

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»**

Кафедра інформаційних систем в економіці

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Інформаційні управляючі системи і технології»**

галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки

Форма навчання: очна (денна)

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему: «Розроблення інформаційної системи для управління матеріальними ресурсами компанії»

здобувача Ярцева Семена Дмитровича
(ПІБ, підпис)

Науковий керівник: Галузинський Георгій Петрович
(науковий ступінь, учене звання, ПІБ)

(підпис)

**Робота допущена до захисту перед екзаменаційною комісією з атестації
здобувачів вищої освіти (ЕК)**

Завідувач кафедри: _____
(науковий ступінь, учене звання, ПІБ)

(підпис)

Київ 2025

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА

«Інформаційні управляючі системи і технології»

галузь знань

12 Інформаційні технології

спеціальність

122 Комп'ютерні науки

ПОГОДЖЕНО:

Керівник проєктної групи (гарант)
освітньо-професійної програми

ЗАТВЕРДЖУЮ:

Завідувач кафедри інформаційних
систем в економіці

_____ Устенко С.В.
« ____ » _____ 202_ р.

_____ Тішков Б.О.
« ____ » _____ 202_ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Ярцева Семена Дмитровича

(прізвище, ім'я, по батькові)

_____ очної (денної) _____ форми навчання

очної (денної), заочної, дистанційної

на підготовку кваліфікаційної магістерської роботи

на тему: «Розроблення інформаційної системи для управління матеріальними ресурсами компанії»

Тему затверджено наказом ректора Університету від « ____ » _____ 202_ р. № _____

Кваліфікаційна магістерська робота виконується на матеріалах

Інтернет джерел

План кваліфікаційної магістерської роботи

Розділ I Дослідження та аналіз підходів до створення інформаційних управляючих систем предметної області

(назва розділу)

Розділ II Характеристика інформаційних управляючих систем та методи і моделі

(назва розділу)

Розділ III Розроблення проєктних рішень та їх реалізація

(назва розділу)

Об'єкт дослідження: Процеси автоматизованої обробки та обліку матеріальних ресурсів

Предмет дослідження: Сукупність теоретичних, методичних і практичних підходів до технології проєктування системи для управління матеріальними ресурсами компанії

Мета кваліфікаційної магістерської роботи: Вдосконалення процесу управління матеріальними ресурсами компанії шляхом проєктування інформаційної системи

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I Провести аналіз актуальності теми розробки веб-платформи для управління матеріальними ресурсами компанії на основі статистичних даних. Вивчити основні процеси у сфері управління ресурсами, організаційну структуру, ключові поняття, законодавчі норми та види інформаційних систем для управління матеріальними ресурсами. Провести огляд та порівняльний аналіз існуючих платформ і сервісів. Визначити перспективи розвитку майбутньої системи. Сформулювати початковий стек технологій та функціоналу

У розділі II Визначити функціональні та нефункціональні вимоги до системи. Розробити бізнес-процеси основних сценаріїв користування платформою. Побудувати діаграми процесів і класів, які відображають взаємодію користувачів із системою та структуру даних. Визначити методології які можна задіяти в платформі

У розділі III Створити інфологічну модель бази даних, адаптовану до функціональних вимог системи. Обґрунтувати вибір бази даних та розробити фізичну модель. Оцінити обсяги зберігання даних та їх структуру. Розробити архітектуру програмного забезпечення, врахувати питання безпеки. Провести тестування системи. Представити результати розробки, функціонал та користувацький інтерфейс

**Завдання підготував
науковий керівник**

(підпис)

(ініціали, прізвище)

« ____ » _____ 202_ р.

**Завдання одержав
здобувач**

(підпис)

(ініціали, прізвище)

« ____ » _____ 202_ р.

Реферат

Кваліфікаційна магістерська робота містить 77 сторінок, 3 таблиць, 20 рисунків, список використаних джерел з 50 найменувань, додатки.

«Проектування інформаційної системи управління матеріальними ресурсами компанії»

Об'єктом дослідження кваліфікаційної магістерської роботи є процеси автоматизованої обробки та обліку матеріальних ресурсів

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів до технології проектування системи для управління матеріальним ресурсами компанії.

Мета і завдання дослідження. Основною метою кваліфікаційної магістерської роботи є вдосконалення процесу управління матеріальним ресурсами компанії шляхом проектування інформаційної системи.

Відповідно до поставленої мети визначені такі *завдання*:

- дослідити предметну область, виконати аналіз існуючих систем;
- представити обґрунтування вибору підходів і технологій для створення системи;
- визначити структуру та представити характеристику системи;
- описати методи та моделі в системі;
- спроектувати базу даних для системи;
- описати програмне забезпечення для системи;
- описати технічне забезпечення для системи;
- представити результати реалізації.

Теоретична, методична та практична значущість отриманих результатів. Під час дослідження було досліджено предметну область, схожі програмні продукти, варіанти доступу та формати ведення даних, визначено перспективність удосконалення процесу взаємодії з контрагентами за допомогою розробки спеціалізованого програмного забезпечення. Практичні результати дослідження полягають у вдосконаленні процедури управління матеріальними ресурсами, що призведе до підвищення ефективності роботи з клієнтами та якості взаємодії з контрагентами.

Рік виконання кваліфікаційної магістерської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: система, ресурси, облік, обробка, інформаційна система.

Відгук
про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту «Інститут інформаційних технологій в економіці»
освітньо-професійної програми «Інформаційні управляючі системи і технології»
Ярцева Семена Дмитровича
на тему «Розроблення інформаційної системи для управління матеріальними ресурсами компанії»

1. Актуальність теми: дослідження визначається високим рівнем розроблення веб-платформи для управління матеріальними ресурсами компанії, розробленням архітектури програмного забезпечення.

2. Позитивні риси кваліфікаційної магістерської роботи: Реалізована інформаційна система успішно інтегрує сучасні технологічні рішення та продуману бізнес-логіку для підтримки діяльності, створюючи комфортне, безпечне та прозоре середовище. Платформа забезпечує автоматизацію ключових процесів, підтримує масштабування й подальше вдосконалення функціоналу, що дозволяє ефективно реагувати на виклики важливих ініціатив та сприяти розвитку.

3. Наявність і цінність самостійних розробок автора: пов'язана з розробкою інформаційної управляючої системи, що базується на продуманій інфологічній та фізичній моделі бази даних, які відображають ключові бізнес-процеси платформи. Використання SQLite забезпечує надійність, масштабованість та продуктивність, що підтверджено оцінкою прогнозних обсягів даних. Реалізована базова логіка інтелектуального аналізу у вигляді автоматизованих правил дозволяє ефективно обробляти інформацію, залишаючи можливість для подальшого розвитку інтелектуальних сервісів.

4. Рекомендація: Розроблена програма як частина платформи інформаційної системи може бути впроваджена для діяльності компаній різних напрямів.

5. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК: є завершеним проєктом, що поєднує теоретичну новизну, практичну значущість і відповідає вимогам ОПП «ІУСТ» спеціальності F3 «Комп'ютерні науки». Під час тестування програми платформи перевірено її працездатність, зручність користувацьких сценаріїв, коректність обробки запитів, відповідність інтерфейсу очікуванням користувачів. КМР виконана високому професійному рівні, а її автор заслуговує на високу оцінку.

Науковий керівник д.е.н., доцент Г.П. Галузинський 16.05.2025

аналітики на базі штучного інтелекту як перспективного напрямку подальшого розвитку.

Практична значимість висновків і рекомендацій. Запропонована інформаційна система довела свою ефективність під час апробації в умовах, наближених до виробничих. Вона здатна значно підвищити точність обліку, знизити витрати на логістику, покращити прогнозування і прийняття рішень. Система може бути легко адаптована до потреб інших підприємств, що свідчить про її універсальність і практичну цінність.

Старший науковий співробітник
Інституту кібернетики
імені В.М. Глушкова НАН України
канд. фіз.-мат. наук

(підпис, ПІБ)

Ніколаєвська О.А.

Підпис засвідчую:

Зав. секцією
(посада, підпис)

М.П.



Перелік умовних скорочень	9
ВСТУП	10
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.1 Аналіз існуючих інформаційних управляючих систем	12
1.1.1 Дослідження предметної області	12
1.1.2 Дослідження інформаційних управляючих систем	21
1.2 Обґрунтування вибору підходів і технологій для створення інформаційної управляючої системи	25
РОЗДІЛ 2 ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ТА МЕТОДИ І МОДЕЛІ	31
2.1 Структура і характеристика системи	31
2.2 Методи та моделі в інформаційних управляючих системах і технологіях	40
РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ	43
3.1 Проектування бази даних та/або сховища даних для інформаційної управляючої системи	43
3.1.1 Проектування інфологічної моделі БД	43
3.1.2 Обґрунтування вибору СКБД та проектування фізичної моделі БД в середовищі вибраної СКБД	45
3.1.3 Контрольний приклад та обрахунок прогнозних обсягів БД	48
3.2 Проектування бази знань інформаційної управляючої системи та/або засоби інтелектуального аналізу даних	50
3.2.1 Реалізація інтелектуального аналізу великих даних з метою наповнення бази знань	50
3.2.2 Проектування та реалізація бази знань інформаційної управляючої системи ...	52
3.3. Програмне забезпечення	56
3.3.1 Проектування програмного забезпечення	56
3.3.2 Вибір архітектури програмного забезпечення	62
3.3.3 Інструментальні засоби розробки для створення прикладного програмного забезпечення інформаційних управляючих систем	65
3.3.4 Тестування програмного забезпечення	67
3.3.5 Керівництво (інструкція) користувача	68
3.4. Технічне забезпечення	69
3.4.1 Обґрунтування вибору технічного забезпечення	69
3.4.2 Ресурсні вимоги до технічного забезпечення	73
3.5 Реалізація інформаційної управляючої системи	75
ВИСНОВКИ	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	83
ДОДАТКИ	87

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CSS	Cascading Style Sheets
HTML	HyperText Markup Language
MVC	Model View Controller
RAD	Rapid Application Development
DDoS	Distributed Denial of Service
URL	Uniform Resource Locator
SQL	Structured Query Language
БД	База даних
СУБД	Система управління базами даних

ВСТУП

Актуальність дипломної роботи полягає у тому, що компанії зацікавлені у розробці своєчасних і ефективних рішень щодо управління матеріальними ресурсами з використання різноманітних аналітичних веб-сервісів. Оскільки бізнес-процеси на підприємстві постійно змінюються, то виникає задача у постійному зборі та аналізі даних. Підприємства зацікавлені у розробці нових і/або удосконаленні існуючих інформаційних систем з управління матеріальних ресурсів

Аналіз останніх досліджень і публікацій

Далі необхідно визначити *наукову проблему* та сформулювати її суть. У вступі також бажано вказати відомості про наукове використання результатів досліджень або рекомендації щодо їх використання, визначити *особистий внесок автора*.

Мета і завдання дослідження. Основною метою кваліфікаційної магістерської роботи є підвищення ефективності процесів управління ІТ інфраструктурою АТ "Київгаз" шляхом розроблення та використання

інформаційної управляючої системи сервісної підтримки ІТ інфраструктури

Відповідно до поставленої мети визначені такі завдання:

- дослідити предметну область, виконати аналіз існуючих процесів;
- обґрунтувати підходи і технологій для створення системи;
- визначити структуру та представити характеристику системи;
- описати методи та моделі в системі;
- спроектувати базу даних;
- описати програмне забезпечення для інформаційної системи;
- описати технічне забезпечення для інформаційної системи;
- представити результати реалізації.

Об'єктом дослідження кваліфікаційної магістерської роботи є інформаційна система для управління матеріальними ресурсами компанії.

Предметом дослідження є методи та прийоми проектування інформаційної системи для управління матеріальними ресурсами компанії.

Теоретична, методична та практична значущість отриманих результатів. Під час дослідження було досліджено предметну область, проаналізовано існуючі рішення, визначені методи і моделі для підвищення ефективності управління матеріальними ресурсами, спроектована база даних, визначені показники ефективності. Практичні результати дослідження полягають у вдосконаленні процесів управління матеріальними ресурсами шляхом розроблення та використання інформаційної управляючої системи.

Інформаційна база дослідження відображає узагальнений перелік джерел інформації (праці вітчизняних та зарубіжних науковців і практиків, статистичні дані офіційних сайтів, звіти рейтингових агентств, фінансові звіти тощо)

Загальний обсяг роботи становить 77 сторінок друкованого тексту, 20 рисунків, 3 таблиць, 1 додаток . Список використаних джерел налічує 50 найменувань.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз існуючих інформаційних управляючих систем

1.1.1 Дослідження предметної області

Сучасний етап розвитку економіки супроводжується глибокою цифровою трансформацією бізнесу, що обумовлює зростаючу роль інформаційних систем у забезпеченні ефективної діяльності підприємств. В умовах глобалізації, жорсткої конкуренції та високої динаміки ринку особливого значення набуває раціональне управління матеріальними ресурсами — як одного з ключових компонентів у забезпеченні безперервності виробництва, оптимізації витрат і підвищенні операційної ефективності.

Неефективне управління матеріальними ресурсами призводить до затримок у постачанні, перевитрат, надлишкових запасів або дефіциту необхідних матеріалів, що безпосередньо впливає на якість продукції чи послуг, фінансові результати та конкурентоспроможність підприємства. Саме тому виникає потреба в побудові сучасної, гнучкої та інтегрованої інформаційної системи, яка забезпечить повний контроль над усіма процесами, пов'язаними з обігом матеріальних ресурсів.

Метою цієї роботи є створення ефективної інформаційної системи для автоматизації процесів управління матеріальними ресурсами підприємства, що охоплює облік, зберігання, внутрішнє переміщення, контроль залишків, планування закупівель та генерацію звітності. Запропонована система повинна забезпечувати прозорий та оперативний облік матеріальних ресурсів, підтримку прийняття управлінських рішень на основі аналітичних даних, а також інтеграцію з іншими інформаційними середовищами компанії.

Для досягнення поставленої мети в рамках проєкту передбачається виконати низку послідовних дій. Насамперед буде здійснено аналіз сучасних підходів до автоматизації процесів управління матеріальними ресурсами, з метою виявлення переваг і недоліків існуючих рішень. Далі відбудеться дослідження

архітектурних моделей, зокрема клієнт-серверної архітектури, та обґрунтований вибір найбільш доцільної для розробки системи. Наступним етапом стане проєктування реляційної бази даних, що забезпечить надійне зберігання й структуровану організацію інформації про запаси, постачання, внутрішні запити та облік руху матеріалів. Після цього планується реалізація програмного забезпечення. Крім того, буде розроблено функціонал для формування аналітичної звітності та реалізовано інструменти прогнозування потреб у матеріальних ресурсах, що базуються на аналізі історичних даних. Особливу увагу буде приділено питанням інформаційної безпеки, цілісності даних та масштабованості системи. Важливим аспектом є також розробка і впровадження технічного забезпечення, яке забезпечуватиме надійну роботу системи у виробничому середовищі. Це охоплює вибір серверного та мережевого обладнання, налаштування середовища розгортання та організацію резервного копіювання.

На заключному етапі реалізації проєкту буде проведено всебічне тестування інформаційної системи, включаючи функціональну, навантажувальну та інтеграційну перевірку. Крім того, планується навчання персоналу, що забезпечить успішну адаптацію нової системи у виробничій практиці та мінімізацію перехідних витрат.

Очікуваним результатом впровадження системи є підвищення ефективності управління матеріальними ресурсами за рахунок автоматизації рутинних процесів, зниження обсягів непродуктивних витрат, забезпечення точного контролю за запасами та оперативного прийняття рішень на основі достовірних даних.

Таким чином, реалізація запропонованої інформаційної системи створить основу для цифрової трансформації управлінських процесів, підвищить конкурентоспроможність підприємства та сприятиме його стабільному розвитку в умовах ринкової нестабільності та технологічних змін.

Аналіз існуючих інформаційних систем управління матеріальними ресурсами є одним із ключових етапів у процесі проєктування власного рішення, орієнтованого на потреби конкретного підприємства. Вивчення доступних рішень дозволяє не лише ознайомитися з їхніми функціональними можливостями, але й

критично оцінити їх сильні та слабкі сторони, виявити типові помилки реалізації, визначити рівень відповідності сучасним технологічним вимогам, а також закласти основу для формування вимог до майбутньої системи. Такий підхід дає змогу уникнути дублювання недоліків, покращити ергономіку, підвищити продуктивність розробки та забезпечити реальну цінність ІС для бізнесу.

Інформаційні системи управління матеріальними ресурсами охоплюють комплекс програмно-технічних засобів, що забезпечують автоматизацію процесів обліку, контролю, аналізу та планування ресурсів на всіх етапах їх життєвого циклу. Їх основними функціями є: точне ведення залишків, своєчасне оновлення даних про запаси, підтримка процесів закупівлі, розподілу матеріалів між підрозділами, інвентаризації та формування звітності. Вивчення досвіду застосування таких систем у різних галузях дає можливість оцінити ефективність реалізованих функцій, їхню масштабованість, адаптивність до змін у діяльності підприємства та ступінь інтеграції з іншими інформаційними компонентами.

Таким чином, аналіз сучасних ІС управління матеріальними ресурсами дозволяє зробити обґрунтований вибір архітектурних рішень, функціонального наповнення та технологічного стеку, що максимально відповідають потребам і можливостям підприємства, яке впроваджує цифрову систему управління ресурсами.

Управління матеріальними ресурсами є складовою частиною системи загального управління підприємством, що охоплює планування, облік, контроль, аналіз і оптимізацію процесів забезпечення, використання та зберігання матеріально-технічних цінностей. Раціональне управління матеріальними потоками має вирішальне значення для ефективності діяльності організації, оскільки значна частина витрат припадає саме на ресурси: сировину, матеріали, комплектуючі, запчастини, інструменти, витратні засоби тощо.

Загальні підходи до управління матеріальними ресурсами формувалися на основі класичних теорій логістики, виробничого планування та управління запасами. У традиційній моделі облік матеріалів здійснювався вручну або у вигляді децентралізованих локальних систем, що часто призводило до неузгодженості

інформації, дублювання даних, перевитрат або дефіциту ресурсів. Сучасні підходи ґрунтуються на автоматизації та цифровізації всього ланцюга управління матеріальними ресурсами, починаючи від закупівель і до моменту списання або повторного використання.

Одним із найбільш поширених методів у цій сфері є система управління запасами, яка передбачає облік залишків, контроль критичних рівнів, ведення карток руху матеріалів та інвентаризацій. До ефективних методик належать FIFO (first-in-first-out), LIFO (last-in-first-out), JIT (just-in-time) та ABC/XYZ-аналіз. Застосування цих підходів у межах інформаційної системи дозволяє адаптивно регулювати рівень запасів залежно від реального споживання та потреб виробництва, мінімізуючи витрати на зберігання й уникнення надлишків.

Сучасні підприємства дедалі частіше впроваджують інтегровані інформаційні системи для управління матеріальними ресурсами, які включають модулі з планування потреб, обліку надходжень і витрат, розподілу ресурсів між структурними підрозділами, моніторингу залишків у реальному часі, аналітики та формування звітності. Ці системи можуть бути частиною корпоративного ERP-рішення (наприклад, SAP, Oracle ERP, Microsoft Dynamics) або розроблені індивідуально з урахуванням специфіки конкретного підприємства.

На підприємствах малого та середнього бізнесу часто використовуються легкі інформаційні системи власної розробки або відкриті рішення з обмеженою функціональністю. Їх основною перевагою є адаптивність, простота обслуговування та можливість налаштування під конкретні бізнес-процеси. В таких системах зазвичай реалізовані основні функції обліку: ведення номенклатури матеріалів, створення складських документів, генерація залишків та облік переміщень.

Технологічно сучасні системи управління матеріальними ресурсами реалізуються на основі клієнт-серверної архітектури, що дозволяє розподілити обробку даних між сервером, який виконує основну бізнес-логіку і керує БД, та клієнтськими додатками, які забезпечують взаємодію з користувачами. Для зберігання даних застосовуються реляційні СУБД (наприклад, PostgreSQL,

MySQL, SQLite), що забезпечують надійність, консистентність та швидкий доступ до великого обсягу інформації.

Ключові складові будь-якої ІС управління матеріальними ресурсам:

- блок обліку матеріалів, який фіксує надходження, видачу, залишки, списання;
- блок закупівель, який забезпечує автоматизацію замовлень, ведення даних постачальників, контроль виконання контрактів;
- аналітичний модуль, що дозволяє оцінювати ефективність використання ресурсів, проводити прогнозування та оптимізацію запасів;
- модуль інтеграції, який забезпечує взаємодію з бухгалтерськими, фінансовими та виробничими підсистемами.

Такі системи також часто мають рольову модель доступу, що передбачає розмежування прав користувачів залежно від їхньої функціональної ролі — наприклад, комірник, аналітик, керівник складу, адміністратор. Це дозволяє підвищити безпеку системи та запобігти несанкціонованим змінам у даних.

У межах цифрової трансформації бізнесу дедалі актуальнішим стає використання інструментів інтелектуального аналізу даних (BI). Такі інструменти, як Power BI, Tableau або власні візуалізаційні модулі, дають змогу в реальному часі формувати дашборди з інформацією про стан запасів, виконання замовлень, динаміку витрат і рівень використання матеріалів, що значно підвищує якість управлінських рішень.

Загальні підходи до управління матеріальними ресурсами підприємства сьогодні базуються на інтеграції перевірених логістичних практик із сучасними ІТ-рішеннями. Інформаційні системи виконують роль не лише інструменту обліку, а й інтелектуального середовища для аналітики, контролю та прогнозування. Їх впровадження дозволяє підприємствам підвищити прозорість ресурсних потоків, знизити витрати, уникнути дефіциту або надлишкових закупівель і, як наслідок, зміцнити позиції на ринку в умовах високої конкуренції.

Основні типи інформаційних систем управління матеріальними ресурсами.

Локальні облікові системи. Прості програми для обліку ресурсів,

встановлені на одному робочому місці. Використовуються малими підприємствами або для обмежених задач (наприклад, облік на одному складі).

Корпоративні ERP-системи. Інтегровані програмні платформи, що охоплюють усі функціональні підрозділи підприємства, включаючи матеріально-технічне забезпечення, фінанси, бухгалтерію, виробництво та ін.

Системи класу MRP/MRP II (Material Requirements Planning). Спеціалізовані рішення для планування потреб у матеріалах. В основі — розрахунок графіків закупівель і виробництва.

Інформаційні системи власної розробки. ІС, створені спеціально під потреби конкретного підприємства. Мають адаптивну функціональність і можуть будуватись на сучасній клієнт-серверній архітектурі.

Хмарні системи обліку ресурсів. Веб-орієнтовані рішення, які не потребують локального встановлення. Доступні через браузер, працюють на основі SaaS (Software as a Service) (таблиця 1.1)

Таблиця 1.1 – Порівняльна таблиця типів інформаційних систем

Тип ІС	Переваги	Недоліки	Сфера застосування
Локальні облікові системи	Простота використання, швидке впровадження, невисока вартість	Обмежена масштабованість, немає інтеграції між підрозділами	Малі підприємства, облік на окремих складах
ERP-системи (SAP, 1C, Oracle)	Комплексне охоплення процесів, інтеграція між модулями, гнучкість	Висока вартість, складність впровадження, потреба у підготовці персоналу	Великі та середні підприємства з розгалуженою структурою
MRP/MRP II системи	Орієнтація на планування потреб, інтеграція з виробництвом	Вузька спеціалізація, необхідність інтеграції з іншими модулями	Виробничі підприємства з великою номенклатурою
ІС власної розробки	Повна адаптація до бізнес-процесів, контроль над логікою системи	Потреба у підтримці, часова тривалість розробки	Підприємства з унікальними або змінними процесами
Хмарні ІС	Швидкий доступ, відсутність витрат на інфраструктуру, мобільність	Залежність від інтернету, менший контроль над даними	Малі та середні підприємства, підприємства з віддаленим доступом
MRP/MRP II системи	Орієнтація на планування потреб, інтеграція з виробництвом	Вузька спеціалізація, необхідність інтеграції з іншими модулями	Виробничі підприємства з великою номенклатурою

Джерело: сформовано автором на основі виконаного дослідження

SAP S/4HANA — це високопродуктивна, модульна платформа корпоративного рівня, що забезпечує управління всіма основними ресурсами підприємства, включаючи матеріальні запаси, закупівлі, логістику, фінанси та виробництво. У модулі Materials Management (MM) реалізовано облік складів, контроль закупівель, планування постачання, управління ланцюгами поставок.

Особливості:

- підтримка роботи в реальному часі;
- глибока інтеграція з іншими модулями (фінанси, виробництво);
- потужна аналітика на основі in-memory обробки даних;
- висока вартість впровадження, але виняткова масштабованість.

Підходить для великих виробничих підприємств, транснаціональних корпорацій (рис. 1.1)

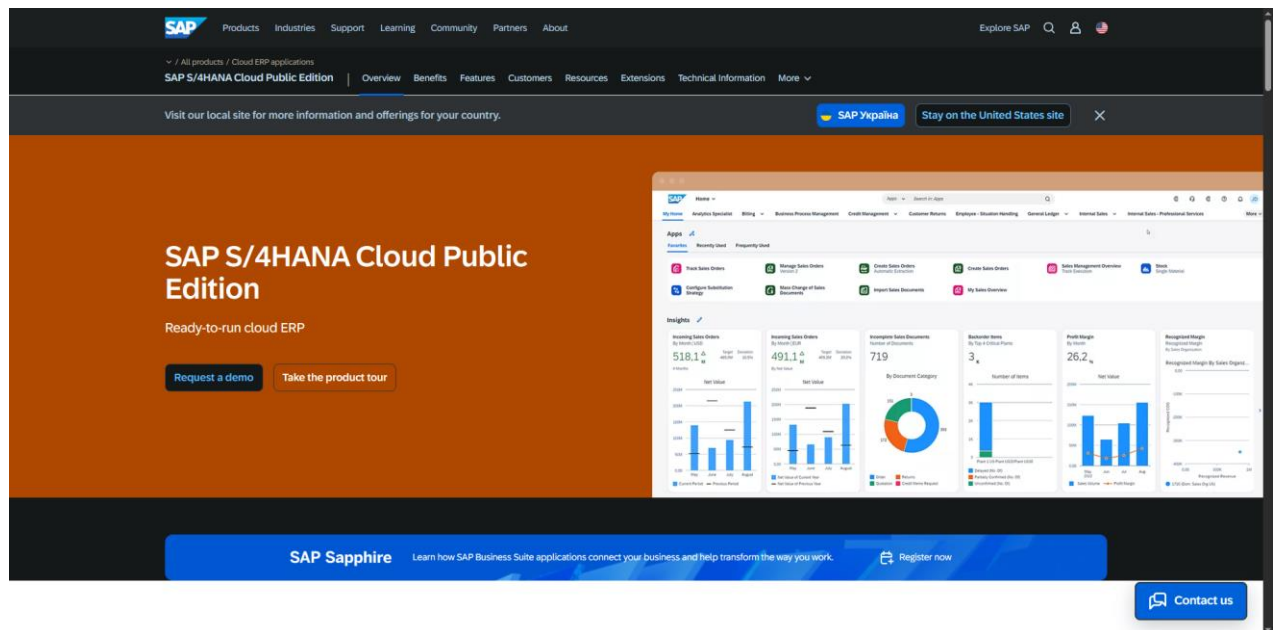


Рисунок 1.1 - SAP S/4HANA

Джерело: <https://www.sap.com/products/erp/s4hana.html>

Microsoft Dynamics 365 Supply Chain Management — є частиною великої екосистеми Microsoft і дозволяє керувати матеріальними потоками, складськими операціями, виробничими циклами, плануванням ресурсів. Вбудовані інструменти аналітики Power BI дають змогу оперативно візуалізувати та аналізувати дані.

Особливості:

- повна інтеграція з Microsoft 365 (Excel, Teams, Power Apps);
- підтримка мобільних пристроїв та хмарної роботи;
- гнучка модель ліцензування;
- зручний інтерфейс для користувачів без технічної підготовки.

Підходить для середніх і великих підприємств із потребою в інтеграції з офісною інфраструктурою (рис. 1.2)

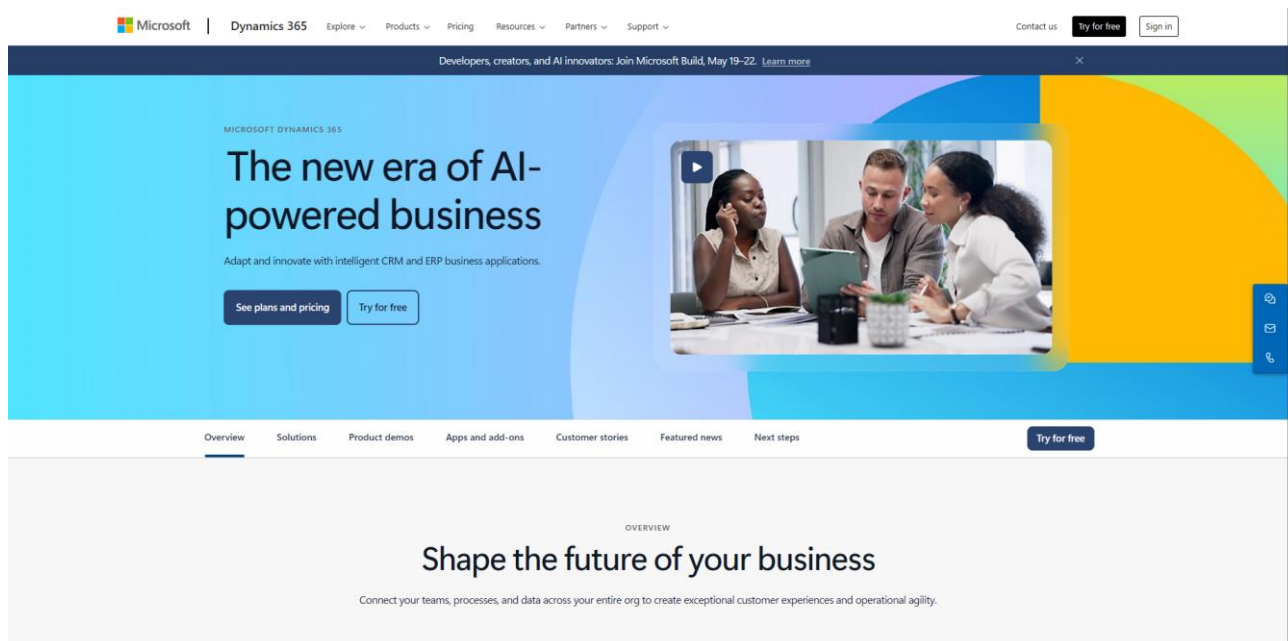


Рисунок 1.2 — Microsoft Dynamics

Джерело: <https://www.microsoft.com/en-us/dynamics-365>

Odoo — це модульна ERP-система, що підтримує управління запасами, закупівлями, обліком, CRM, виробництвом тощо. Модулі **Inventory** та **Purchase** дозволяють створити повноцінну систему управління матеріальними ресурсами із функціями обліку залишків, керування постачальниками, автоматичного поповнення складу, звітності.

Особливості:

- відкрита архітектура, можливість самостійної модифікації;
- працює в браузері, хмарно або локально;
- висока гнучкість при низькій вартості;
- велика спільнота розробників та доступ до тисяч модулів.

Підходить для середніх і великих підприємств із потребою в інтеграції з офісною інфраструктурою (рис. 1.3)

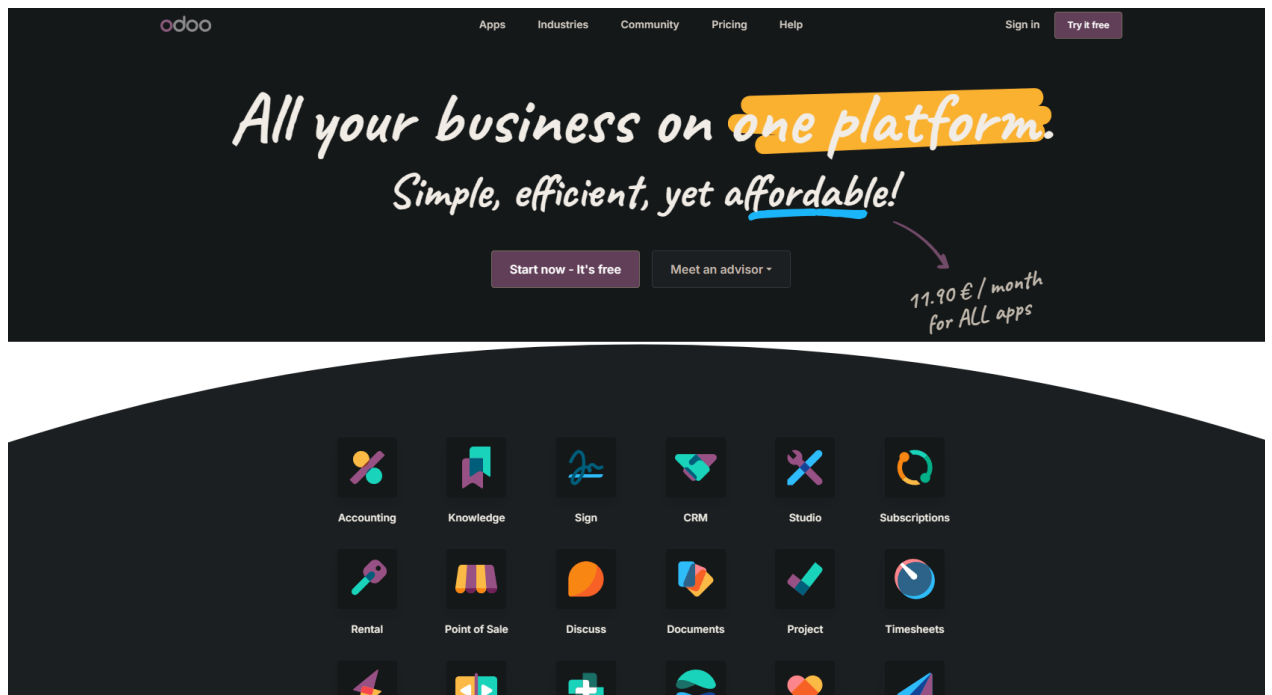


Рисунок 1.3 — Odoo

Джерело: <https://www.odoo.com/>

1.1.2 Дослідження інформаційних управляючих систем

Дослідження предметної області є першим кроком у процесі розробки інформаційної системи для управління матеріальними ресурсами компанії. Воно передбачає глибокий аналіз усіх аспектів діяльності підприємства, пов'язаних із закупівлею, обліком, зберіганням, переміщенням та використанням матеріальних цінностей. Розуміння предметної області є критично важливим для формування вимог до системи, виявлення ключових проблем у процесах управління ресурсами та побудови ефективного цифрового рішення.

Матеріальні ресурси займають центральне місце у виробничо-господарській діяльності будь-якої компанії, незалежно від галузі. До них можуть належати сировина, напівфабрикати, комплектуючі, допоміжні матеріали, інструменти, запасні частини, витратні ресурси тощо. Управління цими ресурсами включає весь

життєвий цикл — від моменту надходження на склад до повного використання або списання. Ефективність управління матеріальними ресурсами безпосередньо впливає на продуктивність, рентабельність та конкурентоспроможність підприємства.

Однією з ключових особливостей предметної області є необхідність точного, своєчасного й повного обліку матеріалів. Без якісного обліку виникають перевитрати, нестача, затримки у виробництві або простої. Крім того, деякі ресурси мають обмежений термін зберігання або вимагають особливих умов (температурний режим, вологість, захист від впливу зовнішнього середовища), що також має бути враховано при побудові інформаційної системи.

Система управління матеріальними ресурсами повинна забезпечувати автоматизований контроль над усіма операціями: надходження, внутрішнє переміщення, списання, повернення, інвентаризація. Одним з критично важливих аспектів є ведення обліку за партіями, серійними номерами, номенклатурними групами, що дає змогу ефективно відстежувати історію кожного ресурсу та виконувати аудит.

Також необхідно враховувати вимоги до аналітичної звітності. Керівництво компанії має отримувати актуальну інформацію про залишки, рівень використання ресурсів, динаміку витрат, а також прогноз потреб на основі історичних даних. Це дає змогу оптимізувати закупівлі, уникати надлишкових запасів та знижувати витрати.

У межах дослідження предметної області важливо проаналізувати бізнес-процеси, пов'язані з матеріальними потоками. Це включає в себе процеси закупівлі, приймання, внутрішньої логістики, зберігання, підготовки до виробництва та списання. Аналіз дозволяє виявити критичні точки, дублювання дій, залежності від людського фактору, які знижують загальну ефективність. Наприклад, ручне внесення даних про надходження часто призводить до помилок, а відсутність оперативного обліку в реальному часі — до інформаційного вакууму.

Важливим елементом є взаємодія з постачальниками та внутрішніми службами. Компанія повинна мати можливість автоматизовано створювати

замовлення постачальникам, контролювати виконання контрактів, вести облік рекламаций. З іншого боку, виробничі підрозділи мають оперативно отримувати інформацію про наявність ресурсів та терміни їх доставки. Все це вимагає гнучкого, інтегрованого підходу до побудови інформаційної системи.

Однією з основних цілей дослідження предметної області є виявлення існуючих проблем у процесах управління матеріальними ресурсами. До них можуть належати: відсутність єдиної бази даних, використання неузгоджених форматів обліку, недостатній рівень автоматизації, слабка інтеграція з виробничими системами. Визначення цих проблем дає змогу сформулювати конкретні вимоги до функціоналу майбутньої системи.

Таким чином, дослідження предметної області є ключовим етапом на шляху створення ефективної інформаційної системи управління матеріальними ресурсами. Воно охоплює аналіз складу ресурсів, бізнес-процесів, вимог до обліку, звітності, безпеки та аналітики, що дозволяє забезпечити максимальну відповідність розроблюваної системи потребам підприємства.

У сучасних умовах функціонування підприємств, незалежно від галузі, матеріальні ресурси залишаються одним із ключових чинників забезпечення безперервності виробничо-господарських процесів. До них належать різноманітні групи матеріалів — від сировини, комплектуючих та допоміжних засобів до витратних матеріалів, інструментів, запасних частин тощо. Управління цими ресурсами охоплює увесь життєвий цикл — починаючи з формування потреби, здійснення закупівлі, надходження на склад, зберігання, розподілу між підрозділами та завершуючи їх використанням, списанням або повторним введенням у господарський обіг.

Однією з головних вимог до системи управління матеріальними ресурсами є забезпечення точного, своєчасного і достовірного обліку кожної одиниці ресурсу. Неточності в обліку призводять до значних фінансових втрат, перевитрат, накопичення надлишкових залишків або, навпаки, нестачі критичних матеріалів у найвідповідальніші моменти. Особливо це стосується підприємств, що працюють

зі складним виробництвом або мають розгалужену структуру з кількома складами, відокремленими підрозділами та складною логістикою.

Важливо також враховувати, що частина матеріальних ресурсів має специфічні умови зберігання — температурний режим, рівень вологості, захист від механічного впливу або шкідливого середовища. У разі порушення таких умов виникають ризики втрат та псування, що робить критично необхідним впровадження засобів контролю й сповіщення в реальному часі. Тому при дослідженні предметної області слід приділити окрему увагу не лише економічним, а й технічним аспектам управління ресурсами.

Інформаційна система має забезпечувати автоматизований супровід усіх операцій з матеріальними ресурсами: надходження, розміщення на складах, внутрішнє переміщення, інвентаризація, списання, повернення залишків, облік втрат тощо. Для більшості підприємств доцільним є впровадження багаторівневого обліку: за партіями, серійними номерами, категоріями, кодами номенклатури та місцезнаходженням. Такий рівень деталізації дозволяє здійснювати повноцінне відстеження руху ресурсів у часовому та просторовому вимірах, виконувати аудит та формувати прозорі звіти для управлінських рішень.

Крім функціонального аспекту, при дослідженні предметної області слід враховувати й управлінські потреби підприємства. Керівництво зазвичай потребує гнучких інструментів для моніторингу стану залишків, контролю динаміки витрат, оцінки рівня використання ресурсів та прогнозування майбутніх потреб. Саме тому система має містити засоби аналітичної обробки даних, побудови візуалізацій, формування індивідуальних звітів, а також можливості інтеграції з іншими модулями або системами підприємства (наприклад, бухгалтерськими чи виробничими).

Значна увага у процесі дослідження предметної області приділяється аналізу внутрішніх бізнес-процесів підприємства, які стосуються закупівлі, логістики, обліку, контролю й використання матеріалів. На практиці часто виявляються проблеми дублювання даних, залежності від людського фактору, відсутності централізованого управління або розірваності інформаційного ланцюга між

службами. Наприклад, затримки з обробкою первинної документації або неузгодженість даних між відділами можуть знижувати ефективність усього підприємства. Тому важливо формалізувати та оптимізувати бізнес-процеси ще до початку технічної реалізації ІС.

Ще одним важливим аспектом дослідження є взаємодія підприємства із зовнішніми контрагентами — постачальниками, підрядниками, логістичними компаніями. Інформаційна система має враховувати можливість автоматизованого обміну документами, сповіщень про статус замовлень, інтеграції з електронними системами закупівель тощо. З іншого боку, всередині підприємства слід забезпечити чітку регламентацію доступу до даних для різних ролей — складських працівників, логістів, фінансистів, керівництва.

Під час дослідження предметної області зазвичай виявляються типові проблеми, характерні для більшості підприємств: фрагментованість обліку, залежність від ручної обробки, низький рівень прозорості руху ресурсів, застарілі технічні рішення, слабка інтеграція з іншими ІТ-системами. Визначення таких проблем є критично важливим, оскільки дозволяє сформувати цільовий образ системи, яка усуває наявні недоліки, підвищує ефективність управління та створює основу для цифрової трансформації логістичних і облікових процесів.

Отже, дослідження предметної області є стратегічною передумовою для успішної розробки інформаційної системи управління матеріальними ресурсами. Воно дозволяє не лише глибше зрозуміти специфіку функціонування підприємства, а й забезпечити максимальну адаптацію майбутнього рішення до реальних умов експлуатації. Комплексний підхід до аналізу ресурсної бази, внутрішніх процесів, інформаційних потоків, ролей користувачів і зовнішніх взаємозв'язків створює умови для розробки надійної, масштабованої та ефективної інформаційної системи, яка відповідатиме сучасним вимогам підприємств різного профілю.

1.2 Обґрунтування вибору підходів і технологій для створення інформаційної управляючої системи

Обґрунтування вибору підходів і технологій для створення інформаційної системи для управління матеріальними ресурсами компанії є критично важливим етапом, що визначає ефективність і довгостроковий успіх майбутньої системи. Розуміння специфічних вимог медичної галузі, потреб підприємства та аналіз існуючих рішень допомагає вибрати оптимальні підходи та технології, які найкраще відповідають цілям проекту.

Одним із основних завдань при виборі підходів є визначення архітектури системи. Клієнт-серверна архітектура є оптимальним рішенням для інформаційної системи, оскільки дозволяє розбити систему на окремі рівні, які можуть працювати незалежно один від одного. Це забезпечує високу гнучкість і масштабованість системи, дозволяючи легко додавати нові функції або змінювати існуючі без значних втручань у загальну структуру. Клієнт-серверна архітектура також спрощує підтримку та розширення системи, що є важливим для забезпечення її довгострокової ефективності.

Для забезпечення централізованого управління даними і їхньої доступності, реляційні бази даних є найбільш підходящим вибором. Реляційні бази даних, такі як MySQL, SQLite або PostgreSQL, забезпечують надійність, високий рівень захисту даних, підтримку складних транзакцій і можливість виконання складних запитів. Ці бази даних є ідеальним вибором для підприємства, де точність і надійність даних є критично важливими. Реляційні бази даних дозволяють ефективно керувати великими обсягами даних, що стосуються товарних запасів, замовлень, постачальників та клієнтів, забезпечуючи їхню інтеграцію та доступність у режимі реального часу.

Для прискорення доступу до даних і зменшення навантаження на основну базу даних, доцільно використовувати технології кешування, такі як Redis. Redis забезпечує швидкий доступ до часто використовуваних даних і дозволяє зберігати їх у пам'яті, що значно підвищує продуктивність системи. Це особливо важливо для інформаційної системи, де швидкий доступ до інформації про матеріальні запаси може мати критичне значення для ефективного управління ресурсами компанії.

У процесі створення інформаційної системи управління матеріальними ресурсами першим і одним із найважливіших етапів є дослідження предметної області. Саме на цьому етапі відбувається вивчення організаційних, економічних і технічних аспектів управління матеріальними потоками в межах підприємства. Такий аналіз дозволяє глибше зрозуміти внутрішню структуру облікових, логістичних та управлінських процесів, а також виявити специфічні потреби, які повинна задовольняти майбутня інформаційна система.

Система управління матеріальними ресурсами повинна бути здатною інтегрувати в собі функції обліку, контролю, планування та аналітики, що дозволяє підприємству ефективно управляти запасами, забезпечувати своєчасне постачання ресурсів до відповідних підрозділів та уникати як надлишків, так і дефіциту матеріалів. Особливу увагу варто приділити контролю за строками зберігання, умовами обліку, механізмами списання та ротації матеріальних запасів.

Оскільки система має реалізовуватися за архітектурою клієнт-сервер, це забезпечує розподілення обчислювального навантаження між клієнтською частиною (користувацький інтерфейс) та серверною частиною (обробка бізнес-логіки, зберігання даних, забезпечення доступу). Такий підхід забезпечує масштабованість рішення, можливість централізованого адміністрування, кращу безпеку даних та легшу інтеграцію з іншими компонентами корпоративної ІТ-інфраструктури.

У рамках сучасного підходу до розробки таких систем доцільно розглядати можливість інтеграції з ERP-компонентами або впровадження окремих модулів, що будуть відповідати вимогам підприємства. Зокрема, функціонал повинен охоплювати ведення складу, облік руху матеріалів, формування звітності, планування потреб у ресурсах та взаємодію з постачальниками.

Не менш значущим є аспект інформаційної безпеки. При клієнт-серверній архітектурі важливо реалізувати надійні механізми автентифікації та авторизації користувачів, захисту передаваних даних (з використанням протоколів SSL/TLS), а також забезпечення цілісності даних на рівні бази. Важливим також є аудит дій

користувачів, журналювання операцій та резервне копіювання критичної інформації.

Ще одним ключовим компонентом системи є інтерфейс користувача. Його розробка повинна базуватися на принципах інтуїтивної зрозумілості, адаптивності до різних пристроїв та відповідності специфіці бізнес-процесів. Для цього можуть бути використані сучасні фреймворки, такі як React, Vue.js або Angular, що дозволяють створювати динамічні та зручні веб-інтерфейси.

Крім того, у рамках реалізації системи важливо забезпечити можливості для аналітичної обробки даних. Це дозволяє не лише відслідковувати поточний стан матеріальних ресурсів, але й прогнозувати потреби, аналізувати ефективність використання запасів, виявляти аномалії у витратах. Для цього можуть бути інтегровані засоби бізнес-аналітики (BI), такі як Power BI або Metabase.

Інформаційна система управління матеріальними ресурсами повинна володіти високим рівнем адаптивності до змін внутрішнього і зовнішнього середовища підприємства, а також мати потенціал до масштабування відповідно до зростаючих потреб організації. У цьому контексті ключову роль відіграє правильний вибір архітектурного підходу. Найбільш доцільною є реалізація системи у вигляді модульної структури, яка передбачає поділ функціональності на окремі логічно завершені компоненти. Це дозволяє поступово впроваджувати систему в дію, починаючи з базових підсистем (наприклад, облік надходження матеріалів), а згодом інтегрувати більш складні модулі (аналітика, звітність, автоматизоване планування тощо).

Важливим архітектурним рішенням є використання мікросервісного підходу, за якого кожна функціональна частина реалізується у вигляді окремого сервісу, що виконує строго визначену задачу. Кожен сервіс може масштабуватися незалежно, оновлюватися без впливу на інші компоненти та забезпечує гнучкість під час інтеграції нових функціональностей. Такий підхід істотно полегшує підтримку великої системи, сприяє кращому управлінню складністю та забезпечує стійкість до збоїв — у разі виходу з ладу одного мікросервісу вся система продовжує функціонувати.

Ключовим доповненням до мікросервісної архітектури є використання клієнт-серверної моделі взаємодії, яка визначає чіткий розподіл функцій між фронтом (клієнтська частина) та бекендом (серверна частина). У такій моделі клієнтська частина відповідає за відображення інтерфейсу користувача, прийом та передавання запитів, тоді як серверна частина виконує обробку бізнес-логіки, керування даними та взаємодію з базами даних. Це дозволяє централізовано зберігати та захищати інформацію, а також реалізовувати гнучке управління правами доступу користувачів.

Клієнт-серверна архітектура є також основою для реалізації взаємодії між мікросервісами, які можуть спілкуватися через RESTful API або інші протоколи, такі як gRPC або GraphQL. Завдяки цьому можна забезпечити високу швидкодію, мінімізувати затримки в обробці запитів та підтримувати паралельне обслуговування великої кількості клієнтів.

Особливої уваги потребує також питання масштабованості. На практиці застосовуються два основних підходи — вертикальне масштабування (збільшення обчислювальних ресурсів окремих вузлів) та горизонтальне масштабування (додавання нових вузлів до системи). У разі мікросервісної клієнт-серверної архітектури більш ефективним є горизонтальне масштабування, яке дозволяє незалежно масштабувати окремі компоненти — наприклад, виділяти додаткові ресурси лише для сервісу обробки аналітичних запитів у періоди пікового навантаження.

У поєднанні з використанням хмарних платформ (таких як Amazon Web Services, Microsoft Azure або Google Cloud Platform) клієнт-серверна і мікросервісна архітектури створюють технологічну основу для створення надійної, масштабованої та адаптивної інформаційної системи, здатної не лише відповідати поточним запитам підприємства, але й розвиватися разом із ним. Застосування контейнеризації (Docker) та систем оркестрації (Kubernetes) дозволяє автоматизувати процеси розгортання, моніторингу та підтримки системи, знижуючи навантаження на ІТ-відділ та підвищуючи загальну стабільність платформи.

Таким чином, синергія модульного, мікросервісного та клієнт-серверного підходів забезпечує стратегічну технологічну перевагу в розробці інформаційних систем, орієнтованих на довгострокову експлуатацію, простоту розширення функціоналу, підвищену продуктивність та гнучке реагування на зміну бізнес-вимог..

Проведене дослідження предметної області дозволяє сформулювати чіткі функціональні та нефункціональні вимоги до системи, виявити слабкі місця існуючих підходів до управління матеріальними ресурсами та закласти надійну основу для подальшої розробки програмного забезпечення. Саме системний і методологічний підхід до цього етапу забезпечує високу відповідність інформаційної системи реальним потребам підприємства, що є запорукою її успішної реалізації та подальшої експлуатації.

РОЗДІЛ 2 ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ТА МЕТОДИ І МОДЕЛІ

2.1 Структура і характеристика системи

Інформаційна система управління матеріальними ресурсами є одним з ключових елементів цифрової інфраструктури сучасного підприємства. Її впровадження дозволяє автоматизувати процеси обліку, контролю, аналізу та прогнозування матеріальних потоків, що в умовах високої конкуренції, складної логістики та мінливої ринкової ситуації набуває вирішального значення для стабільної та ефективної діяльності організації.

Структура такої системи зазвичай базується на клієнт-серверній архітектурі, яка забезпечує надійний розподіл функціональних компонентів між серверною частиною (що відповідає за бізнес-логіку, зберігання даних, обробку запитів) та клієнтською частиною (інтерфейс користувача, взаємодія з операторами). Завдяки цьому модель є гнучкою, масштабованою та зручною для експлуатації у підприємствах різного масштабу.

Центральною складовою системи є реляційна база даних, яка виконує функцію зберігання структурованої інформації про матеріальні ресурси підприємства: номенклатуру, партії, серійні номери, одиниці виміру, місця зберігання, акти переміщення, запити, надходження, списання тощо. У системі також зберігаються дані про користувачів, лог доступу, записи інвентаризацій та технічну документацію.

У якості СУБД можуть використовуватись рішення, які підтримують транзакційність, ACID-властивості, реплікацію та резервування, наприклад PostgreSQL, MySQL або SQLite (для вбудованих або локальних систем). У невеликих компаніях або в умовах обмежених ресурсів доцільним є застосування SQLite через його простоту, швидкість і відсутність необхідності в серверному розгортанні. При цьому база має бути оптимізована на рівні запитів, індексів та нормалізації таблиць.

Сервер відповідає за приймання запитів від клієнтів, виконання бізнес-логіки, доступ до БД, обробку результатів і відправку відповідей. У контексті C++-реалізації доцільно використовувати легковаговий HTTP-фреймворк Crow, що дозволяє створювати RESTful API з високою продуктивністю. На серверному рівні реалізуються всі основні модулі системи: облік надходжень, переміщень, автоматичне списання, обробка запитів між підрозділами, формування звітності, контроль залишків.

Для забезпечення безпеки серверної частини застосовуються механізми шифрування (наприклад, HTTPS на базі TLS), токенизація доступу (JWT), авторизація на основі ролей, логування дій.

Фронтенд системи — це інструмент взаємодії персоналу з інформаційною системою. Веб-інтерфейс реалізується за допомогою стандартних вебтехнологій: HTML5, CSS3, JavaScript, з можливістю використання сучасних фреймворків на кшталт React або Vue.js. Це дозволяє забезпечити адаптивний дизайн, інтерактивність, швидку навігацію та гнучке налаштування вигляду під різні ролі користувачів (складський персонал, логіст, адміністратор, керівник).

Усі операції здійснюються через браузер, а взаємодія з сервером реалізується через асинхронні HTTP-запити (AJAX або Fetch API). Це дозволяє забезпечити оновлення даних у реальному часі без перезавантаження сторінки та значно підвищує продуктивність праці.

Для підтримки безперебійного функціонування ІС необхідна стабільна мережа з високою пропускною здатністю. Доцільним є використання оптоволоконного з'єднання, керованих комутаторів з QoS, VLAN для сегментації трафіку та VPN-з'єднань для захищеного доступу. У розподілених структурах — філіях, віддалених складах — обов'язкове впровадження маршрутизаторів із підтримкою шифрування, моніторингу та резервних каналів зв'язку.

Для резервного копіювання та захисту даних від збоїв рекомендується впровадити систему резервного зберігання, як локального (NAS/RAID), так і хмарного рівня (Google Cloud Storage, Amazon S3), що дозволяє відновити критичну інформацію у разі аварії.

Характеристики ефективної інформаційної системи управління матеріальними ресурсами

- ефективна ІС повинна відповідати таким критеріям:
- надійність — стабільна робота при навантаженнях, збереження цілісності даних;
- масштабованість — можливість розширення функціоналу або підключення нових підрозділів без зміни архітектури;
- інтеграційність — підтримка взаємодії з іншими системами ;
- безпека — контроль доступу, захист даних, аудит дій;
- інтуїтивність — зрозумілий інтерфейс;
- аналітичність — вбудовані інструменти звітності та прийняття рішень на основі реальних даних.

Структура інформаційної системи управління матеріальними ресурсами повинна забезпечувати цілісний і гнучкий підхід до автоматизації всіх ключових процесів, пов'язаних з матеріальним забезпеченням. Її ефективність визначається не лише технічними характеристиками окремих компонентів, а й узгодженістю їх взаємодії, відповідністю бізнес-процесам підприємства та здатністю адаптуватися до змін зовнішнього середовища. Впровадження такої системи дає змогу не лише зменшити втрати, пов'язані з неефективним обліком, а й сформуванню підґрунтя для стратегічного управління ресурсами на основі об'єктивних даних.

Інформаційні системи управління матеріальними ресурсами є невід'ємною складовою сучасного підприємства, оскільки забезпечують автоматизацію, моніторинг та контроль усіх процесів, пов'язаних із надходженням, зберіганням, обліком, переміщенням та використанням матеріально-технічних цінностей. Ефективність такої системи безпосередньо впливає на стабільність виробничих процесів, оптимізацію витрат, своєчасність постачання ресурсів і, як наслідок, на загальну результативність діяльності підприємства.

Структура інформаційної системи управління матеріальними ресурсами, як правило, має багаторівневу архітектуру, до складу якої входять кілька ключових компонентів. Кожен із них виконує свою функцію та відповідає за конкретний

аспект управління ресурсами. До основних елементів системи належать база даних, серверна інфраструктура, клієнтські інтерфейси, мережеве середовище та прикладне програмне забезпечення, що забезпечує логіку функціонування.

Центральним елементом інформаційної системи виступає база даних, яка акумулює всю критично важливу інформацію про матеріальні ресурси: номенклатуру, залишки, постачання, внутрішні переміщення, акти видачі, інвентаризацію, а також супровідну документацію. Реляційні бази даних — такі як PostgreSQL, SQLite або MySQL — найчастіше використовуються для таких цілей завдяки своїй структурованості, підтримці транзакцій, можливостям масштабування і забезпеченню цілісності даних. База даних повинна бути належним чином оптимізована: включати систему індексів, забезпечувати контроль цілісності записів, підтримку зовнішніх ключів і обмежень, що є особливо актуальним у випадках великого обсягу даних і високих вимог до надійності обліку.

Таким чином, ефективне функціонування бази даних є критичним для успішної реалізації усіх інших модулів інформаційної системи, оскільки від неї залежить не лише швидкодія системи, але й достовірність та актуальність усіх облікових і аналітичних даних, на основі яких приймаються управлінські рішення.

Серверна інфраструктура відіграє центральну роль у функціонуванні інформаційної системи управління матеріальними ресурсами, оскільки саме сервери відповідають за обробку запитів від клієнтських додатків, реалізацію бізнес-логіки, доступ до бази даних та забезпечення взаємодії між модулями системи. Для досягнення високої продуктивності, відмовостійкості та масштабованості доцільно реалізувати серверну частину у вигляді кластерної архітектури, яка дозволяє розподіляти навантаження між декількома вузлами. Це забезпечує безперебійну роботу системи навіть у разі виходу з ладу окремих компонентів, а також підвищує швидкість обробки паралельних запитів від користувачів.

Залежно від масштабів підприємства та інтенсивності операцій з матеріальними ресурсами, можливе використання локальних серверів або

впровадження рішень на основі хмарної інфраструктури, таких як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform. Хмарні технології надають додаткову гнучкість — дозволяють автоматично масштабувати обчислювальні потужності відповідно до навантаження, оптимізувати витрати на IT-інфраструктуру та забезпечити резервне зберігання даних.

Клієнтські додатки забезпечують інтерфейс взаємодії кінцевого користувача з інформаційною системою. Вони є ключовим інструментом для виконання повсякденних операцій — введення даних про надходження та переміщення ресурсів, реєстрації запитів на видачу, перегляду залишків, формування звітності, моніторингу руху матеріалів та керування структурою складів. Якість інтерфейсу безпосередньо впливає на ефективність роботи персоналу, тому він має бути інтуїтивно зрозумілим, доступним на різних пристроях (ПК, планшетах, мобільних терміналах) і адаптованим під специфіку бізнес-процесів підприємства.

Сучасні вебтехнології, такі як HTML5, CSS3, JavaScript та фреймворки React, Vue.js або Angular, дозволяють створювати динамічні, адаптивні та масштабовані інтерфейси з високою швидкістю відгуку. Використання таких технологій забезпечує підтримку асинхронної взаємодії з сервером (через AJAX, Fetch API або WebSocket), що сприяє ефективній роботі в режимі реального часу.

Мережеве обладнання виконує критично важливу функцію забезпечення стабільного зв'язку між усіма компонентами інформаційної системи — клієнтами, серверами, базою даних та зовнішніми сервісами. Надійна та високошвидкісна мережа є обов'язковою умовою для безперебійної роботи системи, особливо у випадках, коли підприємство має розгалужену структуру або віддалені філії. Для цього використовуються оптоволоконні з'єднання, керовані комутатори, багатофункціональні маршрутизатори з підтримкою VPN, QoS (Quality of Service) та інші засоби оптимізації трафіку.

Таким чином, усі технічні компоненти інформаційної системи повинні бути узгоджено інтегровані та налаштовані відповідно до реальних потреб підприємства. Їх грамотне поєднання забезпечує стабільність, масштабованість,

швидкодію та зручність експлуатації системи в умовах реального виробничого або логістичного середовища.

Для інформаційної системи з управління матеріальними ресурсами компанії було розроблено ряд діаграм.

Діаграма класів (Class Diagram)

Діаграма класів для системи відображає зв'язки між основними сутностями та їхніми атрибутами (рис. 2.1)

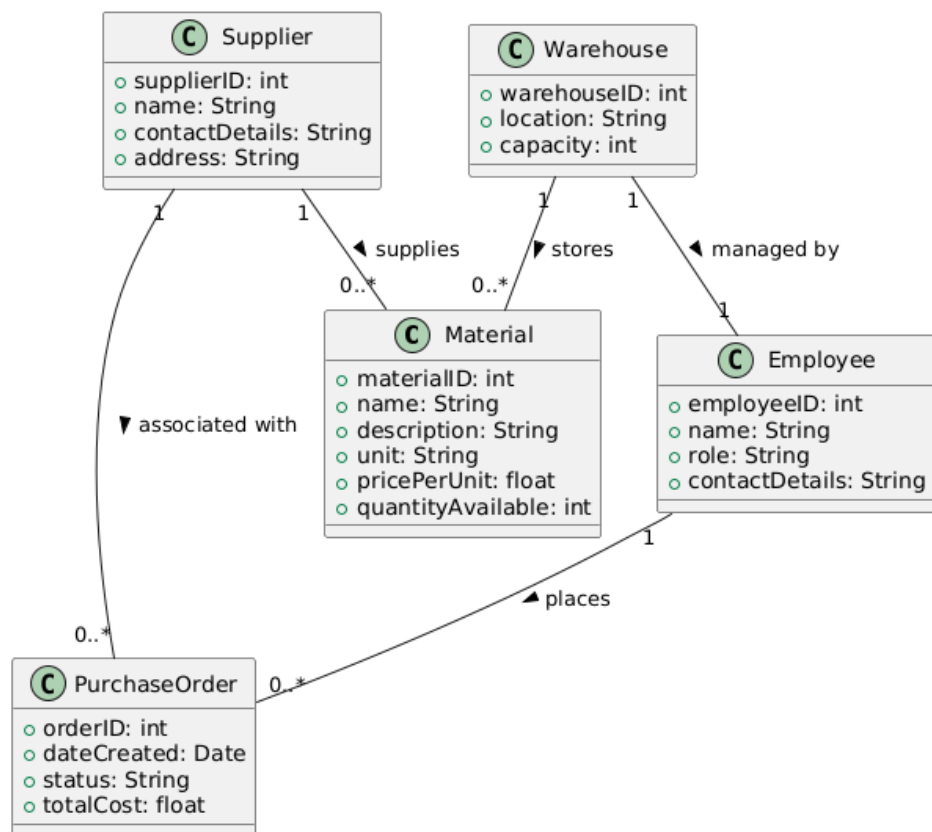


Рисунок 2.1. - Діаграма класів

Джерело: сформовано автором на основі виконаного дослідження

Діаграма прецедентів (Use Case діаграма) показує основні взаємодії між користувачами системи (акторами) та функціями, які система повинна виконувати. Діаграма процесів (діаграма станів) відображає різні стани об'єктів у системі та переходи між цими станами. Діаграма класів описує структуру системи на рівні класів, показуючи їхні атрибути та методи, а також зв'язки між класами.

Діаграма прецедентів (Use Case Diagram) є важливим інструментом для візуалізації взаємодії користувачів із системою та ілюстрації функціональних вимог. Нижче наведено детальний опис цієї діаграми для інформаційної системи

Діаграма ролей (Roles Diagram) (рис. 2.2)

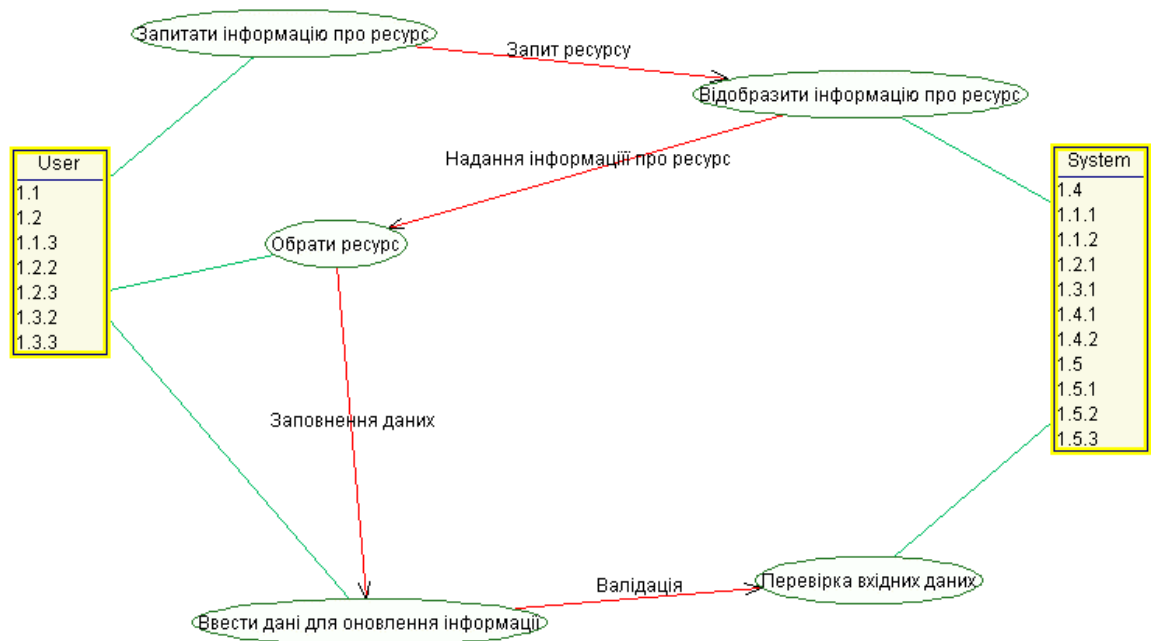


Рисунок 2.2 - Діаграма ролей

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності (Roles Diagram)

Діаграма послідовності для роботи зі списком ресурсів (рис. 2.3)

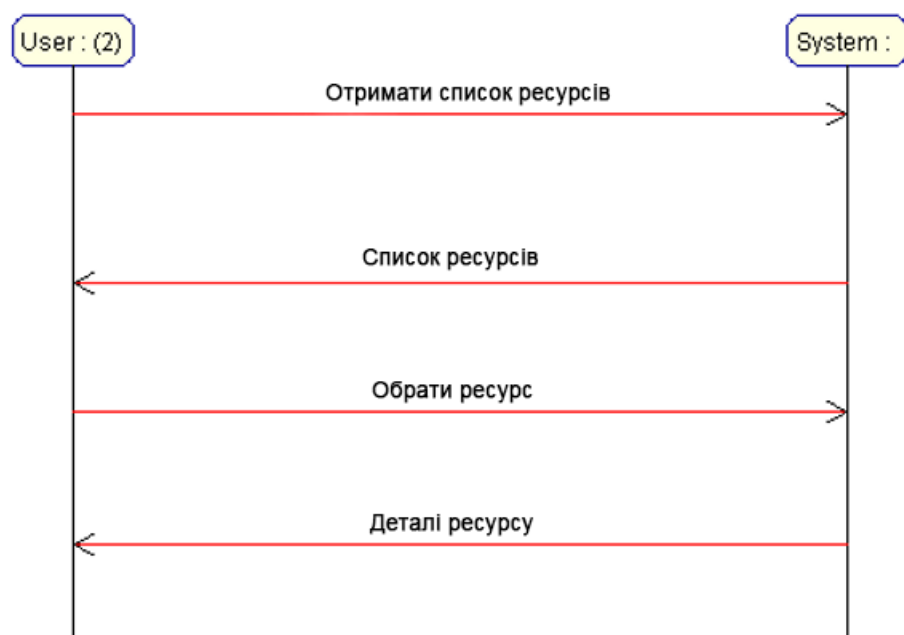


Рисунок 2.3 - Діаграма роботи зі списком ресурсів

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності роботи з редагуванням ресурсу (рис. 2.4)

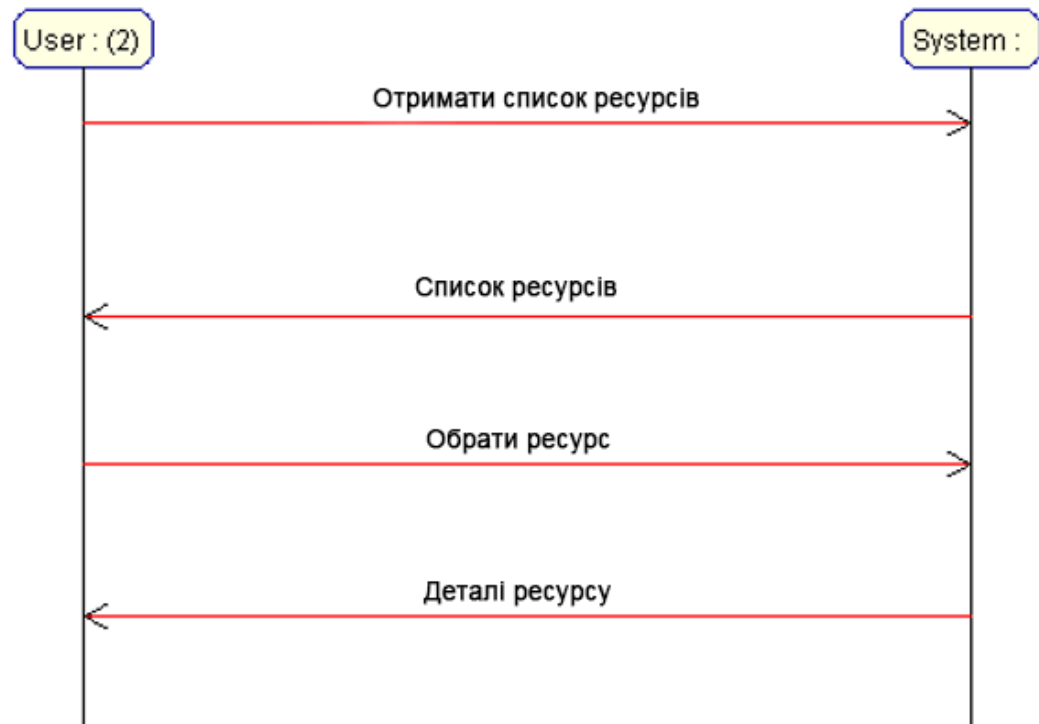


Рисунок 2.4 - Діаграма роботи з редагуванням ресурсу

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності для роботи валідації (рис. 2.5)

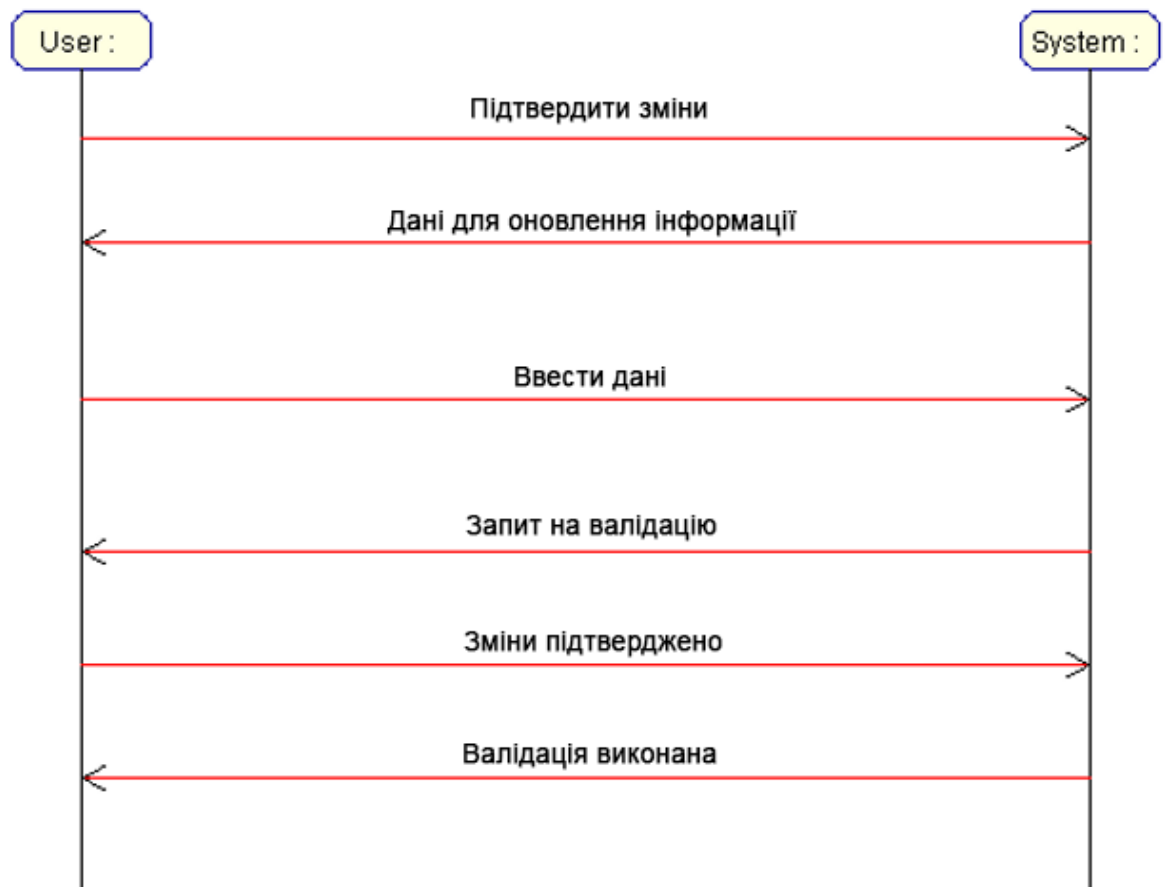


Рисунок 2.5 - Діаграма для роботи валідації

Джерело: сформовано автором на основі виконаного дослідження

Діаграма стану (State Diagram)

Діаграма стану для ResourceManager (рис. 2.6)

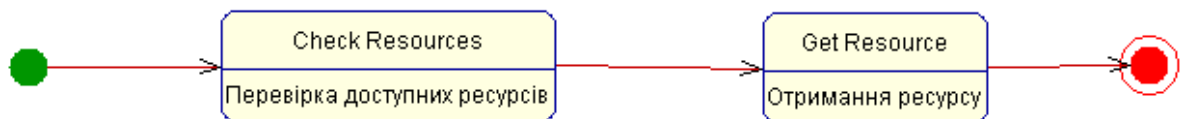


Рисунок 2.6 - Діаграма стану ResourceManager

Джерело: сформовано автором на основі виконаного дослідження

Діаграма стану для Validation (рис. 2.7)

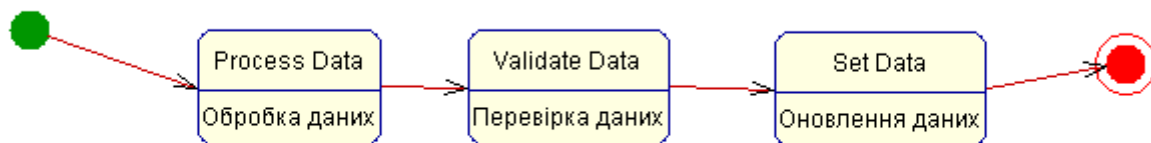


Рисунок 2.7 - Діаграма стану Validation

Джерело: сформовано автором на основі виконаного дослідження

Ці діаграми ілюструють взаємодії користувачів із системою, процеси, які відбуваються всередині системи, та структуру системи на рівні класів.

Вони допомагають зрозуміти загальну архітектуру та функціональність інформаційної системи, забезпечуючи чітке уявлення про те, як система повинна працювати та як різні її компоненти взаємодіють один з одним.

2.2 Методи та моделі в інформаційних управляючих системах і технологіях

Сучасні підприємства функціонують в умовах постійних змін, підвищеної конкуренції та великого обсягу даних, що потребують своєчасної обробки та аналізу. У контексті управління матеріальними ресурсами це означає, що рішення мають прийматись швидко, обґрунтовано та з урахуванням багатьох факторів — економічних, логістичних, технічних, організаційних тощо. Для досягнення такої якості управління необхідним є застосування сучасних методів і моделей, реалізованих у складі інформаційних управляючих систем (ІУС).

ІУС для управління матеріальними ресурсами повинна забезпечувати ефективне функціонування як у структурованих, так і в слабоформалізованих умовах. З огляду на це, до складу системи мають входити як класичні економіко-

математичні моделі, так і модулі інтелектуального аналізу даних, що опираються на методи штучного інтелекту.

Для вирішення структурованих задач, до яких належать моделі обліку запасів, оптимізації закупівель, логістики та складування, доцільно використовувати кількісні методи аналізу: моделі лінійного і динамічного програмування, транспортні моделі, методи прогнозування часових рядів, а також теорію масового обслуговування для оцінки ефективності переміщення ресурсів. Такі моделі дозволяють формалізувати процеси, побудувати аналітичні залежності між ключовими змінними та забезпечити оптимальність рішень у межах заданих ресурсних обмежень.

При вирішенні слабоструктурованих задач, які супроводжуються як кількісними, так і якісними параметрами, доцільно впроваджувати моделі на основі нечітких множин та методів нечіткої логіки. Це дозволяє враховувати невизначеність при оцінюванні ризиків постачання, аналізі стабільності постачальників, адаптації до сезонності попиту чи внутрішніх організаційних змін. У таких випадках використовуються нечіткі продукційні правила, нечіткі ієрархічні дерева рішень, а також алгоритми нечіткої класифікації для оцінки та ранжування альтернатив.

Особливу цінність в ІУС набувають інструменти штучного інтелекту, що дозволяють вирішувати нестандартні та неструктуровані задачі, наприклад, прогнозування змін у структурі споживання матеріалів, пошук аномалій у витратах ресурсів, автоматичну класифікацію типових сценаріїв нестачі чи перенасичення складу. У таких випадках використовуються нейронні мережі, методи глибокого навчання, моделі підтримки прийняття рішень на основі машинного навчання, дерева рішень, імовірнісні графи та генетичні алгоритми.

Для досягнення гнучкості та адаптивності системи в умовах змін зовнішнього середовища (наприклад, порушення логістичних ланцюгів або коливання цін постачальників) застосовуються еволюційні методи, серед яких: генетичне програмування, мурашині та роїнгові алгоритми, імітаційне моделювання та гібридні нейро-нечіткі системи. Вони дозволяють системі

самонавчатись, адаптуватись до нових даних та сценаріїв і забезпечувати прийняття рішень у режимі реального часу з високим ступенем автономності.

В окремих випадках використовується апарат експертних систем, які дозволяють формалізувати знання досвідчених співробітників та моделювати їхню логіку прийняття рішень. Такі підходи особливо корисні в системах підтримки логістичних рішень, де значну роль відіграє досвід і практичне знання специфіки матеріальних потоків на конкретному підприємстві.

Крім того, актуальними є гібридні моделі, що об'єднують класичні кількісні підходи з елементами штучного інтелекту. Наприклад, прогнозування попиту може ґрунтуватися на часових рядах, але доповнюватись кластеризацією клієнтів і виявленням закономірностей у споживанні ресурсів за допомогою алгоритмів машинного навчання.

Узагальнюючи, можна стверджувати, що ефективна ІУС для управління матеріальними ресурсами підприємства повинна містити у своїй структурі набір моделей, які забезпечують:

- аналіз та оптимізацію складських операцій;
- багатокритеріальне прийняття рішень у логістичних задачах;
- адаптивне планування постачань;
- виявлення аномалій та відхилень у матеріальних потоках;
- автоматичну класифікацію та прогнозування потреб у ресурсах.

Таким чином, застосування математичних методів і моделей у складі інформаційної системи дає змогу перетворити складні та неструктуровані бізнес-процеси у формалізовані та керовані, забезпечуючи тим самим високий рівень обґрунтованості управлінських рішень на всіх рівнях функціонування підприємства.

РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ

3.1 Проектування бази даних та/або сховища даних для інформаційної управляючої системи

3.1.1 Проектування інфологічної моделі БД

Інфологічне моделювання є фундаментальним етапом у процесі проектування бази даних для інформаційної системи управління матеріальними ресурсами підприємства. На цьому рівні виконується абстрактне, логічне представлення предметної області у вигляді взаємозв'язаних інформаційних об'єктів, що підлягають зберіганню, обробці та аналізу в межах майбутньої ІС.

У межах даної предметної області — управління матеріальними ресурсами — базовими сутностями, які потребують структурованого представлення у БД, є: матеріали, категорії матеріалів, склади, внутрішні переміщення, постачання, використання ресурсів, постачальники, відповідальні особи, а також користувачі системи. Кожна сутність характеризується певним набором атрибутів, що відображають її властивості у реальному бізнес-процесі.

Сутність Матеріал включає такі атрибути, як унікальний ідентифікатор, назва, одиниця виміру, категорія, поточна кількість на складі, мінімальний залишок, термін придатності (якщо застосовно), статус.

Сутність Постачання описується такими характеристиками: дата постачання, постачальник, перелік матеріалів, кількість, вартість, номер документа.

Сутність Переміщення фіксує логістичні операції в межах підприємства: звідки і куди переміщується ресурс, обсяг, дата, підпис відповідального.

Між сутностями визначено структурні зв'язки:

- один постачальник може здійснювати багато постачань → зв'язок один до багатьох (1:N);
- один матеріал належить до однієї категорії → зв'язок багато до одного (N:1);

- один матеріал може зберігатись на різних складах → зв'язок реалізується через окрему таблицю складування або обліку залишків;
- всі дії користувачів фіксуються у журналах змін → зв'язок один користувач — багато записів журналу.

Після виявлення сутностей було проведено нормалізацію логічної структури до третьої нормальної форми (3НФ), що дозволяє усунути надлишковість даних, уникнути аномалій оновлення та забезпечити логічну цілісність. Основні принципи нормалізації полягали у виділенні окремих таблиць для груп атрибутів, які мають залежність від неключових полів або від частини складного ключа. Таким чином, досягнуто оптимальної структури БД з точки зору підтримки консистентності та ефективності запитів.

На завершення побудовано інфологічну (логічну) ER-модель, що відображає всі сутності, їх атрибути та зв'язки між ними.

Схема інфологічної моделі бази даних подана нижче (включається у текст роботи або розміщується у додатку у вигляді зображення). Вона служить основою для подальшого переходу до фізичного рівня проєктування, де будуть визначені типи даних, індекси, зовнішні ключі та параметри оптимізації продуктивності.

У результаті логічного моделювання сформовано цілісну структуру даних, яка повністю відображає інформаційні потреби системи управління матеріальними ресурсами, відповідає специфіці предметної області та закладає надійне підґрунтя для реалізації на фізичному рівні в середовищі обраної СКБД (рис. 3.1)

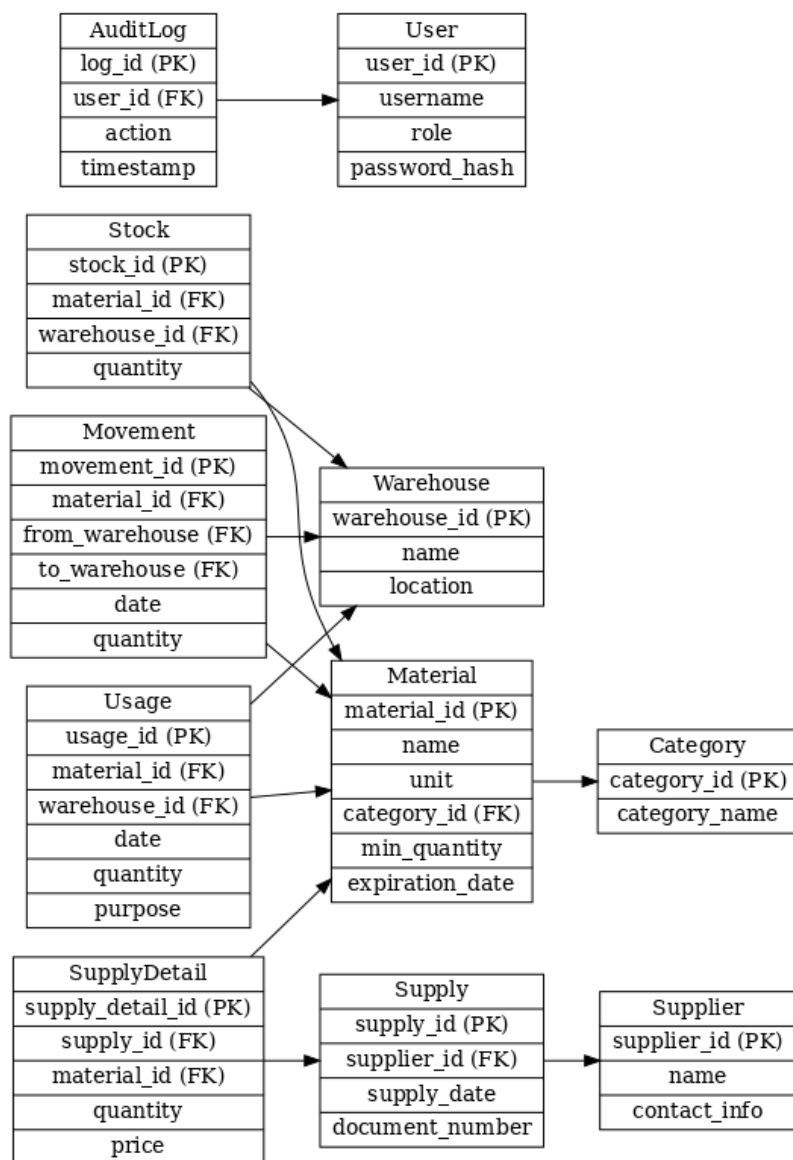


Рисунок 3.1 – Інфологічна модель бази даних

Джерело: сформовано автором на основі виконаного дослідження

3.1.2 Обґрунтування вибору СКБД та проєктування фізичної моделі БД в середовищі вибраної СКБД

Проектування бази даних для інформаційної системи з управління матеріальними ресурсами компанії є ключовим етапом у створенні ефективної та надійної системи. У даному проєкті було створено базу даних, яка складається з кількох взаємопов'язаних таблиць, які забезпечують комплексний облік та управління всіма аспектами товарного обліку. Ця база даних була спроектована з

урахуванням вимог до зберігання, обробки та аналізу даних, необхідних для ефективного управління запасами, обліку та відображення.

Основною таблицею в базі даних є таблиця `Users`, яка зберігає інформацію про всіх користувачів системи. Ця таблиця включає такі поля, як `id`, `name`, `email`, `role` та `password`. Поле `id` є первинним ключем і автоматично для кожного нового запису. Інші поля містять відповідну інформацію про користувача, включаючи електронну пошту, яка є унікальною для кожного користувача, та роль, яка визначає права доступу в системі (рис. 3.2)

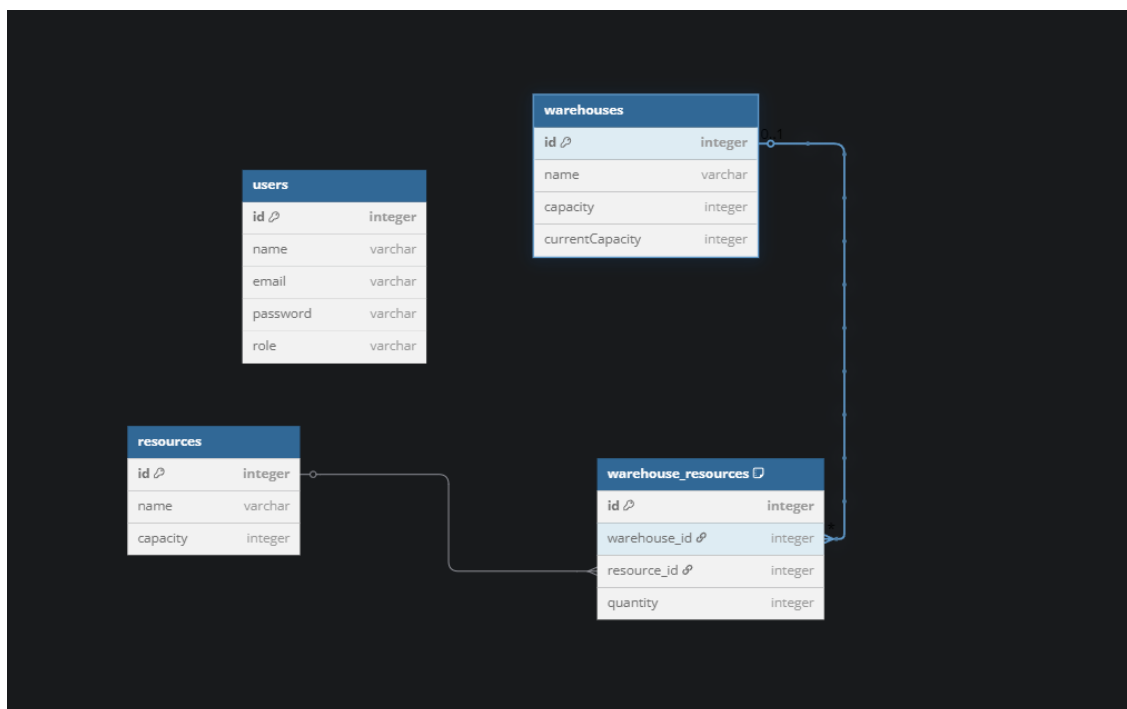


Рисунок 3.2 - Модель бази даних

Джерело: сформовано автором на основі виконаного дослідження

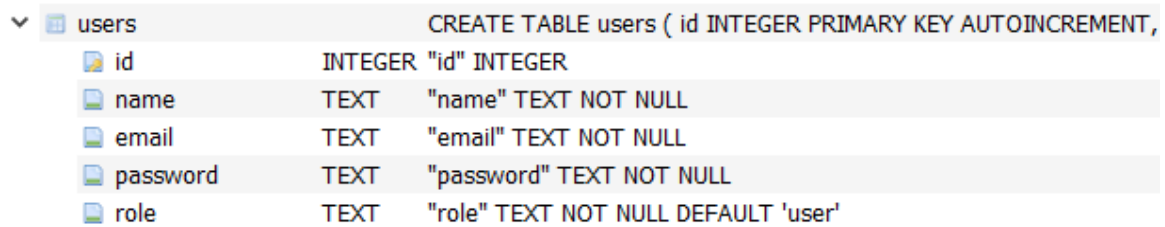
Таблиця `resources` зберігає інформацію про ресурс, включаючи поля `id`, `name` та `capacity`. Поле `id` є первинним ключем і автоматично інкрементується (рис. 3.3)

resources	CREATE TABLE resources (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, capacity INTEGER NOT NULL)	
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT NOT NULL
capacity	INTEGER	"capacity" INTEGER NOT NULL

Рисунок. 3.3 - Таблиця Resources

Джерело: сформовано автором на основі виконаного дослідження

Таблиця `users` зберігає інформацію про користувача, включаючи поля `id`, `name`, `email`, `password` та `role`. Поле `id` є первинним ключем і автоматично інкрементується (рис. 3.4)

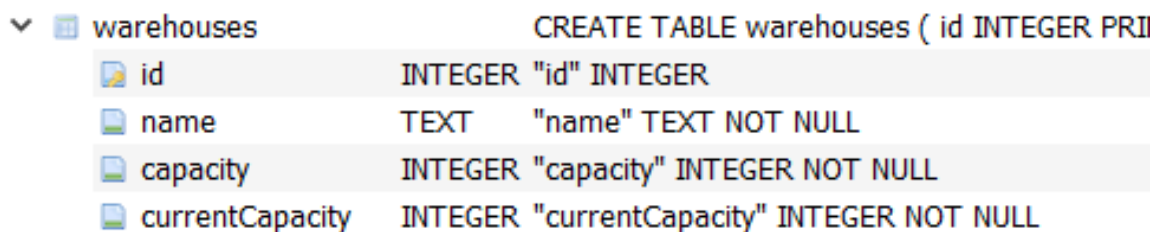


users	CREATE TABLE users (id INTEGER PRIMARY KEY AUTOINCREMENT,	
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT NOT NULL
email	TEXT	"email" TEXT NOT NULL
password	TEXT	"password" TEXT NOT NULL
role	TEXT	"role" TEXT NOT NULL DEFAULT 'user'

Рисунок. 3.4 - Таблиця Users

Джерело: сформовано автором на основі виконаного дослідження

Таблиця `warehouses` зберігає інформацію про склади, включаючи поля `id`, `name`, `capacity` та `currentCapacity`. Поле `id` є первинним ключем і автоматично інкрементується (рис. 3.5)



warehouses	CREATE TABLE warehouses (id INTEGER PRI	
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT NOT NULL
capacity	INTEGER	"capacity" INTEGER NOT NULL
currentCapacity	INTEGER	"currentCapacity" INTEGER NOT NULL

Рисунок. 3.5 - Таблиця Warehouses

Джерело: сформовано автором на основі виконаного дослідження

Загальна структура бази даних забезпечує надійне зберігання та обробку даних, необхідних для управління матеріальними ресурсами підприємства. Використання зовнішніх ключів забезпечує цілісність даних та їхню взаємозв'язаність, що дозволяє ефективно відстежувати всі процеси обліку матеріальних ресурсів.

3.1.3 Контрольний приклад та обрахунок прогнозних обсягів БД

Контрольний приклад є важливим інструментом перевірки коректності структури бази даних, а також необхідною умовою для проведення розрахунків прогнозованого обсягу збережених даних у реальних умовах експлуатації. Цей етап дозволяє не лише здійснити початкове завантаження бази тестовими записами, а й оцінити ефективність її структури, потенційну продуктивність, а також виявити можливі обмеження у масштабуванні системи.

Вимоги до контрольного прикладу.

Контрольний приклад був сформований з урахуванням типових сценаріїв роботи підприємства з матеріальними ресурсами. Передбачалося, що система буде використовуватись на середньому підприємстві з кількома складами, регулярними постачаннями, міжскладськими переміщеннями та обліком використання матеріалів. Для моделювання було створено наступний обсяг тестових даних:

- користувачі: 10 записів (адміністратори, менеджери складу, аналітики);
- категорії матеріалів: 8 записів (електроніка, упаковка, будматеріали тощо);
- матеріали: 100 записів;
- склади: 3 записи;
- залишки (stock): 300 записів (поєднання складів і матеріалів);
- постачальники: 15 записів;
- постачання: 50 записів;
- деталі постачань: 250 записів (у середньому по 5 позицій на постачання);
- переміщення: 70 записів;
- використання: 90 записів;
- журнал дій користувачів (AuditLog): 200 записів;

Цей обсяг дозволив повністю перевірити цілісність та узгодженість логічних зв'язків між таблицями, зокрема: зовнішні ключі, каскадні оновлення, цілісність атрибутів та підтримку нормалізованої структури.

Прогнозування обсягів бази даних.

Оцінка прогнозованого обсягу бази даних базувалась на розрахунках довжини записів у таблицях, поточного обсягу даних та середнього темпу зростання в кількісному вимірі. Для прикладу обрано часовий горизонт у 12 місяців, із урахуванням щомісячного приросту записів (таблиця 3.1)

Таблиця 3.1 – Приклад обчислення довжини запису для таблиці Material

Поле	Тип даних	Довжина (байт)
material_id (INT)	INTEGER	4
name (VARCHAR 100)	TEXT (avg)	50
unit (VARCHAR 10)	TEXT (avg)	10
category_id (INT)	INTEGER	4
min_quantity (INT)	INTEGER	4
expiration_date	DATE	8

Джерело: сформовано автором на основі виконаного дослідження

Разом – 80 байт.

Аналогічно були проаналізовані всі ключові таблиці. Приблизний розмір одного повного запису в таблиці SupplyDetail складає ~60 байт, у таблиці AuditLog ~80 байт, Usage ~70 байт тощо (таблиця 3.2)

Таблиця 3.2 – Прогнозований річний обсяг даних

Таблиця	Довжина 1 запису	Приріст/міс	Обсяг за 12 міс (байт)
Material	80	+10	$80 \times 10 = 800$
SupplyDetail	60	+250	$60 \times 250 \times 12 = 180,000$

AuditLog	80	+300	$80 \times 300 \times 12 = 288,000$
Usage	70	+200	$70 \times 200 \times 12 = 168,000$
Stock	40	+100	$40 \times 100 \times 12 = 48,000$
Movement	60	+100	$60 \times 100 \times 12 = 72,000$

Джерело: сформовано автором на основі виконаного дослідження

Загалом – приблизно 756,800 байт або 739 КБ.

З урахуванням індексів, накладних системних структур та зберігання журналів змін прогнозується, що річний обсяг бази даних не перевищить 3–5 МБ у форматі SQLite, що є прийнятним показником для локальної або веборієнтованої ІС.

Контрольне завантаження бази даних підтвердило її працездатність, коректність структурних зв'язків і готовність до масштабування. Проведений аналіз прогнозного зростання обсягів даних показав, що навіть за активної експлуатації система залишатиметься ефективною та не потребуватиме зміни СУБД у межах перших років експлуатації. Така оцінка дає підстави вважати обрану архітектуру та модель зберігання даних доцільною з технічного, логічного та ресурсного погляду.

3.2 Проектування бази знань інформаційної управляючої системи та/або засоби інтелектуального аналізу даних

3.2.1 Реалізація інтелектуального аналізу великих даних з метою наповнення бази знань

У цьому розділі буде описано реалізацію інтелектуального аналізу даних для управління матеріальними ресурсами компанії, а також проектування і реалізацію бази знань. Для досягнення цієї мети необхідно застосувати сучасні методи аналізу даних та створити систему, яка дозволить ефективно керувати

інформацією про клієнтів і їх поведінку. Основною метою є підвищення ефективності управління матеріальними ресурсами за допомогою інтелектуальних алгоритмів і створення бази знань для кращого розуміння потреб користувачів.

Інтелектуальний аналіз даних є невід'ємною частиною сучасних інформаційних систем. Для управління матеріальними ресурсами компанії використовується аналіз великих обсягів даних, що дозволяє виявити закономірності і тренди в поведінці користувачів та ефективно вести облік ресурсів. Основні етапи реалізації інтелектуального аналізу даних включають збір даних, їх попередню обробку, вибір моделей та алгоритмів для аналізу, навчання моделей і їх тестування.

Для збору даних використовується інформація з різних джерел, включаючи дані про продажі, дані про постачання, інформацію про користувачів та їх потреби. Попередня обробка даних включає очищення даних, нормалізацію, усунення пропусків і аномалій. Далі обираються моделі і алгоритми, які найбільше підходять для аналізу цих даних. Це можуть бути кластеризація, регресійний аналіз, алгоритми машинного навчання тощо.

База знань є основним компонентом системи управління інформацією про клієнтів та їх поведінку. Вона дозволяє зберігати і аналізувати дані про клієнтів, їх покупки і поведінку, а також використовувати ці дані для прийняття рішень. Основними завданнями при проектуванні бази знань є визначення структури бази, вибір інструментів для її реалізації і забезпечення інтеграції з іншими компонентами системи.

База знань може включати інформацію про клієнтів, сегментацію клієнтів, дані про їх покупки, картки лояльності та інші релевантні дані. Наприклад, система може ділити споживачів на сегменти і збирати інформацію про їх купівлі та продажі за допомогою карток лояльності. Це дозволяє відстежувати поведінку клієнтів і використовувати інтелектуальні методи аналізу даних для виявлення закономірностей у їх поведінці. Завдяки цьому можна краще розуміти потреби клієнтів, прогнозувати їхні майбутні покупки і ефективніше керувати запасами товарів.

Для реалізації інтелектуального аналізу даних використовуються сучасні технології і алгоритми машинного навчання. Наприклад, кластеризація клієнтів дозволяє виділити групи клієнтів з подібними характеристиками і поведінкою, що допомагає налаштувати маркетингові кампанії і акції для кожного сегменту. Регресійний аналіз і прогнозування дозволяють передбачити майбутні потреби в товарах, що зменшує ризики надлишкових запасів або дефіциту товарів.

Інтелектуальні методи аналізу даних також використовуються для виявлення аномалій у даних про продажі та постачання, що допомагає швидко реагувати на можливі проблеми і ухвалювати відповідні заходи. Завдяки цьому система стає більш гнучкою і ефективною в управлінні товарними потоками.

Реалізація інтелектуального аналізу даних і проектування бази знань є важливими етапами в створенні інформаційної системи для управління товарними потоками. Вони дозволяють більш ефективно використовувати наявні дані, покращувати управління запасами, розуміти потреби клієнтів і прогнозувати їхні майбутні дії. Це сприяє підвищенню ефективності бізнесу і забезпечує конкурентні переваги на ринку.

3.2.2 Проектування та реалізація бази знань інформаційної управляючої системи

Процес проектування та реалізації бази знань у контексті розробки інформаційної управляючої системи для управління матеріальними ресурсами підприємства є одним з ключових етапів створення інтелектуального програмного рішення, яке не лише фіксує інформацію, а й підтримує прийняття управлінських рішень на основі отриманих закономірностей, правил та прогнозних висновків.

Етап ідентифікації бази знань.

На першому етапі було визначено задачі, які має підтримувати база знань у рамках функціонування ІУС. Серед них — автоматизований аналіз динаміки витрат матеріалів, прогнозування потреб у закупівлях, виявлення відхилень у споживанні ресурсів, формування рекомендацій щодо оптимального розподілу

залишків між складами. Для кожної задачі було складено неформальні описи: вхідні дані (виписки з облікових систем, логістичні дані, календарі подій), очікувані результати (рекомендації, прогнози, сповіщення) та критерії якості (точність, достовірність, оперативність).

- Цілями створення бази знань є:
- формалізація знань, отриманих внаслідок ІАД;
- автоматизація рутинних процедур контролю за матеріальними потоками;
- підтримка прийняття рішень з боку менеджерів матеріально-технічного забезпечення;
- можливість масштабування на інші виробничі або логістичні сценарії.

Етап концептуалізації бази знань.

Було здійснено деталізований аналіз предметної області — управління матеріальними ресурсами, в результаті якого ідентифіковано основні поняття (матеріал, постачання, витрата, критичний запас, залишок), а також встановлено їхні взаємозв'язки: «матеріал використовується у виробничому циклі», «залишок формується внаслідок постачання», «запит формується за результатами дефіциту» тощо. Обрано тип знань — процедурні (регулярності та шаблони дій), евристичні (підказки) та логічні (причинно-наслідкові правила).

Було визначено типи задач, які потребують вирішення в ІУС:

- класифікація матеріалів за ризиком дефіциту;
- побудова прогнозу споживання на основі історичних даних;
- виявлення аномалій у частоті переміщень;
- формування повідомлень про можливе перевищення ліміту витрат.

Етап формалізації бази знань

Для подання знань доцільно обрати гібридну модель: продукційні правила (if-then) і логічні зв'язки у вигляді умовно-логічних блоків. Базу знань структурувати у вигляді набору фактів і правил, які інтерпретуються механізмом логічного висновку на основі прямої логіки. Тип механізму — детермінований, із

можливістю розширення fuzzy-модулями для роботи з неточними оцінками (наприклад, «низький залишок», «часте переміщення»).

Для реалізації бази знань слід обрати середовище Orange Data Mining із можливістю візуального конструювання сценаріїв, а також модуль Rules Evaluation для виведення логіки прийняття рішень.

Етап реалізації бази знань.

На цьому етапі база знань може бути безпосередньо наповнена виведеними з правилами, зокрема правилами класифікації матеріалів за ступенем важливості, шаблонами поведінки складів (наприклад, які часто мають надлишки або нестачі), рекомендаційними правилами щодо термінів закупівлі. Кожен модуль знань інтегруватися з операційною базою даних системи через API-запити до SQLite, що забезпечити динамічну актуалізацію знань.

Можлива інтеграція з інтерфейсом користувача — у вигляді дашборду рекомендацій і системи попереджень, що використовує висновки бази знань у реальному часі.

Етап тестування.

Для перевірки якості бази знань слід створити тестовий набір сценаріїв із реальних даних. Система повинна демонструвати здатність правильно класифікувати матеріали, коректно формувати рекомендації та виявляти аномалії. Тест граничних випадків (великі обсяги даних, нульові залишки, перерви в постачанні), що підтвердить надійність логіки виводу.

Етап дослідної експлуатації.

База знань можк бути впроваджена у тестову версію системи управління, де надалі пройде перевірку з боку кінцевих користувачів — завідувачів складів, логістів та аналітиків. Відгуки можуть підтвердити користь рекомендацій, особливо у випадках раптового зростання споживання ресурсів. Було також запропоновано розширити базу знань сценаріями розподілу надлишків між складами, що увійшло у план подальшого вдосконалення.

Реалізація бази знань забезпечить перехід інформаційної системи управління матеріальними ресурсами від простого інструмента обліку до засобу

аналітичної підтримки рішень. Завдяки застосуванню CRISP-DM методології, сучасних засобів машинного навчання і візуального аналізу, система отримала функціонал, здатний адаптуватися до змін бізнес-середовища, мінімізувати ризики дефіциту ресурсів та підвищити якість логістичних рішень.

3.3. Програмне забезпечення

3.3.1 Проектування програмного забезпечення

Функціональні вимоги до інформаційної системи управління матеріальними ресурсами визначають базовий обсяг можливостей, який повинна забезпечувати система з метою автоматизації обліку, контролю та аналізу всіх процесів, пов'язаних із рухом та використанням матеріальних цінностей на підприємстві. Насамперед, система повинна передбачати можливість введення, редагування, перегляду та видалення записів про матеріальні ресурси, з фіксацією їх основних характеристик: номенклатурного найменування, одиниць виміру, кількісних залишків, термінів придатності (за наявності), місць зберігання, облікових кодів, а також призначення чи належності до конкретних структурних підрозділів підприємства. Кожна одиниця матеріалу повинна бути асоційована з відповідними метаданими, включаючи джерело надходження, умови зберігання, поточний статус у логістичному ланцюгу та історію руху в межах організації.

Інформаційна система повинна забезпечувати наскрізну автоматизацію обробки запитів на видачу, повернення або списання матеріалів, а також облік внутрішніх переміщень ресурсів між підрозділами. Операції мають бути реалізовані із забезпеченням контролю за залишками на складах у режимі реального часу, з можливістю формування супровідних документів, зведених актів, накладних або звітів для керівництва. Крім того, необхідно передбачити механізми відстеження змін статусів облікових записів, контролю над нормативними рівнями запасів та створення сповіщень у випадках виявлення критичних відхилень — наприклад, зниження запасів нижче мінімального рівня або порушення умов зберігання. Вбудована система звітності має надавати змогу формувати як стандартні, так і довільні аналітичні звіти за вказаними критеріями: періодом, видами матеріалів, підрозділами, відповідальними особами, обсягами витрат чи залишків.

Окрім функціональних, до інформаційної системи висуваються низка нефункціональних вимог, які визначають її технічні, експлуатаційні та якісні

характеристики. Однією з ключових вимог є забезпечення високої продуктивності системи, що означає здатність обробляти значну кількість транзакцій одночасно без втрати швидкодії або точності розрахунків. Це особливо актуально в умовах масштабних підприємств, де кількість взаємодій із системою з боку різних користувачів досягає десятків або сотень щодня. Масштабованість системи повинна гарантувати її здатність до розширення функціоналу та нарощування обсягів даних без потреби в повній реконструкції архітектури чи падіння продуктивності. Зокрема, система має підтримувати гнучке додавання нових модулів, оновлення компонентів та інтеграцію з іншими інформаційними системами підприємства.

Особливу увагу необхідно приділити безпеці, оскільки інформаційна система оперує потенційно конфіденційними та комерційно чутливими даними. Забезпечення безпеки має реалізовуватись через обов'язкову автентифікацію користувачів, авторизацію доступу залежно від ролі в системі, використання зашифрованих каналів зв'язку, регулярне резервне копіювання бази даних, журналювання дій та систему виявлення спроб несанкціонованого втручання. Не менш важливою є зручність у використанні: система повинна мати інтерфейс, зрозумілий як для досвідчених операторів, так і для користувачів із базовими навичками роботи з інформаційними системами. Дизайн має бути логічно структурованим, із чіткою навігацією, візуальними підказками та доступом до довідкових матеріалів без потреби у тривалому навчанні персоналу.

Інформаційна система також повинна підтримувати диференційований доступ, який забезпечує розмежування повноважень між категоріями користувачів: наприклад, звичайний користувач може здійснювати лише перегляд і введення даних у межах своєї зони відповідальності, тоді як адміністратор має право редагування, управління доступом та контроль над усією системою. Реалізація цих вимог гарантує, що розроблена інформаційна система буде не лише функціонально повною, а й технологічно надійною, безпечною, зручною та здатною адаптуватися до змін у внутрішньому середовищі підприємства.

У практичній реалізації зазначених вимог важливу роль відіграє правильно спроектована архітектура інформаційної системи. Наприклад, у рамках клієнт-серверного підходу система може бути побудована з використанням C++ як основної мови для створення серверної частини, що обробляє всі запити, реалізує бізнес-логіку та забезпечує інтеграцію з реляційною базою даних. Застосування бібліотеки Crow дає змогу реалізувати легкий і продуктивний HTTP-сервер, що ідеально підходить для побудови RESTful API, через які взаємодіють веб-інтерфейси з базою даних.

Для зберігання інформації доцільним є використання SQLite — ефективної вбудованої СУБД, яка не потребує розгортання окремого серверного середовища. Це значно спрощує конфігурацію, зменшує витрати на обслуговування та дозволяє розробляти автономні рішення, особливо актуальні для невеликих підприємств або мобільних робочих місць.

Клієнтська частина, реалізована з використанням HTML, CSS та JavaScript, може бути розгорнута у вигляді вебінтерфейсу, що працює у браузері. У поєднанні з сучасними JavaScript-фреймворками (наприклад, React або Vue.js), інтерфейс може надавати розширений функціонал: фільтрацію даних, візуалізацію таблиць, миттєве оновлення даних без перезавантаження сторінки (через AJAX), а також інтерактивні панелі для перегляду аналітики в реальному часі.

Додатково до функціональних модулів система повинна містити підсистему журналювання подій та дій користувачів, яка дає змогу вести хронологічний запис усіх ключових операцій у системі: змін у даних, доступу до окремих модулів, спроб аутентифікації, помилок виконання запитів. Це не тільки підвищує прозорість використання системи, але й забезпечує відповідність внутрішнім політикам безпеки та зовнішнім вимогам аудиту.

У довгостроковій перспективі доцільно передбачити можливість масштабування функціоналу — зокрема, інтеграцію з бухгалтерським обліком, системами фінансового планування, виробничими та логістичними підсистемами. Це забезпечить перехід від ізольованої ІС управління матеріальними ресурсами до повноцінного модуля у складі єдиної цифрової екосистеми підприємства, що

дозволить досягти максимальної автоматизації і прозорості всіх ресурсних процесів.

Таким чином, правильне поєднання функціональних і нефункціональних вимог із технічно грамотно реалізованою архітектурою створює підґрунтя для побудови ефективної, гнучкої та надійної інформаційної системи, яка відповідає реальним потребам сучасного підприємства в частині управління матеріальними ресурсами.

Особливу увагу в процесі розробки доцільно приділити реалізації архітектурної моделі MVC (Model–View–Controller), яка забезпечує чітке розділення логіки даних, представлення інформації та обробки користувацьких запитів. Завдяки цьому спрощується супровід, модифікація та тестування програмного забезпечення. Компонент Model відповідає за взаємодію з базою даних, структурування та обробку матеріальних записів, View — за формування інтерфейсу користувача і подання інформації у зручному для сприйняття вигляді, а Controller — за обробку дій користувачів, маршрутизацію запитів і виклик відповідної логіки. Застосування шаблону MVC дає змогу реалізувати масштабовану та структуровану організовану систему, де зміни в одному компоненті не призводять до необхідності переписувати інші частини системи.

У контексті розробки на C++ із використанням фреймворку Crow така модель може бути реалізована через розділення коду на класи моделей, маршрутизатори (контролери) та HTML-шаблони або API-відповіді, що формують представлення. Це дозволяє гнучко керувати логікою системи, спрощувати розширення функціоналу в майбутньому та забезпечити чисту архітектуру з дотриманням принципів інженерного проєктування.

Застосування MVC-підходу, у поєднанні з клієнт-серверною структурою та сучасними вебтехнологіями, гарантує не лише відповідність високим стандартам розробки, але й створення програмного продукту, що легко адаптується до нових вимог, технологічно масштабовується та зручно підтримується у виробничих умовах.

Вибір відповідного архітектурного підходу є критично важливим етапом у процесі розробки складного програмного забезпечення, зокрема інформаційної системи для управління матеріальними ресурсами підприємства. Архітектурні патерни дозволяють стандартизувати структуру додатку, зробити її зрозумілою, гнучкою та легко масштабованою. Застосування таких патернів забезпечує високий рівень організованості коду, спрощує тестування, підтримку, а також подальший розвиток системи.

Патерн Model–View–Controller (MVC).

Один із найбільш поширених і ефективних архітектурних підходів — це Model–View–Controller (MVC). Цей патерн структурує програмний код шляхом чіткого поділу на три компоненти: Model, View та Controller, кожен з яких відповідає за окремий аспект роботи системи.

Model — компонент, який відповідає за бізнес-логіку та взаємодію з даними. У контексті інформаційної системи управління матеріальними ресурсами модель відповідає за структуру бази даних, операції над таблицями, наприклад: додавання, оновлення, пошук і видалення записів про матеріальні ресурси, постачання, залишки та облік переміщень.

View — це компонент, що забезпечує відображення даних користувачу. У нашому випадку це вебінтерфейс, побудований з використанням HTML, CSS і JavaScript або відповідних фреймворків. Представлення має бути адаптивним, зрозумілим і зручним для роботи персоналу підприємства: складу, логістики, адміністрації.

Controller — координує взаємодію між Model і View. Він отримує введення від користувача, передає ці дані моделі для обробки, а потім забезпечує оновлення представлення на основі зміненого стану даних.

Застосування MVC дозволяє розділити відповідальність між компонентами системи. Наприклад, програмісти можуть оновлювати логіку бізнес-процесів у Model без впливу на View, а дизайнери можуть змінювати інтерфейс користувача без ризику порушити логіку обробки даних. У системі, побудованій на C++ із використанням фреймворку Crow, MVC може реалізовуватися через окремі класи

моделей, маршрутизатори запитів (у ролі контролерів) і шаблони для генерації HTML або JSON-відповідей.

Патерн Layered Architecture (багаторівнева архітектура).

Ще одним важливим архітектурним підходом є багаторівнева архітектура (Layered Architecture), яка передбачає поділ системи на окремі логічні шари. Найчастіше виділяють такі рівні: презентаційний, прикладний (або логічний), рівень доступу до даних та саму базу даних. У цьому підході кожен шар взаємодіє лише з суміжним. Така структура дозволяє ізолювати зміни: наприклад, оновлення інтерфейсу користувача не впливає на рівень логіки або даних.

У розробці ІС для управління матеріальними ресурсами багаторівнева архітектура може поєднуватися з MVC. Наприклад, View належить до презентаційного рівня, Model — до шару доступу до даних, а Controller — до прикладного шару.

Патерн Repository.

У більш складних системах доцільно впроваджувати шаблон Repository, який дозволяє абстрагуватися від конкретної реалізації зберігання даних. Це означає, що бізнес-логіка системи взаємодіє не з самою базою даних напряму, а з проміжним шаром — репозиторієм, який надає уніфіковані методи доступу до сутностей. Наприклад, замість прямого SQL-запиту до таблиці materials, модуль взаємодіє з методом `getMaterialsByType(type)`. Такий підхід підвищує гнучкість системи та полегшує перехід між різними СУБД або впровадження кешування.

Патерн Dependency Injection (впровадження залежностей).

З метою підвищення тестованості та модульності важливим є застосування патерну Dependency Injection. Його суть полягає в тому, що залежності класів (наприклад, сервіси, репозиторії або конфігурації) не створюються всередині самих класів, а передаються ззовні. Це дозволяє під час тестування замінювати залежності на "заглушки", проводити ізольоване тестування окремих частин системи, а також легко змінювати реалізацію компонентів без зміни клієнтського коду.

Патерн Singleton.

Для компонентів, які повинні існувати в єдиному екземплярі у межах всієї системи (наприклад, менеджер з'єднання з базою даних), доцільно реалізовувати патерн Singleton.

У контексті ІС управління ресурсами Singleton гарантує, що протягом усього життєвого циклу системи існує лише одне підключення до бази або лише один об'єкт, який відповідає за конфігурацію доступу до сховища.

Застосування архітектурних патернів, зокрема MVC, багаторівневої архітектури, Repository, Dependency Injection та Singleton, створює потужну архітектурну основу для розробки сучасної, надійної та масштабованої інформаційної системи управління матеріальними ресурсами. Така структурна організація програмного продукту не лише забезпечує його відповідність поточним функціональним і нефункціональним вимогам, але й закладає фундамент для його подальшої еволюції — додавання нових модулів, зміни технологічної платформи або інтеграції з іншими корпоративними системами. У результаті підприємство отримує інструмент, який не лише вирішує поточні завдання обліку та контролю ресурсів, але й слугує надійною основою для цифрової трансформації в майбутньому.

3.3.2 Вибір архітектури програмного забезпечення

Вибір архітектурного підходу для розробки програмного забезпечення інформаційної системи управління матеріальними ресурсами підприємства є одним із найвідповідальніших етапів проєктування, оскільки саме архітектура визначає загальну структуру програмного продукту, взаємодію між його компонентами, масштабованість, гнучкість, безпеку та подальшу підтримку.

З огляду на поставлені функціональні цілі системи, її багатокomпонентність, а також потребу в обслуговуванні значної кількості запитів від різних категорій користувачів у реальному часі, було обрано клієнт-серверну архітектуру з підтримкою патерну MVC (Model-View-Controller). Такий підхід забезпечує чітке розділення відповідальностей між шарами системи: логікою обробки даних

(Model), інтерфейсом користувача (View) та контролем взаємодії між ними (Controller).

На серверному боці реалізацію архітектури покладено на фреймворк Crow — високопродуктивну C++ бібліотеку для створення RESTful API. Crow дозволяє організувати маршрутизацію HTTP-запитів, обробку сесій, реалізацію контролерів та взаємодію з базою даних у логічному та безпечному вигляді. Збереження даних здійснюється у реляційній БД SQLite, що дозволяє досягти достатньої продуктивності в умовах середніх навантажень і забезпечує простоту розгортання без необхідності в зовнішньому сервері БД.

Клієнтська частина, що виконує роль шару View у парадигмі MVC, реалізована за допомогою HTML, CSS та JavaScript з використанням AJAX-технологій. Це дозволяє організувати динамічний інтерфейс, що реагує на дії користувача без перезавантаження сторінки. Передбачено підтримку форм для введення, редагування, перегляду та фільтрації даних про матеріальні ресурси, а також відображення аналітичної інформації у вигляді таблиць та діаграм.

Вибір розподіленої архітектури (distributed model) з можливістю подальшого переходу до мікросервісної моделі є стратегічно обґрунтованим. Такий підхід дозволяє у майбутньому масштабувати окремі модулі системи, наприклад, модуль аналітики або звітності, без необхідності модифікації всієї системи. Архітектура також передбачає інтеграцію з зовнішніми модулями (API або сторонні сервіси), що дає змогу гнучко адаптувати ІУС під нові бізнес-процеси.

Для реалізації аутентифікації та авторизації користувачів застосовано токенований підхід із використанням JWT (JSON Web Token). Це забезпечує безпечну та масштабовану систему контролю доступу до функціональності системи на основі ролей користувачів (адміністратор, менеджер, аналітик). При кожному запиті клієнта до сервера токен передається у заголовках HTTP, верифікується на сервері та визначає права доступу.

Щодо захисту каналу передачі даних, обрана архітектура підтримує використання SSL/TLS для шифрування всіх з'єднань між клієнтською та

серверною частинами системи, що особливо важливо у випадку передачі конфіденційної інформації, зокрема даних про запаси та фінансові операції.

Крім того, архітектура передбачає реалізацію журналювання дій користувачів (AuditLog) та централізоване логування подій системи, що дає змогу адміністратору відстежувати аномальні дії, здійснювати аудит та реагувати на інциденти.

У якості альтернативної архітектурної моделі для інтерфейсної частини (View) може бути використано підхід MVVM (Model-View-ViewModel) на базі React.js. Такий варіант доцільно реалізовувати у разі потреби масштабування фронтенду або розширення інтерактивності інтерфейсу. Зокрема, ViewModel забезпечить реактивний зв'язок між даними та інтерфейсом, спрощуючи оновлення інформації без прямої взаємодії з базовими елементами DOM.

Таким чином, обрана архітектура відповідає сучасним вимогам до побудови інформаційних управляючих систем: вона є масштабованою, безпечною, модульною та гнучкою, що створює основу для подальшого розширення системи, її обслуговування та інтеграції з іншими корпоративними рішеннями підприємства.

3.3.3 Інструментальні засоби розробки для створення прикладного програмного забезпечення інформаційних управляючих систем

Розробка програмного забезпечення для управління матеріальними ресурсами є складним, багаторівневим процесом, що охоплює повний цикл створення інформаційної системи — від вибору технологічного стеку до написання коду, тестування, налаштування бази даних, створення інтерфейсів взаємодії та підготовки документації для користувачів. На кожному з етапів необхідно забезпечити відповідність розробки поставленим функціональним, технічним та експлуатаційним вимогам.

В якості основної мови програмування для реалізації серверної логіки було обрано C++ — мову, яка відзначається високою швидкодією, контрольованістю використання ресурсів та широкими можливостями для побудови продуктивних, безпечних рішень. Для побудови веб-сервера обрано легковаговий фреймворк Crow, який реалізує RESTful-підхід і дозволяє інтегрувати HTTP-інтерфейси без зайвих накладних витрат.

Для реалізації бази даних використано SQLite — вбудовану реляційну СУБД, яка дозволяє зберігати всі дані у локальному .db файлі. Цей вибір був зумовлений простотою налаштування, відсутністю необхідності розгортання окремого серверного ПЗ, а також повною відповідністю вимогам автономної або локально мережевої експлуатації системи на малих та середніх підприємствах.

Клієнтська частина реалізована з використанням HTML, CSS та JavaScript, що забезпечує кросплатформеність, адаптивність до різних пристроїв і гнучкість інтерфейсу користувача. Основна взаємодія між фронтендом і бекендом відбувається за допомогою HTTP-запитів через AJAX або Fetch API, що відповідає стандартам клієнт-серверної моделі.

Архітектура бази даних має реляційну структуру з чітким логічним поділом на сутності та встановленими зв'язками між ними. До складу БД входять наступні таблиці:

- users — облікові записи співробітників;

- resources — інформація про матеріальні ресурси;
- warehouses — список доступних для використання складів;
- warehouse_resources — деталі про ресурси та їх кількість у конкретному складі;

Реляційні зв'язки реалізовані через зовнішні ключі з правилами каскадного оновлення та заборони видалення зв'язаних записів. Це забезпечує цілісність даних на рівні структури та підтримує консистентність при виконанні транзакцій.

На рівні логіки додатка було розроблено низку REST-ендпоінтів для виконання базових операцій CRUD (Create, Read, Update, Delete) над усіма ключовими сутностями системи. Наприклад:

- POST /materials/add — додавання нового ресурсу;
- GET /materials/list — виведення списку наявних ресурсів;
- PUT /materials/update/:id — оновлення атрибутів ресурсу;
- DELETE /materials/remove/:id — видалення об'єкта з бази.

Для підвищення надійності кожна операція супроводжується валідацією вхідних даних та обробкою виключень. Логіка взаємодії з базою інкапсульована у функціональні модулі з чітко визначеним інтерфейсом.

Тестування охоплювало:

- структурну перевірку БД — правильність створення таблиць і зв'язків;
- функціональне тестування API — відповідність відповідей очікуваним результатам;
- навантажувальне тестування — оцінка продуктивності при великій кількості запитів;
- перевірка надійності — симуляція збоїв і перевірка відновлення даних.

Було створено набір тестових сценаріїв, що моделюють типові бізнес-випадки: надходження нових матеріалів, переміщення між складами, формування звітів за період, списання з причини псування тощо. На підставі тестування сформовано чек-листи й внесено відповідні коригування в кодову базу.

Також передбачена можливість розширення системи через плагіни або додаткові модулі, включно з аналітичними панелями або зовнішніми ВІ-системами.

Таким чином, розроблене програмне забезпечення реалізує повноцінну інформаційну систему для управління матеріальними ресурсами підприємства. Воно забезпечує:

- зберігання структурованої інформації;
- безпечну й швидку взаємодію з даними;
- можливість масштабування;
- адаптацію під специфіку підприємства.

Архітектура є відкритою для подальшого вдосконалення та дозволяє легко інтегрувати її з іншими підсистемами управління, формуючи основу для єдиного корпоративного цифрового середовища.

3.3.4 Тестування програмного забезпечення

Тестування програмного забезпечення інформаційної системи управління матеріальними ресурсами підприємства проводилось з метою перевірки коректності реалізації функціоналу, виявлення помилок та оцінки загальної стабільності роботи системи. Особливу увагу було приділено перевірці відповідності логіки роботи ІУС поставленим вимогам користувача, а також тестуванню стійкості системи до помилкових або непередбачуваних дій.

Тестування здійснювалося у кількох етапах: модульне тестування окремих компонентів (робота з базою даних, обробка запитів, генерація звітів), інтеграційне тестування взаємодії між модулями, а також системне тестування з імітацією типових користувацьких сценаріїв. Для перевірки коректності обробки даних використовувалися контрольні приклади з типових логістичних операцій: надходження, переміщення, списання ресурсів.

У ході тестування було задіяно як ручний, так і автоматизований підхід. Зокрема, було розроблено кілька скриптів перевірки API-запитів до серверної

частини та SQL-запитів до бази даних. Крім цього, перевірялися час відгуку на запити, коректність відображення інтерфейсу на клієнтській частині та відповідність повідомлень про помилки встановленим правилам валідації.

Результати тестування підтвердили працездатність системи, її функціональну повноту та стабільність при типових сценаріях використання. Під час тестування були також виявлені окремі незначні помилки інтерфейсної частини, які були усунуті в процесі налагодження.

3.3.5 Керівництво (інструкція) користувача

Інформаційна управляюча система для управління матеріальними ресурсами підприємства розроблена як вебзастосунок з клієнт-серверною архітектурою. Система забезпечує можливість реєстрації, обліку, контролю та аналізу руху матеріальних ресурсів компанії в режимі реального часу. Інтерфейс користувача реалізовано засобами HTML, CSS і JavaScript, а обробку запитів — за допомогою серверної частини на C++ з використанням фреймворку Crow.

Для користування системою необхідно мати сучасний браузер (Google Chrome, Mozilla Firefox, Microsoft Edge) та доступ до локального сервера або розгорнутого хостингу, де розміщено виконуваний файл сервера. Для підтримки кінцевих користувачів розроблено інструкцію з розгортання, використання та адміністрування системи.

Порядок роботи з системою.

Запуск системи.

Для активації роботи системи необхідно запустити виконуваний файл сервера (./material_system) у терміналі або командному рядку. Після запуску сервер прослуховує порт (за замовчуванням 18080) і стає доступним для взаємодії через браузер.

Авторизація.

Після відкриття вебінтерфейсу за адресою <http://localhost:18080/> користувачеві пропонується авторизуватись за допомогою логіна та пароля.

Залежно від ролі (адміністратор, оператор, аналітик) користувач отримує доступ до відповідного функціоналу.

Основні функції:

- облік матеріалів;
- управління ресурсами.
- робота із постачанням;
- формування звітності.

Навігація та інтерфейс.

Усі функції згруповані у відповідних розділах меню, доступ до яких залежить від ролі користувача. Для зручності реалізовано швидкий пошук, сортування таблиць та експорт звітів у форматі CSV.

Завершення роботи.

Після завершення сеансу рекомендується виконати вихід із системи через пункт меню «Вийти». Сервер можна зупинити за допомогою команди Ctrl+C у терміналі.

Технічна підтримка.

У разі виникнення помилок у роботі системи або потреби в оновленні, адміністратор може звернутись до розробника або технічного фахівця для оновлення серверного ПЗ або бази даних.

Керівництво користувача передбачено для типового середовища експлуатації системи в межах локальної мережі підприємства. У разі інтеграції з хмарними сервісами або масштабування системи доступ через інтернет, структура взаємодії може бути адаптована.

3.4. Технічне забезпечення

3.4.1 Обґрунтування вибору технічного забезпечення

Ефективна інформаційна система управління матеріальними ресурсами підприємства неможлива без належного технічного забезпечення. Саме від якості,

надійності, масштабованості та безпеки обраного апаратного й програмного комплексу залежить здатність системи своєчасно обробляти великі обсяги даних, забезпечувати цілісність інформації, підтримувати оперативність бізнес-процесів і адаптуватися до змін у діяльності підприємства. У межах цього етапу відбувається комплексний аналіз вимог до інфраструктури, вибір оптимального технічного рішення, його налаштування, тестування й підготовка до впровадження.

Передусім необхідно забезпечити наявність серверної інфраструктури, яка здатна підтримувати безперервну роботу системи, обробку запитів користувачів, зберігання великої кількості даних про матеріальні ресурси, а також забезпечувати безпеку та резервування інформації. Системи, що працюють із матеріальними потоками, повинні підтримувати режим реального часу або бути наближеними до нього. Це особливо важливо в умовах інтенсивної виробничої діяльності або багаторівневої логістики.

Оптимальним рішенням є використання серверів корпоративного класу від провідних виробників, таких як Dell PowerEdge, HPE ProLiant чи Lenovo ThinkSystem. Вибір конфігурації залежить від масштабу підприємства та кількості одночасних користувачів. Доцільним є використання багатоядерних процесорів, таких як Intel Xeon Scalable або AMD EPYC, обсяг оперативної пам'яті не менше 64 ГБ для середніх систем, а також використання твердотільних накопичувачів (SSD NVMe) у конфігурації RAID 10 або RAID 6 для одночасного забезпечення швидкодії та відмовостійкості.

Особливу увагу слід приділити мережевому обладнанню. Для уникнення "вузьких місць" у передачі даних між серверами, користувацькими пристроями й системами обліку, доцільним є застосування керованих комутаторів із підтримкою протоколів QoS (Quality of Service), VLAN і SNMP. У випадку, якщо підприємство має розгалужену географічну структуру (філії, склади в різних містах), доцільним є використання VPN-рішень і захищених каналів зв'язку з резервуванням.

Для забезпечення стабільного живлення серверного обладнання необхідно впровадити джерела безперебійного живлення (UPS), що підтримують як основне,

так і резервне живлення, а також дозволяють коректно завершувати роботу систем у разі аварійного вимкнення електроенергії.

Важливим елементом інфраструктури є системи резервного копіювання. У рамках забезпечення інформаційної безпеки та безперервності бізнесу слід організувати багаторівневе резервне копіювання. Воно може включати локальні бекапи на NAS-сховища, зовнішні сервери, а також хмарні сервіси. До прикладу, можна реалізувати комбіновану стратегію на базі Veeam Backup для локального копіювання та AWS S3/Glacier або Microsoft Azure Backup для віддаленого зберігання. Це дозволяє захистити дані від втрат через апаратні збої, атаки програм-вимагачів або людський фактор.

Щодо програмної частини, її підбір має відповідати обраній архітектурі інформаційної системи та підтримувати необхідні функціональні модулі: управління обліком, інвентаризацією, закупівлями, переміщенням матеріалів, формування аналітичних звітів та інтеграцією з іншими інформаційними системами.

У якості серверної платформи доцільно використовувати ОС на базі Linux (наприклад, Ubuntu Server або CentOS), яка забезпечує стабільну роботу, масштабованість, безпеку та відкритість коду. Веб-додаток можна реалізувати з використанням високопродуктивного фреймворку, наприклад, Crow для C++ або інших у залежності від технологічного стеку.

Для роботи з базами даних рекомендується використовувати PostgreSQL або SQLite, залежно від складності та навантаження на систему. PostgreSQL забезпечує гнучку схему зберігання, підтримку складних запитів, тригерів, реплікації, а також високу надійність транзакцій.

На клієнтській стороні інтерфейс користувача реалізовується через вебтехнології — HTML5, CSS3, JavaScript. Використання бібліотек, таких як Bootstrap або Materialize, дозволяє створити адаптивний інтерфейс, зручний для користувача. Для інтерактивності та роботи з API може застосовуватися AJAX, а для побудови графіків і таблиць — Chart.js або D3.js.

Окремо слід виділити вимоги до інтеграції з іншими системами підприємства — бухгалтерськими, ERP-системами, сервісами електронного документообігу. Для цього застосовуються RESTful API або SOAP-сервіси, які дозволяють забезпечити взаємодію між модулями без ручного дублювання інформації.

Безпека інформації — ще один критично важливий аспект. Необхідно впровадити багаторівневу систему доступу з ролями, зашифрованим зберіганням даних (використання AES-256), захистом від SQL-ін'єкцій та XSS-атак, а також системами моніторингу підозрілої активності (наприклад, на основі Fail2Ban чи Wazuh).

Для аналітичної обробки даних варто впровадити модулі звітності з побудовою графіків, аналізом динаміки використання ресурсів, прогнозуванням на основі історичних даних. Доцільно інтегрувати бібліотеки Pandas, NumPy, а для складнішого прогнозування — модулі машинного навчання, такі як Scikit-learn.

Наступним кроком після вибору компонентів є налаштування технічного середовища. Встановлюється серверне ПЗ, конфігурується веб-сервер (наприклад, Nginx), створюється база даних, налаштовується мережева інфраструктура з врахуванням безпеки й резервування. Окрему увагу приділяють автоматизації — застосовують скрипти розгортання (наприклад, Ansible або Docker для контейнеризації системи).

Після цього система проходить всебічне тестування: функціональне, навантажувальне, модульне, а також перевірку на вразливості. Виявлені недоліки усуваються до моменту передачі системи в експлуатацію.

На етапі впровадження система інсталується на виробничому середовищі, проводиться інтеграція з суміжними службами підприємства, організовується навчання персоналу та надається документація користувача й адміністратора. У разі потреби — налаштовується технічна підтримка та супровід.

Такий підхід до вибору й налаштування технічного забезпечення дозволяє створити стабільну, масштабовану, безпечну інформаційну систему, яка відповідає

вимогам сучасного підприємства та підтримує ефективне управління всіма етапами життєвого циклу матеріальних ресурсів.

3.4.2 Ресурсні вимоги до технічного забезпечення

Належне функціонування інформаційної системи управління матеріальними ресурсами залежить не лише від програмної реалізації, але й від технічного забезпечення, яке відіграє критичну роль у забезпеченні стабільності, надійності та продуктивності програмного середовища. Технічні ресурси повинні відповідати вимогам архітектури системи, обсягам оброблюваних даних, числу активних користувачів та необхідному рівню доступності. У цьому контексті важливо надати обґрунтований підхід до формування вимог до апаратного і мережевого забезпечення для підтримки клієнт-серверної архітектури, реалізованої засобами C++, Crow, SQLite та вебтехнологій.

Вимоги до серверного обладнання.

Серверна частина системи виконує основне навантаження, оскільки обробляє запити від клієнтів, взаємодіє з базою даних, виконує логіку обробки ресурсів та відповідає за доступ до ключових функцій. У базовій конфігурації для невеликого або середнього підприємства, що обслуговує до 50 одночасних користувачів, рекомендується використовувати сервер з такими характеристиками:

Процесор: не менше ніж 4-ядерний (Intel Xeon / AMD Ryzen / аналогічний серверний).

Оперативна пам'ять (RAM): від 16 ГБ для забезпечення буферизації, обробки запитів і зменшення навантаження на диск.

Накопичувач: SSD-накопичувач з обсягом від 256 ГБ для швидкого доступу до даних і забезпечення стабільної роботи бази даних.

Мережева карта: гігабітний Ethernet для забезпечення швидкого обміну даними з клієнтами.

Для більших підприємств з декількома підрозділами або інтенсивною взаємодією з базою даних доцільно розглядати розподілену серверну

інфраструктуру (кластер), або віртуалізацію в хмарному середовищі (наприклад, AWS EC2, Azure VM), що дозволяє масштабувати обчислювальні ресурси динамічно відповідно до навантаження.

Вимоги до клієнтських пристроїв.

Клієнтська частина інформаційної системи представлена вебінтерфейсом, реалізованим за допомогою HTML, CSS і JavaScript. Такий підхід дозволяє запускати клієнтську програму в стандартному браузері без потреби встановлення додаткового програмного забезпечення. Відповідно, технічні вимоги до клієнтських пристроїв є помірними:

- процесор: 2-ядерний (Intel Core i3 або аналогічний);
- RAM: від 4 ГБ;
- операційна система: Windows 10/11, Linux або macOS з актуальними оновленнями;
- браузер: сучасний (Google Chrome, Mozilla Firefox, Microsoft Edge) з підтримкою ES6+;
- роздільна здатність екрану: від 1366×768 для коректного відображення інтерфейсу.

Цього рівня достатньо для забезпечення комфортної роботи користувачів у підрозділах постачання, складу або адміністрації.

Вимоги до мережевої інфраструктури.

Оскільки система базується на моделі клієнт–сервер, якість мережевої взаємодії є критично важливою. Мережа повинна забезпечувати стабільний обмін даними з мінімальними затримками.

Рекомендується використовувати:

Локальну мережу з гігабітною пропускною здатністю

- комутатори 3-го рівня з підтримкою QoS, для пріоритезації критичного трафіку;
- VPN-з'єднання для віддалених підрозділів з підтримкою шифрування (IPSec, OpenVPN);
- захищені маршрутизатори з firewall-функціями.

Для забезпечення безперервної роботи бажано мати резервний канал зв'язку або балансування навантаження між провайдерами.

Вимоги до системи резервного копіювання та безпеки.

Зважаючи на важливість даних, які зберігає система (облікова, логістична та управлінська інформація про матеріальні запаси), необхідно впровадити систему резервного копіювання. Доцільно щодня створювати автоматизовані повні бекапи бази даних на зовнішньому сховищі або в хмарі (Amazon S3, Google Drive for Business тощо). Для забезпечення інформаційної безпеки повинні використовуватись шифровані канали зв'язку (HTTPS), рольова модель доступу, багаторівнева автентифікація та періодичне логування всіх операцій у системі.

Таким чином, обґрунтування ресурсних вимог до технічного забезпечення базується на балансі між ефективністю роботи системи, стабільністю її функціонування та доступністю обслуговування. Запропонована конфігурація є масштабованою та адаптивною до змін у структурі підприємства, що забезпечує її придатність як для поточних задач, так і для подальшого розширення системи в міру зростання бізнесу.

3.5 Реалізація інформаційної управляючої системи

Реалізація інформаційної системи з управління матеріальними ресурсами компанії є важливим кроком у підвищенні ефективності операційних процесів і покращенні управління ресурсами. Процес реалізації включає кілька етапів, від проектування системи до її впровадження та оцінки результатів. Ступінь реалізації або готовності системи можна оцінити за кількома ключовими показниками.

На першому етапі було проведено детальне проектування бази даних, яка є основою інформаційної системи. База даних включає кілька таблиць, які зберігають інформацію про товари, склади, користувачів та облік. Кожна таблиця була

ретельно спроектована з урахуванням вимог системи, що забезпечує цілісність і надійність даних.

Після завершення етапу проектування була розроблена інтелектуальна система аналізу даних і база знань. Ці компоненти дозволяють здійснювати глибокий аналіз даних, виявляти закономірності в поведінці клієнтів, проводити сегментацію клієнтів і прогнозувати потреби в ресурсах. Це значно покращує процес прийняття рішень і дозволяє ефективно управляти матеріальними ресурсами та вести їх подальший облік.

Система також включає модулі для управління ресурсами, моніторингу, аналітики та обліку, забезпечення безпеки даних та інтеграції з іншими системами підприємства. Кожен модуль був розроблений і протестований окремо, щоб забезпечити його ефективність і надійність. Реалізація цих модулів дозволяє автоматизувати багато рутинних операцій і зменшити ймовірність помилок, що підвищує загальну продуктивність підприємства.

Що стосується ступеня реалізації або готовності системи, на даний момент основні компоненти системи розроблені і протестовані. Система готова до впровадження на підприємстві, хоча деякі модулі можуть вимагати подальшого налаштування відповідно до специфічних вимог користувачів. Це стосується, наприклад, налаштування інтерфейсів користувача або інтеграції з існуючими системами підприємства.

Економічні результати впровадження системи можна оцінити за кількома показниками. По-перше, система дозволяє значно знизити витрати на управління ресурсами завдяки більш точному прогнозуванню потреб і зменшенню надлишкових запасів. Це також знижує витрати на зберігання товарів і зменшує ризики пов'язані з псуванням або втратами товарів.

По-друге, автоматизація процесів управління замовленнями і постачаннями дозволяє скоротити час обробки замовлень і знизити операційні витрати. Це підвищує задоволеність клієнтів і сприяє збільшенню продажів. Крім того, система дозволяє виявляти найбільш використовувані склади і ресурси, що допомагає ефективніше проводити облік і збільшувати прибутковість бізнесу.

Результати реалізації інформаційної системи (рис 3.6 – 3.10)

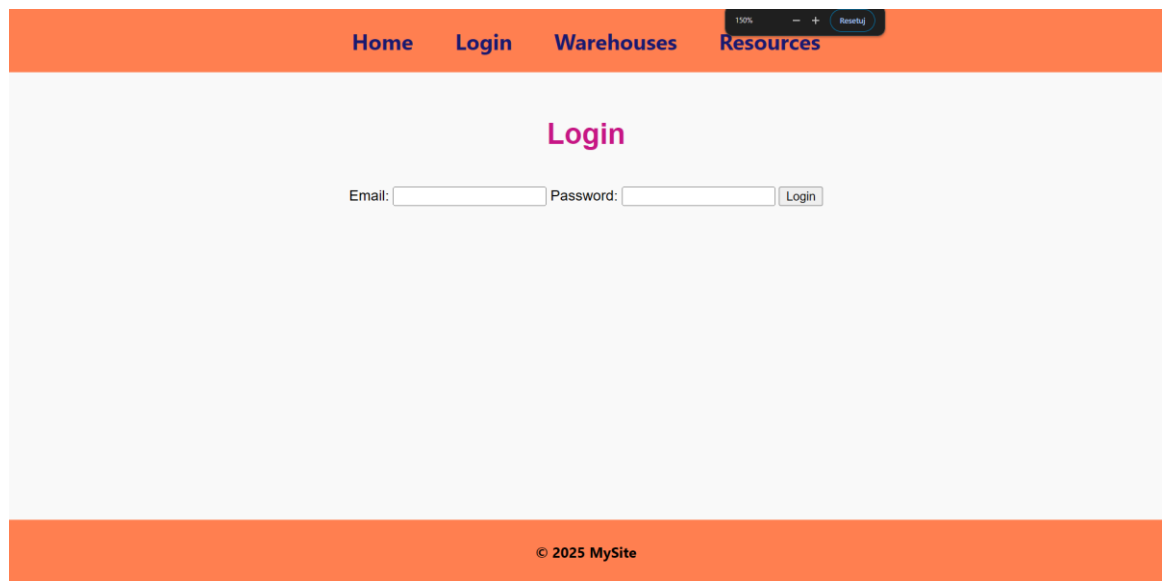


Рисунок 3.6 – Сторінка Login

Джерело: сформовано автором на основі виконаного дослідження

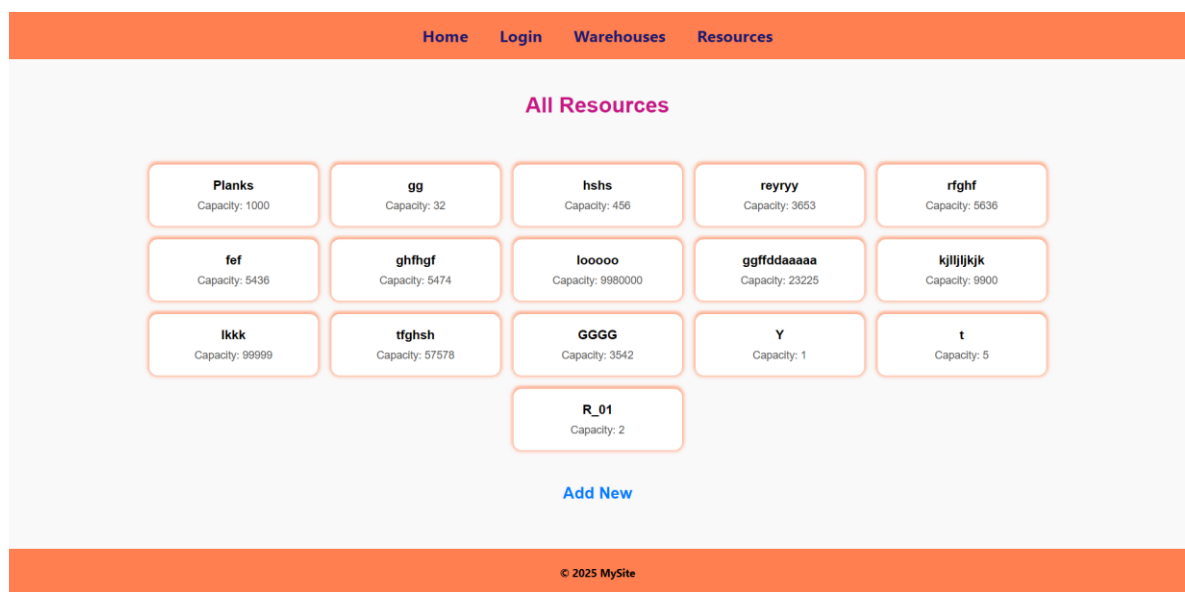


Рисунок 3.7 - Сторінка Resources

Джерело: сформовано автором на основі виконаного дослідження

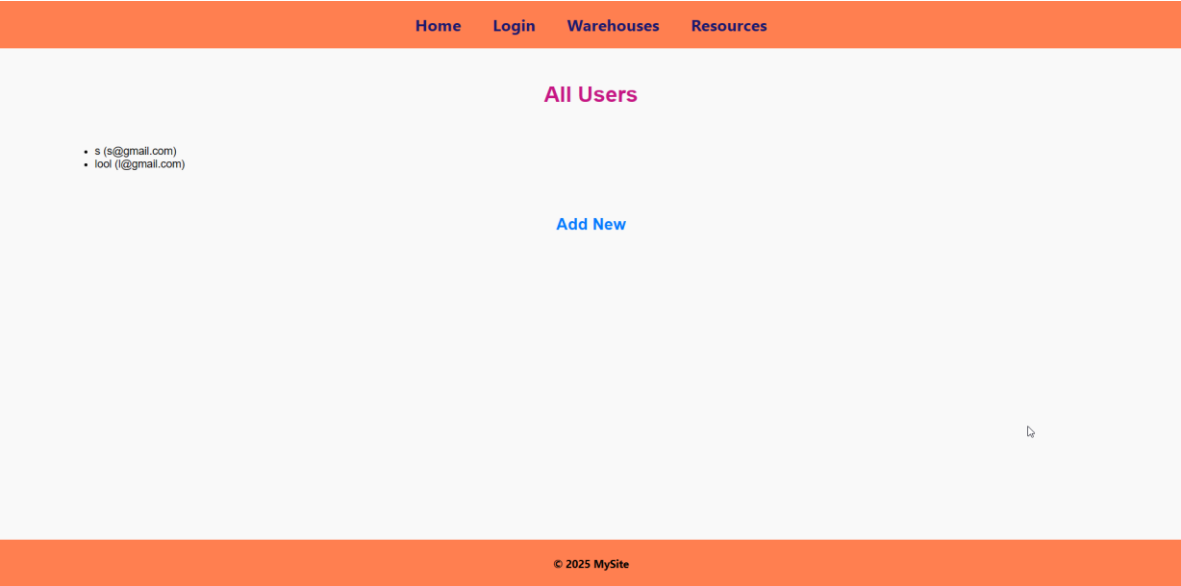


Рисунок 3.8 - Сторінка Users

Джерело: сформовано автором на основі виконаного дослідження

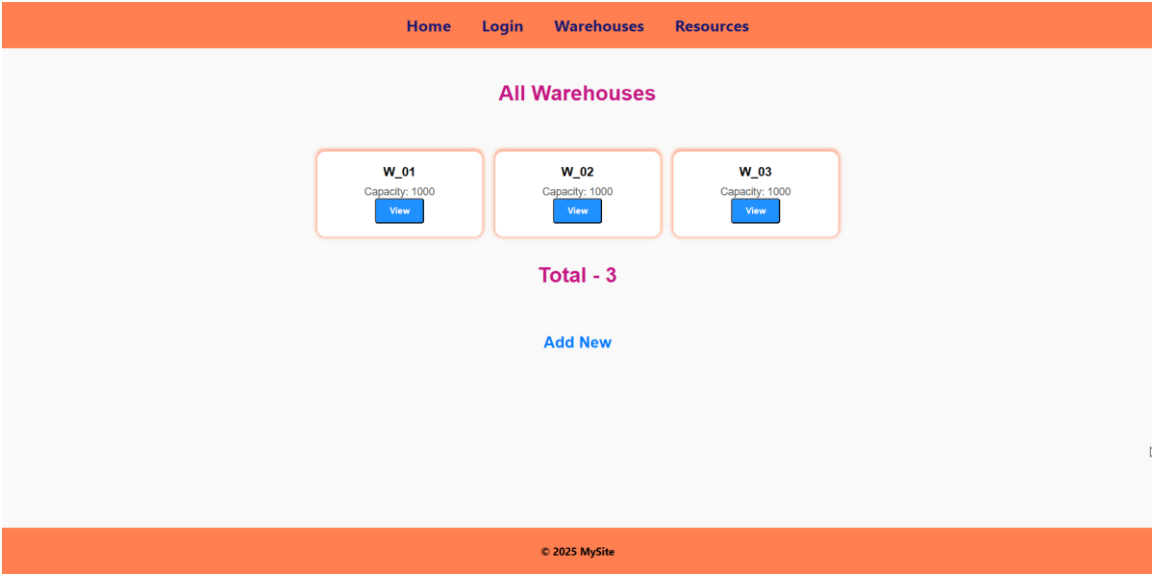


Рисунок 3.9 - Сторінка Warehouses

Джерело: сформовано автором на основі виконаного дослідження

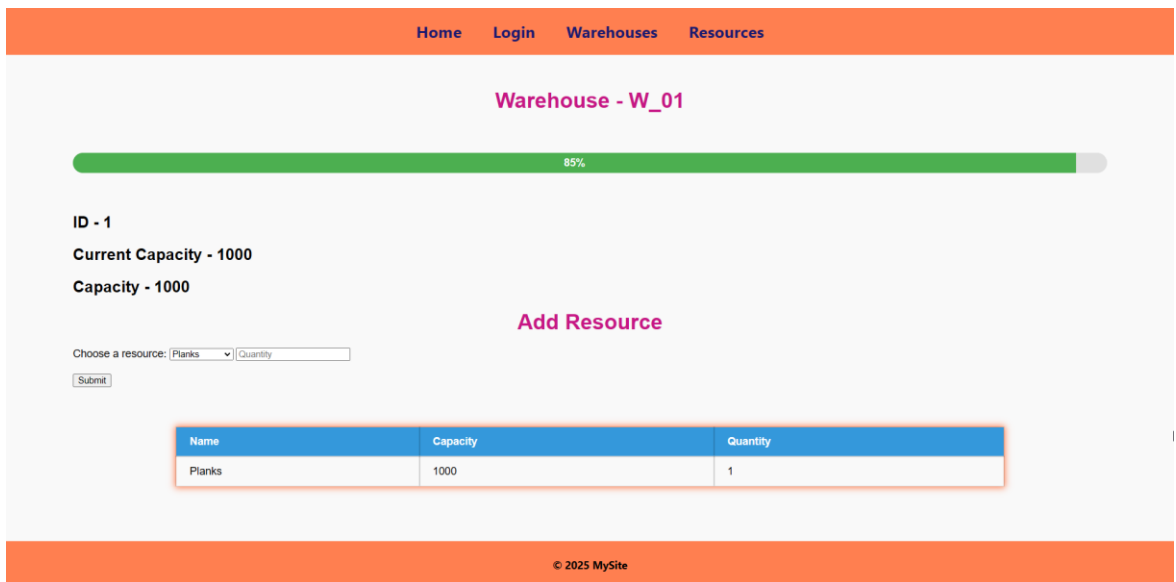


Рисунок 3.10 - Сторінка Warehouse

Джерело: сформовано автором на основі виконаного дослідження

З точки зору ефективності системи, результати апробації показали, що система значно покращує точність і швидкість обробки даних. Це дозволяє приймати більш обґрунтовані рішення і швидко реагувати на зміни в ринковій ситуації. Система також забезпечує високу надійність і безпеку даних, що важливо для захисту конфіденційної інформації клієнтів і підприємства.

Економічна доцільність впровадження системи також підтверджується результатами тестування. Система дозволяє знизити витрати і підвищити прибутковість бізнесу, що робить її впровадження економічно обґрунтованим і вигідним. Загалом, реалізація інформаційної системи з управління матеріальними ресурсами компанії є успішним кроком до підвищення ефективності операцій і покращення управління ресурсами.

ВИСНОВКИ

У процесі виконання магістерської роботи на тему «Розроблення ІУС для управління матеріальними ресурсами компанії» було досягнуто низки вагомих результатів, які підтверджують ефективність і доцільність впровадження розробленої системи в діяльність сучасного підприємства. В умовах динамічного розвитку ринку, посилення конкуренції та необхідності раціонального використання матеріальних ресурсів, автоматизація процесів їх обліку, контролю та аналітики стає стратегічно важливим завданням. Побудова сучасної, надійної, масштабованої інформаційної системи дозволила оптимізувати управління товарно-матеріальними потоками, мінімізувати втрати, пов'язані з людським фактором, підвищити швидкість обробки запитів та значно покращити прозорість логістичних операцій. Як результат — підвищення загальна ефективність управлінських процесів на підприємстві.

У рамках дослідження проведено всебічний аналіз чинних підходів до управління матеріальними ресурсами в контексті економічної діяльності підприємств. На перше місце виходить контроль якості, термінів придатності, умов зберігання продукції та безперебійність обліку. Встановлено, що одним із ключових факторів успішного управління матеріальними потоками є своєчасне отримання достовірної інформації про наявність, переміщення та потребу в ресурсах. На основі цього було розроблено концептуальну модель інформаційної системи, яка відповідає сучасним вимогам і включає в себе не лише облік, а й аналітичні інструменти для прийняття управлінських рішень.

Ретельний аналіз функціональних і нефункціональних вимог дав змогу обґрунтувати доцільність використання клієнт-серверної архітектури, яка забезпечує ефективний розподіл навантаження, підвищену безпеку даних і гнучкість масштабування. Серверна частина, реалізована мовою програмування C++ з використанням високопродуктивного фреймворку Crow, виконує обробку запитів, бізнес-логіку та управління базою даних. Обрано SQLite як легку,

вбудовану СУБД, що повністю задовольняє потреби системи в контексті локального зберігання з високою швидкістю доступу до даних та мінімальними вимогами до ресурсів. Завдяки цьому забезпечено стабільну роботу системи навіть на апаратно обмежених платформах.

Фронтенд-середовище системи реалізовано з використанням HTML, CSS та JavaScript. Такий підхід гарантує кросбраузерність, адаптивність до різних пристроїв, а також зручність і швидкість взаємодії з інтерфейсом. Зокрема, розроблений користувацький інтерфейс забезпечує швидкий доступ до ключових функцій системи, інтуїтивно зрозумілу навігацію та можливість персоналізації відповідно до ролі користувача.

Особлива увага приділялася питанням інформаційної безпеки. У системі реалізовано багаторівневу модель доступу, яка передбачає автентифікацію та авторизацію користувачів, контроль прав доступу до окремих функцій і даних, шифрування конфіденційної інформації та журналювання дій. Для захисту від втрати даних передбачено механізми автоматичного резервного копіювання. Усе це дозволяє гарантувати високий рівень надійності, відповідність законодавчим вимогам щодо захисту персональних і корпоративних даних, а також забезпечити сталу роботу системи навіть у разі виникнення позаштатних ситуацій.

У ході реалізації проєкту були спроектовані основні функціональні модулі, зокрема: модуль обліку матеріальних ресурсів, модуль управління залишками та логістичними процесами, а також модулі інтеграції з іншими інформаційними системами підприємства. Розробка бази даних здійснювалася з урахуванням нормалізації, забезпечення цілісності, зв'язності та можливості швидкої модифікації структури при зміні бізнес-процесів. Завдяки модульній архітектурі система має високий потенціал масштабування та подальшого розвитку без порушення основного функціоналу.

Система була піддана комплексному тестуванню, яке включало перевірку на відповідність функціональним вимогам, стійкість до навантажень, відмовостійкість, коректність обробки виключних ситуацій та швидкодію. Результати тестування показали, що розроблена система є стабільною, надійною та

готовою до використання в умовах реального підприємства. Особливо слід відзначити покращення у швидкості обробки запитів, зниження кількості помилок у процесі обліку ресурсів та зростання прозорості управлінських дій.

Впровадження інформаційної системи на підприємстві дозволить не лише автоматизувати облік матеріальних ресурсів, а й створити єдине інформаційне середовище для взаємодії підрозділів, що відповідають за постачання, зберігання та використання матеріалів. Це сприяло підвищенню точності прогнозування потреб у ресурсах, зниженню обсягів надлишкових запасів, оптимізації витрат і покращенню оперативності прийняття управлінських рішень. Позитивні зміни, зафіксовані в результаті апробації системи, свідчать про її значний потенціал для впровадження на інших підприємствах з аналогічними логістичними потребами.

Таким чином, виконана магістерська робота повною мірою підтвердила гіпотезу щодо ефективності застосування сучасних інформаційних технологій у сфері управління матеріальними ресурсами. Запропоноване рішення поєднує у собі гнучкість, безпеку, високу продуктивність та відповідність специфіці діяльності підприємства. У перспективі система може бути доповнена новими модулями, зокрема для інтеграції з мобільними додатками, використанням хмарних технологій, застосуванням аналітики на основі штучного інтелекту тощо.

Отже, розроблена інформаційна система є не лише завершеним технічним продуктом, але й основою для подальшої цифрової трансформації підприємства у напрямку побудови повноцінного інформаційного середовища управління матеріальними ресурсами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Artificial intelligence as an enabler for achieving human resource resiliency: past literature, present debate and future research directions / G. Panda, M. Dash, A. Samadhiya, A. Kumar, E. Mulat-weldemeskel // International Journal of Industrial Engineering and Operations Management. — 2023. — <https://doi.org/10.1108/ijieom-05-2023-0047>.
2. Гнатська Т. М., Яковенко А. О., Котик Н. М. ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПОТРЕБ ОБЛІКУ СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВ. *Економічний вісник Причорномор'я*. 2023. № 4. С. 16–28.
3. Застосування технологій штучного інтелекту. «Делойт» в Україні. «Делойт» в Україні. URL: <https://www2.deloitte.com/ua/uk/pages/about-deloitte/articles/ce-cfo-survey-automation.html> (дата звернення: 25.03.2024).
4. Барановська С. П. *Інформаційні системи в управлінні підприємством*.
5. Малахов В. Н. *Проектування інформаційних систем*.
6. Вітлінський В. В., Бланк І. А. *Системний аналіз у бізнесі*.
7. Олійник О. В. *Системи підтримки прийняття рішень*.
8. Ковальчук А. В. *СУБД та SQL: проектування і реалізація баз даних*.
9. Ковбасюк Ю. І. *Бази знань та експертні системи*.
10. Климчук О. *Архітектура інформаційних систем*.
11. Литвиненко І. О. *Інформаційні технології в менеджменті*.
12. Лещенко М. А. *Проектування баз даних. Нормалізація*.
13. Войцеховський М. — *Програмне забезпечення інформаційних систем*.
14. Crow C++ Microframework Documentation - <https://crowcpp.org/master/>
15. SQLite Documentation - <https://sqlite.org/docs.html>
16. W3C HTML & JavaScript Standards - <https://www.w3.org/>
17. PostgreSQL Official Docs - <https://www.postgresql.org/docs/>
18. MDN Web Docs – JavaScript - <https://developer.mozilla.org/en-US/>

19. ReactJS Documentation - <https://react.dev/>
20. OpenAI - <https://openai.com/research/>
21. ERDPlus - <https://erdplus.com/>
22. dbdiagram.io - <https://dbdiagram.io/home>
23. Mitchell T. *Machine Learning*.
24. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*.
25. Hastie T., Tibshirani R., Friedman J. *The Elements of Statistical Learning*.
26. Han J., Kamber M. *Data Mining: Concepts and Techniques*.
27. Murphy K. *Probabilistic Machine Learning*.
28. Литовченко, І. В. (2018). Інформаційні системи і технології в управлінні організаціями. Дніпро: ДНУ
29. Мазур, В. А. (2019). Інформаційні системи і технології на підприємствах. Київ: ВД "Київський університет".
30. Михайленко, О. М. (2020). Інформаційні технології в управлінні підприємствами. Одеса: ОНПУ
31. Cimini, C., Pirola, F., Pinto, R., & Cavalieri, S. (2017). A Human-in-the-loop Manufacturing Control Architecture for the next generation of Production Systems. IFAC-PapersOnLine, 50(1), 12229-12234.
32. Dalenogare, L. S., Benitez, G. B., Ayala, N. F., & Frank, A. G. (2018). The expected contribution of Industry 4.0 technologies for industrial performance. International Journal of Production Economics, 204, 383-394.
33. Davies, R. (2015). Industry 4.0: Digitalisation for productivity and growth. European Parliamentary Research Service, 10(11), 12-23.
34. Hermann, M., Pentek, T., & Otto, B. (2016). Design principles for industrie 4.0 scenarios: A literature review. Technische Universität Dortmund, Dortmund, Germany
35. Zadeh L. *Fuzzy Sets and Systems*
36. Schmidhuber J. *Deep Learning in Neural Networks: An Overview*.
37. Quinlan J. *Induction of Decision Trees*.

38. Tan P.-N., Steinbach M., Kumar V. *Introduction to Data Mining*.
39. Witten I. H., Frank E. *Data Mining: Practical Machine Learning Tools*.
40. Gamma E., Helm R., Johnson R., Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*.
41. Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18-23.
42. Liao, Y., Deschamps, F., Loures, E. de F. R., & Ramos, L. F. P. (2017). Past, present and future of Industry 4.0—a systematic literature review and research agenda proposal. *International Journal of Production Research*, 55(12), 3609-3629.
43. Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6, 1-10.
44. Müller, J. M., Buliga, O., & Voigt, K. I. (2018). Fortune favors the prepared: How SMEs approach business model innovations in Industry 4.0. *Technological Forecasting and Social Change*, 132, 2-17.
45. Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, 92(11), 64-88.
46. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber-physical systems: The next computing revolution. *Proceedings of the 47th Design Automation Conference*, 731-736.
47. Schmidt, R., Möhring, M., Härting, R. C., Reichstein, C., Neumaier, P., & Jozinović, P. (2015). Industry 4.0-potentials for creating smart products: Empirical research results. *Business Information Systems*, 663-674.
48. Schuh, G., Anderl, R., Gausemeier, J., ten Hompel, M., & Wahlster, W. (2017). *Industrie 4.0 maturity index: Managing the digital transformation of companies*. acatech—National Academy of Science and Engineering.
49. Wang, S., Wan, J., Li, D., & Zhang, C. (2016). Implementing smart factory of Industrie 4.0: an outlook. *International Journal of Distributed Sensor Networks*
50. Поліщук, С. М. (2017). Системи управління підприємством. Одеса:

ДОДАТКИ

Додаток А

```
#include <iostream>

#include "src/App.h"
#include "src/Database/DatabaseManager.h"
#include "controllers/UserController.h"
#include "View/UserView.h"

#include "crow.h"
#include "src/sqlite3.h"

#include <fstream>
#include <sstream>
#include "controllers/ResourceController.h"
#include "controllers/LoginController.h"
#include "controllers/WarehouseController.h"

std::string readFile(const std::string& path)
{
    std::ifstream file(path);
    std::stringstream buffer;
    buffer << file.rdbuf();
    return buffer.str();
}

void StartCrow()
{
    crow::SimpleApp crowApp;

    //APP::Database::DatabaseManager db("MyDB.db");

    std::shared_ptr<VIEW::MainView> mainView = std::make_shared<VIEW::MainView>();

    CONTROLLERS::LoginController loginController(mainView);
    CONTROLLERS::UserController userController(mainView);
    CONTROLLERS::ResourceController resourceController(mainView);
    CONTROLLERS::WarehouseController warehouseController(mainView);

    //-----

    CROW_ROUTE(crowApp, "/frnt/static/<string>")
    ([&](const std::string& filename) {
        std::string path = "frnt/static/style.css";
        std::ifstream file(path);

        if (!file.is_open())
            return crow::response(404);

        std::ostringstream contents;
        contents << file.rdbuf();

        crow::response res(contents.str());
        res.set_header("Content-Type", "text/css"); // для CSS
        return res;
    });

    CROW_ROUTE(crowApp, "/frnt/js/<string>")
    ([&](const std::string& filename) {
        std::string path = "frnt/js/loader.js";
        std::ifstream file(path);
```

```

        if (!file.is_open())
            return crow::response(404);

        std::ostringstream contents;
        contents << file.rdbuf();

        crow::response res(contents.str());
        res.set_header("Content-Type", "application/javascript"); // для CSS
        return res;
    });

CROW_ROUTE(crowApp, "/frnt/templates/<path>")
    ([&](const std::string& filename) {
        std::string path = "frnt/templates/" + filename;
        std::ifstream file(path);

        if (!file.is_open())
            return crow::response(404);

        std::ostringstream contents;
        contents << file.rdbuf();

        crow::response res(contents.str());
        res.set_header("Content-Type", "text/html"); // для CSS
        return res;
    });
//-----

CROW_ROUTE(crowApp, "/form")([&userController] {

    auto result = userController.RenderForm();

    return crow::response(result);
});

CROW_ROUTE(crowApp, "/login")([&loginController] {

    auto result = loginController.RenderForm();

    return crow::response(result);
});

CROW_ROUTE(crowApp, "/submitLogin").methods("POST"_method)([&](const crow::request& req) {

    auto result = loginController.HandleFormSubmission(req);

    return result;
});

CROW_ROUTE(crowApp, "/formResource")([&] {

    auto result = resourceController.RenderForm();

    return crow::response(result);
});

CROW_ROUTE(crowApp, "/submit").methods("POST"_method)([&](const crow::request& req) {

    auto result = userController.HandleFormSubmission(req);

    return result;
});

CROW_ROUTE(crowApp, "/submitRes").methods("POST"_method)([&](const crow::request& req) {

    auto result = resourceController.HandleFormSubmission(req);

```

```

        return result;
    });

    CROW_ROUTE(crowApp, "/formWarehouse")([&] {
        auto result = warehouseController.RenderForm();
        return crow::response(result);
    });

    CROW_ROUTE(crowApp, "/warehousePage").methods("POST"_method)([&](const crow::request& req)
{
    auto result = warehouseController.RenderWarehouse(req);
    return crow::response(result);
});

    CROW_ROUTE(crowApp, "/addResToWar").methods("POST"_method)([&](const crow::request& req) {
        auto result = warehouseController.AddResourceToWarehouse(req);
        return crow::response(result);
    });

    CROW_ROUTE(crowApp, "/submitWarehouse").methods("POST"_method)([&](const crow::request&
req) {
        auto result = warehouseController.HandleFormSubmission(req);
        return result;
    });

    CROW_ROUTE(crowApp, "/getWarehouses")([&] {
        auto result = warehouseController.ListWarehouses();
        return result;
    });

    CROW_ROUTE(crowApp, "/users")([&] {
        auto result = userController.ListUsers();
        return result;
    });

    CROW_ROUTE(crowApp, "/getResources")([&] {
        auto result = resourceController.ListResources();
        return result;
    });

    CROW_ROUTE(crowApp, "/user")([&]
    {
        return crow::response(readFile("frnt/templates/user.html"));
    });

```

```

        // Route to serve the main HTML page
        CROW_ROUTE(crowApp, "/")([&] {
            return crow::response(readFile("frnt/index.html"));
        });

    crowApp.port(3000).multithreaded().run();
}

int main()
{
    APP::App app;
    app.Start();

    StartCrow();

    return 0;
}

#ifndef RESOURCE_CONTROLLER_H
#define RESOURCE_CONTROLLER_H

#include "../src/Resource.h"
#include "../View/ResourcesView.h"

#include <crow.h>

namespace CONTROLLERS
{
    class ResourceController
    {
    public:
        ResourceController(std::shared_ptr<VIEW::MainView> main)
        {
            m_mainView = main;
            m_resourcesView.SetMainView(main);
        }

        std::string RenderForm();

        crow::response HandleFormSubmission(const crow::request& req);

        crow::response ListResources();

    private:
        APP::Resource m_resource;
        VIEW::ResourcesView m_resourcesView;

        std::shared_ptr<VIEW::MainView> m_mainView;
    };
}

#endif // !RESOURCE_CONTROLLER_H

#include "ResourceController.h"

std::string CONTROLLERS::ResourceController::RenderForm()
{
    return m_resourcesView.RenderResource();
}

crow::response CONTROLLERS::ResourceController::HandleFormSubmission(const crow::request& req)
{
    auto params = req.get_body_params();
    std::string name = params.get("name") ? params.get("name") : "";
}

```

```
std::string capacity = params.get("capacity") ? params.get("capacity") : "";
m_resource.InsertResource(name, std::stoi(capacity));

std::string html = m_mainView->GetView().load_file("frnt/formResource.html");

return crow::response{ html };
}

crow::response CONTROLLERS::ResourceController::ListResources()
{
    auto resources = m_resource.GetAllResources();
    std::string html = m_resourcesView.GetAllResources(resources);

    return crow::response{ html };
}
```



Звіт подібності

метадані

Назва організації

Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок

Розроблення інформаційної системи для управління матеріальними ресурсами компанії

Автор

Науковий керівник / Експерт

Ярцев Семен ДмитровичГалузинський Георгій Петрович

підрозділ

кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

Довжина фрази для коефіцієнта подібності 2



КП 2

14194

Кількість слів



КЦ

117677

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		23
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		70

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копію тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	424 2.99 %
2	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	232 1.63 %
3	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	181 1.28 %
4	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	119 0.84 %
5	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	81 0.57 %

6	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	78 0.55 %
7	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	56 0.39 %
8	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	53 0.37 %
9	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	51 0.36 %
10	https://ir.kneu.edu.ua/bitstreams/725b64da-359e-479b-b2e4-50ed2d83d86c/download	43 0.30 %
з бази даних RefBooks (0.14 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИФІЧНИХ СЛІВ (ФРАГМЕНТІВ)
джерело: Paperity		
1	DIGITALIZATION OF THE HOSPITALITY SPHERE AS THE BASIS FOR CREATING A COMPETITIVE BUSINESS Yurii Hurtovyi, Botsian Tetiana, Ruslan Fostolovych, Valentyna Fostolovych;	20 (4) 0.14 %
з домашньої бази даних (0.27 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИФІЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Інформаційна управляюча система інтелектуального підбору персоналу на підприємстві 6/16/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	22 (2) 0.15 %
2	Інформаційна система он-лайн бізнес-планування 6/19/2018 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	10 (1) 0.07 %
3	Інформаційна управляюча система обліку мешканців студентського містечка 6/22/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	6 (1) 0.04 %
з програми обміну базами даних (0.24 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИФІЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	2024_015.39_Труш Дмитро Миколайович 12/14/2024 Lutsk National Technical University (Lutsk National Technical University)	14 (2) 0.10 %
2	Курсова робота ФБ-35 Яковенко Анастасія 12/16/2024 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute course papers (ФТІ, К-ра інформаційної безпеки)	14 (2) 0.10 %
3	IT-M24-11353445 12/29/2024 Bohdan Khmelnytsky National University of Cherkasy (Кафедра прикладної математики і інформатики)	6 (1) 0.04 %
з Інтернету (13.00 %)		
ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИФІЧНИХ СЛІВ (ФРАГМЕНТІВ)

1	https://ir.kneu.edu.ua/bitstreams/6e5d65c5-8134-40f6-9310-dec1b2ef2e8d/download	1470 (23) 10.36 %
2	https://ir.kneu.edu.ua/bitstreams/725b64da-359e-479b-b2e4-50ed2d83d86c/download	107 (6) 0.75 %
3	https://ir.kneu.edu.ua/bitstreams/6ccdb206-8499-47f8-b15b-5d42b9b14ce2/download	89 (8) 0.63 %
4	https://kneu.edu.ua/userfiles/Department_of_Administration_and_Marketing_Personn/MM_cshod_o_vikonannya_KBD0A0.pdf	34 (2) 0.24 %
5	http://diqplus.com.ua/metodicheskiye-ukazaniya-i-informatsiya/article_post/informatsiyi-upravyayuchi-sistemi-ta-tehnologii	28 (4) 0.20 %
6	https://zhalk.zt.ua/wp-content/uploads/2024/04/metodichka_diplom_programisti.pdf	27 (2) 0.19 %
7	https://ir.kneu.edu.ua/bitstreams/f2d2da61-14eb-4ba9-b82f-2663cc9ae485/download	21 (2) 0.15 %
8	https://kneu.edu.ua/userfiles/Faculty_of_Information_Systems_and_Technology/kaf%20ise/Metod_rekomendacii_8_050101.doc	21 (2) 0.15 %
9	http://ir.znau.edu.ua/bitstream/123456789/10602/1/Pekh_LI_KR_O_071_2020.pdf	15 (1) 0.11 %
10	https://managementmania.com/en/hifo-highest-in-first-out	11 (1) 0.08 %
11	https://essuir.sumdu.edu.ua/bitstream/123456789/92885/1/Voytenko_bac_rob.pdf	11 (1) 0.08 %
12	https://uchika.in.ua/metodichni-rekomendaciyi-shodo-vikonannya-magistersekoyi-diplo.html?page=3	11 (1) 0.08 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------