

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Розробка автоматизованого додатку обліку товарів на складі»

Пивовар Артем Олександрович

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ
к.т.н., доцент Гончаренко Т. А
„___” _____ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Розробка автоматизованого додатку обліку товарів на складі»

Виконав: студент 4-го курсу, групи КНс-21
Спеціальності: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційні
управляючі системи і технології»
(шифр і назва напрямку підготовки, спеціальності)
Пивовар А.О.
(прізвище та ініціали)
Керівник к.т.н., доц. Горда О.В.
(прізвище та ініціали)
Рецензент к.т.н., доц. Шабала Є. Є.
(прізвище та ініціали)

Київ 2024 р

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій
Кафедра: інформаційних технологій
Освітній рівень: «бакалавр» за ОП
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: Інформаційні управляючі системи і технології.

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ
к.т.н. Гончаренко Т.А.
„ 29 ” лютого 2024 року

**ЗАВДАННЯ
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Пивовар Артем Олександрович

Тема роботи: Розробка автоматизованого додатку обліку товарів на складі затверджена наказом ректора КНУБА № 433/2 від « 29 » лютого 2024 р.

2. Керівник роботи: к.т.н, доцент Горда Олена Володимирівна,
кафедри інформаційних технологій проектування і прикладної математики

3. Строк подання студентом роботи до захисту: _____.

4. . Зміст пояснювальної записки за розділами:

P.1. Аналіз предметної області та постановка задачі

P.2. Проектування інформаційного забезпечення

P.3. Практична реалізація

P.4. Ергономіка в ІТ

5. Інформаційні слайди:

C.1. Вступ

C.2. Дерево основних цілей

C.3. Порівняння розробок автоматизованих систем обліку товарів на складі

C.4. Порівняння баз даних

C.5. Дерево функцій

С.6. Вибір типу бази даних

С.7. Вибір СУБД

С.8. Логічна модель представлення

С.9. Фізична модель представлення

С.10. Вибір програмного інструментарію

С.11. Приклад взаємодії з базою даних

С.12. Висновки

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Лютий 2022 р.
Р. 2. Алгоритмічне та математичне забезпечення	Лютий 2024 р.
Р. 3. Розробка програмного забезпечення	Квітень 2022 р.
Р. 4. Ергономіка в ІТ	Травень 2024 р.
Направлення роботи для перевірки на плагіат	Червень 2024 р.
Остаточне оформлення роботи	Червень 2024 р.
Направлення роботи на рецензування	Червень 2024 р.
Попередній захист роботи на кафедрі	Червень 2024 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали посада консультанта, представника комісії	дата	підпис
Ергономіка інформаційних технологій	к.т.н., доц. Рябчун Ю.В.		
Прийом програмного продукту	к.т.н., доц. Шабала Є.Є.		

8. Дата видачі завдання: 29 лютого 2024 р.

Зав. кафедри

(підпис)

Гончаренко Т.А.

(прізвище та ініціали)

Керівник

(підпис)

Горда О.В.

(прізвище та ініціали)

Здобувач

(підпис)

Пивовар А.О.

(прізвище та ініціали)

АНОТАЦІЯ

Пивовар А.О. Розробка автоматизованого додатку обліку товарів на складі.

Дипломна бакалаврська робота за спеціальністю – «Комп’ютерні науки» – Київський національний університет будівництва та архітектури, Київ, 2023 рік.

Бакалаврську роботу присвячено дослідженню основних тенденцій, принципів та реалізацій обліку товарів на складі, їх інформаційного забезпечення та баз даних в цілому.

У роботі досліджено основні принципи та технології розробки інформаційного забезпечення для автоматизованих систем обліку товарів на складі. Розроблено базу даних для зберігання та обробки інформації про товари, включаючи їх надходження, зберігання та відправку.

Ключовими словами є: "інформаційне забезпечення", "складський облік", "база даних", "товари", "функціональні можливості".

ABSTRACT

Demianov N.V. Development of information support system and databases for a "smart" home.

Bachelor's thesis in Computer Science – Kyiv National University of Construction and Architecture, Kyiv, 2023.

The bachelor's thesis is devoted to the study of the main trends, principles and implementations of accounting for goods in the warehouse, their information support and databases in general.

Key words are: "information support", "warehouse accounting", "database", "goods", "functionality".

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРОБЛЕМ.....	11
1.1 Загальні та теоретичні відомості, концепції та проблематика автоматизованого обліку товарів на складі.....	11
1.2 Постановка та аналіз проблеми.....	13
1.3 Дерево основних цілей.....	15
1.4 Аналіз існуючих рішень.....	16
1.4.1 Рішення систем обліку товарів на складі.....	16
1.4.2 Рішення баз даних для обліку товарів.....	17
1.5 Постановка задачі.....	24
1.6 Проблеми взаємозв'язку баз даних з реалізаціями автоматизованого додатку обліку товарів на складі.....	26
2.ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	29
2.1. Ключові питання постановки завдання проектування інформаційного забезпечення.....	29
2.2. Дерево функцій.....	30
2.3. Вибір технології зв'язку.....	32
2.4. Вибір технології забезпечення безпеки.....	34
2.5. Вибір інтерфейсу користувача.....	36
2.5.1. Вимоги до інтерфейсу.....	36
2.5.2. Загальні можливості структури інтерфейсу.....	38
2.5.3. Вибір типу інтерфейсу користувача.....	41
2.6. Проектування моделі бази даних.....	44
2.6.1. Створення логічної моделі бази даних.....	47
2.6.2. Створення фізичної моделі бази даних.....	48

2.7. Загальний вигляд сутностей бази даних	50
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ	54
3.1. Вибір програмного інструментарію	54
3.1.1. Бази даних	54
3.1.2. Мова програмування	56
3.1.3. Програмне середовище	58
4. ЕРГОНОМІКА В ІТ	63
4.1. Вступ до ергономіки в ІТ	63
4.2. Вимоги до програмного забезпечення та підходи до його проектування з погляду користувача	64
4.3. Параметри, які необхідно враховувати при розробці інтерфейсу користувача	67
4.4. Проблеми та особливості розробки прототипу інтерфейсу	69
4.5. Вимоги до процесів інтерфейсу та проектування і реалізації компонентів ..	71
4.5.1. Проектування компонентів інтерфейсу	71
4.5.2. Реалізація компонентів інтерфейсу	71
4.6. Психологічні аспекти ергономіки	72
4.6.1. Мотивація користувачів	72
4.6.2. Зниження стресу	73
4.7. Фізичний аспект ергономіки	75
4.7.1. Запобігання професійним захворюванням	75
4.7.2. Оптимізація робочого середовища	75
ВИСНОВОК	77
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	80
Додаток 1 Програмна реалізація	83
Додаток 2 Презентація	88

ВСТУП

Актуальність

В сучасному світі все більшу популярність набуває автоматизація процесів у різних галузях, включаючи управління складськими запасами. Розробка автоматизованого додатку обліку товарів на складі є актуальною темою дослідження, оскільки це дозволяє забезпечити ефективне управління товарними запасами, зменшити людські помилки, підвищити швидкість обробки даних та знизити витрати на обслуговування складу.

Автоматизація складських процесів може значно підвищити продуктивність, забезпечити точний облік товарів, оптимізувати просторове розташування товарів на складі, а також забезпечити своєчасне поповнення запасів. Крім того, автоматизована система обліку товарів може забезпечити виявлення та запобігання втратам та крадіжкам, що підвищує безпеку та надійність зберігання товарів.

У сучасних умовах все більше компаній впроваджують автоматизовані системи обліку товарів на складі, тому розробка нових технологій та систем управління є дуже важливим напрямом досліджень у галузі логістики та складського господарства.

Мета дослідження

Покращення ефективності управління товарними запасами на складі, зменшення витрат на обслуговування складу, підвищення точності обліку та безпеки зберігання товарів.

Об'єкт дослідження

Складські системи та процеси автоматизації обліку товарів.

Предмет дослідження

Розробка ефективної та зручної системи управління товарними запасами на складі з використанням сучасних технологій та методів.

Методи дослідження

Аналіз технічних можливостей та особливостей різних систем автоматизації обліку товарів.

Вивчення популярних технологій та стандартів управління складськими системами.

Визначення потреб та вимог користувачів щодо функціональності та зручності системи управління.

Розробка та тестування програмного забезпечення для управління обліком товарів на складі.

Встановлення та налаштування обладнання для роботи системи управління.

Експериментально-теоретичне тестування та оцінка ефективності системи.

Завдання дослідження

Аналіз ринку технологій автоматизації обліку товарів на складі та визначення найбільш популярних та ефективних рішень.

Розробка програмного забезпечення для управління обліком товарів та інтеграція різних систем управління в єдину платформу.

Вивчення потреб користувачів та визначення функціональних вимог до системи управління.

Встановлення та налаштування обладнання для роботи системи управління.

Тестування та оцінка ефективності системи.

Розробка методів оптимізації управління товарними запасами та зменшення витрат на обслуговування складу.

Забезпечення безпеки користування системою управління та захисту від несанкціонованого доступу.

Наукова новизна

Розробка цілісної системи управління товарними запасами на складі на основі сучасних технологій забезпечує зручний та ефективний контроль за станом запасів і становить наукову новизну у цій галузі. Крім того, дослідження можуть включати нові підходи до оптимізації управління товарними запасами та забезпечення безпеки користування системою управління, що також представляють наукову новизну.

Практична значимість

Проаналізована система управління може допомогти значно підвищити ефективність управління товарними запасами, зменшити витрати на обслуговування складу та підвищити точність обліку товарів. Автоматизована система управління може автоматично реєструвати надходження та витрати товарів, контролювати залишки, генерувати звіти та багато іншого. Крім того, така система може забезпечити безпеку зберігання товарів, відслідковувати їх стан та запобігати втратам та крадіжкам.

Результати дослідження

Проаналізована розробка прототипу системи управління, що дозволяє контролювати облік товарів, управління запасами, генерацію звітів та інші параметри складського господарства. Крім того, дослідження включають вивчення технологічних рішень для підвищення ефективності управління товарними запасами, зменшення витрат на обслуговування складу та покращення безпеки зберігання товарів.

Результати дослідження можуть бути використані для практичної реалізації автоматизованих систем управління товарними запасами на складі та інших типах складів. Крім того, результати дослідження можуть бути використані для покращення існуючих систем управління товарними запасами, що допоможе їм стати більш ефективними та зручними у використанні.

1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРОБЛЕМ

1.1 Загальні та теоретичні відомості, концепції та проблематика автоматизованого обліку товарів на складі.

Автоматизований облік товарів на складі – це інноваційна технологія, яка надає можливість автоматизувати та контролювати різноманітні процеси управління запасами. Така система є не тільки більш зручною для ведення бізнесу, але й більш ефективною та безпечною. У зв'язку зі зростаючим попитом на цю технологію, дослідження у цій області стали особливо актуальними.[1]

Однак, використання автоматизованого обліку товарів також викликає певні проблеми та виклики. Однією з найбільш суттєвих є залежність від технологій та необхідність навчання персоналу для ефективного використання системи, хоча для кожного підприємства це може бути суб'єктивною річчю.[2]

Тому метою даної дипломної роботи є детальний аналіз інноваційної технології автоматизованого обліку товарів на складі та її інформаційного забезпечення, вивчення основних принципів її функціонування та застосування в різних сферах логістики. Також робиться спроба оцінити позитивні та негативні аспекти впровадження автоматизованого обліку та запропонувати шляхи вдосконалення даної технології.

Для досягнення поставленої мети, в роботі є намір розглянути основні аспекти автоматизованого обліку товарів, зокрема, архітектурні особливості, системи управління, типи та реалізації баз даних, комунікації та безпеки. Також проаналізовані приклади впровадження автоматизованого обліку в різних країнах світу, а також наукові статті та дослідження в цій галузі.

Для забезпечення ефективної роботи автоматизованого обліку товарів необхідно використовувати сучасні технології, такі як штучний інтелект, машинне навчання та інтернет речей. Тому у дипломній роботі було застосовано саме ці інформаційні ресурси та детально проаналізовано

технології, що лежать в основі автоматизованих систем обліку товарів, та їхні можливості. [3]

У контексті впровадження автоматизованого обліку товарів у логістику, розглянуто соціально-економічні та екологічні аспекти. Дослідження ролі автоматизованих систем у зниженні витрат та підвищенні ефективності управління запасами є важливою складовою даної теми.[4]

На основі існуючого на сьогоднішній день досвіду були запропоновані рекомендації для вдосконалення системи автоматизованого обліку товарів та покращення ефективності управління на складі. Існуючі рекомендації можуть бути корисними для розробників технологій автоматизованого обліку, інженерів, дослідників та всіх, хто цікавиться розвитком інформаційних технологій та їхнього впливу на логістику.

У цілому, дана дипломна робота є важливим кроком у розумінні та дослідженні технології автоматизованого обліку товарів, її інформаційних можливостей та ризиків. Результати можуть бути корисними для всіх зацікавлених сторін, які працюють над вдосконаленням та розвитком цієї галузі. Крім того, дана робота може бути корисною для підприємств, які планують впровадити автоматизований облік товарів у своїй діяльності, дозволяючи їм краще розуміти можливості та переваги цієї технології.

Однак, варто зазначити, що технологія автоматизованого обліку товарів ще не є повністю зрілою та має свої виклики та обмеження. Серед них можна відзначити високу вартість встановлення та підтримки систем, несумісність між різними системами, можливість хакерських атак та порушення приватності.[9]

У дипломній роботі проаналізовані ці виклики та обмеження, а також теоретично запропоновано можливі шляхи їхнього подолання.[10]

Висновки роботи є важливим етапом у розвитку технології автоматизованого обліку товарів та її впровадженні у логістичні процеси. Вони допоможуть краще зрозуміти потенціал цієї технології та сприятимуть її подальшому вдосконаленню.

Загальна мета дипломної роботи – дослідити технологію автоматизованого обліку товарів на складі, а саме її інформаційного забезпечення, проаналізувати її особливості, можливості та виклики, а також запропонувати рекомендації для вдосконалення та підвищення ефективності управління запасами. Результати дослідження можуть бути корисними для всіх зацікавлених сторін та внести свій внесок у розвиток цієї важливої технології.[11]

1.2 Постановка та аналіз проблеми.

Однією з головних проблем при впровадженні або навіть теоретичному описі автоматизованих систем обліку товарів на складі є проблема безпеки. Оскільки такі системи пов'язані з Інтернетом та мережею зв'язку, їх компоненти можуть стати об'єктом кібератак та викрадення даних. Зловмисник може отримати доступ до фізичних пристроїв, що керують обліком та управлінням товарів, змінювати їх налаштування та виконувати інші дії, що можуть призвести до втрати або пошкодження товарів, а також збоїв у логістичних процесах. Питання приватності також є важливим, оскільки бази даних обліку товарів містять конфіденційну інформацію про підприємство, клієнтів та постачальників. Недостатній захист може призвести до витоку комерційної інформації та порушення конфіденційності. Тому розробникам необхідно включати у свої системи механізми захисту та шифрування даних, а користувачам – знати основи забезпечення інформаційної безпеки.[12]

Іншою проблемою є вартість встановлення та обслуговування автоматизованих систем обліку товарів. Необхідно встановлювати спеціальні пристрої та програмне забезпечення, а також проводити їх регулярне оновлення. Це може призвести до високих витрат на встановлення та підтримку системи, що може бути проблематичним для невеликих підприємств.

Також виникає проблема відсутності єдиного стандарту для автоматизованих систем обліку товарів. Різні виробники використовують різні протоколи та стандарти, що може призвести до несумісності між різними

компонентами системи та ускладнити їх інтеграцію та обмін даними. Це може призвести до того, що деякі компоненти системи не будуть працювати належним чином або не будуть взаємодіяти між собою, що погіршить функціональність системи в цілому.[13]

Реалізація автоматизованих систем обліку товарів також потребує встановлення великої кількості пристроїв, що може створити екологічні загрози. З однієї сторони це веде до збільшення споживання електроенергії, що спричиняє додаткові витрати, а з іншої – створює додаткові джерела електромагнітного випромінювання.[14]

Нарешті, існує проблема недостатньої надійності автоматизованих систем обліку товарів. У разі збою в системі, підприємства можуть втратити контроль над своїми запасами, що може призвести до збоїв у постачаннях та інших негативних наслідків. Для запобігання таким ситуаціям, необхідно розробляти системи, які будуть працювати належним чином навіть у разі виникнення непередбачуваних ситуацій.[16]

З іншого боку, автоматизовані системи обліку товарів можуть стати джерелом нових можливостей для управління запасами та зменшення витрат. Наприклад, системи управління запасами можуть забезпечувати оптимальний рівень товарних запасів та зменшувати витрати на зберігання. Крім того, автоматизовані системи можуть інтегруватися зі смарт-мережами та іншими інфраструктурами підприємства, що дозволить забезпечити ефективне управління ресурсами та зменшення негативного впливу на довкілля.[15]

Отже, розробка інформаційного забезпечення автоматизованих систем обліку товарів на складі є складним та багатогранним завданням, яке вимагає врахування різних аспектів та проблем. Проте, правильний аналіз та розв'язання цих проблем дозволить створити ефективну, економічну та безпечну систему обліку товарів, яка відповідатиме сучасним вимогам та потребам підприємств.

1.3 Дерево основних цілей

Метою досліджень в роботі є розробка автоматизованого додатку обліку товарів на складі. Дерево цілей опису даної мети наведено на рисунку 1.1.

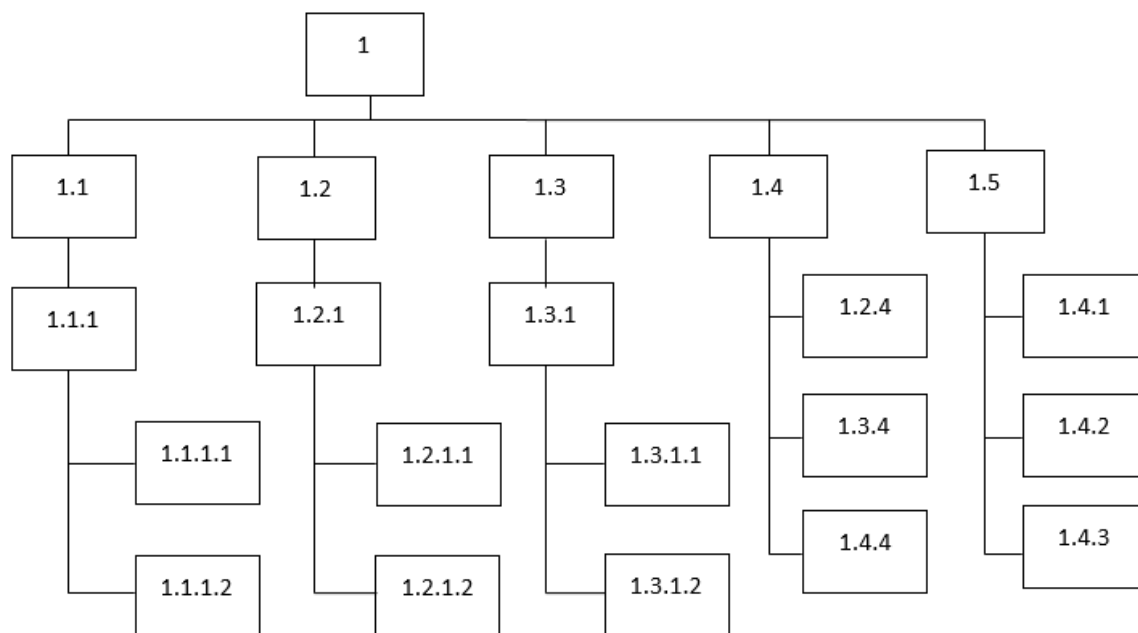


Рисунок 1.1 – Дерево цілей досліджень роботи

1. Розробка інформаційного забезпечення системи та баз даних для автоматизованого обліку товарів на складі

1.1. Розробка програмного забезпечення для контролю та управління товарними запасами

1.1.1. Розробка модулів для обліку та контролю за наявністю товарів

1.1.1.1. Розробка модулів для реєстрації надходжень та витрат товарів.

1.1.1.2. Розробка інтерфейсу для відстеження залишків товарів.

1.2. Розробка програмного забезпечення для оптимізації логістичних процесів

1.2.1. Розробка модулів для планування та оптимізації розміщення товарів

1.2.1.1. Розробка модулів для визначення оптимального розташування товарів на складі.

1.2.1.2. Розробка інтерфейсу для візуалізації складу та розміщення товарів.

1.3. Розробка мобільного додатку для управління складом

1.3.1. Розробка інтерфейсу мобільного додатку

1.3.1.1. Розробка інтерфейсу для доступу до основних функцій управління складом.

1.3.1.2. Розробка функціоналу для сканування штрих-кодів товарів.

1.4. Забезпечення безпеки та захисту даних складу

1.4.1. Розробка системи автентифікації та авторизації користувачів

1.4.2. Захист даних користувачів та інформації про товарні запаси

1.4.3. Виявлення та виправлення можливих вразливостей системи безпеки та захисту даних

1.5. Підвищення ефективності та енергоефективності складських операцій

1.5.1. Розробка алгоритмів оптимального керування складськими процесами для забезпечення енергоефективності

1.5.2. Використання датчиків та інших пристроїв для моніторингу та аналізу енергоспоживання в складських операціях

1.5.3. Визначення та підвищення ефективності роботи складських систем, зокрема, шляхом оптимізації ресурсів

1.4 Аналіз існуючих рішень.

1.4.1 Рішення систем обліку товарів на складі

На сучасному ринку існує ряд програмних рішень для автоматизованого обліку товарів на складі. Розглянемо деякі з найбільш популярних та ефективних з них:

ERP-системи (Enterprise Resource Planning): Ці комплексні програмні продукти надають широкий спектр функціоналу для управління бізнес-процесами, включаючи облік товарів на складі. Вони забезпечують цілісний

погляд на всі аспекти бізнесу, включаючи закупівлі, складське управління, виробництво та фінанси.[2]

Складські системи управління (WMS - Warehouse Management Systems): Ці системи спеціалізуються на ефективному управлінні операціями на складі. Вони дозволяють вести облік товарів, оптимізувати розміщення на складі, виконувати відбірку та відправлення замовлень, а також забезпечувати контроль за залишками товарів.[4]

1.4.2 Рішення баз даних для обліку товарів

Бази даних грають ключову роль у зберіганні та організації даних про товари на складі. Розглянемо деякі з найпопулярніших рішень на ринку:

1.MySQL: Це одна з найпопулярніших відкритих реляційних систем управління базами даних. MySQL надає широкі можливості для зберігання та обробки даних, є стабільним та ефективним рішенням для обліку товарів на складі.

2.Microsoft SQL Server: Ця реляційна система управління базами даних має широкий функціонал для роботи з даними. Вона забезпечує високу швидкість обробки даних, можливості для резервного копіювання та відновлення, що робить її привабливим варіантом для обліку товарів на складі.

Ці рішення можуть допомогти підприємствам ефективно вести облік товарів на складі, забезпечуючи точність, швидкість та надійність управління даними.

1.4.2 Рішення баз даних для обліку товарів.

Огляд систем управління базами даних

Бази даних є невід'ємною частиною багатьох програмних продуктів та інформаційних систем. На ринку існує велика кількість різних рішень баз даних, які надають користувачам можливість зберігати, організувати та оптимізувати свої дані. У даному розділі розглянуті деякі з найбільш поширених та ефективних рішень баз даних, що вже існують на ринку,

включаючи їх переваги та недоліки. Ви дізнаєтесь, які можливості надають різні системи баз даних та як вони можуть допомогти зробити роботу з даними більш ефективною та зручною.

1.Oracle Database:

Переваги:

Потужність та розширюваність, підтримка для широкого спектру додатків, включаючи бізнес-аналітику та високошвидкісні транзакційні системи. Забезпечує масштабування та високу доступність, має широкий набір інструментів для розробки та управління базами даних.

Недоліки:

Висока вартість ліцензії, складність налаштування та управління для новачків.

2.Microsoft SQL Server:

Переваги:

Популярна реляційна система управління базами даних, підтримка різних типів даних, різноманітні інструменти для аналізу та візуалізації даних. Забезпечує високу швидкість транзакцій та масштабування, висока доступність.

Недоліки: Вартість ліцензії, прив'язка до операційної системи Windows.

3.MySQL:

Переваги:

Найпопулярніша відкрита реляційна система управління базами даних. Підтримка широкого спектру додатків, включаючи веб-додатки та електронну комерцію. Забезпечує масштабування та високу доступність, широкий набір інструментів для розробки та управління базами даних.

Недоліки:

Обмежені можливості в порівнянні з комерційними рішеннями, менш потужна підтримка транзакцій.

4.MongoDB:

Переваги:

Документальна система управління базами даних, що дозволяє зберігати та опрацьовувати нереляційні дані. Підтримує масштабування та високу доступність, вбудований механізм шарування для розміщення даних на різних серверах.

Недоліки:

Менш оптимізована для складних транзакцій, обмежена підтримка складних запитів.[6]

5.PostgreSQL:

Переваги:

Відкрита реляційна система управління базами даних з розширюваною архітектурою. Підтримує багато типів даних, масштабування та високу доступність, великий набір інструментів для розробки та управління базами даних.

Недоліки:

Складніша в налаштуванні та управлінні в порівнянні з іншими відкритими системами.

6.SQLite:

Переваги:

Легка реляційна система управління базами даних, що дозволяє зберігати дані на клієнтському комп'ютері. Підтримує велику кількість типів даних, має малу вимогу до ресурсів системи, ідеальне рішення для додатків з обмеженими ресурсами.

Недоліки:

Обмежені можливості масштабування, менш оптимальна для великих систем.

7.Firebase:

Переваги:

Хмарна система управління базами даних від Google, спеціалізується на розробці мобільних та веб-додатків. Підтримує реальний час для збереження та

синхронізації даних, має функції аутентифікації користувачів, зберігання файлів та аналітики.

Недоліки:

Обмежена підтримка складних запитів, залежність від хмарного сервісу.

8.InfluxDB:

Переваги:

Нереляційна система управління базами даних, спеціалізується на зберіганні та аналізі даних датчиків та IoT-пристроїв. Має вбудовані механізми масштабування та високої доступності, функції запитів за часовими мітками, статистичні аналізи та візуалізацію даних.

Недоліки:

Обмежені можливості для традиційних транзакцій, потребує специфічних знань для оптимального використання.[5]

9.MS Access:

Переваги:

Легка у використанні система управління базами даних, інтегрована з Microsoft Office. Забезпечує зручні інструменти для створення та управління базами даних, підтримка невеликих бізнес-додатків, швидке створення прототипів.

Недоліки:

Обмежені можливості масштабування, не підходить для великих систем з високими вимогами до продуктивності, менша кількість інструментів для аналітики в порівнянні з іншими системами.[9]

Таблиця 1.2 – опис рішень баз даних для написання під системи IoT.

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
MongoDB	NoSQL	MongoDB	Windows, Linux, macOS	Гнучкість у роботі зі структурованими та неструктурованими

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
				даними, горизонтальне масштабування
MySQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, гарна підтримка індексів
PostgreSQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, можливість використовувати різні типи даних, розширені можливості з безпеки
InfluxDB	Time Series	InfluxQL	Windows, Linux, macOS	Оптимізована для роботи з часовими рядами, гарна підтримка агрегаційних функцій

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
Cassandra	NoSQL	CQL	Windows, Linux, macOS	Гнучкість у роботі зі структурованими та неструктурованими даними, можливість масштабування на кілька вузлів, відмінна мінливість, підтримка транзакцій на рівні запису
Firebase	Real-time	Firebase API	Windows, Linux, macOS	Інтегрований механізм синхронізації, можливість прямого зв'язку з мобільними додатками
Oracle Database	Relational	SQL	Windows, Linux, macOS	Надійна платформа, підтримка великих обсягів даних, можливість використовувати більшість функцій SQL, відмінна масштабованість

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
SQLite	Relational	SQL	Windows, Linux, macOS	Невеликий розмір, легкість використання та налаштування, відсутність необхідності в окремому сервері для роботи з базою даних, підтримка транзакцій, швидкий доступ до даних
Microsoft SQL Server	Relational	T-SQL	Windows, Linux, macOS	Гарна підтримка індексів та транзакцій, можливість використовувати більшість функцій SQL, відмінна масштабованість
MS Access	Relational	SQL	Windows	Легкість використання для малих та середніх проектів, інтеграція з Microsoft Office, не потребує окремого сервера, зручний

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
				інтерфейс для користувачів, обмежена масштабованість

MS Access надає зручний та інтуїтивний інтерфейс для створення та управління базами даних, що робить його підходящим для невеликих проектів, де не потрібна висока масштабованість та продуктивність.

1.5 Постановка задачі.

Метою дипломного проекту є створення автоматизованого додатку обліку товарів на складі, який дозволить оптимізувати процеси управління товарними запасами, підвищити точність обліку та ефективність роботи складу. Реалізація проекту передбачає розробку бази даних, інтерфейсу для користувача, алгоритмів автоматизації процесів обліку, системи контролю доступу до даних, а також проведення тестування та налагодження системи. Очікуваними результатами є створення робочої інформаційної системи зручного та зрозумілого інтерфейсу для користувача, а також функціональності для автоматизації обліку товарів на складі та забезпечення контролю доступу до даних.

Основні завдання дипломного проекту:

1. Розробка бази даних для зберігання інформації про товари, що зберігаються на складі.

Створення структури бази даних, яка забезпечує зберігання, оновлення та зчитування даних про товари.

Визначення необхідних атрибутів товарів та їх характеристик.

2. Розробка інтерфейсу для управління обліком товарів на складі.

Створення зручного та зрозумілого інтерфейсу для користувача.

Забезпечення функцій для введення, редагування, перегляду та видалення інформації про товари.

3. Розробка алгоритмів для автоматизації процесів обліку товарів.

Розробка алгоритмів для прийому, зберігання, переміщення та відпуску товарів.

Інтеграція алгоритмів з базою даних для автоматичного оновлення інформації.

4. Розробка системи контролю доступу до даних.

Визначення рівнів доступу для різних категорій користувачів.

Реалізація механізмів автентифікації та авторизації користувачів.

5. Тестування та налагодження системи на реальних даних та в реальних умовах використання.

Проведення тестування функціональності системи.

Виявлення та виправлення помилок та недоліків.

Аналіз результатів тестування та внесення необхідних коригувань.

Очікувані результати дипломного проекту:

1. Розроблена база даних та інформаційна система для обліку товарів на складі.

Функціонуюча база даних, яка забезпечує зберігання та обробку інформації про товари.

2. Розроблений зручний та зрозумілий інтерфейс для користувача.

Інтерфейс, який дозволяє легко та ефективно управляти даними про товари.

3. Розроблені та впроваджені алгоритми для автоматизації процесів обліку товарів на складі.

Алгоритми, які забезпечують автоматизацію процесів прийому, зберігання, переміщення та відпуску товарів.

4. Розроблена система контролю доступу до даних.

Реалізовані механізми автентифікації та авторизації користувачів, що забезпечують захист даних.

5. Проведене тестування та налагодження системи.

Система, що пройшла тестування та налаштування, готова до впровадження в реальних умовах.

1.6 Проблеми взаємозв'язку баз даних з реалізаціями автоматизованого додатку обліку товарів на складі

Автоматизовані додатки обліку товарів на складі можуть включати різні функції, такі як моніторинг запасів, обробка даних про переміщення товарів, інтеграція з системами постачальників та інших підрозділів компанії. Ці дані можуть зберігатись у базах даних, які повинні взаємодіяти з іншими системами управління складом. Проте, можуть виникати проблеми з взаємозв'язком баз даних з реалізаціями автоматизованих додатків обліку товарів на складі. Наприклад, можуть виникати проблеми зі сумісністю різних пристроїв та протоколів, які використовуються для передачі даних між системами обліку та базами даних. Також можуть виникати проблеми з недостатнім захистом даних, що може призвести до втрати інформації та зловживань даними.[3]

Іншою проблемою може бути складність інтеграції баз даних з іншими системами управління складом, особливо якщо ці системи розроблені різними виробниками та використовують різні протоколи та стандарти. Для уникнення цих проблем, необхідно ретельно планувати та досліджувати взаємозв'язок між базами даних та системами обліку перед їх впровадженням. Важливо враховувати вимоги щодо захисту даних та використання стандартів та протоколів, які забезпечують сумісність та інтеграцію з іншими системами управління складом.[2]

З впровадженням автоматизованих додатків обліку товарів на складі може з'явитися необхідність зберігати та обробляти велику кількість різноманітних даних, які можуть включати інформацію про стан товарів, їх кількість,

місцезнаходження, терміни зберігання та інші параметри. Ці дані збираються та передаються до центральної системи для обробки та аналізу. Для забезпечення роботи таких систем необхідні бази даних, які можуть зберігати, оновлювати та зчитувати дані з різних джерел.[7]

Проте, при взаємозв'язку баз даних з автоматизованими додатками обліку товарів на складі можуть виникати різні проблеми. Однією з таких проблем є сумісність між різними протоколами та стандартами, які використовуються для передачі даних між пристроями та базами даних. Наприклад, якщо одна система використовує один протокол для передачі даних, а інша — інший, можуть виникати складнощі з передачею даних між ними.[4]

Також можуть виникати проблеми зі стандартами забезпечення безпеки та конфіденційності даних, що може призвести до можливого зловживання даними. Іншою проблемою може бути складність інтеграції баз даних з іншими системами управління складом, особливо якщо ці системи розроблені різними виробниками та використовують різні протоколи та стандарти. Це може призвести до складнощів зі зберіганням та передачею даних між різними системами, що може вплинути на роботу всієї системи обліку.

Для забезпечення ефективної та безпечної роботи автоматизованих додатків обліку товарів на складі необхідно ретельно планувати та досліджувати потенційні проблеми, які можуть виникнути при взаємозв'язку баз даних з системами обліку. Іншою проблемою може бути обмежена масштабованість баз даних, особливо якщо обсяг даних постійно збільшується. Чим більше даних, тим більше ресурсів необхідно для їх зберігання та обробки. Якщо база даних не може масштабуватися, це може призвести до проблем з продуктивністю та надійністю всієї системи.

Якщо дані зберігаються в різних форматах або з різними структурами, це може ускладнити їх обробку та аналіз, що може призвести до зниження ефективності та точності роботи системи обліку товарів на складі. Крім того, при роботі з базами даних можуть виникати проблеми з безпекою та конфіденційністю даних. Недостатня захищеність баз даних може призвести до

викрадення та витоку даних, що може мати серйозні наслідки для безпеки та фінансових показників компанії.

Отже, для успішної реалізації автоматизованих додатків обліку товарів на складі необхідно забезпечити ефективний взаємозв'язок між базами даних та іншими системами, забезпечити стандартизацію даних та їх форматів, розробити ефективні засоби захисту та безпеки даних та забезпечити масштабованість баз даних для забезпечення продуктивності та надійності системи обліку.[4]

2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Ключові питання постановки завдання проєктування інформаційного забезпечення

Для розробки інформаційного забезпечення автоматизованого додатку обліку товарів на складі необхідно розглянути декілька ключових питань, таких як:

1. Визначення дерева функцій та можливостей системи обліку товарів на складі:

Це допоможе зрозуміти, які бази даних і яке програмне забезпечення необхідно розробити. Основні функції можуть включати моніторинг запасів, обробку даних про прийом та видачу товарів, відстеження переміщень товарів у межах складу та генерацію звітів.

2. Вибір технології зв'язку:

Для автоматизованої системи обліку товарів на складі можуть використовуватися різні технології зв'язку, такі як Wi-Fi, Bluetooth, RFID тощо. Необхідно визначитись з технологією, що буде використовуватись, і забезпечити підтримку цієї технології в базі даних та програмному забезпеченні.

3. Забезпечення безпеки:

Необхідно розглянути питання забезпечення безпеки даних та зв'язку в системі обліку товарів. Для цього можна використовувати різні технології, такі як шифрування даних, автентифікація користувача, контроль доступу та резервне копіювання даних.

4. Розробка інтерфейсу користувача:

Для забезпечення зручності користування системою обліку товарів необхідно розробити інтерфейс користувача. Функціональність інтерфейсу може включати інструменти для внесення даних про товари, перегляду наявних запасів, створення звітів та налаштування системи.

5. Проектування моделі бази даних (концептуальна, фізична та логічна представлення):

Для правильної реалізації інформаційного забезпечення та баз даних необхідно визначити правильну концептуальну, логічну та фізичну форми представлення бази даних:

Концептуальна модель описує відносини між сутностями в системі та їх атрибутами. Це допомагає визначити, які сутності повинні бути включені до бази даних та як вони пов'язані між собою. Концептуальна модель допомагає уникнути зайвих даних.

Логічна модель описує відносини між таблицями та структуру даних, які використовуються для зберігання. Вона дозволяє визначити, які таблиці повинні бути створені та які взаємозв'язки повинні бути встановлені між ними.

Фізична модель описує, як дані фактично зберігаються на диску або в інших засобах зберігання. Вона визначає, які типи даних використовуються для зберігання, які індекси та ключі повинні бути встановлені та як дані можна розмістити на диску для оптимальної продуктивності.

6. Тестування та налагодження:

Після розробки програмного забезпечення та баз даних необхідно провести тестування та налагодження системи. Це допоможе виявити та виправити помилки та недоліки системи перед її використанням. Тестування повинне включати перевірку функціональності, продуктивності, безпеки та сумісності системи.

Ці питання є ключовими для успішного проектування та реалізації автоматизованого додатку обліку товарів на складі, забезпечення його продуктивності, безпеки та зручності користування.

2.2 Дерево функцій

Дерево функцій – це інструмент для опису ієрархічної структури функцій, які повинні бути виконані для досягнення певної мети. У контексті розробки інформаційного забезпечення та баз даних для автоматизованого додатку

обліку товарів на складі, дерево функцій використовується для опису функціональних блоків системи, які повинні бути реалізовані, щоб забезпечити користувачам зручне та ефективне управління запасами. В науково-дослідницькій роботі дерево функцій слугує для відображення можливих функцій і не потребує виконання всіх аспектів в повній мірі.

Головна мета роботи представлена вершиною дерева функцій, яка включає гілки, що представляють різні функції або завдання. Ці завдання пов'язані з досягненням головної мети та інших підзавдань. Функціональна декомпозиція системи базується на її функціональних частинах, кожен елемент яких розглядається окремо з урахуванням його ролі в системі.

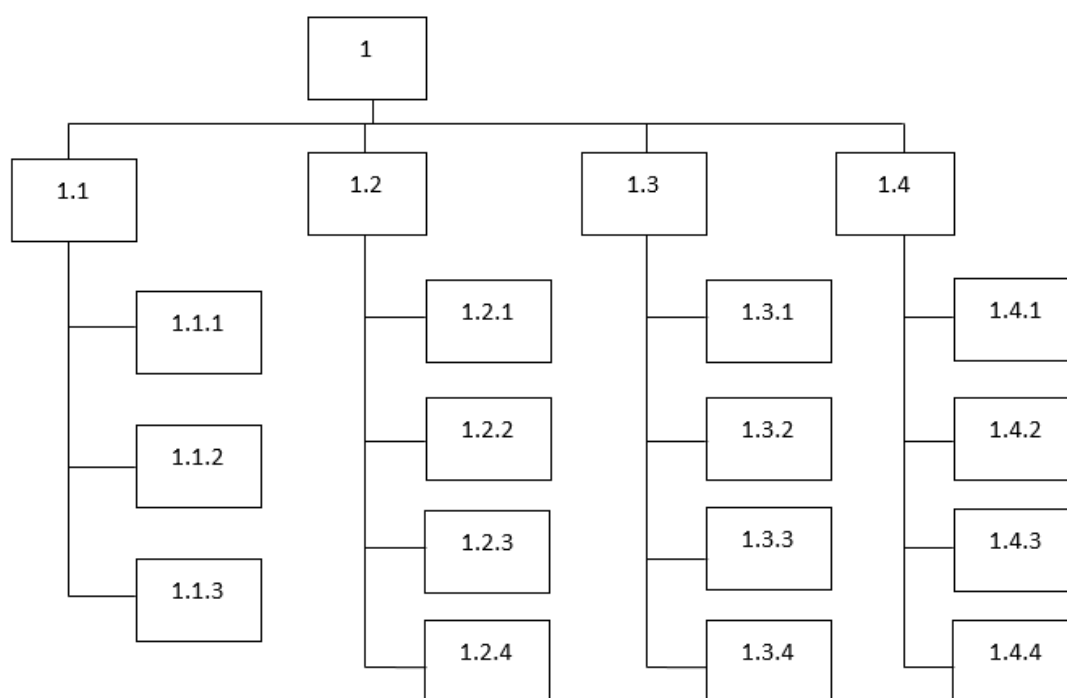


Рисунок 2.1 – Дерево функцій розроблюваного проєкту.

1. Створення автоматизованої системи обліку товарів на складі

1.1. Функції збору інформації про товари

1.1.1. Зчитування даних про прийом товарів

1.1.2. Зчитування даних про видачу товарів

1.1.3. Зчитування даних про переміщення товарів всередині складу

1.2. Функції керування запасами

- 1.2.1. Управління прийомом товарів
- 1.2.2. Управління видачею товарів
- 1.2.3. Управління переміщенням товарів
- 1.2.4. Управління інвентаризацією

1.3. Функції моніторингу системи

- 1.3.1. Аналіз даних про запаси для підтримки оптимального рівня
- 1.3.2. Моніторинг витрат товарів
- 1.3.3. Моніторинг рівня запасів для уникнення надлишків та дефіцитів
- 1.3.4. Оповіщення користувачів про критичні рівні запасів та інші проблеми

1.4. Функції збереження та аналізу даних

- 1.4.1. Зберігання даних про товари у базі даних
- 1.4.2. Аналіз даних для забезпечення максимальної ефективності роботи системи
- 1.4.3. Підготовка звітів та статистики щодо використання товарів
- 1.4.4. Забезпечення можливості редагування та налаштування системи

2.3. Вибір технології зв'язку

Працездатність системи автоматизованого додатку обліку товарів на складі потребує вибору технології зв'язку, яка дозволить підключати до неї різні пристрої та сенсори, а також забезпечить безпеку даних, переданих між ними та сервером. Найбільш популярними технологіями зв'язку для таких систем є Wi-Fi, Bluetooth, ZigBee та Z-Wave. Розглянемо їх характеристики у порівнянні:

Таблиця 2.1 – Порівняльна характеристика технологій зв'язку

Технологія	Частота	Швидкість передачі даних	Дальність передачі даних	Безпека
Wi-Fi	2.4GHz/5GHz	до 300Mbps/600Mbps	до 100м	WPA2-PSK
Bluetooth	2.4GHz	до 24Mbps	до 100м	AES-128
ZigBee	2.4GHz	до 250kbps	до 75м	AES-128
Z-Wave	868.4MHz/908.4 MHz	до 100kbps	до 100м	AES-128

Аналіз технологій зв'язку

1. Wi-Fi:

Переваги:

Висока швидкість передачі даних, широка доступність, простота налаштування.

Недоліки:

Високе енергоспоживання, залежність від наявності стабільного Wi-Fi з'єднання, можливі проблеми з безпекою в громадських мережах.

2. Bluetooth:

Переваги:

Низьке енергоспоживання, висока швидкість передачі даних, проста інтеграція з мобільними пристроями.

Недоліки:

Обмежена дальність передачі даних, можливі проблеми з сумісністю між різними версіями Bluetooth.

3. ZigBee:

Переваги:

Низьке енергоспоживання, стійкість до перешкод, висока надійність, підтримка великої кількості підключених пристроїв.

Недоліки:

Низька швидкість передачі даних, обмежена дальність передачі.

4. Z–Wave:

Переваги:

Низьке енергоспоживання, висока надійність, стійкість до перешкод, можливість створення масштабованих мереж.

Недоліки:

Низька швидкість передачі даних, обмежена кількість підключених пристроїв.

Рекомендації

Найбільш популярні технології – Wi–Fi та Bluetooth – мають досить високу швидкість передачі даних, але можуть бути не досить надійними та безпечними в деяких умовах. ZigBee та Z–Wave є більш стійкими до перешкод та мають надійну безпеку, проте швидкість передачі даних є меншою.

Кожна з технологій має свої переваги та недоліки. На основі порівняння у таблиці 2.1 та враховуючи специфіку даної роботи, рекомендується використання ZigBee або Z–Wave технологій зв'язку. Ці технології є більш стійкими до перешкод та мають надійну безпеку, що є важливим для системи обліку товарів на складі, де забезпечення конфіденційності та цілісності даних є критично важливою. Також, ZigBee та Z–Wave мають достатню швидкість передачі даних для потреб обліку товарів, що дозволяє забезпечити зручність використання системи користувачем.

Однак, для дослідницької роботи обрано використовувати технології Wi–Fi та Bluetooth. Це рішення обґрунтовано їхньою широкою доступністю, простотою налаштування та можливістю легкої інтеграції з існуючими мобільними пристроями та мережами.

2.4. Вибір технології забезпечення безпеки

Для забезпечення безпеки інформації в системі автоматизованого додатку обліку товарів на складі можна використовувати різні технології, такі як:

SSL / TLS: Ці протоколи забезпечують безпеку передачі даних, шифруючи їх і запобігаючи доступу до них з боку зловмисників.

VPN: Це безпечний тунель між двома пристроями, що забезпечує захист передачі даних з одного кінця до іншого.

Firewalls: Це пристрої, які контролюють доступ до системи і забезпечують захист від несанкціонованого доступу.

Антивірусні програми: Ці програми виявляють інфіковані файли і захищають систему від шкідливих програм.

Ідентифікація та автентифікація: Ці технології забезпечують захист від несанкціонованого доступу.

Таблиця 2.2 – Порівняльна характеристика технологій безпеки

Технологія	Переваги	Недоліки
SSL / TLS	Ефективний захист від перехоплення даних	Потребує додаткових ресурсів для шифрування даних
VPN	Забезпечує безпечну передачу даних	Потребує додаткових налаштувань
Firewalls	Ефективний захист від несанкціонованого доступу	Може обмежувати легітимний доступ до системи
Антивірусні програми	Захист від шкідливих програм	Може сповільнювати роботу системи
Ідентифікація та автентифікація	Забезпечує захист від несанкціонованого доступу	Можливість зламування захисту

Рекомендації

Найкращою технологією забезпечення безпеки буде використання SSL / TLS. Ця технологія є ефективною у захисті від перехоплення даних та не вимагає значних ресурсів для їх шифрування. Застосування інших технологій,

таких як VPN, Firewalls, Антивірусні програми та ідентифікація та автентифікація, може бути виправданим залежно від конкретних потреб проекту. Проте в даному випадку SSL / TLS є оптимальним вибором з точки зору ефективності та витрат на реалізацію.

Додаткові заходи безпеки

Окрім використання SSL / TLS, для забезпечення безпеки інформації в системі автоматизованого обліку товарів на складі доцільно впровадити наступні заходи:

Регулярне оновлення програмного забезпечення:

Забезпечує захист від нових загроз та вразливостей.

Резервне копіювання даних:

Регулярне створення резервних копій даних допоможе відновити систему у разі втрати даних.

Моніторинг та аудит:

Постійний моніторинг активності системи та аудит дій користувачів допоможе виявити та попередити потенційні загрози.

Навчання користувачів:

Проведення тренінгів та навчання користувачів з питань безпеки інформації зменшить ризик ненавмисних помилок та дій, що можуть призвести до втрати даних або компрометації системи.

Рольове управління доступом:

Забезпечення доступу до системи та даних тільки тим користувачам, яким це необхідно для виконання службових обов'язків, допоможе мінімізувати ризик несанкціонованого доступу.

Впровадження цих заходів допоможе створити надійну та безпечну систему автоматизованого обліку товарів на складі, яка буде ефективно захищена від внутрішніх та зовнішніх загроз.

2.5. Вибір інтерфейсу користувача

2.5.1. Вимоги до інтерфейсу

Для розробки інформаційного забезпечення та баз даних автоматизованого додатку обліку товарів на складі необхідний інтерфейс, який би дозволяв легко та зручно взаємодіяти з даними та змінювати їх. Основні вимоги до інтерфейсу створення інформаційного забезпечення та баз даних мають бути такі:

Легкість використання:

Інтерфейс повинен бути простим та зрозумілим для будь-якого користувача без великого досвіду роботи з базами даних та програмним забезпеченням.

Функціональність:

Інтерфейс повинен забезпечувати можливість додавання, видалення та редагування даних у базах даних, а також взаємодію з іншими системами, такими як мобільні додатки та інтернет-сервіси.

Зручність відображення даних:

Інтерфейс повинен мати зручну та логічну структуру відображення даних, що дозволить користувачеві легко орієнтуватися та знайти потрібну інформацію.

Надійність та безпека:

Інтерфейс повинен мати захист від несанкціонованого доступу до даних та забезпечувати надійне збереження даних.

Можливість масштабування:

Інтерфейс повинен дозволяти легко розширювати бази даних та додавати нові функціональні можливості без перезапуску системи.

Підтримка стандартів:

Інтерфейс повинен дотримуватися стандартів програмування та баз даних, що дозволить його легко інтегрувати з іншими системами та розробляти додаткові функції.

Підтримка мови SQL (Structured Query Language):

Інтерфейс повинен підтримувати мову структурованого запиту до баз даних SQL, що є стандартом в області управління реляційними базами даних.

Відстеження та логування дій користувачів:

Інтерфейс повинен мати можливість відслідковувати та логувати дії користувачів. Це дозволить забезпечити контроль за даними та зберіганням історії змін, що може бути корисним у випадку виявлення помилок або потреби відновити попередні налаштування.

Автоматичне оновлення та поновлення баз даних:

Інтерфейс повинен дозволяти автоматичне оновлення та поновлення баз даних, що дозволяє уникнути можливих проблем зі сумісністю між версіями та забезпечити надійну та стабільну роботу системи.

Імпорт та експорт даних:

Інтерфейс повинен забезпечувати можливість імпорту та експорту даних з інших джерел. Це дозволяє користувачам ефективно обмінюватись даними з іншими системами та розробляти різноманітні інтеграції.

Загалом, інтерфейс створення інформаційного забезпечення та баз даних має бути зрозумілим, зручним та надійним, надаючи додаткові функції для користувачів.

2.5.2 Загальні можливості структури інтерфейсу

Для забезпечення зручного та інтуїтивно зрозумілого інтерфейсу розробки інформаційного забезпечення та баз даних для автоматизованого додатку обліку товарів на складі, його структура може включати наступні компоненти:

1. Панель навігації:

Це важливий елемент інтерфейсу, що дозволяє користувачам легко переміщатись між різними розділами програми та виконувати необхідні дії. Панель навігації може містити посилання на такі розділи, як "Домашня сторінка", "Керування товаром", "Управління користувачами", "Системні налаштування" тощо.

2. Розділ "Керування товаром":

Цей розділ може містити інформацію про всі товари, що зберігаються на складі, та дозволяти користувачам керувати ними. Тут можуть бути доступні такі функції, як додавання нових товарів, оновлення інформації, перегляд історії переміщень, створення та редагування інвентаризацій тощо.

3. Розділ "Управління користувачами":

Цей розділ може містити інформацію про всіх користувачів системи та їх рівень доступу до функцій та даних. Тут можуть бути доступні такі функції, як створення нових користувачів, надання та зміна рівнів доступу, налаштування паролів та інші аспекти управління користувачами.

4. Розділ "Системні налаштування":

Цей розділ може містити інформацію про налаштування системи та баз даних, а також дозволяти користувачам змінювати різні параметри. Тут можуть бути доступні такі функції, як налаштування резервного копіювання, налаштування безпеки, налаштування зв'язку з іншими системами тощо.

5. Розділ "Домашня сторінка":

Цей розділ може містити основну інформацію про систему та найважливіші функції, які можуть бути виконані з використанням інтерфейсу. Наприклад, на цій сторінці можуть бути доступні швидкі посилання на найбільш використовувані функції, а також повідомлення про стан товарів або інформація про сповіщення та нагадування.

6. Розділ "Створення та налаштування баз даних":

Цей розділ може містити інструменти для створення та редагування баз даних, налаштування зв'язків між ними, додавання нових таблиць та полів, встановлення індексів, забезпечення безпеки та інші налаштування.

7. Розділ "Пошук та фільтрація даних":

Цей розділ може містити інструменти для пошуку та фільтрації даних, що містяться в базі даних. Користувачі можуть шукати та фільтрувати дані за певними параметрами, такими як дата, тип товару, постачальник тощо.

8. Розділ "Візуалізація даних":

Цей розділ може містити інструменти для візуалізації даних, зібраних системою обліку товарів на складі. Користувачі можуть створювати графіки, діаграми та інші візуальні елементи, щоб краще розуміти дані та виявляти тенденції.

9. Розділ "Налаштування сповіщень":

Цей розділ може містити інструменти для налаштування сповіщень, які надходять з системи обліку товарів. Користувачі можуть встановлювати правила та налаштовувати рівні сповіщень для різних подій, щоб бути в курсі стану системи.

10. Розділ "Аналітика та статистика":

Цей розділ може містити інструменти для аналізу та відображення статистики використання системи обліку товарів на складі. Користувачі можуть переглядати статистику руху товарів, аналізувати тенденції та виявляти незвичайні показники.

11. Розділ "Журнал подій":

Цей розділ може містити інформацію про всі події, які відбуваються в системі обліку товарів на складі, такі як зміни стану товарів, нагадування про розклад робіт, сповіщення про непередбачені ситуації тощо. Журнал подій може допомогти користувачам відслідковувати та аналізувати події в системі.

12. Розділ "Налаштування профілю користувача":

Цей розділ може містити інформацію про профіль користувача, таку як ім'я, контактну інформацію, налаштування безпеки та приватності. Користувачі можуть змінювати свої налаштування, щоб забезпечити безпеку та приватність своїх даних.

13. Розділ "Довідка та підтримка":

Цей розділ може містити інформацію про довідку та підтримку для користувачів. Користувачі можуть отримувати допомогу та відповіді на свої запитання, переглядати часті запитання та відповіді, а також звертатися до служби підтримки.

Ці компоненти інтерфейсу допоможуть створити зручну та ефективну систему для управління обліком товарів на складі, що дозволить користувачам легко взаємодіяти з даними та керувати системою.

Графічне представлення вимог зображено на рисунку 2.2

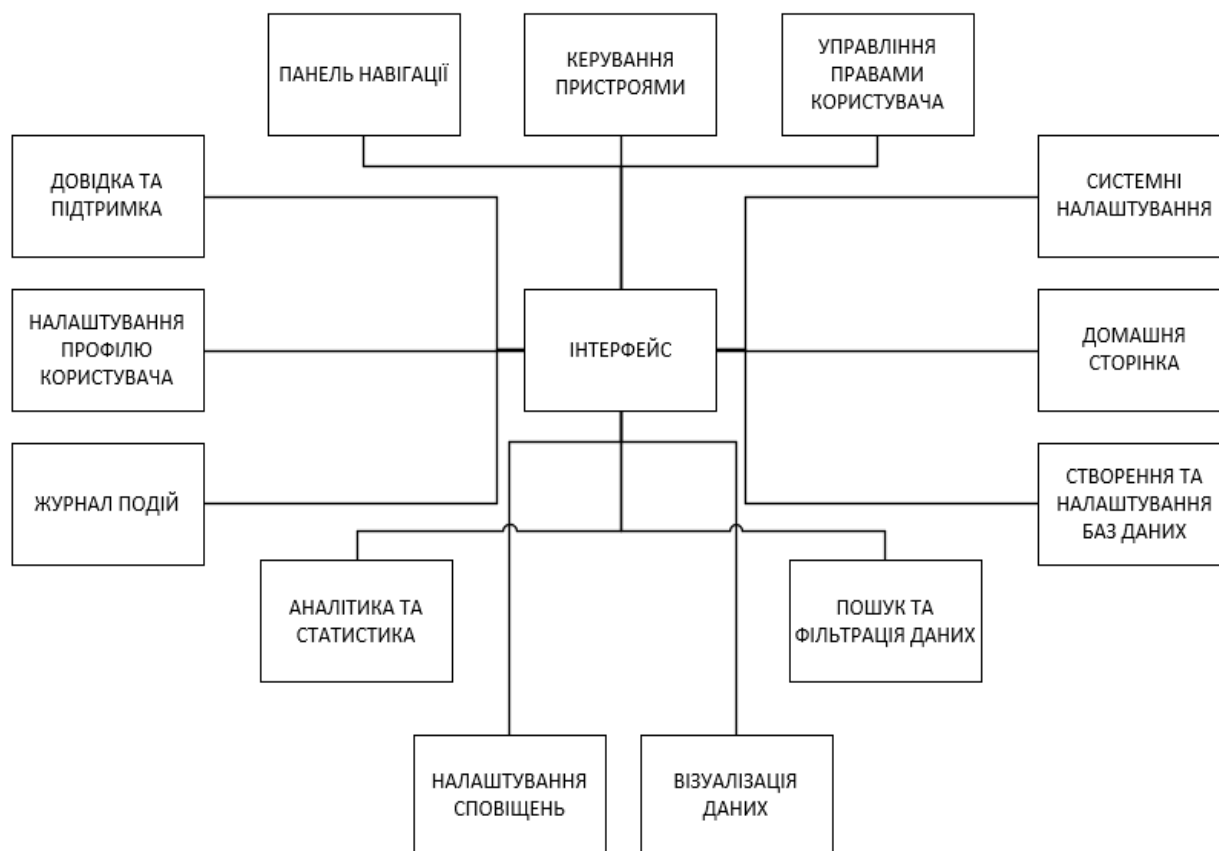


Рисунок 2.2 – графічне представлення загальних можливостей структури інтерфейсу

2.5.3 Вибір типу інтерфейсу користувача

Дана підтема досліджує різні типи інтерфейсу, такі як текстовий, графічний, віртуальна реальність та інші. Для кожного типу інтерфейсу розглядаються його переваги та недоліки, включаючи легкість використання, швидкість виконання завдань, підтримку користувачів з обмеженими можливостями та ін. Опис «простих» типів представлений в таблиці нижче.

Таблиця 2.3 – Порівняльна характеристика типів інтерфейсів

Тип інтерфейсу	Опис	Переваги	Недоліки
Графічний	Інтерфейс з використанням графічних елементів	Легкий у використанні	Потребує певних знань, обмежує функціонал
Текстовий	Інтерфейс з використанням текстових команд	Дозволяє доступ до всіх функцій, легкий у використанні	Складний у вивченні, не так зручний у використанні
Голосовий	Інтерфейс на основі розпізнавання мови	Забезпечує зручність, легкий у використанні	Потребує використання спеціального обладнання
Сенсорний	Інтерфейс на основі сенсорних елементів	Забезпечує зручність, не вимагає додаткових знань	Дорогий, складний у використанні
VR інтерфейс	Інтерфейс, заснований на використанні VR	Забезпечує максимальну іммерсивність, зручність	Дорогий

Опис «складних» варіантів, виведених із таблиці вище:

1. Веб–інтерфейс:

Опис:

Інтерфейс користувача, який доступний через веб–браузер.

Переваги:

Простий веб–інтерфейс, що дає можливість користувачеві взаємодіяти з домашніми пристроями з будь–якого місця, де є доступ до Інтернету. Не потребує спеціального програмного забезпечення, оскільки працює на будь–якому браузері.

Недоліки:

Може мати обмежену функціональність у порівнянні з настільними додатками.

2. Мобільний додаток:

Опис:

Додаток, який можна встановити на смартфон або планшет.

Переваги:

Мобільний додаток може мати більш простий та зручний інтерфейс, спеціально адаптований для маленьких екранів. Крім того, мобільний додаток може використовувати додаткові можливості смартфона, такі як GPS або датчики руху.

Недоліки:

Потребує розробки для різних мобільних платформ (iOS, Android).

3. Голосовий інтерфейс:

Опис:

Інтерфейс, що дає можливість користувачеві керувати домашніми пристроями за допомогою голосових команд.

Переваги:

Голосовий інтерфейс може бути зручним для людей, які мають обмежені можливості взаємодії з домашніми пристроями через інші інтерфейси. Забезпечує високу зручність використання.

Недоліки:

Потребує використання спеціального обладнання для розпізнавання мови. Можливі проблеми з розпізнаванням команд через акценти або фонового шуму. Отже, з огляду на дослідницьку мету проєкту, оптимальним вибором став графічний інтерфейс. Однак, з урахуванням складності в деяких аспектах,

робота продемонструє використання різних інтерфейсів. Вони забезпечать простоту використання та масштабованість, знизять вартість розробки та підтримки. Також, дані інтерфейси забезпечать широку доступність для користувачів, оскільки можуть бути реалізованими на будь-якому пристрої з доступом до Інтернету.

Вибір різних типів інтерфейсів дозволить покрити різні потреби користувачів:

Графічний інтерфейс: для зручного доступу та використання можливостей з візуалізацією інформації через графічні елементи.

Веб-інтерфейс: для доступу до системи з будь-якого пристрою та місця.

Мобільний додаток: для зручного доступу та використання можливостей смартфона.

Голосовий інтерфейс: для забезпечення зручності використання для користувачів з обмеженими можливостями.

Це забезпечить простоту використання, масштабованість та зниження вартості розробки та підтримки системи автоматизованого обліку товарів на складі.

2.6. Проектування моделі бази даних

Проектування моделі бази даних

База даних автоматизованого додатку обліку товарів на складі повинна містити три рівні представлення: концептуальний, фізичний та логічний.

Концептуальне представлення бази даних[7]

Концептуальне представлення бази даних – це високорівневе описання структури та залежностей між даними в предметній області обліку товарів на складі. На цьому рівні визначаються основні сутності та їх взаємозв'язки. Наприклад, можна визначити такі таблиці:

Товари: містить інформацію про всі товари, що зберігаються на складі (ID, назва, опис, категорія, ціна, кількість, дата надходження).

Категорії: містить інформацію про категорії товарів (ID, назва, опис).

Постачальники: містить інформацію про постачальників (ID, назва, контактна інформація).

Рух товарів: містить інформацію про переміщення товарів (ID, ID товару, тип операції, кількість, дата, примітки).

Користувачі: містить інформацію про користувачів системи (ID, ім'я, роль, логін, пароль).

Ролі: містить інформацію про ролі користувачів (ID, назва, опис).

Концептуальне представлення допомагає зрозуміти взаємозв'язки між різними об'єктами в системі та визначити ключові аспекти бази даних.[8]

Фізичне представлення бази даних

Фізичне представлення бази даних – це опис технічних аспектів зберігання даних, таких як формати файлів, типи дискових пристроїв, що використовуються для зберігання інформації про товари на складі. Фізичне представлення включає в себе детальні відомості про розташування та організацію даних на дисках, а також процеси забезпечення безпеки і зменшення ризиків втрати даних.

Формати файлів: SQL файли для створення та наповнення таблиць.

Типи дискових пристроїв: SSD для швидкого доступу до даних.

Резервне копіювання: автоматичне створення резервних копій даних на зовнішніх носіях або у хмарі.

Безпека: використання шифрування даних та контроль доступу до бази даних.[4]

Логічне представлення бази даних

Логічне представлення бази даних – це опис способів, якими дані можуть бути доступні та опрацьовуватися за допомогою запитів. Логічне представлення включає в себе опис схем бази даних, запитів, що дозволяють виконувати різні операції з даними, та інших елементів, що дозволяють взаємодіяти з інформацією про товари на складі.

Схеми бази даних: ERD (Entity-Relationship Diagram), що відображає взаємозв'язки між таблицями.

Запити: SQL-запити для додавання, видалення, оновлення та вибірки даних.

Індекси: створення індексів для прискорення пошуку даних.

Тригери та процедури: автоматичне виконання певних дій при зміні даних у таблицях.

Фізичне представлення

Технічні аспекти зберігання даних:

Формати файлів: SQL-файли для збереження структури таблиць та даних.

Типи дискових пристроїв: SSD для швидкого доступу до даних.

Резервне копіювання: регулярне резервне копіювання даних на зовнішніх носіях або в хмарі.

Безпека: використання SSL для шифрування даних при передачі, контроль доступу на основі ролей.

Логічне представлення

Схема бази даних та SQL-запити:

Схема бази даних (ERD): відображає взаємозв'язки між таблицями та атрибутами.[17],[18],[19],[20]

SQL-запити:

Додавання нового товару:

```
INSERT INTO Товари (Назва, Опис, Категорія, Ціна, Кількість,
Дата_надходження)
```

```
VALUES ('Товар1', 'Опис товару', 1, 100, 10, '2024-05-01');
```

Оновлення інформації про товар:

```
UPDATE Товари
```

```
SET Кількість = 20
```

```
WHERE ID = 1;
```

Вибірка товарів за категорією:

```
SELECT * FROM Товари
```

```
WHERE Категорія = 1;
```

Індекси:

Створення індексу для прискорення пошуку товарів за назвою:

```
CREATE INDEX idx_товари_назва
```

```
ON Товари (Назва);
```

Тригери та процедури:

Приклад тригера, який автоматично оновлює кількість товару при його видачі:

```
CREATE TRIGGER on_товар_виданий
```

```
AFTER INSERT ON Рух_товарів
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Товари
```

```
    SET Кількість = Кількість - NEW.Кількість
```

```
    WHERE ID = NEW.ID_товару;
```

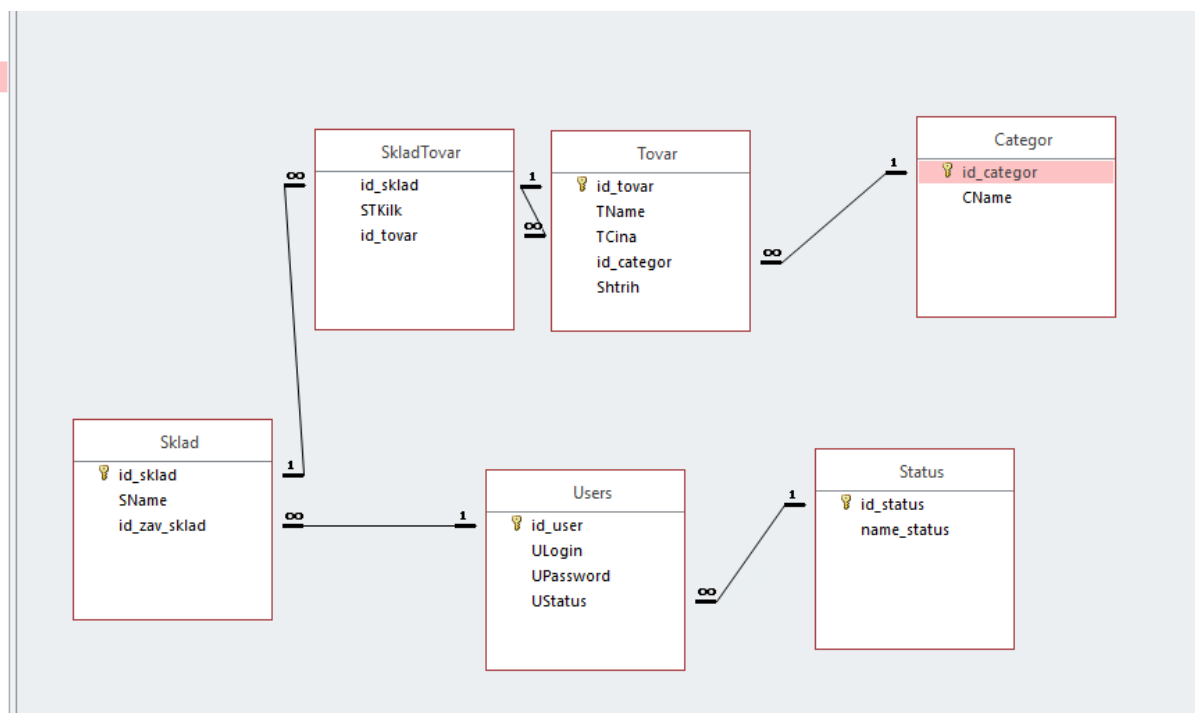
```
END;
```

Цей детальний опис кожного рівня представлення бази даних допоможе забезпечити ефективне та надійне управління даними в автоматизованому додатку обліку товарів на складі.

2.6.1. Створення логічної моделі бази даних

Після створення концептуального представлення бази даних для автоматизованого додатку обліку товарів на складі, наступним етапом стало створення логічного представлення. Логічне представлення полягає в перетворенні концептуальної моделі в модель, яка враховує особливості конкретної системи управління базою даних.

Детальний опис визначено та показано на рисунку 2.4 нижче.



Рисунку 2.4

Логічна модель

На етапі створення логічного представлення визначено сутності, атрибути та зв'язки між ними, відповідно до концептуальної моделі. Основні сутності логічної моделі включають:

На етапі створення логічного представлення були визначені обмеження цілісності даних, такі як обов'язковість заповнення певних атрибутів, обмеження на діапазон значень атрибутів та заборона на видалення записів, що мають зв'язки з іншими записами в базі даних. [17],[18],[19],[20]

2.6.2. Створення фізичної моделі бази даних

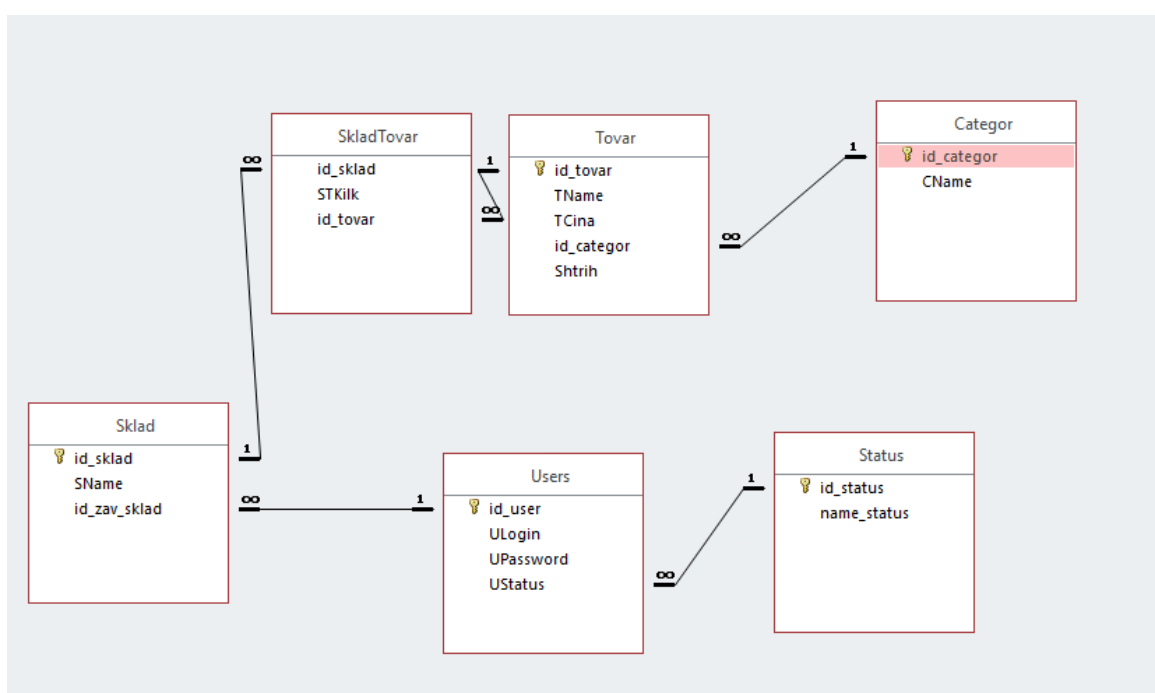
Створення фізичної моделі бази даних для автоматизованого додатку обліку товарів на складі

Після створення логічного представлення бази даних, наступним етапом стало створення фізичного представлення. Фізичне представлення полягає в

перетворенні логічної моделі в модель, яка відображає реальні характеристики фізичної реалізації бази даних та їх типів даних.

Фізична модель

На етапі створення фізичного представлення була визначена структура та параметри фізичної реалізації бази даних. Це включило в себе вибір типу сервера баз даних, налаштування параметрів бази даних та визначення способу зберігання даних.



Параметри забезпечення безпеки даних

Аутентифікація та авторизація: процеси ідентифікації користувачів та визначення їхніх прав доступу до даних. Використовуються різні механізми, такі як логіни та паролі, двофакторна аутентифікація, біометричні технології тощо.

Шифрування даних: процес перетворення звичайного тексту в зашифрований, що забезпечує конфіденційність даних. Використовуються різні алгоритми шифрування, такі як AES, RSA, Blowfish тощо.

Захист від SQL-ін'єкцій: захист від вразливості, при якій зловмисник може використати SQL-запит для витягування або зміни даних в базі даних.

Використовуються параметризовані запити та фільтрація введених користувачем даних.

Резервне копіювання: забезпечення можливості відновлення даних в разі їх втрати або пошкодження. Використовуються різні методи резервного копіювання, такі як повний бекап, інкрементальне копіювання тощо.

Моніторинг та аудит: процес відстеження доступу до даних та змін в базі даних з метою виявлення потенційних загроз безпеці. Використовуються системи моніторингу та аудиту, такі як Microsoft SQL Server Audit, Oracle Audit Vault тощо.

Завершуючи створення фізичного представлення бази даних, забезпечено її реалізацію згідно з встановленими параметрами та правильну інтеграцію з системою управління базою даних. Це дозволило забезпечити ефективне та безпечне функціонування бази даних та системи обліку товарів на складі в цілому. [17],[18],[19],[20]

2.7. Загальний вигляд сутностей бази даних

Створення логічної та фізичної моделі бази даних для автоматизованого додатку обліку товарів на складі

Після опису створення концептуального, логічного та фізичного представлення бази даних, сформовано загальний вигляд сутностей (таблиць) для початку розробки, програмування та подальшої розробки елементарної системи автоматизованого обліку товарів на складі.

Таблиця "Products" (Товари)

- ProductID (INT, PRIMARY KEY) – унікальний ідентифікатор товару
- Name (VARCHAR) – назва товару
- CategoryID (INT, FOREIGN KEY) – ідентифікатор категорії
- SupplierID (INT, FOREIGN KEY) – ідентифікатор постачальника
- QuantityPerUnit (VARCHAR) – кількість одиниць в упаковці
- UnitPrice (FLOAT) – ціна за одиницю
- UnitsInStock (INT) – кількість одиниць на складі
- UnitsOnOrder (INT) – кількість одиниць в замовленні

- ReorderLevel (INT) – рівень для повторного замовлення
- Discontinued (BOOLEAN) – позначка про зняття з продажу

Таблиця "Categories" (Категорії)

- CategoryID (INT, PRIMARY KEY) – унікальний ідентифікатор категорії
- CategoryName (VARCHAR) – назва категорії
- Description (TEXT) – опис категорії

Таблиця "Suppliers" (Постачальники)

- SupplierID (INT, PRIMARY KEY) – унікальний ідентифікатор постачальника
- CompanyName (VARCHAR) – назва компанії
- ContactName (VARCHAR) – ім'я контактної особи
- ContactTitle (VARCHAR) – посада контактної особи
- Address (VARCHAR) – адреса
- City (VARCHAR) – місто
- Region (VARCHAR) – регіон
- PostalCode (VARCHAR) – поштовий індекс
- Country (VARCHAR) – країна
- Phone (VARCHAR) – телефон
- Fax (VARCHAR) – факс
- HomePage (VARCHAR) – веб-сайт

Таблиця "Orders" (Замовлення)

- OrderID (INT, PRIMARY KEY) – унікальний ідентифікатор замовлення
- CustomerID (INT, FOREIGN KEY) – ідентифікатор клієнта
- EmployeeID (INT, FOREIGN KEY) – ідентифікатор працівника
- OrderDate (TIMESTAMP) – дата замовлення
- RequiredDate (TIMESTAMP) – дата вимоги
- ShippedDate (TIMESTAMP) – дата відправлення
- ShipVia (INT, FOREIGN KEY) – ідентифікатор транспортної компанії
- Freight (FLOAT) – вартість перевезення
- ShipName (VARCHAR) – назва одержувача

- ShipAddress (VARCHAR) – адреса одержувача
- ShipCity (VARCHAR) – місто одержувача
- ShipRegion (VARCHAR) – регіон одержувача
- ShipPostalCode (VARCHAR) – поштовий індекс одержувача
- ShipCountry (VARCHAR) – країна одержувача

Таблиця "OrderDetails" (Деталі замовлень)

- OrderID (INT, PRIMARY KEY, FOREIGN KEY) – унікальний ідентифікатор замовлення
- ProductID (INT, PRIMARY KEY, FOREIGN KEY) – унікальний ідентифікатор товару
- UnitPrice (FLOAT) – ціна за одиницю товару
- Quantity (INT) – кількість замовлених одиниць
- Discount (FLOAT) – знижка

Вибір створеної бази даних

Логічне представлення бази даних включає сутності, їх атрибути та взаємозв'язки між ними, а також обмеження цілісності даних.

Логічна модель з урахуванням взаємозв'язків

Вибір показу таблиць створеної бази даних

Фізичне представлення бази даних визначає структуру та параметри фізичної реалізації бази даних.

Параметри забезпечення безпеки даних

Аутифікація та авторизація: процеси ідентифікації користувачів та визначення їхніх прав доступу до даних. Використовуються різні механізми, такі як логіни та паролі, двофакторна аутифікація, біометричні технології тощо.

Шифрування даних: процес перетворення звичайного тексту в зашифрований, що забезпечує конфіденційність даних. Використовуються різні алгоритми шифрування, такі як AES, RSA, Blowfish тощо.

Захист від SQL-ін'єкцій: захист від вразливості, при якій зловмисник може використати SQL-запит для витягування або зміни даних в базі даних.

Використовуються параметризовані запити та фільтрація введених користувачем даних.

Резервне копіювання: забезпечення можливості відновлення даних в разі їх втрати або пошкодження. Використовуються різні методи резервного копіювання, такі як повний бекап, інкрементальне копіювання тощо.

Моніторинг та аудит: процес відстеження доступу до даних та змін в базі даних з метою виявлення потенційних загроз безпеці. Використовуються системи моніторингу та аудиту, такі як Microsoft SQL Server Audit, Oracle Audit Vault тощо.

Завершуючи створення фізичного представлення бази даних, забезпечено її реалізацію згідно з встановленими параметрами та правильну інтеграцію з системою управління базою даних. Це дозволило забезпечити ефективне та безпечне функціонування бази даних та системи обліку товарів на складі в цілому.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1. Вибір програмного інструментарію

3.1.1. Бази даних

Для реалізації автоматизованого додатку обліку товарів на складі, було обрано використання Microsoft Access як системи управління базами даних. MS Access надає зручний інтерфейс для створення, управління та маніпуляції даними, що робить його підходящим інструментом для даного проекту.

Переваги використання MS Access:

Легкість використання:

MS Access пропонує користувачам інтуїтивно зрозумілий інтерфейс, що дозволяє швидко створювати та налаштовувати бази даних без глибоких знань у програмуванні.

Містить багато вбудованих шаблонів і майстрів, які спрощують процес створення бази даних.

Інтеграція з іншими продуктами Microsoft:

Легка інтеграція з іншими продуктами Microsoft, такими як Excel, Word та Outlook, що спрощує імпорт та експорт даних.

Підтримка OLE (Object Linking and Embedding), що дозволяє зберігати документи та зображення у базі даних.

Функціональність для малих та середніх проектів:

MS Access підходить для невеликих та середніх баз даних, забезпечуючи необхідну функціональність для обліку товарів на складі.

Пропонує достатні можливості для розробки форм, звітів, запитів та макросів.

Швидкість розгортання та доступність:

MS Access дозволяє швидко створити прототип та розгорнути базу даних, що особливо важливо для проектів з обмеженими термінами.

Доступний у складі пакету Microsoft Office, що знижує витрати на придбання додаткового ПЗ.

Недоліки використання MS Access:

Обмежена масштабованість:

MS Access має обмежені можливості для роботи з великими обсягами даних та одночасного доступу багатьох користувачів, що може бути недостатнім для дуже великих проектів.

Для великих систем рекомендується розглянути інші СУБД, такі як SQL Server або MySQL.

Продуктивність:

При роботі з великими обсягами даних продуктивність MS Access може знижуватися.

Не підходить для дуже великих баз даних або високонавантажених систем.

Безпека:

MS Access забезпечує базові механізми безпеки, але вони можуть бути недостатніми для критично важливих систем, що потребують високого рівня захисту даних.

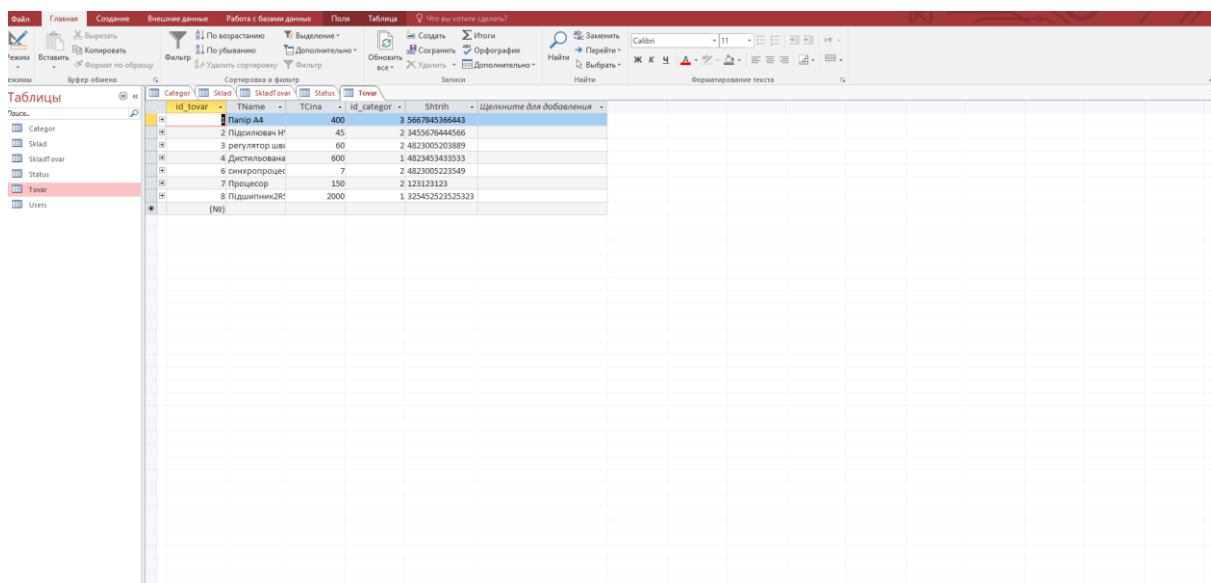


Рисунок 3.1 Інтерфейс MS Access

З огляду на специфіку та масштаб проекту, використання Microsoft Access є оптимальним вибором для створення бази даних автоматизованого додатку обліку товарів на складі. Він забезпечує необхідну функціональність, простоту

використання та швидкість розгортання, що дозволяє швидко створити та налаштувати базу даних для ефективного обліку товарів.[22]

3.1.2. Мова програмування

Для проектування інформаційного забезпечення автоматизованого додатку обліку товарів на складі необхідною задачею є вибір мови програмування, яка найкраще підходить для вирішення поставлених задач. Вибір мови програмування залежить від різних факторів, таких як складність проекту, доступність ресурсів, вимоги до продуктивності, масштабованості та безпеки.

Даний розділ проектування інформаційного забезпечення для автоматизованого додатку обліку товарів на складі описує ресурсну базу наступних завдань:

Вибір мови програмування для розробки додатків та систем.

Розробка структури бази даних та вибір системи управління базами даних (СУБД).

Визначення алгоритмів та інтерфейсів для взаємодії між різними пристроями та системами.

Вибір системи зберігання даних та забезпечення їх безпеки.

Мови програмування для проектування автоматизованого додатку обліку товарів на складі

Python – це інтерпретована мова програмування високого рівня, яка зазвичай використовується в розробці веб-додатків та наукових обчислень. Python також може бути використаний для створення програмного забезпечення для управління складськими запасами завдяки своєму читабельному та простому синтаксису, що робить його добрим вибором для розробки прототипів та швидкого прототипування.

Java – це об'єктно-орієнтована мова програмування, що має широке застосування в розробці додатків. Java є платформонезалежною, тому програми, написані на Java, можуть працювати на будь-якій операційній

системі. Java підтримує багато фреймворків, бібліотек та інших інструментів, що робить її відмінним вибором для розробки складних систем управління складом.

JavaScript – це мова програмування, яка часто використовується в розробці веб-додатків, але може бути використана й для розробки додатків для управління складом. JavaScript дозволяє створювати динамічні та інтерактивні веб-сторінки та додатки. JavaScript має велику кількість бібліотек та фреймворків, що полегшують розробку програмного забезпечення.[6]

C#/C++ – це компільовані мови програмування, які використовуються в розробці високопродуктивного програмного забезпечення. Однією з головних їх переваг є те, що вони дають розробникам більший контроль над пам'яттю та швидкістю виконання програми. Ці мови є дуже потужними, але вимагають від розробника більш високого рівня знань та досвіду. C++ та C# можуть бути використані для розробки додатків для управління складом, які потребують високої продуктивності та низької затримки.

Таблиця 3.2 – Порівняння мов програмування для використання в практичній реалізації продукту

Мова програмування	Переваги	Недоліки
Python	Простий синтаксис, велика кількість бібліотек, хороша продуктивність	Менш ефективний для високопродуктивних додатків
Java	Велика кількість фреймворків, хороша масштабованість, висока продуктивність	Вимагає великої кількості ресурсів для запуску, менш простий синтаксис
JavaScript	Легка взаємодія з компонентами веб-додатків, широкі можливості для	Менш ефективний для розробки масштабних додатків

Мова програмування	Переваги	Недоліки
	розробки інтерфейсів	
C++/C#	Висока продуктивність, можливість взаємодії з апаратними пристроями, ефективні алгоритми	Складний синтаксис, вимагає великої кількості коду

Тож, вибір мови програмування для проектування інформаційного забезпечення автоматизованого додатку обліку товарів на складі залежить від різних факторів. Для простих додатків можна використовувати мови з простим синтаксисом, такі як Python або JavaScript. Для більш складних додатків з високою продуктивністю та ефективними алгоритмами можна використовувати мови, такі як Java або C++/C#. Крім того, важливим фактором є наявність бібліотек та фреймворків для розробки додатків, які можуть значно полегшити процес розробки.

Під час вибору мови програмування для проектування автоматизованого додатку обліку товарів на складі, враховано вимоги до продуктивності, функціональності та зручності розробки, щоб забезпечити ефективну та швидку розробку інформаційного забезпечення та баз даних.

У цьому проекті обрано мову програмування C#.[21]

3.1.3. Програмне середовище

Для розробки автоматизованого додатку обліку товарів на складі обрано програмне середовище Visual Studio. Visual Studio є одним з найпопулярніших інтегрованих середовищ розробки (IDE), що надає потужні інструменти для створення високоякісного програмного забезпечення. Воно підтримує широкий

спектр мов програмування та технологій, що робить його універсальним інструментом для розробників.

Переваги використання Visual Studio

Підтримка мов програмування:

Visual Studio підтримує безліч мов програмування, таких як C#, C++, Python, JavaScript та інші, що дозволяє розробляти різноманітні проекти в одному середовищі.

Інтеграція з .NET Framework:

Глибока інтеграція з .NET Framework та .NET Core забезпечує легку розробку, тестування та розгортання додатків.

Інструменти для налагодження та тестування:

Вбудовані інструменти для налагодження та тестування допомагають розробникам швидко знаходити та виправляти помилки.

Інтеграція з Azure:

Можливість інтеграції з хмарними сервісами Azure для розробки, тестування та розгортання хмарних додатків.

Розширення та підтримка:

Величезна кількість доступних розширень дозволяє налаштувати IDE під потреби конкретного проекту.

Інші програмні середовища для порівняння

1. PyCharm

Підтримка мов програмування:

Основна підтримка для Python, але також підтримує HTML, CSS, JavaScript та SQL.

Інструменти для налагодження та тестування:

Вбудовані інструменти для налагодження, тестування та профілювання Python-коду.

Інтеграція з веб-фреймворками:

Підтримка веб-фреймворків, таких як Django, Flask та інші.

Переваги:

Потужне середовище для Python-розробки з безліччю інструментів для підвищення продуктивності.

Недоліки:

Обмежена підтримка інших мов програмування.

2. IntelliJ IDEA

Підтримка мов програмування:

Підтримка широкого спектру мов, включаючи Java, Kotlin, Groovy, Scala та інші.

Інтеграція з фреймворками:

Підтримка популярних фреймворків, таких як Spring, Hibernate та інших.

Інструменти для налагодження та тестування:

Потужні інструменти для налагодження та тестування, включаючи інтеграцію з JUnit та TestNG.

Переваги:

Інтуїтивно зрозумілий інтерфейс, потужні інструменти для налагодження та підтримка великих проектів.

Недоліки:

Може бути занадто важким для невеликих проектів.

3. Eclipse

Підтримка мов програмування:

Підтримує Java, C++, Python, PHP та багато інших мов через плагіни.

Інтеграція з фреймворками:

Підтримка різних фреймворків та інструментів, таких як Spring, Hibernate, Jenkins та інших.

Інструменти для налагодження та тестування:

Вбудовані інструменти для налагодження та тестування, включаючи підтримку JUnit.

Переваги:

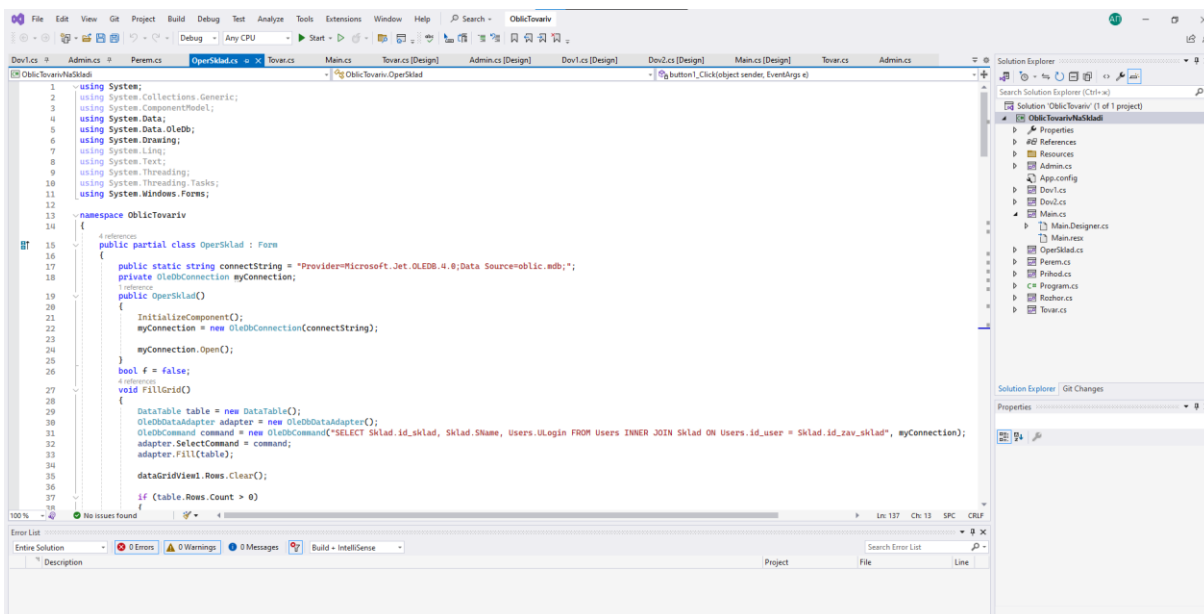
Відкрите програмне забезпечення з великою кількістю плагінів та розширень.

Недоліки:

Менш інтуїтивно зрозумілий інтерфейс порівняно з іншими IDE.

Таблиця 3.3 – Порівняльна характеристика програмних середовищ

Програмне середовище	Переваги	Недоліки
Visual Studio	Підтримка багатьох мов програмування, глибока інтеграція з .NET, потужні інструменти для налагодження та тестування, інтеграція з Azure	Високі системні вимоги, платне ПЗ (є безкоштовна версія з обмеженою функціональністю)
PyCharm	Потужне середовище для Python, підтримка веб-фреймворків, інструменти для налагодження та тестування	Обмежена підтримка інших мов програмування, платне ПЗ (є безкоштовна версія з обмеженою функціональністю)
IntelliJ IDEA	Підтримка багатьох мов, потужні інструменти для налагодження та тестування, підтримка великих проектів	Високі системні вимоги, платне ПЗ (є безкоштовна версія з обмеженою функціональністю)
Eclipse	Відкрите програмне забезпечення, велика кількість плагінів, підтримка багатьох мов	Менш інтуїтивно зрозумілий інтерфейс, потребує налаштування



4. ЕРГОНОМІКА В ІТ

4.1. Вступ до ергономіки в ІТ

Ергономіка в інформаційних технологіях (ІТ) є наукою, яка займається оптимізацією взаємодії людини з комп'ютерними системами та програмним забезпеченням. Вона спрямована на забезпечення комфортних, безпечних та продуктивних умов праці для користувачів та розробників, що, в свою чергу, сприяє зниженню ризиків для їхнього здоров'я та підвищенню загальної ефективності роботи.[24]

Основні завдання ергономіки в ІТ:

Створення комфортних умов праці: Забезпечення фізичного та психологічного комфорту для користувачів та розробників, враховуючи робоче місце, освітлення, рівень шуму та інші фактори.

Забезпечення безпеки: Впровадження заходів, що запобігають травмам та професійним захворюванням, таких як тунельний синдром, проблеми зі спиною та очима.

Підвищення продуктивності: Оптимізація робочих процесів та інтерфейсів користувача для забезпечення максимальної ефективності та мінімізації помилок.

Зниження стресу: Створення інтуїтивно зрозумілих та легких у використанні систем, що зменшують когнітивне навантаження на користувачів.

Важливість ергономіки в ІТ:

Підвищення задоволеності користувачів: Добре спроектовані системи та інтерфейси сприяють позитивному користувацькому досвіду, що підвищує задоволеність користувачів та їх лояльність.

Зменшення витрат на навчання: Інтуїтивно зрозумілі системи скорочують час, необхідний для навчання нових користувачів, що знижує витрати на тренінги та підтримку.

Зменшення помилок та підвищення якості: Ергономічно спроектовані системи допомагають зменшити кількість помилок, що покращує якість продуктів та послуг.

Покращення здоров'я та благополуччя: Зменшення фізичних та психічних навантажень на користувачів сприяє покращенню їхнього здоров'я та загального благополуччя.

Основні напрямки ергономіки в ІТ:

Дизайн інтерфейсів користувача: Розробка інтуїтивних, простих та естетично привабливих інтерфейсів, які полегшують взаємодію з програмним забезпеченням.

Організація робочого місця: Оптимізація робочого місця для забезпечення комфортних умов праці, включаючи розташування обладнання, освітлення та акустичний комфорт.

Розробка програмного забезпечення: Врахування ергономічних принципів на всіх етапах розробки програмного забезпечення, від проектування до тестування та впровадження.

Психологічні аспекти: Врахування психічного стану та мотивації користувачів, створення умов для зниження стресу та підвищення мотивації.

Ергономіка в ІТ є багатогранною дисципліною, яка включає в себе знання з галузі психології, фізіології, дизайну та інженерії. Її головною метою є створення таких систем, які б максимально відповідали потребам та можливостям користувачів, забезпечуючи при цьому їх комфорт, безпеку та ефективність.

4.2. Вимоги до програмного забезпечення та підходи до його проектування з погляду користувача

Ергономічні цілі та показники якості програмного продукту для автоматизованого додатку обліку товарів на складі

Зручність використання

Програмне забезпечення для обліку товарів на складі повинно бути простим та інтуїтивним у використанні. Це означає, що користувачі, включаючи складських працівників та менеджерів, повинні швидко освоювати роботу з системою без тривалого навчання. Інтерфейс має бути зрозумілим і доступним для всіх категорій користувачів, забезпечуючи легкий доступ до основних функцій, таких як додавання нових товарів, перегляд наявності, формування звітів та проведення інвентаризації.

Продуктивність

Система повинна забезпечувати швидке та ефективне виконання завдань, пов'язаних з обліком товарів на складі. Це включає швидку обробку запитів, миттєвий доступ до даних про товари, автоматичне оновлення інформації в режимі реального часу та швидке генерування звітів. Висока продуктивність системи дозволить мінімізувати затримки та підвищити ефективність роботи складу.

Безпека та надійність

Захист від помилок та збоїв є критично важливим для системи обліку товарів на складі. Програмне забезпечення повинно мати механізми для виявлення та виправлення помилок, а також забезпечувати резервне копіювання даних для запобігання їх втраті. Крім того, система повинна забезпечувати високий рівень безпеки даних користувачів, включаючи захист від несанкціонованого доступу та витоку інформації. Це може бути досягнуто за допомогою аутентифікації користувачів, шифрування даних та моніторингу дій у системі.

Адаптивність

Можливість налаштування інтерфейсу під потреби різних користувачів є важливим аспектом ергономіки програмного забезпечення для обліку товарів на складі. Інтерфейс повинен бути гнучким та дозволяти користувачам налаштовувати його відповідно до своїх потреб та уподобань. Це може включати зміну мови інтерфейсу, налаштування відображення даних, вибір

режимів роботи (наприклад, денний або нічний режим) та інші параметри, що підвищують зручність роботи з системою.

Реалізація ергономічних цілей

Реалізація зазначених ергономічних цілей в автоматизованому додатку обліку товарів на складі вимагає дотримання декількох ключових підходів до проектування:

Аналіз потреб користувачів: Вивчення реальних потреб та вимог користувачів, включаючи складських працівників, менеджерів та адміністраторів системи. Це допоможе визначити необхідні функції та оптимальний дизайн інтерфейсу.

Прототипування та тестування: Створення прототипів інтерфейсу та проведення тестувань з користувачами для виявлення та виправлення недоліків. Це дозволить зробити інтерфейс максимально зручним та інтуїтивним.

Модульність та масштабованість: Розробка програмного забезпечення з використанням модульного підходу, що дозволяє легко додавати нові функції та налаштування. Це забезпечить гнучкість та адаптивність системи.

Інтеграція з іншими системами: Забезпечення можливості інтеграції з іншими системами, такими як ERP, CRM та бухгалтерські системи, для підвищення ефективності управління складом.

Навчання та підтримка користувачів: Надання детальної документації, навчальних матеріалів та підтримки для користувачів. Це допоможе швидко освоїти роботу з системою та підвищить її ефективність.

Завдяки дотриманню цих підходів, автоматизований додаток обліку товарів на складі зможе забезпечити високу зручність, продуктивність, безпеку та адаптивність, що сприятиме ефективній роботі складу та задоволенню потреб користувачів.

4.3. Параметри, які необхідно враховувати при розробці інтерфейсу користувача

Основні характеристики

Простота

Простота є ключовою характеристикою при розробці інтерфейсу користувача для автоматизованого додатку обліку товарів на складі. Інтерфейс має бути зрозумілим навіть для новачків, які можуть не мати великого досвіду роботи з комп'ютерними системами. Це означає, що всі елементи інтерфейсу повинні бути інтуїтивно зрозумілими, з мінімальною кількістю необхідних дій для виконання основних завдань, таких як додавання нових товарів, оновлення інформації про запаси, створення звітів та проведення інвентаризації.

Послідовність

Всі елементи інтерфейсу повинні бути послідовними у зовнішньому вигляді та поведінці. Це включає використання однакових шрифтів, кольорових схем, іконок та розташування елементів на всіх сторінках додатку. Послідовність дозволяє користувачам швидко освоїти систему та передбачити її поведінку, що знижує ймовірність помилок та підвищує ефективність роботи.

Зворотний зв'язок

Система повинна надавати користувачу зворотний зв'язок про виконані дії. Наприклад, після додавання нового товару або оновлення інформації про запаси, система повинна відображати повідомлення про успішне виконання операції. Якщо виникла помилка, користувач повинен отримати детальну інформацію про причину помилки та можливі способи її виправлення. Зворотний зв'язок допомагає користувачам розуміти, що їхні дії призвели до очікуваних результатів, та сприяє швидкому виявленню та виправленню помилок.

Стандартизація

Інтерфейс повинен відповідати загальноприйнятим стандартам дизайну та юзабіліті. Це включає використання стандартних елементів управління, таких як кнопки, поля введення, списки та діалогові вікна, а також дотримання

принципів доступності та зручності використання. Стандартизація полегшує навчання користувачів, оскільки вони вже знайомі з основними елементами управління з інших програм.

Приклади реалізації

Простота

На головній сторінці додатку розміщуються найбільш використовувані функції, такі як додавання нового товару, перегляд поточного стану запасів та створення звітів.

Використання зрозумілих іконок та підказок при наведенні курсору на елементи інтерфейсу допомагає користувачам швидко орієнтуватися в системі.

Послідовність

Всі форми введення даних мають однакову структуру та розташування полів, що дозволяє користувачам швидко заповнювати їх без необхідності адаптації до нового формату.

Використання єдиної кольорової схеми для всіх елементів інтерфейсу забезпечує візуальну гармонію та полегшує сприйняття інформації.

Зворотний зв'язок

Після додавання нового товару система відображає повідомлення "Товар успішно доданий" з можливістю перегляду доданої інформації.

У разі помилки, наприклад, при спробі додати товар з уже існуючим кодом, відображається повідомлення "Помилка: товар з таким кодом уже існує. Будь ласка, перевірте введені дані."

Стандартизація

Використання стандартних компонентів управління, таких як кнопки з підписами "Додати", "Оновити", "Видалити", забезпечує зрозумілість дій для користувачів.

Дотримання принципів доступності, таких як можливість навігації за допомогою клавіатури та підтримка екранних читалок для користувачів з обмеженими можливостями.

Ці характеристики допоможуть створити зручний, ефективний та надійний інтерфейс для автоматизованого додатку обліку товарів на складі, що забезпечить високий рівень задоволеності користувачів та підвищить продуктивність їхньої роботи.

4.4. Проблеми та особливості розробки прототипу інтерфейсу

Принципи реалізації призначеного для користувача інтерфейсу

Ітеративний підхід

Розробка інтерфейсу для автоматизованого додатку обліку товарів на складі повинна здійснюватися через серію ітерацій з постійним вдосконаленням на основі зворотного зв'язку від користувачів. Цей підхід дозволяє виявляти та виправляти помилки, а також додавати нові функції на кожному етапі розробки. На початкових етапах можна розробити базовий інтерфейс з основними функціями, такими як додавання нових товарів, оновлення інформації про запаси та створення звітів. Після отримання зворотного зв'язку від користувачів можна внести необхідні корективи та додати нові функціональні можливості.

Прототипування

Створення прототипів є важливим етапом у розробці інтерфейсу для автоматизованого додатку обліку товарів на складі. Прототипи дозволяють тестувати та виявляти проблеми на ранніх етапах розробки, що допомагає уникнути великих змін та витрат на пізніх етапах. Прототипи можуть бути як паперовими, так і цифровими, включаючи макети інтерфейсу, що дозволяють візуалізувати розташування елементів, навігацію та взаємодію з користувачем. Це допомагає забезпечити зручність та ефективність роботи з системою.

Юзабіліті-тестування

Проведення тестування з користувачами є важливим кроком у виявленні та виправленні недоліків інтерфейсу. Юзабіліті-тестування дозволяє оцінити, наскільки інтуїтивним та зручним є інтерфейс для кінцевих користувачів. Для

автоматизованого додатку обліку товарів на складі це може включати тестування з реальними складськими працівниками та менеджерами. Під час тестування користувачі виконують типові завдання, такі як додавання нових товарів, оновлення запасів та формування звітів, а розробники спостерігають за їх діями та збирають зворотний зв'язок. Це допомагає виявити проблемні зони та оптимізувати інтерфейс для покращення продуктивності та зручності використання.

Приклади реалізації

Ітеративний підхід

Перший ітераційний цикл включає розробку базового інтерфейсу з функціями додавання нових товарів та перегляду наявних запасів.

Після отримання зворотного зв'язку від користувачів, інтерфейс оновлюється для покращення навігації та додавання нових функцій, таких як автоматичне оновлення даних та інтеграція зі сканерами штрих-кодів.

Прототипування

На початковому етапі розробки створюються паперові прототипи інтерфейсу, які дозволяють швидко оцінити розташування елементів та основну навігацію.

Далі створюються цифрові прототипи, що дозволяють інтерактивно тестувати інтерфейс та виявляти проблеми на ранніх етапах.

Юзабіліті-тестування

Проведення тестування з реальними користувачами, такими як складські працівники, які виконують типові завдання, дозволяє виявити недоліки інтерфейсу та отримати цінний зворотний зв'язок.

Виправлення виявлених проблем та повторне тестування забезпечує постійне вдосконалення інтерфейсу.

Застосування цих принципів у розробці інтерфейсу для автоматизованого додатку обліку товарів на складі допоможе створити зручну, ефективну та надійну систему, яка відповідає потребам користувачів та забезпечує високу продуктивність роботи складу.

4.5. Вимоги до процесів інтерфейсу та проектування і реалізація його компонентів

4.5.1. Проектування компонентів інтерфейсу

Планування

Планування є першим етапом у проектуванні компонентів інтерфейсу для автоматизованого додатку обліку товарів на складі. На цьому етапі визначаються вимоги до інтерфейсу, включаючи функціональні та нефункціональні вимоги. Основними аспектами планування є визначення цільових користувачів системи (складські працівники, менеджери), їхніх потреб та сценаріїв використання додатку. Планування також включає розробку технічного завдання, в якому описуються основні компоненти інтерфейсу, їхня функціональність та взаємозв'язки.

Розробка макетів

На цьому етапі створюються візуальні макети інтерфейсу, які допомагають візуалізувати дизайн та структуру системи. Макети можуть бути як низької деталізації (паперові ескізи), так і високої деталізації (цифрові прототипи). Вони відображають розташування елементів інтерфейсу, таких як форми введення даних, кнопки, меню та інформаційні панелі. Розробка макетів допомагає отримати зворотний зв'язок від користувачів та команд розробників до початку фактичної реалізації.

Інтеграція

Інтеграція включає об'єднання всіх компонентів інтерфейсу в єдину систему. Це означає, що всі макети, створені на попередньому етапі, перетворюються в інтерактивні елементи, які взаємодіють один з одним. Інтеграція також включає налаштування зв'язків між інтерфейсом та базою даних, забезпечення правильного відображення інформації про товари, запаси та інші дані, що зберігаються в системі.

4.5.2. Реалізація компонентів інтерфейсу

Розробка

На цьому етапі відбувається написання коду для реалізації функціоналу інтерфейсу. Розробка включає створення форм для введення даних про товари, інтерфейсу для перегляду та редагування інформації про запаси, створення звітів та інших функцій. Використовуються сучасні мови програмування та фреймворки. Важливо забезпечити, щоб інтерфейс був адаптивним та міг працювати на різних пристроях, включаючи комп'ютери, планшети та смартфони.

Тестування

Після завершення розробки необхідно провести тестування працездатності інтерфейсу та його відповідності вимогам. Тестування включає перевірку функціональності всіх елементів інтерфейсу, виявлення та виправлення помилок, а також проведення юзабіліті-тестування з реальними користувачами. Тестування допомагає переконатися, що інтерфейс працює стабільно, забезпечує коректне введення та відображення даних, а також відповідає вимогам щодо зручності використання.

Впровадження

На цьому етапі інтерфейс інтегрується в кінцевий продукт та запускається для використання. Впровадження включає налаштування програмного забезпечення на робочих станціях користувачів, проведення навчання для персоналу, а також забезпечення технічної підтримки. Після впровадження система повинна бути готова до використання в реальних умовах роботи складу, забезпечуючи ефективне управління запасами та облік товарів.

Застосування цих процесів у розробці інтерфейсу для автоматизованого додатку обліку товарів на складі допоможе створити зручну, ефективну та надійну систему, яка відповідатиме потребам користувачів та забезпечуватиме високу продуктивність роботи складу.

4.6. Психологічні аспекти ергономіки

4.6.1. Мотивація користувачів

Гейміфікація

У процесі розробки автоматизованого додатку обліку товарів на складі, використання елементів гейміфікації може значно підвищити мотивацію користувачів. Це може включати систему нагород за виконання певних завдань, досягнення певних цілей, наприклад, ведення обліку без помилок протягом певного періоду часу, або швидке виконання завдань. Введення рейтингової системи або таблиці лідерів серед працівників складу також може мотивувати користувачів працювати ефективніше та уважніше.

Задоволення від роботи

Створення інтерфейсу, який приносить задоволення від роботи, є важливим аспектом підвищення продуктивності та мотивації користувачів. Інтерфейс додатку повинен бути привабливим, зручним у використанні, мати інтуїтивно зрозумілу навігацію та приємний дизайн. Наприклад, використання яскравих кольорів, зрозумілих іконок та анімацій може зробити роботу з системою більш приємною. Крім того, швидка реакція системи на дії користувачів, відсутність затримок і помилок також сприяють підвищенню задоволення від роботи.

4.6.2. Зниження стресу

Простота в роботі

Зменшення складності завдань та забезпечення чітких інструкцій є ключовими факторами для зниження стресу користувачів. Інтерфейс додатку повинен бути максимально простим та інтуїтивним, щоб навіть новачки могли легко зрозуміти, як ним користуватися. Важливо передбачити чіткі інструкції та підказки для виконання основних операцій, таких як введення даних про нові товари, оновлення інформації про запаси, створення звітів. Інструкції можуть бути представлені у вигляді спливаючих підказок, інтерактивних турів або відеоінструкцій.

Підтримка користувачів

Наявність підтримки та допомоги для користувачів у разі виникнення проблем є важливим аспектом для зниження стресу. Користувачі повинні мати можливість швидко отримати допомогу у разі виникнення будь-яких труднощів

або питань. Це може бути реалізовано через систему онлайн-чату, телефонну підтримку або розділ з часто задаваними питаннями (FAQ). Крім того, важливо передбачити можливість зворотного зв'язку, щоб користувачі могли повідомляти про проблеми або давати свої пропозиції щодо покращення системи.

Приклади реалізації

Гейміфікація

Запровадження системи досягнень та нагород для працівників, які ефективно використовують додаток.

Введення щомісячних нагород для найкращих працівників, які досягли високих результатів у веденні обліку товарів.

Задоволення від роботи

Використання сучасного та привабливого дизайну інтерфейсу з інтуїтивно зрозумілою навігацією.

Впровадження швидкої реакції системи на дії користувачів та мінімізація затримок.

Простота в роботі

Створення чітких та зрозумілих інструкцій для виконання основних операцій у додатку.

Використання інтерактивних турів для нових користувачів, які пояснюють основні функції додатку.

Підтримка користувачів

Наявність системи онлайн-чату для швидкого отримання допомоги.

Розділ FAQ з відповідями на найпоширеніші питання користувачів.

Врахування психологічних аспектів ергономіки у розробці автоматизованого додатку обліку товарів на складі допоможе створити зручну, ефективну та комфортну систему, яка задовольняє потреби користувачів та сприяє підвищенню їхньої продуктивності та задоволеності роботою.[25]

4.7. Фізичні аспекти ергономіки

4.7.1. Запобігання професійним захворюванням

Регулярні перерви

Під час розробки та впровадження автоматизованого додатку обліку товарів на складі, важливо враховувати фізичні аспекти ергономіки для запобігання професійним захворюванням серед користувачів. Робота з комп'ютером на складі може призвести до напруги очей та м'язів. Рекомендується проводити регулярні перерви для зняття напруги. Працівникам слід робити перерви кожні 60 хвилин на 5-10 хвилин, під час яких вони можуть виконувати легкі вправи для рук та очей, відпочивати від екрану.

Вправи для рук та очей

Виконання простих вправ для рук та очей може допомогти запобігти розвитку тунельного синдрому та інших проблем, пов'язаних із тривалим використанням комп'ютера. Наприклад, розтягування пальців, обертання зап'ясть, фокусування погляду на віддалених об'єктах. Такі вправи можна легко інтегрувати в робочий графік, забезпечуючи працівникам необхідну підтримку для підтримання здоров'я.

4.7.2. Оптимізація робочого середовища

Температура та вентиляція

Оптимізація робочого середовища є важливим аспектом ергономіки для забезпечення комфорту та продуктивності працівників. На складах, де використовується автоматизований додаток для обліку товарів, необхідно забезпечити комфортну температуру та добру вентиляцію. Оптимальна температура повинна бути в межах 20-24 градусів Цельсія. Вентиляція повинна забезпечувати постійний приток свіжого повітря, що сприяє зменшенню втоми та підвищенню продуктивності працівників.

Акустичний комфорт

Зниження рівня шуму в робочому середовищі також є важливим фактором для підвищення комфорту та продуктивності працівників. Високий рівень шуму

може викликати стрес та знижувати концентрацію. Використання звукоізолюючих матеріалів, розміщення обладнання, що створює шум, в окремих приміщеннях, а також використання навушників із захистом від шуму може допомогти знизити рівень шуму та забезпечити комфортні умови для роботи.

Приклади реалізації

Регулярні перерви

Впровадження нагадувань у додатку для обліку товарів про необхідність зробити перерву кожну годину.

Створення програми, яка пропонує виконувати вправи для рук та очей під час перерв.

Вправи для рук та очей

Включення в інструкції з використання додатку рекомендацій щодо виконання вправ для зняття напруги.

Проведення тренінгів для працівників щодо виконання вправ для запобігання професійним захворюванням.

Температура та вентиляція

Встановлення системи кондиціонування та вентиляції у приміщеннях складу для забезпечення оптимальної температури та циркуляції повітря.

Регулярне обслуговування систем вентиляції для підтримання їх ефективної роботи.

Акустичний комфорт

Використання звукоізолюючих матеріалів у приміщеннях складу для зниження рівня шуму.

Забезпечення працівників навушниками із захистом від шуму для підвищення їхнього комфорту та концентрації під час роботи.

Врахування фізичних аспектів ергономіки у розробці та впровадженні автоматизованого додатку обліку товарів на складі сприятиме створенню комфортного та безпечного робочого середовища, підвищенню продуктивності працівників та зниженню ризиків для їхнього здоров'я.

ВИСНОВОК

З метою забезпечення ефективного та зручного обліку товарів на складі, з'явилася потреба в автоматизованих системах обліку. Ці системи дозволяють підвищити ефективність управління складськими запасами, зменшити ймовірність помилок, пов'язаних з ручним введенням даних, та забезпечити високу точність обліку товарів.

У цьому проекті розроблено інформаційне забезпечення для автоматизованого додатку обліку товарів на складі, що включає в себе базу даних та програмне забезпечення для її використання. Для цього були використані сучасні технології програмування, які дозволяють ефективно працювати з базою даних та забезпечувати її надійність і безпеку.

Результатом даної роботи є створення функціональних частин системи обліку товарів на складі, які можуть бути використані для базового створення реалізації ідеї ефективного управління складом з можливістю подальшого розширення та вдосконалення. Це може бути корисним як для приватних підприємців, так і для великих бізнес-організацій, які бажають забезпечити високий рівень ефективності та точності в управлінні своїми запасами.

Важливість створення інформаційного забезпечення для систем обліку товарів на складі полягає в тому, що воно є необхідною складовою для ефективною та надійною роботи системи. Без відповідної бази даних та програмного забезпечення система обліку не може функціонувати належним чином і не може забезпечити потрібний рівень точності та ефективності в управлінні складом.

Створення інформаційного забезпечення для систем обліку товарів на складі є ключовим елементом у забезпеченні ефективною та точною роботи таких систем. Застосування відповідних технологій та підходів у процесі

проектування та розробки дозволяє досягти максимальної продуктивності та забезпечити користувачів необхідними функціями та можливостями.

Крім того, системи обліку товарів на складі стають все більш популярними та доступними для звичайних користувачів, оскільки вони дозволяють автоматизувати багато рутинних процесів, що сприяє економії часу та підвищенню продуктивності. Важливою перевагою є також забезпечення точного обліку товарів, що допомагає уникнути помилок і недочетів, що можуть виникнути при ручному введенні даних. Завдяки відомостям даної дипломної роботи, людина, яка мінімально розбирається в управлінні складськими процесами, може створити елементарну систему обліку, розібравшись у програмному забезпеченні приладу та підключенні/створенні своєї бази даних.

Таким чином, створення інформаційного забезпечення для систем обліку товарів на складі є необхідним елементом у забезпеченні їх продуктивності, точності та ефективності. Розробка таких систем вимагає знань та досвіду у галузі комп'ютерних наук та програмування, а також уважного аналізу потреб та вимог користувачів. Але попри це, дана ніша робить значний внесок у розвиток сфери ІТ в цілому.

У першому розділі було описано аналіз та дослідження проблеми розробки інформаційного забезпечення та баз даних системи обліку товарів на складі, досліджені питання: аналіз існуючих рішень, постановка задачі, проблематика взаємозв'язку баз даних з реалізаціями наявних систем.

Другий розділ описує процес проектування інформаційного забезпечення, а саме: розглянуті ключові питання постановки завдання проектування ІЗ, дерева функцій, вибору технології зв'язку, вибору технології забезпечення безпеки, вибору інтерфейсу користувача (вимог до інтерфейсу, загальних можливостей структури інтерфейсу, вибору типу інтерфейсу користувача) та в подальшому проектування моделі бази даних.

Третій розділ виконує задачу вибору програмного інструментарію для подальшого створення та розробки бази даних та інформаційного забезпечення системи обліку товарів на складі.

Четвертий розділ забезпечує інформацію щодо ергономіки в ІТ, описуючи вимоги до програмного забезпечення, принципи проектування інтерфейсу, а також психологічні та фізичні аспекти, які враховуються під час розробки та впровадження системи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Silberschatz A., Korth H. F., & Sudarshan S. Database System Concepts - McGraw-Hill. 2010 https://nibmehub.com/opac-service/pdf/read/Python%20Programming%20_%20an%20introduction%20to%20computer%20science-%203rd%20Edition.pdf
2. John M. Zelle. Python Programming - Franklin, Beedle. 2004 https://nibmehub.com/opac-service/pdf/read/Python%20Programming%20_%20an%20introduction%20to%20computer%20science-%203rd%20Edition.pdf
3. Chaidhary H. Introduction to C++ - CreateSpace Independent Publishing Platform. 2014 https://learn.saylor.org/course/view.php?id=65&gad_source=1&gclid=Cj0KCQjw4MSzBhC8ARIsAPFOuyXUBg2RGl-Lhczmhe1SaVu2MUxTXgo3_CZ4duGeZSYpJr8WV5ZH04aAs_xEALw_wcB
4. Kaufmann, M., Meier A. SQL and NoSQL Databases - Springer Nature Switzerland. 2023 https://www.hs-albsig.de/weiterbildungsangebot/databases?gad_source=1&gclid=Cj0KCQjw4MSzBhC8ARIsAPFOuyXBigm6vQRMnCvW0yXJH0mRPpjKtR0nUrqe22jUhnLtuApcb4At1CsaAt9nEALw_wcB
5. Finch B. How to Write a Business Plan - Kogan Page. 2010 https://books.google.com.ua/books/about/How_to_Write_a_Business_Plan.html?id=oTVxD3hzxc4C&redir_esc=y
6. Platt M. Document Oriented Databases and Text Processing - GRIN Verlag. 2014 <https://theses.hal.science/tel-04500393v1/document>
7. Kahate A. Introduction to Database Management Systems. 2004 https://books.google.com.ua/books/about/Introduction_to_Database_Management_System.html?id=mxYESolfLfoC&redir_esc=y
8. Singh S. K. Database Systems. Concepts, Design and Applications - Pearson Education. 2009

https://books.google.com.ua/books?id=UBPanQEACAAJ&pg=PR5&hl=uk&source=gbv_selected_pages&cad=1#v=onepage&q&f=false

9. What is MongoDB Atlas? [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/docs/atlas/>

10. Indexes - Режим доступу: <https://www.mongodb.com/docs/manual/indexes/>

11. What is MongoDB Compass? [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/docs/compass/current/>

12. What is a Cloud Database? [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/resources/basics/databases/cloud-databases>

13. Visual Studio - Режим доступу: <https://visualstudio.microsoft.com/vs/>

14. JSON and BSON [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/resources/basics/json-and-bson>

15. CouchDB - Режим доступу: <https://couchdb.apache.org/>

16. What is MongoDB? [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/company/what-is-mongodb>

17. PostgreSQL Global Development Group. "PostgreSQL 13 Documentation." PostgreSQL, 2020. <https://cran.r-project.org/web/packages/RSQLite/RSQLite.pdf>

18. MySQL Documentation Team. "MySQL 8.0 Reference Manual." Oracle, 2020. <https://cran.r-project.org/web/packages/RSQLite/RSQLite.pdf>

19. Microsoft. "SQL Server Documentation." Microsoft, 2020. <https://cran.r-project.org/web/packages/RSQLite/RSQLite.pdf>

20. SQLite Consortium. "SQLite Documentation." SQLite, 2020. <https://cran.r-project.org/web/packages/RSQLite/RSQLite.pdf>

21. C#: що це за мова та де її використовують [Електронний ресурс] - Режим доступу: <https://robotdreams.cc/uk/blog/284-s-cho-to-eto-za-yazyk-i-gde-ego-ispolzuyut>

22. Microsoft Access [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Access

23. Microsoft Visual Studio [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio

24. ЕРГОНОМІКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ [Електронний ресурс] -

Режим

доступу:

https://org2.knuba.edu.ua/pluginfile.php/196069/mod_resource/content/2/%D0%95%D0%86%D0%A2_%D0%9A%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82.pdf

25. Using ergonomics to reduce pain from technology use [Електронний ресурс] -

Режим доступу: <https://www.sciencedaily.com/releases/2021/12/211217123806.htm>

Додатки

Додаток 1

```

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

namespace OblicTovariv
{
    public partial class OperSklad : Form
    {
        public static string connectString =
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=oblic.mdb;";
        private OleDbConnection myConnection;
        public OperSklad()
        {
            InitializeComponent();
            myConnection = new OleDbConnection(connectString);

            myConnection.Open();
        }
        bool f = false;
        void FillGrid()
        {
            DataTable table = new DataTable();
            OleDbDataAdapter adapter = new OleDbDataAdapter();
            OleDbCommand command = new OleDbCommand("SELECT
Sklad.id_sklad, Sklad.SName, Users.ULogin FROM Users INNER JOIN Sklad
ON Users.id_user = Sklad.id_zav_sklad", myConnection);
            adapter.SelectCommand = command;
            adapter.Fill(table);

            dataGridView1.Rows.Clear();

            if (table.Rows.Count > 0)
            {
                OleDbDataReader reader = command.ExecuteReader();

```

```

int i = 0;

while (reader.Read())
{
    dataGridView1.Rows.Add();
    dataGridView1[0, i].Value = reader[0];
    dataGridView1[1, i].Value = reader[1];
    dataGridView1[2, i].Value = reader[2];
    i++;
}
reader.Close();
}
}
private void OperSklad_Load(object sender, EventArgs e)
{
    FillGrid();
    DataTable table1 = new DataTable();
    OleDbDataAdapter adapter1 = new OleDbDataAdapter();
    OleDbCommand command1 = new OleDbCommand("SELECT ULogin
FROM Users", myConnection);
    adapter1.SelectCommand = command1;
    adapter1.Fill(table1);

    if (table1.Rows.Count > 0)
    {
        OleDbDataReader reader1 = command1.ExecuteReader();

        while (reader1.Read())
        {
            comboBox1.Items.Add(reader1[0]);
        }
        reader1.Close();
    }
}
private void panel1_Click(object sender, EventArgs e)
{
}
private void pictureBox1_Click(object sender, EventArgs e)
{
    panel1.BackColor = Color.Gray;
    if (MessageBox.Show("Видалити обраний склад в таблиці?",
"Підтвердження дії", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button1) == DialogResult.Yes)
    {

```

```

        OleDbCommand command = new OleDbCommand("DELETE
FROM Sklad WHERE id_sklad = @a", myConnection);
        command.Parameters.Add("@a", OleDbType.Integer).Value =
dataGridView1[0, dataGridView1.CurrentRow.Index].Value;
        command.ExecuteNonQuery();
        FillGrid();
    }
}

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    panel1.BackColor = Color.Red;
}

private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    panel1.BackColor = Color.Gray;
}

private void pictureBox3_MouseDown(object sender, MouseEventArgs e)
{
    panel2.BackColor = Color.Red;
}

private void pictureBox3_MouseUp(object sender, MouseEventArgs e)
{
    panel2.BackColor = Color.Gray;
}

private void pictureBox3_Click(object sender, EventArgs e)
{
    label3.Text = dataGridView1[0,
dataGridView1.CurrentRow.Index].Value.ToString();
    panel2.BackColor = Color.Gray;
    button1.Text = "Зберегти зміни";
    textBox1.Text = dataGridView1[1,
dataGridView1.CurrentRow.Index].Value.ToString();
    comboBox1.Text = dataGridView1[2,
dataGridView1.CurrentRow.Index].Value.ToString();
    f = true;
}

private void dataGridView1_CurrentCellChanged(object sender,
EventArgs e)
{

```

```

if(f)
{
    button1.Text = "Додати склад";
    textBox1.Text = "";
    comboBox1.Text = "";
    //f = false;
}
}

private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text != "" && comboBox1.Text != "")
    {

        string NAME_S = textBox1.Text;
        label4.Text = textBox1.Text;

        DataTable table = new DataTable();
        OleDbDataAdapter adapter = new OleDbDataAdapter();
        OleDbCommand command = new OleDbCommand("SELECT
id_user FROM Users WHERE ULogin = " + comboBox1.Text + "",
myConnection);
        adapter.SelectCommand = command;
        adapter.Fill(table);

        dataGridView1.Rows.Clear();
        string k = "10";
        if (table.Rows.Count > 0)
        {
            OleDbDataReader reader = command.ExecuteReader();

            while (reader.Read())
            {
                k = reader[0].ToString();
            }
            reader.Close();
        }

        if (f == false)
        {
            OleDbCommand command1 = new OleDbCommand("INSERT
INTO Sklad (SName, id_zav_sklad) VALUES (" + textBox1.Text + ", " + k +
")", myConnection);
            command1.ExecuteNonQuery();
            FillGrid();
        }
    }
}

```

```
        MessageBox.Show("Успішно!", "Дія закінчена");
        textBox1.Text = "";
        comboBox1.Text = "";
    }
    else
    {
        int kkk = Convert.ToInt32(label3.Text);
        int iii = Convert.ToInt32(k);
        // NAME_S = textBox1.Text;
        label3.Text = textBox1.Text;
        command = new OleDbCommand("UPDATE Sklad SET SName =
" + NAME_S + ", id_zav_sklad = "+iii+" WHERE id_sklad = "+kkk+",
myConnection);
        // command.Parameters.Add("@a", OleDbType.Integer).Value =
Convert.ToInt32(label3.Text);
        // command.Parameters.Add("@b", OleDbType.Integer).Value =
Convert.ToInt32(k);
        command.ExecuteNonQuery();
        FillGrid();
        MessageBox.Show("Успішно!", "Дія закінчена");
        f = false;
    }

}

}
else
    MessageBox.Show("Заповніть всі поля!", "Помилка");
}
}
}
```

Додаток 2

«РОЗРОБКА АВТОМАТИЗОВАНОГО ДОДАТКУ ОБЛІКУ ТОВАРІВ НА СКЛАДІ»

Виконав студент: Пивовар А.О.
Керівник: к.т.н., доц. Горда О. В.

ВСТУП

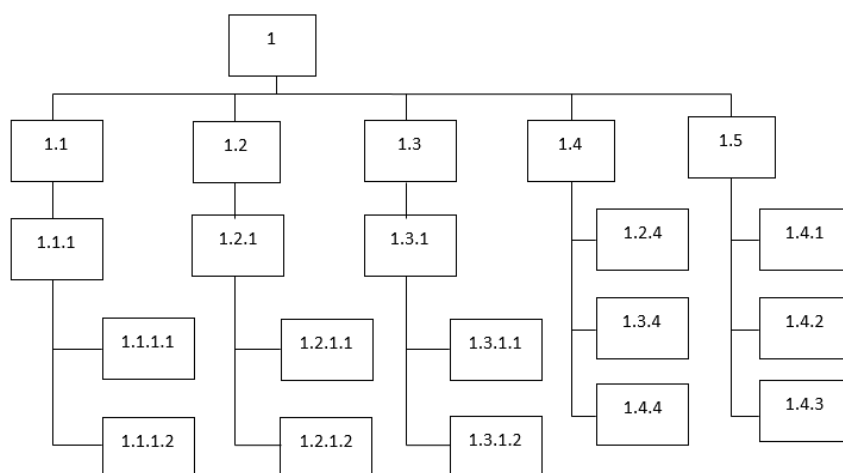
Актуальність дослідження зумовлена необхідністю ефективного управління запасами та обліку товарів на складі, що є критичним для зменшення витрат та підвищення продуктивності підприємства. В умовах зростання обсягів торгівлі та логістичних операцій, автоматизація процесів обліку товарів на складі стає особливо важливою.

Мета дослідження – розробити інформаційне забезпечення для автоматизованого додатку обліку товарів на складі, що забезпечить точний та своєчасний облік товарів, оптимізацію складських процесів та зменшення витрат на їхнє управління.

Об'єктом дослідження є автоматизована система обліку товарів на складі, а предметом дослідження – структура та зміст бази даних цієї системи.

Методика включає аналіз літературних джерел, визначення вимог до системи, проектування бази даних, реалізацію програмного забезпечення, тестування та впровадження автоматизованої системи обліку товарів на складі.

Практична значимість полягає в можливості розробки ефективної системи управління запасами, яка сприятиме зниженню витрат, покращенню обслуговування клієнтів, оптимізації складських процесів та підвищенню загальної ефективності підприємства.



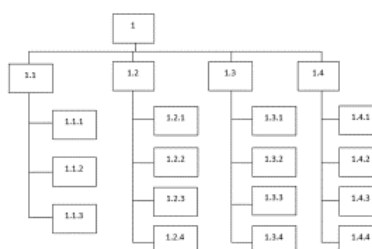
Порівняння розробок автоматизованих систем обліку товарів на складі

Назва системи	Країна виробник	Скільки років на ринку	Особливості
SAP Warehouse Management	Німеччина	Понад 20 років	Потужна інтеграція з іншими модулями SAP, високий рівень автоматизації складських процесів.
Oracle Warehouse Management	США	Понад 20 років	Висока масштабованість, підтримка хмарних рішень, інтеграція з ERP Oracle.
Microsoft Dynamics 365 for Finance and Operations	США	Понад 15 років	Гнучкість налаштувань, інтеграція з іншими продуктами Microsoft, підтримка хмарних рішень.
Infor Warehouse Management	США	Понад 20 років	Потужні аналітичні інструменти, модульна структура, інтеграція з іншими продуктами Infor.

Порівняння баз даних

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
Oracle Database	Relational	SQL	Windows, Linux, macOS	Підтримка великих обсягів даних, можливість використовувати більшість функцій SQL
MySQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, гарна підтримка індексів
MongoDB	NoSQL	MongoDB	Windows, Linux, macOS	Гнучкість у роботі зі структурованими та неструктурованими даними
PostgreSQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, можливість використовувати різні типи даних
MS Access	Relational	SQL	Windows	Легкість використання для малих та середніх проектів, інтеграція з Microsoft Office, не потребує окремого сервера, зручний інтерфейс для користувачів, обмежена масштабованість

Дерево функцій



Вибір типу бази даних

Для розробки автоматизованого додатку обліку товарів на складі було обрано реляційну базу даних.

Реляційні бази даних зберігають дані у вигляді таблиць, де кожна таблиця складається з рядків та стовпців. Рядки представляють окремі записи, а стовпці — атрибути або поля записів. Така структура дозволяє організувати дані у вигляді взаємопов'язаних таблиць, що забезпечує ефективне управління даними та їх обробку.

Переваги:

- **Структурованість:** Дані зберігаються в чітко визначених таблицях, що полегшує їх організацію та доступ.
- **Цілісність даних:** Використання ключів та обмежень забезпечує цілісність даних, унеможливаючи дублювання або втрату інформації. Швидкий доступ до даних
- **Підтримка транзакцій:** Забезпечує надійність виконання операцій, дозволяючи виконувати їх повністю або відмінати при виникненні помилок.
- **Запити на основі SQL:** Мова SQL дозволяє виконувати складні запити для вибірки, оновлення та видалення даних.

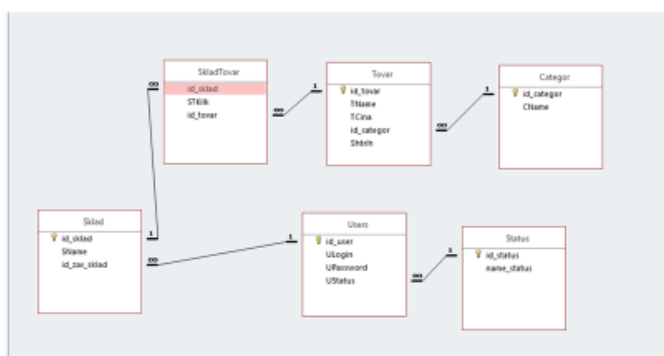
Вибір СУБД

Для проекту був обраний MS Access через його інтеграцію з Microsoft Office, зручний інтерфейс для створення та управління базами даних, а також можливості для швидкого прототипування та розробки додатків. MS Access забезпечує високу продуктивність для невеликих і середніх обсягів даних, що робить його оптимальним вибором для розробки автоматизованого додатку обліку товарів на складі.

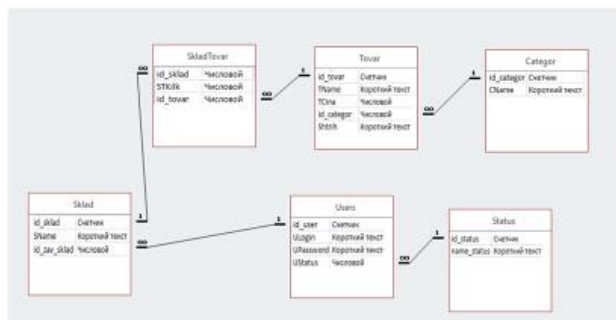
MS Access є настільною СУБД, яка зберігає всі дані у файлі формату .accdb. Це робить її зручною для локального використання та розгортання на одному комп'ютері або в невеликій мережі. Вибір MS Access забезпечує легкість налаштування та управління базою даних без необхідності налаштовувати складну інфраструктуру або хмарні сервіси.



Логічна модель представлення



Фізична модель представлення



Вибір програмного інструментарію

Для розробки автоматизованого додатку обліку товарів на складі я обрав середовище розробки **Visual Studio** та мову програмування **C#**

Переваги вибору Visual Studio та C#:

- **Інтуїтивно зрозумілий інтерфейс:** Visual Studio надає зручний інтерфейс з численними інструментами для автоматизації процесів розробки.
- **Підтримка інтеграції з базами даних:** Visual Studio легко інтегрується з MS Access та іншими реляційними базами даних, що спрощує роботу з даними.
- **Ефективні інструменти налагодження:** Visual Studio надає потужні інструменти для налагодження та тестування коду, що підвищує продуктивність розробки.

Приклад взаємодії з базою даних

Visual Studio

```

private void button1_Click(object sender, EventArgs e)
{
    int id_sklad = 1;
    if (MessageBox.Show("Вибрати склад") == DialogResult.No)
    {
        return;
    }
    string sql = "SELECT TOP 10 * FROM Sklad WHERE id_sklad = @id_sklad";
    OleDbCommand cmd = new OleDbCommand(sql, conn);
    cmd.Parameters.AddWithValue("@id_sklad", id_sklad);
    OleDbDataReader reader = cmd.ExecuteReader();
    FillDataGridView();
}
    
```

Програмний продукт

Назва складу:

Вибір складу:

Назва	Відповідаєльний
Склад №1	ZavSklad1
Склад №2	Admin
Склад №2	Tovarivod

MS Access

id_sklad	SName	id_zav_sklad
1	Склад №1	0
14	Склад №2	4
16	Склад №2	5

Висновки

У результаті виконання даного дипломного проекту розроблено інформаційне забезпечення автоматизованої системи обліку товарів на складі. Для досягнення цієї мети було проведено аналіз предметної області, проектування задачі, практичну реалізацію та розробку ергономічної складової.

Результатом роботи є функціональне інформаційне забезпечення, яке задовольняє визначені цілі та вимоги. Розроблена система може стати основою для подальшого розвитку та вдосконалення систем обліку товарів на складі, сприяючи зручності, безпеці та ефективності управління складськими процесами. Вона має потенціал для впровадження у різні бізнес-середовища, що може значно покращити процеси обліку, управління запасами та логістики.