси, що підвищує зручність роботи для співробітників, які керують запасами. Крім того, програми, написані на мові C#, можуть легко масштабуватися, щоб обробляти збільшені обсяги даних без втрати продуктивності, що є важливим для великих підприємств з великою кількістю товарів та постачальників.

Для створення програмного забезпечення для управління запасами обрано середовище розробки Visual Studio, яке є основним інструментом для розробки програм на мові C# і платформі .NET, що дозволяє максимально ефективно використовувати можливості цієї мови. Всі необхідні інструменти для роботи з C#, включаючи інтегровані фреймворки та бібліотеки, доступні безпосередньо з середовища Visual Studio. Крім того, Visual Studio надає ефективні інструменти налагодження, включаючи підтримку покрокового виконання коду, перевірки значень змінних у реальному часі та моніторинг ресурсів.

Створено модель взаємодії користувача з програмним забезпеченням для управління запасами, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для управління запасами.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для управління запасами

Структуру програмного забезпечення для управління запасами наведено на рис. 3.1.

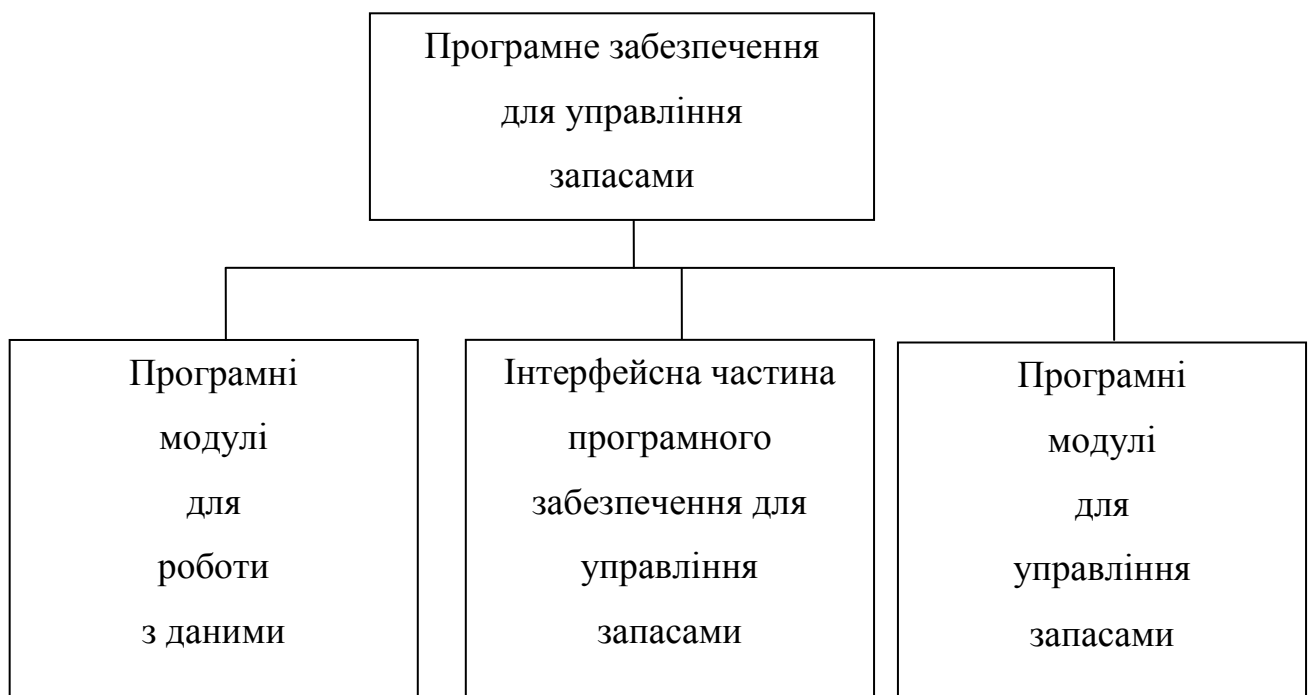


Рисунок 3.1 – Структура програмного забезпечення для управління запасами

Програмне забезпечення для управління запасами включає три основні компоненти:

– програмні модулі для роботи з даними забезпечують функції відкриття та збереження файлів з даними для управління запасами. Також ці модулі забезпечують завантаження та попередній аналіз вхідних даних. До складу цієї групи модулів входить модуль введення даних, який виконує завантаження вихідної інформації, зокрема, вибір файлу з даними для управління запасами, введення інформації, попередню перевірку правильності

введених значень для розв'язання задачі управління запасами, з виведенням повідомлень про помилки з детальними поясненнями у разі їх виявлення, налаштування граничних параметрів, цільових функцій та часу виконання методів управління запасами. Модуль виведення даних зберігає результати обчислень задачі управління запасами у файл звіту з можливістю друку або відображає їх на екрані у зручному форматі для подальшого використання користувачем. Довідковий модуль забезпечує надання доступу до інформації про елементи інтерфейсу, надає детальну довідкову інформацію про різні класи задач управління запасами та методи їх розв'язку;

– інтерфейсна частина – містить набір користувацьких інтерфейсних форм для взаємодії з програмним забезпеченням для управління запасами. У конфігураційних файлах інтерфейсної частини програмного забезпечення для управління запасами зберігаються дані про початкові та граничні умови, необхідні для обчислення значень цільової функції. Програмне забезпечення для управління запасами автоматично перевіряє відповідність конфігураційного файлу обраному алгоритму та контролює правильність його формату. У разі виявлення помилок відображаються дружні повідомлення з поясненнями та коментарями. Модуль вибору алгоритму управління запасами дозволяє користувачеві вибрати алгоритм для вирішення задач управління запасами. Після вибору класу задач управління запасами програма автоматично підбирає оптимальний алгоритм. Якщо вибір зробити неможливо, користувач може самостійно вибрати алгоритм управління запасами;

– програмні модулі для управління запасами – включають реалізацію різних підходів для управління запасами. У цих модулях реалізовано алгоритми для вирішення задач управління запасами, зокрема, управління запасами при стабільному попиті та періодичних поставках, управління запасами при безперервному розподілі попиту, управління багатомноменклатурними запасами з обмеженою складською ємністю, управління запасами з дискретним розподілом попиту, динамічне управління

запасами. Модуль аналізу та обробки результатів перевіряє та аналізує коректність отриманих рішень. Він також дає змогу налаштовувати необхідну точність розв'язку та допустимий час роботи методів.

3.2 Функціонування програмного забезпечення для управління запасами

Функціонування програмного забезпечення для управління запасами подамо за допомогою схеми, зображеної на рис. 3.2.

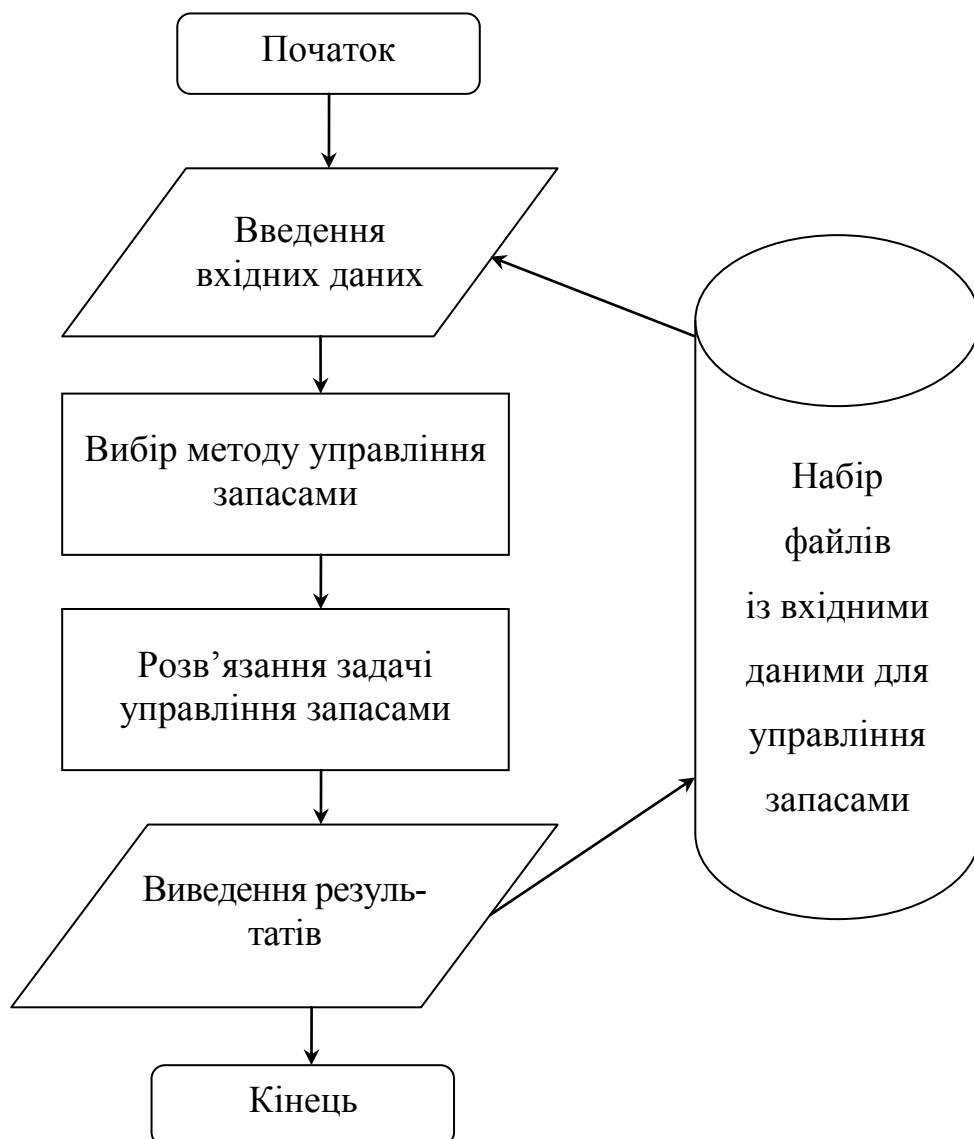


Рисунок 3.2 – Схема функціонування програмного забезпечення для управління запасами

Робота програмного забезпечення для управління запасами розпочинається з введення користувачем вхідних даних щодо складу, задачі управління запасами. На наступному етапі обирається метод управління запасами. Після цього виконується розв'язання задачі управління запасами. Далі виводиться результат розв'язання задачі управління запасами.

3.3 Використання генетичних алгоритмів для управління запасами

Оскільки завдання управління запасами є завданнями оптимізації, доцільно для її розв'язання використовувати генетичні методи [20]-[23].

Під час вирішення задач управління запасами необхідно враховувати декілька важливих факторів, які визначають контекст. Сюди входять обсяг продукції, моделі попиту, розвиток системи, можливий дефіцит, затримки постачання тощо. Велика кількість цих факторів збільшує складність задач управління запасами. Зокрема, робота з багатьма типами продукції є одним із ключових ускладнюючих аспектів. Через значну кількість можливих варіантів стає практично неможливим розрахувати всі складні варіанти вручну, що також впливає на витрати або можливу економію [20]-[23].

Загалом задача управління запасами полягає в ефективному плануванні доставки товарів на склад з метою мінімізації витрат на зберігання та обробку. Використання генетичних алгоритмів для вирішення цих задач показує добрі результати, особливо при роботі з великою кількістю варіантів рішень. Особливість управління запасами також полягає в динамічному характері вхідних даних. Початковий план постачань ґрунтується на прогнозах, зроблених раніше, але під час отримання актуальних даних необхідно коригувати рішення безпосередньо в процесі доставки, що допомагає вирішувати динамічні задачі інвентаризації [20]-[23].

Оскільки точні методи вирішення задач управління запасами підходять лише для найпростіших випадків, на практиці широко застосовують

наближені методи. У генетичних алгоритмах для пошуку оптимального рішення використовують набір хромосом. Кожна хромосома складається з декількох полів (генів), що містять інформацію про певні параметри, а значення представляють шукані характеристики. За допомогою механізмів відбору батьків, схрещування, мутації та відбору найбільш пристосованих хромосом відбувається еволюція рішень, адаптованих до заданих умов, з метою пошуку оптимального простору рішень [20]-[23].

Методи локальної оптимізації після застосування генетичного пошуку можуть використовуватися для покращення найкращої хромосоми шляхом оцінки значення цільової функції навколо вибраної точки в просторі параметрів. Одна з переваг генетичних алгоритмів полягає в тому, що вони дозволяють включати інформацію про розподіл хромосом. Кожна хромосома не лише представляє можливе рішення, але й під час еволюційного процесу вона оптимізується без необхідності подальших доопрацювань. У задачах управління запасами це допомагає ефективно відображати дані у вигляді єдиного набору параметрів, хоча генетичні оператори не завжди здатні повністю контролювати і коригувати результати своїх дій [20]-[23].

У процесі розв'язання задачі управління запасами на основі генетичного підходу поступово усуваються негативні генетичні значення, а хромосоми відбираються на основі найкращого набору характеристик, тобто мінімізації цільової функції, яка відображає загальні витрати на доставку та зберігання. Щоб прискорити цей процес, може використовуватися метод локальної оптимізації, який працює з окремою хромосомою. Окрім прискорення еволюційних процесів, цей підхід динамічно коригує ймовірності появи нових генетичних значень під час перебудови хромосом і створює нові хромосоми для поповнення популяції [20]-[23].

На певному етапі, після початку оцінки генома, стартує процес визначення рівня генетичної значущості позитивних хромосом. Кожне можливе генетичне значення кількісно оцінюється як найкраща хромосома

серед наявних варіантів. Ймовірні значення можуть суттєво відрізнятися через мутацію або появу нових хромосом [20]-[23].

Страховий запас використовується для покриття потреб у товарах, які залежать від фактичного попиту та очікуваного часу виконання замовлень. У спеціалізованому програмному забезпеченні рівень страхового запасу можна встановити відповідно до прогнозованих потреб [20]-[23].

Однією з основних переваг генетичних алгоритмів для вирішення задач управління ресурсами є їхня проста адаптивність (у порівнянні з традиційними методами), яка дозволяє застосовувати їх до різних завдань. Використовуючи один і той же алгоритм, можна вирішувати як прості, так і складні задачі, що включають час чи змінний попит, з обмеженнями або без них. Надійність вихідних даних забезпечується властивостями стабільності генетичного пошуку, що може змінюватися відповідно до умов роботи. Крім того, цей метод можна комбінувати з іншими підходами, які передбачають коригування розкладу в залежності від різниці між фактичними і прогнозованими параметрами [20]-[23].

Існують різні моделі для вирішення задач управління запасами. Основні характеристики цих моделей включають: ланцюг поставок, попит і пропозицію, відновлення запасів, вартість обслуговування запасів, обмеження, управління ризиками.

Попит на продукцію може бути постійним або непередбачуваним, фіксованим чи випадковим. Існують різні підходи до постачання: безпосереднє постачання; постачання після певного періоду; та випадкові постачання. Функціональна вартість визначає ефективність запропонованого рішення і включає адміністративні витрати, витрати на запаси, витрати на кожне нове замовлення та штрафи за відсутність товару. Можливі різні комбінації цих елементів.

Витрати на зберігання залежать від кількості товару на складі в кінці періоду, а також від нелінійної функції, яка визначає оптимальну кількість запасів. Витрати на доставку залежать від фіксованої кількості поставок і

пропорційної кількості транспортних засобів, необхідних для транспортування. Штрафи за недотримання умов постачання можуть базуватися на середній кількості помилок (збоїв) за певний період або на кількості невиконаних замовлень у кінці періоду.

В управлінні запасами існують обмеження, такі як максимальна ємність для зберігання, допустима вага, важливі статистичні показники, швидкість постачання за певний період, витрати на транспортування, швидкість доставки та можливість виникнення помилок.

Ефективна стратегія управління запасами повинна мінімізувати функцію витрат, яка є індикатором ефективності. Для досягнення максимального результату в основному процесі, розробники дослідили основні варіанти управління складом, такі як управління запасами при фіксованому попиті та чергових поставках, управління кількома видами продукції з обмеженим простором для зберігання, а також управління динамічними запасами для забезпечення попиту.

Для кожного типу задач визначається оптимальний алгоритм вирішення. Розглядаються властивості алгоритмів, призначених для вирішення задач управління запасами з різними умовами. Проблеми управління запасами з ймовірнісними параметрами ілюструються через аналіз попиту, який є складним елементом у прогнозуванні реальних сценаріїв. Важливо зазначити, що дискретні та безперервні методи потребують відповідного підходу для виконання різних функцій. У ролі статистичного критерію обирається стандартне правило, яке є ідеальною моделлю стохастичних процесів у логістиці.

Для демонстрації вирішення різних проблем управління складом використовується типовий приклад, оскільки це є важливим аспектом при роботі з кількома складами. В контексті динамічного управління запасами, алгоритми для оптимізації планування допускають різну кількість розрахункових періодів. Використання ітераційних алгоритмів дозволяє виконувати пошук без спостереження, що робить розрахунки більш точними

та наближеними до реальних умов. Для спрощення обчислень вартість виконання замовлень та запаси вважаються лінійними функціями.

Оскільки адміністрування замовлень вимагає мінімальних ресурсів, розробник прагнув знайти рішення для всіх можливих ситуацій. Це також допомагає приховати роботи, які можуть становити загрозу, але не впливають на швидко зростаючі популяції. Існує можливість вказати максимальний обсяг складу та терміни доставки. Для досягнення оптимальної продуктивності кожен тип послуги супроводжується відповідними міркуваннями.

Генетичний алгоритм є потужним інструментом для вирішення складних задач динамічного зберігання з багатьма параметрами. Ці алгоритми, які є адаптивними методами пошуку, активно використовуються для оптимізації. Вони ґрунтуються на генетичних процесах, що відбуваються в живих організмах, де популяція еволюціонує відповідно до законів природного відбору. Існує різні способи реалізації концепції життєвого циклу в генетичних алгоритмах [20]-[23].

При реалізації генетичного алгоритму застосовуються три основні операції: відбір, перехід та мутація, які відповідають етапам еволюції. У цьому випадку використовується метод відбору на основі випадкового вибору невеликої популяції, що дозволяє передавати найкращі індивіди до наступної генерації, що підтверджує їхню ефективність у створенні різноманітних рішень. Також впроваджена оптимізована модель, яка швидко ідентифікує та зберігає найкращі рішення, якщо такі є. Випадкові перетворення, хоча й знижують узгодженість алгоритму, значно розширюють область пошуку.

3.4 Проєктування інтерфейсу програмного забезпечення для управління запасами

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для управління запасами виконано з урахуванням вимог

технічного завдання. При розробленні програмного забезпечення для управління запасами враховані функціональні вимоги до програми:

- підтримка можливості управління запасами різними методами;
- підтримка можливості завдання значень параметрів (можливість дефіциту, розподіл попиту та ін.) підходу до управління запасами;
- можливість табличного подання розрахункових значень запасів товарів на складі;
- можливість динамічного керування запасами (визначення рівня запасів товарів для різних періодів);
- подання інформації про програмне забезпечення для управління запасами;
- підтримка можливості роботи з файлами даних.

Інтерфейс головної форми програмного забезпечення для управління запасами наведено на рис. 3.5.

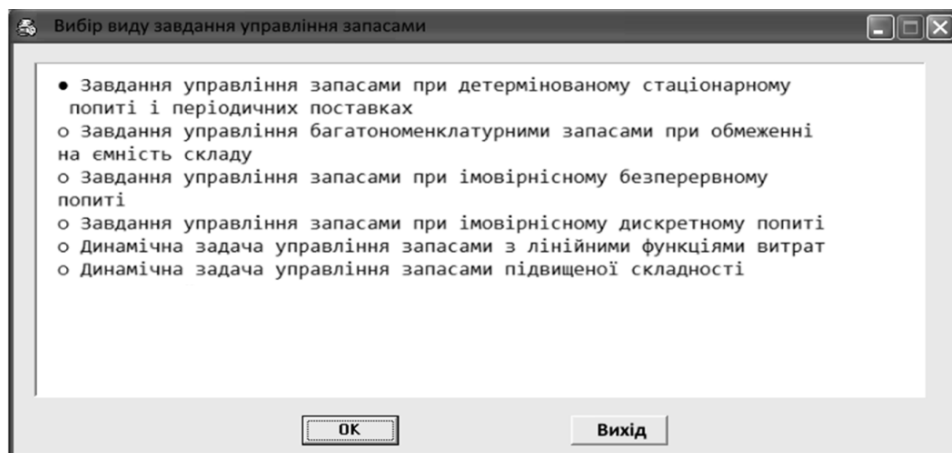


Рисунок 3.5 – Інтерфейс головної форми програмного забезпечення для управління запасами

Форма введення вхідних даних при розв'язанні завдання управління запасами при постійному попиті та періодичних поставках наведена на рис. 3.6.

Завдання управління запасами при детерміновано...

Файл Налаштування Розрахунок Допомога

Витрати на зберігання од.продукту в од.часу, грн. 2

Уд.штраф за дефіцит од.продукції в од.часу, грн. неможливий

Витрати за одну поставку, грн. 100

Інтенсивність попиту, кг/міс. 86

Інтенсивність поставок, кг/міс. МИТТЄВІ

Рекомендовані значення

Оптимальний період поставок, міс. 1.078

Оптимальний розмір партії, кг. 92.736

Витрати в одиницю часу, грн. 185.472

Рисунок 3.6 – Форма введення вхідних даних при розв’язанні завдання управління запасами при постійному попиті та періодичних поставках

Форму введення вхідних даних та розв’язання завдання управління запасами різних товарів наведено на рис. 3.7.

Завдання управління багатомономенклатурними запасами при обмеженні на ємність складу

Файл Налаштування Розрахунок Допомога

Кількість видів продукції, шт. 12

Обмеження на ємність складу, м.куб. 80

Вид продукції	Вит. на замов., грн.	Витр. на зберіг., грн.	Інтенс. попиту	Пр-ір для од.товару, м.куб	Опт. ро-ір з
сахар	10	0,3	2	1	4.984
соль	5	0,1	4	1	5.326
мука 1-й сорт	15	0,2	4	1	8.915
мука 2-й сорт	7	0,3	3	1	5.108
крахмал	10	0,2	5	1	8.138
сода	15	0,6	2	1	5.605
крупа овсяная	5	0,1	2	0,5	5.147
крупа гречневая	3	0,1	3	0,5	4.883
крупа рисовая	10	0,4	7	1	9.048
макарони "Спагетти"	10	0,5	2	1	4.701
макарони "Рожки"	12	0,3	10	2	9.066
сухие сливки	10	0,3	2	1	4.984

Рисунок 3.7 – Форма введення вхідних даних та розв’язання завдання управління запасами різних товарів

Форму введення вхідних даних та розв’язання завдання управління запасами при ймовірнісному попиті наведено на рис. 3.8.

Рисунок 3.8 – Форма введення вхідних даних та розв’язання завдання управління запасами при ймовірнісному попиті

Форму введення вхідних даних та розв’язання завдання управління динамічними запасами наведено на рис. 3.9.

Період	Попит	Витрати	Р-ня про закуп.	Опт. р-р замовл.
1-й	90	300.000	закупівля	215
2-й	125	550.000		0
3-й	140	850.000	закупівля	240
4-й	100	1050.000		0
5-й	45	1230.000	закупівля	105
6-й	60	1470.000		0
7-й	130	1770.000	закупівля	130

Рисунок 3.9 – Форма введення вхідних даних та розв’язання завдання управління динамічними запасами

Форма введення вхідних даних для розв’язання задачі управління запасами підвищеної складності наведена на рис. 3.10.

Налаштування генетичного алгоритму

☒ Завантаження даних з конфігураційного файлу

☐ Введення даних в ручному режимі

Кі-ть розрахункових періодів

Кількість хромосом

Макс. розмір замовлення

☒ Наявність штрафів за дефіцит товару

Рисунок 3.10 – Форма введення вхідних даних для завдання управління запасами підвищеної складності

Форму з результатами розв’язання завдання управління запасами підвищеної складності наведено на рис. 3.11.

Динамічна задача управління запасами підвищеної складності

Файл Розрахунок Допомога

Період	1-й	2-й	3-й	4-й	5-й	6-й
Полит	32	180	92	16	33	27
Опт. р-р замов.	118	125	38	41	79	95

Ціна доставки: 55

Витрати на зберіг.: 10

Штраф: 49

Тривалість розрахунку: 00:01:14

Розрахувати

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	118	125	38	41	79	95	39	119	285	160	115	198	48
2	118	125	38	41	79	95	39	119	285	160	115	198	48
3	118	125	38	41	79	95	39	119	285	160	393	198	48
4	118	125	38	41	79	95	39	119	285	160	115	198	48
5	118	125	38	41	79	95	39	119	285	160	115	198	48
6	118	125	38	41	79	95	39	119	285	160	115	198	48
7	118	125	38	41	79	95	39	119	285	160	115	198	48
8	118	125	38	41	79	95	39	119	285	160	115	198	48
9	118	125	38	41	79	95	39	119	285	160	115	198	48
10	118	125	38	41	79	95	39	119	285	160	115	198	48
11	118	125	38	41	79	95	39	119	285	160	115	198	48
12	118	125	38	41	79	95	39	119	285	160	115	198	48
13	118	125	38	41	79	95	39	119	285	160	115	198	48
14	118	125	38	41	79	95	39	119	285	160	115	198	48
15	118	125	38	41	79	95	39	119	285	160	73	198	48
16	118	125	38	41	79	95	39	119	285	160	115	198	48

Рисунок 3.11 – Форма з результатами розв’язання завдання управління запасами підвищеної складності

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для управління запасами.

Запропоновано структуру програмного забезпечення для управління запасами. Визначено, що програмне забезпечення для управління запасами включає три основні компоненти: програмні модулі для роботи з даними, інтерфейсна частина, програмні модулі для управління запасами.

Описано функціонування програмного забезпечення для управління запасами. Визначено, що робота програмного забезпечення для управління запасами розпочинається з введення користувачем вхідних даних щодо складу, задачі управління запасами, на наступному етапі обирається метод управління запасами, після цього виконується розв'язання задачі управління запасами, далі виводиться результат розв'язання задачі управління запасами.

Описано особливості реалізації програмного забезпечення для управління запасами. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для управління запасами.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу управління запасами. Програма написана на мові C#, яка має високу продуктивність завдяки ефективній роботі з пам'яттю, а також дозволяє створювати зручні та інтерактивні користувацькі інтерфейси. В якості середовища розробки для створення програмного забезпечення для управління запасами обрано Visual Studio, яке є основним інструментом для розробки програм на мові C# і платформі .NET, що дозволяє максимально ефективно використовувати можливості цієї мови.

Програмне забезпечення для управління запасами забезпечує виконання таких функцій:

- підтримка можливості управління запасами різними методами;
- підтримка можливості завдання значень параметрів (можливість дефіциту, розподіл попиту та ін.) підходу до управління запасами;
- можливість табличного подання розрахункових значень запасів товарів на складі;
- можливість динамічного керування запасами (визначення рівня запасів товарів для різних періодів);
- подання інформації про програмне забезпечення для управління запасами;
- підтримка можливості роботи з файлами даних.

Для експлуатації програмного забезпечення для управління запасами необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять мінімум 4 Гб;
- мінімум 500 Мб вільного простору на диску;
- процесор не гірше, ніж двох ядровий з тактовою частотою 2,0 ГГц;
- встановлена операційна система Windows 10 або новіша версія.

Функціональні характеристики розробленого програмного забезпечення для управління запасами наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для управління запасами наведено у додатку Б.

4.2 Характеристики програми для управління запасами

Програмне забезпечення для підтримки процесу управління запасами підтримує автоматизацію та оптимізацію процесів контролю та зберігання товарів.

Програмне забезпечення для управління запасами включає три основні компоненти: програмні модулі для роботи з даними, інтерфейсну частину та програмні модулі для управління запасами. Програмні модулі для роботи з даними забезпечують функції відкриття та збереження файлів з даними для управління запасами. Інтерфейсна частина містить набір користувацьких інтерфейсних форм для взаємодії з програмним забезпеченням для управління запасами. У конфігураційних файлах інтерфейсної частини програмного забезпечення для управління запасами зберігаються дані про початкові та граничні умови, необхідні для обчислення значень цільової функції. Програмне забезпечення для управління запасами автоматично перевіряє відповідність конфігураційного файлу обраному алгоритму та контролює правильність його формату. У разі виявлення помилок відображаються дружні повідомлення з поясненнями та коментарями. Модуль вибору алгоритму управління запасами дозволяє користувачеві вибрати алгоритм для вирішення задач управління запасами. Після вибору класу задач управління запасами програма автоматично підбирає оптимальний алгоритм. Програмні модулі для управління запасами включають реалізацію різних підходів для управління запасами. У цих модулях реалізовано алгоритми для вирішення задач управління запасами, зокрема, управління запасами при стабільному попиті та періодичних поставках, управління запасами при безперервному розподілі

попиту, управління багатомножинними запасами з обмеженою складською ємністю, управління запасами з дискретним розподілом попиту, динамічне управління запасами.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для управління запасами треба запустити виконуваний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для управління запасами є тип розв'язуваної задачі управління запасами, параметри розрахунку, параметри управління складом та ін.

Вихідними даними програмного забезпечення для управління запасами є результати розв'язання задачі управління запасами, наприклад, витрати на закупівлю товарів у кожний розрахунковий період та ін.

4.3.3 Повідомлення

Користувач програмного забезпечення для управління запасами може отримати такі повідомлення: повідомлення про некоректні значення вхідних параметрів, неправильний формат файлу та ін.

4.4 Виконання програмного забезпечення для управління запасами

Після запуску програмного забезпечення для управління запасами користувачу відображається головна форма (рис. 4.1), де користувач повинен

обрати завдання управління запасами, яке необхідно розв'язати за допомогою програми.

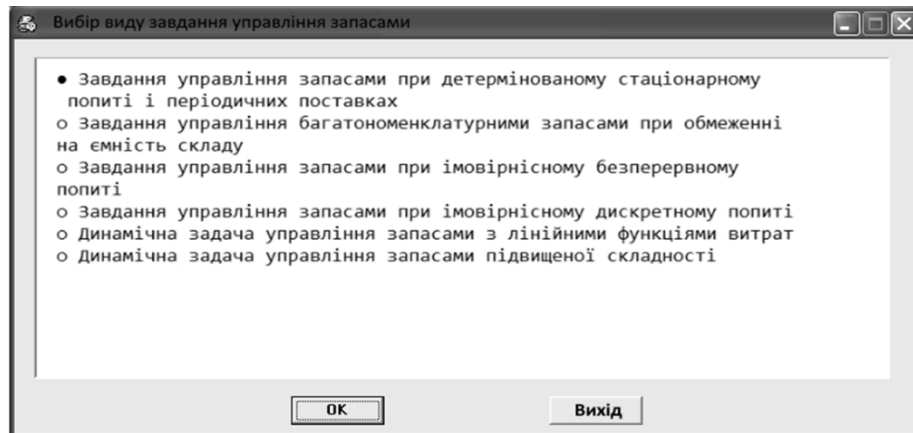


Рисунок 4.1 – Інтерфейс головної форми програмного забезпечення для управління запасами

Після вибору типу завдання управління запасами, яке необхідно розв'язати, відкривається форма, де необхідно задати вхідні дані. Як приклад такої форми наведено форму для введення вхідних даних при розв'язанні завдання управління запасами при постійному попиті та періодичних поставках (рис. 4.2).

Рисунок 4.2 – Форма введення вхідних даних при розв'язанні завдання управління запасами при постійному попиті та періодичних поставках

Після введення вхідних даних, зокрема, витрат на поставку та зберігання одиниці товару, інтенсивність попиту та поставок, у відповідних полях форми виводяться вихідні значення розрахунку, зокрема, оптимальний період поставок, розмір партії та ін.

Форму введення вхідних даних та розв'язання завдання управління запасами різних товарів наведено на рис. 4.3. Тут користувач задає кількість видів продукції, обмеження на розмір складу, а також деякі параметри, що характеризують особливості кожного виду товару. Результати розв'язання завдання управління запасами різних типів товарів наводяться у табличному вигляді у стовбці оптимального розміру замовлення. З метою скорочення часу введення користувачем вхідних даних їх можна завантажити з файлу, натиснувши відповідну кнопку. Результати можна завантажити у файл.

Завдання управління багатомножинними запасами при обмеженні на ємність складу

Файл Налаштування Розрахунок Допомога

Кількість видів продукції, шт. 12

Обмеження на ємність складу, м.куб. 80

Вид продукції	Вит. на замов., грн.	Витр. на зберіг., грн.	Інтенс. попиту	Пр-ір для од.товару, м.куб	Опт. ро-ір з
сахар	10	0,3	2	1	4.984
соль	5	0,1	4	1	5.326
мука 1-й сорт	15	0,2	4	1	8.915
мука 2-й сорт	7	0,3	3	1	5.108
крахмал	10	0,2	5	1	8.138
сода	15	0,6	2	1	5.605
крупа овсяная	5	0,1	2	0,5	5.147
крупа гречневая	3	0,1	3	0,5	4.883
крупа рисовая	10	0,4	7	1	9.048
макарони "Спагетти"	10	0,5	2	1	4.701
макарони "Рожки"	12	0,3	10	2	9.066
сухие сливки	10	0,3	2	1	4.984

Рисунок 4.3 – Форма введення вхідних даних та розв'язання завдання управління запасами різних товарів

Форму введення вхідних даних та розв'язання завдання управління запасами при ймовірнісному попиті наведено на рис. 4.4. Тут користувач задає ймовірнісні характеристики попиту, зокрема, тип розподілу, математичне очікування та середньоквадратичне відхилення, можливість дефіциту та інші характеристики. Результат розв'язання завдання управління запасами наводиться у полі оптимального розміру замовлення.

Рисунок 4.4 – Форма введення вхідних даних та розв'язання завдання управління запасами при ймовірнісному попиті

Форму введення вхідних даних та розв'язання завдання управління динамічними запасами наведено на рис. 4.5. При розв'язанні цього типу задачі користувач задає параметри попиту на продукцію у відповідні часові періоди. Результат розв'язання завдання управління динамічними запасами наводиться у стовбці оптимального розміру замовлення.

Період	Попит	Витрати	Р-ня про закуп.	Опт. р-р замовл.
1-й	90	300.000	закупівля	215
2-й	125	550.000		0
3-й	140	850.000	закупівля	240
4-й	100	1050.000		0
5-й	45	1230.000	закупівля	105
6-й	60	1470.000		0
7-й	130	1770.000	закупівля	130

Рисунок 4.5 – Форма введення вхідних даних та розв’язання завдання управління динамічними запасами

У випадку вибору користувачем на початковій формі необхідності розв’язання задачі управління запасами підвищеної складності відображується форма введення вхідних даних для відповідної задачі (рис. 4.6). Тут користувач обирає варіант введення вхідних даних (за допомогою конфігураційного файлу або ручне введення). Також вводиться кількість розрахункових періодів. Крім того, оскільки це оптимізаційне завдання розв’язується за допомогою генетичного алгоритму, необхідно ввести кількість хромосом (розв’язків, що аналізуються на кожній ітерації оптимізаційного процесу).

Налаштування генетичного алгоритму

☒ Завантаження даних з конфігураційного файлу

☐ Введення даних в ручному режимі ☒

Кі-ть розрахункових періодів

Кількість хромосом

Макс. розмір замовлення

☒ Наявність штрафів за дефіцит товару ☒

Рисунок 4.6 – Форма введення вхідних даних для завдання управління запасами підвищеної складності

Форму з результатами розв’язання завдання управління запасами підвищеної складності наведено на рис. 4.7. Тут відображаються оптимальні розміри замовлення для кожного періоду, а також деякі інші характеристики.

Динамічна задача управління запасами підвищеної складності

Файл Розрахунок Допомога

Період 1-й 2-й 3-й 4-й 5-й 6-й

Попит	32	180	92	16	33	27
Опт. р-р замов.	118	125	38	41	79	95

Ціна доставки 55

Витрати на зберіг. 10

Штраф 49

Тривалість розрахунку 00:01:14

Розрахувати

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	118	125	38	41	79	95	39	119	285	160	115	198	48
2	118	125	38	41	79	95	39	119	285	160	115	198	48
3	118	125	38	41	79	95	39	119	285	160	393	198	48
4	118	125	38	41	79	95	39	119	285	160	115	198	48
5	118	125	38	41	79	95	39	119	285	160	115	198	48
6	118	125	38	41	79	95	39	119	285	160	115	198	48
7	118	125	38	41	79	95	39	119	285	160	115	198	48
8	118	125	38	41	79	95	39	119	285	160	115	198	48
9	118	125	38	41	79	95	39	119	285	160	115	198	48
10	118	125	38	41	79	95	39	119	285	160	115	198	48
11	118	125	38	41	79	95	39	119	285	160	115	198	48
12	118	125	38	41	79	95	39	119	285	160	115	198	48
13	118	125	38	41	79	95	39	119	285	160	115	198	48
14	118	125	38	41	79	95	39	119	285	160	115	198	48
15	118	125	38	41	79	95	39	119	285	160	73	198	48
16	118	125	38	41	79	95	39	119	285	160	115	198	48

Рисунок 4.7 – Форма з результатами розв’язання завдання управління запасами підвищеної складності

При виборі пункту меню Допомога користувачу відображається довідка по роботі з програмним забезпеченням для управління запасами.

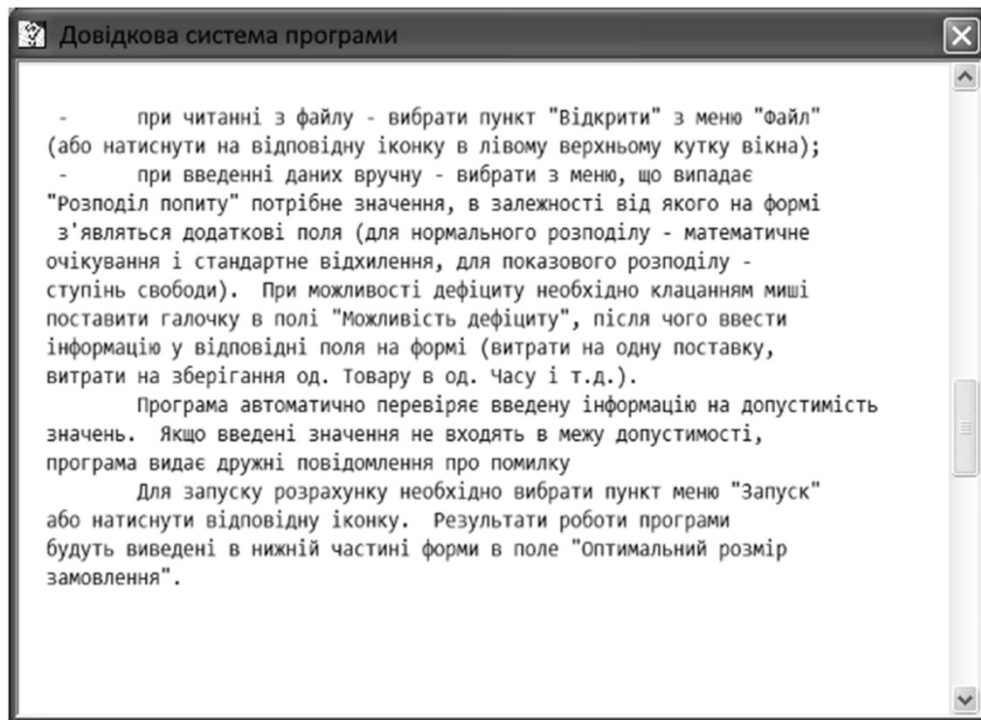


Рисунок 4.8 – Довідка по роботі з ПЗ для управління запасами

4.5 Тестування програмного забезпечення для управління запасами

Виконано тестування програмного забезпечення для управління запасами.

Виявлено, що програма для підтримки процесу управління запасами функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу управління запасами.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу управління запасами.

4.6 Висновки за розділом 4

Описано програмне забезпечення для управління запасами. Виконано тестування розробленого програмного забезпечення для управління запасами. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління запасами.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для управління запасами.

Відзначено, що управління запасами допомагає компаніям приймати обґрунтовані рішення про те, що і в яких кількостях варто зберігати. Цей процес забезпечує своєчасну ідентифікацію наявних запасів та відповідну реакцію на зміни, щоб завжди мати достатню кількість товарів для виконання замовлень клієнтів і вчасно попереджати про можливі нестачі. Існують різні підходи до управління запасами, деякі з них використовують математичні моделі та аналітику для оптимізації процесів, інші базуються на певних процедурах. Проаналізовано підходи до розв'язання задачі управління запасами. Проаналізовано етапи створення програмного забезпечення управління запасами. Відзначено, що важливим є прогнозування рівня запасів з метою забезпечення наявності достатньої кількості запасів для виконання очікуваних продажів протягом визначеного періоду. Показано, що однією з поширених проблем управління складом є зайве накопичення запасів, яке може створювати фінансовий тягар і призводити до застарілих товарів, з іншого боку, недостатні запаси можуть спричинити їх дефіцит через затримки у виробництві та незадоволеність клієнтів. Тому для усунення зазначених недоліків доцільно використовувати спеціалізоване програмне забезпечення для управління запасами.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для управління запасами. Відзначено, що деякі програмні засоби для управління запасами можуть бути досить дорогими, особливо для малого та середнього бізнесу. Деякі програмні засоби для управління запасами потребують складного налаштування та адаптації під конкретні бізнес-процеси. Інші програмні системи для управління запасами можуть не бути достатньо гнучкими для масштабування, що у деяких

випадках може вимагати оновлення або повної заміни програмного забезпечення, що супроводжується додатковими витратами та зусиллями працівників підприємства. Крім того, деякі програми для управління запасами можуть не враховувати унікальні вимоги або специфіку конкретного бізнесу. Тому актуальною є розробка програмного забезпечення для управління запасами.

Сформульовано функціональні вимоги до програмного забезпечення для управління запасами.

Для реалізації програмного забезпечення для управління запасами обрано мову програмування C#, яка має високу продуктивність завдяки ефективній роботі з пам'яттю, а також дозволяє створювати зручні та інтерактивні користувацькі інтерфейси, що підвищує зручність роботи для співробітників, які керують запасами. Крім того, програми, написані на мові C#, можуть легко масштабуватися, щоб обробляти збільшені обсяги даних без втрати продуктивності, що є важливим для великих підприємств з великою кількістю товарів та постачальників.

Для створення програмного забезпечення для управління запасами обрано середовище розробки Visual Studio, яке є основним інструментом для розробки програм на мові C# і платформі .NET, що дозволяє максимально ефективно використовувати можливості цієї мови. Всі необхідні інструменти для роботи з C#, включаючи інтегровані фреймворки та бібліотеки, доступні безпосередньо з середовища Visual Studio. Крім того, Visual Studio надає ефективні інструменти налагодження, включаючи підтримку покрокового виконання коду, перевірки значень змінних у реальному часі та моніторинг ресурсів.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для управління запасами, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для управління запасами.

Практична цінність результатів роботи полягає у тому, що розроблено програмне забезпечення для управління запасами.

Запропоновано структуру програмного забезпечення для управління запасами. Визначено, що програмне забезпечення для управління запасами включає три основні компоненти: програмні модулі для роботи з даними, інтерфейсна частина, програмні модулі для управління запасами.

Описано функціонування програмного забезпечення для управління запасами. Визначено, що робота програмного забезпечення для управління запасами розпочинається з введення користувачем вхідних даних щодо складу, задачі управління запасами, на наступному етапі обирається метод управління запасами, після цього виконується розв'язання задачі управління запасами, далі виводиться результат розв'язання задачі управління запасами.

Описано особливості реалізації програмного забезпечення для управління запасами. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для управління запасами.

Виконано тестування розробленого програмного забезпечення для управління запасами. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління запасами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Inventory Management Guide: Definition, Process & Techniques [Electronic resource]. – Access mode: <https://bestarion.com/inventory-management/>.
2. The complete guide to inventory management [Electronic resource]. – Access mode: <https://monday.com/blog/project-management/inventory-management/>.
3. What is Inventory Management? Methods, Examples & Benefits [Electronic resource]. – Access mode: <https://tipalti.com/financial-operations-hub/inventory-management/>.
4. Inventory Control Defined: Best Practices, Systems, & Management [Electronic resource]. – Access mode: <https://www.netsuite.com/portal/resource/articles/inventory-management/what-are-inventory-management-controls.shtml>.
5. What is Inventory Management? Benefits, Types, & Techniques [Electronic resource]. – Access mode: <https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management.shtml>.
6. Essential Tips for Effective Inventory Management [Electronic resource]. – Access mode: <https://www.businessnewsdaily.com/10613-effective-inventory-management.html>.
7. Mastering Inventory Management: A Guide to Success in Business [Electronic resource]. – Access mode: <https://overstocktrader.com/blog/inventory-management/>.
8. Inventory Management Techniques to Optimize Your Inventories [Electronic resource]. – Access mode: <https://creately.com/guides/inventory-management-techniques/>.
9. Best Inventory Management Software [Electronic resource]. – Access mode: <https://www.g2.com/categories/inventory-management>.

10. Find the best Inventory Management Software [Electronic resource]. – Access mode: <https://www.softwareadvice.com/inventory-management/>.

11. Best Inventory Management Software [Electronic resource]. – Access mode: <https://softwareconnect.com/roundups/best-inventory-management-software/>.

12. The 14 Best Inventory Management Software for Small Businesses [Electronic resource]. – <https://www.business.org/finance/inventory-management/best-inventory-management-software/>.

13. Ordoro [Electronic resource]. – Access mode: <https://www.ordoro.com/features/inventory-management>.

14. Inflow Inventory [Electronic resource]. – Access mode: <https://www.inflowinventory.com/>.

15. Cin7 [Electronic resource]. – Access mode: <https://www.cin7.com/>.

16. Learn C# [Electronic resource]. – Access mode: <https://dotnet.microsoft.com/en-us/learn/csharp#>.

17. C# Tutorial [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/csharp-programming-language/>.

18. Introduction to Visual Studio [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/introduction-to-visual-studio/>.

19. Microsoft's do-it-all IDE Visual Studio 2022 came out late last year [Electronic resource]. – Access mode: https://www.theregister.com/2022/01/25/visual_studio_2022/.

20. The Computer Science of Evolution: an Introduction to Genetic Algorithms [Electronic resource]. – Access mode: <https://www.freecodecamp.org/news/the-computer-science-of-evolution-an-introduction-to-genetic-algorithms-b3871286c7e7/>.

21. Genetic Algorithms: Search and Optimization by Natural Selection [Electronic resource]. – Access mode: <https://www.toptal.com/algorithms/genetic-algorithms>.

22. An Introduction to Genetic Algorithms [Electronic resource]. – Access mode: <https://www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pdf>.

23. Genetic Algorithms - Introduction [Electronic resource]. – Access mode: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для управління запасами.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів управління запасами», затверджене наказом Національного університету «Запорізька політехніка» № 398 від 07 жовтня 2024 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу управління запасами.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для управління запасами забезпечує виконання таких функцій:

- підтримка можливості управління запасами різними методами;
- підтримка можливості завдання значень параметрів (можливість дефіциту, розподіл попиту та ін.) підходу до управління запасами;
- можливість табличного подання розрахункових значень запасів товарів на складі;
- можливість динамічного керування запасами (визначення рівня запасів товарів для різних періодів);

- подання інформації про програмне забезпечення для управління запасами;
- підтримка можливості роботи з файлами даних.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для управління запасами повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для управління запасами повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для управління запасами необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для управління запасами необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять мінімум 4 Гб;
- мінімум 500 Мб вільного простору на диску;
- процесор не гірше, ніж двох ядровий з тактовою частотою 2,0 ГГц;
- встановлена операційна система Windows 10 або новіша версія.

А.3.6 Вимоги до маркування і пакування

Програма для управління запасами може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для управління запасами».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для управління запасами є тип розв'язуваної задачі управління запасами, параметри розрахунку, параметри управління складом та ін.

Вихідними даними програмного забезпечення для управління запасами є результати розв'язання задачі управління запасами, наприклад, витрати на закупівлю товарів у кожний розрахунковий період та ін.

ДОДАТОК Б
Фрагмент тексту програми

```

public class SkldItm
{
    public int Id { get; set; }
    public string Imja { get; set; }
    public int Kilkist { get; set; }
    public decimal Cina { get; set; }
    public DateTime LastOnovld { get; set; }
}

public class SkldManagement
{
    private List<SkldItm> _SkldItms;

    public SkldManagement()
    {
        _SkldItms = new List<SkldItm>();
    }

    // Додати товар до складу
    public void AddItm(SkldItm Itm)
    {
        _SkldItms.Add(Itm);
        Console.WriteLine($"Товар      '{Itm.Imja}'      додано      до
складу.");
    }

    // Оновити інформацію про товар
    public void OnovlItm(int ItmId, int newKilkist)
    {
        var Itm = _SkldItms.FirstOrDefault(i => i.Id == ItmId);
        if (Itm != null)
        {
            Itm.Kilkist = newKilkist;
            Itm.LastOnovld = DateTime.Now;
        }
    }
}

```

```

        Console.WriteLine($"Кількість товару '{Itm.Imja}'
оновлено до {newKilkist}.");
    }
    else
    {
        Console.WriteLine($"Товар з ID: {ItmId} не
знайдено.");
    }
}

// Видалити товар з складу
public void DelItm(int ItmId)
{
    var Itm = _SkldItms.FirstOrDefault(i => i.Id == ItmId);
    if (Itm != null)
    {
        _SkldItms.Del(Itm);
        Console.WriteLine($"Товар '{Itm.Imja}' видалено з
складу.");
    }
    else
    {
        Console.WriteLine($"Товар з ID: {ItmId} не
знайдено.");
    }
}

// Отримати інформацію про товар
public SkldItm GetItm(int ItmId)
{
    return _SkldItms.FirstOrDefault(i => i.Id == ItmId);
}

// Отримати всі товари на складі
public List<SkldItm> GetAllItms()
{

```

```

        return _SkldItms;
    }
}

public class SkldPrognosing
{
    // Прогнозування запасів за середнім методом
    public static decimal PrognozUsingAverage(List<decimal>
istorychnData)
    {
        if (istorychnData == null || !istorychnData.Any())
        {
            throw new ArgumentException("Istorychn data cannot be
null or empty.");
        }

        decimal average = istorychnData.Average();
        Console.WriteLine($"Прогноз запасів за середнім методом:
{average}");
        return average;
    }

    // Прогнозування запасів за експоненціальним згладжуванням
    public static decimal
PrognozUsingExponentialSmoothing(List<decimal> istorychnData,
decimal alght)
    {
        if (istorychnData == null || !istorychnData.Any())
        {
            throw new ArgumentException("Istorychn data cannot be
null or empty.");
        }

        if (alght < 0 || alght > 1)
        {

```

```

        throw new ArgumentOutOfRangeException("Alght must be
between 0 and 1.");
    }

    decimal Prognoz = istorychnData.First(); // Початкове
значення
    foreach (var data in istorychnData)
    {
        Prognoz = alght * data + (1 - alght) * Prognoz;
    }

    Console.WriteLine($"Прогноз запасів за експоненціальним
згладжуванням: {Prognoz}");
    return Prognoz;
}

// Прогнозування запасів за допомогою простого лінійного
регресування
public static decimal
PrognozUsingLinearRegression(List<decimal> istorychnData)
{
    if (istorychnData == null || !istorychnData.Any())
    {
        throw new ArgumentException("Istorychn data cannot be
null or empty.");
    }

    int n = istorychnData.Count;
    decimal smX = 0, smY = 0, smXY = 0, smX2 = 0;

    for (int i = 0; i < n; i++)
    {
        smX += i + 1; // X - індекс (1, 2, 3, ...)
        smY += istorychnData[i]; // Y - значення запасів
        smXY += (i + 1) * istorychnData[i];
        smX2 += (i + 1) * (i + 1);
    }
}

```

```

    }

    // Формули для обчислення коефіцієнтів a і b
    decimal b = (n * smXY - smX * smY) / (n * smX2 - smX *
smX);

    decimal a = (smY - b * smX) / n;

    // Прогнозування для наступного періоду
    decimal Prognoz = a + b * (n + 1);
    Console.WriteLine($"Прогноз запасів за допомогою простого
лінійного регресування: {Prognoz}");
    return Prognoz;
}
}

public class Postachaln
{
    public string Imja { get; set; }
    public string ContactInfo { get; set; }

    public Postachaln(string imja, string contactInfo)
    {
        Imja = imja;
        ContactInfo = contactInfo;
    }
}

public class Zamovl
{
    public int ZamovlId { get; private set; }
    public Postachaln Postachaln { get; private set; }
    public List<string> Itms { get; private set; }
    public DateTime ZamovlDate { get; private set; }
    public string Stds { get; set; } // Наприклад:
"Обробляється", "Відправлено", "Доставлено"

```

```

        public Zamovl(int ZamovlId, Postachaln Postachaln,
List<string> Itms)
        {
            ZamovlId = ZamovlId;
            Postachaln = Postachaln;
            Itms = Itms;
            ZamovlDate = DateTime.Now;
            Stds = "Обробляється"; // Початковий статус
        }
    }

public class PostachalnZamovl
{
    private List<Postachaln> Postachalns;
    private List<Zamovl> Zamovls;
    private int nextZamovlId;

    public PostachalnZamovl()
    {
        Postachalns = new List<Postachaln>();
        Zamovls = new List<Zamovl>();
        nextZamovlId = 1;
    }

    // Додавання постачальника
    public void AddPostachaln(Postachaln Postachaln)
    {
        Postachalns.Add(Postachaln);
        Console.WriteLine($"Постачальника {Postachaln.Imja}
додано.");
    }

    // Створення замовлення
    public void CreateZamovl(Postachaln Postachaln, List<string>
Itms)
    {

```

```

        if (!Postachalns.Contains(Postachaln))
        {
            Console.WriteLine("Постачальника не знайдено.");
            return;
        }

        Zamovl newZamovl = new Zamovl(nextZamovlId++, Postachaln,
Items);
        Zamovls.Add(newZamovl);
        Console.WriteLine($"Замовлення {newZamovl.ZamovlId}
створено для постачальника {Postachaln.Imja}.");
    }

    // Оновлення статусу замовлення
    public void OnovlZamovlStds(int ZamovlId, string Stds)
    {
        var Zamovl = Zamovls.Find(o => o.ZamovlId == ZamovlId);
        if (Zamovl != null)
        {
            Zamovl.Stds = Stds;
            Console.WriteLine($"Статус замовлення {ZamovlId}
оновлено на: {Stds}.");
        }
        else
        {
            Console.WriteLine("Замовлення не знайдено.");
        }
    }

    // Виведення інформації про всі замовлення
    public void ShowZamovls()
    {
        Console.WriteLine("Список замовлень:");
        foreach (var Zamovl in Zamovls)
        {

```

```

        Console.WriteLine($"ID: {Zamovl.ZamovlId},
Постачальник: {Zamovl.Postachaln.Imja}, Дата:
{Zamovl.ZamovlDate}, Статус: {Zamovl.Stds}, Товари:
{string.Join(", ", Zamovl.Itms)}");
    }
}

public class SkldManagement
{
    private List<SkldItm> Skld;

    public SkldManagement()
    {
        Skld = new List<SkldItm>();
    }

    // Додавання нового товару в інвентар
    public void AddItm(string imja, int Kilkist, double Cina)
    {
        var Itm = new SkldItm(imja, Kilkist, Cina);
        Skld.Add(Itm);
        Console.WriteLine($"Товар {imja} додано в інвентар з
кількістю {Kilkist} і ціною {Cina}.");
    }

    // Видалення товару з інвентарю
    public void DelItm(string imja)
    {
        var Itm = Skld.Find(i => i.Imja.Equals(imja,
StringComparison.OrdinalIgnoreCase));
        if (Itm != null)
        {
            Skld.Del(Itm);
            Console.WriteLine($"Товар {imja} видалено з
інвентарю.");
        }
    }
}

```

```

    }
    else
    {
        Console.WriteLine($"Товар {imja} не знайдено в
інвентарі.");
    }
}

// Оновлення кількості товару
public void OnovlItmKilkist(string imja, int newKilkist)
{
    var Itm = Skld.Find(i => i.Imja.Equals(imja,
StringComparison.OrdinalIgnoreCase));
    if (Itm != null)
    {
        Itm.Kilkist = newKilkist;
        Console.WriteLine($"Кількість товару {imja} оновлено
на {newKilkist}.");
    }
    else
    {
        Console.WriteLine($"Товар {imja} не знайдено в
інвентарі.");
    }
}

// Виведення всіх товарів в інвентарі
public void ShowSkld()
{
    Console.WriteLine("Список товарів в інвентарі:");
    foreach (var Itm in Skld)
    {
        Console.WriteLine($"Назва: {Itm.Imja}, Кількість:
{Itm.Kilkist}, Ціна: {Itm.Cina}");
    }
}

```

```
// Отримання загальної вартості інвентарю
public double GetZaglSkldValue()
{
    double zaglValue = 0;
    foreach (var Itm in Skld)
    {
        zaglValue += Itm.Kilkist * Itm.Cina;
    }
    return zaglValue;
}
}
```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів управління запасами

Виконав: Данило СТАРИХ
ст. групи КНТ-133м

Керівник: Тетяна КОЛПАКОВА
к.т.н., доцент

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

- **Об'єкт дослідження** – процес розробки програмного забезпечення для управління запасами.
- **Предмет дослідження** – програмні засоби для підтримки процесу управління запасами.
- **Метою роботи** є дослідження та розробка програмного забезпечення для управління запасами.

2

Рисунок В.2 – Слайд 2

Завдання роботи

- виконати аналіз предметної області та програмних засобів для управління запасами;
- розробити модель взаємодії користувача з програмним забезпеченням для управління запасами;
- здійснити проєктування програмного забезпечення для управління запасами;
- створити програмне забезпечення для управління запасами;
- виконати тестування розробленого програмного забезпечення для управління запасами.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Ordoro	inFlow Inventory	Cin7
Зручність налаштування	–	+	+–
Автоматизація закупівель	+–	+	+
Засоби управління багатоканальним продажем	+	+	+–
Аналітика та звітування	+–	+–	+
Масштабованість	+–	+–	–
Інтеграція з програмним забезпеченням провідних перевізників	+	–	+

4

Рисунок В.4 – Слайд 4

Вимоги до програмного забезпечення

- підтримка можливості управління запасами різними методами;
- підтримка можливості завдання значень параметрів (можливість дефіциту, розподіл попиту та ін.) підходу до управління запасами;
- можливість табличного подання розрахункових значень запасів товарів на складі;
- можливість динамічного керування запасами (визначення рівня запасів товарів для різних періодів);
- подання інформації про програмне забезпечення для управління запасами;
- підтримка можливості роботи з файлами даних.

5

Рисунок В.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	C#	C++	Ruby
Зручність роботи з інтерфейсом користувача	+	–	+–
Простота у використанні	+	+–	+
Висока продуктивність	+	+	–
Зручність для створення ПЗ для управління запасами	+	–	–
Масштабованість	+	+–	+

6

Рисунок В.6 – Слайд 6

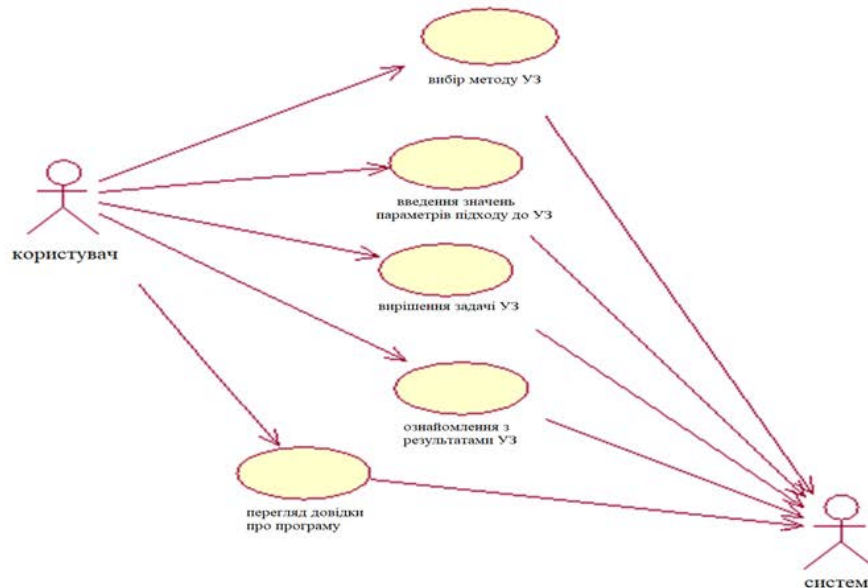
Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	Atom	Xcode
Глибока інтеграція з C# та .NET	+	—	—
Підтримка кросплатформних рішень	+	+	+—
Засоби для налагодження і тестування	+	+—	+
Інтегровані фреймворки та бібліотеки для мови C#	+	—	—
Зручність інструментів для створення інтерфейсу користувача	+	+	+—

7

Рисунок В.7 – Слайд 7

Модель взаємодії користувача з ПЗ



8

Рисунок В.8 – Слайд 8

Структура ПЗ для управління запасами



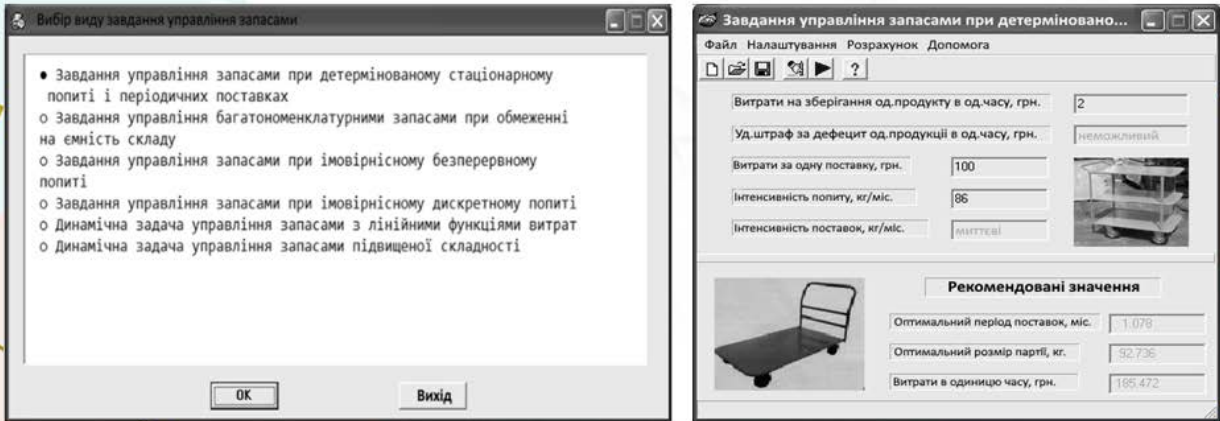
Рисунок В.9 – Слайд 9

Схема функціонування програми



Рисунок В.10 – Слайд 10

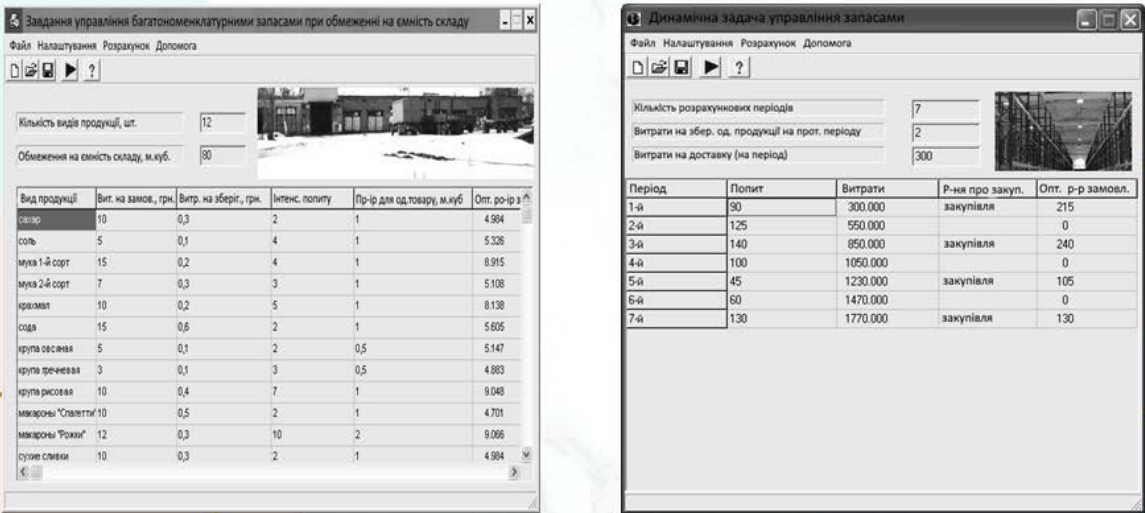
Інтерфейс програми. Головна форма. Завдання управління запасами при постійному попиті



11

Рисунок В.11 – Слайд 11

Інтерфейс програми. Завдання управління запасами різних товарів та управління динамічними запасами



12

Рисунок В.12 – Слайд 12

Інтерфейс програми. Використання генетичних алгоритмів.

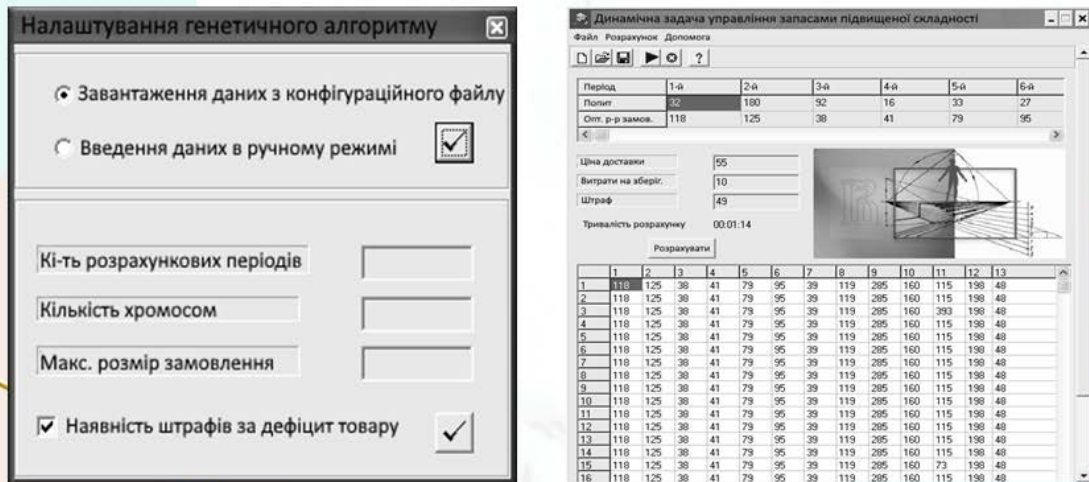


Рисунок В.13 – Слайд 13

Висновки

В ході виконання кваліфікаційної роботи було проаналізовано та досліджено процес розробки програмного забезпечення для управління запасами.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для управління запасами, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для управління запасами.

Практична цінність результатів роботи полягає у тому, що розроблено програмне забезпечення для управління запасами.

Усі завдання кваліфікаційної роботи повністю виконано.

Рисунок В.14 – Слайд 14