

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.457

До захисту допущено
Завідувач кафедри ММСА

Оксана ТИМОЩУК

«___» _____ 2022 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 «Системний аналіз»
на тему: «Методи і механізми управління і підтримки комплексом управління
складом»

Виконала:

студентка 2 курсу, групи КА-13мп
Торліна Надія Михайлівна

Керівник: професор кафедри ММСА,
д.т.н., проф. Мухін В.Є.

Рецензент: декан факультету
інформатики та обчислювальної техніки
КПІ ім. Ігоря Сікорського
д.т.н., Корнага Я.І.

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань

Студент (підпис): _____

Київ
2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність (ОПП) — 124 «Системний аналіз» («Системний аналіз
фінансового ринку»)

ЗАТВЕРДЖУЮ
Завідувач кафедри ММСА
Оксана ТИМОЩУК
«___» _____ 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студентці Торліній Надії Михайлівні

1. Тема дисертації: «Методи і механізми управління і підтримки комплексом управління складом», науковий керівник дисертації Мухін Вадим Євгенович, д.т.н., професор, затверджені наказом по університету від «03» листопада 2022 р. № 4046-с

2. Термін подання студентом дисертації: 12.12.2022 р. _____

3. Об'єкт дослідження: методологія впровадження і підтримки комплексу управління складом

4. Предмет дослідження: методи і механізми управління і підтримки комплексом управління складом

5. Перелік завдань, які потрібно розробити:

1) провести огляд існуючої методології впровадження комплексу управління складом;

2) провести огляд існуючої методології підтримки комплексу управління складом;

3) розробити програмний продукт для мінімізації часу та зусиль витрачених на підтримку і впровадження комплексу управління складом на основі методу з використанням механізмів управління;

4) продемонструвати результати роботи програмного продукту;

5) проаналізувати ефективність роботи програмного продукту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- 1) функціональні блок-схеми (рис.);
- 2) інформаційні зображення (рис.);
- 3) приклади роботи програми (рис.);
- 4) таблиці у розділі стартап-проекту, таблиці з даними та порівняльні таблиці результатів.

7. Орієнтовний перелік публікацій:

1. Торліна Н.М., Мухін В.Є. Методи і механізми управління і підтримки комплексом управління складом // 1-а Всеукраїнська конференція «Системний аналіз та інформатика», м. Київ, 22-29 листопада, 2022 року. С. 251-256.

8. Дата видачі завдання: 01.09.2022_____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Затвердження теми магістерської дисертації (МД)	01.09.2022-07.09.2022	виконано
2	Аналіз актуальності задач стосовно тематики дослідження	01.09.2022-19.09.2022	виконано
3	Формування задач дослідження	01.09.2022-19.09.2022	виконано
4	Аналіз існуючих рішень стосовно тематики дослідження	01.09.2022-19.09.2022	виконано
	Завершення роботи над першим варіантом основної частини МД	08.09.2022-19.09.2022	виконано
5	Розробка програмного продукту на основі метода з технічним механізмом, оформлення розділу зі стартап проекту	15.09.2022-19.10.2022	виконано
6	Оформлення звіту з переддипломної практики	20.09.2022-26.10.2022	виконано
7	Оформлення пояснювальної записки	01.12.2022-10.12.2022	виконано
8	Підготовка презентації до захисту	01.12.2022-10.12.2022	виконано

Студент

Надія ТОРЛІНА

Науковий керівник дисертації

Вадим МУХІН

РЕФЕРАТ

Магістерська дисертація 103 с., 13 рис., 47 табл., 2 додатки, 27 джерел

АВТОМАТИЗАЦІЯ СКЛАДСЬКИХ ПРОЦЕСІВ, СИСТЕМА УПРАВЛІННЯ СКЛАДОМ, МЕХАНІЗМ УПРАВЛІННЯ, ВПРОВАДЖЕННЯ, ПІДТРИМКА, КОМПЛЕКС УПРАВЛІННЯ СКЛАДОМ.

Метою дослідження є скорочення часу, зусиль і витрат на управління і підтримку комплексу управління складом на основі методу з використанням технічних механізмів управління.

Об'єктом даної роботи є дослідження та аналіз методології управління та підтримки системи управління складом, а також розробка програмного продукту, для мінімізації часу та зусиль витрачених на підтримку і впровадження комплексу управління складом на основі методу з використанням механізмів управління.

Предметом дослідження є методи і механізми управління і підтримки комплексом управління складом.

Актуальність теми полягає в тому, що автоматизація бізнес-процесів, а також процесів складської логістики, грає суттєву роль у економічному розвитку компанії. Це в свою чергу підвищує ефективність функціонування підприємства або компанії в будь-яких умовах, що є однією з найважливіших задач, що сприяє розвитку ринкових відносин в умовах конкуренції.

У результаті виконання роботи було розглянуто та проаналізовано методологію впровадження та підтримки системи управління складом. А також створено програмний продукт для скорочення часу, зусиль і витрат на управління і підтримку комплексу управління складом.

Отримані під час роботи результати можна покращити за рахунок впровадження нового функціоналу в систему та оптимізації реалізованого функціоналу.

ABSTRACT

Master thesis: 103 p., 13 fig., 47 tabl., 2 append., 27 sources.

AUTOMATION OF WAREHOUSE PROCESSES, WAREHOUSE MANAGEMENT SYSTEM, MANAGEMENT MECHANISM, IMPLEMENTATION, SUPPORT, WAREHOUSE MANAGEMENT COMPLEX.

The purpose of this work is to reduce the time, effort and costs of managing and maintaining a warehouse management system based on the method using technical management mechanisms.

The object of this work is the research and analysis of the methodology of management and support of the warehouse management system, as well as the development of a software product to minimize the time and effort spent on the maintenance and implementation of the warehouse management complex based on the method using management mechanisms.

The subject of the research is the methods and mechanisms of management and support of the warehouse management system.

The relevance of the topic is that the automation of business processes, as well as warehouse logistics processes, plays a significant role in the economic development of the company. This, in turn, increases the efficiency of the operation of the enterprise or company in any conditions, which is one of the most important tasks that contributes to the development of market relations in conditions of competition.

As a result of the work, the methodology of implementation and support of the warehouse management system was considered and analyzed. A software product was also created to reduce the time, effort and costs of managing and maintaining the warehouse management complex.

The results obtained during operation can be improved by introducing new functionality into the system and optimizing the implemented functionality.

ЗМІСТ

РОЗДІЛ 1	ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ТА МЕХАНІЗМІВ ПІДТРИМКИ І УПРАВЛІННЯ КОМПЛЕКСОМ УПРАВЛІННЯ СКЛАДОМ	11
1.1	Актуальність, суть, значення та завдання комплексу управління складом	
1.2	Організація впровадження комплексу управління складом	
1.2.1	Етапи впровадження комплексу управління складськими процесами	14
1.2.2	Підходи впровадження комплексу управління складськими процесами	19
1.2.3	Порівняння підходів впровадження комплексу управління складськими процесами	21
1.3	Організація підтримки комплексу управління складськими процесами	22
1.4	Механізми управління комплексом управління складом	28
1.5	Проблеми впровадження комплексів управління складськими процесами	31
1.6	Постановка задачі	33
1.7	Висновки до розділу	34
РОЗДІЛ 2	ОРГАНІЗАЦІЯ БАЗИ ДАНИХ СИСТЕМИ УПРАВЛІННЯ СКЛАДСЬКИМИ ПРОЦЕСАМИ	36
2.1	Загальні поняття і методи проектування баз даних для комплексу управління складом	36
2.2	Опис даних системи управління складськими процесами. Особливості	38
2.3	Огляд літератури	43
2.4	Висновки до розділу	45
РОЗДІЛ 3	ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДЛЯ ПІДТРИМКИ І УПРАВЛІННЯ СИСТЕМИ УПРАВЛІННЯ СКЛАДОМ	46
3.1	Вибір засобів та механізмів для реалізації застосунку для підтримки і управління системи управління складом	46

3.2 Опис загального механізму роботи з застосунком для підтримки і управління системою управління складом	47
3.3 Розробка функціоналу застосунку для підтримки і управління системою управління складом	49
3.4 Розробка інтерфейсу застосунку для підтримки і управління системою управління складом	51
3.5 Результати використання застосунку для підтримки і управління комплексом управління складом	55
3.6 Висновки до розділу	59
РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП ПРОЕКТУ	60
4.1 План розробки стартапу та масштабування його на ринок	60
4.2 Опис ідеї стартап-проекту	62
4.3 Технологічний аудит ідеї проекту	63
4.4 Аналіз ринкових можливостей запуску стартап-проекту	66
4.5 Розроблення ринкової стратегії стартап-проекту	76
4.6 Розроблення маркетингової програми стартап-проекту	79
4.7 Висновки до розділу	82
ВИСНОВКИ	83
СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ	84
ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ	86

ВСТУП

Комплекс управління складом – це система управління складськими операціями, завдання якої є забезпечити автоматизовану діяльність складських операцій. До таких операцій належать приймання, розміщення, зберігання, облік, підготовка для подальшого споживання та відправка товару до місця призначення.

Системи управління та підтримки складів відіграють важливу роль у складських та логістичних функціях в цілому. За допомогою таких систем управління великими складами або цілими складськими комплексами стає простішим і легшим, оскільки виконується повна автоматизація, що, в свою чергу, значно полегшує пошук інформації, контроль та зберігання для кожного продукту на складі. За допомогою такої системи майже всі процеси на складі можна автоматизувати.

Складність впровадження і підтримки комплексу управління складом залежить від сфери діяльності бізнесу, масштабності проекту та доступних людських ресурсів. Впровадження системи управління складом займає зазвичай декілька місяців, і вимагає великої кількості фінансових і робочих затрат. Є багато методів та механізмів ведення проектів, які організовують і скорочують час на впровадження системи управління складом, проте практично немає системи, яка б скоротила час та зусилля на впровадження системи управління складом з технічної сторони для більшості команд для впровадження системи управління складом та операторів системи. Це означає, що тема роботи є актуальною і несе в собі новизну в сфері впровадження і підтримки комплексів управління складом.

Об'єктом даної роботи є дослідження та аналіз методології управління та підтримки системи управління складом, а також розробка програмного продукту, для мінімізації часу та зусиль витрачених на підтримку і

впровадження комплексу управління складом на основі методу з використанням механізмів управління.

Предметом дослідження є методи і механізми управління і підтримки комплексом управління складом.

Метою дослідження є скорочення часу, зусиль і витрат на управління і підтримку комплексу управління складом.

Наукова новизна дослідження полягає в тому, що розроблено метод на основі технічного механізму управління, за допомогою якого пропонується вирішити проблему витрат на підтримку і впровадження комплексу управління складом.

Перший розділ даної роботи присвячений аналізу предметної області. В ньому більш детально описана актуальність впровадження системи управління складом, загальна методологія впровадження і підтримки системи управління складом, розглянуто і проаналізовано різні методи і механізми управління, розглянуто проблеми, які можуть виникнути під час впровадження комплексу управління складом.

Другий розділ описує більш систему управління складом зі сторони технічного підходу. Розглянуто загальні поняття та методи проектування баз даних для систем управління складом, а також даних, що можуть використовуватись в таких системах. Також зроблено огляд літератури по темі роботи.

Третій розділ присвячений огляду, аналізу та вибору засобів для розробки програмного продукту, опису роботи та огляду розробленого програмного продукту та завдань, які він може виконувати. Проведено аналіз ефективності роботи програмного продукту через огляд витрат на впровадження і підтримку комплексу управління складом з та без використання методу з технічним механізмом управління.

Четвертий розділ присвячено розробці власного стартап проекту на тему роботи.

РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ТА МЕХАНІЗМІВ ПІДТРИМКИ І УПРАВЛІННЯ КОМПЛЕКСОМ УПРАВЛІННЯ СКЛАДОМ

1.1 Актуальність, суть, значення та завдання комплексу управління складом

Практично кожне виробниче або торгове підприємство має склад для зберігання, на якому розміщується сировина або власна продукція. Так як складські операції є однією з найважливіших складових в ціноутворенні товару, організація складу є важливим етапом розвитку будь-якої фірми. Тому на шляху товару від виробника до кінцевого споживача можна досягти мінімізації витрат навіть якщо не оптимально організований процес виробництва, купівлі-продажу, фінансових розрахунків, тощо. Недооцінка важливості складських операцій з боку керівників підприємств веде до високих витрат при обробці або збереганні товарів.

Склад використовується для зберігання запасів на всіх етапах логістичного процесу. Існують два типи запасів:

- сировина, компоненти і запасні частини;
- готова продукція (будь-який матеріальний потік, що починається з запасу) [2].

В ході досліджень складських операцій було встановлено, що на їх суть і значення впливають два фактори: час та якість. Час є фактором ефективності складування. Чим швидше виконується складська операція, тим краще це для функціонування складу в цілому. Наприклад, від швидкості збирання, пакування, завантаження вихідного замовлення, напряду залежить час, коли товар потрапить до клієнта або кінцевого споживача. З цього випливає, що найкращі складські операції – ті, які спроектовані з урахуванням мети скорочення кожного кроку операції.

Якість є не менш важливим фактором. На ринку клієнтів складських послуг ключовим є завдання максимально якісної роботи складу. Товар має

зберігатись у відповідних умовах, що критично важливо для галузі фармацевтики, наприклад [3].

Система управління складом – це система із застосуванням комп'ютеризованих технологій, за допомогою якої оптимізується та автоматизується вся складська діяльність. Скорочена назва цієї системи – WMS (англ. Warehouse Management System – система управління складом) [4]. Призначення система управління складом полягає в підвищенні ефективності роботи складу, що, в свою чергу, забезпечує підвищення продуктивності робочого персоналу. Система управління складськими процесами є необхідною в підприємствах будь-якої галузі: логістика, виробництво, оптова або роздрібна торгівля, фармацевтика, тощо. Основною задачею даної системи є управління матеріальними потоками. Система управління складом вирішує такі питання, як пропускна здатність зберігання, якість і швидкість збірки та пакування замовлення, його відвантаження. Робота з системою управління складом починається на етапі планування її впровадження. При впровадженні даної системи на підприємстві вирішуються такі задачі:

- прийом товарів або сировини;
- складування;
- управління запасами;
- контроль запасів та їх поповнення;
- збір, пакування та відвантаження замовлення;
- зберігання.

1.2 Організація впровадження комплексу управління складом

Впровадження системи управління складом несе економічний сенс всюди, де може здійснюватись приймання, зберігання, переміщення, відправка та облік товарів, сировини, відправлень, тощо, в тому числі архівів. Сенс застосування системи управління складом є не тільки на великих логістичних центрах, підприємствах чи складах з оптової торгівлі, а також у невеликих складах, з невеликими операційними обсягами або невеликою кількістю товару. Адже незалежно від кількості товару, замовлень, розмірів приміщення, автоматизація значно зменшує витрати часу, і внаслідок цього коштів.

Впровадження системи управління складом дозволяє:

- організувати раціональне розміщення і зберігання продукції;
- забезпечити ефективне управління товарами з обмеженим терміном придатності;
- спростити виконання всіх функцій, усунути роботу з паперами, що є малоефективною та трудомісткою;
- прискорити і спростити отримання інформації про кількість і місце розташування товару;
- забезпечити ефективне управління прийомом і відвантаженням товарів, прискорити формування замовлень, виключити помилки при підготовці відвантажень;
- підвищити якість та контроль над роботою персоналу складу;
- оптимізувати використання складських площ;
- мінімізувати роботи по інвентаризації складу.

Рішення впровадити систему управління складом може бути прийнято через багато різних проблем, таких як продуктивність, не оптимізовані процеси, багато помилок при обробці замовлень внаслідок людського фактору, тощо. У таких випадках оптимізація бізнес-процесів і впровадження автоматизації допоможуть підвищити ефективність та точність виконання процесів.

Підвищена точність, якість, легкість та швидкість у виконанні складських процесів і є основними причинами впровадження системи управління складом.

Однак успіх впровадження системою управління складом базується на деяких організаційних та технічних факторах. Наприклад, пошук часу на впровадження, вибір досвідченого керівника проекту, пошук та навчання співробітників, пошук консультантів та команди впровадження, тестування системи.

Крім того, перед початком впровадження системи управління складом потрібно переконатись, що процеси загалом підібрані правильно. Якщо ж компанія вже використовує якусь систему автоматизації, наприклад ERP, і бажає інтегрувати її в нову систему, необхідно переконатись, що дані системи сумісні і можуть використовуватись разом без перебоїв. Також важливо строго дотримуватись методів управління проектом.

1.2.1 Етапи впровадження комплексу управління складськими процесами

Загалом процес впровадження системи управління складськими процесами можна поділити на 7 етапів [5]:

1. На першому етапі відбувається загальний огляд та аналіз процесів. Для початку необхідно визначити необхідні внутрішні процеси та дослідити функціональні можливості для складських операцій. Після знайомства з командою впровадження, першим кроком, є функціональний аналіз складу. Суть аналізу полягає в розгляді та аналізі всіх внутрішніх складських процесів, які потрібно буде в майбутньому налаштувати в системі. Для цього знадобиться залучення людей з відділів, які беруть участь у цих процесах, як правило, з операційних та ІТ. Найкращий спосіб зібрати цю важливу інформацію – організувати зустрічі на місці, або, якщо можливо, на відеодзвінок, якщо команда знаходиться на відстані. Це дозволить ретельно переглянути та оптимізувати наявні процеси, визначити стандартні та

користувацькі процеси. На цьому етапі кожен, хто працює над проектом впровадження, має зрозуміти, як фізично досягаються процеси, перш ніж перейти до їх програмної розробки.

2. Комунікація та розробка. Після завершення огляду та аналізу процесів можна почати конфігурацію системи. Основною метою цього кроку є відтворення потреб і вимог процесів, які були задокументовані та узгоджені під час зустрічей та наступних обговорень. Ідея полягає в тому, щоб максимально обмежити кількість помилок та некоректне використання системи. Важливим фактором успіху цього кроку є підтримка постійної комунікації з командою впровадження. Хорошою практикою є встановлення щотижневих дзвінків. Ці зустрічі дозволяють відслідковувати кожен крок проекту впровадження та отримати коментарі та відгуки щодо кожної функції.
3. Підготовка середовища. Даний етап відбувається після узгодження планування складу та потреб в обладнанні. Якість виконання цього етапу впливає на продуктивність впроваджуваної системи управління складом в майбутньому. Під підготовкою середовища мається на увазі фізичне обладнання складського середовища: нове обладнання (наприклад, навантажувачі), додаткове програмне забезпечення для роботи та підтримки системи управління складом, модеми, принтери для документів та етикеток, радіочастотні пристрої. На цьому етапі також узгоджується організація нумерації палет, контейнерів, одиниць товару.
4. Оцінка ризиків та їх мінімізація. Етап оцінки і мінімізації ризиків триває протягом всього проекту з впровадження системи. Його необхідно включити в планування та врахувати при прогнозуванні бюджету, якщо станеться непередбачувана ситуація. Більш ризикованим є одночасне впровадження декількох систем, наприклад, ERP і WMS. Також на даному етапі важливо визначити пріоритети та початковий обсяг проекту для подальшого їх дотримання. Якщо під

час впровадження системи управління складом з'являються нові вимоги, вони зазвичай або додаються до загального обсягу, або розглядаються на майбутній фазі проекту.

5. Тестування. Етап тестування відбувається для перевірки командою проекту налаштувань системи і інтерфейсу, щоб переконатися, що все працює належним чином. Важливо проводити тестування з обох боків операційної команди проекту та командою впровадження системи для того щоб переконатись, що налаштований програмний процес збігається з фізичним процесом на складі. Далі клієнт затверджує процеси один за одним і після цього процеси блокуються, що означає неможливість змінювати їх з ризиком перевищення витрат та часу. Звісно на вимогу клієнта процеси в подальшому можуть бути змінені або розширені, але по завершенню даного проекту з провадження системи управління складом.
6. Навчання персоналу та управління змінами. Даний етап передбачає, що після перевірки та узгодження всіх процесів з клієнтом, консультанти з системи управління складом навчають операційний персонал щодо функцій і процесів системи. Суперкористувачі зазвичай проходять навчання першими і навчають кінцевих користувачів – операторів складу, збиральників, пакувальників, тощо. Важливо щоб усі зацікавлені працівники брали участь в проекті впровадження на всіх рівнях. Наприклад, якщо система управління складом впроваджується в пік сезону, коли навантаження на склад доволі високе, це може призвести до затримок і проект не буде успішним через високу зайнятість персоналу. Оператори складу найкраще знають, як відбуваються процеси на їх складі, тому важливо, щоб у них була можливість висловити свою думку. Це не тільки позитивно впливає на процес налаштування складських процесів, а ще і показує працівникам, що вони є частиною проекту і їх думка справді важлива. Проектна група має складатись з різних ієрархічних рівнів і

заохочення такої групи до існування навіть після завершення проекту буде корисним для вдосконалення системи управління складом.

7. Остаточна підготовка та введення в експлуатацію. Останній етап проекту з впровадження – підготовка та введення в експлуатацію. На даному етапі в систему вводиться інформація про запаси вже наявні на складі, перевірка важливих функцій та вихідних даних. Зазвичай складається список пунктів, які мають бути верифіковані до введення в експлуатацію. Після того як все буде підтверджено, систему управління складом вводять в експлуатацію. Хорошою практикою є процес «гіперопіки» системи консультантами протягом перших двох тижнів. Під час «гіперопіки» консультанти знаходяться на складі та переконуються, що все працює належним чином.

Основні етапи наведеного переліку зображено на рисунку 1.1.

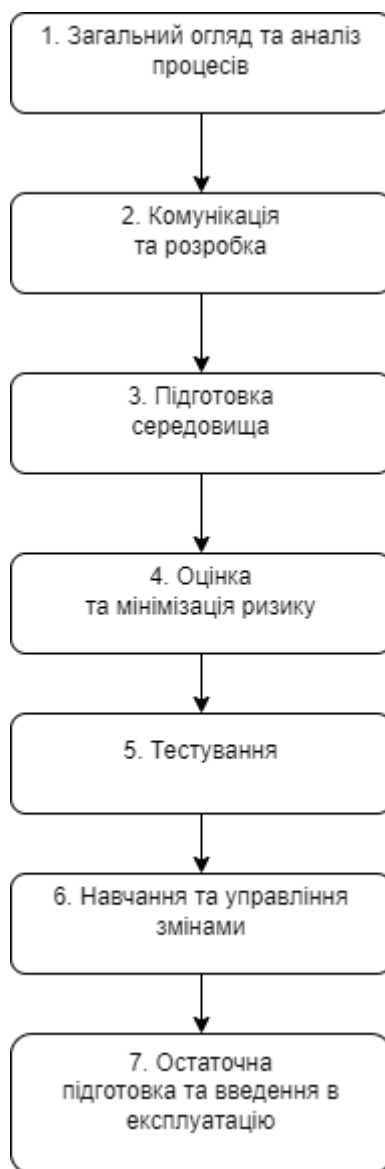


Рисунок 1.1 – Етапи впровадження системи управління складськими процесами

Отже, впровадження системи управління складом приносить багато переваг, як зниження витрат, підвищення продуктивності операцій. Такі системи також допомагають у веденні обліку, адже з них можна отримати різні звіти, що дозволяє керівникам складу проаналізувати ефективність тих чи інших процесів на складі. Вирішальними елементами успішного впровадження системи управління складом є наявність досвідчених людських ресурсів для проекту, час та належне управління організаційними змінами.

1.2.2 Підходи впровадження комплексу управління складськими процесами

Для успішної реалізації проекту компаніям часто доводиться вносити зміни в проект через зміни вимог, пріоритетів, ризиків, тощо. Це також стосується і впровадження комплексу управління складом. Для того щоб прискорити впровадження, покращити управління проектом та підвищити продуктивність використовуються гнучкі методи управління. Головна суть гнучких методів – прийняття змін, адже зміни є природньою частиною проекту і має бути можливість їх розглянути під час проекту. Найпоширеніші з гнучких методів – Scrum, Kanban, а також Scrumban.

Scrum — це гнучкий метод управління, який містить принципи та практики, що допомагають командам якнайшвидше постачати нові продукти або проекти за умови постійного вдосконалення та швидкої адаптації до змін. Scrum має три ролі: власник продукту (клієнт, який відповідає за рентабельність проекту), Scrum Master (спостерігає за командою, гарантує, що правила Scrum не порушуються, і усуває будь-які перешкоди, які команда може мати) і команда (міжфункціональна команда, яка відповідає за розробку продукту). Спринт є ітерацією фіксованої довжини і представляє основну одиницю етапу розробки. Перед кожним спринтом відбувається подія планування спринту, у якій визначається відставання спринту. Усі спринти закінчуються зустрічами та ретроспективним оглядом спринтів. Команда та власник продукту беруть участь у оглядах спринту і шукають можливості для вдосконалення [19, 20].

Kanban — це метод управління процесами, заснований на досвіді інших гнучких методів. Основна мета даного методу – усунення затримок, що позитивно впливає на оптимізацію робочого процесу. Він заснований на техніці для планування завдань, яка вимагає точного визначення та реалізації завдання якомога пізніше в робочому процесі, щоб позбутися непотрібного повторного планування [21]. Зазвичай в даному методі використовується дошка (Рис. 1.2), на якій у стовпцях дошки чітко визначені всі необхідні етапи завдання.

Завдання розподіляються за пріоритетом і розміщуються у відповідній колонці, яка найкраще визначає поточний стан завдання. Завдання переміщуються між станами, доки вони не перейдуть у стан «Зроблено» — мета полягає в тому, щоб якнайшвидше завершити завдання, які вже є в потоці, замість того, щоб починати нові [21]. Кожен крок у завдання має обмеження незавершеної роботи (WIP), яке визначається відповідно до складності, кількості працівників та інших параметрів. Таке обмеження змушує зосереджуватися на одному завданні замість того, щоб виконувати кілька завдань одночасно. Kanban заохочує використання гнучких показників для вимірювання продуктивності, моніторингу прогресу та покращує ефективність робочого процесу [21].

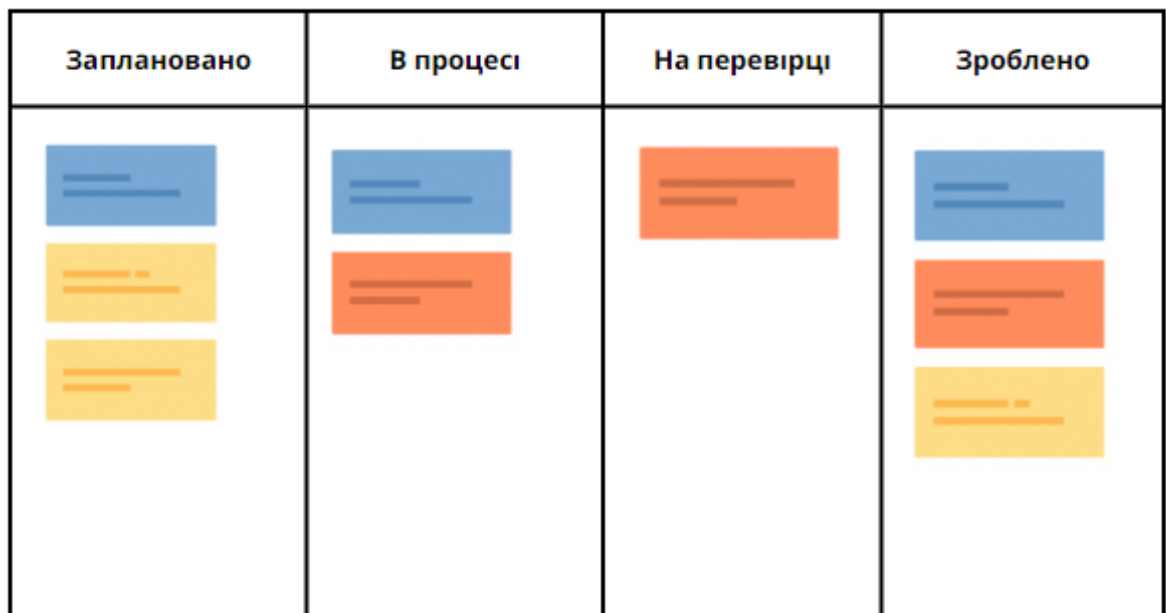


Рисунок 1.2 – Приклад дошки Kanban

Scrumban – це суміш методів Scrum і Kanban, він містить основні властивості Scrum і гнучкість Kanban. Довгострокові цілі розвитку в Scrumban визначаються за допомогою планування розміру сегмента. Кожен сегмент містить план розвитку, який потрібно реалізувати за визначений час: наприклад, 3 місяці. Подібний сегмент буде містити детально визначені завдання, проте сегмент на 1 рік містить лише чернетку завдань. Це завдяки техніці для планування завдань з методу Kanban, яка вимагає детального планування якомога пізніше. В Scrumban також заохочуються щоденні зустрічі для вирішення поточних перешкод. Scrumban передбачає ітерації, які не

повинні тривати довше 2 тижнів, але на відміну від Scrum, він дозволяє довгострокові завдання, які можуть тривати кілька ітерацій [22, 23].

1.2.3 Порівняння підходів впровадження комплексу управління складськими процесами

Порівняємо методи відповідно до їх основних функцій:

1. Дошка: використовується у всіх методах, проте у кожному методі по-різному. Наприклад, дошка Scrumban оновлюється з кожним спринтом, це означає, що всі завдання поміщаються в стовпець «Заплановано до роботи». При використанні Kanban таке оновлення не відбувається, тому що немає ітерацій – нові завдання надаються постійно.
2. Завдання: Scrum вимагає чітко визначеного беклогу, резерву спринту, в той час як Kanban не вимагає жодного конкретного беклогу, а Scrumban потребує створення планованих сегментів.
3. Ітерації: Scrum визначає ітерації (які називаються спринтами) як частину життєвого циклу. Вони можуть тривати від одного до трьох тижнів. Scrumban також має ітерації, які не є чітко визначеними щодо завдань та тривалості, однак їх тривалість не повинна перевищувати два тижні, оскільки коротші ітерації дозволяють швидше адаптуватися до змін. Kanban не визначає ітерації.
4. Пріоритет: у Scrum пріоритетність завдань визначається під час планування спринту, у той час як пріоритезація завдань Kanban виконується щодня за принципом своєчасного планування та вилучення. Scrumban спочатку визначає пріоритетність роботи за допомогою планування розміру сегмента, потім завдання

визначаються та встановлюються пріоритети для кожної ітерації а згодом щодня, як і в Kanban.

5. Команда: Scrum-команди мають бути багатфункціональними, Kanban і Scrumban дозволяють створювати міжфункціональні та спеціалізовані команди залежно від типу продукту та того, що найкраще підходить для певного сценарію.
6. Ролі: Scrum передбачає ролі власника продукту, команди розробників і майстеру Scrum. Kanban і Scrumban не визначають жодних спеціальних ролей.
7. Зміни в ході роботи: Scrum не дозволяє змінювати робочий план під час спринту, У Scrumban і Kanban немає правил, які забороняють змінювати робочий план у будь-який момент часу.

Отже, у кожного методу є свої переваги та недоліки, але необхідно мати на увазі, що жоден із них не принесе користі бізнесу, якщо використовувати його неправильно. Тому важливо вибрати один гнучкий метод, який найкраще відповідає вимогам і побажанням компанії. Scrum найкраще працює в компаніях, які мають досвідчені команди, які мають досвід роботи більше одного року. Для компаній із безперервним виробництвом, які потребують швидкого реагування на зміни, і груп, які працюють над підтримкою та обслуговуванням продукту, краще підійде Kanban. Scrumban найкраще підходить для молодих невеликих компаній, оскільки він містить гнучкість Kanban та основні характеристики Scrum.

1.3 Організація підтримки комплексу управління складськими процесами

Після того, як система управління складськими процесами була введена в експлуатацію, її підтримка здійснюється операторами складу, суперкористувачами та/або службою підтримки.

Технічна підтримка або служба підтримки – це служба, яка підтримує користувачів технологічних продуктів або послуг. Технічна підтримка зазвичай зосереджується не на навчанні персоналу, а на допомозі з конкретною проблемою користувача [6].

Технічна підтримка може складатися з одного співробітника для невеликих компаній або може включати багато відділів і працівників. Наприклад, великі компанії часто мають два типи команд:

1. Внутрішня команда IT-підтримки, яка допомагає співробітникам, коли вони мають справу з технічною проблемою. Наприклад, ноутбуком, принтером, системними застосунками, тощо.
2. Команда зовнішньої підтримки допомагає користувачам системи, в даному випадку системи управління складськими процесами.

Технічна підтримка може надаватися різними способами, залежно рівня підтримки. Наприклад, через електронну пошту, форум, текстовим онлайн чатом, відео- або телефонними дзвінками, онлайн інструкціями або технічною документацією, тікет системи, тощо.

Зазвичай технічна підтримка складається з декількох рівнів підтримки (Рис. 1.3). Така структура ефективна з багатьох причин:

- швидке вирішення дрібних або простих проблем;
- визначення протоколу для складніших проблем;
- стратегічне задоволення потреб клієнтів;
- створення позитивного досвіду клієнтів;
- підвищення задоволеності співробітників;
- отримання відгуків та пропозицій щодо розробки продукту відповідно до кожної лінії підтримки.

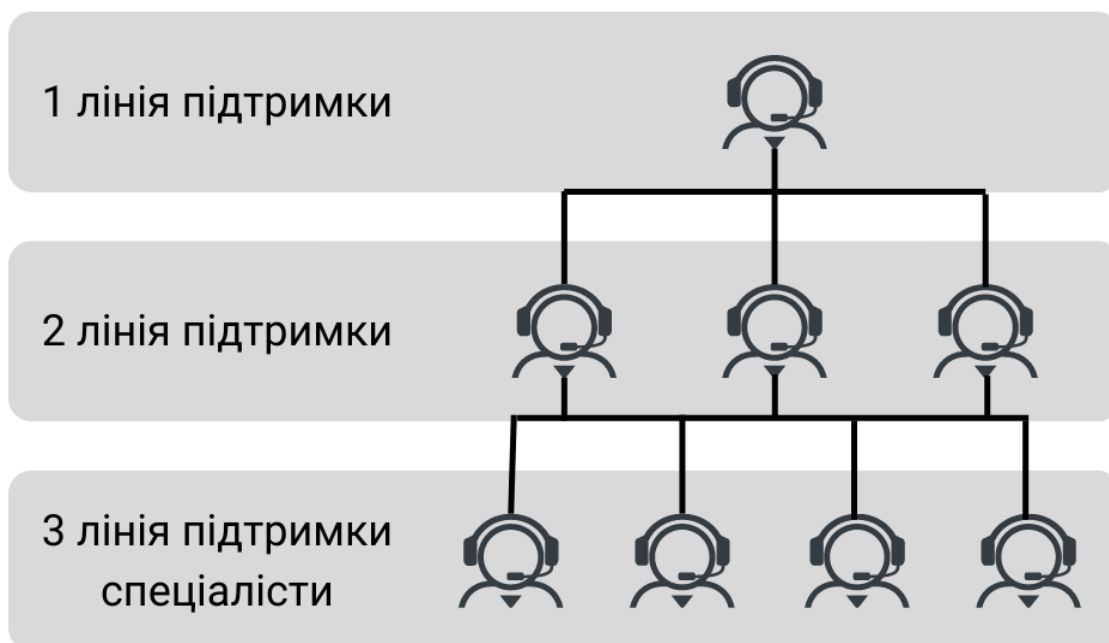


Рисунок 1.3 – Схема рівнів підтримки технічної підтримки

Звісно структура служби підтримки варіюється від компанії до компанії. Типова структура технічної підтримки зазвичай організовується навколо рівнів підтримки, наведених у таблиці 1.1.

Компанії зазвичай змінюють цей шаблон і поєднують рівні підтримки відповідно до своїх ресурсів в цілях його економії. У деяких організаціях функції рівня 0 і рівня 1 виконує один і той же персонал. Інші компанії можуть об'єднати функції рівня 2 і рівня 3, наприклад.

Також компанії можуть використовувати часову шкалу і пріоритетність для передачі проблеми на наступний рівень. Наприклад, якщо рішення проблеми з високим пріоритетом займає більше 1 години у рівня 1, вона має бути передана на рівень 2.

Таблиця 1.1 Структура технічної підтримки

Лінія підтримки	Функції	Методологія підтримки	Потреби співробітників
0 лінія підтримки	Самодопомога та інформація, яку отримує користувач	Користувачі отримують інформацію про підтримку з веб- та мобільних сторінок або додатків, включаючи поширені запитання, детальну інформацію про продукт і технічну інформацію, дописи в блогах, технічну документацію та функції пошуку. Користувачі також можуть запитувати колег та отримувати послуги без залучення технічної підтримки. Електронна пошта, веб-форми та соціальні контакти, використовуються для надсилання запитань і запитів до вищих рівнів підтримки або персоналу компанії.	Рівень 0 потребує технічних ресурсів для створення, підтримки та оновлення інформації про продукт. Створенням веб-сайтів і додатків займається команда розробників. Модератори використовуються для моніторингу форумів клієнтів. Персонал рівня 1 відповідає на запити, отримані через електронну пошту, веб-сайти чи соціальні мережі.

Продовження таблиці 1.1

		Форуми клієнтів дозволяють користувачам використовувати краудсорсингові рішення, як правило, без участі персоналу компанії.	
1 лінія підтримки	Базове рішення служби підтримки	Підтримка основних проблем клієнтів, таких як вирішення проблем використання та виконання запитів служби підтримки, які потребують участі ІТ. Якщо рішення не вдається знайти, персонал 1 рівня передає інциденти на вищий рівень.	Технічний персонал нижчого рівня, навчений вирішувати відомі проблеми та виконувати запити на обслуговування за допомогою сценаріїв.
2 лінія підтримки	Поглиблена технічна підтримка	Досвідчені та обізнані технічні спеціалісти оцінюють проблеми та надають рішення для проблем, які не можуть бути врегульовані за допомогою рівня 1. Якщо рішення не	Персонал із глибокими знаннями про продукт чи послугу, але не обов'язково інженерів чи програмістів, які розробили та створили продукт.

Продовження таблиці 1.1

		вдається знайти, підтримка рівня 2 переносить інцидент до рівня 3.	
3 лінія підтримки	Експертна підтримка продуктів і послуг	Доступ до найвищих доступних технічних ресурсів для вирішення проблем або створення нових функцій. Техніки рівня 3 намагаються повторити проблеми та визначити першопричини, використовуючи дизайн продуктів, код або специфікації. Після виявлення причини компанія вирішує, чи створювати нове рішення, залежно від причини проблеми. Нові виправлення задокументовані для використання персоналом рівня 1 і рівня 2.	Фахівці рівня 3, як правило, є найбільш висококваліфікованими спеціалістами з продуктів, і можуть включати творців, головних архітекторів або інженерів, які створили продукт або послугу.

Продовження таблиці 1.1

4 лінія підтримки	Зовнішня підтримка проблем, які не підтримуються організацією	Підтримка за контрактом для товарів, що надаються організацією, але не обслуговуються безпосередньо, включаючи підтримку принтерів, підтримку програмного забезпечення постачальників, технічне обслуговування та інші послуги стороннього виконавця. Проблеми або запити пересилаються до підтримки рівня 4 і контролюються організацією для реалізації.	Переважають постачальники та бізнес-партнери, які надають підтримку та послуги для товарів, які надає компанія.
-------------------	---	---	---

1.4 Механізми управління комплексом управління складом

Поняття «механізм управління» не має чіткого і уніфікованого визначення, адже використовується в різних сферах з різним підходом. Загалом для розуміння цього терміну виділяють два підходи. У вузькому розумінні «механізм управління» — це інструмент, система, за допомогою якої створюються управлінські явища та процеси в організації. У широкому тлумаченні «механізм управління» прийнять розглядаєти як засіб реалізації управлінських рішень, приведення в дію окремих ланок керованої системи з метою стимулювання інших ланок системи до необхідного руху, що спрямований на досягнення позитивного результату. Складовими елементами механізму управління є методи, форми, засоби, принципи, об'єкти управління, на які здійснюється вплив для досягнення заздалегідь поставлених завдань і визначених цілей [24]. Загалом виділяють сім видів механізмів управління:

1. адміністративний;
2. організаційний;
3. економічний;
4. технічний;
5. структурний;
6. інформаційний;
7. державний.

Механізм адміністративного управління – це система скоординованих дій, процедур, що забезпечують стабільний розвиток і діяльність організації шляхом підтримки дисципліни і порядку, що забезпечує отримання позитивних економічних результатів і підвищення ефективності діяльності. Безпосередню участь у цьому механізмі управління бере керівник підприємства, який визначає правила гри, стандарти, норми, формує бізнес-політику, тобто, по суті, це законодавча гілка влади в організації, яка додатково виконує виконавчу. функції в управлінні та приймає стратегічні рішення та відповідає за них.

Організаційний механізм управління — це сукупність елементів і компонентів, які створюють організаційну основу процесів і явищ. Загальноприйнятого тлумачення сутності цього виду не існує, кожен з вчених

розуміє його по-своєму, одні вважають, що це певна кількість взаємопов'язаних процесів, які відбуваються в організації, інші стверджують, що це життєві цикли організації, правила та стандарти, за якими функціонує вся система управління суб'єкта господарювання.

Механізм управління економікою - це сукупність елементів планування і прогнозування, за допомогою яких здійснюється формування політик ціноутворення, економічне стимулювання, фінансове кредитування, налагодження системи формування і розподілу прибутку, стимулювання продуктивності трудових ресурсів і вдосконалення матеріально-технічного забезпечення відбувається в організації [25]. Цей механізм залежить від зовнішніх і внутрішніх факторів ринкового середовища, зміни в якому постійно відбуваються, тому доцільно в цьому контексті також використовувати інноваційний менеджмент, адже сьогодні інновації виступають рушійною силою підвищення конкурентоспроможності підприємства та її ефективність у виробничо-господарській діяльності, що є основними завданнями в реалізації економічного механізму.

Сутність механізму технічного управління полягає у впровадженні сучасних цифрових технологій у виробничий процес підприємницької діяльності, автоматизації шляхом заміни обладнання на нове. При цьому управління всіма процесами здійснюється в режимі реального часу та з урахуванням змін у зовнішніх умовах. За рахунок цього відбувається значна економія ресурсів компанії, що дозволяє досягти швидкого результату щодо досягнення поставленої мети. У цьому типі механізму управління важливу роль відіграє застосування Інтернет-технологій, оскільки вони забезпечують швидкий зв'язок між персоналом і машинами.

Структурний механізм управління – це прийоми послідовного розподілу обов'язків у механізмі управління та між підрозділами, загальноприйняті дії, які мають терміни виконання, при цьому всім відомо, що порушення термінів позначиться на загальному результаті та призведе до відсутності мотивації для кожного з учасників.

Механізм інформаційного управління полягає в забезпеченні збору та швидкої доставки нової, достовірної, актуальної інформації з питань зовнішнього середовища до управлінського персоналу всіх рівнів підприємства, яка структурована за функціями та функціями та виконуваних ними завдань.

Механізм державного управління – важелі, інструменти, елементи, процеси та закони соціально-економічного характеру, за допомогою яких органи виконавчої та законодавчої влади впливають на економічну, фінансову, виробничу діяльність підприємств, організацій з метою розвитку та стабілізації існуючої соціально-економічної системи.

Комплексне та успішне впровадження всіх видів механізмів управління призводить до зміцнення конкурентних позицій на ринку, підвищення конкурентоспроможності компанії, зміцнення позицій на ринку та досягнення цілей, які були передбачені на етапі стратегічного планування діяльності з розвитку. При правильному управлінні кожен елемент механізму повинен мати своє відображення у своєму зовнішньому вигляді, оскільки це невід'ємні частини системи управління.

1.5 Проблеми впровадження комплексів управління складськими процесами

Кожна компанія або бізнес має свій підхід до вирішення завдання автоматизації складських процесів. Проте всі вони стикаються з низкою проблем у процесі впровадження.

Основною проблемою реалізації є недостатня оптимізація складу. Для ефективної роботи навіть без автоматизації складу необхідно максимально раціонально використовувати його площу та об'єм. Стелажі слід підбирати відповідно до висоти стелі і розташовувати так, щоб ширина проходів була оптимальною для переміщення обладнання. Крім того, для раціонального управління складом його слід розділити на різні зони, призначені для певних

функцій. У складі кожна комірка повинна мати адресу, щоб забезпечити швидкий пошук і розміщення товарів.

Поширеною проблемою під час впровадження комплексу управління складом є коли кілька людей намагаються керувати впровадженням. Під час впровадження ролі персоналу мають бути чітко розподілені для розуміння власних відповідальностей, а також відповідальностей колег.

Ще одна проблема – навчання персоналу користуванню системою управління складом. Деякі співробітники можуть бути не готові до змін у своїй роботі, не довіряти системним рішенням, а функціональність системи вимагає чіткої послідовності дій.

Програмне забезпечення не повністю відповідає вимогам бізнесу. Сучасні готові платформи пропонують широкий спектр функціональних можливостей, окрім традиційних можливостей, але програмне забезпечення може не обов'язково відповідати всім потребам бізнесу. У цих випадках можуть знадобитися незначні модифікації для задоволення будь-яких потреб. Однак важливо визначити, чи потрібні чи необхідні зміни. Також поширеною проблемою є стислі терміни та недостатнє фінансування проектів автоматизації, оскільки кожен проект індивідуальний і може потребувати різного функціоналу та стикатися з різними збоями. У цьому випадку необхідно спочатку вирішити найважливіші і базові для складу проблеми і реалізувати більш прості функції для їх вирішення, а більш складні залишити для довгострокового вирішення. Також варто зазначити, що навіть після повного впровадження автоматизації необхідно постійно підтримувати роботу системи у разі виявлення тих чи інших несправностей, а також фінансувати подальший розвиток та вдосконалення впровадженого функціоналу.

Крім того, проблема може виникнути з боку постачальника та клієнта. Важливо налаштувати автоматичну відправку інформації від постачальника або клієнта і постачальнику або клієнту. Адже кожна сторона може відправляти і отримувати інформацію в певних форматах, які повинні підтримуватися і складською системою, і постачальником, і клієнтом. Також всі товари, які

будуть прийняті на склад, повинні бути промарковані відповідним чином, тому що система управління складом висуває жорсткі вимоги до приймання товарів.

Ще однією проблемою під час впровадження є недостатня кількість даних про товар або низька якість даних. Робота автоматизованої системи на складі вимагає чітко організованої інформації про термін придатності товару, серійний номер, штрих-код тощо.

Недостатнє тестування також може спричинити проблеми. Тестування має вирішальне значення для впровадження, а якість тестування має значення.

Всі групи рішень логістичних завдань, пов'язаних з реорганізацією та оптимізацією роботи складу, взаємопов'язані. Спочатку потрібно проаналізувати потік продукту, його розмір, його динаміку. Потім сформулювати бізнес-вимоги до складу, скласти конкретний перелік завдань, які необхідно виконати, забезпечити умови зберігання продукції. І лише на завершальному етапі оцінити поточний стан складу, його обсяг, організаційну структуру, систему обліку, використовувані технології.

Аналіз проблем і невирішених завдань передуює прийняттю рішення про усунення недоліків, що в свою чергу запобігає їх появі після впровадження автоматизованої системи управління та підтримки складських процесів [7].

1.6 Постановка задачі

Основною задачею магістерської роботи є на основі проведеного аналізу реалізувати програмний продукт, який має автоматизувати процеси під час впровадження та підтримки системи управління складом для мінімізації часу та зусиль витрачених на дані процеси на основі методу з використанням механізмів управління.

Тобто до задач даної магістерської дисертації відносяться:

8. провести огляд існуючої методології впровадження комплексу управління складом;
9. провести огляд існуючої методології підтримки комплексу управління складом;
10. розробити програмний продукт для мінімізації часу та зусиль витрачених на підтримку і впровадження комплексу управління складом на основі методу з використанням механізмів управління;
11. продемонструвати результати роботи програмного продукту;
12. проаналізувати ефективність роботи програмного продукту.

1.7 Висновки до розділу

Таким чином, автоматизація складських процесів відіграє важливу роль у розвитку бізнесу за рахунок скорочення часу складських операцій, підвищення точності та забезпечення збереження товарів, що зберігаються на складі. Впровадження системи управління складом має економічний доціль там, де відбувається зберігання, облік і рух будь-яких одиниць зберігання. Важливу роль у впровадженні системи управління складськими процесами відіграє також організація технічної підтримки. Правильно організовані логістичні процеси на складі також повинні відповідати специфіці та необхідним параметрам роботи складу, забезпечувати безпечне зберігання товарів і оптимальний рівень обслуговування.

Для швидкого і ефективного впровадження варто використовувати гнучкі методи управління, як Scrum, Kanban, Scrumban. Вони допоможуть легко реагувати та включати зміни в проект.

Від ефективності роботи складу, своєчасного вирішення проблем, які можуть виникнути на складі під час і після впровадження автоматизованого комплексу, впровадження механізмів управління, залежить успіх і

конкурентоспроможність як самої логістичної компанії, так і компаній-замовників.

РОЗДІЛ 2 ОРГАНІЗАЦІЯ БАЗИ ДАНИХ СИСТЕМИ УПРАВЛІННЯ СКЛАДСЬКИМИ ПРОЦЕСАМИ

2.1 Загальні поняття і методи проектування баз даних для комплексу управління складом

Комплекс управління складом по суті становить базу даних, в якій містяться дані про конфігурацію системи, про вхідні, вихідні замовлення, процеси системи.

На сьогоднішній день існує багато підходів і моделей зберігання даних та доступу до них. Серед них є:

- ієрархічна модель;
- схема зірки і модель сніжинки;
- модель «сутність-зв'язок»;
- реляційна модель;
- документо-орієнтована модель;
- об'єктно орієнтована модель.

Хоча кожна з цих моделей досить цікава, але, якщо говорити про проектування бази даних, то найбільш поширеною стала реляційна модель. Ця модель використовує мову структурованих запитів SQL (англ. structured query language) для доступу до даних. Використовується в абсолютній більшості баз даних. Більшість мов програмування мають багато інструментів для інтеграції з реляційними базами даних. Це дозволяє розробити гнучку архітектуру програмного забезпечення. Таким чином, великою перевагою реляційного підходу можна назвати його популярність і велику кількість інструментів, які допомагають працювати з такими базами даних.

По-перше, потрібно визначити, що таке проектування бази даних. Т. Коннолі та К. Бегг у своїй книзі "Бази даних. Дизайн, реалізація та підтримка. Теорія та практика" визначають проектування бази даних таким чином: «Проектування бази даних – це процес створення проекту бази даних, яка

призначена для підтримки функціонування підприємства і яка допомагає досягненню його цілей» [8]. Таким чином, основна мета проектування бази даних – забезпечити можливість максимально успішно керувати бізнесом, а від вдалого проектування залежать такі фактори, як легкість подальшої розробки, можливість розширення майбутньої системи новими функціональними блоками, модулями, підсистемами. Невдале планування бази даних призведе до значних ускладнень і, отже, втрати коштів. Тому проектування бази даних слід сприймати дуже серйозно. Дослідники розрізняють два основні підходи до дизайну бази даних: висхідний та низхідний [7].

Висхідний підхід – це декомпозиція предметної області, починаючи з найнижчого рівня, тобто, власне, починаючи з аналізу сутностей і зв'язків, і тільки потім переходити до самої бази даних. При такому підході процес навчання необхідних функцій для виконання задачі можна вважати пріоритетним. Лише тоді нормалізація оптимізує раніше визначені зв'язки. Цей підхід досить цікавий і, очевидно, підходить для побудови бази даних з невеликої предметної області. Але при великій кількості можливих підключень і дуже складній системі є ризик плутанини і ускладнень при проектуванні.

Низхідний підхід охоплює потреби складних баз даних з великою кількістю можливих відносин та взаємозв'язків між ними. Значення підходу полягає в тому, щоб визначити більші сутності, які лежать на поверхні та подальшу декомпозицію з необхідними деталями.

Існують також інші підходи до проектування бази даних. Вони певною мірою є узагальненнями вищезазначеного. Наприклад, «від загального до окремого». Це, як і висхідний підхід, з початковим визначенням суб'єктів та відносин між собою, але подальше планування пов'язане з додаванням суб'єктів, які взаємодіють з уже доступним. У висхідному підході, така взаємодія не є необхідною. Також цікавим методом дизайну бази даних може бути «змішана стратегія проектування», який в той же час використовує основні методи: висхідний і низхідний. Цей метод проектування полягає в тому, що різні частини моделей формуються за допомогою двох методів. Після

цього ці частини доведеться об'єднати, щоб утворити модель, яка є результатом.

Незважаючи на те, що існують дуже добре описані методи проектування баз даних, інформація про які наведена вище, слід зазначити, що всі вони є більш емпіричним підходом. Звичайно, на даному етапі проектування системи без суб'єктивного підходу не обійтися. Але, оскільки визначити чіткі правила правильного проектування складно, це важко або навіть неможливо, це завдання вимагає не тільки серйозних теоретичних знань фахівця, а й його досвіду в подібній практиці, а це робить це завдання нетривіальним.

2.2 Опис даних системи управління складськими процесами. Особливості

Після завершення етапу проектування бази даних, тобто після декомпозиції предметної області в реляційну базу даних, наступним етапом є моделювання даних. У певному сенсі моделювання – це збір документації для нової схеми бази даних. Модель повинна розкривати суть взаємозв'язків і зв'язків між ними і слугувати певною інструкцією для розробників, які створюватимуть базу даних, а також для тих, хто буде програмувати серверну частину додатку. Документація є важливою для успішної розробки, отже, завдання якісного моделювання моделі даних. Звичайно, при розробці бази даних, а потім і серверної частини програмного комплексу і навіть клієнтської частини, виникнуть питання щодо вимог до даних. Тому дуже важливо мати чітке формулювання таких вимог. Крім того, поділ предметної області в схемі бази даних може змінюватися при зміні завдання і повинен бути достовірно вказаний в документації. Це важливо, оскільки сутності, які формують масиви, зв'язки між ними та поля масивів, які потім також увійдуть до об'єктної моделі, є фундаментальними для системи. Але якщо ці аспекти не будуть належним чином відображені в документації, це призведе до серйозних наслідків і

проблем у розвитку. Моделювання значно спрощує розуміння деяких нюансів, які виникають у процесі роботи з даними:

- використання даних відповідно до предметної області, встановлення валідних меж використання;
- характер даних;
- вимоги до даних відповідно до предметної області та задачі.

Також традиційно виділяють наступні вимоги для моделі даних [13]:

Таблиця 2.1 – Вимоги до моделі даних

Показник	Характеристика показнику
Структурна достовірність	Відповідність способу визначення і організації інформації на даному підприємстві
Цілісність	Узгодженість з методом використання і управління інформацією в підприємстві
Відсутність збитковості	Будь-яка частина даних має бути представлена тільки один раз, тобто необхідно досягти виключення зайвої інформації
Можливість розширення в майбутньому	Можливість легко адаптувати модель під нові вимоги, які можуть виникнути
Можливість сумісного використання	Відсутність належності до якогось конкретного додатку або технології. Тобто необхідно гарантувати можливість використання моделі в багатьох програмних комплексах, а не тільки в одному
Виразність	Необхідність чіткого представлення даних, зв'язків між ними, відмінностей

Продовження таблиці 2.1

Простота	Модель має бути легкодоступною для розуміння, як професіоналами, так і людьми, які не мають достатнього рівня експертизи
Схематичне представлення	Необхідність представлення моделі у вигляді схематичного зображення

Слід зазначити, що ці критерії є у широкому сенсі ідеалами моделі даних. Навряд чи можна створити модель, яка б відповідала кожному з цих критеріїв. Тому, успішна модель – це компроміс, метою якого є досягнення максимального обсягу даних і вимог.

Сучасні СУБД дозволяють обробляти дані найрізноманітніших типів, серед яких найпоширенішими є [14]:

1. Цілі числа. У стовпцях, що мають цей тип даних, зазвичай зберігаються дані про ціни, кількості, вік співробітників і т.д. Цілочисельні стовпці часто використовуються також для зберігання ідентифікаторів, таких як ідентифікатор клієнта, товару або замовлення.
2. Десяткові числа. У стовпцях цього типу зберігаються числа, що мають дрібну частину, але які необхідно обчислювати точно, наприклад курси валют та відсотки. Крім того, у таких стовпцях часто зберігаються грошові величини.
3. Числа з плаваючою комою. Стовпці такого типу використовуються для зберігання значень, які можна обчислювати приблизно, наприклад, значення ваги і відстані. Числа з плаваючою комою можуть становити більш широкий діапазон значень, ніж десяткові числа, проте при обчисленнях можливі похибки через заокруглення.
4. Рядки символів постійної довжини. Стовпці з таким типом даних зазвичай зберігають імена людей та компаній, адреси тощо.

5. Символи змінної довжини. Стівпці такого типу дозволяють зберігати ряди символів, довжина яких змінюється в певному діапазоні. Стандарт SQL1 визначив лише рядок фіксованої довжини, що простіше обробити, але вимагає більше місця для зберігання.
6. Грошові величини. Багато СУБД підтримується тип даних MONEY чи CURRENCY, який зазвичай зберігається як десяткового чи числа з плаваючою комою. Наявність окремого типу даних для представлення фінансових величин дозволяє правильно форматувати їх при виведенні на екран.
7. Дата і час. Підтримка значень дати/часу також поширена в різних СУБД, хоча його методи застосування зовсім відрізняються один від одного. Як правило, над значеннями цього даних можна виконувати різні операції. Стандарт SQL2 включає визначення типів даних DATE, TIME, TIMESTAMP та INTERVAL, а також підтримку часових поясів та можливість вказати точності часу (наприклад, десяти або соті частки секунди).
8. Булеві величини. Деякі СУБД, такі як Informix Universal Server, явно підтримують логічні значення (TRUE або FALSE), інші СУБД дозволяють виконувати в інструкціях SQL логічні операції (порівняння, логічне І/АБО та інших.) над даними.
9. Довгий текст. Багато СУБД підтримують стівпці, в яких зберігаються довгі текстові рядки (як правило довжиною до 32000 або 65000 символів, а в деяких випадках і більше). Це дозволяє зберігати цілі документи в базі даних.
10. Неструктуровані потоки байтів. Сучасні СУБД дозволяють зберігати та витягувати неструктуровані потоки байтів змінної довжини. Стівпці з таким типом даних зазвичай використовуються для зберігання графіки та відео, виконуваних файлів та інших неструктурованих даних.

Список всіх типів даних SQL наведений у таблиці 2.2.

Таблиця 2.2 – Типи даних SQL

Тип даних	Опис
CHAR	Рядки символів постійної довжини
VARCHAR	Рядки символів змінної довжини
NCHAR	Рядки локалізованих символів постійної довжини
NCHAR VARYING	Рядки локалізованих символів змінної довжини
INTEGER	Цілі числа
SMALLINT	Малі числа
BIT	Ланцюги бітів постійної довжини
BIT VARYING	Ланцюги бітів змінної довжини
NUMERIC	Числа із заданою точністю та ступенем
DOUBLE PRECISION	Числа з плаваючою комою низької точності
DATE	Дата
TIME	Час
TIMESTAMP	Дата і час
INTERVAL	Часовий інтервал
DECIMAL	Числа з плаваючою комою
REAL	Числа з плаваючою комою низької точності

У відношенні системи управління складськими процесами будуть використовуватись такі типи даних, як CHAR, INTEGER, DECIMAL та TIME.

2.3 Огляд літератури

У статті «Фактори, що впливають на успішне впровадження системи управління складом (WMS)» [15] вивчаються та ідентифікуються ключові фактори, які можуть вплинути на успішну реалізацію системи управління складськими процесами. В роботі проведено тематичне дослідження провідного постачальника логістичних та ланцюгів поставок, щоб зрозуміти та оцінити ці критичні фактори, що сприяють успішному впровадженню WMS на основі фактичних спостережень щодо впровадження проекту та документації проекту. Нижче наведені ключові фактори, що сприяють успішному впровадженню системи управління процедурою складу:

1. Система управління складськими процесами реалізована в хмарному середовищі. Переваги хмарного середовища полягають у тому, що вони дозволяють швидко впровадити систему управління складом, не потребують часу на технічне обслуговування та менше ІТ. Систему управління в хмарному середовищі можна налаштувати відповідно до унікальних потреб бізнесу та додаткових функцій. Така система управління складом ідеально підходить для середніх і великих підприємств, а також для компаній більшої складності.
2. Галузеві стандарти. Деякі постачальники пропонують галузеві стандарти (шаблони), в яких використовуються найкращі практики та процеси. Завдяки цьому очікується скорочення циклу впровадження, що скоротить час і вартість впровадження.
3. Інтеграція з системою ERP — інтеграція ERP та WMS є важливим фактором, який слід враховувати, коли бізнес-модель є складною для управління та вимагає системного зв'язку в режимі реального часу;

4. Конфігурація та налаштування - постачальники виявляють переваги гнучкого підходу до роботи за потреби, щоб система могла легко адаптуватися та покращувати свою продуктивність. Можливість легко і швидко адаптувати рішення в певному сенсі є ключовим необхідним елементом WMS для будь-якого бізнесу.
5. Програми впровадження Agile проти Waterfall. Деякі постачальники систем управління складами мають гнучкі або швидкі програми впровадження, розроблені для швидкого запуску бізнесу протягом обмеженого часу. Крім того, деякі використовують підхід Waterfall, заснований на графіку впровадження та часовій шкалі.
6. Сучасний інтерфейс. Простота використання WMS для бізнесу може значно зменшити навчання, необхідне для системи, а також полегшити навчання нових працівників у майбутньому.

У статті «Об'єктно-орієнтований підхід до реалізації системи баз даних» [16] розглядається об'єктно-орієнтоване програмування як техніка реалізації систем баз даних. Об'єктно-орієнтований підхід об'єднує представлення об'єктів бази даних і зв'язки з процесами, які ними маніпулюють. Щоб досягти цього, автори спочатку визначають абстракції імітаційних структур моделі даних, які описують їхні загальні властивості та поведінку. Потім представляють типи сутностей і типи зв'язків у концептуальній схемі та внутрішній схемі за допомогою об'єктів, які є екземплярами цих абстракцій. Загальні процедури (процедури обробки даних), які складають інтерфейс користувача, тепер можна реалізувати як виклики процедур, пов'язаних із цими об'єктами.

Представлено та обговорено загальну процедурну модель техніки реалізації бази даних. Кілька сучасних методів реалізації системи баз даних представлено як приклади цієї моделі, після чого наведено критичний аналіз методів реалізації на основі об'єктів. Автори демонструють, що об'єктно-орієнтований підхід має такі переваги, як незалежність даних, ефективність

виконання за рахунок усунення доступу до системних дескрипторів і підтримка низькорівневих представлень.

У статті «Порівняння баз даних SQL і NoSQL і їхній зв'язок з аналітикою великих даних» [17] розглядаються бази даних SQL і NoSQL, порівняння традиційних баз даних SQL з базами даних NoSQL для аналітики великих даних, моделі даних NoSQL, типи зберігання даних NoSQL, особливості та характеристики кожного сховища даних, переваги та недоліки NoSQL та RDBMS. На завершення автори зазначають, що NoSQL забезпечує сховище даних і транзакцій без схем, які дозволяють підприємствам вільно додавати поля до записів, а не вимоги структурованої обробки схеми, що є основним обмеженням баз даних SQL і гнучкого моделювання бази даних NoSQL добре підходить для динамічної масштабованості та підвищення ефективності аналізу великих даних.

2.4 Висновки до розділу

У даному розділі було оглянуто як може бути організована база даних системи управління складськими процесами та модель бази даних. Також було розглянуто основні типи, що використовуються в базах даних.

Крім того, був проведений огляд літератури, яка описує ключові фактори, які можуть вплинути на успішне впровадження системи управління складськими процесами, розглядає бази даних SQL та об'єктно-орієнтоване програмування як техніку реалізації для систем баз даних.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДЛЯ ПІДТРИМКИ І УПРАВЛІННЯ СИСТЕМИ УПРАВЛІННЯ СКЛАДОМ

3.1 Вибір засобів та механізмів для реалізації застосунку для підтримки і управління системи управління складом

Доступний широкий спектр інструментів для реалізації програми для підтримки та керування системою управління складом, але він залежить від вибору системи управління базою даних. Оскільки дана тема магістерської роботи є продовженням бакалаврської роботи зі створення програмного комплексу для управління та підтримки процесів складської компанії, то для розробки використовувалася та ж мова SQL та система управління базами даних MySQL. Вибір мови програмування для реалізації функціональних можливостей програми та графічного інтерфейсу не залежить від вибору системи управління базою даних, але оскільки було вирішено реалізувати цю програму у вигляді веб-додатку, функціоналу було обрано скриптову мову програмування PHP, а для реалізації інтерфейсу – мову HTML для розмітки та CSS для реалізації графічного інтерфейсу.

MySQL являє собою універсальну систему управління базами даних з відкритим кодом [18].

MySQL можна знайти у вільному доступі, хоча система має комерційну ліцензійну версію. MySQL в основному використовується для керування базами даних малого та середнього розміру. За механізмом доступу до бази даних MySQL відноситься до клієнт-серверної системи, тобто система управління розташована разом з базою даних на сервері, а основне завдання клієнта – організація доступу користувачів до сервера. MySQL – одна з найпопулярніших систем керування базами даних, яка має власний сервер. Він простий у використанні, підтримує практично всі функції SQL, має високий рівень захисту та високу швидкість [18].

Для реалізації функціоналу була обрана мова програмування PHP. PHP — це скриптова мова програмування, яка дозволяє створювати веб-сторінки на стороні веб-сервера. Ця мова є однією з найпоширеніших у сфері веб-розробки, підтримується більшістю провайдерів, має досить простий, але в той же час гнучкий синтаксис і інструменти для роботи з базами даних MySQL.

Для організації інтерфейсу веб-додатку використовується мова HTML і модуль CSS, тому додаток займає менше ресурсів сервера, ніж сайти, що використовують систему CMS.

Механізм взаємодії з SQL-сервером реалізований через додаток OpenServer Panel. OpenServer — це професійний інструмент веб-розробки, розроблений для операційної системи Windows і не залежить від конкретного комп'ютера. Основними перевагами OpenServer є:

- простота у використанні;
- швидка робота;
- можливість роботи з профілем налаштувань;
- доступ до різних шаблонів налаштування модулів;

3.2 Опис загального механізму роботи з застосунком для підтримки і управління системою управління складом

Застосунок для підтримки і управління системою управління складом може використовуватись як на етапі впровадження проекту, так і на етапі обслуговування. В рамках кількох екземплярів проекту. Перший варіант використання — використання програми під час налаштування процесу системи управління складом. Якщо потрібно налаштувати складний процес, рекомендується розділити конфігурацію на етапи. Після кожної частини етапу конфігурації користувач перевіряє, чи конфігурація працює правильно, і переносить конфігурацію на другий екземпляр проекту. Якщо на певному етапі

система не працюватиме, буде простіше розпізнати налаштування, які не працюють, і повторно налаштувати процес.

Другий кейс призначений для усунення несправностей. Можна швидко перевірити, чи була нещодавно змінена відповідна конфігурація. Однак один екземпляр проекту має працювати справно. Хорошою практикою є створення такого екземпляру під час введення проекту в експлуатацію.

Можна перенести конфігурацію з одного екземпляра в інший. Або, наприклад, було зроблено конфігурацію в двох екземплярах, і потрібно перевірити, чи вони збігаються.

Передумовами використання застосунку в будь-якому випадку буде встановлення та конфігурація системи управління складом. Основний потік дій користувача і системи наведений у таблиці 3.1.

Таблиця 3.1 – Основний потік дій користувача і системи

№	Користувач	Система
1	Користувач вводить логін та пароль.	
2		Система підключається до серверу.
3		Система перевіряє чи правильно введений логін та пароль. Дані введено правильно
4		Система відображає головний екран застосунку
5	Користувач заповнює джерела даних: вводить базу даних та склад для порівняння	
6	Користувач ініціює порівняння джерел даних	

Продовження таблиці 3.1

7		Система перевіряє чи правильно введені дані. Дані введено правильно
8		Система виконує алгоритм порівняння.
9		Система виводить результат на відповідні вкладки.
10	Користувач аналізує отриманий результат.	

3.3 Розробка функціоналу застосунку для підтримки і управління системи управління складом

Мета розробки застосунку для підтримки і управління системи управління складом є скорочення часу та зусиль на впровадження системи управління складом з технічної сторони за рахунок автоматичного порівняння налаштувань системи управління складом. Отже, функції, які має виконувати застосунок є наступним:

- підключення до серверу системи управління складом;
- підключення до потрібної бази даних – складів;
- порівняння налаштувань двох складів;
- виведення результатів порівняння – ідентичні об'єкти, відсутні об'єкти та об'єкти, що мають розбіжність в полях;

Для отримання налаштувань з бази даних було використано функцію SELECT. Для порівняння налаштувань були використані оператори WHERE, AND, OR, NOT IN.

Приклад запиту і результат його виконання зображений на рис. 3.1 і рис. 3.2 відповідно.

```

1 SELECT * FROM `DB WMS2`.`Zone`
2 WHERE Zone NOT IN (SELECT Zone FROM `DB WMS`.`Zone`);

```

Рисунок 3.1 – Запит на виведення інформації про те, яка зона відсутня у базі даних DB WMS порівняно з базою даних DB WMS2

+ Параметры

	Zone	Description	Building	PutawaySequence	PickSequence
☐   	RACK01	Racking01	B1	1	2

☐ Отметить все С отмеченными:    

Рисунок 3.2 – Приклад результату запиту

Для того щоб перевірити коректність введення бази даних, використовується програмний код РНР. Приклад валідації і результат роботи, якщо поле було залишено пустим, наведено на рис. 3.3 і рис. 3.4 відповідно.

```

// Check if Data Source 1 is empty
if(empty(trim($_POST["db1"]))) {
    if(empty(trim($_POST["whs1"]))) {
        $db1_err = "Please enter Data Base 1";
        $whs1_err = "Please enter Warehouse 1";
    } else {
        $db1 = trim($_POST["db1"]);
        $whs1 = trim($_POST["whs1"]);
    }
}

```

Рисунок 3.3 – Перевірка чи було введено дані в джерело даних 1

The screenshot shows a light blue rectangular box with the title "Data Source 1". Inside the box, there are two input fields. The first field is labeled "Database 1" and has a red border with a red exclamation mark icon in the top right corner. Below it, the text "Please enter Data Base 1" is displayed in red. The second field is labeled "Warehouse 1" and also has a red border with a red exclamation mark icon in the top right corner. Below it, the text "Please enter Warehouse 1" is displayed in red.

Рисунок 3.4 – Результат перевірки, якщо не було введено даних в джерело даних 1

Після того як користувач ввів дані в джерела даних 1 та 2 і натиснув кнопку «Порівняти» система виконує функцію порівняння введених джерел даних і виводить результати. Результати порівняння діляться на три категорії: ідентичні об’єкти, відсутні об’єкти та об’єкти, що мають розбіжність в полях.

Також після порівняння користувач має змогу переглянути розбіжності через застосунок.

3.4 Розробка інтерфейсу застосунку для підтримки і управління системи управління складом

Для полегшення інтерпретації отриманих результатів роботи програми та покращення користувацького досвіду роботи з програмою, важливо розробити зручний та інтуїтивно зрозумілий інтерфейс. Це значно скоротить час для вивчення застосунку і швидко зробить з користувача-початківця користувача, що досконало володіє функціоналом застосунку.

Навігація по веб-застосунку має бути зрозумілою для користувача. Проте інтерфейс не має бути перенавантаженим. Користувач має бачити лише легко інтерпритований результат роботи та виконувати дії, які очікує програма.

Для реалізації інтерфейсу було обрано мову HTML та CSS.

Рисунок 3.5 ілюструє форму входу для підключення користувача до серверу та початку роботи з програмою.

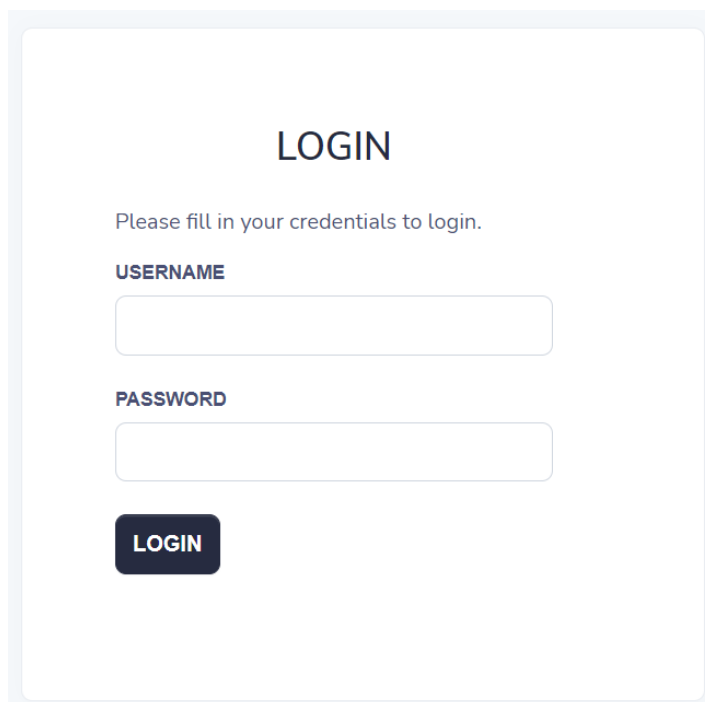


Рисунок 3.5 – Форма для входу в систему

На рис. 3.6 зображена головна вкладка меню програми, де користувач має ввести два джерела даних – для складу 1 та складу 2 і натиснути кнопку «Compare» для їх порівняння. Якщо дані не було внесено або у внесених даних є помилка, буде відображатись відповідна помилка джерел даних і користувач повинен буде внести свої дані знову.

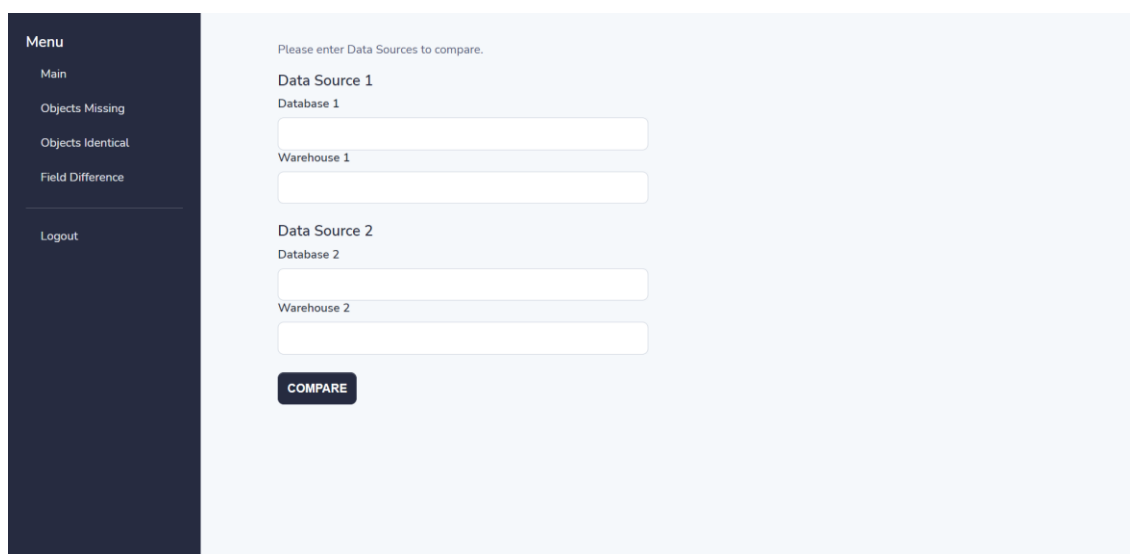


Рисунок 3.6 – Головна вкладка меню

Користувач має змогу переключатись між вкладками меню для перегляду результатів порівняння. Також користувач має змогу вийти з застосунку та завершити сесію, натиснувши кнопку «Logout».

Результати порівняння виводяться на відповідні вкладки меню: ідентичні об'єкти (вкладка «Objects Identical»), відсутні об'єкти (вкладка «Objects Missing») та об'єкти, що мають розбіжність в полях (вкладка «Field Difference»). Відсутні об'єкти – об'єкти які відсутні у відповідних таблицях на першій чи другій базі даних. Користувач має можливість обрати, у якому джерелі даних переглянути відсутні об'єкти. Сторінка, що відображає відсутні об'єкти зображена на рис. 3.7.

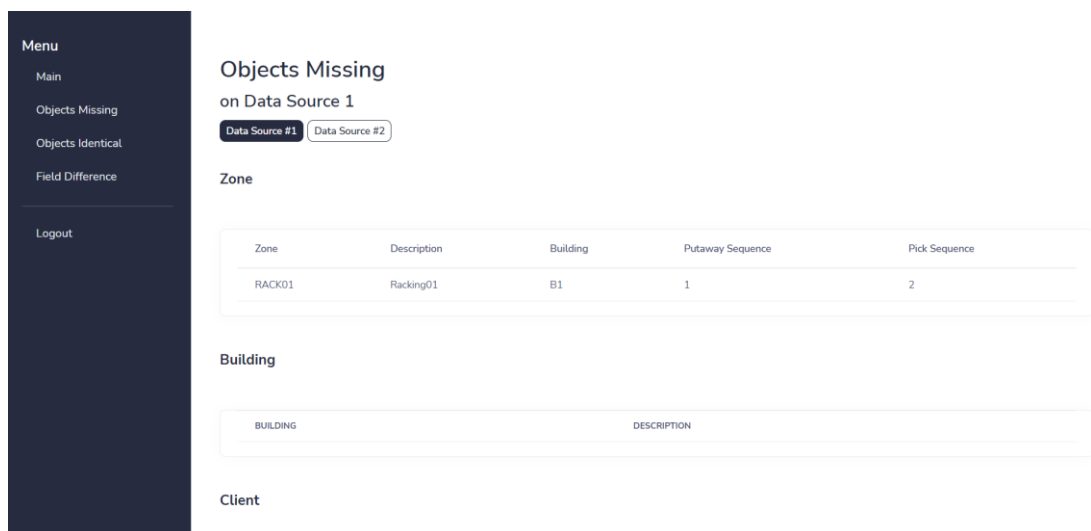


Рисунок 3.7 – Вкладка «Objects Missing»

Ідентичні об'єкти відображають дані з відповідно зазначених таблиць, які є однаковими в обох базах даних – складах. Сторінка, що відображає ідентичні об'єкти зображена на рис. 3.8.

Menu
Main
Objects Missing
Objects Identical
Field Difference
Logout

Objects Identical

on both Data Sources

Zones

Zone	Description	Building	Putaway Sequence	Pick Sequence
INBSTAGE	Inbound Staging	B1	0	0
OUTSTAGE	Outbound Staging	B1	0	0
PICK	Picking Zone	B1	0	1

Building

Building	Warehouse
B1	NT
B2	NT

Рисунок 3.8 – Вкладка «Objects Identical»

Об’єкти з розбіжностями присутні в обох базах даних, проте мають поле або поля, які відрізняються. Сторінка, що відображає об’єкти з розбіжностями зображена на рис. 3.9.

Menu
Main
Objects Missing
Objects Identical
Field Difference
Logout

Field Differences

Zones

Data Source 1
Data Source 2

Zone	Description	Building	Putaway Sequence	Pick Sequence	Zone	Description	Building	Putaway Sequence	Pick Sequence
RACK	Racking	B1	1	2	RACK	Racking 2	B1	1	2

Buildings

Data Source 1
Data Source 2

Building	Warehouse	Building	Warehouse

Clients

Data Source 1
Data Source 2

Client	Warehouse	Client	Warehouse

Рисунок 3.9 – Вкладка «Field Difference»

3.5 Результати використання застосунку для підтримки і управління комплексом управління складом

Для дослідження ефективності програмного продукту були відібрані дані на основі даних логістичної компанії, яка займається впровадженням та підтримкою комплексів управління складськими процесами. Було відібрано 5 проектів, які були реалізовані в період 2020-2022 років. Також представлені дані щодо витрат на обслуговування системи управління складськими процесами за 2020-2022 роки.

Аналіз існуючих підходів до оцінки логістичних витрат показує, що переважна більшість дослідників [3, 26, 27], визначаючи цей вид витрат, зосереджують свою увагу, здебільшого, на базовій логістичній активності або на конкретному перелік бізнес-процесів, які відбуваються всередині підприємства.

У таблиці 3.2 наведено витрати на впровадження комплексу управління складськими процесами на п'ятих проектах різних за сферою діяльності та обсягом зберігання без використання технічних механізмів управління.

Таблиця 3.2 – Витрати на впровадження комплексу управління складом без використання технічних механізмів управління

№ проекту	Початок проекту	Кінець проекту	Загальна кількість робочих днів	Вартість послуг за робочий день (дол.)	Вартість комплексу управління складом (дол./місяць)	Загальна вартість проекту (дол.)
1	29.09.2020	01.04.2021	128	1000	2000	136533
2	16.11.2020	26.07.2021	172	1000	2000	183352

3	01.02.2021	15.07.2021	113	1000	2000	120458
---	------------	------------	-----	------	------	--------

Продовження таблиці 3.2

4	23.02.2021	14.01.2022	223	1000	2000	237718
5	03.05.2021	05.01.2022	169	1000	2000	180154

Розглянемо кожен проект на етапі реалізації більш детально (Табл. 3.3).

Таблиця 3.3 – Витрати на етапі реалізації проекту без використання технічних механізмів управління

№ про ект у	Початок етапу	Кінець етапу	Загаль на кількіс ть робочи х днів	Кількіс ть годин на порівн яння середо вищ	Вартість послуг за робочий день (дол.)	Вартість комплек су управлін ня складом (д ол./міся ць)	Грошо ві витрат и на порівн яння середо вищ (дол.)
1	23.10.2020	08.03.2021	93	58	1000	2000	7728,5
2	11.03.2021	22.06.2021	70	50	1000	2000	6662,5
3	12.06.2021	28.06.2021	9	8	1000	2000	1066
4	08.04.2021	30.07.2021	77	56	1000	2000	7462
5	09.06.2021	06.12.2021	125	760	1000	2000	101270

В розробленому програмному продукті порівняння середовищ займає від 1 до 10 хвилин, в залежності від кількості даних з налаштувань системи управління складом. Витрати на етапі реалізації проекту з використанням технічного механізму управління представлені у таблиці 3.4.

Таблиця 3.4 – Витрати на етапі реалізації проекту з використанням технічних механізмів управління

№ проекту	Загальна кількість днів (робочих)	Кількість годин на порівняння середовищ	Вартість послуг (дол/робочий день)	Вартість комплексу управління складом (дол/місяць)	Грошові витрати на порівняння середовищ (дол)
1	86	0,16	1000	2000	21,32
2	64	0,16	1000	2000	21,32
3	8	0,16	1000	2000	21,32
4	70	0,16	1000	2000	21,32
5	95	0,16	1000	2000	21,32

Значення 10 хвилин було прийнято як максимальна кількість часу для порівняння середовищ. У результаті в середньому грошові витрати на час, витрачений на порівняння середовищ, можна зменшити більш ніж у 100 разів.

У таблиці 3.5 наведено порівняння витрат на впровадження комплексу управління складом без та з використанням технічних механізмів управління.

Таблиця 3.5 – Порівняння витрат на впровадження комплексу управління складом

	Без використання технічних механізмів управління	З використанням технічних механізмів управління
Середні витрати робочих днів на впровадження комплексу управління складом	161	151

Продовження таблиці 3.5

Середні грошові витрати на впровадження комплексу управління складом (дол.)	180421	146805
---	--------	--------

У результаті використання розробленого програмного продукту на етапі реалізації витрати часу зменшились на 6,2%, а грошові витрати на 18,6%.

У таблиці 3.6 розглядаються витрати на підтримку комплексу управління складом за 2020-2022 роки.

Таблиця 3.6 – Витрати на підтримку комплексу управління складом без використання технічних механізмів управління

Рік	Тип проблеми	Кількість запитів	Мінімальний час на вирішення 1 проблеми (год)	Середній час на вирішення 1 проблеми (год)	Максимальний час на вирішення 1 проблеми (год)	Вартість послуг зовнішньої підтримки (дол/год)
2020	Налаштування	35	0,8	312	2736	1000
2021	Налаштування	44	0,25	192,75	1210	1000
2022	Налаштування	38	1.23	446	3718,5	1000

Розглядалися лише запити з типом проблеми «Налаштування», що відноситься до випадків, коли змінилася конфігурація системи і процес перестав працювати коректно. У середньому за три роки середній час вирішення 1 задачі становить 316,9 години. Також варто зазначити, що час вирішення запиту залежить від кількості ескалацій проблеми на вищі рівні підтримки.

У таких випадках також можна використовувати розроблений програмний продукт для порівняння середовищ і визначення конфігурації, яка змінилася, і скорочення часу на пошук проблеми. У цьому випадку стороння підтримка може бути не залучена до вирішення проблеми. Таким чином, грошова вартість послуг зовнішньої підтримки для задачі типу «Налаштування» буде дорівнювати нулю. Що призведе до загального зниження грошової вартості зовнішньої підтримки комплексу управління складом. Порівняння середовищ усе ще займає від 1 до 10 хвилин. Зміна відповідної конфігурації та перевірка коректності процесу займе в середньому 1 робочий день або 8 годин. Таким чином, витрати часу на обслуговування комплексу управління складським господарством скоротяться в середньому в 40 разів.

3.6 Висновки до розділу

В цьому розділі було розглянуто та проаналізовано обрані засоби та механізми для розробки застосунку для підтримки і управління системи управління складом. Також було розглянуто основні ситуації, в яких має використовуватись програмний продукт та розроблено функціонал та інтерфейс програми. Було проаналізовано ефективність використання програмного продукту.

В результаті отримано мінімально життєздатний програмний продукт, готовий до використання у поставлених задачах.

Програму реалізовано за допомогою SQL, PHP, HTML та CSS мов на сервері OpenServer.

В майбутньому в програмі може бути розширений функціонал в залежності від вимог клієнтів. Наприклад, створення фільтру, синхронізація налаштувань в рамках застосунку та ін.

РОЗДІЛ 4 РОЗРОБКА ВЛАСНОГО СТАРТАП ПРОЕКТУ

Так як в сучасному світі практично кожне виробниче або торгове підприємство має склад для зберігання власної продукції або сировини, в підприємств наявна потреба у організації складу та автоматизації складських процесів. Впровадження системи управління складом займає зазвичай декілька місяців, і вимагає великої кількості фінансових і робочих затрат. Є багато методів та механізмів ведення проектів, які організовують і скорочують час на впровадження системи управління складом, проте практично немає системи, яка б скоротила час та зусилля на впровадження системи управління складом з технічної сторони для більшості команд для впровадження системи управління складом та суперюзерів системи.

Це означає, що проблема, яку вирішує стартап, являється актуальною і може стати успішною на ринку та стати основним вибором для переважної кількості цільової аудиторії.

4.1 План розробки стартапу та масштабування його на ринок

Наведемо план розробки стартапу та виведення його на ринок.

Спочатку треба провести маркетинговий аналіз, який включає в себе:

- конкурентний аналіз, щоб зрозуміти, якими методами вирішення проблем вже користуються люди;
- формування ідеї самого проекту та виділення цільової аудиторії;
- розробити стратегію виведення товару на ринок, базуючись на аналізі ринкового середовища.

Наступним кроком являється організація самого стартапу. На цьому етапі мають бути:

- складений весь план та побудований таймлайн розробки та запуску продукту;
- запланований обсяг виробництва та оцінений потенційний обсяг ресурсу, який буде потрібен для виконання плану;
- розраховані витрати, необхідні для реалізації проекту, та витрати на запуск проекту.

Далі необхідно виконати фінансово-економічний аналіз та оцінити ризики стартап-проекту, в межах якого:

- визначити обсяг інвестиційних втрат;
- розрахувати основні фінансово-економічні показники проекту (собівартість, ціну продукту/послуги, податковий збір та чистий прибуток) та визначити показники інвестиційної привабливості проекту (рентабельність продажів, період окупності проекту);
- визначити основні ризики проекту та способи для їх запобігання.

Фінальним кроком являється розробка заходів з комерціалізації продукту. Цей крок являється важливим для масштабування та збільшення розмірів продукту.

Для того, щоб залучити інвесторів та знайти різні способи фінансування проекту, необхідно:

- провести дослідження на предмет інтересів потенційних інвесторів та бізнесів;
- скласти інвестиційну пропозицію, яка включає в себе як опис самого продукту та його теперішні розміри, так і можливі шляхи розширення та розвитку;
- обрати канали комунікації із потенційно зацікавленими персонами.

Далі наведемо результати виконання кожного з описаних кроків.

4.2 Опис ідеї стартап-проекту

Стартап-проект полягає у вирішенні проблеми трудомісткості впровадження системи управління складом. Суть продукту полягає у функціональному рішенні для управління налаштуваннями системи управління складом, що дозволяє швидко помічати різницю в налаштуваннях та усувати неполадки. Це скорочує час і зусилля на впровадження та підтримку операцій в системі управління складом. Застосунок можна використовувати для перевірки налаштувань в межах одного або декількох середовищ (складів).

У таблиці 4.1 наведена інформаційна карта стартапу.

Таблиця 4.1 – Інформаційна карта стартап-проекту

Назва проекту	WMS Compare
Автори проекту	Торліна Надія Михайлівна
Коротка анотація	Додаток буде порівнювати налаштування двох середовищ в одній системі управління складом, що дозволить швидко помічати різницю в налаштуваннях та усувати неполадки і підвищить якість виконання та ведення операцій в системі управління складом.
Термін реалізації проекту	12 місяців
Необхідні ресурси	Приміщення з комп'ютерами, доступом до Інтернету, доступ до електромережі.

Продовження таблиці 4.1

	Програмне забезпечення для розробки, сервер для даних, антивірусні програми. Фінансові кошти на оплату заробітної плати виконавцям на термін 12 місяців, а також на такі витрати як: оренда приміщення, комунальні послуги, оренда серверу тощо.
Опис проблеми, яку вирішує проект	Продукт вирішує задачу трудомісткості впровадження системи управління складом.
Головні цілі та завдання проекту	Метою проекту є створення веб-додатку для скорочення часу на впровадження системи управління складом та її підтримку, підвищення якості виконання та ведення операцій в системі управління складом.
Очікувані результати	Привернення логістичних компаній до нашого стартапу, автономна система для управління налаштуваннями системи управління складом.

4.3 Технологічний аудит ідеї проекту

Тепер можна розібрати ідею стартапу та провести конкурентний аналіз. У таблиці 4.2 наведений опис ідеї стартапу.

Таблиця 4.2 – Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Основна ідея: функціональне рішення для управління налаштуваннями системи управління складом	Порівняння налаштувань системи управління складом	Користувач зможе автоматично знаходити різницю в налаштуваннях
	Усунення невідповідностей налаштувань системи управління складом	Система буде оновлюватись і функціонал буде розширюватись

Далі проведемо порівняльний аналіз конкурентів проекту та наведемо результати у таблиці 4.3.

Таблиця 4.3 – Порівняльний аналіз конкурентів проекту

п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Власний проект	Вбудована функція порівняння БД MS Access	SQL Management Studio for MySQL	dbForge Data Compare for MySQL			
	Порівняння налаштувань двох баз даних	Надає доволі хороший результат	Свій власний алгоритм	Свій власний алгоритм	Свій власний алгоритм			+

Продовження таблиці 4.3

	Доступність по ціні	Безкоштовний при запуску MVP	Лише при наявності MS Office і бази даних в MS Access	Безкоштовний пробний період, далі – \$480 на рік	Безкоштовний пробний період, далі – \$169.95 на рік		+	
	Продукт, розроблений для системи управління складськими процесами	Так	Ні	Ні	Ні		+	

Далі аналізуємо реальність технічно здійснити ідею проекту (таблиця 4.4).

Таблиця 4.4 – Технологічна здійсненність продукту

п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
	Створення веб застосунку для управління налаштуваннями	Використання мови програмування Python	Наявні, необхідні допрацювання	Доступні

	системи управління складом	Використання мови	Наявні	Доступні
--	-------------------------------	----------------------	--------	----------

Продовження таблиці 4.4

		програмування PHP		
		Використання мови програмування JSP.Net	Наявні, необхідні допрацювання	Доступні
Обрана технологія реалізації ідеї проекту: PHP				

4.4 Аналіз ринкових можливостей запуску стартап-проекту

Далі проведемо попередній аналіз ринку для запуску стартап-проекту (таблиця 4.5).

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту

п/п	Показники ринку (найменування)	Характеристика
	Кількість головних гравців, од	3
	Загальний обсяг продаж, грн/ум.од	6000
	Динаміка ринку (якісна оцінка)	Позитивна, зростає

	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
	Специфічні вимоги до стандартизації та сертифікації	Відсутні

Продовження таблиці 4.5

	Середня норма рентабельності в галузі (або по ринку), %	12%
--	---	-----

Тепер проведемо характеристику потенційних клієнтів, які можуть бути зацікавлені

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

п/п	Потреби, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Скорочення часу і зусиль на впровадження та підтримку операцій в системі управління складом	Команди з впровадженн я систем управління складом, менеджери з проектів, клієнти систем управління складом	Цікавить функціонал та ефективність використання загалом	Стабільна робота
	Усунення	Користувачі,	Особлива потреба	Простота

	неполадок	команди з	у можливості	використання,
--	-----------	-----------	--------------	---------------

Продовження таблиці 4.6

		впровадженн я систем управління складом	виявити та усунути неполадку за допомогою застосунку	зрозумілий результат
	Синхронізація налаштувань	Користувачі, команди з впровадженн я систем управління складом	Необхідність у синхронізації налаштувань двох складів	Можливість працювати з великою кількістю налаштувань

Обрахуємо фактори загроз (таблиця 4.7) та можливостей (таблиця 4.8). Проаналізуємо загрози, щоб зрозуміти можливі перешкоди при запуску продукту на ринок. Фактори можливостей же треба обрахувати, щоб знати усі сприятливі умови та по можливості ними скористатися.

Таблиця 4.7 – Фактори загроз

п/п	Фактор	Зміст загрози	Можлива реакція компанії
	Конкуренція	Хоча ринок є відкритим і неосвоїним, на ньому вже є кілька великих гравців,	Знайти точки додаткової цінності для користувача

		які відомі на ринку, які вже мають свою цільову групу покупців	
--	--	--	--

Продовження таблиці 4.7

	Ціна збуту	Конкуренти можуть коштувати менше через нижчу якість	Сфокусуватися на якості роботи застосунку та продумати маркетингову стратегію
	Якість аналізу	Через комплексність задачі, система може мати проблеми на певних нових налаштуваннях, так наприклад деякі налаштування можуть визначатися не в базі даних, а в коді системи управління складом	Мати достатній штат і ресурси, для розвитку застосунку

Таблиця 4.8 – Фактори можливостей

п/п	Фактор	Зміст можливості	Можлива реакція компанії
-----	--------	------------------	--------------------------

	Універсальні сь	Продукт не залежить від конкретної системи управління складом	Зробити акцент при маркетингу, продовжувати розвиток як окремого продукту
--	--------------------	--	--

Продовження таблиці 4.8

	Простота у використанні	Від користувача треба лише підключення до серверу і ввести потрібний склад	Реалізувати зручний інтерфейс
	Якість та гарантії	Надавати найбільш якісні послуги та сервіси	Надавати усю необхідну технічну підтримку
	Безкоштовний сервіс при MVP	Максимально швидко набрати базу своїх клієнтів та заявити про себе на ринку	Розгорнути широкий маркетинг, а також активно боротися за клієнтів конкурентів

Далі розглянемо питання конкуренції, а саме визначимо її тип та рівень (таблиця 4.9).

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції: недосконала конкуренція	Представлено мало продуктів та експертів	Зробити максимальним збут застосунку
2. За рівнем конкурентної боротьби: міжнародний	Наявні проекти, розроблені та можуть бути	Розширити цільову аудиторію, розробити інтерфейс на різних мовах

Продовження таблиці 4.9

	доступні у всьому світі	
3. За галузевою ознакою: внутрішньогалузева	Можуть працювати в одній галузі	Розширити функціонал
4. Конкуренція за видами товарів: товарно-родова	Конкуренція з аналізами інших систем та експертів	Підтримувати та покращувати якість існуючих функцій
5. За характером конкурентних переваг: нецінова	Різні компанії пропонують застосунки, не прив'язані до галузі	Розробляти найбільш зрозумілий інтерфейс, орієнтуватись на галузь
6. За інтенсивністю: марочна	Вже представлені компанії із сильним брендом	Предметно створити комунікаційну стратегію для вибудови свого бренду

Далі необхідно виконаємо аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера (таблиця 4.10).

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товарозамінники
	Інші існуючі системи та продукти	Якість, ціни, кількість користувачів, капіталовкладення	Фактори сили постачальників	Контроль якості, порівняння цін	Сила бренду, якість, ціна, масштаби
Висновки	Конкуренція з невеликою інтенсивністю, а також підігрітий ринок	Можливості входження на ринок, нові потенційні конкуренти	Постачальники відсутні	Клієнти не диктують умови роботи на ринку	Товарозамінники відсутні

Маючи результати аналізу конкуренції (таблиця 4.10), характеристики ідеї стартап-проекту (таблиця 4.5), характеристики потенційних клієнтів і їх вимоги до продукту (таблиця 4.6) та фактори ринкового середовища (таблиці 4.7 і 4.8) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності (таблиця 4.11).

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
	Універсальність	Продукт не залежить від конкретної системи управління складом
	Простота у використанні	Від користувача треба лише підключення до серверу і ввести потрібний склад
	Якість та гарантії	Надавати найбільш якісні послуги та сервіси
	Безкоштовний сервіс при MVP	Максимально швидко набрати базу своїх клієнтів та заявити про себе на ринку

Тепер можна провести аналіз сильних та слабких сторін продукту (таблиця 4.12).

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін системи

п/п	№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
				-3	-2	-1	0	1	2	3
1		Універсальність	13						+	
2		Простота у використанні	18						+	

Продовження таблиці 4.12

3	Якість та гарантії	16		+					
4	Безкоштовний сервіс при MVP	18	+						

Далі проведемо SWOT-аналіз продукту (таблиця 4.13).

Таблиця 4.13 – SWOT-аналіз стартап-проекту

Сильні сторони Універсальність Простота у використанні, зрозумілий інтерфейс Регулярні оновлення застосунку Якість та гарантії Безкоштовний сервіс при MVP	Слабкі сторони Відсутність сильного бренду Не сформована база клієнтів Не підключені альтернативні канали маркетингу Вузький сектор ринку Охоплення не усього функціоналу одразу
Можливості Покращення системи Розширення функціоналу за рахунок ідей від користувачів Залучення інвесторів	Загрози Нові системи та експерти Збут Потенційна складність з охоплення великих масивів даних Потенційна складність з актуалізації налаштувань системи управління складом

Дякуючи проведенню SWOT-аналізу, ми змогли визначити сильні та слабкі сторони, можливості та загрози, пов'язані з конкуренцією та плануванням стартап-проекту. Далі спроектуємо альтернативну ринкову поведінку для інтеграції стартап-проекту на ринок та приблизний час реалізації системного комплексу, з урахуванням потенційних проектів, що можуть бути виведені на ринок та наведемо результати у таблиці 4.14.

Таблиця 4.14 – Альтернативи ринкового впровадження стартап проекту

п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	Вихід на ринок з нижчою якістю застосунку	70%	8 місяців
	Пропонувати одразу платне використання	50%	6 місяців
	Представлення користувачам системи без інтерфейсу	60%	5 місяців

У даному пункті був проведений детальний аналіз ринку та продукту. Також відп висновок, що існують дуже сприятливі умови для виходу продукту на ринок.

4.5 Розроблення ринкової стратегії стартап-проекту

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

п / п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит у межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Персональні користувачі	Середня	20%	Низька	Середня
	Великі бізнеси	Середня	30%	Середня	Середня
	Малі та середні бізнеси	Висока	30%	Низька	Середня
Які цільові групи обрано: 2, 3					

Маючи аналіз цільових груп, далі визначимо базову стратегію розвитку продукту (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії розвитку

п/п	Обрана альтернатива	Стратегія охоплення ринку	Ключові конкурентоспроми	Базова стратегія
-----	---------------------	---------------------------	--------------------------	------------------

	розвитку проекту		ожні позиції відповідно до обраної альтернативи	розвитку*
	2 та 3	Диференційованого маркетингу	Масштабування та максимізація	Оптимальних витрат

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 4.17, 4.18).

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем » на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
Ні	Так	Так, порівняння і синхронізація даних	Виклику лідера

Таблиця 4.18 – Визначення стратегії позиціонування

Вимоги до товару цільової	Базова стратегія розвитку	Ключові конкурентоспром	Вибір асоціацій, які мають
------------------------------	------------------------------	----------------------------	-------------------------------

аудиторії		ожні позиції власного стартап- проекту	сформувати комплексну позицію власного проекту (три ключових)
Універсальність Простота у використанні Якість результатів	Оптимальних витрат	Універсальність Простота у використанні Якість та гарантії Безкоштовне використання при MVP	Система, яка скорочує час і зусилля на впровадження та підтримку операцій в системі управління складом Система з простим інтерфейсом

4.6 Розроблення маркетингової програми стартап-проекту

Після проведеного комплексного аналізу, можемо повноцінно описати ключові переваги концепції потенційного товару (таблиця 4.19) та побудувати концепцію маркетингових комунікацій (таблиця 4.20).

Таблиця 4.19 – Ключові переваги концепції потенційного товару

	Потреба	Вигода, яку	Ключові переваги перед
--	---------	-------------	------------------------

п/п		пропонує товар	конкурентами (існуючі або такі, що потрібно створити)
	Управління налаштуваннями системи управління складом	Порівняння і синхронізація налаштувань системи управління складом	Постійне вдосконалення функціоналу, який зможе аналізувати більше налаштувань
	Універсальність	Система не залежить конкретної бази даних зі складом	Такого виду систему може використовувати будь який користувач
	Простий інтерфейс	Система дуже проста у використанні	Система із інтуїтивно зрозумілим інтерфейсом, який вимагає всього лише підключення до серверу з базою даних та введення назви складу

Таблиця 4.20 – Концепція маркетингових комунікацій

п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Пошук спеціалізованих систем	b2b продажі Зв'язок через теплі контакти Публікація в спеціалізованих виданнях, журналах Участь в професійних ярмарках	Точність Якість Універсальність	Поєднати повідомлення про те, що це якісна система, яка є незалежною	Реклама на цільову аудиторію
	Пошук доступного продукту	Рекламні банери в Інтернеті, форуми, реклами від постачальників систем управління складом	Простота Безкоштовне використання MVP	Вселити довіру у бренд та продукт	Реклама у постачальників в Рекламу на цільову аудиторію

4.7 Висновки до розділу

Даний розділ був присвячений дослідженню стартап-проекту. В якості такого була представлена механізм управління та підтримки системи управління складом: система для управління налаштуваннями системи управління складом.

У рамках розділу було досліджено розробку стратегій виходу на ринок та маркетинг-стратегії для цього. Зокрема, даний ринок являється сприятливим з невеликою кількістю представлених компаній конкурентів. Оскільки вони не орієнтовані на галузь логістики та системи управління складом, а запропонована система є доволі універсальною та доступною, то у стартап-проекті є всі шанси стати монополістами на ринку.

Також були опрацьовані сильні та слабкі сторони проекту, SWOT аналіз, аналіз конкурентів та цільової аудиторії. На основі всіх досліджень був сформований концепт маркетингової стратегії для обраних цільових аудиторій.

ВИСНОВКИ

В даній магістерській роботі було проведено огляд існуючих методів та механізмів для управління і підтримки системи управління складом, зокрема при її впровадженні на підприємство. Було розроблено програмний продукт, що вирішує задачу по управлінню і підтримки системи управління складом.

В результаті проведеної роботи можна сказати, що використання програмного продукту скорочує час і зусилля при впровадженні системи управління складом за рахунок часткової автоматизації процесу конфігурації системи за вирішення проблем під час налаштування процесів.

Запропонований в роботі застосунок для управління і підтримки системи управління складом є мінімально життєздатним продуктом, що забезпечує базовий процес порівняння налаштувань системи управління складом. Проте він повністю функціональний та легкий у використанні за рахунок простого та інтуїтивно зрозумілого інтерфейсу.

В ході розробки програми для управління і підтримки системи управління складом було в якості системи управління складом було взято систему з бакалаврської роботи з потрібними зв'язками. Було також описано функціонал та приклад роботи з веб додатком. Розробка програмного продукту відбувалася на мові програмування PHP, мові HTML та CSS, на базі системи управління базами даних MySQL, а також серверу OpenServer.

Застосування даного застосунку можливе в рамках невеликої системи управління складом без доробки або з мінімальними змінами через індивідуальність кожної системи управління складом.

СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дудар Т. Г. Основи логістики: навч. посіб. Тернопіль: Економічна думка, 2006. 163 с.
2. Чухрай Н. Логістичне обслуговування: підручник для вузів. М-во освіти і науки України. Л.: Львів. політехніка, 2006, 292 с.
3. Шевців Л. Ю. Логістичні витрати підприємства: формування та оцінювання: [монографія]. ЛНУ «Львівська політехніка», 2011, 244 с.
4. Warehouse management system
URL: https://en.wikipedia.org/wiki/Warehouse_management_system (Last accessed 20.06.2022)
5. 7 Steps Towards a Successful WMS Implementation. URL: <https://www.linkedin.com/pulse/7-steps-towards-successful-wms-implementation-marcil-msc-> (Last accessed 21.06.2022)
6. IT Support Levels Clearly Explained: L1, L2, L3 & More. URL: <https://www.bmc.com/blogs/support-levels-level-1-level-2-level-3/> (Last accessed 21.06.2022)
7. Мамонова Ю.С., Русина Ю.В. Проблемы внедрения автоматизированной системы управления складом. 2013. URL: <https://cyberleninka.ru/article/n/problemy-vnedreniya-avtomatizirovannoy-sistemy-upravleniya-skladom> (Дата звернення: 20.06.2022)
8. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика = Database Systems: A Practical Approach to Design, Implementation, and Management. — 3-е изд. — М.: «Вильямс», 2003. — 340 с.
9. C. J. Date. What First Normal Form Really Means /C. J. Date. Date on database: Writings 2000—2006, Apress, 2006. — 45 p.
10. Базові знання з нормалізації баз даних Майк Чеппл (About.com) (<https://www.lifewire.com/database-normalization-basics-1019735>)

11. Kent W. «A Simple Guide to Five Normal Forms in Relational Database Theory», Communications of the ACM 26 (2), Feb. 1983, 120—125 pp.
12. К. Дж. Дейт. Введение в системы баз данных = Introduction to Database Systems. — 8-е изд. — М.: «Вильямс», 2006. — 437 с.
13. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика = Database Systems: A Practical Approach to Design, Implementation, and Management. — 3-е изд. — М.: «Вильямс», 2003. — 340 с.
14. Мулеса О.Ю. Основи мови запитів SQL. — Ужгород, 2015. — 48 с.
15. Natesan Andiyappillai. Factors Influencing the Successful Implementation of the Warehouse Management System (WMS). 2020. URL: <https://www.ijais.org/archives/volume12/number35/andiyappillai-2020-ijais-451896.pdf> (Last accessed 21.06.2022)
16. A. James Baroody, David J. Dewitt. An Object-Oriented Approach to Database System Implementation. URL: <https://dl.acm.org/doi/pdf/10.1145/319628.319645> (Last accessed 22.06.2022)
17. Wajid Ali, Muhammad Usman Shafique, Muhammad Arslan Majeed, Ali Raza. Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. 2019. URL: https://www.researchgate.net/profile/Wajid-Ali-30/publication/336686999_Comparison_between_SQL_and_NoSQL_Databases_and_Their_Relationship_with_Big_Data_Analytics/links/5dada9e1a6fdccc99d927b31/Comparison-between-SQL-and-NoSQL-Databases-and-Their-Relationship-with-Big-Data-Analytics.pdf?sg%5B0%5D=started_experiment_milestone&origin=journalDetail (Last accessed 22.06.2022)
18. Tim King, George Reese, Randy Yarger, Hugh E. Williams, Managing & Using MySQL: Open Source SQL Databases for Managing Information & Web Sites. USA: O'Reilly Media. 2002. 444 p.

19. Pichler, R., Wesley A., Agile Product Management with Scrum: Creating Products That Customers Love. 2010. URL: <http://www.ultering.com/itstuff/wp-content/uploads/2018/08/SE-Scrum-Agile-Product-Management-with-Scrum-Roman-Pichler-BIN532.pdf> (Last accessed 10.11.2022)
20. Swisher W. P., Implementing Scrumban. 2014. URL: https://switchingtoscrum.files.wordpress.com/2013/12/implementing-scrumban_v1-32.pdf (Last accessed 10.11.2022)
21. Brechner, E., Agile Project Management with Kanban. 2015. URL: <https://ptgmedia.pearsoncmg.com/images/9780735698956/samplepages/9780735698956.pdf> (Last accessed 10.11.2022)
22. Bieliūnas E., Scrum-ban for Project Management. 2014. URL: <https://eylean.com/wp-content/uploads/2019/12/Scruban-for-Project-Management.pdf> (Last accessed 10.11.2022)
23. Sutherland J., Schwaber, K. The Scrum Guide. 2013. URL: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf> (Last accessed 10.11.2022)
24. Гончаров А. Б., Олейникова Н. М. Економічне управління підприємством : конспект лекцій – Вид. ХНЕУ, 2009. – 366 с.
25. Економіка підприємства: підручник / ред. С.Ф. Покропивного. вид. 2-ге, переробл. та доп. – К. : КНЕУ, 2011. – 528 с.
26. Сумець О. М., Логістичні витрати підприємства: теоретичний аспект, монографія. Харків: КП «Міськдрук», 2013. 224 с.
27. Лифар В. В., Комерційна логістика та методика розрахунку логістичних витрат. Вісник Національного Університету «Львівська політехніка». 2011. Вип. 416. С. 293 – 297.

ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ

Модуль connection.php

```
<?php
$connect = mysqli_connect('localhost', 'root', "");
$select_db = mysqli_select_db($connect,'DB WMS');
$select_db2 = mysqli_select_db($connect,'DB WMS2');
?>
```

Модуль main.php

```
<?php
session_start();
require('connection.php'); // подключаем скрипт

// Define Data Sources: db and whs, and initialize with empty values
$db1 = $db2 = "";
$db1_err = $db2_err = "";
$whs1 = $whs2 = "";
$whs1_err = $whs2_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Check if Data Source 1 is empty
    if(empty(trim($_POST["db1"]))) {

        $db1_err = "Please enter Data Base 1";

    } else {
        $db1 = trim($_POST["db1"]);
    }
}
```

```

// Check if datasource2 is empty
if(empty(trim($_POST["db2"]))) {

    $db2_err = "Please enter Data Base 2";

} else {
    $db2 = trim($_POST["db2"]);

}

// Store data in session variables
$_SESSION["db1"] = $db1;
$_SESSION["whs1"] = $whs1;
$_SESSION["db2"] = $db2;
$_SESSION["whs2"] = $whs2;

(випущено)

// Close connection
mysqli_close($connect);
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
<link rel="apple-touch-icon" sizes="120x120" href="assets/img/favicon/apple-touch-ic

```

```

<link          rel="icon"          type="image/png"          sizes="32x32"
href="assets/img/favicon/favicon-32x32.png">
<link          rel="icon"          type="image/png"          sizes="16x16"
href="assets/img/favicon/favicon-16x16.png">
<link rel="manifest" href="assets/img/favicon/site.webmanifest">
<link      rel="mask-icon"      href="assets/img/favicon/safari-pinned-tab.svg"
color="#ffffff">
<meta      name="msapplication-TileColor"      content="#ffffff"><meta
name="theme-color" content="#ffffff">
<link      type="text/css"      href="vendor/@fortawesome/fontawesome-
free/css/all.min.css" rel="stylesheet">
<!-- Notyf -->
<link type="text/css" href="vendor/notyf/notyf.min.css" rel="stylesheet">
<!-- Volt CSS -->
<link type="text/css" href="css/volt.css" rel="stylesheet">
<meta  http-equiv="Content-Type"  content="width=device-width,  initial-
scale=1; text/html; charset=utf-8" >
<!-- Primary Meta Tags -->
<title>WMS</title>
<link      rel="canonical"      href="https://themesberg.com/product/admin-
dashboard/volt-bootstrap-5-dashboard">
<!--      Favicon      --><link      rel="apple-touch-icon"      sizes="120x120"
href="assets/img/favicon/apple-touch-icon.png">
<link          rel="icon"          type="image/png"          sizes="32x32"
href="assets/img/favicon/favicon-32x32.png">
<link          rel="icon"          type="image/png"          sizes="16x16"
href="assets/img/favicon/favicon-16x16.png">
<link rel="manifest" href="assets/img/favicon/site.webmanifest">
<link      rel="mask-icon"      href="assets/img/favicon/safari-pinned-tab.svg"
color="#ffffff">

```

```

    <meta      name="msapplication-TileColor"      content="#ffffff"><meta
name="theme-color" content="#ffffff">

```

```

    <!--          <link                      rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">-->

```

```

    <style>

```

```

        body{ font: 14px sans-serif; }

```

```

        .wrapper{ width: 360px; padding: 20px; }

```

```

    </style>

```

```

</head>

```

```

<body class="bg-soft">

```

```

<!-- Sidebar/menu -->

```

```

    <nav id="sidebarMenu" class="sidebar d-md-block bg-primary text-white
collapse" data-simplebar><br>

```

```

        <div class="sidebar-inner px-4 pt-3"><div class="user-card d-flex d-md-none
align-items-center justify-content-between justify-content-md-center pb-4">

```

```

            <div class="d-flex align-items-center">

```

```

                <div class="d-block">

```

```

                    <a href="pages/examples/sign-in.html" class="btn btn-secondary text-dark btn-
xs">

```

```

                        <span class="mr-2"><span class="fas fa-sign-out-alt"></span></span>Sign
Out</a></div></div>

```

```

                <div class="collapse-close d-md-none">

```

```

                    <a href="#sidebarMenu" class="fas fa-times" data-toggle="collapse" data-
target="#sidebarMenu" aria-controls="sidebarMenu" aria-expanded="true" aria-
label="Toggle navigation"></a>

```

```

                </div></div>

```

```

            <ul class="nav flex-column">

```

```

<li class="nav-item">
<h5>Menu</h5>
<a href="main.php" class="nav-link"><span class="sidebar-icon"></span>
<span>Main</span></a></li>
<li class="nav-item"><a href="objectsmissingpage.php" class="nav-
link"><span class="sidebar-icon"></span> <span>Objects Missing</span></a></li>
<li class="nav-item"><a href="objectsidentical.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Objects Identical</span></a></li>
<li class="nav-item"><a href="fielddifference.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Field Difference</span></a></li>
<hr>
<li class="nav-item"><a href="logout.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Logout</span></a></li>
</nav>

<!-- !PAGE CONTENT! -->

<div class="w3-main" style="margin-left:300px;margin-top:43px;">
<div class="container">
<div class="signin-inner my-3 my-lg-0 w-100 fmxxw-500">

<p>Please enter Data Sources to compare.</p>

<?php
if(!empty($login_err)){
    echo '<div class="alert alert-danger">' . $login_err . '</div>';
}
?>

<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);
?>" method="post">

```

```

        <div class="form-group mb-4">
            <h5>Data Source 1</h5>
            <h6>Database 1</h6>
            <input type="text" name="username" class="form-control" <?php
echo (!empty($db1_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $db1; ?>">
            <span class="invalid-feedback"><?php echo $db1_err; ?></span><h6>Ware
            <input type="text" class="form-control" <?php echo (!empty($whs1_err)) ? 'is-invalid'
            <span class="invalid-feedback"><?php echo $whs1_err; ?></span>
        </div>
        <div class="form-group mb-4">
            <h5>Data Source 2</h5><h6>Database 2</h6>
            <input type="text" name="username" class="form-control" <?php
echo (!empty($db2_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $db2; ?>">
            <span class="invalid-feedback"><?php echo $db2_err;
?></span><h6>Warehouse 2</h6>
            <span class="invalid-feedback"><?php echo $whs2_err; ?></span>
        </div>
        <div class="form-group">
            <input type="submit" class="btn btn-primary" value="COMPARE">
        </div>
    </form></div></div>
</body>
</html>

```

Модуль objects.identical

```

<!DOCTYPE html>
<html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<!-- Primary Meta Tags -->
<title>WMS</title>

```

```

<link      rel="canonical"      href="https://themesberg.com/product/admin-
dashboard/volt-bootstrap-5-dashboard">

<!--      Favicon      --><link      rel="apple-touch-icon"      sizes="120x120"
href="assets/img/favicon/apple-touch-icon.png">

<link      rel="icon"            type="image/png"            sizes="32x32"
href="assets/img/favicon/favicon-32x32.png">

<link      rel="icon"            type="image/png"            sizes="16x16"
href="assets/img/favicon/favicon-16x16.png">

<link rel="manifest" href="assets/img/favicon/site.webmanifest">

<link      rel="mask-icon"      href="assets/img/favicon/safari-pinned-tab.svg"
color="#ffffff">

<meta      name="msapplication-TileColor"      content="#ffffff"><meta
name="theme-color" content="#ffffff">

<!--      Fontawesome      --><link      type="text/css"
href="vendor/@fontawesome/fontawesome-free/css/all.min.css" rel="stylesheet">

<!--      Notyf      --><link      type="text/css"      href="vendor/notyf/notyf.min.css"
rel="stylesheet">

<!-- Volt CSS --><link type="text/css" href="css/volt.css" rel="stylesheet">

<!-- Sidebar/menu -->

<nav id="sidebarMenu" class="sidebar d-md-block bg-primary text-white
collapse" data-simplebar><br>

  <div class="sidebar-inner px-4 pt-3"><div class="user-card d-flex d-md-none
align-items-center justify-content-between justify-content-md-center pb-4">

    <div class="d-flex align-items-center">

      <div class="d-block">

        <a href="pages/examples/sign-in.html" class="btn btn-secondary text-dark btn-
xs">

          <span class="mr-2"><span class="fas fa-sign-out-alt"></span></span>Sign
Out</a></div></div>

```

```

<div class="collapse-close d-md-none">
  <a href="#sidebarMenu" class="fas fa-times" data-toggle="collapse" data-
target="#sidebarMenu" aria-controls="sidebarMenu" aria-expanded="true" aria-
label="Toggle navigation"></a>
</div></div>
<ul class="nav flex-column">
  <li class="nav-item">
    <h5>Menu</h5>
    <a href="main.php" class="nav-link"><span class="sidebar-icon"></span>
<span>Main</span></a></li>
    <li class="nav-item"><a href="objectsmissingpage.php" class="nav-
link"><span class="sidebar-icon"></span> <span>Objects Missing</span></a></li>
    <li class="nav-item"><a href="objectsidentical.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Objects Identical</span></a></li>
    <li class="nav-item"><a href="fielddifference.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Field Difference</span></a></li>
    <hr>
    <li class="nav-item"><a href="logout.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Logout</span></a></li>
  </nav>

<!-- !PAGE CONTENT! -->
<div class="w3-main" style="margin-left:300px;margin-top:43px;">
  <!-- Header -->
  <header class="w3-container" style="padding-top:22px">
    <h2>Objects Identical</h2>
    <h4>on both Data Sources</h4>
  </header>
  <br></br>
  <h5><b>Zones</b></h5>

```

```

<div class="w3-row-padding w3-margin-bottom">
    <div class="card card-body border-light shadow-sm table-wrapper table-
responsive          pt-0"><table          class="table          table-
hover"><th>Zone</th><th>Description</th><th>Building</th><th>Putaway
Sequence</th><th>Pick Sequence</th>
    <?php
require('connection.php'); // подключаем скрипт

(випущено)

    // очищаем результат
    mysqli_free_result($result);

}
?>

<br></br>
<h5><b>Building</b></h5>
<div class="w3-row-padding w3-margin-bottom">
    <div class="card card-body border-light shadow-sm table-wrapper table-
responsive          pt-0"><table          class="table          table-
hover"><th>Building</th><th>Warehouse</th>
    <?php
require('connection.php');

$query = "SELECT
t1.Building, t1.Warehouse
FROM
`DB WMS`.`Building` t1,
`DB WMS2`.`Building` t2

```

WHERE

t1.Building = t2.Building AND t1.Warehouse = t2.Warehouse";

```
$result = mysqli_query($connect, $query) or die("Error" .
mysqli_error($connect));
```

```
if($result )
```

```
{
```

```
    $rows = mysqli_num_rows($result); // количество полученных строк
```

```
    echo "";
```

```
    for ($i = 0 ; $i < $rows ; ++$i)
```

```
    {
```

```
        $row = mysqli_fetch_row($result);
```

```
        echo "<tr>";
```

```
            for ($j = 0 ; $j < 2 ; ++$j) echo "<td>$row[$j]</td>";
```

```
        echo "</tr>";
```

```
    }
```

```
    echo "</table>";
```

```
echo "</table>";
```

```
echo "</div>";
```

```
echo "</div>";
```

```
    mysqli_free_result($result);
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

Модуль objectsmissingpage.php

```

<!DOCTYPE html>
<html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<!-- Primary Meta Tags -->
<title>WMS</title>
<link      rel="canonical"      href="https://themesberg.com/product/admin-
dashboard/volt-bootstrap-5-dashboard">
<!--      Favicon      --><link      rel="apple-touch-icon"      sizes="120x120"
href="assets/img/favicon/apple-touch-icon.png">
<link      rel="icon"           type="image/png"           sizes="32x32"
href="assets/img/favicon/favicon-32x32.png">
<link      rel="icon"           type="image/png"           sizes="16x16"
href="assets/img/favicon/favicon-16x16.png">
<link rel="manifest" href="assets/img/favicon/site.webmanifest">
<link      rel="mask-icon"      href="assets/img/favicon/safari-pinned-tab.svg"
color="#ffffff">
<meta      name="msapplication-TileColor"      content="#ffffff"><meta
name="theme-color" content="#ffffff">
<!--      Fontawesome      --><link      type="text/css"
href="vendor/@fontawesome/fontawesome-free/css/all.min.css" rel="stylesheet">
<!--      Notyf      --><link      type="text/css"      href="vendor/notyf/notyf.min.css"
rel="stylesheet">
<!-- Volt CSS --><link type="text/css" href="css/volt.css" rel="stylesheet">

<!-- Sidebar/menu -->
<nav id="sidebarMenu" class="sidebar d-md-block bg-primary text-white
collapse" data-simplebar><br>
<div class="sidebar-inner px-4 pt-3"><div class="user-card d-flex d-md-none
align-items-center justify-content-between justify-content-md-center pb-4">
<div class="d-flex align-items-center">

```

```

<div class="d-block">
  <a href="pages/examples/sign-in.html" class="btn btn-secondary text-dark btn-
xs">
    <span class="mr-2"><span class="fas fa-sign-out-alt"></span></span>Sign
Out</a></div></div>
  <div class="collapse-close d-md-none">
    <a href="#sidebarMenu" class="fas fa-times" data-toggle="collapse" data-
target="#sidebarMenu" aria-controls="sidebarMenu" aria-expanded="true" aria-
label="Toggle navigation"></a>
  </div></div>
  <ul class="nav flex-column">
    <li class="nav-item">
      <h5>Menu</h5>
      <a href="main.php" class="nav-link"><span class="sidebar-icon"></span>
<span>Main</span></a></li>
      <li class="nav-item"><a href="objectsmissingpage.php" class="nav-
link"><span class="sidebar-icon"></span> <span>Objects Missing</span></a></li>
      <li class="nav-item"><a href="objectsidentical.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Objects Identical</span></a></li>
      <li class="nav-item"><a href="fielddifference.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Field Difference</span></a></li>
      <hr>
      <li class="nav-item"><a href="logout.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Logout</span></a></li>
    </nav>

<!-- !PAGE CONTENT! -->
<div class="w3-main" style="margin-left:300px;margin-top:43px;">
  <!-- Header -->
  <header class="w3-container" style="padding-top:22px">

```

```

<h2>Objects Missing</h2>
<h4>on Data Source 1</h4>
<div class="w3-row-padding w3-margin-bottom">
  <a href='datasource1objectsmissing.php' class="btn btn-sm btn-outline-
primary">Data Source #1</a>
  <a href='datasource2objectsmissing.php' class="btn btn-sm btn-outline-
primary">Data Source #2</a>
  <br></br>
</header>
<h5><b>Zone</b></h5>
<div class="w3-row-padding w3-margin-bottom">
  <br></br>
  <div class="card card-body border-light shadow-sm table-wrapper table-
responsive pt-0"><table class="table table-
hover"><th>Zone</th><th>Description</th><th>Building</th><th>Putaway
Sequence</th><th>Pick Sequence</th>
  <?php
  require('connection.php'); // подключаем скрипт
  //echo "<a href='addwarehouse.php' class='btn btn-lg btn-primary';>Add</a>";
  //echo " <a href='deletewarehouse.php' class='btn btn-lg btn-
primary';>Delete</a>";

  $query = "SELECT * FROM `DB WMS2`.`Zone`
  WHERE Zone NOT IN (SELECT Zone FROM `DB WMS`.`Zone`)";

  $result = mysqli_query($connect, $query) or die("Error" .
mysqli_error($connect));

  if($result )

```

```

{
    $rows = mysqli_num_rows($result); // количество полученных строк

    echo "";
    for ($i = 0 ; $i < $rows ; ++$i)
    {
        $row = mysqli_fetch_row($result);
        echo "<tr>";
        for ($j = 0 ; $j < 5; ++$j) echo "<td>$row[$j]</td>";
        echo "</tr>";
    }
    echo "</table>";
echo "</div>";
echo "</div>";
    // очищаем результат
    mysqli_free_result($result);

}
?>

```

```

<br></br>
<h5><b>Building</b></h5>
<div class="w3-row-padding w3-margin-bottom">
<br></br>
    <div class="card card-body border-light shadow-sm table-wrapper table-
responsive          pt-0"><table          class="table          table-
hover"><thead><tr><th>Building</th><th>Warehouse</th></table>
    <?php
require('connection.php'); // подключаем скрипт
//echo "<a href='addwarehouse.php' class='btn btn-lg btn-primary';>Add</a>";

```

```
//echo "<a href='deletewarehouse.php' class='btn btn-lg btn-
primary';>Delete</a>";
```

(випущено)

```
mysqli_free_result($result);
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

Модуль fielddifference.php

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="width=device-width, initial-
scale=1; text/html; charset=utf-8" >
```

```
<!-- Primary Meta Tags -->
```

```
<title>WMS</title>
```

```
<link rel="canonical" href="https://themesberg.com/product/admin-
dashboard/volt-bootstrap-5-dashboard">
```

```
<!-- Favicon --><link rel="apple-touch-icon" sizes="120x120"
href="assets/img/favicon/apple-touch-icon.png">
```

```
<link rel="icon" type="image/png" sizes="32x32"
href="assets/img/favicon/favicon-32x32.png">
```

```
<link rel="icon" type="image/png" sizes="16x16"
href="assets/img/favicon/favicon-16x16.png">
```

```
<link rel="manifest" href="assets/img/favicon/site.webmanifest">
```

```
<link rel="mask-icon" href="assets/img/favicon/safari-pinned-tab.svg"
color="#ffffff">
```

```

    <meta      name="msapplication-TileColor"      content="#ffffff"><meta
name="theme-color" content="#ffffff">

    <!--      Fontawesome      --><link      type="text/css"
href="vendor/@fortawesome/fontawesome-free/css/all.min.css" rel="stylesheet">

    <!--  Notyf  --><link  type="text/css"  href="vendor/notyf/notyf.min.css"
rel="stylesheet">

    <!-- Volt CSS --><link type="text/css" href="css/volt.css" rel="stylesheet">

    <style>
    * {
        box-sizing: border-box;
    }

    /* Create two equal columns that floats next to each other */
    .column {
        float: left;
        width: 50%;
        padding: 10px;
    }

    .row {
        display: flex;
    }
    }

</style>
</head>

<body>
<!-- Sidebar/menu -->

<nav id="sidebarMenu" class="sidebar d-md-block bg-primary text-white
collapse" data-simplebar><br>

```

```

<div class="sidebar-inner px-4 pt-3"><div class="user-card d-flex d-md-none
align-items-center justify-content-between justify-content-md-center pb-4">
  <div class="d-flex align-items-center">
    <div class="d-block">
      <a href="pages/examples/sign-in.html" class="btn btn-secondary text-dark btn-
xs">
        <span class="mr-2"><span class="fas fa-sign-out-alt"></span></span>Sign
Out</a></div></div>
      <div class="collapse-close d-md-none">
        <a href="#sidebarMenu" class="fas fa-times" data-toggle="collapse" data-
target="#sidebarMenu" aria-controls="sidebarMenu" aria-expanded="true" aria-
label="Toggle navigation"></a>
      </div></div>
      <ul class="nav flex-column">
        <li class="nav-item">
          <h5>Menu</h5>
          <a href="main.php" class="nav-link"><span class="sidebar-icon"></span>
<span>Main</span></a></li>
          <li class="nav-item"><a href="objectsmissingpage.php" class="nav-
link"><span class="sidebar-icon"></span> <span>Objects Missing</span></a></li>
          <li class="nav-item"><a href="objectsidentical.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Objects Identical</span></a></li>
          <li class="nav-item"><a href="fielddifference.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Field Difference</span></a></li>
          <hr>
          <li class="nav-item"><a href="logout.php" class="nav-link"><span
class="sidebar-icon"></span> <span>Logout</span></a></li>
        </nav>

<!-- !PAGE CONTENT! -->

```

```

<div class="w3-main" style="margin-left:300px;margin-top:43px;">
  <!-- Header -->
  <header class="w3-container" style="padding-top:22px">
    <h2>Field Differences</h2>
  </header>

  <h5><b>Zones</b></h5>
  <div class="row">
    <div class="column" >
      <h5>Data Source 1</h5>
    </div>
    <div class="column" >
      <h5>Data Source 2</h5>
    </div>
  </div>

  <div class="w3-row-padding w3-margin-bottom">
    <div class="card card-body border-light shadow-sm table-wrapper table-
responsive          pt-0"          ><table          class="table          table-
hover"><th>Zone</th><th>Description</th><th>Building</th><th>Putaway
Sequence</th><th>Pick
Sequence</th><th>Zone</th><th>Description</th><th>Building</th><th>Putaway
Sequence</th><th>Pick Sequence</th>
    <?php
      require('connection.php'); // подключаем скрипт

      $datasource1 = 'Data Source 1';
      $datasource2 = 'Data Source 2';// змінні щоб виводилась відповідна бд
      $items = array ();

      $query = "SELECT

```

```

*

FROM
`DB WMS`.`Zone` t1,
`DB WMS2`.`Zone` t2
WHERE
t1.Zone = t2.Zone AND (t1.Description != t2.Description OR t1.Building !=
t2.Building OR t1.PutawaySequence != t2.PutawaySequence OR t1.PickSequence !=
t2.PickSequence)";

$result = mysqli_query($connect, $query) or die("Error" .
mysqli_error($connect));

if($result )
{
    $rows = mysqli_num_rows($result);          // количество полученных строк
    //$columns = mysqli_num_fields($result); // кількість колонок

    echo "";
    //                                     while ($row = $result->fetch_assoc()){
    //
    //                                     }
    for ($i = 0 ; $i < $rows ; ++$i)
    {
        $row = mysqli_fetch_row($result);
        echo "<tr>";
        for ($j = 0 ; $j < 10 ; ++$j) echo "<td>$row[$j]</td>";
        echo "</tr>";
    }

    echo "</table>";

```

```
echo "</div>";  
echo "</div>";
```

(випущено)

?>

</body>

</html>