

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту, назва факультету (відділення))

Кафедра обчислювальної техніки
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Комп'ютерна система керування виробництвом матеріалів для зварювальних робіт»

08-54.МКР.001.00.000 ПЗ

Виконав: студент 2-го курсу, групи
1КІ-23м спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва напрямку підготовки, спеціальності)

Босак І.С.
(прізвище та ініціали)

Керівник: к. т. н., доц. каф. ОТ

Савицька Л.А.
(прізвище та ініціали)

«___» _____ 2024 р.

Опонент: д.т.н., доц. каф.

Шиян А.А.
(прізвище та ініціали)

«___» _____ 2024 р.

Допущено до захисту
Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.
(прізвище та ініціали)

«___» _____ 2024 р.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
 Факультет інформаційних технологій та комп'ютерної інженерії
 Кафедра обчислювальної техніки
 Рівень вищої освіти II-й (магістерський)
 Галузь знань — 12 «Інформаційні технології»
 Спеціальність — 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ
 д.т.н., професор Азаров О.
 “18” вересня 2024 року



ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТА

Босаку Іллі Сергійовичу

1 Тема роботи: «Комп'ютерна система керування виробництвом ма-
 зварювальних робіт.», Керівник роботи Савицька Л.А. к.т.н., доцент к-
 затверджені наказом вищого навчального закладу від “17” вересня 2024

2 Термін подання студентом роботи 20.12.2024

3 Вихідні дані до роботи: глобальна комп'ютерна мережа; операці-
 Windows; підтримка усіх доступних браузерів; протокол передачі да-
 фреймворк клієнта jquery.js, оперативна пам'ять 2 GB.

4 Зміст текстової частини

— аналіз підходів до автоматизації логістичних цілей в системах п-
 ресурсами підприємства;

— технології розробки модуля «Волічіння зварювального дроту по-

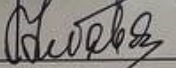
— вибір технологій для реалізації підсистеми управління логістич-
 цілями ERP «ПлазмІС»;

— розробка програмного модуля «Волічіння зварювального дроту
 попереднє».

- діаграма варіантів використання;
- діаграма діяльності для серверної частини;
- діаграма послідовностей процесів;
- загальний вигляд інтерфейсу модуля.

6 Консультанти розділів роботи показані в таблиці 1

Таблиця 1— Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпи
		завдання видав
Спеціальна частина	к.т.н., доцент кафедри ОТ Савицька Л.А.	
Економічна частина	Небава М. І., доцент, к.е.н., професор каф. ЕПВМ	

7 Дата видачі завдання 18.09.2024 . Календарний план виконання наведено в таблиці 2.

Таблиця 2 — Календарний план

№	Назва та зміст етапу	Термін виконання	
		початок	закінчення
1.	Вибір, узгодження та затвердження теми МКР	19.09.2024	19.09.2024
2.	Дослідження предметної області	20.09.2024	25.09.2024
3.	Обґрунтування методів реалізації та контролю	26.09.2024	05.10.2024
4.	Розробка структури та функціональної схеми	06.10.2024	26.10.2024
5.	Розробка програмного забезпечення	11.11.2024	11.11.2024
6.	Тестування та перевірка припущень	12.11.2024	12.11.2024
7.	Підтвердження економічної частини	15.11.2024	16.11.2024
8.	Оформлення матеріалів до захисту МКР	17.11.2024	17.12.2024

АНОТАЦІЯ

УДК 004.4:234

Босак І.С. Комп'ютерна система керування виробництвом матеріалів для зварювальних робіт. Магістерська кваліфікаційна робота зі спеціальності 123 – Комп'ютерна інженерія. Вінниця: ВНТУ, 2024. 111 с.

На укр.мові. Бібліогр.: 37 назв; рис.:20; табл.: 6.

У магістерській кваліфікаційній роботі розроблено модуль для керування виробництвом зварювального дроту у системі управління ресурсами підприємства (ERP) ПлазмІС. Реалізація програмного модуля використовує такі технології, як Oracle APEX, JavaScript, JQuery, PL/SQL і інші. Робота включає опис створення інтерфейсу модуля та взаємодії з базою даних, а також реалізацію процесів додавання, редагування та видалення даних. Також проведено аналіз витрат на розробку та оцінку економічного ефекту від можливого впровадження запропонованої системи.

Ключові слова: логістика, заявка на перевезення, програмне забезпечення.

ANNOTATION

Bosak I.S. Computer System for Managing Material Production for Welding Works. Master's Thesis in the specialty 123 – Computer Engineering. Vinnytsia: VNTU, 2024. 111 c.

In the Ukrainian language. Bibliography: 37 titles; Fig.: 20; tab.: 6. The master's thesis develops a module for managing the production of welding wire within the enterprise resource planning (ERP) system PlazmIS. The software implementation of the module is developed using technologies such as Oracle APEX, JavaScript, JQuery, PL/SQL, and others. The thesis covers the creation of the module's interface and interaction with the database, implementing processes for adding, editing, and deleting data. The costs of development are calculated, and an economic impact assessment is conducted for the potential implementation of the proposed system.

Keywords: logistics, application for transportation, software.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПІДХОДІВ ДО СИСТЕМ КЕРУВАННЯ ВИРОБНИЦТВОМ МАТЕРІАЛІВ ДЛЯ ЗВАРЮВАЛЬНИХ РОБІТ.....	11
1.1 Загальні відомості про інформаційні ERP – системи та їх приклади..	11
1.2 Огляд відомих програм для автоматизації логістичних цілей в ERP системах.....	20
1.2.1 Система SAP Extended Warehouse Management.....	20
1.2.2 Система Oracle Transportation Management.....	22
1.2.3 Система Microsoft Dynamics 365 Supply Chain Management.....	23
1.3 Особливості інформаційної ERP системи «ПлазмІС» та вимоги до її виробництва зварювального дроту.....	25
2 ТЕХНОЛОГІЇ РОЗРОБКИ МОДУЛЯ «ВОЛОЧІННЯ ЗВАРЮВАЛЬНОГО ДРОТУ».....	28
2.1 Обґрунтування вибору технологій для розробки користувацької частини	28
2.2 Вибір інструментів для розробки серверної частини.....	33
2.3 Порівняння обраних інструментів для розробки з аналогами	35
3 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПІДСИСТЕМИ УПРАВЛІННЯ МАТЕРІАЛІВ ДЛЯ ЗВАРЮВАЛЬНИХ РОБІТ ERP «ПЛАЗМІС».....	41
3.1 Аналіз функціональних вимог користувача для модуля «Заявка на перевезення» в логістичній підсистемі	41
3.2 Проектування бази даних.....	488
3.3 Розробка алгоритму функціонування модуля «Волочіння зварювального дроту попереднє» підсистеми виробництво	500

					08-54.МКР.001.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розробив		Босак І.С.			Комп'ютерна система керування виробництвом матеріалів для зварювальних робіт Пояснювальна записка	Літ.	Аркуш	Акрушів
Перевірила		Савицька Л. А.					6	115
Опонент		Шиян А. А.				ВНТУ, гр. 1КІ-23м		
Н. Контр.		Швець С. І.						
Затвердив		Азаров О. Д.						

4	РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ «ВОЛОЧІННЯ ЗВАРЮВАЛЬНОГО ДРОТУ»	53
4.1	Проектування користувацької частини інтерфейсу функціонального модулю підсистеми логістики.....	53
4.2	Реалізація обробки даних в базі даних.....	56
4.3	Реалізація взаємодії клієнтської та серверної частини в підсистемі ...	57
4.4	Тестування розробки.....	60
5.	ЕКОНОМІЧНИЙ РОЗДІЛ	62
5.1	Технологічний аудит розробленого програмного забезпечення для..... автоматизації логістичних цілей у інформаційній системі «ПлазмІС».....	62
5.2	Розрахунок витрат на розроблення програмного забезпечення	66
5.3	Розрахунок економічного ефекту від можливої комерціалізації розробленого програмного забезпечення для автоматизації матеріалів для зварювальних робіт у інформаційній системі «ПлазмІС»	70
	ВИСНОВКИ	80
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	82
	ДОДАТОК А Технічне завдання	86
	ДОДАТОК Б Ілюстративна частина	91
	ДОДАТОК В Лістинг програми	95
	ДОДАТОК Г Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	113

					08-54.МКР.001.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність в оптимізації роботи в сучасному менеджменті великих і середніх підприємств відіграє інформаційна система планування ресурсів, тобто ERP (Enterprise Resource Planning). Як правило, система ERP складається з базового набору функцій і інструментів обробки даних, а також включає різні модулі для створення єдиного циклу, який охоплює всі ключові бізнес-процеси компанії. Серед основних операцій, які реалізують такі системи, варто відзначити автоматизоване управління та моніторинг операцій з отримання ресурсів, автоматичний облік складських запасів, планування виробничих потужностей та управління транспортними процесами за замовленнями клієнтів тощо [1-4].

Вимоги до сучасної системи планування ресурсів підприємства включають поєднання вищезазначених функціональних вимог і вимог нових технологій. Ці нові вимоги включають використання реляційних баз даних, архітектурні характеристики систем клієнт-сервер і можливість інтеграції з існуючими програмними та апаратними платформами. На сучасному ринку найбільш популярними ERP-системами є [5]: знамениті Acumatica ERP і Oracle ERP, ERP-система Sage X3 на базі системи управління базами даних Oracle Database, а також ERP-система Unit4 з розширеними функціями управління підпорядкованими підрозділами. Однак варто зазначити, що ці системи орієнтовані на певну сферу застосування і не є абсолютно універсальними. Високі витрати та обмежена гнучкість для конкретних бізнес-потреб можуть призвести до необхідності розробки нових систем ERP для більш вузької сфери застосування.

Тому ще одним прикладом ERP-системи з урахуванням особливостей планування ресурсів підприємства групи промислових компаній ТОВ «ПлазмаТек» є інформаційна система «ПлазмІС». Система знаходиться в постійному процесі розвитку та вдосконалення, адаптуючись до динамічних змін у потребах заводів, що виробляють сучасні зварювальні прутки, починаючи з

постачання матеріалів і сировини і закінчуючи виробництвом готової продукції та її супутніми поставками. світ.

Інтенсивний розвиток і створення нових частин в інформаційній системі «ПлазмІС» відповідає поставленим завданням і призводить до розвитку нових функцій системи, особливо в області планування виробництва.

Мета і задачі магістерської роботи є розширення функціональних можливостей підсистеми управління виробництвом зварювального дроту в системі управління ресурсами підприємства (ERP) «ПлазмІС». Для досягнення поставленої мети необхідно вирішити наступні завдання:

Аналіз шляхів автоматизації управління виробництвом зварювального дроту в системі управління ресурсами підприємства «ПлазмІС». Це включає в себе вивчення та оцінку існуючих методів і практик автоматизації в системах ERP (зокрема «ПлазмІС»).

Впровадити варіанти інформаційних технологій. На основі аналізу визначте найкращу технологічну платформу або інструмент, який може допомогти автоматизувати функції керування виробництвом зварювального дроту.

Розробити програмне забезпечення для модуля «Змінне завдання: волочіння зварювального дроту попереднє» в підсистемі ERP «ПлазмІС». Створення програмного коду для додаткових модулів, які доповнюють функціональність системи «ПлазмаІС» і дозволяють автоматизувати процеси, пов'язані з виробництвом зварювального дроту.

Тестувати розроблені функціональні модулі. Після завершення розробки розроблені модулі перевіряються на функціональність і відповідність задокументованим вимогам і стандартам. Тестування допоможе виявити та виправити можливі помилки та недоліки перед впровадженням.

Магістерська робота спрямована на підвищення функціональності системи «ПлазмІС» у сфері контролю виробництва зварювального дроту та допоможе підприємствам ефективніше справлятися з поставленими завданнями.

Об'єкт дослідження — процес автоматизації виробництва зварювального дроту в системах керування ресурсами підприємства системи.

Предметом дослідження є програмна реалізація функціональних модулів, спрямованих на автоматизацію процесів в системі управління виробництвом зварювального дроту інформаційної системи «ПлазмІС».

Методи дослідження: системний аналіз, методи інженерії знань, особливо побудова моделі знань на основі систем продуктів, аналіз виробничих бізнес-процесів, об'єктно-орієнтоване проектування, методи розробки програмного забезпечення клієнт-сервер.

Науково-технічним результатом розробки програмного забезпечення автоматизації виробництва зварювального дроту в інформаційній системі «ПлазмІС» є створення інтегрованого динамічного програмного продукту, який суттєво сприяє управлінню логістичними процесами підприємства ТОВ «ПлазмаТек».

Практична цінність роботи полягає в розробці алгоритмів і програмних засобів, спрямованих на розробку програмного забезпечення для автоматизації виробництва зварювального дроту в інформаційній системі «ПлазмІС».

Апробація результатів. І. С. Босак, Л. А. Савицька, О. В. Дудник. Комп'ютерна система керування виробництвом // [Електронний ресурс] / Л. А. Савицька // Матеріали LIV Всеукраїнської науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії, Вінниця, 24-27 березня 2025 р. – Електрон. текст. дані. – 2024.

1 АНАЛІЗ ПІДХОДІВ ДО СИСТЕМ КЕРУВАННЯ ВИРОБНИЦТВОМ МАТЕРІАЛІВ ДЛЯ ЗВАРЮВАЛЬНИХ РОБІТ.

1.1 Загальні відомості про інформаційні ERP – системи та їх приклади

На сучасному етапі соціальної глобалізації та автоматизації бізнес-процесів на різних підприємствах існує велика потреба в технологічних рішеннях для спрощення роботи підприємств, від виробничих ліній до управління персоналом. У сучасному світі системи планування ресурсів підприємства, а саме ERP (Enterprise Resource Planning), стали незамінним інструментом, який широко використовується підприємствами будь-якого розміру в різних галузях. Посилення використання цих систем пов'язане з необхідністю конкурувати в глобалізованому світі з великою кількістю динамічних зв'язків, що вимагає значного підвищення темпу роботи та ефективного управління. Понад 88% компаній, що працюють у різних ринках і секторах, вважають впровадження ERP-систем успішним і необхідним [1].

Основними завданнями ERP-системи є автоматизація бізнес-процесів і надання інформаційних повідомлень агентам (користувачам з різними рівнями доступу) про події, що відбуваються в системі, а також забезпечення внутрішнього контролю.

Такі системи використовують центральну базу даних, яка містить інформацію, зібрану з різних відділів, таких як бухгалтерія, виробництво, ланцюг постачання, кадри, маркетинг і управління. За допомогою центральної бази даних менеджери різних рівнів і в різних відділах можуть отримати доступ до зібраної інформації для подальшого аналізу сценаріїв, оптимізації процесів і підвищення ефективності за допомогою різних методів. Як правило, підвищення продуктивності компанії завдяки використанню системи ERP оцінюється за такими критеріями, як економія коштів і підвищення продуктивності. Легкий доступ до даних та їх систематизація також значно скорочує витрати часу менеджерів і дає можливість легше розширювати виробництво за потреби.

Тому в багатьох випадках система ERP є основним програмним забезпеченням у бізнес-структурі, що дозволяє менеджерам отримувати доступ до максимального обсягу зібраних структурованих даних у найкоротші терміни. Вони формують єдине бачення реальності в межах конкретного підприємства, що впроваджує систему [2].

Сучасна архітектура системи ERP зазвичай включає три рівні, як показано на рисунку 1.1.

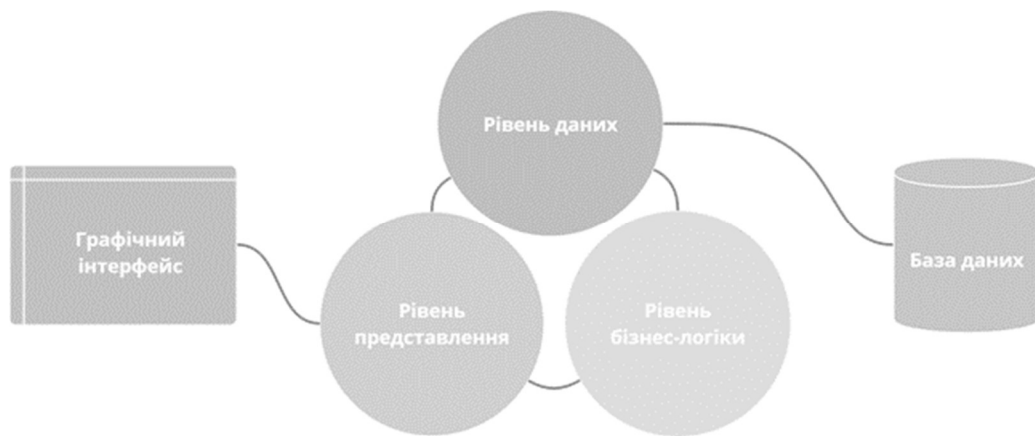


Рисунок 1.1 — Трьохрівнева архітектура ERP-систем

Рівень даних відповідає за зберігання даних, структурування даних і реагування на запити користувачів. Усі важливі дані компанії зберігаються та керуються на цьому рівні. Він є основою системи та забезпечує надійну та послідовну роботу програм ERP.

Рівень бізнес-логіки— це програми, які відповідають за обробку та аналіз даних, а також забезпечують взаємодію між різними компонентами системи. Програми ERP включають модулі в різних функціональних областях, таких як фінанси, логістика, виробництво та людські ресурси. Вони забезпечують автоматизацію бізнес-процесів і можливості аналізу [7].

Презентаційний рівень — дозволяє користувачам безпосередньо взаємодіяти з системою. Графічні інтерфейси включають веб-портали, настільні програми та інші інтерфейси, які дозволяють користувачам виконувати дії та

отримувати доступ до інформації. Це спрощує роботу користувача та забезпечує інтуїтивно зрозумілий інтерфейс для взаємодії з системою.

Ця трирівнева архітектура дозволяє забезпечити ефективну роботу вашої ERP-системи та підтримує інтеграцію з іншими програмами та розширеннями відповідно до потреб компанії.

Важливо враховувати, що ідеальна ERP-система повинна включати якісну розробку всіх трьох необхідних і достатніх компонентів. На початку існування таких систем замість графічного інтерфейсу можна було використовувати командний рядок, але це ускладнювало роботу та підвищувало вимоги до управлінської кваліфікації. Тому вони повинні бути не тільки досвідченими користувачами, але й володіти спеціальними навичками в галузі інформаційних технологій. Розробка та підтримка високоякісних графічних інтерфейсів може уникнути цих вимог до менеджерів і полегшити поширення систем і залучити більше компаній, які їх використовують [3].

ERP – це платформа, яка допомагає планувати ресурси та відстежувати бізнес-процеси.

Основні функції ERP включають:

- фінансовий облік;
- керування складом;
- управління виробництвом;
- управління ланцюгом поставок;
- управління продажами та маркетингом;
- управління людськими ресурсами;
- аналіз і звітність;
- управління проектами;
- управління взаємовідносинами з клієнтами (CRM);
- бізнес-аналітика та аналітика.

Фінансовий облік — це система ERP яка дозволяє вести фінансову документацію вашої компанії, включаючи бухгалтерський та фінансовий облік, виставлення рахунків, податковий облік, рух грошових коштів і фінансовий

аналіз. Він допомагає підприємствам підтримувати фінансову дисципліну та створювати фінансові звіти для внутрішнього та зовнішнього використання.

Керування складом — це функція дозволяє підприємствам відстежувати товари, керувати запасами, прогнозувати попит, розміщувати замовлення та контролювати логістичні операції. Це допомагає уникнути дефіциту або надлишку запасів.

Управління виробництвом — це системи ERP які можуть планувати та контролювати виробничі процеси, включаючи виробництво продукції, контроль якості та оптимізацію виробничих потужностей.

Управління ланцюгом поставок — це функція яка допомагає компаніям взаємодіяти з постачальниками, керувати замовленнями, оптимізувати логістичні процеси та гарантувати Сировина надходить вчасно[8].

Управління продажами та маркетингом — система ERP яка допомагає відстежувати клієнтів, обробляти замовлення, аналізувати ринок і розробляти маркетингові стратегії для збільшення продажів і залучення нових клієнтів.

Управління людськими ресурсами — це розділ який включає облік працівників, управління заробітною платою, професійний розвиток та управління персоналом. ERP допомагає в ефективному управлінні людськими ресурсами та підтримує внутрішні процеси людських ресурсів [4].

Аналіз і звітність — це системи ERP які збирають дані та надають аналітичні інструменти для прийняття рішень. Це включає створення звітів, аналіз даних і прогнозування бізнес-процесів.

Управління проектами — ERP дозволяє планувати та контролювати проекти, розподіляти ресурси та записувати всі аспекти проекту.

Управління взаємовідносинами з клієнтами (CRM) — це функція, що забезпечує зберігання та аналіз інформації про клієнтів для збереження історії взаємодії та покращення відносин із клієнтами.

Бізнес-аналітика та аналітика — ERP використовує дані для прогнозування та аналізу бізнес-процесів. Це дозволяє вам визначити тенденції та зрозуміти, які аспекти вашої діяльності потребують покращення.

Системи ERP об'єднують ці функції в одну систему, спрощуючи управління бізнесом і роблячи його більш ефективним і конкурентоспроможним.

Значення ERP у кожному конкретному випадку використання залежить від функції, яку він виконує в конкретній компанії. Загалом ERP керує багатьма бізнес-функціями, оптимізуючи та автоматизуючи повсякденні бізнес-процеси. Програмне забезпечення не тільки підвищує ефективність роботи, але й надає повний огляд усіх аспектів бізнесу. Якщо ERP-система спеціалізована і розроблена спеціально для потреб компанії, вона може автоматизувати велику кількість дрібних операцій і виконувати повний аналіз бізнес-процесів. У випадку класичного ERP важливо знайти баланс між потребами конкретної компанії та загальними принципами роботи та управління, втіленими в системі. Використання програмного забезпечення ERP підвищує ефективність і продуктивність користувачів і забезпечує більшу гнучкість роботи [5].

Системи ERP мають багато переваг, таких як:

- автоматизація бізнес-процесів;
- єдина централізована база даних;
- підвищення продуктивності;
- Скорочення витрат;
- більш точні інформаційні дані;
- покращене обслуговування клієнтів;
- зручний аналіз даних;
- гнучкість і масштабованість;
- покращені управлінські рішення;
- підвищення конкурентоспроможності.

Автоматизація бізнес-процесів — ERP дозволяє автоматизувати багато щоденних операцій і завдань, оптимізуючи вашу роботу та зменшуючи ризик помилок.

Єдина централізована база даних — ERP використовує єдину базу даних, щоб дозволити різним відділам і функціональним групам взаємодіяти та обмінюватися інформацією, покращуючи співпрацю всередині компанії.

Підвищення продуктивності — завдяки автоматизації та оптимізації процесів ERP допомагає підвищити продуктивність співробітників.

Скорочення витрат — використання ERP може допомогти зменшити витрати на виробництво, логістику та управління запасами за рахунок оптимізації процесів.

Більш точні інформаційні дані — ERP забезпечує точність і надійність інформації за допомогою централізованої системи управління даними.

Покращене обслуговування клієнтів — завдяки доступу до важливої інформації про клієнтів компанії можуть краще задовольняти їхні потреби та покращувати обслуговування.

Зручний аналіз даних — ERP надає інструменти аналізу даних і звітності, які допомагають приймати обґрунтовані рішення.

Гнучкість і масштабованість — системи ERP можуть адаптуватися до потреб конкретної компанії та масштабуватися в міру зростання бізнесу.

Покращені управлінські рішення — завдяки доступу до важливої інформації в режимі реального часу керівництво може приймати швидкі обґрунтовані рішення.

Підвищення конкурентоспроможності — впровадження ERP може допомогти компаніям стати більш конкурентоспроможними на ринку завдяки більш ефективному управлінню та ресурсам.

Впровадження системи ERP, безсумнівно, має свої недоліки та проблеми, зокрема:

- тривалість і вартість впровадження;
- навчання користувачів;
- вимоги до обладнання та захисту даних;
- сервери та інфраструктура;
- системи безпеки;
- інтеграція з існуючими системами.

Тривалість і вартість впровадження — впровадження системи ERP може бути тривалим і дорогим процесом, і витрати не завжди окупаються відразу.

Компанії повинні бути готові інвестувати в процес і мати на увазі довгострокові вигоди.

Навчання користувачів — перехід на ERP вимагає навчання персоналу. Для цього може знадобитися час і ресурси, а користувачам може знадобитися час, щоб адаптуватися до нової системи.

Вимоги до обладнання та захисту даних — оптимальна робота ERP вимагає потужного обладнання та високого рівня захисту даних. Для цього можуть знадобитися додаткові інвестиції.

Сервери та інфраструктура — налаштування серверів для зберігання даних і резервного копіювання може бути складним завданням і збільшити витрати.

Системи безпеки — зберігання та передача корпоративних даних вимагає високого рівня захисту та контролю доступу, що може бути дорогим і складним завданням.

Інтеграція з існуючими системами — для багатьох компаній інтеграція ERP з існуючими системами може бути складним завданням, яке вимагає додаткових зусиль і ресурсів.

Незважаючи на ці проблеми, правильно спроектована та впроваджена ERP-система може значно підвищити ефективність бізнесу та управління.

Це робить його важливим інструментом для багатьох компаній.

ERP-системи поділяються на:

- комплексні;
- відділ;
- для малого бізнесу.

Комплексні системи ERP — ці системи розроблені для великих і середніх підприємств і охоплюють більшість аспектів бізнесу, таких як фінанси, логістика, виробництво, кадри та маркетинг. Вони пропонують комплексні рішення для всієї організації, часто з широким функціональним розмахом.

Галузеві системи ERP — ці системи розроблені для певної галузі чи галузі. Їх можна налаштувати відповідно до конкретних потреб і вимог конкретних галузей, таких як виробництво, логістика, фармацевтика, освіта, охорона

здоров'я тощо. г. Галузеві ERP-системи можуть бути більш спеціалізованими та відповідати певним галузевим стандартам.

Системи ERP для малого бізнесу — ці системи розроблені для малих підприємств, яким може не потрібен такий же рівень функціональності, як великим компаніям. Вони часто мають спрощену функціональність і можуть бути більш доступними для малого бізнесу з точки зору вартості та ресурсів. [9]

Крім наведених вище класифікацій, ERP-системи також можна розділити за архітектурними характеристиками. Архітектурні характеристики програмного продукту визначають можливість інтеграції конкретної ERP-системи за конкретних умов використання. Від відповідності ERP-системи залежить успішне впровадження та подальше використання ERP-системи на підприємстві, а також складність подальшого розвитку системи при появі нових функціональних вимог або необхідності інтеграції з новим програмним забезпеченням. Архітектура та потреби підприємства. За цим параметром можна виділити лише дві категорії програмних продуктів:

- має централізовану архітектуру;
- розподілена архітектура.

Централізована архітектура — у системі цього типу центральний сервер або центральна база даних відіграє ключову роль у зберіганні та обробці даних. Усі користувачі та відділи підприємства підключаються до центрального сервера для доступу до інформації та виконання дій. Це робить архітектуру централізованої системи ERP відносно простою в управлінні та обслуговуванні, але може створити проблеми з надійністю, оскільки всі запити обробляються центральним сервером.

У цьому типі системи ERP обробка даних і функціональні можливості розподіляються між різними серверами або вузлами мережі. Це забезпечує більшу гнучкість і масштабованість, оскільки завдання можна розподіляти між декількома серверами. Однак ця архітектура може бути складнішою для налаштування та керування та вимагає додаткових зусиль для забезпечення синхронізації даних між різними вузлами.

Asumatica вважається однією з найпопулярніших систем на сучасному ринку ERP-систем. Завдяки хмарній технології він дуже зручний, що дозволяє легко впроваджувати його в малому та середньому бізнесі. Крім того, конкурентна ціна робить Asumatica [10] доступною для багатьох компаній. Ця ERP-система має багатий досвід і довіру користувачів завдяки своїй тривалій присутності на ринку та широкому набору функцій.

ERP-система Oracle [11] давно популярна на ринку і користується високим попитом. Oracle ERP використовує Інтернет для оптимізації та інтеграції діяльності великих і середніх підприємств. Вона є лідером у сфері фінансового менеджменту та прогнозування. Оптимізація тимчасових ресурсів при використанні цього ERP особливо помітна завдяки наявному функціоналу системи.

Система ERP Unit4 [12] також заслуговує на увагу своїми універсальними програмними засобами, функціями нового покоління та допоміжними засобами управління. Система широко поширена і має лояльні ціни навіть на тлі зростаючої конкуренції.

ERP-система Sage X3[13] є потужною та масштабованою та призначена для комплексного управління ресурсами підприємства. Однією з його основних функцій є централізація інформації для підвищення продуктивності компанії. Він особливо популярний серед виробників технологій, оскільки допомагає керувати великими обсягами даних, які важко контролювати та планувати.

Отже, можна зробити висновок, що якісних і популярних ERP-систем на ринку достатньо. Однак важливо враховувати, що кожен з них спеціалізований для певної області застосування і не є абсолютно універсальним. Крім того, високі витрати та обмежена гнучкість у контексті потреб конкретного підприємства можуть призвести до необхідності розробки нових систем ERP, які спеціалізуються на більш вузьких областях.

1.2 Огляд відомих програм для автоматизації логістичних цілей в ERP системах

Сьогодні існує безліч інформаційних систем різного типу. Підприємства розробляють систему аутсорсингом або наймають персонал для подальшого розвитку та підтримки. Давайте розглянемо системи, які пропонують більш вузькі завдання і доступні на різних національних і міжнародних ринках.

1.2.1 Система SAP Extended Warehouse Management

SAP Extended Warehouse Management (SAP EWM) — це програмний пакет, який спеціалізується на управлінні складами та транспортними операціями. Розглянемо його призначення, функціонал і переваги докладніше.

SAP EWM призначений для оптимізації та ефективного управління всіма аспектами логістики на складі. Він призначений для компаній, які мають як склад, так і складний ланцюг постачання, і хочуть підвищити ефективність управління ресурсами та вантажами.

SAP EWM пропонує різноманітні можливості, зокрема:

- управління запасами;
- ефективне зберігання;
- управління замовленнями;
- інтеграція з транспортом;
- відстеження.

Управління запасами — комплексний контроль запасів, який включає отримання, передачу, переміщення та планування на детальному рівні.

Ефективне зберігання — розміщення товарів на складі є ефективним для максимального використання простору.

Управління замовленнями — відстеження та обробка замовлень клієнтів з можливістю швидкого виконання.

Інтеграція з транспортом — засоби для управління логістичними процесами, пов'язаними з транспортуванням.

Відстеження — система полегшує моніторинг товарів на складі під час їх руху в режимі реального часу та контролює роботу пристроїв і обладнання.

Переваги системи SAP Extended Warehouse Management включають такі аспекти:

- поєднання з SAP ERP;
- ефективність;
- надійність

Поєднання з SAP ERP — спільні зусилля з іншими модулями SAP ERP, які мають значний обсяг потенційної функціональності.

Ефективність — завдяки функціональності SAP EWM компанії можуть скоротити свої логістичні процеси, що призводить до підвищення ефективності та зниження витрат.

Надійність — SAP відомий своєю надійністю та підтримкою, тому користувачі можуть очікувати, що він буде надійним і послідовним[14].

Неефективність системи включає в себе наступні аспекти:

- вартість ліцензії;
- залежність від інших систем.

Вартість ліцензії — вартість ліцензії та підтримки SAP EWM є змінною та залежить від обсягу функціональності та потреб користувача.

Залежність від інших систем — якщо корпорація вже використовує інші системи, інтеграція з SAP EWM буде складною та потребуватиме додаткових зусиль.

Не підходить для невеликих компаній: SAP EWM може бути надто потужним або складним для невеликих компаній, які не мають великого обсягу сховищ.

SAP Extended Warehouse Management — це потужний програмний інструмент, який можна використовувати для оптимізації логістики на складі. Цей інструмент особливо корисний для компаній, які мають різноманітну складну інфраструктуру на складі, але обмеження цієї системи повинні бути визнані та враховані. .

1.2.2 Система Oracle Transportation Management

Програмне забезпечення Oracle Transportation Management (OTM) призначене для управління логістичними процесами та оптимізації управління транспортуванням і вантажами в компаніях. Розглянемо його призначення, функціонал і переваги докладніше.

OTM призначений для організацій, які мають великий або середній парк транспортних засобів і мають логістичні проблеми. Це полегшує оптимізацію планування логістики, управління вантажами та вибір маршруту, а також моніторинг транспортної діяльності.

Платформа Oracle Transportation Management (OTM) надає різноманітні можливості, зокрема:

- планування маршруту;
- відстеження;
- планування вантажів;
- управління витратами;
- інтеграція.

Планування маршруту — визначення найкращих маршрутів для транспортування вантажу з урахуванням різних факторів, включаючи час, вартість, вагу тощо.

Відстеження — онлайн-обстеження транспортної діяльності та вантажів для забезпечення їх безпеки та своєчасної доставки.

Планування вантажів — керуйте завантаженням транспортних засобів і вантажів, щоб максимізувати їхню ефективність.

Управління витратами — контроль транспортних витрат і оптимізація логістичних бюджетів.

Інтеграція — можливість об'єднання з іншими системами, такими як ERP та управління запасами.

Переваги системи Oracle Transportation Management включають такі аспекти:

- зменшення витрат;
- підвищення продуктивності;
- підвищена задоволеність клієнтів.

Зменшення витрат — це допомагає компаніям знизити витрати на транспортування та логістику шляхом ефективного планування та моніторингу.

Підвищення продуктивності — полегшує логістичні процеси, що призводить до вищої продуктивності праці та транспортних засобів.

Підвищена задоволеність клієнтів — гарантує своєчасну та точну доставку товарів, що призводить до підвищення задоволеності клієнтів.

До негативних сторін цієї системи можна віднести наступні фактори:

- високі витрати;
- складність.

Високі витрати — провадження та обслуговування ОТМ є дорогим для компанії.

Складність — для повного використання всіх можливостей системи необхідні навчання та досвід.

Oracle Transportation Management System — це потужний програмний інструмент, який полегшує керування транспортною логістикою, але вимагає зобов'язань щодо впровадження та підтримки з боку користувача.

1.2.3 Система Microsoft Dynamics 365 Supply Chain Management

Microsoft Dynamics 365 Supply Chain Management — це програмний пакет, призначений для оптимізації та автоматизації процесів управління ланцюгами поставок у компаніях.

Метою Microsoft Dynamics 365 Supply Chain Management є підвищення ефективності логістики та операцій, пов'язаних із постачанням. Він допомагає компаніям краще керувати транспортуванням, зберіганням, виробництвом та іншими аспектами ланцюга постачання.

Microsoft Dynamics 365 Supply Chain Management надає різноманітні можливості, зокрема:

- управління запасами;
- планування виробництва;
- управління вантажами та логістикою;
- розумний;
- інтеграція.

Управління запасами — відстеження та покращення рівня запасів з метою забезпечення достатнього запасу товарів і матеріалів.

Планування виробництва — планування та оптимізація виробництва на основі попиту та ресурсів.

Управління вантажами та логістикою — відстежуйте та керуйте транспортною діяльністю, включаючи маршрутизацію та доставку.

Розумний — прийняття рішень на основі даних і підвищення продуктивності.

Інтеграція — можливість об'єднання з іншими системами, включаючи управління запасами та облік.

Переваги системи включають наступні аспекти:

- підвищення продуктивності;
- підвищена точність;
- усунення людської праці.

Підвищення продуктивності — підвищує ефективність процесів ланцюга постачання, що призводить до підвищення продуктивності та зниження витрат.

Підвищена точність — допомагає уникнути помилок в управлінні запасами та плануванні виробництва, це підвищує якість продукту та послуг.

Усунення людської праці — автоматизація багатьох процесів зменшує необхідність людського втручання та підвищує точність даних.

До негативних сторін цієї системи можна віднести наступні фактори:

- ціна;
- складність інтеграції в інші системи;
- розвиток персоналу.

Ціна — витрати на впровадження та підтримку Microsoft Dynamics 365 переважно несуть малі підприємства.

Складність інтеграції в інші системи — інтеграція в існуючі системи може вимагати додаткових зусиль і ресурсів.

Розвиток персоналу — компанії може знадобитися підвищити кваліфікацію персоналу, щоб використовувати Microsoft Dynamics 365.

Загалом, керування ланцюгом поставок Microsoft Dynamics 365 — це потужна система ERP, яка має широкий спектр функціональних можливостей, корисних для середніх і великих компаній, але дорога для впровадження та вимагає навчання працівників.

1.3 Особливості інформаційної ERP системи «ПлазмІС» та вимоги до її виробництва зварювального дроту

Інформаційна система «ПлазмаІС» спочатку була створена як результат потреб підприємств ТОВ «ПлазмаТек», найбільшого виробника електродів і сировини для зварювання в Україні, вперше впроваджена в 2011 році. Вона продовжує розвиватися та вдосконалюватись відповідно до потреб заводів ТОВ «ПлазмаТек» та їхніх нових бажань.

Система «ПлазмІС» охоплює всі етапи виробництва, від закупівлі матеріалів до видачі готової продукції, а також враховує специфіку виробничих процесів на підприємстві. У «ПлазмІС» є розділи, які документують документообіг, ведуть кадри, розраховують табелі, виробляють продукцію, ведуть облік матеріальних цінностей, створюють звіти та розрахунки, мають цілі логістики, такі як план, графік, замовлення від постачальників та ін. На рисунку 1.2 наведена схема, яка узагальнює всі розділи інформаційної системи «ПлазмаІС».

Активне культивування та розширення інформаційної системи «ПлазмаІС» відповідно до вимог ПрАТ «ПлазмаТек» призвело до необхідності розширення функціональних можливостей логістики шляхом додавання нових компонентів. За допомогою виробництва співробітники можуть виконувати кілька важливих

завдань, включаючи: управління складськими процесами та відвантаженням необхідних товарів, обіг матеріалів на заводах, документування всіх виробничих операцій, автоматичне сповіщення відповідних осіб і драйвери, а також імпорту інформації з системи.



Рисунок 1.2 — Схема поділу розділів інформаційної системи «ПлазмІС»

Виробництво у системі оптимізує усі процеси та забезпечує належну ефективність виробництва. У даному розділі наявні такі підрозділи: змінні завдання, операційні карти, журнал актів браку, технолог.

Розробка та доповнення функціоналу модулю змінне завдання волочіння зварювального дроту попереднє є задачею даної магістерської роботи.

Інформаційна система «ПлазмІС» має багато переваг, зокрема:

- надання комплексних рішень з управління виробничими процесами, характерними для даної галузі;
- підвищення продуктивності виробництва;
- централізація інформації підприємства;
- підтримка безперебійного зв'язку між відділами та співробітниками системи.
- здатність динамічно змінювати інформацію;
- перегляд і редагування застарілого контенту в цій комунікаційній системі.

«ПлазмаІС» — комплексний програмний комплекс, який дозволяє динамічно

розвиватися з часом. Одним із основних атрибутів є наявність на підприємстві спеціального інформаційного відділу, який відповідає за підтримку системи у відповідності до вимог виробництва. Програмне забезпечення платформи використовується для легкого додавання нових компонентів і розділів.

Одним з важливих компонентів є зручний, інтуїтивно зрозумілий інтерфейс, який не вимагає спеціальних знань з інформаційних технологій. Комплексна комбінація компонентів системи дозволяє ефективно контролювати всі аспекти діяльності компанії. «ПлазмаІС» знижує трудомісткість складних процесів, одночасно підвищуючи продуктивність.

2 ТЕХНОЛОГІЇ РОЗРОБКИ МОДУЛЯ «ВОЛОЧІННЯ ЗВАРЮВАЛЬНОГО ДРОТУ»

Веб-розробка поділяється на дві основні області: інтерфейс (клієнт) і сервер (сервер). Основна відмінність між ними полягає в тому, що зовнішній код взаємодіє з користувачем, тоді як серверний код взаємодіє з даними на сервері.

Під час розробки інтерфейсу інструменту «Волочіння зварювального дроту попереднє» використовувалися наступні інструменти: HTML, CSS, JavaScript і платформа KendoUI.

2.1 Обґрунтування вибору технологій для розробки користувацької складової.

Процес вибору технологій, які використовуватимуться для розробки спеціального компонента проекту, як правило, базується на кількох важливих факторах. Спочатку ви повинні розглянути функціональні вимоги проекту та його масштаб, оскільки різні технології мають різні можливості та діапазони масштабування. Крім того, важливо враховувати наявність ресурсів, призначених для розробки та підтримки вибраної технології, якість документації та підтримку спільноти.

HTML відіграє значну роль у створенні будь-якої веб-сторінки, це фундаментальний компонент, який все ще необхідний незалежно від складності веб-сайту чи кількості використаних технологій. Це не просто важливий навик для будь-якого веб-розробника, але він важливий для кожного, хто хоче навчитися мистецтву створення вмісту для Інтернету.

Мова гіпертексту — HTML, що характеризується використанням тегів для розрізнення різних типів вмісту та опису їхньої ролі на веб-сайті. І навпаки, CSS або каскадні таблиці стилів дають змогу вказати спосіб, у який HTML-компоненти відображаються на веб-сайті, що забезпечує кращий вигляд презентації.

HTML тісно пов'язаний із CSS, останній забезпечує вихідні матеріали для організації вмісту на веб-сайті. Ці дві гіпертекстові розмітки зберігаються окремо, це підвищує ефективність створення веб-сайтів до їх показу.

Особливу увагу приділено фреймворку Kendo UI. Kendo UI — це не просто набір із понад 10 віджетів (таблиць, графіків, списків, розкритих списків, дерев і численних селекторів), а й платформа для розробки клієнта на основі jQuery. Сьогодні використовуються сучасні технології, такі як HTML5 і CSS3, які дозволяють Kendo UI підтримувати старіші браузери без шкоди для продуктивності. Цей фреймворк ефективний із зв'язуванням даних, шаблонами, анімацією, можливістю перетягувати об'єкти (drag-and-drop), приховувати їх або, навпаки, показувати в інтерфейсі, збільшувати чи зменшувати розмір стовпця тощо [18].

JavaScript — це високоякісна інтерпретована мова, яка використовується для створення динамічних інтерактивних веб-сайтів. Спочатку мова була призначена для використання на стороні клієнта (у браузерах) для полегшення взаємодії з користувачем і створення динамічного вмісту.

У 1995 році була створена мова програмування JavaScript, яка є основою для розробки динамічних та інтерактивних сайтів. Дослідження історії цього етапу включає кілька значущих подій.

На початку 90-х веб-сайти були переважно статичними, а їх взаємодія з громадськістю була обмеженою. Популярність таких браузерів для Інтернету, як Mosaic і Netscape Navigator, була значною [19].

Майкл Еддісон, програміст Netscape Communications Corporation, визнав необхідність створення мови, яка б полегшила створення динамічних веб-сайтів. Його зусилля привели до випуску мови, яка спочатку називалася «LiveScript», у серпні 1995 року.

Однак одночасно Sun Microsystems випустила мову програмування Java. Щоб більше просувати свій продукт, Netscape вирішив змінити бренд «LiveScript» на «JavaScript».

Завдяки партнерству з Microsoft JavaScript було включено в браузер Internet Explorer, що сприяло його широкому впровадженню серед обох популярних браузерів.

У 1997 році JavaScript було формалізовано як ECMAScript, що сприяє стабільності та зменшує складність розробки. З часом JavaScript став широко використовуватися у веб-розробці, нові версії стандарту ECMAScript додали нові функції та можливості.

У 2000-х роках була популяризована технологія Ajax, яка дозволяла асинхронно передавати дані на сервер без перезавантаження сторінки. Зростання популярності таких бібліотек, як jQuery, спростило використання JavaScript.

Сьогодні JavaScript все ще залишається однією з найпопулярніших мов програмування, він використовується як на клієнтській, так і на серверній стороні веб-розробки, крім того, він використовувався з мовою програмування Node.js для мобільної розробки (з використанням таких фреймворків, як React Native).

JavaScript часто використовується в більшості веб-додатків і веб-сайтів. Нижче ми надаємо кілька прикладів використання JavaScript у популярних програмах:

— карти Google;

Карти Google — використовується для інтерактивного відображення карт, подорожей та інших служб на Картах Google.

— сторінки Facebook призначені для асинхронного отримання нових повідомлень, оновлення своїх стрічок новин і відображення інтерактивного вмісту на сторінці.

Twitter — використовується для динамічного завантаження нових твітів без необхідності перезавантажувати сторінку разом із використанням відео та зображень, вбудованих у сторінку.

— JavaScript використовується для полегшення асинхронного завантаження та оновлення електронних листів, а також для інтерактивних функцій, таких як перетягування повідомлень.

— використовується для створення інтерактивного відеопрогравача, який взаємодіє з коментарями, асинхронно завантажує відео та дає рекомендації.

— у веб-версії Slack JavaScript використовується для полегшення зв'язку з каналом, оновлення каналу та взаємодії з різними компонентами інтерфейсу.

— використовує JavaScript для створення інтерактивного та динамічного інтерфейсу, а також для перегляду відео та керування аудіо/відеопотоками.

Ці приклади демонструють різні способи використання JavaScript у веб-додатках, зокрема створення динамічного інтерактивного інтерфейсу, асинхронну обробку даних, анімацію та інші функції.

Давайте обговоримо основні переваги мови програмування JavaScript, які полегшать реалізацію продукту в майбутньому.

JavaScript використовується для створення динамічних та інтерактивних веб-сторінок на стороні клієнта. Це полегшує зміну вмісту сторінки без необхідності повторного завантаження, взаємодії з користувачем і створення кількох цікавих веб-компонентів.

JavaScript підтримується всіма сучасними веб-переглядачами, що дозволяє йому бути стандартом для інтерфейсної розробки.

Багато розробників вважають JavaScript простим для вивчення. Синтаксис мови подібний до інших мов програмування, таких як C++ і Java.

Здатність JavaScript асинхронно виконувати код дає змогу ефективно взаємодіяти з сервером без негативного впливу на інші операції.

Доступно багато бібліотек і фреймворків JavaScript, зокрема JQuery, React, Angular або Vue.js, які підвищують ефективність і швидкість розробки.

Багато випущених змін до JavaScript не впливають негативно на зворотну сумісність із попередніми версіями мови, це робить старий код, написаний у попередніх версіях, актуальним із часом.

JavaScript використовується не лише на клієнтській стороні веб-розробки, але й на сервері, що використовує платформу Node.js. Це полегшує використання однієї мови як для передньої, так і для задньої частини веб-програми.

У 2017 році IBM назвала JavaScript однією з найважливіших мов програмування сьогодні. Він використовується як на стороні клієнта, так і на стороні сервера, що полегшує створення простих і ефективних інтерфейсів. Функціонал JavaScript вважається найдосконалішим. Діапазон JavaScript не обмежується певною сферою - він також використовується в програмах для ПК, мобільних пристроїв і серверів.

Дуже важливо усвідомлювати різницю між рамками JavaScript і бібліотеками. Фреймворки схожі на рамки, які обмежують дизайн програми. Розробник розширює цю базу, але тільки відповідно до раніше встановлених правил щодо використання фреймворків JavaScript. Такого регламенту в бібліотеці немає. Найпоширеніші фреймворки JavaScript для Angular: Angular JS, Aurelia JS, Backbone.js, dhtmlx Suite, Ember.js, jQuery, KnockoutJS, Meteor.js, Node.js, Polymer.js і React.js.

у — це бібліотека JavaScript з відкритим кодом, яка полегшує взаємодію між документом HTML/CSS і JavaScript. Зокрема, це об'єктно-орієнтована бібліотека, яка полегшує зв'язування документів із відповідними об'єктами. Він здатний переглядати та змінювати HTML-документи, обробляти події браузера, виконувати дії DOM, взаємодіяти з AJAX і розробляти кросбраузерні програми

Кілька важливих факторів, які сприяють популярності jQuery, він полегшує легкий і ефективний контроль над HTML і CSS, що зменшує труднощі роботи з веб-документами. Бібліотека пропонує простий доступ і легкий контроль над елементами Document Object Model (DOM), що дозволяє обробляти та змінювати ці елементи. jQuery надає прості методи для ініціювання та керування

різноманітними подіями на веб-сайті, такими як натискання миші чи клавіатури, що полегшує взаємодію з користувачем більш ефективно. Бібліотека зменшує труднощі використання AJAX для взаємодії з сервером в асинхронному режимі, ця технологія дозволяє оновлювати частини сторінки без необхідності повністю її перезавантажувати. jQuery є відносно легким і простим у використанні, цей атрибут головним чином відповідає за привабливість програмного забезпечення для розробників, особливо тих, хто новачок у веб-розробці. Спільнота є великою та активною, яка підтримує швидку розробку jQuery та усунення несправностей. Плагіни jQuery мають різноманітні функції, які дозволяють розробникам розширити його можливості та використовувати готові рішення для різноманітних завдань.[21]

Вибір інструментів для розробки серверної частини

Вибір інструментів для розробки серверної складової проекту має вирішальне значення для успіху, масштабованості та надійності системи.

Oracle Application Express (APEX) — це інструмент розробки веб-додатків, який полегшує створення та розповсюдження веб-додатків на базі даних. APEX надає інструменти для створення інтерактивних і динамічних веб-інтерфейсів, які можуть підключатися до баз даних Oracle [22].

Лінійка продуктів корпорації Oracle для APEX призначена для створення, розробки та розгортання функціональних, зручних і адаптивних веб-сайтів, які взаємодіють із базою даних через веб-браузер. Oracle Application Express — це безкоштовна програма, яка служить інтерфейсом до бази даних Oracle. За допомогою програмного забезпечення APEX можна створювати універсальні веб-додатки, ефективні в більшості сучасних браузерів [23, 24].

Переваги Oracle Application Express:

- швидкість розробки продукту;
- асоціація з базою даних Oracle;

Швидкість розробки продукту — APEX надає розробникам швидкі засоби для створення веб-додатків за допомогою внутрішніх інструментів і компонентів.

Асоціація з базою даних Oracle, APEX — це простий метод створення веб-додатків, які взаємодіють із базами даних Oracle через SQL і PL/SQL.

Інструмент надає готові шаблони та параметри налаштування, призначені для створення естетично привабливих веб-інтерфейсів.

Універсальності APEX достатньо для створення як простих, так і складних програм, які можуть розвиватися від невеликих проєктів до великих корпоративних систем.

Безкоштовно та бюджетно, APEX входить до випуску Oracle Database і доступний безкоштовно для користувачів платформи без додаткових витрат.

Інструменти безпеки включені в APEX, які запобігають уразливості програм до потенційних небезпек і вторгнень.

APEX полегшує включення рідного JavaScript, CSS і бібліотек сторонніх розробників на додаток до інших доповнень.

Інструмент Oracle APEX популярний завдяки швидкому розвитку веб-додатків, особливо в контексті бази даних Oracle.

Procedural Language/Structured Query Language) — це мова програмування, яка спеціально створена для взаємодії зі сховищами даних Oracle. Це полегшує створення процедур, функцій, тригерів та інших структур для обробки та маніпулювання даними в базі даних Oracle [25].

Основними атрибутами PL/SQL є:

PL/SQL містить процедурні конструкції програмування, такі як умови, цикли та блоки коду, які використовуються для полегшення структури даних і маніпулювання ними.

PL/SQL дозволяє виконувати оператори SQL безпосередньо в коді за допомогою таких конструкцій, як SELECT, INSERT, UPDATE, DELETE та інших.

Мова має додаткові функції, які дозволяють їй більш витончено обробляти винятки та мати більш стабільний код.

Використання інструментів бази даних Oracle. PL/SQL поєднує в собі функціональність бази даних Oracle із можливістю обробки даних, транзакцій та інших аспектів бази даних.

Поряд з Oracle Application Express (APEX), PL/SQL використовується для створення бізнес-логіки та взаємодії з базою даних у веб-додатках, розроблених за допомогою APEX. APEX надає інструменти для створення веб-інтерфейсів, а PL/SQL використовується для операцій з базою даних, перевірки даних та інших завдань на рівні бази даних.

Взаємодія між PL/SQL і Oracle APEX дозволяє створювати потужні та ефективні веб-додатки, які взаємодіють із базою даних Oracle за допомогою процедурного програмування.

2.3 Порівняння обраних інструментів для розробки з аналогами

Порівняння вибраних інструментів для розробки з іншими проектами має вирішальне значення для успіху вашого проекту.

Серед інших мов на основі JavaScript аналогом слід визнати мову Python. Python — це інтерпретована мова високого рівня, яка широко використовується у веб-розробці, наукових дослідженнях, штучному інтелекті та інших сферах. Основні переваги мови Python:

Python відомий своїм гігієнічним і простим стилем навчання, що ідеально підходить для новачків. Його точний синтаксис полегшує ефективне вираження ідей, це популярно в освітніх ініціативах.

Велика екосистема Python включає численні бібліотеки та корисні фреймворки. Це допомагає розробникам швидше завершувати свої проекти та мати більш пряму взаємодію з іншими технологіями.

Python відомий своєю високою продуктивністю та простим використанням. Розробники можуть швидко розширювати функції та з легкістю виконувати завдання завдяки великій кількості вбудованих функцій.

Кросплатформенність. Код, написаний на Python, може працювати в різних операційних системах без серйозних змін. Це робить Python універсальним і портативним як платформа для розробки.

Велика та віддана спільнота розробників Python завжди підтримує та покращує мову. Це полегшує доступ до великої кількості ресурсів і знань, які можна використовувати для вирішення проблем.

Python використовується в багатьох дисциплінах, включаючи веб-розробку, обробку даних, штучний інтелект, машинне навчання тощо. Різноманітність її використання є корисною для мови в багатьох сферах технологій.

Мова Python є універсальною та дозволяє розробникам завершувати проекти в багатьох дисциплінах і областях, ця мова також полегшує реалізацію нових ідей у різних сферах.

Крім того, серед аналогів на основі JavaScript варто виділити мову програмування Java. Java — це потужна об'єктно-орієнтована мова програмування, яка набула значного поширення в різних технологічних сферах. Основними перевагами мови Java є:

Мова Java існує вже понад 20 років і є однією з найбільш послідовних і надійних мов програмування. Мільярди програм, написаних на Java, працюють і сьогодні.

Однією з найвидатніших особливостей Java є здатність виконувати код на різних платформах без необхідності переписування. Це полегшує розробку універсальних програм для різних платформ.

Java сприяє використанню об'єктно-орієнтованого програмування, яке дозволяє створювати модульний код, який легко розширюється. Цей метод підвищує ефективність організації та полегшує розуміння та зміну програм.

Java має різноманітні потужні бібліотеки та фреймворки, які сприяють розробці та розповсюдженню програм. Це сприяє ефективному використанню заздалегідь розроблених рішень для різних завдань.

Java надає розробникам численні інструменти, які вони можуть використовувати для розробки, тестування та налагодження програм. Розширена підтримка інтегрованої розробки (IDE) та інших засобів, що полегшує створення програм.

Java асоціюється з її високою безпекою. Такі механізми, як віртуальна машина Java (JVM), полегшують ізоляцію програм і запобігають потенційним проблемам із безпекою.

Велика та активна спільнота Java підтримує, розробляє нові технології та сприяє обміну досвідом між розробниками.

Java продовжує залишатися потужною платформою для створення численних додатків у світі програмування, її високий ступінь надійності та універсальності робить її популярною.

Інші мови програмування, такі як PHP, також слід розглядати як аналоги мови JavaScript. PHP — це мова програмування високого класу, яка в основному використовується для створення динамічних веб-сайтів і програм. Основними перевагами мови PHP є:

З самого початку PHP в першу чергу призначений для веб-розробки. У ньому є готові функції, які обробляють HTTP-запити, використовують форми, керують сесіями та взаємодіють із базою даних, що ідеально підходить для створення веб-сайтів і програм.

Простий і легкий для вивчення синтаксис PHP в основному використовується початківцями, які хочуть швидко вивчити його та розробити ефективні веб-програми.

Існує багато фреймворків PHP, у тому числі Laravel, Symfony, CodeIgniter тощо, які сприяють швидкій і легкій розробці великих і складних веб-проектів.

PHP сумісний з кількома системами керування базами даних, включаючи MySQL, PostgreSQL, SQLite та інші, що забезпечує ефективний обмін даними.

Завдяки простоті використання та численним готовим рішенням PHP дозволяє розробникам швидко створювати та розгортати веб-програми.

PHP успішно працює на багатьох операційних системах, що робить його універсальною платформою для створення веб-додатків на різних пристроях.

Велика та різноманітна спільнота розробників PHP підтримує, просуває та сприяє розвитку мови.

РНР продовжує залишатися популярним інструментом для веб-розробки, він надає різноманітні функції, необхідні для створення різноманітних веб-проектів.

Значну роль у процесі розробки відіграє вибір платформи розробки для серверної складової програми, ця платформа має істотний вплив на надійність, масштабованість і продуктивність системи. Вибрана серверна архітектура може вплинути на продуктивність програми, її здатність витримувати навантаження та потенційну легкість майбутнього розширення функціональності. Вибір відповідної технології для серверної частини бази даних дозволяє підтримувати високий ступінь безпеки, ефективно управляти базою даних і швидко обмінюватися інформацією між клієнтами і сервером.

Сьогодні на ринку представлено багато програмного забезпечення, призначеного для впровадження ERP-систем різного рівня складності. Серед конкурентів Oracle Application Express можна виділити Apache OpenOffice Base, Appian, Appery.io та інші.

Apache OpenOffice Base — це повна система керування базами даних, призначена для задоволення потреб різноманітних користувачів. Від керування особистими колекціями компакт-дисків до створення щомісячних звітів, призначених для корпоративного використання, він повний функцій. База даних містить внутрішні інструменти, які полегшують створення таблиць, запитів, форм і звітів, а також створений користувачем набір значень для відстеження активів, клієнтів, замовлень, рахунків-фактур та інших об'єктів.

Base надає повноцінну базу даних для реляційних даних, налаштовану на роботу з одним користувачем, із даними, що зберігаються безпосередньо в Base, а також підтримує файли dBase, які є плоскими.

Для користувачів, які мають розширені бізнес-інтереси, Base забезпечує підтримку різноманітних складних багатокористувацьких механізмів баз даних: MySQL, Adabas D, MS Access і PostgreSQL. Крім того, підтримка загальних драйверів JDBC і ODBC дозволяє підключатися майже до будь-якої бази даних [31].

Appian — це провідна платформа з низьким кодом, яка спеціалізується на швидкій розробці комерційних додатків і автоматизації процесів для бізнесу. Ось список основних функцій і можливостей платформи Appian:

- управління бізнесом і процесами;
- розмовні боти та віртуальні помічники;
- інтеграція з UIPath;
- інтеграція із зовнішніми ресурсами;
- оцінка розвідки та звітність;
- широкий спектр можливостей низькорівневого кодування.

Appian — це універсальна та потужна платформа, яка сприяє швидкій та ефективній розробці додатків великих корпорацій.

Microsoft Power Apps — це низькорівнева програмна платформа, призначена для швидкої розробки великих корпоративних програм без тривалого програмування. Ось деякі з ключових атрибутів і можливостей платформи Microsoft Power Apps:

Візуальне створення програмних додатків. (PowerApps пропонує візуальний підхід до розробки додатків. Розробники можуть використовувати компоненти інтерфейсу перетягування та персоналізувати їх без глибоких знань програмування.)

Інтеграція з Microsoft 365 та іншими програмними додатками. (PowerApps легко поєднується з іншими службами Microsoft, такими як SharePoint, Excel і Microsoft Azure. Це полегшує використання даних і служб екосистеми Microsoft).

Автоматизація бізнес-процесів. (PowerApps дозволяє автоматизувати звичайні бізнес-процеси за допомогою попередньо встановлених потоків і автоматизації).

Доступність даних в режимі реального часу. (Платформа надає можливість використовувати дані в режимі реального часу, це надає користувачам найновішу інформацію).

Низькорівневе кодування та можливості розширення. (Power Apps полегшує використання низькорівневого кодування для створення більш складних функцій і доповнень.)

Повний доступ і функції безпеки. (PowerApps пропонує кілька методів керування доступом до даних і підвищення безпеки програми).

Microsoft Power Apps — це потужна платформа для організацій, які хочуть швидко створювати та впроваджувати програми для великих підприємств, не витрачаючи багато грошей на програмування.

3 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПІДСИСТЕМИ УПРАВЛІННЯ МАТЕРІАЛІВ ДЛЯ ЗВАРЮВАЛЬНИХ РОБІТ ERP «ПЛАЗМІС»

Аналіз функціональних вимог користувача для модуля «Волочіння зварювального дроту» в підсистемі

Проектування інформаційних систем має вирішальне значення для розробки програмного забезпечення. Без належного планування неможливо виконати вимоги користувачів і основні вимоги системи, такі як збереження якості даних, їх ефективне використання, швидкий доступ до інформації, проста реалізація запитів користувачів і надійний захист.

Крім того, дуже важливо враховувати майбутні потреби користувачів системи.

Створення програмних компонентів, які будуть об'єднані з існуючою системою, включає спочатку визначення різних функціональних можливостей, які повинні бути реалізовані цими компонентами, а також опис способу взаємодії цих компонентів з користувачем і системою. Для спрощення технічного завдання, яке виражається в письмовій формі, використовуються діаграми поведінки UML, які сприяють формалізації послідовності дій і зв'язку між ними [34, 35].

Діаграма випадку UML призначена для того, щоб показати функціональність системи без урахування структури системи. Він описує технічне завдання з точки зору користувача, надаючи водночас вичерпний і стислий опис представлення системи та рівнів доступу до функцій, а також потенційне індивідуальне застосування кожної функції. Узагальнення та визначення, надані цією діаграмою, сприяють подальшій розробці програмного продукту та є важливими в документації програмного продукту.

У даному випадку інтерфейс «Волочіння зварювального дроту попереднє» розроблений для користування усіма користувачами ERP системи «ПлазмІС». Однак, у підсистемі є перевірка на наявність прав користувача під назвою:

«Виробництво: Журнали реєстрації партій». У даному випадку, якщо у користувача наявні ці права, у нього відкривається можливість користування усіма можливостями підсистеми виробництва. Це необхідно для уникнення помилок під час пошуку та створення партій, розрахунки заробітної плати, тощо. Наприклад, операціоніст може шукати партію на одну вагу, а користувач в будь-який момент може змінити вагу і тоді створиться конфліктна ситуація. Щоб цього уникнути, було розроблено права доступу «Виробництво: Журнали реєстрації партій» [36].

На рисунку 3.1 наведено діаграму варіантів використання функціонального модуля «Волочіння зварювального дроту попереднє» для звичайного користувача ERP системи «ПлазмІС». Даний модуль передбачає дві основні діяльності користувача: проходження процедури авторизації та власне робота з створенням змінного завдання. Процедура авторизації передбачає вибір персональних даних для прив'язки сеансу роботи з базою даних. Варто зазначити, що цей функціональний підблок є залежним від блоку авторизації, тому користувачеві безпосередній зв'язок з ним не надається.

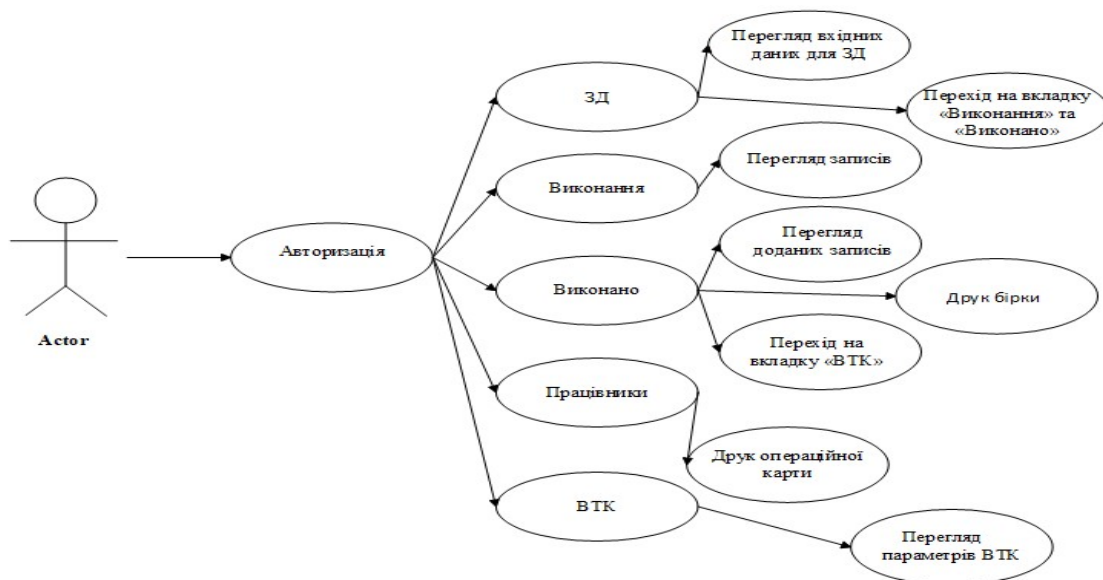


Рисунок 3.1 — Діаграма варіантів використання функціонального модуля звичайного користувача

Робота з інтерфейсом заявки на перевезення для звичайного користувача передбачає операції перегляду, друку форм.

На рисунку 3.2 наведено діаграму варіантів використання функціонального модуля «Волочіння зварювального дроту попереднє» для волочильника або відповідальної особи з правами «Виробництво: Журнали реєстрації партій». Відповідальна особа має ті самі можливості, що і звичайний користувач, але до них додаються інші можливості такі як: сканування вхідних матеріалів, що в свою чергу приводить до створення операційної карти, додавання записів в операційну карту з вкладки «Виконання», додавання параметрів ВТК тощо.

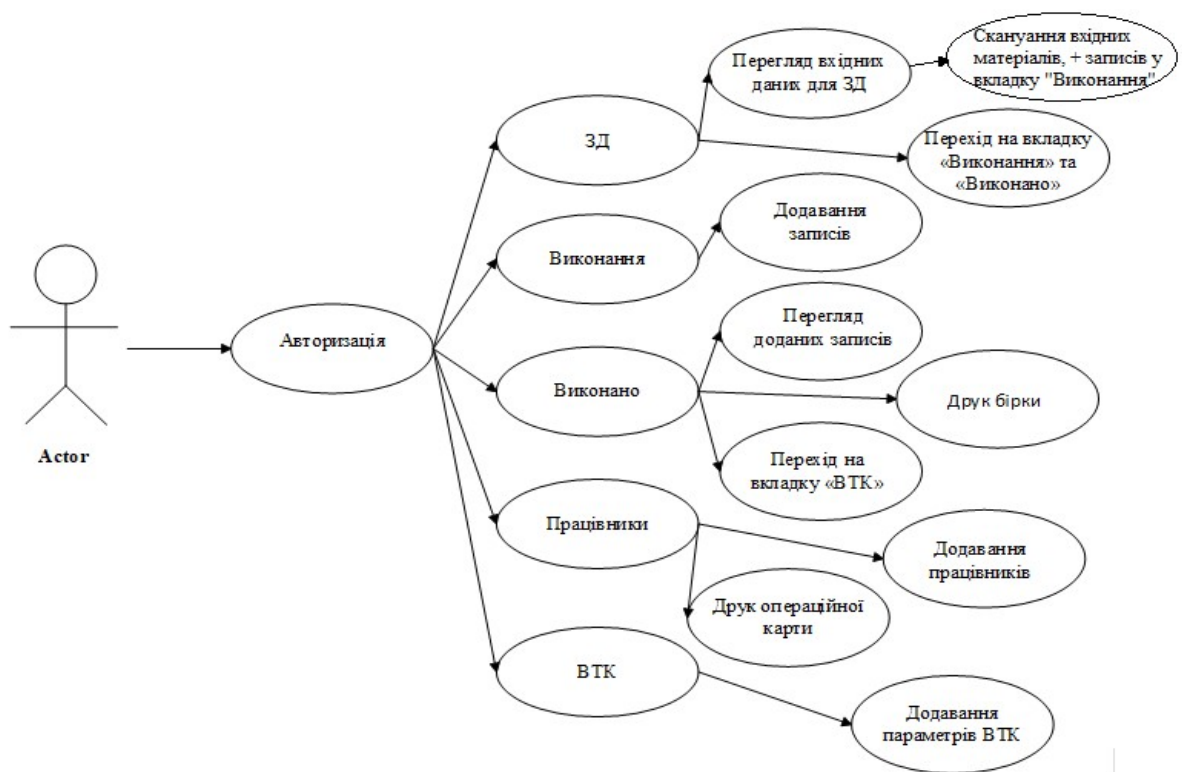


Рисунок 3.2 — Діаграма варіантів використання функціонального модуля відповідальної особи(логіста)

У результаті ці статистичні дані дозволяють нам зробити висновок про масштаби майбутньої розробки програмного забезпечення, але вони не визначають конкретні структурні компоненти програмного забезпечення. Це означає, що не рекомендується використовувати виключно діаграму випадків на

етапі розробки. У результаті слід використовувати більш детальні діаграми, такі як діаграма послідовності.

Діаграма послідовності ілюструє життєвий цикл об'єкта через взаємодію різних індивідів на шкалі часу. Цей малюнок показує поведінку програмних компонентів і користувачів, він дозволяє продемонструвати ступінь паралелізму між їхніми діями та розкрити специфіку функціоналу, описаного на попередньому малюнку.

Іншим корисним аспектом діаграми послідовності є її здатність описувати складні процеси за допомогою потоків керування та складного набору об'єктів.

На рисунку 3.3 представлена діаграма послідовності для користувача, а на рисунку 3.4 - для відповідальної особи.

У результаті ці статистичні дані дозволяють нам зробити висновок про масштаби майбутньої розробки програмного забезпечення, але вони не визначають конкретні структурні компоненти програмного забезпечення. Це означає, що не рекомендується використовувати виключно діаграму випадків на етапі розробки. У результаті слід використовувати більш детальні діаграми, такі як діаграма послідовності.

Діаграма послідовності ілюструє життєвий цикл об'єкта через взаємодію різних індивідів на шкалі часу. Цей малюнок показує поведінку програмних компонентів і користувачів, він дозволяє продемонструвати ступінь паралелізму між їхніми діями та розкрити специфіку функціоналу, описаного на попередньому малюнку.

Іншим корисним аспектом діаграми послідовності є її здатність описувати складні процеси за допомогою потоків керування та складного набору об'єктів.

Якщо авторизацію здійснив звичайний користувач він може виконувати наступні дії.

Під час взаємодії з другим об'єктом (ЗД), головний інтерфейс активує базу даних, запитуючи інформацію про підприємство та права користувача. Отримані результати відображаються у вкладці «ЗД».

Вкладка «Виконання», що слугує третім об'єктом системи, реагує на запит користувача відкривання її та відображає усі додані записи з вкладки «ЗД».

Четвертим об'єктом є вкладка «Виконано». Користувач має змогу переглянути рядки з які додали на попередній вкладці, або ж рядки з операційної карти, роздрукувати необхідну йому друковану бірку для подальшого використання.

П'ятим об'єктом є вкладка «Працівники». Дана вкладка дає змогу здійснити друк операційної карти.

Шостим об'єктом є вкладка «ВТК». За допомогою цієї вкладки користувач здійснює перегляд параметрів ВТК.

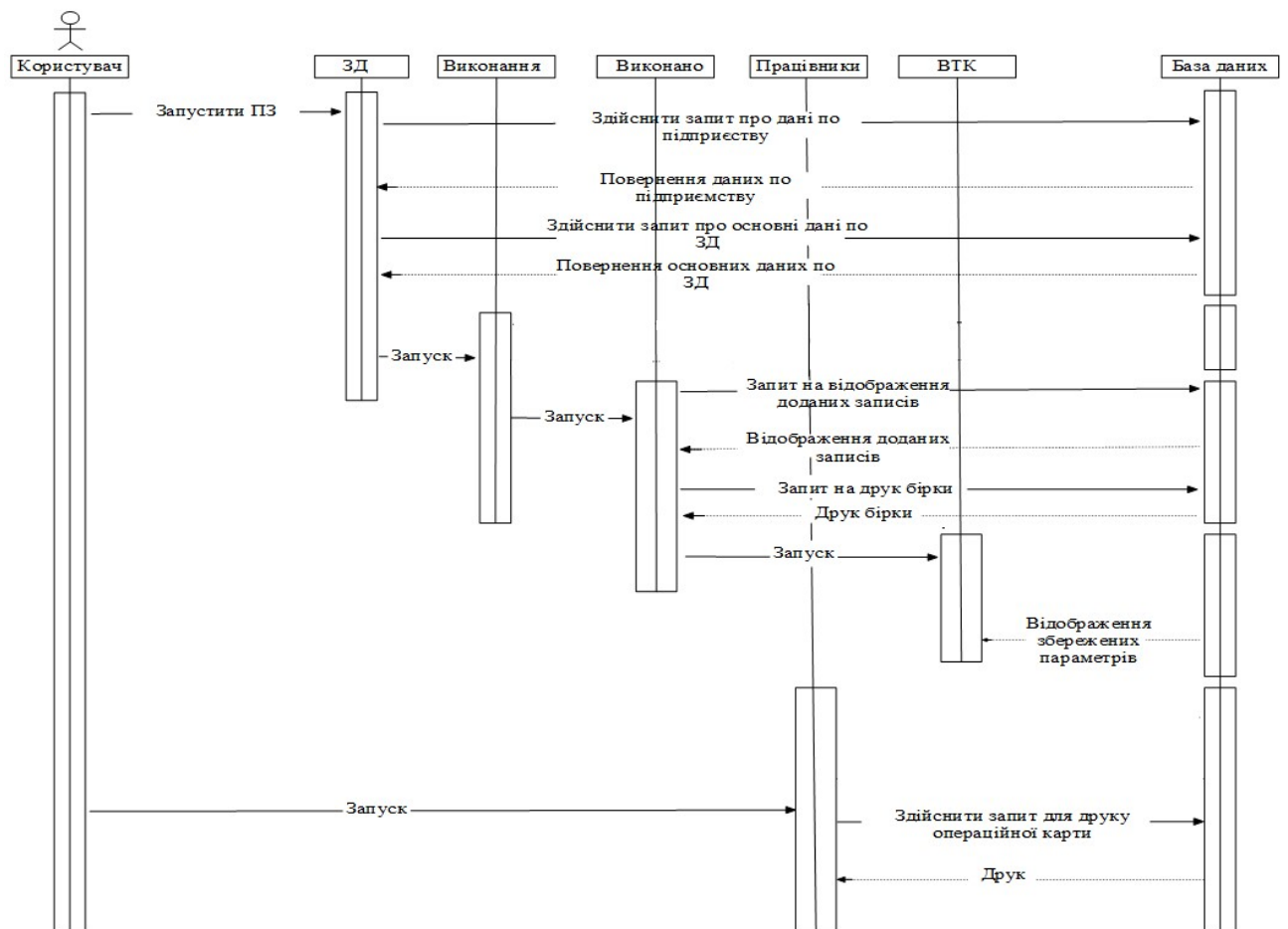


Рисунок 3.3 — Діаграма послідовностей функціонального модуля «Волочіння зварювального дроту попереднє» користувача

Коли авторисувалась відповідальна особа пояснення діаграми послідовностей настуна.

Під час взаємодії з другим об'єктом (ЗД), головний інтерфейс активує базу даних, запитуючи інформацію про підприємство та права користувача. Отримані результати відображаються у вкладці «ЗД». Під час роботи з цією вкладкою відбувається збереження відсканованих вхідних матеріалів до бази даних, після чого вікно повідомляє про успішне збереження. У цей ж момент створюється документ «Операційна карта» у статусі «редагується» після чого, користувач може додавати нові записи в цей документ.

Вкладка «Виконання», що слугує третім об'єктом системи, реагує на запит користувача відкривання її та відображає усі додані записи з вкладки «ЗД», після чого ми можемо додавати рядки в операційну карту з данної вкладки, якщо документ «Операційна карта», не в статусі «проведено».

Четвертим об'єктом є вкладка «Виконано». Користувач має змогу переглянути рядки з які додали на попередній вкладці, або ж рядки з операційної карти, роздрукувати необхідну йому друковану бірку для подальшого використання.

П'ятим об'єктом є вкладка «Працівники». Дана вкладка дає змогу додавати працівників в операційну карту та здійснити друк операційної карти.

Шостим об'єктом є вкладка «ВТК». За допомогою цієї вкладки ми здійснюється додавання параметрів ВТК та їх перегляд.

Деталізація за допомогою діаграми послідовності полегшує розробку майбутнього програмного забезпечення більш комплексно та визначає функціональні компоненти, які, як очікується, будуть реалізовані.

Як наслідок, використання інструменту UML продемонструвало принципи роботи модуля «Запит на транспортування» у зрозумілій та інформативній формі. У цьому випадку діаграма варіантів використання UML сприяла більш розширеному опису конкретного варіанту використання, цей опис включав послідовність дій, що проходять через систему. Діаграма послідовності надає

найдетальніший опис зв'язків у коді програми, тоді як попередня діаграма допомагає визначити, які обов'язки пов'язані з попередніми ролями.

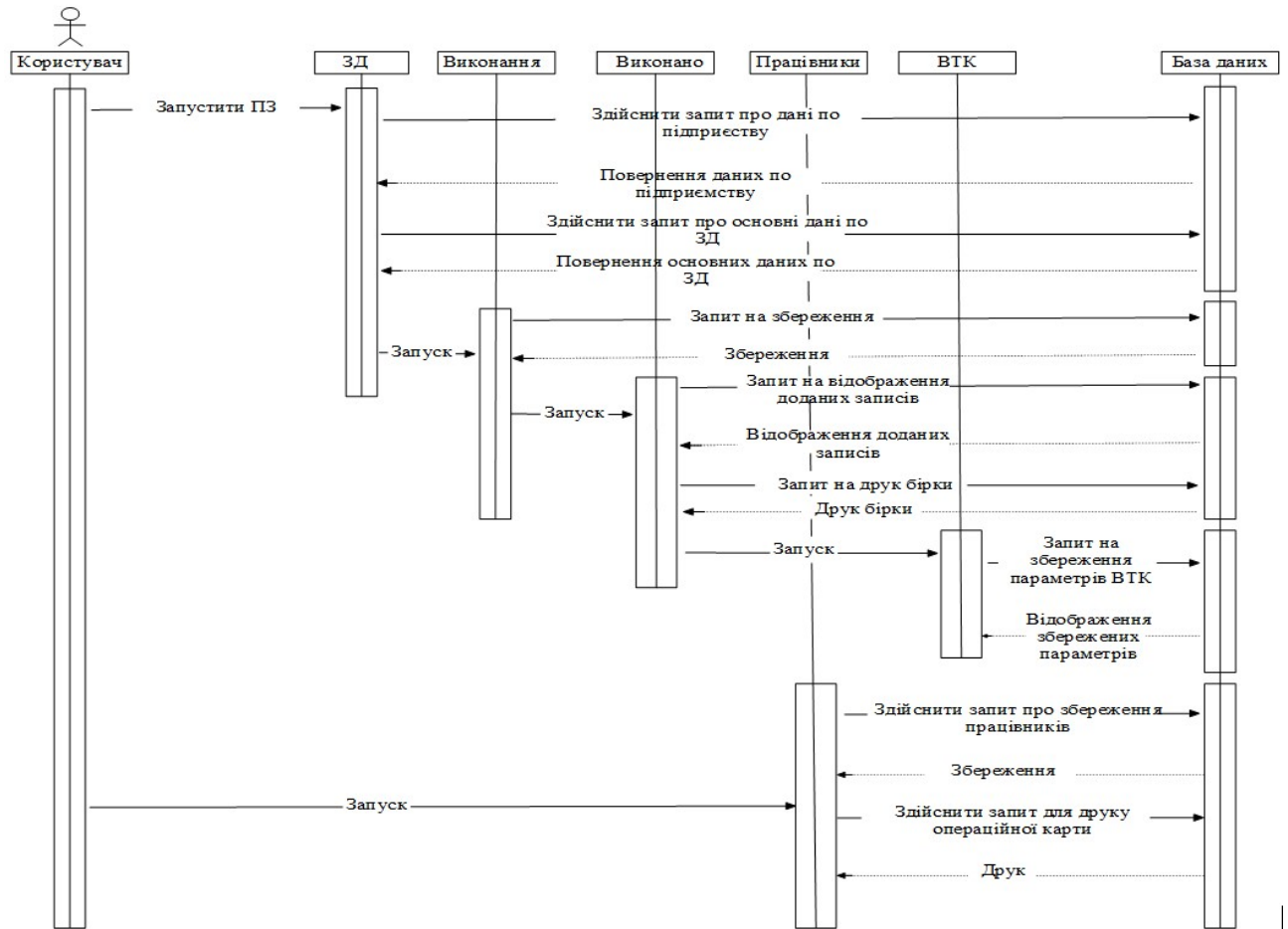


Рисунок 3.4 — Діаграма послідовностей функціонального модуля відповідальної особи

Деталізація за допомогою діаграми послідовності полегшує розробку майбутнього програмного забезпечення більш комплексно та визначає функціональні компоненти, які, як очікується, будуть реалізовані.

Як наслідок, використання інструменту UML продемонструвало принципи роботи модуля «Запит на транспортування» у зрозумілій та інформативній формі. У цьому випадку діаграма варіантів використання UML сприяла більш розширеному опису конкретного варіанту використання, цей опис включав послідовність дій, що проходять через систему. Діаграма послідовності надає

найдетальніший опис зв'язків у коді програми, тоді як попередня діаграма допомагає визначити, які обов'язки пов'язані з попередніми ролями.

За допомогою розширеної документації щодо графіки ви можете переходити до етапів проектування бази даних і розробки програмного забезпечення.

3.2 Проектування бази даних

Розробка бази даних для реляційної моделі, яка використовує колекції, здійснюється шляхом створення таблиць і встановлення зв'язків між ними для заповнення цих таблиць інформацією. У реляційній моделі дані представлені таблицями, які мають стовпці (атрибути) і рядки (кортежі). Колекції можна використовувати для підтримки та обробки даних у моделі реляційної бази даних.

Основні переваги моделі реляційної бази даних:

- чіткість;
- гнучкість;
- зв'язки між даними;
- незалежність фізичного зберігання даних;
- запитання та можливості для аналізу;
- обробки даних.

Чіткість — реляційна модель має визначену структуру, яка організовує дані у формі таблиць із стовпцями та рядками. Це сприяє простому розумінню та сприйняттю інформації.

Гнучкість — реляційна модель не може змінювати дані без зміни структури бази даних. Це підвищує гнучкість і універсальність системи для задоволення мінливих потреб.

Зв'язки між даними — реляційна модель полегшує зв'язок таблиць за допомогою ключів. Це дає змогу зв'язувати дані з різних таблиць, що забезпечує ефективний і простий доступ до даних.

Незалежність фізичного зберігання даних — реляційна модель дає змогу відмежуватися від специфіки фізичного зберігання даних, це дає змогу використовувати різні методи зберігання (наприклад, у пам'яті чи на диску), не змінюючи логіку бази даних.

Запитання та можливості для аналізу — реляційна модель пропонує широкий набір операцій і запитань, які дозволяють виконувати різноманітні аналізи.

Обробки даних — це полегшує виконання складних запитів і отримання потрібної інформації.

Загалом модель реляційної бази даних із колекціями має численні переваги, зокрема структуру, гнучкість, взаємозв'язки даних, фізичну незалежність зберігання даних і широкі можливості аналізу. Це робить її однією з найпопулярніших і часто використовуваних моделей баз даних.

Основна мета даної розробки - трансформація даних про виконану роботу на виробництві «Волочіння зварювального дроту попереднє» в зміну статусу корпорації ПлазмаТек.

Оскільки база даних системи «ПлазмІС» заснована на відношеннях, то в даному випадку також була використана модель відношень. Враховуючи предметну область і діаграми UML, можна було створити структуру частини бази даних, пов'язаної з підсистемою «Волочіння зварювального дроту попереднє», як показано на рисунку 3.5.

Відповідно до поставленого завдання, модель БД вимагає п'яти логічних компонентів, якими є: сторони, підприємство, матеріальна цінність, документи-предки та таблиця, яка з'єднує два документи (змінне завдання та оперативну карту). Відповідно до цих сутностей було створено п'ять таблиць, які відображаються у форматі БД із пов'язаними зв'язками.

Таблиця «Річ за документом» (`doc_item`) описує необхідні умови зберігання та містить сім полів. Серед них першочергове значення має поле `doci_id`, воно є первинним ключем, `doci_doc` є параметром документа, а також є вторинним ключем для взаємодії з таблицею.

«Партії», ідентифікатор партії є первинним ключем, він використовується для взаємодії з таблицею «Речі», а також сировиною, продуктами, і всі вони ідентифікуються відповідними ідентифікаторами.

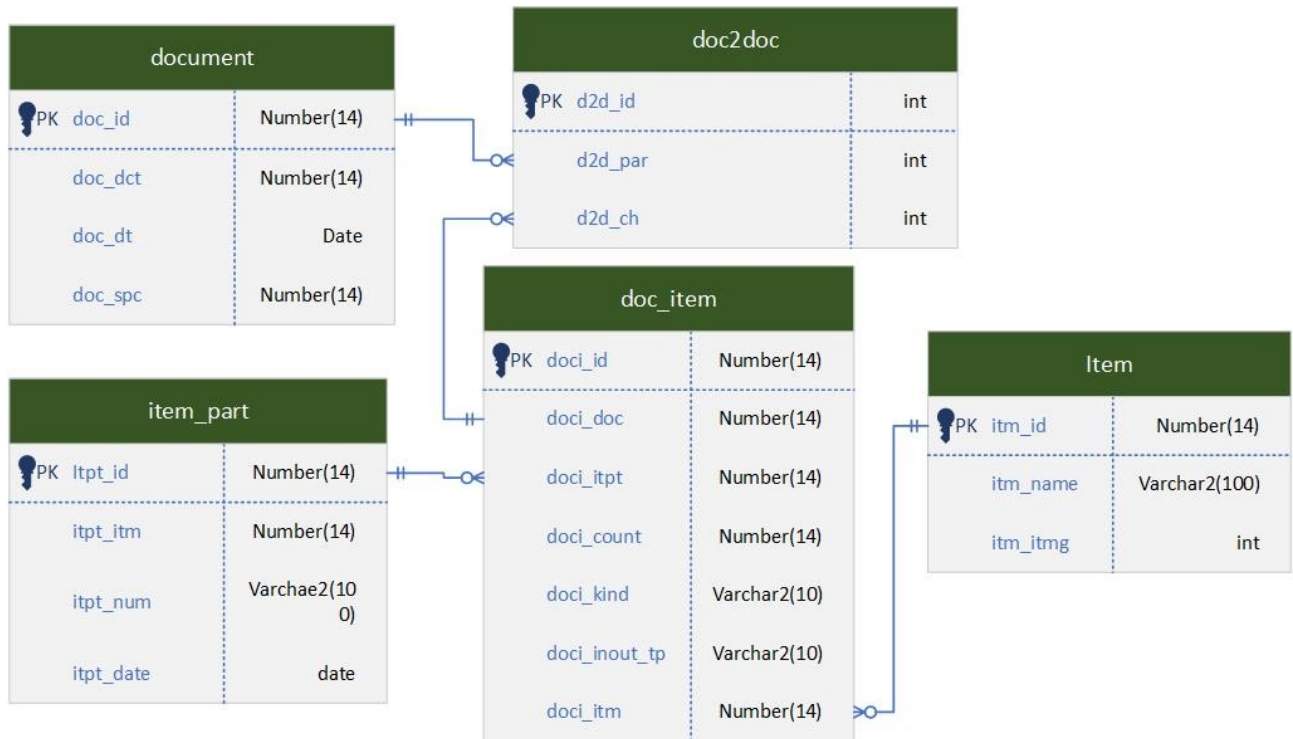


Рисунок 3.5 - Структура частини БД підсистеми для модуля

3.3 Розробка алгоритму функціонування модуля «Заявка на перевезення» підсистеми логістики

Алгоритм - це послідовність чітко визначених кроків, спрямованих на вирішення певної задачі чи виконання обчислень. Ці інструкції створюються для визначення точного порядку виконання завдань. Алгоритми є важливим інструментом для обробки даних та виконання обчислень. Сучасні алгоритми, які базуються на принципах штучного інтелекту, можуть надавати автоматизовані рішення та керувати виконанням програмного коду, використовуючи різноманітні шляхи, які не завжди можна передбачити заздалегідь [19].

На рисунку 3.6 відображено блок-схему функціонування модуля «Волочіння зварювального дроту попердне» підсистеми виробництва.

Алгоритм відображає принцип роботи підрозділу для звичайного користувача та для відповідальної особи(логіста), його основні функції (створення нового запису, редагування діючого запису, переведення статусу, сповіщення відповідальної особи, коректне заповнення полів тощо). Як результат – актуальні дані підсистеми будуть збережені.

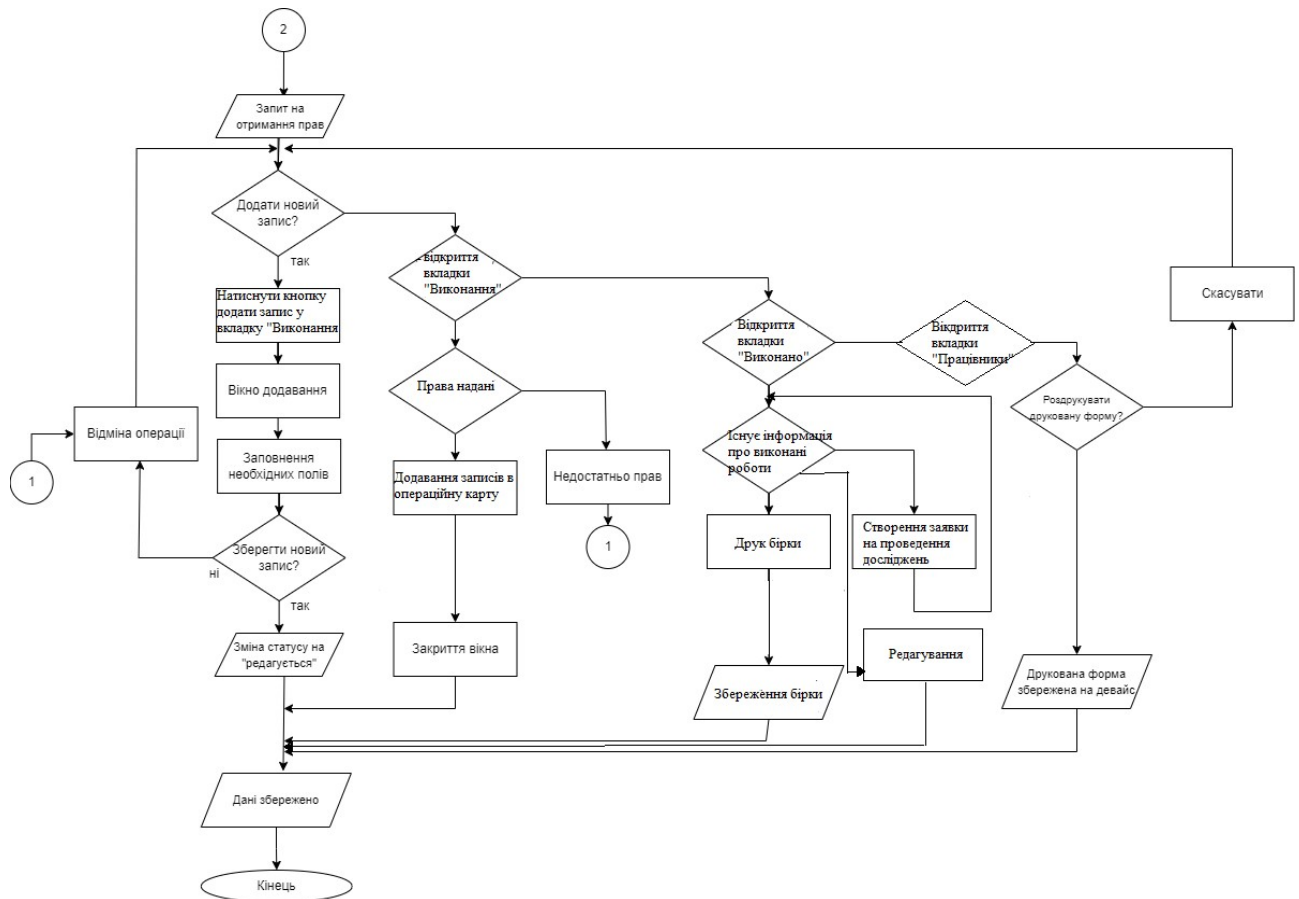


Рисунок 3.6 – Блок-схема функціонування модуля «Волочіння зварювального дроту попереднє» підсистеми виробництво

Першим етапом роботи програми є запит на отримання даних. Якщо операцію пошуку схвалено, дані виводяться на графічному інтерфейсі згідно з запитом.

Після цього здійснюється перевірка потреби здійснення нового запису. Користувач повинен натиснути кнопку «створити новий запис», заповнити параметри у вікні додавання та відправити запис на збереження. В процесі

збереження можна здійснити відміну додавання. Після цього алгоритм повертається до запиту необхідності додавання нового запису.

Другим варіантом дій користувача є перехід на вкладку «Виконання» натиснувши на її назву, тут користувач переходить до редакції нового запису(якщо це потрібно), що включає в себе натиснення кнопки «редагувати запис», появу вікна редагування, перевіркою чи документ у статусі «Редагується» та заповнення необхідних полів з подальшим збереженням, що передбачає також можливість відміни операції.

Четвертим варіантом є друк форми, та додавання працівників.

Усі чотири варіанти можна запускати в довільному порядку, необхідному користувачеві, будь-яку кількість разів до моменту переведення статусу документу в «Проведено». Після збереження необхідних змін користувач може вийти з програми та завершити її роботу, що відображено як у діаграмі послідовностей, так і на блок-схемі.

4 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ «ВОЛОЧІННЯ ЗВАРЮВАЛЬНОГО ДРОТУ»

4.1 Проектування користувацької частини інтерфейсу функціонального модулю підсистеми змінні завдання

Підрозділ «Волочіння зварювального дроту » створено як окрему веб сторінку в складі виробництва. Цей розділ має складатися з чотирьох вкладко. На рисунку 4.1 наведено інтерфейс сторінки «Волочіння зварювального дроту попереднє».

Завдання	Виконання	Виконано	Працівники	
ЗдВчО-2412-089 ві...				
G4Si1	6.1	345/61/1/0т-12	дріт д/омд d2	+1
G4Si1	6.1	345/55/1/0т-12	дріт д/омд d2	+1
G4Si1	6.1	345/22/1/0т-12	дріт д/омд d2	+1
G4Si1	6.1	345/53/1/0т-12	дріт д/омд d2	+1

Рисунок 4.1 — Вигляд сторінки підрозділу «Волочіння зварювального дроту попереднє».

Основна мета дизайну інтерфейсу - забезпечити його зрозумілість, потужність і чуйність для користувача під час взаємодії з екраном. Прагнення до ефективності та доступності є вигідним, оскільки це гарантує легкість взаємодії користувача з системою. Крім того, тестування інтерфейсу на різних платформах і пристроях є надзвичайно важливим для забезпечення належного відображення та функціональності інтерфейсу.

На першій вкладці «ЗД» користувач бачить номер змінного завдання, рядки із вхідними даними(з чого і що потрібно зробити). Перша колонка це марка катанки, наспуна колонка діаметр катанки, третя колонка є контрольний код катанки, четверта що потрібно зробити, остання колонка це кнопка для відкривання вікна, щоб додати записи у вкладку «Виконання». Натиснувши на рядок користувачу відкривається деталізація з поясненням даних, деталізація

наведена на рисунку 4.2. На рисунку 4.3 наведено вікно додавання записів у вкладку «Виконання».

Завдання	Виконання	Виконано	Працівники
ЗдВЧО-2412-089 ві...			
G4Si1	6.1	345/61/1/0т-12	дріт д/омд d2
Марка дроту: G4Si1			+1
Діаметр: 6.1			+1
Партія: 345/61/1/0т-12			+1
Запланована кількість: 0.6			+1
Маршрут: 13/2			+1
Виготовити: дріт д/омд d2			+1
	Виконання	Виконано	

Рисунок 4.2 — Вигляд деталізації на вкладці «ЗД».

×

Волочіння зварювального дроту попереднє

Код: 345/61/1/0т-12 [G4Si1 Ø6.1]

Внесіть штрих-код

Обладнання:

ЛПВ {ЦО-001} {ЦО-001} {SN: 55453}

✓ Зберегти

✗ Відмінити

Рисунок 4.3 — Вигляд вікна додавання записів у вкладку «Виконання».

В данному вікні користувачу показується інформація, що якій із бухт катанки йому потрібно просканувати штрих-код, та на якому обладнанні він працює. Кнопка «Відмінити» закриває вікно, кнопка «Зберегти» після успішного сканування стає доступною та після її натискання додає рядок у вкладку «Виконання», кнопка «штрих-код» відкриває вікно для підбору штрих-кода по останнім символам штрих-кода(на випадок якщо фізичний штрих-код не сканується), вікно для підбору штрих-коду наведено на рисунку 4.4.

Приклад другої вкладки «Виконання» з розгорнутою деталізацією наведено на рисунку 4.5. На цій вкладці користувач, а саме волочильник дроту, бачить що він вже взяв в роботу, деталізацію з поясненням та додає рядки в операційну

карту, щоб додати рядок потрібно натиснути кнопку «+1» після чого відкриється вікно додавання, вигляд вікна наведено на рисунку 4.6.

Рисунок 4.4 — Вигляд вікна «Підбір штрих-коду».

Приклад другої вкладки «Виконання» з розгорнутою деталізацією наведий на рисунку 4.5. На цій вкладці користувач, а саме волочильник дроту, бачить що він вже взяв в роботу, деталізацію з поясненням та додає рядки в операційну карту, щоб додати рядок потрібно натиснути кнопку «+1» після чого відкриється вікно додавання, вигляд вікна наведено на рисунку 4.6.

Рисунок 4.5 – Вигляд вкладки «Виконання» з розгорнутою деталізацією

На третій вкладці «Виконано» волочильник бачить, що він вже зробив, може роздрукувати бірку для подальшого використання, відредагувати данний рядок та створити заявку на проведення досліджень приклад вкладки «Виконано» з розгорнутою деталізацією наведено на рисунку 4.7.

Волочіння зварювального дроту попереднє

ЛПВ {ЦО-001} {ЦО-001} {SN: 55453}

Код: 345/61/1/0т-12

Код нащадка 345/61/1/0т-12/3

№ тари Вага тари

Вага загальна Вага продукції

Зберегти Відмінити

Рисунок 4.6 – Вигляд вікна додавання записів в операційну карту.

Завдання	Виконання	Виконано	Працівники
Балтянский В. В. O2/1-2412-044 від 16.12.24;			
ЛПВ {ЦО-001} {ЦО-001} {SN: 55453}		345/61/1/0т-12/2 дріт д/омд d2	586
Обладнання: ЛПВ {ЦО-001} {ЦО-001} {SN: 55453}			
Партія вхідна : 345/61/1/0т-12 дріт d6.1			
Партія вихідна: 345/61/1/0т-12/2 дріт д/омд d2			
Вага, (кг): 586			
Вага браку, (кг): 14			
Дата і час: 16.12.2024 09:53:11			
ЗмВМП-2412-1918			
ЗмВМП-2412-1919			

Рисунок 4.7 – Вигляд вкладки «Виконано» з розгорнутою деталізацією.

На четвертій вкладці «Працівники» відповідальна особа або ж сам волочильник може додати працівників в операційну карту та роздрукувати саму операційну карту, приклад вкладки наведено на рисунку 4.8.

Завдання	Виконання	Виконано	Працівники
Балтянский Володимир Вікторович		волочильник дроту	
Додати працівника			
Операційна карта			

Рисунок 4.8 – Вигляд вкладки «Працівники»

4.2 Реалізація обробки даних в базі даних

У модулі «Волочіння зварювального дроту попереднє» можна проводити операції: додавання даних в таблицю, та редагування даних в таблиці.

Для виконання процедури додавання є спеціальна функція, яка заповнює масив даних, та передає це масив в процес який використовує процедуру зберігання. На вкладці «Виконання» у вікні додавання рядків користувач повинен першочергово внести номер тари, після чого автоматично підставиться ваги цієї тари, далі потрібно заповнити загальну вагу, після цього автоматично заповниться вага продукції, після натискання на кнопку «Зберегти» відпрацює процедура додавання.

Для виконання процедури редагування є спеціальна функція, яка заповнює масив даних, та передає це масив в процес який використовує процедуру редагування. На вкладці «Виконано» у вікні редагування рядків користувачу дається можливість редагування тари та загальної ваги, вага продукції та тари автоматично розраховується, після натискання на кнопку «Зберегти» відпрацює процедура редагування.

еалізація взаємодії клієнтської та серверної частини в підсистемі

Платформа Oracle Application Express (APEX) розроблена для взаємодії з серверною частиною та інтерфейсом користувача. Основною частиною APEX є App Builder, який є невід’ємною частиною створення програм. Тут доступні кілька варіантів [22]:

— I

- використовується для створення нових програм з інтерфейсу.
- ця функція дозволяє імпортувати програми шляхом передачі їх з одного сервера на інший у форматі текстового файлу.
- надає інформацію та засоби для перегляду розробки програм.
- додаткові інструменти, які допомагають в адмініструванні програм і робочих процесів.

Рисунок 4.9 ілюструє компонент App Builder.

Oracle Application Express (APEX) — це універсальний інструмент для розробки потужних і надійних рішень сучасних проблем, які одночасно є потужними та надійними.

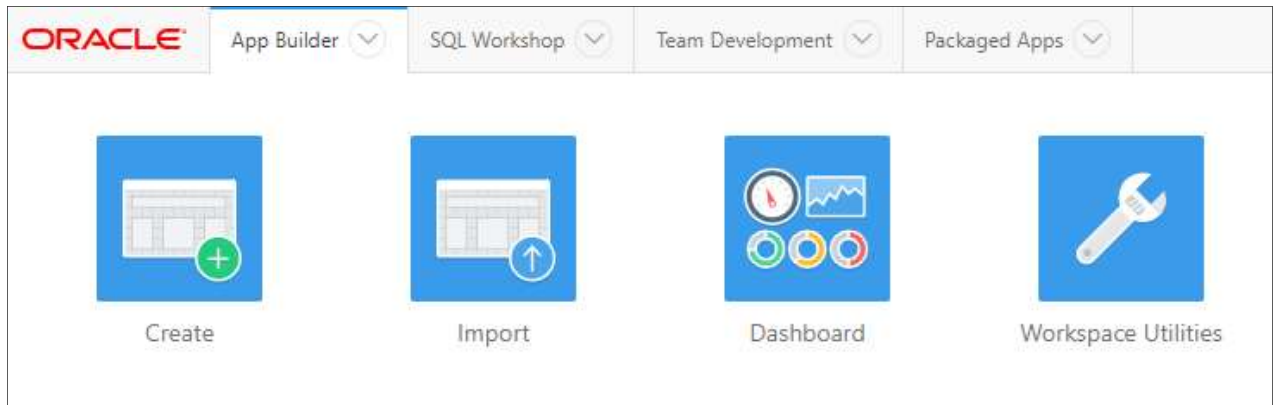


Рисунок 4.9 – Розділ App builder.

Використання JavaScript дає змогу створювати інтерфейси, які максимально підвищують ефективність користувачів. Цей фреймворк постійно доступний із усіх функцій Oracle Database, що надає численні можливості для покращення функціональності програм.

Найважливіше те, що APEX функціонує в кількох сценаріях, він створює прості внутрішні програми компанії, а також великі, складні системи, якими користуються тисячі людей одночасно. Цей фреймворк легко розширюється зі збільшенням обсягу даних, що дозволяє вирішувати найскладніші проблеми.

Однією з його переваг є простота встановлення, оскільки APEX реалізовано за допомогою звичайного веб-браузера. Це значно зменшує труднощі розробки та використання системи, усуває потребу в спеціалізованому програмному забезпеченні для роботи з фреймворком.

Одним із найважливіших компонентів App Builder є Page Designer, який функціонує як автор сторінки. Тут доступні різноманітні інструменти, серед яких: можливість підключення файлів JS/CSS, створення різноманітних процесів (для списків, відображення даних у вигляді дерева, обробки подій кнопок тощо), додаткова персоналізація стилів, фільтрів і макетів. Кожна сторінка присвячена певному номеру.

На малюнку 4.10 показано, як розділ «Page Designer» відображається через розділ «Волочіння зварювального дроту попереднє» як аналог.

Application Express перетинає Інтернет через веб-браузер із веб-приймачем у базі даних Oracle. Це означає, що, окрім браузера, для створення, розгортання та роботи не потрібно жодного іншого клієнтського програмного забезпечення. Сторінки, які стосуються програми Express, відображаються у веб-браузері як вміст HTML. Запити та дії на сторінці виконуються механізмом Application Express у базі даних Oracle.

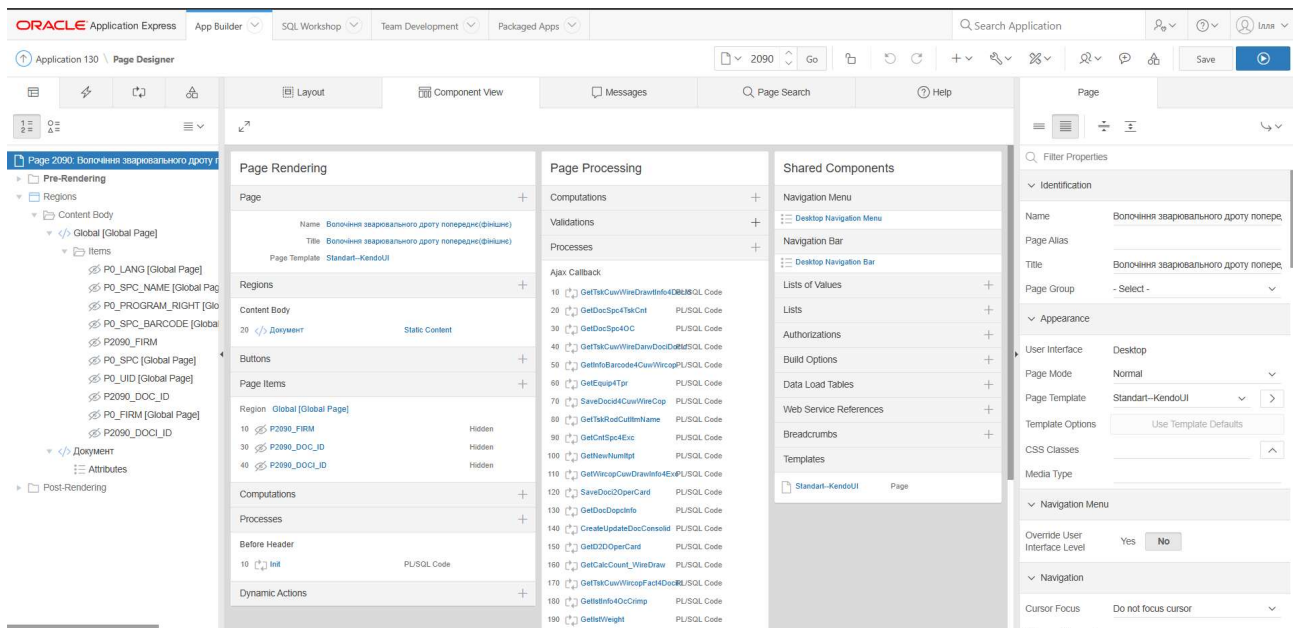


Рисунок 4.10 – Розділ Page Designer

На рисунку 4.11 наведено архітектуру підсистеми виробництво.

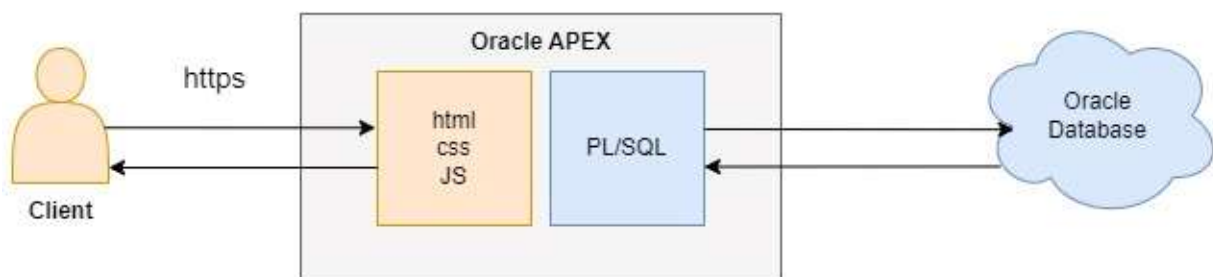


Рисунок 4.11 – Архітектура підсистеми виробництво

Oracle Application Express — це повністю інтегроване програмне забезпечення, яке працює безпосередньо в базі даних, забезпечує керування даними таблиць і виконує коди PL/SQL. Незалежно від того, створюєте ви нову програму чи працюєте над поточним проектом, процедура залишається незмінною. Браузер починає URL-запит, який обробляється системою Application Express за допомогою PL/SQL. База даних інтерпретує цей запит і повертає результат як HTML на ваш комп'ютер. Ці дії зазвичай виконуються під час запиту або завантаження сторінки.

Стан сеансу програми безпосередньо керується в базі даних через систему Application Express, і спеціальне постійне з'єднання не використовується. Кожен новий запит обробляється в окремому сеансі бази даних, що зменшує використання процесора.

4.4 Тестування розробки

Тестування програмного забезпечення — це серія процедур і дій, спрямованих на виявлення помилок, недоліків у дизайні та оцінку функціональності продукту з метою визначення його відповідності вимогам користувача та достатньої якості. Це має вирішальне значення для життєвого циклу програмного забезпечення, яке має на меті переконатися, що програмне забезпечення є надійним, ефективним і функціональним до того, як воно буде опубліковано або використано у виробництві. Тестування передбачає виконання різноманітних тестів, експериментів та аналіз результатів, щоб переконатися, що програма є ефективною та відповідає очікуванням.

Після розробки підсистеми було проведено тестування, яке включало комплексну оцінку функціональності всього блоку:

- Правильна робота кнопок, запис даних і оновлення інформації;
- налаштування даних;
- оцінка ефективності друкованих форм.
- перевірка правильності переходу між вкладками.

У процесі тестування помилок і недоліків у роботі агрегату виявлено не було.

Програма продемонструвала ефективність і ефективність у всіх останніх версіях браузерів для веб-сторінок, що підтверджує її послідовність і сумісність із різними платформами.

Це тестування продемонструвало, що функціональність програмного забезпечення є узгодженою та правильно відображається в різних версіях браузерів, це додало впевненості в його надійності та сумісності з різними середовищами для користувачів.

5. ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Технологічний аудит розробленого програмного забезпечення для автоматизації логістичних цілей у інформаційній системі «ПлазмІС»

Як вже було відзначено раніше, сьогодні у сучасному управлінні великими та середніми підприємствами очевидною є зростаюча роль інформаційних систем планування ресурсів. Сучасні вимоги до систем планування ресурсів підприємств передбачають поєднання функціональних та технологічних вимог, які включають використання реляційних баз даних, особливостей архітектури клієнт-серверних систем і можливостей їх інтеграції з існуючими програмними та апаратними платформами.

Так, підсистема логістики системи «ПлазмІС» реалізує такі функції, як реєстрація відвантажень, оформлення замовлень на матеріальні цінності від постачальників, планування надходження товарів на склад і багато іншого. Разом з тим існує потреба в подальшому розвитку та розширенні функціоналу модуля «Заявка на перевезення» для більш деталізованого керування логістичними процесами.

Тому метою виконаної нами магістерської кваліфікаційної роботи було розширення функціональних можливостей підсистеми логістики в системі планування ресурсів підприємства. Для досягнення цієї мети було розв'язано такі задачі: проведено аналіз підходів до автоматизації логістичних цілей в системі управління ресурсами підприємства; зроблено обґрунтування вибору інформаційної технології для реалізації логістичних цілей; розроблено програмне забезпечення для модулів «Волочіння зварювального дроту попереднє» в підсистемі виробництво; проведено тестування розроблених функціональних модулів тощо.

В результаті було запропоновано програмну реалізацію функціональних модулів, які призначені для автоматизації процесів у системі управління логістичними цілями інформаційної системи «ПлазмІС».

Для встановлення комерційного потенціалу розробленого нами програмного продукту було запрошено 3-х експертів. Визначення комерційного потенціалу розробленого програмного продукту було здійснено за критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання рівня комерційного потенціалу будь-якої розробки і їх бальна оцінка

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

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Запрошені експерти оцінили розроблений нами програмний продукт, який може використовуватися для полегшення управління керуванням виробництва процесами на підприємстві, таким чином (див. таблицю 5.2):

Таблиця 5.2 — Результати технологічного аудиту розробленого програмного продукту (за шкалою оцінювання «0»-«1»-«2»-«3»-«4»)

Критерії	Прізвище, ініціали експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали, що їх виставили експерти:		
1	4	3	3
2	4	3	4
3	3	4	3
4	4	4	4
5	3	3	4
6	4	4	3
7	3	3	4
8	4	3	3
9	4	4	3
10	3	3	4
11	4	4	3
12	3	3	4
Сума балів	СБ ₁ = 43	СБ ₂ = 41	СБ ₃ = 42
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{43 + 41 + 42}{3} = \frac{126}{3} = 42,00$		

Для встановлення комерційного потенціалу розробленого нами програмного продукту скористуємося рекомендаціями, які наведено в таблиці 5.3 [37].

Таблиця 5.3 – Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 42 бали, то це свідчить, що розроблений нами програмний продукт має рівень комерційного потенціалу, який вважається «високим».

Це пояснюється тим, що розроблене нами програмне забезпечення є інтегрованим та динамічним програмним продуктом, який значно полегшує управління логістичними процесами на підприємстві, забезпечуючи йому такі ключові переваги, як економія часу і ресурсів, підвищення ефективності логістичних операцій, точності і надійності їх виконання тощо.

5.2 Розрахунок витрат на розроблення програмного забезпечення

При розробленні програмного забезпечення були зроблені певні витрати.

Основна заробітна плата Z_o розробників, яка визначається за формулою:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн}, \quad (5.1)$$

де M — місячний посадовий оклад розробника, грн; прийmemo, що

$M = (8000 \dots 25000)$ грн/місяць;

T_p — число робочих днів в місяці; прийmemo $T_p = 21$ день;

t — число днів роботи розробників.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник	21500	1023,81	20 годин	≈ 3413
2. Магістрант-студент-виконавець	2000 (беремо 8000)	380,95	79	≈ 30095
3. Консультант	18550	883,33	1,5 години	≈ 221 (при 6-годинному робочому дні)
Загалом				$Z_o = 33729$ грн

Додаткова заробітна плата Z_d розробників розраховується як (10...12)% від величини їх основної заробітної плати, тобто:

$$Z_d = \alpha \cdot Z_o = (0,1...0,12) \cdot Z_o. \quad (5.2)$$

Приймемо, що $\alpha = 0,113$. Тоді для нашого випадку отримаємо:

$$Z_d = 0,113 \times 33729 = 3811,37 \approx 3811 \text{ грн.}$$

Нарахування на заробітну плату $НЗП_{зп}$ розробників (дослідників) розраховуються за формулою:

$$НЗП_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100}, \quad (5.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. $\beta = 22\%$. Тоді:

$$НЗН_{зп} = (33729 + 3811) \times 0,22 = 8258,8 \approx 8259 \text{ грн.}$$

Амортизація основних засобів A , які використовувались під час виконання цієї роботи:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ грн,} \quad (5.4)$$

де $Ц$ – загальна балансова вартість основних засобів, грн;

H_a – річна норма амортизаційних відрахувань, $H_a = (2,5...25)\%$;

T – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 5.5.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, обладнання тощо	76000	25	3,4 (при 80% використанні)	4306,66 ≈ 4307
2. Приміщення	38000	3,0	3,4 при 25% використанні	80,75 ≈ 81
Всього				A = 4388 грн

Витрати на матеріали М розраховуються за формулою:

$$M = \sum_1^n H_i \cdot C_i \cdot K_i - \sum_1^n B_i \cdot C_v \text{ грн.}, \quad (5.5)$$

де H_i — витрати матеріалу i -го найменування, кг;

C_i — вартість матеріалу i -го найменування;

K_i — коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; B_i — маса відходів матеріалу i -го найменування;

C_v — ціна відходів матеріалу i -го найменування; n — кількість видів матеріалів.

Витрати на комплектуючі К розраховуються за формулою:

$$K = \sum_1^n H_i \cdot C_i \cdot K_i \text{ грн.}, \quad (5.6)$$

де H_i — кількість комплектуючих i -го виду, шт.;

C_i — ціна комплектуючих i -го виду;

K_i — коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$;

n — кількість видів комплектуючих.

Під час виконання роботи загальні витрати на матеріали та комплектуючі склали приблизно 1850 грн.

Витрати на силову електроенергію V_e розраховуються за формулою:

$$V_e = \frac{V \cdot P \cdot \Phi \cdot K_p}{K_d}, \quad (5.7)$$

де V – вартість 1 кВт-год. електроенергії, в 2024 р. $V \approx 10,5$ грн/кВт;

P – установлена потужність обладнання, кВт; $P = 1,35$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

Прийmemo, що $\Phi = 335$ годин;

K_p – коефіцієнт використання потужності; $K_p < 1 = 0,84$.

K_d – коефіцієнт корисної дії, $K_d = 0,7$.

Тоді витрати на силову електроенергію будуть дорівнювати:

$$V_e = \frac{V \cdot P \cdot \Phi \cdot K_p}{K_d} = \frac{10,5 \cdot 1,35 \cdot 335 \cdot 0,84}{0,7} = 5698,35 \approx 5698$$

Інші витрати $V_{\text{інш}}$ можна прийняти як (50...300)% від основної заробітної плати розробників. Цей діапазон враховує різноманітні фактори, такі як витрати на офісне устаткування, програмне забезпечення, тренінги та інші ресурси, які можуть бути важливими для успішної розробки та впровадження програмного забезпечення, тобто:

$$V_{\text{інш}} = (0,5 \dots 3) \times Z_o. \quad (5.8)$$

Для нашого випадку отримаємо:

$$V_{\text{інш}} = 1,25 \times 33729 = 42161,25 \approx 42161 \text{ грн.}$$

Сума всіх попередніх статей витрат складає витрати на виконання роботи безпосередньо розробником-магістрантом – В.

$$B = 33729 + 3811 + 8259 + 4388 + 1850 + 5698 + 42161 = 99896$$

Загальні витрати на розроблення програмного забезпечення $B_{\text{заг}}$ розраховуються за формулою:

$$B_{\text{заг}} = \frac{B}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання цієї роботи. Можна прийняти, що, $\beta \approx 0,9$ [37], оскільки робота практично завершена.

$$\text{Тоді: } B_{\text{заг}} = \frac{99896}{0,9} = 110995,6 \text{ грн або приблизно 111 тисячі грн.}$$

Тобто прогнозовані загальні витрати на розробку програмного забезпечення становлять приблизно 111 тисячі грн.

5.3 Розрахунок економічного ефекту від можливої комерціалізації розробленого програмного забезпечення для автоматизації керування матеріалів для зварювальних робіт у інформаційній системі «ПлазмІС»

Економічний ефект від впровадження та можливої комерціалізації розробленого нами програмного забезпечення для автоматизації керування матеріалів для зварювальних робіт у інформаційній системі «ПлазмІС» пояснюється його значно кращими функціональними можливостями та ключовими перевагами, які можуть отримати підприємства, що будуть використовувати цей програмний продукт, а саме: як економією часу і ресурсів, підвищенням ефективності логістичних операцій, точності і надійності їх виконання тощо.

Тому нашу розробку можна реалізовувати на ринку дещо дорожче, ніж аналогічні за функціями розробки. Так, якщо подібні, але гірші за функціями програмні продукти у 2023-2024 роках коштували на ринку приблизно 80 тисяч

грн, то нашу розробку можна буде реалізовувати на ринку приблизно за 100 тисяч грн або на 20 тисяч грн дорожче.

Аналіз місткості ринку показує, що на сьогодні в Україні попит на подібний програмний продукт, особливо в умовах воєнного стану, може бути великим. Це, насамперед, численні перевізники, великі і середні підприємства тощо. Тому можна очікувати стрімке зростання попиту на нашу розробку принаймні протягом 3-х років після її впровадження (до появи нових, більш ефективних розробок).

Тобто, якщо наша розробка буде впроваджена з 1 січня 2025 року, то її результати будуть виявлятися протягом 2025-го, 2026-го та 2027-го років.

Прогноз зростання попиту на нашу розробку складає по роках:

- 2024 рік — базовий попит 50 шт;
- 2025 р. — приблизно +10 шт. до базового року;
- 2026 р. — +20 шт. до базового року;
- 2027 р. — +30 шт. до базового року.

Можливе збільшення чистого прибутку $\Delta\Pi_i$, що його може отримати потенційний інвестор від комерціалізації, тобто виведення нашої розробки на ринок, становитиме:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right), \quad (5.10)$$

де $\Delta\Pi_o$ — покращення основного якісного показника від впровадження результатів нашої розробки у цьому році. Для нашого випадку це є збільшення ціни реалізації нашої розробки $\Delta\Pi_o = (100 - 80) = + 20$ тисяч грн;

N — основний кількісний показник, який визначає обсяг діяльності у році до впровадження результатів розробки; $N = 50$ шт.;

ΔN — покращення основного кількісного показника від впровадження результатів розробки. Таке покращення становитиме: у 2025 році — + 10 шт., у 2026 році — +20 шт., у 2027 році — + 30 шт. (відносно базового 2024 року);

Π_0 – основний якісний показник (тобто ціна), який визначає обсяг діяльності у році після впровадження результатів розробки; $\Pi_0 = 100$ тисяч грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку $n = 3$;

λ – коефіцієнт, враховуючий сплату податку на додану вартість; $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = (0,2 \dots 0,5)$; візьмемо $\rho = 0,5$;

ν – ставка податку на прибуток. У 2024 і наступних роках $\nu = 18\%$.

Тоді можливе зростання чистого прибутку $\Delta \Pi_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2025 р.) складе:

$$\Delta \Pi_1 = [20 \cdot 50 + 100 \cdot 10] \cdot 0,833 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 683 \text{ тис. грн.}$$

Можливе зростання чистого прибутку $\Delta \Pi_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2026 р.) року складе:

$$\Delta \Pi_2 = [20 \cdot 50 + 100 \cdot 20] \cdot 0,833 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 1025 \text{ тис. грн.}$$

Можливе зростання чистого прибутку $\Delta \Pi_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2027 р.) року складе:

$$\Delta \Pi_3 = [20 \cdot 50 + 100 \cdot 30] \cdot 0,833 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 1367 \text{ тис. грн.}$$

Приведена вартість зростання всіх чистих прибутків від можливого впровадження нашої розробки становитиме:

$$\text{ПП} = \sum_1^t \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.11)$$

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

t – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку $t = 3$ роки;

τ – ставка дисконтування. Прийmemo $\tau = 0,10$ (10%);

t – період часу від моменту початку розроблення нашого програмного продукту до моменту отримання можливих чистих прибутків потенційним інвестором.

Тоді приведена вартість зростання всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор від комерціалізації нашої розробки, складе:

$$\text{ПП} = \frac{683}{(1+0,1)^2} + \frac{1025}{(1+0,1)^3} + \frac{1367}{(1+0,1)^4} \approx 564 + 770 + 964 = 2298 \text{ тисяч грн.}$$

Теперішня вартість інвестицій PV , що повинні бути вкладені для реалізації нашої розробки: $PV = (1,0 \dots 5,0) \times B_{\text{заг}}$.

Для нашого випадку $PV = (4,0 \dots 5,0) \times 111 = 4 \times 111 = 444$ тисяч грн.

Абсолютний ефект від можливих вкладених інвестицій $E_{\text{абс}}$.

$$E_{\text{абс}} = \text{ПП} - PV, \quad (5.12)$$

де ПП – приведена вартість збільшення всіх чистих прибутків для інвестора від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій $PV = 428$ тисяч грн.

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$E_{\text{абс}} = 2298 - 444 = 1854 \text{ тисяч грн.}$$

Оскільки $E_{\text{абс}} > 0$, то комерціалізація нашої розробки може бути доцільною.

Далі розрахуємо внутрішню дохідність $E_{\text{в}}$ вкладених інвестицій:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.13)$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 1854$ тис. грн;

PV – теперішня вартість початкових інвестицій $PV = 444$ тис. грн;

$T_{\text{ж}}$ – життєвий цикл розробки, роки.

$T_{\text{ж}} = 4$ років (2024-й, 2025-й, 2026-й, 2027-й роки)

Для нашого випадку отримаємо:

$$E_{\text{в}} = \sqrt[4]{1 + \frac{1854}{444}} - 1 = \sqrt[4]{1 + 4,1756} - 1 = \sqrt[4]{5,1756} - 1 = 1,50 - 1 = 0,50 = 50\%$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційному інвестору не вигідно буде займатися комерціалізацією нашої розробки.

Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau_{\text{мін}} = d + f, \quad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,09...0,14)$;

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

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,14 + 0,050 = 0,19 \text{ або } \tau_{\text{мін}} = 19\%.$$

Оскільки величина $E_B = 50\% > \tau_{\min} = 19\%$, то потенційний інвестор у принципі може бути зацікавлений у фінансуванні та комерціалізації розробленого нами програмного продукту.

Далі розраховуємо термін окупності коштів, вкладених у можливу комерціалізацію розробленого нами програмного продукту.

Термін окупності $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B}. \quad (5.15)$$

Для нашого випадку термін окупності $T_{ок}$ коштів становитиме:

$$T_{ок} = \frac{1}{0,5} = 2 \text{ років} < 3 \text{ років},$$

що свідчить про потенційну доцільність комерціалізації розробленого нами програмного забезпечення для автоматизації логістичних цілей у інформаційній системі «ПлазмІС».

Далі проведено моделювання залежності величини внутрішньої дохідності вкладених потенційним інвестором коштів в комерціалізацію розробленого нами програмного забезпечення для автоматизації керування метеріалів для зварювальних робіт у інформаційній системі «ПлазмІС» від рівня інфляції в країні.

Так, якщо рівень інфляції в країні зросте до 20%, то:

$$ПП = \frac{687}{(1+0,2)^2} + \frac{1025}{(1+0,2)^3} + \frac{1367}{(1+0,2)^4} \approx 477 + 593 + 659 = 1729 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$E_{абс} = 1729 - 444 = 1285 \text{ тисяч грн.}$$

Внутрішня дохідність E_B вкладених інвестицій становитиме:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1,$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 1285$ тисяч грн;

PV –теперішня вартість початкових інвестицій $PV = 444$ тисяч грн.

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{1285}{444}} - 1 = \sqrt[4]{1 + 2,8941} - 1 = \sqrt[4]{3,8941} - 1 = 1,40 - 1 = 0,40 = 40\%$$

Оскільки величина $E_B = 40\% > \tau_{\text{мін}} = 19\%$, то потенційний інвестор у принципі також може бути зацікавлений у фінансуванні та комерціалізації нашої розробки.

Прийнявши рівень інфляції у 30% отримаємо:

$$ПП = \frac{687}{(1+0,3)^2} + \frac{1025}{(1+0,3)^3} + \frac{1367}{(1+0,3)^4} \approx 406 + 466 + 478 = 1350 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки складе:

$$E_{\text{абс}} = 1350 - 444 = 906 \text{ тисяч грн.}$$

Внутрішня дохідність E_B вкладених інвестицій становитиме:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1,$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 906$ тисяч грн;

PV –теперішня вартість початкових інвестицій $PV = 444$ тисяч грн.

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{906}{444}} - 1 = \sqrt[4]{1 + 2,0405} - 1 = \sqrt[4]{3,0405} - 1 = 1,32 - 1 = 0,32 = 32\%$$

Оскільки величина $E_B = 32\% > \tau_{\min} = 19\%$, то потенційний інвестор у принципі може бути і зацікавлений у фінансуванні та комерціалізації нашої розробки, але остаточне рішення щодо цього питання буде прийматися при врахуванні інших обставин (наприклад, шляхом зниження рівня прийнятого ризику з $f = 30\%$ до меншої величини, або шляхом підняття ціни реалізації нашої розробки тощо).

Якщо рівень інфляції досягне 50%, то

$$ПП = \frac{687}{(1+0,5)^2} + \frac{1025}{(1+0,5)^3} + \frac{1367}{(1+0,5)^4} \approx 305 + 303 + 270 = 878 \text{ тисячі грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки складе:

$$E_{абс} = 878 - 444 = 434 \text{ тисяч грн.}$$

Внутрішня дохідність E_B вкладених інвестицій становитиме:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1,$$

де $E_{абс}$ – абсолютний ефект вкладених інвестицій; $E_{абс} = 434$ тисяч грн;

PV –теперішня вартість початкових інвестицій $PV = 444$ тисяч грн.

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{434}{444}} - 1 = \sqrt[4]{1 + 0,9774} - 1 = \sqrt[4]{1,9774} - 1 = 1,18 - 1 = 0,18 = 18\%$$

Оскільки величина $E_v = 18\% > \tau_{\min} = 19\%$, то потенційний інвестор у принципі буде незацікавлений у фінансуванні та комерціалізації нашої розробки, але, знов-таки остаточне рішення щодо цього питання буде прийматися при врахуванні інших обставин (наприклад, шляхом зниження рівня прийнятого ризику з $f = 30\%$ до меншої величини, або шляхом підняття ціни реалізації нашої розробки тощо).

Зроблені розрахунки у вигляді графіків наведено на рис. 5.1.

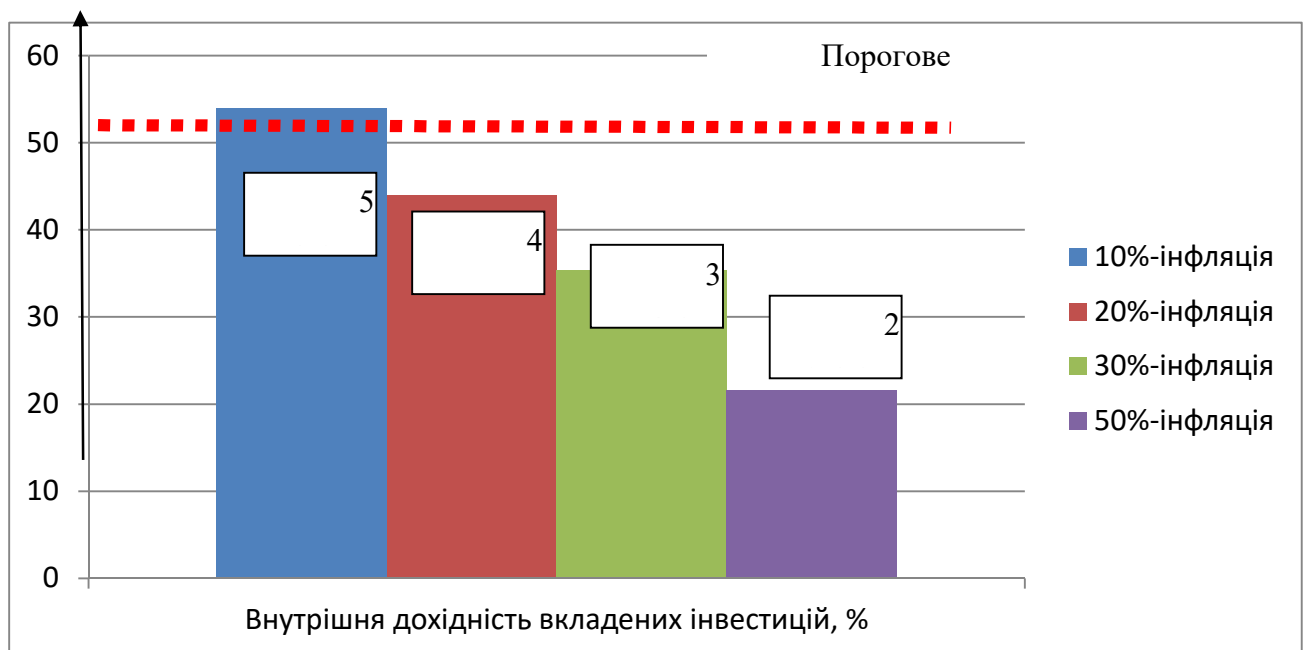


Рисунок 5.1 – Моделювання залежності величини внутрішньої дохідності потенційних інвестицій від рівня інфляції в країні при її значеннях у 10%, 20%, 30% та 50%

Аналіз графіка на рис 5.1 показує, що при рівні інфляції в 10% величина внутрішньої дохідності інвестицій становить $E_v = 50\%$, що значно більше порогового значення $\tau_{\min} = 19\%$ і тому комерціалізація нашої розробки може бути доцільною. При рівні інфляції в 20% величина внутрішньої дохідності

інвестицій, вкладених в комерціалізацію нашої розробки, становить $E_v = 40\%$, що також більше порогового значення $\tau_{\min} = 19\%$, і тому комерціалізація нашої розробки може бути для інвестора доцільною. І тільки при інфляції більше, ніж у 30% величина внутрішньої дохідності інвестицій, вкладених в комерціалізацію нашої розробки, становить менше порогового значення $\tau_{\min} = 19\%$. Тому за таких умов комерціалізація розробленого нами програмного продукту для автоматизації керування матеріалів для зварювальних робіт у інформаційній системі «ПлазмІС» може бути проблематичною і потребує проведення додаткових розрахунків.

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю 5.6.

Таблиця 5.6 – Результати виконаної економічної частини магістерської кваліфікаційної роботи

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку	Не більше 120 тис. грн	111 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розробки, тисяч грн	Не менше 1500 тисяч грн	1744 тисяч грн (при 10%-інфляції)	Виконано
3. Внутрішня дохідність інвестицій, %	не менше 42%	50% (при 10%-інфляції); 40% (при 20%-інфляції);	Досягнуто
4. Термін окупності інвестицій, роки	до 3-х років	1,92 років	Виконано

Таким чином, основні техніко-економічні показники розробленого нами програмного продукту для автоматизації керування матеріалів для зварювальних робіт у інформаційній системі «ПлазмІС», визначені у технічному завданні, виконані.

ВИСНОВКИ

Під час дослідження та аналізу існуючих інформаційних систем щодо планування ресурсів у першому розділі магістерської роботи було виявлено, що кожна з цих систем є дорогою, це викликає питання щодо практичності використання такого програмного забезпечення та потенціалу для розробки власного спеціалізовані модулі. Система «ПлазмІС» є прикладом створення нестандартного програмного продукту, спеціально адаптованого до потреб виробничих потужностей електродів сучасного типу «ПлазмаТек». Розглянуто логістичну підсистему даної системи, яка виконує значну кількість необхідних функцій, але потребує вдосконалення шляхом розробки додаткового функціонального модуля «Волочіння зварювального дроту попереднє».

У другій частині проекту було розпочато процес вибору технологій для розробки програмного забезпечення як для сервера, так і для клієнта. Вибрані технології були детально вивчені, а їхні можливості та унікальні властивості були задокументовані. Розглянуті аспекти включають, але не обмежуються цим, ефективність, масштабованість, продуктивність і легкість інтеграції в практичні проекти. Було проведено всебічне порівняння обраних технологій з існуючими аналогічними технологіями в галузі програмного забезпечення. Основною турботою було забезпечення високої ефективності, здатності розвиватися відповідно до вимог ринку, підвищення продуктивності та спрощення впровадження технології в реальні проекти.

Третя частина магістерської роботи присвячена створенню модуля «Заявка на перевезення» логістичної підсистеми. Виконано процедуру розробки UML-діаграм прикладів і послідовностей, визначено фундаментальні функції модуля та структуру реляційної бази даних, яка включає шість логічних сутностей, а також визначено основні компоненти та зв'язки для забезпечення ефективна робота модуля. Це платформа, на якій зберігаються та обробляються дані, необхідні для ефективного виконання обов'язків модуля. В результаті третій розділ не тільки документує процедуру позначення компонента «Волочіння

зварювального дроту попереднє», але й створює єдине джерело інформації, яке може бути використано для впровадження компонента в загальну логістичну систему та надання високоякісного обслуговування користувача.

У четвертому розділі виконано розробку функціональних модулів, пов'язаних з «Волочіння зварювального дроту попереднє». Ці модулі успішно інтегровані в логістичну складову системи «ПлазмІС», яка дозволяє значно підвищити її ефективність. Описана конструкція клієнт-серверної підсистеми характеризується використанням передових технологій, таких як HTML+CSS і JavaScript разом з бібліотекою jQuery. Іншим програмним застосунком, який використовується, є Oracle Application Express, цей програмний додаток постачається з ліцензією на сервер Oracle Database. Це гарантує надійність, масштабованість і чудову продуктивність розробленої системи.

Вирішальним кроком є успішне впровадження розробленої підсистеми; це пройшло через етапи впровадження та масштабного тестування. Це свідчить про надійність та ефективність розробленого програмного забезпечення при збереженні високого рівня якості та ефективності.

У п'ятому розділі дисертації були проведені розрахунки, які показали, що система при інфляції 10% принесе абсолютний дохід 1 870 000. Це свідчить про практичну доцільність і необхідність використання розробленого програмного продукту в автоматизації логістики в реальних умовах. Оцінка техніко-економічних показників, наведених у технічному завданні, свідчить про успішне виконання поставлених цілей. Система вже існує, вона є високоефективною та прибутковою, що робить її надійним інструментом автоматизації логістичних процесів, а також вигідним вирішенням бізнес-завдань.

Розроблене програмне забезпечення сприяло підвищенню ефективності логістичних процесів у системі «ПлазмІС». Успішне впровадження є свідченням його ефективності та здатності забезпечувати високий дохід, тому це важливо для підприємства в динамічному ринковому середовищі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

а

а(дата звернення 27.09.2024).

а(дата звернення 23.09.2024).

а(дата звернення 28.09.2024).

а(дата звернення 26.09.2024).

. С. Босак, Л. А. Савицька, О. В. Дудник. Комп'ютерна система керування виробництвом // [Електронний ресурс] / Л. А. Савицька // Матеріали LIV Всеукраїнської науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії, Вінниця, 24-27 березня 2025 р. – Електрон.

д

рикавський Є. Логістика та управління ланцюгами поставок Львівська політехніка . – Львів : ЛНТУ, 2011. – 124 с. – ISBN 938-961-121-341-2. (дата звернення 14.11.2024).

Вправління ланцюгами поставок. Конспект лекцій. І.С. Луценко 2022 Київ ім. Ігоря Рікорського – 142 с. – ДК №03056 (дата звернення 15.11.2024).

арі, Джон. «Програмування: Технічні аспекти» - видавництво «Професіонал», 2020. (дата звернення 26.09.2024).

а(дата звернення 26.09.2024).

а(дата звернення 27.09.2024).

а(дата звернення 27.09.2024).

а(дата звернення 27.09.2024).

юдігер Кречмер, Вольфганг Вейс. Developing SAP's R/3 Applications with SAP R/4 Версія 1998. (дата звернення 03.10.2024).

gano.com/oracle-blogs/what-is-oracle-transportation-management/ (дата звернення

о

міт, Емма. «Розробка веб сайтів: Від початку до кінця» - видавництво "Освіта", 2017. (дата звернення 26.09.2024).

м

е

н

endo UI – новітній UI-фреймворк від Telerik. Habr. URL:

t_overview.htm (дата звернення 13.11.2024).

(дата звернення 06.11.2024).

Ц (дата звернення 26.09.2024).

В

Бр. ТUTORIAL по Oracle Application Express. Обзор IDE. URL:

е

<https://www.techtarget.com/searchoracle/definition/PL/SQL> (дата звернення 13.11.2024).

д URL: <https://www.techtarget.com/searchoracle/definition/PL/SQL> (дата звернення

н

н URL: <https://medium.com/edureka/pl-sql-86e7652b2801> (дата звернення

я

(дата звернення 13.11.2024).

в (дата звернення 13.11.2024).

к (дата звернення 07.11.2024) на PHP. URL: <https://habr.com/ru/articles/122803/> (дата звернення 13.11.2024).

.

дата звернення 13.11.2024).

д (дата звернення 13.11.2024).

h

ислий В.М., Біловодська О.А., Олефіренко О.М., 69 Логістика: Теорія та

п

рдар Т.Г., Волошин Р.В. Основи логістики. : Навч. посіб. – 2009. – 115 с. (дата звернення 15.11.2024).

к тодичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. / Укладачі В.О. Козловський, О.Й. Лесько, В.В.Кавецький.

т Вінниця : ВНТУ, 2021. – 42

W

h

a

t

ДОДАТОК А**Технічне завдання**

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

_____ проф., д.т.н. О.Д. Азаров

«29» вересня 2024 року

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

Комп'ютерна система керування виробництвом матеріалів для зварювальних робіт
08-54.МКР.001.00.000 ТЗ

Керівник роботи: к.т.н., доц. каф. ОТ

Савицька Л.А

Виконав студент: гр. 1КІ-23м

_____ Босак І.С.

2024

істава для виконання магістерської кваліфікаційної роботи (МКР)

Розробка інтегрованої платформи для автоматизації виробничих процесів у сфері виготовлення зварювальних матеріалів є важливим кроком для підвищення ефективності підприємства. Така система дозволить оптимізувати розподіл ресурсів, автоматизувати контроль якості продукції та покращити взаємодію між підрозділами. Використання сучасних інформаційних технологій і локальної мережі сприятиме швидкому доступу до даних, оперативному обміну інформацією та мінімізації людського фактору, що позитивно вплине на

та МКР і призначення розробки

ета роботи — магістерської роботи є розширення функціональних можливостей підсистеми управління виробництвом зварювального дроту в системі управління ресурсами підприємства (ERP) «ПлазмІС».

Призначення розробки — розробка комп'ютерної системи керування виробництвом матеріалів для зварювальних робіт полягає у створенні інтегрованої платформи, яка забезпечує автоматизацію виробничих процесів, оперативний обмін інформацією між підрозділами та контроль виконання завдань у реальному часі. Система спрямована на оптимізацію використання ресурсів, зменшення кількості помилок і скорочення часу простоїв, що сприяє підвищенню ефективності роботи підприємства та якості зварювальних матеріалів. Вона також стане інструментом для вдосконалення управління і прийняття стратегічних рішень, забезпечуючи конкурентоспроможність на ринку..

іхідні дані для виконання МКР

система повинна функціонувати в середовищі глобальної комп'ютерної мережі, наприклад, в Інтернеті. Це забезпечує доступність для користувачів у будь-якому місці з підключенням до мережі. Обов'язковим є стабільний зв'язок для обміну даними між клієнтом і сервером.

програмний продукт повинен бути сумісним з ОС Windows, яка є однією з

в

и

найпоширеніших платформ для ПК. Забезпечення сумісності з різними версіями Windows (наприклад, Windows 10 або 11) є важливим для охоплення більшої кількості користувачів. підтримка усіх доступних браузерів;

еб-додаток має коректно працювати у всіх популярних браузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari. Це гарантує зручність використання незалежно від вибору користувача. фреймворк клієнта jquery.js.

для безпечної передачі даних між клієнтом і сервером використовується протокол HTTPS. Він забезпечує шифрування даних, що захищає від перехоплення інформації третіми особами. Вимоги до виконання МКР.

для розробки інтерактивного інтерфейсу користувача використовується jQuery.js, що дозволяє спрощувати роботу з DOM, обробку подій і AJAX-запити. Це забезпечує динамічність та покращує взаємодію користувача з веб-додатком.

інімальна вимога до апаратного забезпечення — наявність 2 GB оперативної пам'яті. Це дозволяє забезпечити стабільну роботу додатка, навіть за умови виконання декількох задач одночасно.

4 Вимоги до виконання МКР

Провести аналіз підходів до автоматизації логістичних цілей в системах

п

л

Обрати технології розробки модуля «Волічіння зварювального дроту

а

н

4.3 Розробити механізми автоматизації розподілу завдань та функціонал для внутрішніх комунікацій.

в

Вибрати технологію для реалізації підсистеми управління логістичними

а

ц

н

р

н

п

я

я

5 Етапи МКР та очікувані результати

м

р

и

е

б

с

р

у

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1.	Вибір, узгодження та затвердження теми МКР	19.09.2024	19.09.2024	виконано
2.	Дослідження предметної області	20.09.2024	25.09.2024	виконано
3.	Обґрунтування методів реалізації та контролю	26.09.2024	05.10.2024	виконано
4.	Розробка структури та функціональної схеми	06.10.2024	26.10.2024	виконано
5.	Розробка програмного забезпечення	11.11.2024	11.11.2024	виконано
6.	Тестування та перевірка припущень	12.11.2024	12.11.2024	виконано
7.	Підтвердження економічної частини	15.11.2024	16.11.2024	виконано
8.	Оформлення матеріалів до захисту МКР	17.11.2024	17.12.2024	виконано

Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами.

орядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

имоги до оформлювання та порядок виконання МКР

При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104–2006 «Єдина система конструкторської документації. Основні написи»;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія»;

— документи на які посилаються у вище вказаних.

Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21

ДОДАТОК Б

Ілюстративна частина

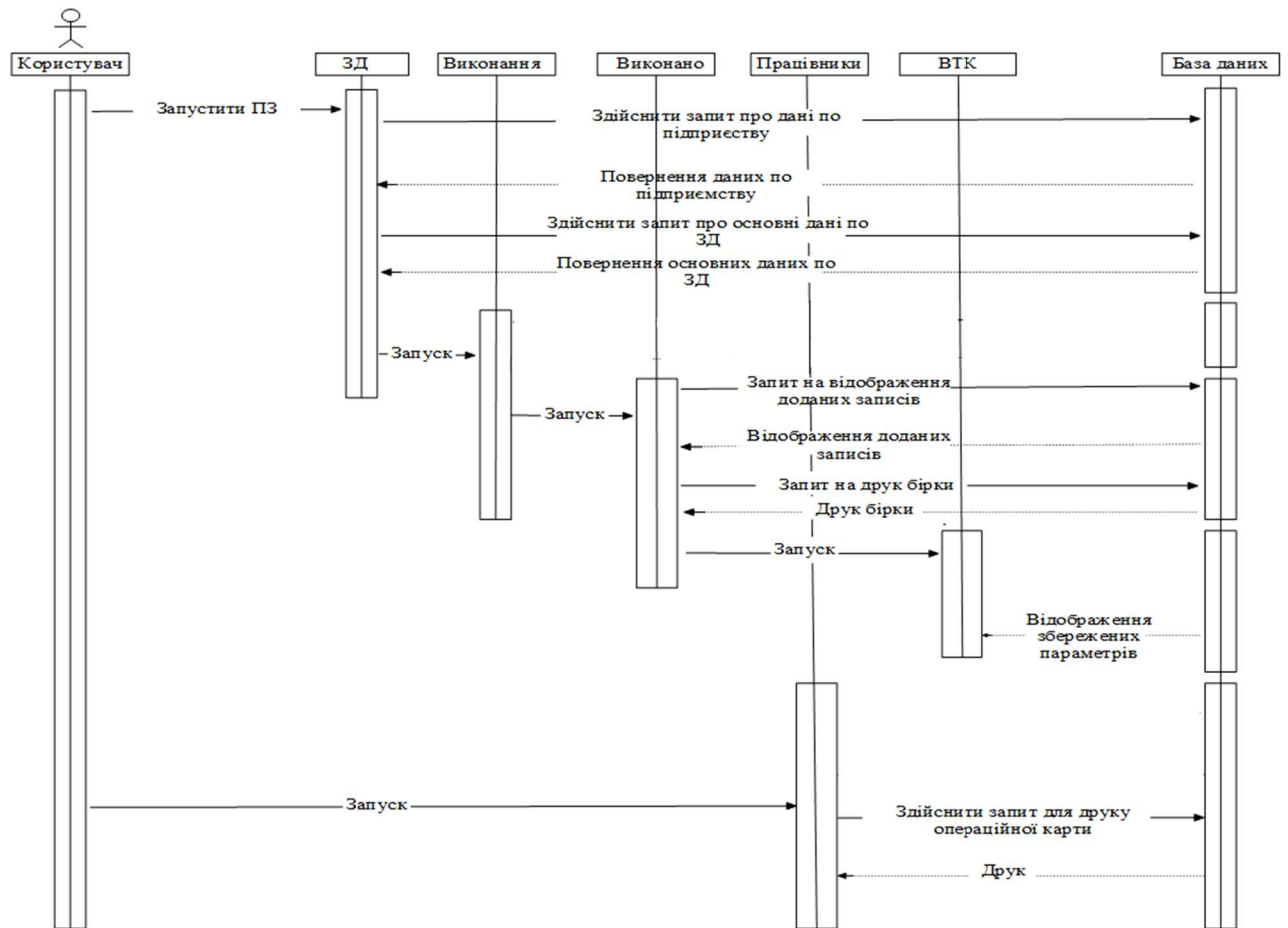


Рисунок Б.1 — Діаграма послідовностей функціонального модуля користувача

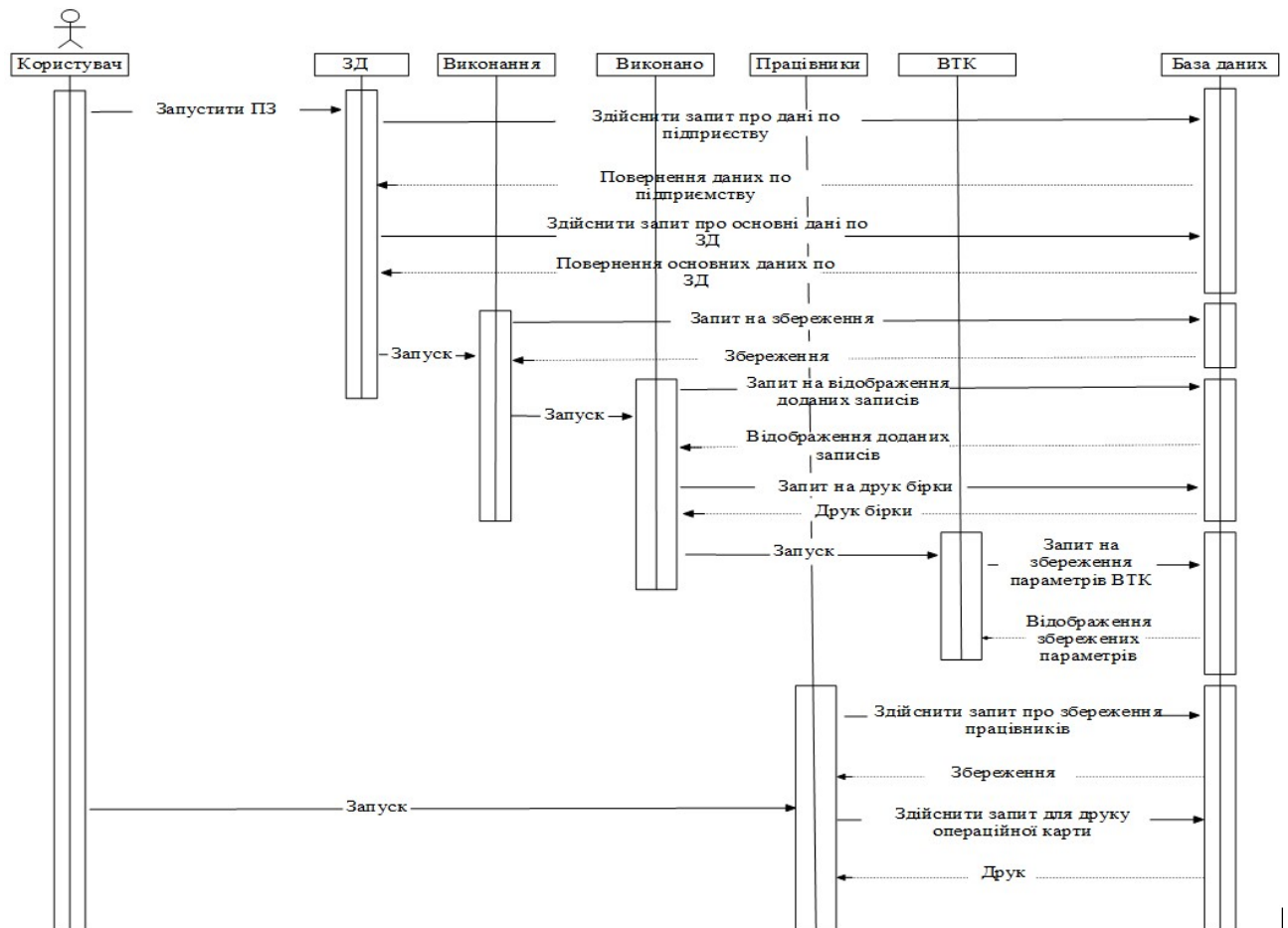
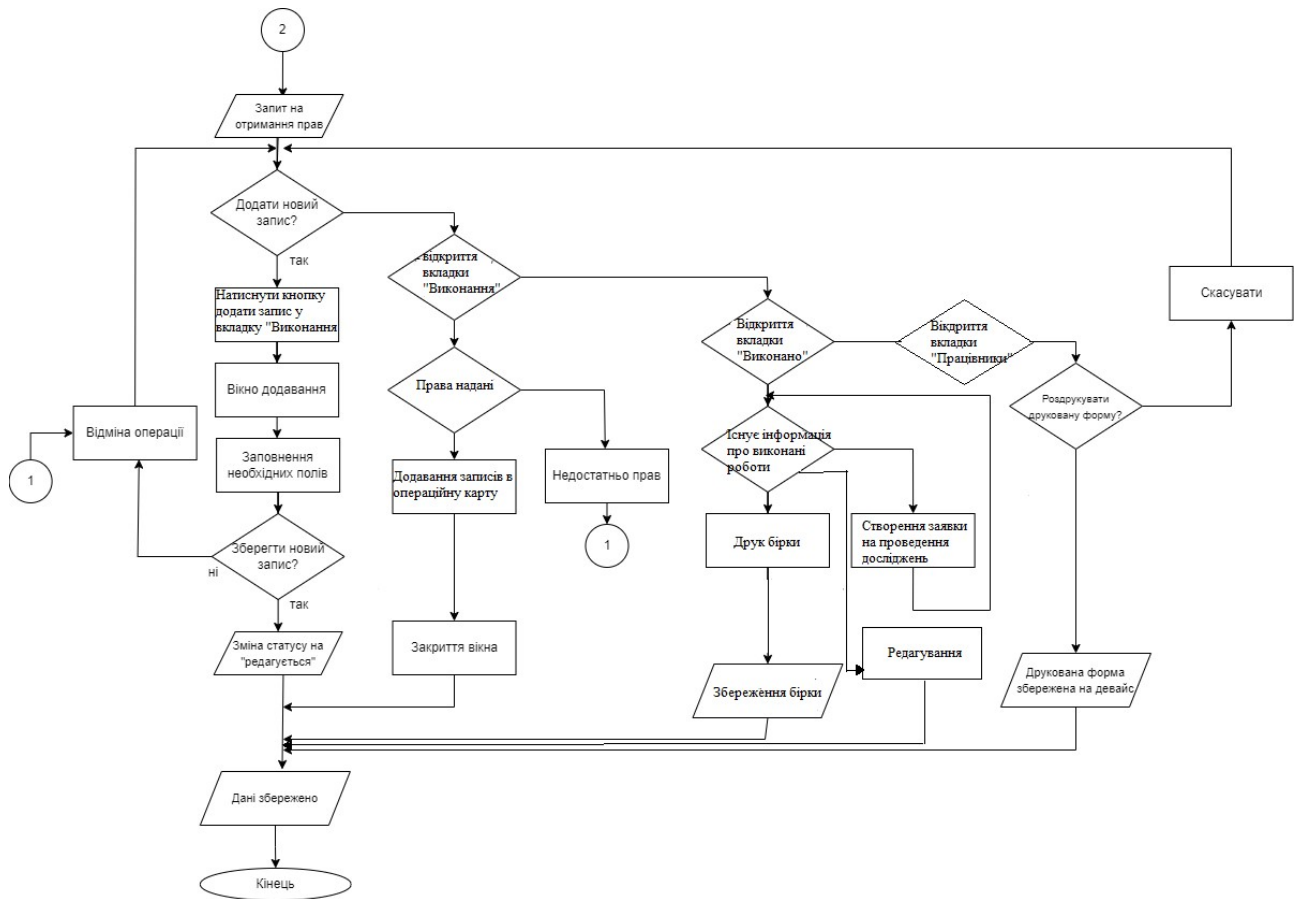


Рисунок Б.2 — Діаграма послідовностей функціонального модуля відповідальної особи



Б.3 — Блок-схема функціонування модуля

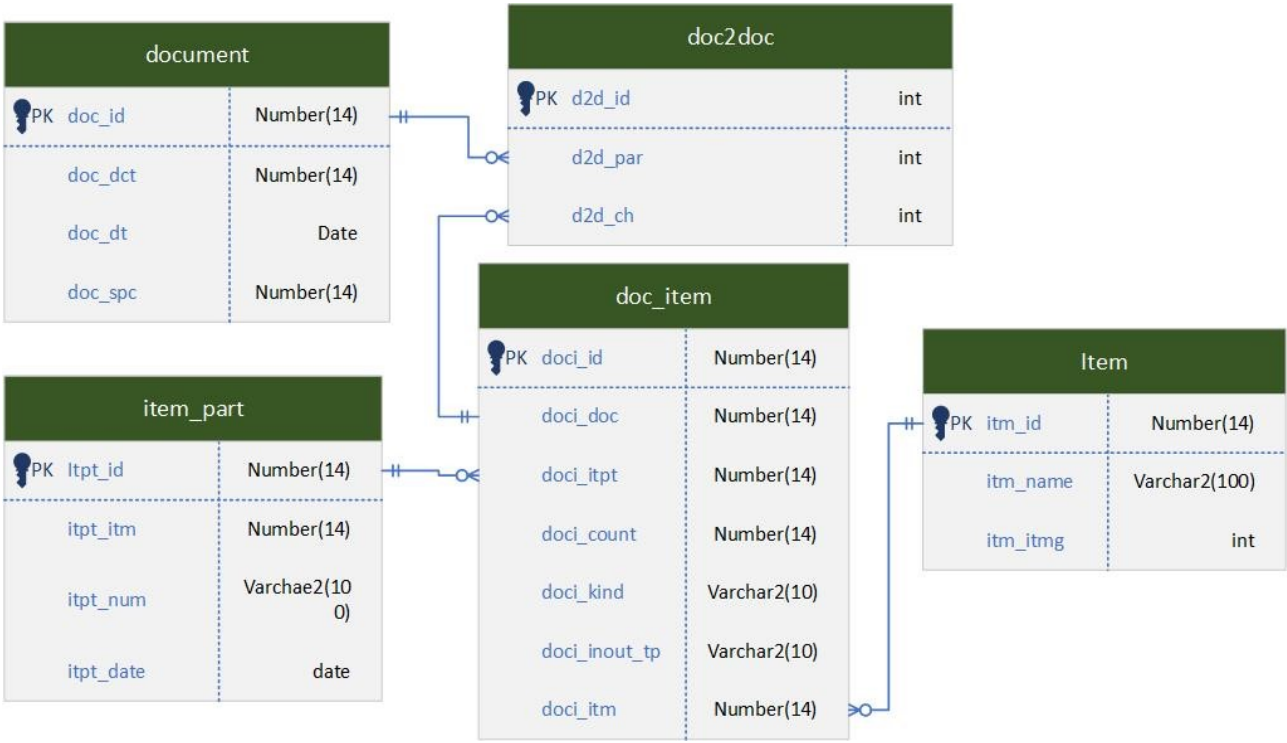


Рисунок Б.4 — Структура частини БД підсистеми для модуля

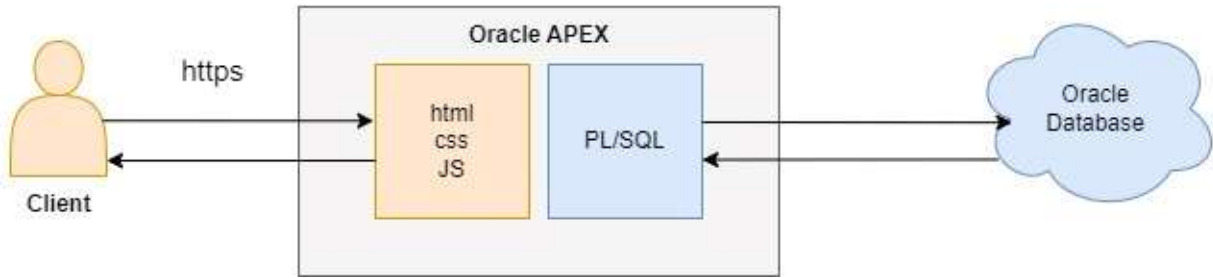


Рисунок Б.5 — Архітектура підсистеми логістики

Додаток В. (обов'язковий)

Лістинг програми

//лістинг процесу на відображення даних у вкладку «ЗД»

```

declare
  ms          varchar2(32767);
  l_cur       sys_refcursor;
  l_doc_id    dis_pltec.v_document.doc_id%type;
  l_dspc_spc  dis_pltec.v_doc_spc.dspc_spc%type;
  l_dct_code  dis_pltec.v_doc_type.dct_code%type;
begin
  l_doc_id := wwv_flow.g_x01;
  l_dspc_spc := wwv_flow.g_x02;
  wwv_flow.g_x01 := null;
  wwv_flow.g_x02 := null;

  select dct.dct_code
  into l_dct_code
  from dis_pltec.v_document doc
  inner join dis_pltec.v_doc_type dct on dct.dct_id = doc.doc_dct
  where doc.doc_id in (l_doc_id);

  if l_dct_code = 'TSK_CUW_WIND' then
  open l_cur for
  with
    consts_ as (
      select l_doc_id as doc_id from dual
    ),
    doci_ as (
      select itpt.itpt_id, itpt.itpt_itm as itm_id,
        doci_in.doci_docip, doci_in.doci_count_pre,
        doci_in.doci_count, doci_in.doci_comment, doci_in.doci_itm,
doci_in.doci_ius as doci_ius_in,
        doci_in.doci_rcpi as doci_rcpi_in, doci_in.doci_id, dct.dct_code, c.doc_id
      from consts_ c
      inner join dis_pltec.v_doc_item doci_in on doci_in.doci_doc = c.doc_id
      inner join dis_pltec.v_item_part itpt on itpt.itpt_id = doci_in.doci_itpt
      inner join dis_pltec.v_document doc on doc.doc_id = c.doc_id
      inner join dis_pltec.v_doc_type dct on dct.dct_id = doc.doc_dct
    ),
    itpt_ as (

```

```

select doci.itpt_id, doci.itm_id, doci.doci_docip,
       nvl(dis_pltec.pltec_common.GetPartInfo(p_itpt => doci.itpt_id, p_type => 3),
itm.itm_mrk) as mrk_id,
       nvl(dis_pltec.pltec_common.GetPartInfo(p_itpt => doci.itpt_id, p_type => 4),
dis_pltec.pltec_common.GetMrkName(p_mrk => itm.itm_mrk)) as mrk_name,
       dis_pltec.to_num(dis_pltec.item_dm.GetItemParamValue2Itm(p_itm =>
doci.itm_id, p_param_code => 'DIAMETER', p_param_tp => ('CONST')))) as diam,
       vm_itpt_par.itpt_num as itpt_num_par, replace(vm_itpt.itpt_name2_f,
vm_itpt_par.itpt_num || '/', ") as ctrl_code,
       doci.doci_count as pst_count_unt, doci.doci_comment,
       doci.doci_ius_in, doci_count_pre, doci_rcpi_in, doci_id, dct_code, doc_id
from doci_doci
       inner join dis_pltec.v_item itm on itm.itm_id = doci.itm_id
       inner join dis_pltec.vm_item_part vm_itpt_par on vm_itpt_par.itpt_id =
nvl(dis_pltec.pltec_common.GetPartInfo(p_itpt => doci.itpt_id, p_type => 1),
doci.itpt_id)
       inner join dis_pltec.vm_item_part vm_itpt on vm_itpt.itpt_id = doci.itpt_id

)
select doci_docip, mrk_name, diam, itpt_num_par, ctrl_code, pst_count_unt,
doci_comment, itpt_id, itm_id as itm_in,
       doci_ius_in, doci_count_pre, doci_rcpi_in,
dis_pltec.pltec_common.GetIusName2Date(p_ius => doci_ius_in, p_type => 2 ) as
itm_name_out, doci_id, dct_code, doc_id
from itpt_
order by doci_docip ;
apex_json_plt.write(l_cur);
else
open l_cur for

with
const_ as (
select l_doc_id as doc_id from dual
),
doci_ as (
select itpt.itpt_id, itpt.itpt_itm as itm_id, c.doc_id, doci_in.doci_docip,
doci_in.doci_id, dct.dct_code,
       doci_in.doci_count, doci_in.doci_comment, doci_in.doci_itm,
doci_in.doci_ius as doci_ius_in,
       doci_out.doci_itm as itm_out, doci_out.doci_ius as doci_ius_out,
doci_out.doci_rcprt, doci_out.doci_itpt as doci_itpt_out,
       doci_in.doci_rcpi as doci_rcpi_in, doci_out.doci_rcpi as doci_rcpi_out,
redci.redci_rpe as rpe_id
from const_ c

```

```

        inner join dis_pltec.v_doc_item doci_in on doci_in.doci_doc = c.doc_id and
        doci_in.doci_inout_tp = 'IN'
        inner join dis_pltec.v_doc_item doci_out on doci_out.doci_doc = c.doc_id
        and doci_out.doci_inout_tp = 'OUT' and doci_out.doci_docip = doci_in.doci_docip
        inner join dis_pltec.v_item_part itpt on itpt.itpt_id = doci_in.doci_itpt
        inner join dis_pltec.v_rcp2doci redci on redci.redci_doci = doci_out.doci_id
        inner join dis_pltec.v_document doc on doc.doc_id = c.doc_id
        inner join dis_pltec.v_doc_type dct on dct.dct_id = doc.doc_dct
    ),
    itpt_as (
        select doci.itpt_id, doci.itm_id, doci.doc_id, doci.doci_docip, doci.dct_code,
            nvl(dis_pltec.pltec_common.GetPartInfo(p_itpt => doci.itpt_id, p_type => 3),
            itm.itm_mrk) as mrk_id,
            nvl(dis_pltec.pltec_common.GetPartInfo(p_itpt => doci.itpt_id, p_type => 4),
            dis_pltec.pltec_common.GetMrkName(p_mrk => itm.itm_mrk)) as mrk_name,
            dis_pltec.to_num(dis_pltec.item_dm.GetItemParamValue2Itm(p_itm =>
            doci.itm_id, p_param_code => 'DIAMETER', p_param_tp => ('CONST'))) as diam,
            vm_itpt_par.itpt_num as itpt_num_par, replace(vm_itpt.itpt_name2_f,
            vm_itpt_par.itpt_num || '/', ") as ctrl_code,
            doci.doci_count as pst_count_unt, doci.doci_rcprt, doci.itm_out,
            dis_pltec.pltec_common.GetItmName2Date(p_itm => doci.itm_out, p_type
            => 1, p_prefix => 0) as itm_name_out,
            dis_pltec.pltec_common.GetItmName2Date(p_itm => doci.itm_out,
            p_type => 1/*, p_prefix => 0*/) as itm_name_out_full, doci.doci_comment,
            doci.doci_ius_in, doci.doci_ius_out, doci.doci_rcpi_in, doci.doci_rcpi_out,
            doci.rcpe_id, rcprt.rcprt_name as route, doci.doci_id,
            --
            doci.doci_itpt_out
        from doci_doci
            inner join dis_pltec.v_item itm on itm.itm_id = doci.itm_id
            inner join dis_pltec.vm_item_part vm_itpt_par on vm_itpt_par.itpt_id =
            nvl(dis_pltec.pltec_common.GetPartInfo(p_itpt => doci.itpt_id, p_type => 1),
            doci.itpt_id)
            inner join dis_pltec.vm_item_part vm_itpt on vm_itpt.itpt_id = doci.itpt_id
            inner join dis_pltec.v_recipe_route rcprt on rcprt.rcprt_id = doci.doci_rcprt
        )
        select doc_id, dct_code, doci_docip, mrk_name, diam, itpt_num_par, ctrl_code,
            pst_count_unt, doci_rcprt, itm_name_out, itm_name_out_full, route, doci_comment,
            itm_out, itpt_id, itm_id as itm_in,
            doci_ius_in, doci_ius_out, doci_rcpi_in, doci_rcpi_out, rcpe_id, doci_itpt_out,
            doci_id
        from itpt_
        order by doci_docip;
    apex_json_plt.write(l_cur);
end if;

```

```

exception
  when others then
    raise_application_error(-20000, dis_sys.dis_message_util.get_message(2,
'onDemand - GetTskCuwWireDarwDociDocId ',
    chr(10) || dbms_utility.format_error_stack ||
dbms_utility.format_error_backtrace))

e
n    //лістинг процесу на відображення даних у вкладку «Виконання»
d
declare
  ms          varchar2(32767);
  l_cur        sys_refcursor;
  l_doci_id     dis_pltec.v_doc_item.doci_id%type;
  l_doc_id      dis_pltec.v_document.doc_id%type;
  l_dspc_spc    dis_pltec.v_doc_spc.dspc_spc%type;
begin
  l_doc_id := wwv_flow.g_x01;
  l_doci_id := wwv_flow.g_x02;
  l_dspc_spc := wwv_flow.g_x03;

  wwv_flow.g_x01 := null;
  wwv_flow.g_x02 := null;
  wwv_flow.g_x03 := null;

  dis_pltec.test_ibosak.GetTskRodCutItmName_WEB(
    p_doc_id => l_doc_id,
    p_doci_id => l_doci_id,
    p_dspc_spc => l_dspc_spc
  );

exception
  when others then
    raise_application_error(-20000, dis_sys.dis_message_util.get_message(2,
'onDemand - GetTskRodCutItmNameDiam ',
    chr(10) || dbms_utility.format_error_stack || dbms_utility.format_error_backtrace

//лістинг процесу на відображення даних у вкладку «Виконано»
begin
  dis_pltec.test_ibosak.GetDociCuwWircop_WEB(
    p_doc_id => wwv_flow.g_x01
  );
  wwv_flow.g_x01 := null;

end;
```

```

// Вкладка 3Д
function GetTskRodCutInfo4DocId(doc_id, tsk_view = true) {
    const that = this

    /* apex.server.process("GetDocSpc4TskCnt", {
        x01: doc_id,
        x02: +$v('P0_SPC')
    }, {
        success: function (pData) {
            if (pData) {
                spc_cnt = pData[0].cnt;
            }
        }
    });

    if (tsk_view) {
        $('#.cards_head').append(`
            <div class='rod-cut-tsk-toolbar' style="display: flex; justify-content: space-between;">
                <div class="dspc_name_recp">
                    <input id="dspc_name_recp" class="k-select" font-size:12px; padding:0; width: max-content;" min="0">
                </div>
                <div style='font-size:16px;' id='doc_num'></div>
            </div>
        `);*/
        $('#dspc_name_recp').kendoDropDownList({
            optionLabel: {
                value: null,

```

```

        spc_namef: " "
    },
    autoBind: true,
    dataTextField: "spc_namef",
    dataValueField: "value",
    height: 'auto',
    change: function (e) {
        $('<div>.cards').children('<div>').remove()
        //GetTskRodCutInfo4DocId(doc_id, false);
    },
    dataSource: {
        transport: {
            read: function (options) {
                apex.server.process("GetDocSpc4OC", { //GetDocSpc4Tsk
                    x01: doc_id,
                    //x02: +$v('P0_SPC')
                }, {
                    success: function (pData) {
                        options.success(pData);
                    }
                });
            },
        },
    },
});

/* $('<div>.rod-cut-tsk-
toolbar').find("#dspc_name_rec<div>p").data('kendoDropDownList').dataSource.read().done((result) => {

```

```

        $('#rod-cut-tsk-
toolbar').find("#dspc_name_recp").data('kendoDropDownList').value((spc_cnt > 0) ?
+$v('P0_SPC') : "");

        if (spc_cnt > 0) $('#rod-cut-tsk-
toolbar').find("#dspc_name_recp").data('kendoDropDownList').enable(false);

        $('#rod-cut-tsk-
toolbar').find("#dspc_name_recp").data('kendoDropDownList').trigger("change");

    });
}*/

```

```

apex.server.process('GetTskCuwWireDrawtInfo4DocId', {
    x01: doc_id,
}, {
    dataType: 'json',
    async: false,
    /*success: function (data) {
        if (data) {
            $('#cards_head').append(`
                <div style="display:grid; grid-template-columns:auto;">
                    <div style='font-size:20px;' id='doc_num'></div>
                </div>
            `);

            $("#doc_num").html(data[0].doc_num + ' ' + lang('Від') + ' ' +
data[0].doc_dt + ');');
        }
    },*/
    success: function (data) {
        if (data && data.length) {
            $('#cards_head').append(`
                <div style="display:grid; grid-template-columns:auto;">

```



```

    ${data[0].doc_cnt > 1 ? `<input id = "doc_list" style="width:
175px;">` : `<div style='font-size:20px;' id='doc_num'></div>`}
    <div style='font-size:20px;' id='doc_num'></div>
</div>

`);
try {
    $("#doc_tabstrip").attr('doc_st', data[0].doc_st);
} catch (error) {
    console.error(error);
}
if (data[0].doc_cnt > 1) { // елемент переходу між документами однієї
людини
    $("#doc_list").kendoDropDownList({
        dataTextField: 'name',
        dataValueField: 'value',
        template: '#=name#' + ' ' + '#=equip_name#',
        autoWidth: window.innerWidth > 800 ? true : false,
        height: 1000,
        value: doc_id,
        dataSource: {
            transport: {
                read: function(options){
                    apex.server.process('GetDoList4Doc', {
                        x01: doc_id
                    }, {
                        dataType: 'json',
                        async: true,
                        success: function(data){

```

```

        if (data && data.err) {
            if (data.err_msg) {
                noty({ text: data.err_msg, type: 'ERROR', layout:
'topRight'}));

                } else {
                    noty({ text: lang("Сталася помилка"), type:
'ERROR', layout: 'topRight'}));
                    console.error(data);
                }
            } else if (data.length) {
                options.success(data);
            } else {
                noty({ text: lang("Сталася помилка"), type: 'ERROR',
layout: 'topRight'}));
                console.error(data);
                options.success([]);
            }
        }
    });
},
},
},
    change: (e) => {
        window.location.href = 'f?p=130:2090:' + $('#pInstance').val() +
'::NO::P2090_DOC_ID:' + e.sender.value();
    }
}).data('kendoDropDownList');
} else {
    $('#doc_num').html(data[0].doc_num + '' + lang('Від') + '' +
data[0].doc_dt + ');');

```

```

    }
  } else {
    noty({ text: lang("Дані не знайдено"), type: 'ERROR', layout: 'topRight'
  });
  }
},
error: function (e) {
  if (e.readyState === 4) {
    noty({ text: e.responseText, type: 'WARNING', layout: 'topRight' });
  } else if (e.readyState === 0) {
    noty({ text: lang("Проблема з інтернетом"), type: 'WARNING', layout:
'topRight' });
  } else {
    noty({ text: lang("Виникла помилка"), type: 'WARNING', layout:
'topRight' });
  }
}
});

apex.server.process('GetTskCuwWireDarwDociDocId', {
  x01: doc_id/,
  x02: $(''.rod-cut-tsk-
toolbar').find("#dspc_name_recp").data('kendoDropDownList').value()*/
  // x03: spc_cnt
}, {
  dataType: 'json',
  async: false,
  success: function (data) {
    for (let i = 0; i < data.length; i++) {
      that.doci_docip = data[i].doci_docip;
    }
  }
});

```

```

        CreateTskRodCutCard(data[i]);
    }
}
});
let zindex = 10;
$("div.card").off('click').on('click', function (e, o) {
    e.preventDefault();
    console.log('here');
    let isShowing = false;
    if ($(this).hasClass("show")) {
        isShowing = true;
    }
    if ($("div.cards").hasClass("showing")) {
        $("div.card.show").removeClass("show");
        if (isShowing) {
            $("div.cards").removeClass("showing");
        } else {
            $(this).css({ zIndex: zindex }).addClass("show");
        }
        zindex++;
    } else {
        $("div.cards").addClass("showing");
        $(this).css({ zIndex: zindex }).addClass("show");
        zindex++;
    }
    console.log($(this).attr('id').replace('doci_', ''), $(this).attr('itpt_id'));
    $("#doc_tabstrip").attr('doci_id', $(this).attr('id').replace('doci_', ''));
    $("#doc_tabstrip").attr('itpt_id', $(this).attr('itpt_id'));
    $("#doc_tabstrip").attr('itm_ist_weight', $(this).attr('itm_ist_weight'));

```

```

    });
}
//вкладка Виконання
function GetTskRodCutExc(params, tsk_view = true) {
    $('.save_card').children('div').remove()
    let { doc_id } = params;
    let doci_id = $("#doc_tabstrip").attr('doci_id')
    apex.server.process("GetCntSpc4Exc", {
        x01: doc_id,
        x02: +$v('P0_SPC')
    }, {
        success: function (pData) {
            if (pData) {
                spc_cnt = pData[0].cnt;
            }
        }
    });

    if (tsk_view) {
        $('.cards_crimp_num').append(`
            <div class='rod-cut-tsk-toolbar' style="display: flex; justify-content: space-between;">
                <div class="dspc_name_">
                    <input id="dspc_name_" class="k-select" font-size:12px; padding:0;
width: max-content;" min="0">
                </div>
                <div style='font-size:16px;' id='doc_num'></div>
            </div>
`);
    }
}

```

```

$(`#dspc_name_`).kendoDropDownList({
  optionLabel: {
    value: null,
    spc_namef: " "
  },
  autoBind: true,
  dataTextField: "spc_namef",
  dataValueField: "value",
  height: 'auto',
  change: function (e) {
    $('#.save_card').children('div').remove()
    GetTskRodCutExc(params, false);
  },
  dataSource: {
    transport: {
      read: function (options) {
        apex.server.process("GetDocSpc4Tsk", { //GetDocSpc4Tsk
          x01: doc_id,
          //x02: +$v('P0_SPC')
        }, {
          success: function (pData) {
            options.success(pData);
          }
        });
      },
    },
  },
});

```

```

    $('rod-cut-tsk-
toolbar').find("#dspc_name_").data('kendoDropDownList').dataSource.read().done((r
esult) => {

    $('rod-cut-tsk-
toolbar').find("#dspc_name_").data('kendoDropDownList').value((spc_cnt > 0) ?
+$v('P0_SPC') : "");

    if (spc_cnt > 0) $('rod-cut-tsk-
toolbar').find("#dspc_name_").data('kendoDropDownList').enable(false);

    $('rod-cut-tsk-
toolbar').find("#dspc_name_").data('kendoDropDownList').trigger("change");

    });
}

```

```

let dspc_spc_namef = "";
apex.server.process('GetTskRodCutItmName', {
    x01: doc_id,
    x02: doci_id,
    x03: $('rod-cut-tsk-
toolbar').find("#dspc_name_").data('kendoDropDownList').value()
}, {
    success: function (pData) {
        console.log(pData);
        $('save_card').children('div').remove()
        for (let i = 0; i < pData.length; i++) {
            if (dspc_spc_namef === pData[i].dspc_spc_namef) {
                null;
            } else {
                dspc_spc_namef = pData[i].dspc_spc_namef;
                $('save_card').append(`<div style="display:grid; font-size: 18px; grid-
template-columns: 1fr; background-color: gainsboro;">
                    <div>${dspc_spc_namef}</div>

```

```

        </div>`);
    }
    SaveTskRodCut4DocId(pData[i], doc_id);
}
let zIndex = 10;
$("#div.card_crimp_fact").off('click').on('click', function (e, o) {
    e.preventDefault();
    console.log('here');
    let isShowing = false;
    if ($(this).hasClass("show")) {
        isShowing = true
    }
    if ($("#div.save_card").hasClass("showing")) {
        $("#div.card_crimp_fact.show").removeClass("show");
        if (isShowing) {
            $("#div.save_card").removeClass("showing");
        } else {
            $(this).css({ zIndex: zIndex }).addClass("show");
        }
        zIndex++;
    } else {
        $("#doc_tabstrip").attr('equip_id', $(this).attr('equip_id'));
        $("#doc_tabstrip").attr('docid_id', $(this).attr('docid_id'));
        $("#doc_tabstrip").attr('docid_obi_ts', $(this).attr('docid_obi_ts'));
        $("#div.save_card").addClass("showing");
        $(this).css({ zIndex: zIndex }).addClass("show");
        zIndex++;
    }
    console.log($(this).attr('equip_id'));

```



```

    $("#doc_tabstrip").attr('equip_id', $(this).attr('equip_id'));
    $("#doc_tabstrip").attr('docid_id', $(this).attr('docid_id'));
    $("#doc_tabstrip").attr('docid_obi_ts', $(this).attr('docid_obi_ts'));
  });
},
error: function (e) {
  if (e.readyState === 4) {
    noty({ text: e.responseText, type: 'WARNING', layout: 'topRight' });
  } else if (e.readyState === 0) {
    noty({ text: lang('Проблема з інтернетом'), type: 'WARNING', layout:
'topRight' });
  } else {
    noty({ text: lang('Виникла помилка'), type: 'WARNING', layout:
'topRight' });
  }
}
});
}
//вкладка Виконано
function rodCutToolbar(doc_id, tsk_view = true) {
  apex.server.process("GetDocSpc4TskCnt", {
    x01: doc_id,
    x02: +$v('P0_SPC')
  }, {
    async: false,
    success: function (pData) {
      if (pData) {
        spc_cnt = pData[0].cnt;
      }
    }
  }

```

```

    }
  });
  if (tsk_view) {
    $('#cards_executed_toolbar').append(
      <div class='rod-cut-toolbar' style="display: flex">
        <div class="dspc_spc">
          <input id="dspc_spc" class="k-select" font-size:12px; padding:0; width:
max-content;" min="0">
        </div>
      </div>
    );
    $('#dspc_spc').kendoDropDownList({
      optionLabel: {
        value: null,
        spc_namef: " "
      },
      autoBind: true,
      dataTextField: "spc_namef",
      dataValueField: "value",
      height: 'auto',
      change: function (e) {
        apex.server.process('GetD2DOperCard', {
          x01: doc_id,
          x02: $('#rod-cut-
toolbar').find("#dspc_spc").data('kendoDropDownList').value()
        }, {
          success: function (pData) {
            for (let i = 0; i < pData.length; i++) {
              info = pData[i];
            }
          }
        });
      }
    });
  }
}

```

```

        let dopc_id = info.doc_id
        $('<div>.cards_crimp_fact').children('div').remove()
        GetTskRodCutFactInfo4DocId(dopc_id, false);
    }
}

});
},
dataSource: {
    transport: {
        read: function (options) {
            apex.server.process("GetDocSpc4OC", { //GetDocSpc4Tsk
                x01: doc_id,
                //x02: +$v('P0_SPC')
            }, {
                async: false,
                success: function (pData) {
                    options.success(pData);
                }
            });
        },
    },
},
});

// процес зберігання
declare
exWrong exception;

ms          varchar2(32767);

l_doci_id    dis_pltec.v_doc_item.doci_id%type;

```

```

l_doc_id      dis_pltec.v_document.doc_id%type;
l_docid_itpt  dis_pltec.v_doc_item_dtl.docid_itpt%type;
l_docid_count number;
l_equip_id    number;
l_docid_id    dis_pltec.v_doc_item_dtl.docid_id%type;
l_docid_obi_ts dis_pltec.v_doc_item_dtl.docid_obi_ts%type;
l_docid_docip dis_pltec.v_doc_item_dtl.docid_docip%type;
l_dspc_id     dis_pltec.v_doc_spc.dspc_id%type;
l_dspc_obi_ts dis_pltec.v_doc_spc.dspc_id%type;
l_dspc_spc     dis_pltec.v_doc_spc.dspc_id%type;
begin
  l_doci_id      := wwv_flow.g_x01; l_doc_id      := wwv_flow.g_x02;
  l_docid_count  := dis_pltec.to_num(wwv_flow.g_x03);
  l_docid_itpt   := wwv_flow.g_x04;
  l_equip_id     := wwv_flow.g_x05; l_docid_docip   := wwv_flow.g_x06;
  wwv_flow.g_x01 := null;
  wwv_flow.g_x02 := null;
  wwv_flow.g_x03 := null;
  wwv_flow.g_x04 := null;
  wwv_flow.g_x05 := null;
  wwv_flow.g_x06 := null;
  dis_pltec.item_dm.Ins_Doc_Item_Dtl(
    p_docid_id => l_docid_id,
    p_docid_obi_ts => l_docid_obi_ts,
    p_docid_doci => l_doci_id,
    p_docid_doc => l_doc_id,
    p_docid_count => l_docid_count,
    p_docid_itpt => l_docid_itpt,
    p_docid_comment => null,

```

```

p_docid_dpr => :P1000_FIRM,
p_docid_tp => null,
p_docid_itm_equip => l_equip_id,
p_docid_docip => l_docid_docip
);
l_dspc_spc := :P0_SPC;
dis_pltec.opercard_dm.Ins_Doc_Spc(
    p_dspc_id    => l_dspc_id,
    p_dspc_obi_ts => l_dspc_obi_ts,
    p_dspc_doc    => null,
    p_dspc_doci   => null,
    p_dspc_spc    => l_dspc_spc,
    p_dspc_jbp    => null,
    p_dspc_docid  => l_docid_id,
    p_dspc_rel    => 'EXEC'
);
http.p({'err':0, "doci_id":"" || l_doci_id || ""});
exception
/* when exWrong then

    rollback;

    http.p({'err':1, "err_text":"" || apex_lang.lang('Не знайдено одиниці зберігання в
топах') || ""});*/

when others then

    rollback;

    ms := 'onDemand SaveRodCut ' || dbms_utility.format_error_stack ||
dbms_utility.format_error_backtrace;

    http.p({'err':1, "err_text":"" ||
replace(replace(replace(htf.escape_sc(ms),chr(10)||chr(13),'<br>'),chr(10),'<br>'),chr(
13),'<br>') || ""});

e
n
d

```

Керівник роботи _____ Савицька Л.А.

(підпис) (прізвище, ініціали)