

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

ДЛЯ ОБЛІКУ ТА КОНТРОЛЮ РУХУ ОБЛАДНАННЯ

SOFTWARE FOR EQUIPMENT

TRACKING AND MOVEMENT CONTROL

Виконав(ла): студент(ка) 4 курсу, групи КНТ-122

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ЄРМАКОВ Д.А.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЛПАКОВА Т.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

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

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЄРМАКОВА Данила Андрійовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для обліку та контролю руху обладнання. Software for Equipment Tracking and Movement Control

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

затверджені наказом закладу вищої освіти від “ 07 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 10 червня 2026 року

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

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

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

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

7. Дата видачі завдання “ 17 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка)

(підпис) Данило ЄРМАКОВ
(Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна КОЛПАКОВА
(Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
87 с., 3 табл., 39 рис., 3 дод., 18 джерел.

PYCHARM, PYTHON, ЗАСТОСУНОК, МОВА ПРОГРАМУВАННЯ,
ОБЛАДНАННЯ, ОБЛІК, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для обліку та контролю руху обладнання.

Предмет дослідження – програмні засоби для обліку та контролю руху обладнання.

Мета роботи – розробка програмного забезпечення для обліку та контролю руху обладнання.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки PyCharm.

Результати. Розроблено проєктні рішення для створення програмного забезпечення для обліку та контролю руху обладнання. Створено програмне забезпечення для обліку та контролю руху обладнання. Проведено тестування розробленої програмної системи для обліку та контролю руху обладнання.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення для обліку та контролю руху обладнання за допомогою мови програмування Python та середовища розробки PyCharm.

Галузь використання – складські системи.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 87 pages, 3 tables, 39 figures, 3 appendixes, 18 sources.

PYCHARM, PYTHON, APPLICATION, PROGRAMMING LANGUAGE, EQUIPMENT, ACCOUNTING, SOFTWARE.

The object of research is the process of developing software for accounting and controlling the movement of equipment.

The subject of the research is software for supporting the accounting and controlling the movement of equipment process.

The purpose of this work is to develop the software to support the accounting and controlling the movement of equipment process.

Materials, methods and technical tools: Python programming language, PyCharm development environment.

Results. Project solutions for the creation of software for accounting and controlling the movement of equipment has been designed. The software to support the process of accounting and controlling the movement of equipment has been developed. The developed software system for accounting and controlling the movement of equipment has been tested.

Conclusions. The goal of the work has been achieved. The software tools for accounting and controlling the movement of equipment using the Python programming language and the PyCharm development environment have been developed.

Scope of use – warehouse systems.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ.....	9
1 Аналіз предметної області.....	12
1.1 Програмне забезпечення для обліку та контролю руху обладнання	12
1.2 Аналіз програмного забезпечення для обліку та контролю руху обладнання	22
1.3 Висновки за розділом 1	31
2 Матеріали і методи	33
2.1 Вибір мови програмування	33
2.2 Вибір середовища розробки для створення програмного забезпечення для обліку та контролю руху обладнання.....	35
2.3 Висновки за розділом 2	38
3 Опис програми	39
3.1 Структура програмного забезпечення для обліку та контролю руху обладнання	39
3.2 Функціонування програмного забезпечення для обліку та контролю руху обладнання	41
3.3 Розробка бази даних програмного забезпечення для обліку та контролю руху обладнання	43
3.4 Проєктування інтерфейсу програмного забезпечення для обліку та контролю руху обладнання	46
3.5 Висновки за розділом 3	49
4 Експлуатація, тестування та експериментальне дослідження програми.....	51
4.1 Призначення й умови застосування програми	51
4.2 Характеристики програми для обліку та контролю руху обладнання....	52
4.3 Інструкція по експлуатації програми.....	54
4.3.1 Звернення до програми	54

4.3.2 Вхідні й вихідні дані.....	54
4.3.3 Повідомлення	54
4.4 Виконання програмного забезпечення для обліку та контролю руху обладнання	55
4.5 Тестування програмного забезпечення для обліку та контролю руху обладнання	61
4.6 Висновки за розділом 4	61
Висновки	62
Перелік джерел посилання	65
Додаток А Технічне завдання	67
Додаток Б Фрагмент тексту програми	72
Додаток В Слайди презентації	80

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ERP – Enterprise Resource Planning;

MVC – Model-View-Controller;

БД – база даних;

ПЗ – програмне забезпечення.

ВСТУП

Актуальність програмного забезпечення для обліку та контролю руху обладнання зумовлюється необхідністю швидкого отримання достовірних відомостей про наявні ресурси та їх використання. У багатьох організаціях обладнання постійно переміщується між підрозділами, використовується у різних процесах, потребує обслуговування або списання, тому ручні методи фіксації таких змін призводять до неточностей, втрати часу та фінансових збитків. Завдяки автоматизованим рішенням забезпечується безперервний моніторинг стану обладнання, фіксується історія його переміщень і гарантується своєчасне реагування на несправності чи потребу в технічному догляді. Підвищення прозорості облікових операцій сприяє раціональному використанню матеріальних ресурсів, запобігає необґрунтованим витратам та дає можливість оптимізувати управлінські рішення [1]-[3].

Таким чином, впровадження програмного забезпечення для контролю руху обладнання стає важливою умовою підвищення ефективності роботи підприємств, що залежить від точності та своєчасності інформації про їх матеріально-технічну базу.

Додаткову цінність таких програмних систем формує можливість інтеграції з різними сервісами управління організацією, що дозволяє використовувати інформацію про обладнання у плануванні виробництва, формуванні бюджету, закупівлях і прогнозуванні потреб. Цифрова обробка даних сприяє узгодженню дій між структурними підрозділами, а також дає змогу формувати обґрунтовані звіти для керівництва та контролюючих органів. У результаті інформація про ресурси перестає існувати в розрізнених джерелах і набуває системного вигляду [4]-[5].

Програмні засоби обліку обладнання також відіграють важливу роль у підвищенні рівня безпеки на підприємстві. Точні дані про технічний стан та строки обслуговування дозволяють своєчасно проводити профілактичні заходи та знижувати ризики аварійних ситуацій. Наявність актуальної

інформації про місце знаходження та умови експлуатації обладнання дає змогу запобігати його несанкціонованому використанню, що позитивно впливає на охорону праці та збереження майна [5]-[6].

Крім того, використання таких програмних рішень полегшує адаптацію підприємства до сучасних стандартів цифрового управління, що є вимогою ринку та умовою конкурентоспроможності. Інформаційні технології перетворюють підхід до керування матеріальними ресурсами, замінюючи складні ручні процедури швидкими, точними та контрольованими процесами. Це сприяє розвитку організаційної культури, в якій рішення ухвалюються на основі перевірених даних, а не припущень чи застарілих відомостей [4]-[6].

Проте важливою перешкодою під час використання таких програмних систем може стати потреба в додаткових витратах на придбання ліцензій, обслуговування програмного продукту, технічну підтримку та навчання персоналу. Складність інтеграції зі сторонніми програмними системами також може знижувати ефективність програмного забезпечення. Не всі продукти забезпечують взаємодію з бухгалтерськими сервісами, системами логістики, закупівель або виробничого планування, що призводить до фрагментації інформаційного простору підприємства. У результаті дані залишаються розділеними між різними платформами, і використання автоматизації не досягає очікуваного рівня [1]-[6].

Тому актуальною є розробка програмного забезпечення для обліку та контролю руху обладнання. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для обліку та контролю руху обладнання.

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

- виконати аналіз предметної області та програмних засобів для обліку та контролю руху обладнання;
- здійснити проектування програмного забезпечення для обліку та контролю руху обладнання;

- створити програмне забезпечення для обліку та контролю руху обладнання;
- виконати тестування розробленого програмного забезпечення для обліку та контролю руху обладнання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для обліку та контролю руху обладнання

Програмне забезпечення для обліку та контролю руху обладнання являє собою спеціалізовану систему, призначену для автоматизації процесів обліку, відстеження та управління обладнанням у різних організаціях. Воно дозволяє централізовано зберігати інформацію про наявність, місцезнаходження, стан та історію використання технічних засобів, що суттєво спрощує управління матеріальними ресурсами та зменшує ризик втрат чи помилок [1].

Серед переваг такого програмного забезпечення можна виділити підвищення ефективності обліку, скорочення часу на рутинні операції, можливість відстежувати рух обладнання в режимі реального часу, а також інтеграцію з іншими системами управління підприємством, що сприяє більш точному фінансовому та виробничому плануванню. Воно також забезпечує створення звітів та аналітики, що полегшує прийняття управлінських рішень і підвищує прозорість процесів [1].

При використанні таких програмних засобів існують певні проблеми з необхідністю початкових витрат на придбання та впровадження системи, можливу складність у налаштуванні та адаптації під специфічні потреби підприємства, а також залежність від наявності відповідного апаратного та програмного забезпечення. Для користувачів, які не мають досвіду роботи з подібними системами, освоєння функціоналу може потребувати певного часу та навчання [1].

Особливістю таких програмних систем є модульна структура, що дозволяє виділяти підсистеми для інтерфейсу користувача, логіки обробки даних і бази даних. Логіка обробки включає функції додавання та редагування даних, виведення інформації, авторизації користувачів та експорту звітів. Також характерним є використання сучасних технологій для інтеграції з

вебінтерфейсом та мобільними пристроями, що забезпечує доступ до даних із будь-якої точки та покращує контроль за рухом обладнання [1].

Таким чином, програмне забезпечення для обліку та контролю руху обладнання виступає ключовим інструментом для організацій, що прагнуть оптимізувати управління своїми ресурсами, підвищити точність обліку та забезпечити прозорість всіх процесів, пов'язаних із використанням технічних засобів [1].

Програмні системи, призначені для контролю запасів і складських операцій, набувають особливого значення у діяльності організацій, що займаються продажем товарів, розповсюдженням продукції чи наданням послуг. Використання таких рішень забезпечує впорядковане відстеження матеріальних ресурсів та їхнє коректне розподілення між користувачами чи партнерами. Під визначенням програмного забезпечення для управління запасами розуміється комплекс інструментів, що дозволяє уникати псування матеріальних цінностей, зменшувати кількість небажаних помилок та крадіжок, а також підтримувати належний рівень задоволеності споживачів [1].

Використання програмних інструментів для управління запасами сприяє точному та своєчасному розподілу продуктів, оперативному реагуванню на потреби клієнтів, а також прискоренню етапів укладення угод і проведення транзакцій. У багатьох випадках такі системи функціонують у хмарному середовищі, поєднуються з іншими інформаційними платформами та підтримують роботу з мобільними пристроями, що особливо корисно для підприємств із розгалуженою структурою [1]-[2].

Позитивний ефект від використання таких програмних рішень проявляється у можливості зберігати дані про запаси в єдиному централізованому середовищі, що полегшує контроль великої кількості складів і прискорює процес інформування клієнтів щодо виконання їхніх замовлень. Наявність якісної системи також дає змогу уникати надмірних витрат часу на ручне ведення документації, оскільки актуальні дані одразу

надходять до пристроїв працівників. Підвищення точності обліку служить основою для зростання кількості успішно завершених угод, адже зменшується ризик втрати клієнтів через неправильну інформацію щодо наявності товарів [1]-[2].

Використання таких програмних систем забезпечує дотримання строків доставки, уникнення затримок та надання можливості користувачам самостійно стежити за виконанням замовлення. Завдяки точному обліку компанія може оптимізувати частоту закупівель, що допомагає уникати накопичення надлишків та зменшує витрати на їх зберігання. Крім того, автоматизація дає змогу реагувати на зміни попиту без додаткових витрат, а в окремих випадках навіть включає функції пакування й відправлення товарів [1]-[2].

Наявність актуальних даних та можливість персоналізованого обслуговування сприяють підвищенню задоволеності клієнтів. Важливим аспектом також є інтеграція з іншими інформаційними системами, зокрема бухгалтерськими чи ERP-платформами. Найбільш універсальними вважаються рішення з відкритим API, оскільки вони дозволяють налаштовувати взаємодію з будь-якими зовнішніми сервісами. Такі платформи здатні збирати аналітичні дані, виявляти закономірності та прогнозувати зміни на ринку, що допомагає будувати стратегію розвитку бізнесу з урахуванням реальних показників ефективності [1]-[2].

Типізація програмних рішень, що використовуються для керування запасами, може здійснюватися за різними принципами, однак ключовим фактором у більшості випадків вважається призначення та спосіб застосування конкретного інструмента. Можливості системи та спосіб її взаємодії з користувачем визначають, до якої групи вона належатиме, оскільки функціонал і модель роботи суттєво впливають на її місце серед інших аналогічних продуктів [2]-[3].

До окремого різновиду належать програмні рішення, що наслідують підхід електронних таблиць і слугують альтернативою паперовому веденню

складів для невеликих підприємств. Такі інструменти створюють основу для первинного цифрового контролю товарів, дозволяють опрацьовувати обмежені обсяги продукції та не позбавляють користувача необхідності самостійно виконувати складні обчислення. Через залежність від ручних дій людський чинник у таких системах залишається основним ризиком неточностей [2]-[3].

Інший різновид забезпечує фіксацію запасів завдяки використанню штрих-кодів та обладнання, пов'язаного з касовими і торговими системами. У такому випадку відомості про продажі та надходження товарів оновлюються автоматично, що дає змогу значно пришвидшити їхній рух між складами, забезпечити точність під час обробки замовлень і зменшити витрати на ручне втручання. Завдяки відсутності фіксованих обмежень щодо розміру складської інфраструктури та можливості інтеграції з іншими програмами такі рішення вважаються більш універсальними [2]-[3].

Більш технологічний підхід реалізується у програмних системах, що використовують спеціальні радіочастотні мітки. Подібні програмні інструменти здатні автоматично контролювати великі потоки товарів, працюють із великими обсягами замовлень та усувають потребу в паралельному контролі кількох складських приміщень. Водночас їхня висока функціональність пов'язана зі складністю впровадження та значними фінансовими витратами, через що такий варіант не завжди доцільний для бізнесу, який тільки розвивається [2]-[3].

Ефективність роботи програм для управління запасами можна зрозуміти завдяки аналізу їхніх функціональних властивостей. Незважаючи на різницю між системами, певні можливості зустрічаються майже в кожному інструменті цього типу. До них належать механізми впорядкування товарів у групи, що дозволяють прогнозувати потреби ринку та виконувати пошук за широким набором критеріїв; функції оброблення замовлень, які поєднують операції з купівлі, продажу і виставлення рахунків; автоматизовані інструменти повторного замовлення, що мінімізують потребу в тривалому

ручному контролю; технології сканування, які створюють основу для електронного супроводу товарів; а також інструменти, які підтримують актуальність даних, роблячи їх доступними для всієї команди. Завдяки цьому управління запасами перетворюється на процес, який не потребує тривалого ручного втручання та забезпечує оперативний доступ до актуальної інформації щодо складів і продажів [3]-[4].

У процесі вибору програмної системи управління запасами передбачається ретельна оцінка її можливостей і відповідності майбутньому розвитку компанії. Не варто поспішати з придбанням програмного забезпечення, якщо обсяг товарів ще невеликий і бізнес лише набирає обертів. Однак доцільно відразу звертати увагу на рішення, здатні адаптуватися до зростання товарообігу та зміни попиту. Розумніше обирати програму, яка зберігатиме стабільність навіть у періоди різких змін продажів і не вимагатиме додаткових витрат у критичних ситуаціях [3]-[4].

Важливо, щоб програмна система забезпечувала розширені можливості взаємодії з клієнтами, пропонуючи оперативний доступ до інформації про стан замовлення та дозволяючи відстежувати статус товарів. Такий підхід формує довіру покупця і підвищує задоволеність обслуговуванням. Разом із цим очікується гарантована технічна підтримка з боку розробника, оскільки навіть найбільш продуманий інструмент може зіткнутися з помилками та збоями. Успішне використання програмного забезпечення неможливе без фахівців, здатних усунути можливі проблеми та забезпечити коректність обчислень і безпеку даних [3]-[4].

Вартість програмних рішень для обліку запасів визначається функціональністю, набором модулів та масштабом використання. На ціну впливає кількість користувачів і товарних позицій, які будуть оброблятися системою. Деякі продукти передбачають диференційований доступ до підтримки або додаткові платежі при збільшенні складів чи необхідності розширеного сервісу. На ринку можна знайти як безкоштовні інструменти для

невеликих підприємств, так і преміальні рішення з повним переліком функцій і персональним тарифом для великих організацій [3]-[4].

Під час роботи із програмною системою управління обладнанням на складі можуть виникати певні труднощі. Компанії, що здійснюють операції в різних валютах, повинні заздалегідь переконатися в готовності програмного забезпечення працювати з валютними курсами. Якщо цього не врахувати, існує ризик втрат при закупівлі товарів у валюті, відмінній від локальної. Контроль за доступом до програми зменшує ймовірність шахрайства, але повністю не виключає його, особливо при великій кількості працівників, які мають право змінювати дані. Окрім цього, покладання на систему надмірних функцій іноді спричиняє зниження уваги менеджерів до якості зберігання товару або стану складських приміщень, що може призвести до втрат через пошкодження продукції [3]-[4].

Розгляд програмних рішень для контролю запасів є важливим кроком у підвищенні ефективності діяльності підприємства. Перед початком використання такого інструменту доцільно визначити ключові труднощі та очікування, пов'язані з обліком товарів. Різні підходи до управління запасами можуть забезпечувати різний результат, тому важливо обирати ту платформу, яка найкраще відповідає конкретним умовам роботи. Особливу увагу варто звертати на здатність системи миттєво відображати зміни у наявності продукції, що робить роботу з матеріальними ресурсами більш впорядкованою. Порівнюючи готові програмні рішення з індивідуальними, необхідно враховувати, наскільки їхня структура узгоджується з потребами бізнесу [4]-[5].

Програмне забезпечення для контролю запасів виступає цифровим інструментом, що дає можливість автоматизувати процеси обліку, контролю кількості товарів та коригування складських залишків. Система має на меті забезпечити точність відомостей про наявність продукції й оптимізувати операції, пов'язані з товарними потоками. Основою таких програм є можливість безперервного відстеження запасів, формування замовлень та

прогнозування потреб у продукції. Окрім цього, такі рішення здатні поєднуватися з іншими сервісами, включно зі спеціалізованими торговельними платформами, системами продажу та складськими інструментами, компенсуючи ризики помилок і створюючи основу для стабільної роботи [4]-[5].

Завдяки використанню даних про попередні операції подібне програмне забезпечення дозволяє передбачати майбутні потреби у товарах та підтримувати обсяг запасів на оптимальному рівні. Це допомагає уникати надлишкових витрат, пов'язаних зі зберіганням непроданих продуктів, а також зменшує ймовірність браку товару. Як наслідок, користування системою сприяє прийняттю більш точних рішень і підвищенню результативності господарської діяльності [4]-[5].

Впровадження програмних інструментів управління запасами відіграє важливу роль у розвитку компаній, які прагнуть підвищити якість логістичних процесів і скоротити витрати. Значна частина підприємств не має ефективного способу контролю товарів без допоміжних технологій, що збільшує ризик неточностей у даних. Великі організації використовують такі системи для оптимізації ланцюгів постачання та покращення взаємодії з клієнтами, що підтверджує вагому роль технологічної підтримки в сучасній торгівлі та виробництві. Поступова заміна ручних реєстрацій цифровими рішеннями забезпечує зниження частоти помилок, підвищує швидкість оновлення інформації та підсилює надійність облікових процесів [4]-[5].

Програмні засоби управління запасами також забезпечують автоматизацію замовлення товарів, точність введення даних завдяки використанню спеціальних ідентифікаторів та створюють інформаційні звіти для аналізу діяльності постачальників і товарообігу. Можливість прогнозувати потреби за допомогою аналізу історичних даних дає можливість зберігати баланс між попитом і доступністю продукції. Таким чином, застосування цих рішень забезпечує раціональне управління товарними ресурсами та сприяє стабільному розвитку бізнесу [4]-[5].

Різні типи програмних систем для обліку обладнання можуть застосовуватися залежно від особливостей підприємства, його масштабу та структури товарних потоків. Одні орієнтовані на постійне оновлення інформації, інші працюють за періодичними перевірками, окремі моделі оптимізують кількість товарів відповідно до виробничих циклів або економічної доцільності. Розуміння різниці між ними дає можливість підібрати той інструмент, який найбільше підходить для конкретної діяльності, забезпечуючи сталий та ефективний контроль над запасами [4]-[5].

Пристрої, що підтримують роботу програмних систем контролю запасів, забезпечують точніший облік та прискорюють виконання операцій. Використання спеціалізованого обладнання допомагає мінімізувати людський фактор і пришвидшує фіксацію даних у процесі переміщення товарів. Зчитувачі штрихових кодів дають змогу оперативно фіксувати необхідну інформацію про продукцію без ручного внесення даних. Технологія RFID дає розширені можливості простеження товарів та дозволяє у будь-який момент отримувати актуальні відомості про їхнє місцезнаходження. Мобільні пристрої створюють можливість керувати запасами без прив'язки до робочого місця, а сенсорні системи Інтернету речей надають інформацію про стан товарів, контролюючи зовнішні умови їх зберігання. Окреме обладнання, інтегроване з пунктами продажу, забезпечує безперервне оновлення інформації про рух продукції під час здійснення транзакцій [5]-[6].

Програмні рішення для обліку запасів діють як автоматизовані інструменти, які дозволяють керувати товарними потоками без постійного ручного втручання. Такі системи фіксують рівень залишків, переміщення продукції та зміни в замовленнях у міру їх виникнення. Обмін даними з іншими бізнес-системами забезпечує єдине інформаційне середовище та допомагає синхронізувати різні процеси в компанії. Завдяки прогнозуванню майбутнього попиту такі системи сприяють підтримці оптимальних складських запасів. Використання сканування та технологій безконтактної

ідентифікації покращує точність аналітики та скорочує час виконання операцій [5]-[6].

Використання програмного забезпечення для контролю запасів та обладнання сприяє зниженню непродуктивних витрат і робить управління ресурсами більш раціональним. Своєчасне оновлення даних усуває ризики нестачі товарів або надлишкового накопичення. Автоматизація дозволяє суттєво скоротити трудомісткість процесів обліку та звільняє підприємства від повторюваних операцій. Покращена аналітика допомагає уникати збиткових рішень і сприяє оптимізації складських витрат. Узгодження інформації з фінансовими та планувальними системами формує більш цілісне бачення діяльності підприємства [5]-[6].

Пошук відповідного програмного рішення для обліку запасів та обладнання залежить від специфіки роботи підприємства, його масштабів та можливостей розвитку. Важливо враховувати необхідний функціонал, вартість системи, а також її адаптацію до потреб користувачів. Від надійності підтримки постачальника та складності впровадження залежить швидкість та якість переходу на нові технології. Чим простішою є система у використанні, тим менше часу витрачається на навчання персоналу та налаштування робочих процесів [5]-[6].

Неточність обліку залишається однією з центральних проблем управління запасами, оскільки неправильні дані безпосередньо впливають на рівень витрат і доступність товарів. Перевищення запасів призводить до непотрібного зростання витрат на їх утримання, тоді як нестача товарів спричиняє втрату покупців. Значна частина розбіжностей виникає через ручне внесення даних, що робить автоматизовані системи ефективним засобом усунення помилок. Використання сучасних програмних рішень дає можливість підтримувати облік із максимальною точністю і підвищувати результативність роботи підприємства [5]-[6].

У динамічному середовищі сучасної торгівлі програмні рішення для контролю товарних запасів та обладнання поступово переходять на новий

рівень. З розвитком технологій очікується більш активне використання алгоритмів, здатних самостійно аналізувати поведінку ринку, визначати попит та допомагати підприємствам точніше формувати обсяги продукції. Застосування пристроїв, що передають дані в режимі миттєвого доступу, посилюватиме контроль над складськими залишками й дасть змогу відстежувати рух товарів у реальних умовах [5]-[6].

Хмарні сервіси продовжуватимуть набирати популярність, забезпечуючи безперешкодний доступ до даних незалежно від місця перебування користувача та дозволяючи масштабувати інструменти відповідно до потреб бізнесу [5]-[6].

Розробка власного програмного інструмента для контролю запасів та обладнання може стати ефективним кроком для компанії, якщо програмний продукт враховуватиме її реальні потреби та специфіку. Створення такої програмної системи передбачає чітке розуміння функціоналу, який повинен забезпечити конкурентні переваги та відповідати очікуванням майбутніх користувачів. Важливим етапом є дослідження ринку, що допомагає окреслити вимоги, технологічну базу та необхідний обсяг майбутньої роботи. Значну увагу приділяють зручності використання, оскільки добре продуманий інтерфейс підвищує ефективність працівників. Початковий варіант програмного продукту створюється для оцінки ключових функцій і збору відгуків, які дозволяють удосконалити рішення до повномасштабного впровадження. Завершальним етапом є перевірка стабільності, надійності та відповідності стандартам безпеки [5]-[6].

Визначено, що програмне забезпечення для обліку та контролю руху обладнання являє собою спеціалізовану систему, призначену для автоматизації процесів обліку, відстеження та управління обладнанням у різних організаціях. Воно дозволяє централізовано зберігати інформацію про наявність, місцезнаходження, стан та історію використання технічних засобів, що суттєво спрощує управління матеріальними ресурсами та зменшує ризик втрат чи помилок.

1.2 Аналіз програмного забезпечення для обліку та контролю руху обладнання

Проаналізуємо програмне забезпечення для обліку та контролю руху обладнання [7]-[14].

Програмне забезпечення Fishbowl (рис. 1.1) є прикладом платформи, що забезпечує централізоване керування товарними залишками та супутніми бізнес-процесами для організацій, які займаються виробництвом чи зберіганням продукції. Програмне забезпечення поєднує операції, пов'язані з обліком продукції, її рухом, реалізацією та плануванням виробництва, дозволяючи контролювати всі процеси в межах однієї системи. За його допомогою інформація про товарні залишки узгоджується з фінансовими операціями та виробничими даними, що забезпечує прозорість обліку та скорочує потребу у дублюванні дій [7], [8], [10], [12].

The screenshot displays the Fishbowl software interface. On the left is a sidebar with navigation options: Dashboard, Materials, Inventory, Sales, Purchasing, Accounting, Reports, Integrations, and Setup. The main area shows an 'Inventory' list with items like B201 Premium Brake Cables, BP100 Quality Brake Pads, BP300 Premium Brake Pads, C201 Single Speed Chain, CK100 Alloy Single Speed Crankset, F202 Access Creator Frame, and FG2100 Extreme Mountain Bike. A detailed view for item B201 is open, showing its status: 15 ON HAND, 0 ON ORDER, 15 AVAILABLE, 0 COMMITTED, and 65 ALLOCATED. It also displays an AVERAGE COST of \$10.75 and a TOTAL COST of \$161.25. Below this, a table shows the inventory location breakdown for 'Bin 100 (Warehouse | Aisle 1)'.

Location	On Hand	On Order	Available	Committed	Allocated
Bin 100 (Warehouse Aisle 1)	15 ea	0 ea	15 ea	0 ea	0 ea
Bin 100 (Warehouse Aisle 1)	15 ea	0 ea	15 ea	0 ea	0 ea
Bin 100 (Warehouse Aisle 1)	15 ea	0 ea	15 ea	0 ea	0 ea
Bin 100 (Warehouse Aisle 1)	15 ea	0 ea	15 ea	0 ea	0 ea

Рисунок 1.1 – Програмне забезпечення Fishbowl [12]

Програмне забезпечення Fishbowl підтримує роботу із замовленнями, відображаючи всі етапи їх обробки та зміни станів. Користувач має доступ до

повної історії оформлення й виконання замовлення: від первинного запиту до передачі товару клієнту. Такий підхід дає змогу швидко реагувати на звернення покупців і надавати їм актуальну інформацію щодо готовності замовлення. У межах складської роботи програмне забезпечення допомагає формувати набори товарів, оптимізувати переміщення працівників та зменшувати витрати часу на виконання операцій. Крім того, підтримується друк маркувальних документів і можливість контролю доставки через інтеграцію з транспортними сервісами [7], [8], [10], [12].

Програмне забезпечення Fishbowl є результативним інструментом для компаній, які мають справу з матеріальними запасами, оскільки програма не лише підтримує облік, а й відстежує рух товарів упродовж усього їх технологічного циклу. Інтеграція з бухгалтерськими програмними системами дає змогу відразу узгоджувати витрати, складування та доходи з фінансовою звітністю, що спрощує підготовку документів та скорочує кількість додаткових операцій [7], [8], [10], [12].

Найбільш доцільно використовувати Fishbowl підприємствам, що займаються виробництвом товарів або оптовою діяльністю та потребують контролю за ресурсами під час їх переміщення й переробки. Програма допомагає невеликим і середнім організаціям ефективніше організовувати роботу зі складом, оптимізувати маршрутизацію та підвищувати продуктивність процесів, забезпечуючи зручний інструмент керування запасами на кожному етапі їх обігу [7], [8], [10], [12].

Основними характеристиками програмного забезпечення Fishbowl є такі [7], [8], [10], [12]:

- інвентаризація: програмне забезпечення Fishbowl підтримує управління запасами в режимі реального часу, дозволяючи відстежувати їх рух і залишки на складі [7], [8], [10], [12];

- замовлення: програмне забезпечення Fishbowl відображає всі етапи обробки замовлень — від котирування до відправки — так, щоб контролювати статуси й ефективно реагувати на запити [7], [8], [10], [12];

- оптимізація складу: програмне забезпечення Fishbowl створює комплектуючі списки та маршрути для працівників, зменшуючи час переміщення на складі й підвищуючи продуктивність [7], [8], [10], [12];

- інтеграція перевізників: програмне забезпечення Fishbowl дозволяє генерувати ярлики для відправлень і відстежувати їх через популярні служби доставки прямо з системи [7], [8], [10], [12];

- фінансова синхронізація: програмне забезпечення Fishbowl інтегрує дані про запаси з бухгалтерськими системами, таким чином пов'язуючи витрати й доходи з фінансовими даними [7], [8], [10], [12];

- безперервний контроль: програмне забезпечення Fishbowl відстежує життєвий цикл запасів, надаючи повну історію переміщень, використання і стану товарів [7], [8], [10], [12];

- гнучка масштабованість: програмне забезпечення Fishbowl підходить і для малих, і для середніх бізнесів, забезпечуючи інструменти управління незалежно від обсягу операцій [7], [8], [10], [12].

Серед недоліків програмного забезпечення Fishbowl можна відзначити залежність від правильного налаштування, оскільки система потребує ретельної конфігурації для повноцінного використання своїх можливостей. Також окремі функції пропонуються як модулі, що потребують додаткової оплати, тому повна функціональність може вимагати значних витрат. Варто враховувати й складність освоєння, адже інтерфейс і розгалужені налаштування можуть вимагати часу для адаптації персоналу. Серед обмежень присутня прив'язка до певних інтеграцій, зокрема повна синхронізація з окремими бухгалтерськими системами можлива лише за наявності відповідних додаткових компонентів [7], [8], [10], [12].

Програмне забезпечення Striven (рис. 1.2) являє собою комплексне рішення, орієнтоване на організацію діяльності підприємств, які працюють у сфері роздрібної торгівлі та оптового постачання. Програмний засіб охоплює інструменти для контролю товарних ресурсів і дає можливість автоматизувати ключові складські процеси. Завдяки поєднанню різних функціональних

модулів на одній платформі забезпечується централізоване керування бізнесом без потреби у використанні додаткових систем. У його склад входять засоби для фінансового обліку, роботи з клієнтами, реалізації проєктів і завдань, контролю запасів, управління персоналом і організації внутрішніх робочих процесів [9], [13].

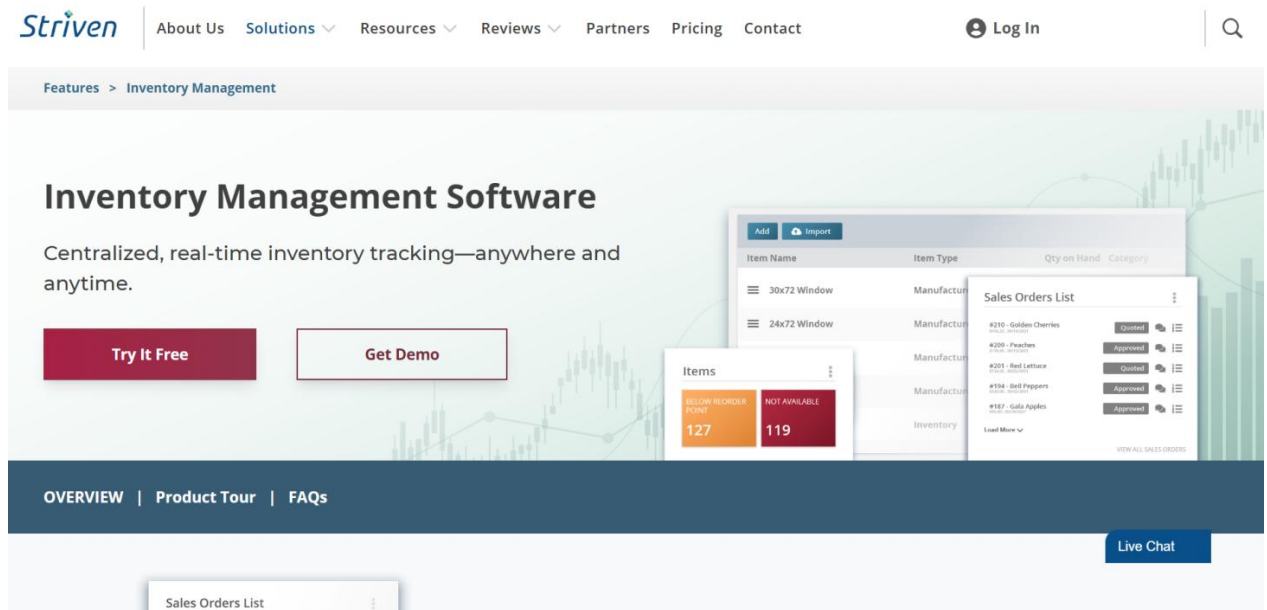


Рисунок 1.2 – Програмне забезпечення Striven [13]

Програмне забезпечення Striven дозволяє працювати з різними обсягами товарних позицій та підтримує широкий спектр номенклатури. Можливість обслуговування дропшипінгу робить програму придатною для компаній, що займаються торгівлею великою кількістю різноманітної продукції. Гнучкість платформи забезпечує адаптацію під потреби конкретного підприємства, що сприяє більш ефективному використанню її інструментів у сфері обліку та контролю руху обладнання [9], [13].

Програмне рішення Striven позиціонується як економічно вигідний варіант у порівнянні з іншими системами подібного призначення. Доступність супроводу та підтримки користувачів сприяє швидкому розв'язанню технічних питань. Разом з тим велика кількість можливостей може вимагати додаткового часу для освоєння, а обмежений перелік доступних інтеграцій

інколи ускладнює сумісність із зовнішніми інструментами. Іноді користувачі відзначають поодинокі технічні неполадки, які можуть виникати під час роботи [9], [13].

Програмне забезпечення Striven має такі особливості [9], [13]:

- комплексність функціоналу: програмне забезпечення Striven забезпечує централізоване керування бізнесом із фінансами, продажами, проектами, персоналом та запасами на єдиній платформі [9], [13];

- підтримка складських операцій: програмне забезпечення Striven дає можливість виконувати контроль руху товарів, обладнання та виконання складських завдань у реальному часі [9], [13];

- автоматизація обліку: програмне забезпечення Striven надає інструменти для фіксації залишків, переміщень, постачань та руху матеріальних активів із подальшою інтеграцією у внутрішні процеси підприємства [9], [13];

- масштабованість товарних баз: програмне забезпечення Striven дозволяє працювати з великою кількістю номенклатури та підтримує різні моделі торгівлі [9], [13];

- налаштування під конкретні потреби: програмне забезпечення Striven передбачає персоналізацію структури, робочих процесів та доступів відповідно до специфіки діяльності підприємства [9], [13];

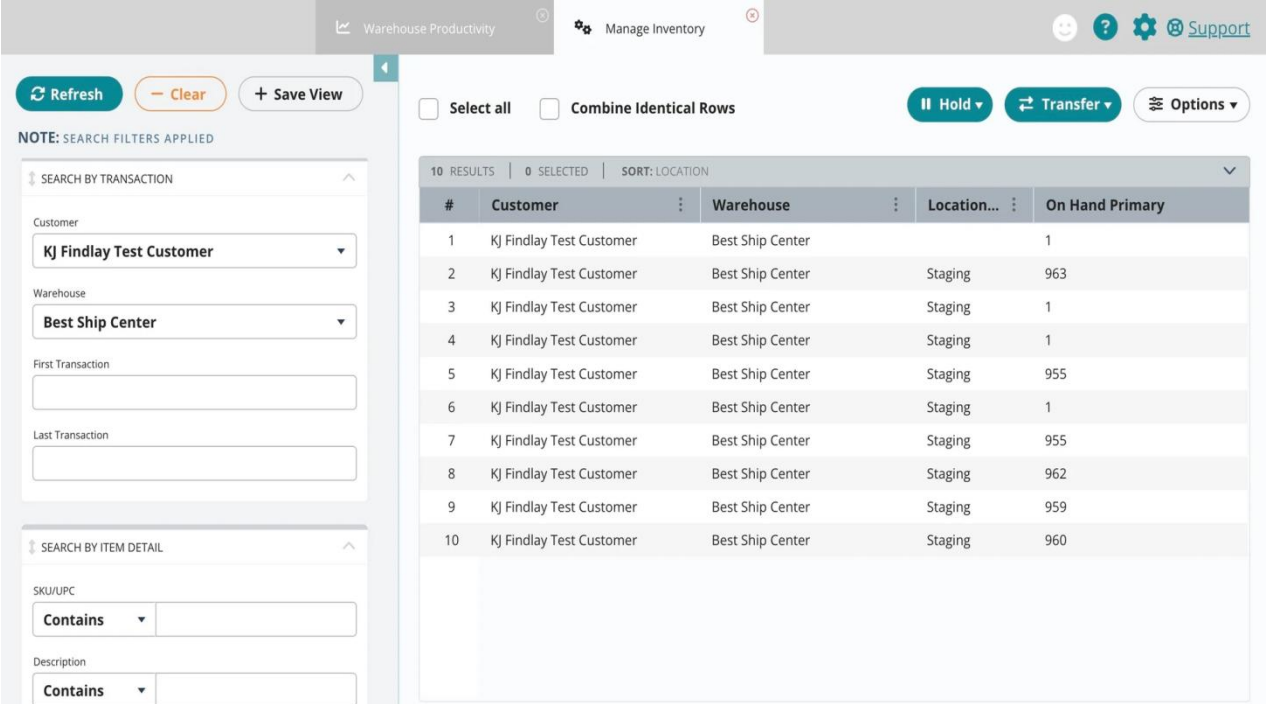
- клієнтська підтримка: програмне забезпечення Striven пропонує повноцінне обслуговування, включаючи консультації та допомогу по телефону, що спрощує впровадження та використання системи [9], [13];

- конкурентна вартість: програмне забезпечення Striven відзначається доступною ціною у порівнянні з іншими платформами комплексного керування бізнесом [9], [13].

Серед недоліків програмного забезпечення Striven можна відзначити складність первинного освоєння через велику кількість інструментів та функцій, що потребують додаткового часу на навчання. Можливості інтеграції з іншими системами є обмеженими, тому програмне забезпечення може не

поєднуватися з усіма зовнішніми сервісами, які використовуються в організації. Також можуть виникати періодичні технічні збої або помилки, здатні впливати на стабільність роботи системи та вимагати діяльності служби підтримки для їх усунення [9], [13].

Програмне забезпечення ZPL Warehouse Manager (рис. 1.3) є хмарною системою для організації складської діяльності, що використовується для контролю руху матеріальних цінностей, оптимізації логістичних операцій і забезпечення клієнтів повною інформаційною прозорістю. У програмі реалізовано перехід від паперових інструкцій до автоматизованого управління процесами, завдяки чому значно зменшується вплив людського фактора та пришвидшується виконання основних завдань. Програмне рішення охоплює цифрову обробку надходжень, виконання замовлень, формування документації та звітів, а також підтримує доступ до актуальних показників складської роботи в режимі онлайн [10], [14].



The screenshot displays the ZPL Warehouse Manager interface. On the left, there are search filters for transactions and item details. The main area shows a table with 10 results, sorted by location. The table columns are #, Customer, Warehouse, Location..., and On Hand Primary. The data rows show various staging locations for the 'KJ Findlay Test Customer' at the 'Best Ship Center' warehouse.

#	Customer	Warehouse	Location...	On Hand Primary
1	KJ Findlay Test Customer	Best Ship Center		1
2	KJ Findlay Test Customer	Best Ship Center	Staging	963
3	KJ Findlay Test Customer	Best Ship Center	Staging	1
4	KJ Findlay Test Customer	Best Ship Center	Staging	1
5	KJ Findlay Test Customer	Best Ship Center	Staging	955
6	KJ Findlay Test Customer	Best Ship Center	Staging	1
7	KJ Findlay Test Customer	Best Ship Center	Staging	955
8	KJ Findlay Test Customer	Best Ship Center	Staging	962
9	KJ Findlay Test Customer	Best Ship Center	Staging	959
10	KJ Findlay Test Customer	Best Ship Center	Staging	960

Рисунок 1.3 – Програмне забезпечення ZPL Warehouse Manager [14]

Програмне забезпечення ZPL Warehouse Manager надає можливість автоматичного та ручного відстеження товарних партій, підтримує роботу зі

штрих-кодами, може самостійно формувати необхідні документи, включаючи накладні та рахунки. Інтерфейс дозволяє одночасно керувати клієнтами та їхньою продукцією без потреби перемикання між окремими вікнами, а функції складських операцій реалізуються з урахуванням різних одиниць вимірювання та трансформацій товарних позицій [10], [14].

Платформа пропонує механізми планування роботи складу, у тому числі розподіл доків, календар управління з можливістю перетягування подій, оповіщення про заплановані дії та систему самообслуговування для перевізників. За допомогою інструментів виставлення рахунків забезпечується гнучка тарифікація з можливістю деталізації оплати за зберігання, відвантаження та інші операції. Є можливість експорту фінансових документів у популярні бухгалтерські платформи, що полегшує фінансовий контроль. Усі операції, пов'язані з відправленням, формуванням накладних і резервуванням вантажів, також виконуються в цифровому середовищі, забезпечуючи зручний пошук та обробку інформації без зайвих переходів між вікнами. Таким чином, система спрямована на підвищення продуктивності складських процесів та покращення якості обслуговування клієнтів завдяки автоматизованому управлінню всіма етапами руху товарів [10], [14].

Програмне забезпечення 3PL Warehouse Manager має такі особливості [10], [14]:

- хмарна платформа для складів: програмне забезпечення 3PL Warehouse Manager дозволяє керувати складськими процесами дистанційно та забезпечує доступ до даних у реальному часі [10], [14];

- автоматизація операцій: програмне забезпечення 3PL Warehouse Manager замінює ручні та паперові дії, мінімізує помилки та прискорює виконання складських завдань [10], [14];

- підтримка штрих-кодування: програмне забезпечення 3PL Warehouse Manager забезпечує швидке сканування та точне відстеження товарів за унікальними ідентифікаторами [10], [14];

– гнучке керування замовленнями: програмне забезпечення ZPL Warehouse Manager дозволяє автоматично або вручну обробляти надходження, комплектацію та відвантаження товарів [10], [14];

– робота з партіями товарів: програмне забезпечення ZPL Warehouse Manager надає можливість створювати, редагувати та відключати контроль партій залежно від потреб підприємства [10], [14];

– інтеграція з бухгалтерією: програмне забезпечення ZPL Warehouse Manager забезпечує експорт документів у популярні фінансові системи для автоматизованого обліку платежів і витрат [10], [14];

– інтелектуальне планування складу: програмне забезпечення ZPL Warehouse Manager пропонує графічний календар, інструменти планування доків і розподілу ресурсів з підказками оптимізації [10], [14];

– портал для перевізників: програмне забезпечення ZPL Warehouse Manager дає перевізникам можливість самостійно отримувати інформацію та записувати вантажні операції без участі співробітників складу [10], [14];

– підтримка різних одиниць вимірювання: програмне забезпечення ZPL Warehouse Manager автоматично розраховує перетворення та забезпечує коректний облік товарних позицій [10], [14];

– розширена звітність: програмне забезпечення ZPL Warehouse Manager формує аналітичні зведення щодо запасів, руху товарів, фінансових операцій та ефективності складських процесів [10], [14].

Серед недоліків програмного забезпечення ZPL Warehouse Manager можна відзначити складність первинного налаштування, що потребує значних витрат часу для адаптації до специфічних бізнес-процесів. Помітним недоліком програмного забезпечення ZPL Warehouse Manager також є залежність від стабільного Інтернет з'єднання, без якого робота у хмарному середовищі стає неможливою. Крім того, обмежені можливості програмного забезпечення ZPL Warehouse Manager щодо інтеграції з деякими зовнішніми системами можуть ускладнювати роботу підприємств, які використовують нестандартні програмні рішення [10], [14].

Порівняльну характеристику програмного забезпечення для обліку та контролю руху обладнання наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для обліку та контролю руху обладнання

Критерій порівняння	Fishbowl [12]	Striven [13]	3PL Warehouse Manager [14]
Автоматизація складських процесів	+-	+-	+
Легкість освоєння функцій	+	+-	-
Інтеграція з бухгалтерськими системами	+	+	+-
Сканування штрих-кодів	+	+-	+
Хмарна доступність	+-	+	+
Інтеграція з транспортними службами	+	+-	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для обліку та контролю руху обладнання. Проте важливою перешкодою під час використання таких програмних систем може стати потреба в додаткових витратах на придбання ліцензій, обслуговування програмного продукту, технічну підтримку та навчання персоналу. Складність інтеграції зі сторонніми програмними системами також може знижувати ефективність програмного забезпечення. Не всі продукти забезпечують взаємодію з бухгалтерськими сервісами, системами логістики, закупівель або виробничого планування, що призводить до фрагментації інформаційного простору підприємства. У результаті дані залишаються розділеними між різними платформами, і використання

автоматизації не досягає очікуваного рівня. Тому актуальною є розробка програмного забезпечення для обліку та контролю руху обладнання.

При розробці програмного забезпечення для обліку та контролю руху обладнання необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення та редагування інформації про обладнання на підприємстві;
- підтримка можливості авторизації користувачів;
- можливість формування журналу операцій стосовно переміщень обладнання;
- можливість формування звітів;
- підтримка можливості експортування звітів у формат pdf;
- підтримка можливості відображення та редагування інформації про постачальників обладнання.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для обліку та контролю руху обладнання являє собою спеціалізовану систему, призначену для автоматизації процесів обліку, відстеження та управління обладнанням у різних організаціях. Воно дозволяє централізовано зберігати інформацію про наявність, місцезнаходження, стан та історію використання технічних засобів, що суттєво спрощує управління матеріальними ресурсами та зменшує ризик втрат чи помилок.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для обліку та контролю руху обладнання. Проте важливою перешкодою під час використання таких програмних систем може стати потреба в додаткових витратах на придбання ліцензій, обслуговування програмного продукту, технічну підтримку та навчання персоналу. Складність інтеграції зі сторонніми програмними системами також може знижувати ефективність програмного забезпечення. Не

всі продукти забезпечують взаємодію з бухгалтерськими сервісами, системами логістики, закупівель або виробничого планування, що призводить до фрагментації інформаційного простору підприємства. У результаті дані залишаються розділеними між різними платформами, і використання автоматизації не досягає очікуваного рівня. Тому актуальною є розробка програмного забезпечення для обліку та контролю руху обладнання.

Сформульовано функціональні вимоги до програмного забезпечення для обліку та контролю руху обладнання.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для обліку та контролю руху обладнання було використано мову програмування Python [15]-[16].

Мова програмування Python спішно використовується як у невеликих автоматизованих рішеннях, так і в масштабних корпоративних системах, що потребують складної логіки та інтеграції з різними сервісами. Її архітектура дозволяє формувати програми, здатні працювати на різних платформах без суттєвих змін у коді, що робить розроблені продукти доступними для широкого кола користувачів. Завдяки цьому забезпечується можливість швидкого розгортання систем обліку на різних пристроях та в різних операційних середовищах [15]-[16].

Важливою особливістю мови програмування Python є активна спільнота, яка постійно розширює можливості мови та підтримує нові стандарти. Наявність навчальних ресурсів, інструментів, бібліотек і прикладів успішних рішень створює сприятливе середовище для підтримки та розвитку вже створених програмних продуктів. Розробникам не потрібно витрачати значні зусилля на пошук складних рішень, адже багато інструментів уже існують і легко адаптуються під конкретні задачі. Така підтримка мови програмування Python сприяє тривалому життєвому циклу програм, що особливо важливо для систем, які використовуються у сфері управління матеріальними ресурсами [15]-[16].

Завдяки високій читабельності коду й простоті його модифікації мова програмування Python полегшує колективну розробку та супровід програмних продуктів. Коли систему обліку обладнання необхідно змінювати, доповнювати новими модулями або передавати іншій команді для підтримки, робота з кодом не перетворюється на тривалий процес аналізу й переосмислення механізмів реалізації. Це зменшує ризик помилок під час модернізації та дозволяє удосконалювати програмні засоби відповідно до

нових управлінських підходів без порушення вже встановленої функціональності [15]-[16].

Вибір мови програмування Python для створення систем обліку та контролю руху обладнання пов'язується з її здатністю забезпечувати швидку розробку функціональних рішень без надмірних витрат часу на реалізацію базових операцій. Завдяки зрозумілому синтаксису й гнучкій структурі програми можна створювати ефективні модулі обробки даних, не перевантажуючи їх технічними деталями. Це дозволяє зосередитися на бізнес-логіці, формуванні інструментів для реєстрації обладнання, фіксації його переміщень та генерації звітів, а не на складних механізмах реалізації внутрішніх процесів [15]-[16].

Мова програмування Python вирізняється широким вибором бібліотек і фреймворків, що підтримують роботу з базами даних, веб-інтерфейсами, мережевою взаємодією та інтеграцією з іншими сервісами. У результаті система обліку може розширюватися без потреби переписувати основні блоки, а її функціональність може змінюватися відповідно до потреб підприємства. Це забезпечує практичну перевагу, яка дозволяє формувати гнучке та масштабоване середовище для управління обладнанням на будь-якому етапі життєвого циклу проєкту [15]-[16].

Застосування мови програмування Python сприяє також швидкій адаптації до нових вимог, коли систему необхідно доповнити алгоритмами аналітики, прогнозування потреб або інструментами моніторингу технічного стану. Можливість підтримки таких функцій без суттєвих змін у загальній архітектурі робить цю мову інструментом для створення програмних продуктів, здатних розвиватися разом з організацією. Це перетворює Python на ефективну основу для розроблення сучасних систем контролю та обліку матеріально-технічних ресурсів [15]-[16].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для обліку та контролю руху обладнання наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для обліку та контролю руху обладнання

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	PHP
Простота та швидкість розробки	+	+—	+—
Читабельність коду та легкість супроводу	+	+	—
Гнучкість масштабування та розширення	+	+—	+
Зручність для створення ПЗ для обліку та контролю руху обладнання	+	—	—
Платформна незалежність	+	+—	+

Отже, для реалізації програмного забезпечення для обліку та контролю руху обладнання обрано мову програмування Python, яка забезпечує простоту розробки, широкий вибір бібліотек, можливості гнучкої інтеграції з аналітичними інструментами та низьким витратам на підтримку.

2.2 Вибір середовища розробки для створення програмного забезпечення для обліку та контролю руху обладнання

В якості середовища розробки для створення програмного забезпечення для обліку та контролю руху обладнання було обрано середовище PyCharm [17]-[18].

PyCharm надає для розробнику цілісне робоче середовище, у якому не потрібно постійно звертатися до зовнішніх інструментів чи окремих сервісів для виконання типових завдань. Розгорнуті можливості роботи з файловою системою, керування версіями, підключення віртуальних середовищ та

конфігурація інтерпретаторів забезпечують повноцінний цикл розробки без додаткових налаштувань. Завдяки цьому програмний продукт формується в упорядкованому просторі, де кожний компонент має своє місце і легко змінюється у разі необхідності [17]-[18].

PyCharm також вирізняється ефективною роботою з базами даних, дозволяючи редагувати структуру таблиць, виконувати запити та переглядати дані без переходу до окремих застосунків. Така можливість набуває особливої важливості під час розробки систем, що працюють із записами про обладнання, його переміщення, технічний стан і відповідальних осіб. Безпосередній доступ до даних під час написання коду дозволяє оперативно перевіряти коректність логіки обробки й уникати помилок, пов'язаних з неправильним моделюванням структури інформації [17]-[18].

Окрему перевагу становить використання інтелектуальних підказок, які допомагають орієнтуватися у великих програмних проєктах та знаходити оптимальні рішення завдяки аналізу коду в реальному часі. Таке середовище не лише підтримує написання програм, а й сприяє вдосконаленню стилю програмування, підвищуючи якість кінцевого продукту. Коли система обліку обладнання розвивається, інтегрує нові можливості й перетворюється на більш складний інструмент управління ресурсами, PyCharm допомагає зберігати цілісність проєкту й підтримувати його еволюцію без втрати зрозумілості та керованості [17]-[18].

Використання середовища PyCharm для розробки програмного забезпечення, спрямованого на облік та контроль руху обладнання, пов'язується з можливістю забезпечення стабільного та структурованого процесу створення коду. Це середовище надає інструменти, що полегшують роботу з мовою Python, роблячи написання й налагодження програм передбачуваним і зручним. Завдяки автоматичній перевірці синтаксису та підсвічуванню помилок знижуються ризики появи некоректних рішень у функціональних модулях, що відповідальні за опрацювання даних щодо переміщення, стану та характеристик обладнання [17]-[18].

PyCharm дозволяє організувати складні програмні проєкти таким чином, щоб різні компоненти системи зберігали логічну ієрархію та були доступними для модифікації. Коли виникає потреба вдосконалити алгоритми обліку, інтегрувати нові модулі моніторингу або змінити структуру бази даних, можливість швидко переходити між файлами та аналізувати взаємозв'язки між ними стає ключовою перевагою. Середовище автоматизує рутинні операції з керування залежностями, що спрощує роботу над масштабованими програмними продуктами [17]-[18].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для обліку та контролю руху обладнання наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для обліку та контролю руху обладнання

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Spyder	Sublime Text
Організація структури проєктів	+	—	—
Автоматизація тестування та налагодження	+	+—	+—
Інтеграція зі системами контролю версій	+	—	—
Інтелектуальні підказки та автодоповнення	+	+—	+—
Зручність для роботи з аналітикою	+	+	+—

Особливу цінність PyCharm створює завдяки підтримці інструментів тестування, роботи з репозиторіями та базами даних, що є важливими

складниками систем, орієнтованих на точний облік матеріальних ресурсів. Наявність цих можливостей у межах одного середовища забезпечує цілісність процесу розробки й дає змогу контролювати якість кожного етапу створення програмного продукту. Використання PyCharm сприяє формуванню надійного та масштабованого рішення для управління обладнанням, оскільки увага розробника зосереджується не на технічному налаштуванні інструментів, а на створенні ефективної логіки програмної системи [17]-[18].

Таким чином, для створення програмного забезпечення для обліку та контролю руху обладнання обрано середовище розробки PyCharm, що забезпечує розробнику розгорнуті можливості роботи з файловою системою, керування версіями, підключення віртуальних середовищ та конфігурація інтерпретаторів забезпечують повноцінний цикл розробки без додаткових налаштувань.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для обліку та контролю руху обладнання обрано мову програмування Python, яка забезпечує простоту розробки, широкий вибір бібліотек, можливості гнучкої інтеграції з аналітичними інструментами та низьким витратам на підтримку.

Для створення програмного забезпечення для обліку та контролю руху обладнання обрано середовище розробки PyCharm, що забезпечує розробнику розгорнуті можливості роботи з файловою системою, керування версіями, підключення віртуальних середовищ та конфігурація інтерпретаторів забезпечують повноцінний цикл розробки без додаткових налаштувань.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для обліку та контролю руху обладнання

Структуру програмного забезпечення для обліку та контролю руху обладнання наведено на рис. 3.1.

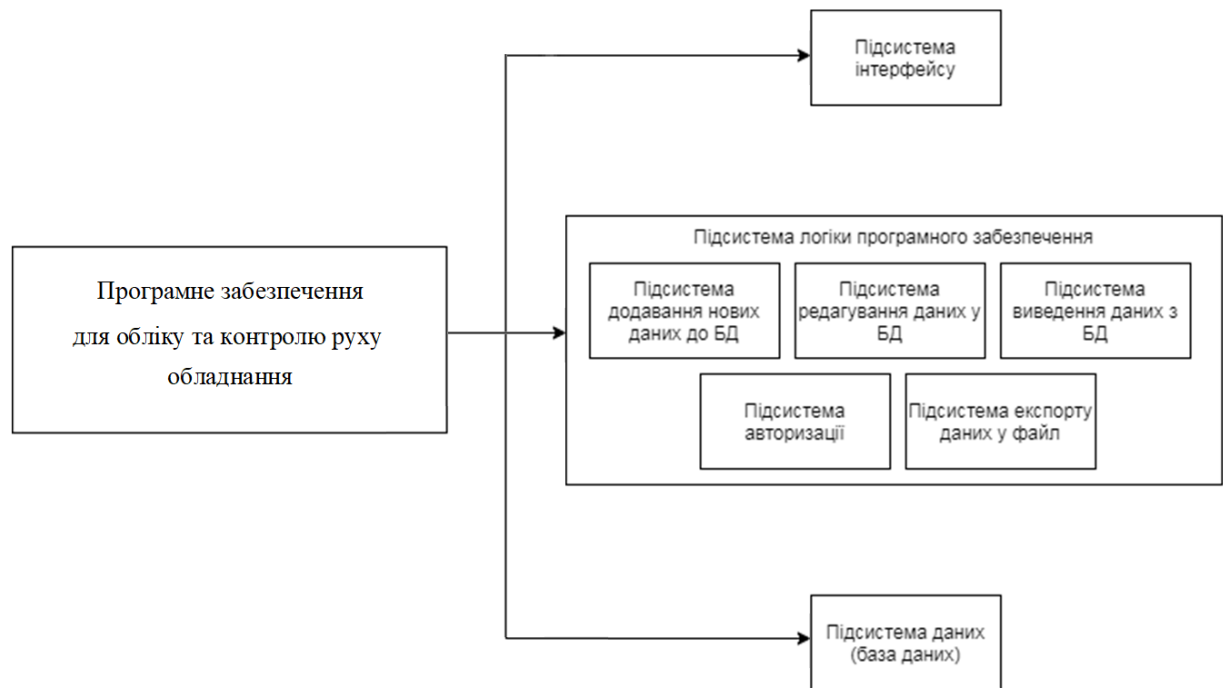


Рисунок 3.1 – Структура програмного забезпечення для обліку та контролю руху обладнання

Як видно з рис. 3.1, структура програмного забезпечення для обліку та контролю руху обладнання має комплексний характер і охоплює взаємопов'язані складові, кожна з яких виконує власну функцію та забезпечує повноцінне функціонування системи. Основу програмного забезпечення для обліку та контролю руху обладнання становлять підсистема інтерфейсу, підсистема програмної логіки та підсистема даних, тобто база даних, у якій зберігаються всі відомості про обладнання, його характеристики, переміщення та стан. Взаємодія цих частин забезпечує можливість введення, обробки й

подальшого використання інформації у процесах управління матеріальними ресурсами. Підсистема логіки складається з модуля додавання нових даних до бази даних, модуля редагування даних у базі даних, модуля виведення даних з бази даних, модуля авторизації, модуля експорту даних у файл.

Візуальна частина програмного забезпечення для обліку та контролю руху обладнання виконує роль середовища комунікації між користувачем та внутрішніми механізмами програми. Через інтерфейс здійснюється доступ до всіх інструментів контролю, перегляду та коригування даних. Від якості побудови цього компонента залежить зрозумілість роботи із системою, адже він має спрощувати виконання операцій, відображати актуальний стан обладнання та не перевантажувати користувача зайвими деталями. Інтерфейс слугує своєрідним посередником, який перетворює користувацькі дії на відповідні запити до інших частин програми.

Усередині застосунку розміщується підсистема логіки, що включає набір функціональних модулів, які забезпечують роботу з даними. Коли виникає потреба додати нові записи про обладнання, до роботи залучається модуль внесення даних, який формує правильні структури для збереження інформації в базі. Зміни у вже існуючих записах здійснюються через окремий компонент редагування, який відповідає за коректне оновлення відомостей без порушення існуючих зв'язків між сутностями. Перегляд наявної інформації забезпечується модулем виведення, що здійснює вибірку даних і представляє їх у вигляді, зручному для оцінки стану та переміщення ресурсів. Право доступу до всіх цих можливостей контролює модуль авторизації, який перевіряє особу користувача та визначає його рівень привілеїв. Крім того, система включає компонент експорту, завдяки якому дані можуть передаватися у зовнішні файли й використовуватися для звітності, аналізу чи резервного збереження поза межами програми.

Усі операції над інформацією пов'язані з роботою підсистеми даних, що зберігає та систематизує записи про обладнання. База даних виступає фундаментом програмного комплексу й гарантує цілісність, доступність і

захищеність інформаційного сховища. Саме в ній фіксуються характеристики предметів обліку, їх технічний стан, місце розташування, переміщення та відповідальні особи. Надійність цієї частини визначає безперервність функціонування програмного забезпечення, адже всі інші блоки лише отримують або змінюють те, що зберігається в структурованому вигляді в базі.

У сукупності згадані компоненти формують єдиний інструмент управління ресурсами, у якому кожен елемент виконує окрему роль, забезпечуючи точний облік і контроль руху обладнання у будь-якій організаційній структурі.

3.2 Функціонування програмного забезпечення для обліку та контролю руху обладнання

Функціонування програмного забезпечення для обліку та контролю руху обладнання подамо за допомогою схеми, зображеної на рис. 3.2.

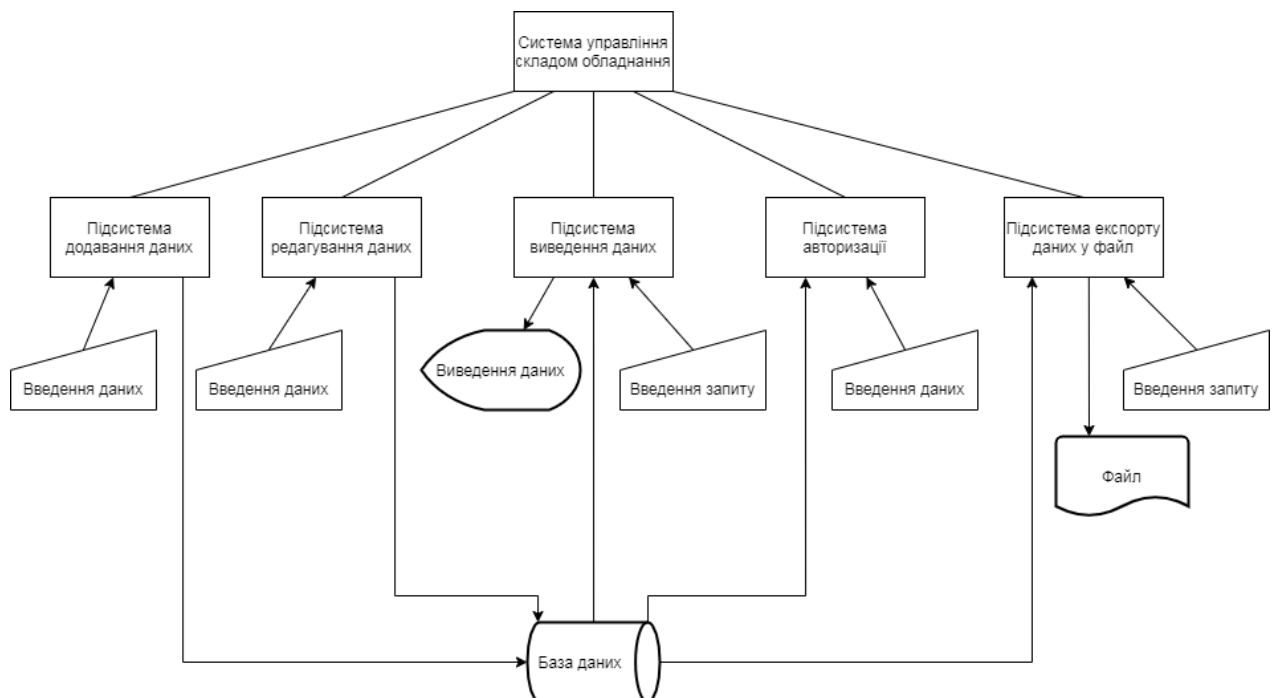


Рисунок 3.2 – Схема функціонування програмного забезпечення для обліку та контролю руху обладнання

Функціонування програмного забезпечення для обліку та контролю руху обладнання ґрунтується на послідовній взаємодії кількох внутрішніх механізмів, кожний з яких виконує окрему роль у роботі з інформацією. У процесі створення системи було сформовано набір процедур, що забезпечують внесення, перегляд, коригування, захист та передачу даних поза межі програми. Усі ці дії зводяться до опрацювання інформації, яка зберігається в базі даних, тому алгоритми обслуговування сховища визначають основні принципи роботи всієї системи.

Механізми, відповідальні за додавання нових відомостей і зміну вже існуючих записів, мають подібний принцип дії. Коли вводяться нові дані, програма спочатку перевіряє їх правильність, що унеможливорює внесення недостовірної або помилкової інформації. Після цього формується запит, який доповнює базу новим записом. При коригуванні відомостей система проходить таку саму перевірку, але замість створення нового об'єкта програмне забезпечення оновлює інформацію про той елемент, який було змінено користувачем. Таким чином, запобігається порушенню цілісності даних, і кожна зміна проходить попередній контроль.

Процедура відображення інформації передбачає вибірку потрібних даних залежно від умов, які задає користувач. Система не обмежується пошуком за єдиним критерієм і дозволяє поєднувати різні параметри. Спершу визначаються можливі варіанти для кожного фільтру, після чого ці множини уточнюються відповідно до введених значень. Лише після цього формується запит до бази даних із врахуванням уточнених параметрів, а отримані результати повертаються для подальшого відображення на екрані. Така структура вибірки дає можливість швидко знаходити конкретні відомості навіть у великому сховищі.

Щоб забезпечити доступ до даних лише авторизованим особам, система виконує процедуру перевірки користувача перед активацією будь-яких механізмів. Для цього у сховищі визначаються облікові записи, і введені дані порівнюються зі збереженими. Якщо співпадіння не виявлено, доступ до

можливостей системи блокується, а при успішній перевірці користувач отримує відповідні права.

Окремо функціонує процедура передачі даних у зовнішні файли. Система може генерувати документи різних форматів, використовуючи інформацію, що зберігається в базі. Для створення текстових файлів із табличною структурою спочатку формується вибірка, після чого відкривається новий файл і починається послідовне додавання даних до нього. Під час створення файлів зі складнішою візуалізацією використовуються шаблони, у які вставляються значення, отримані з бази. Завдяки цьому програма може надавати звіти в готовому вигляді, придатному для друку або перенесення в інші системи. У результаті всі алгоритми взаємодіють між собою, утворюючи єдиний механізм обліку та контролю руху обладнання.

3.3 Розробка бази даних програмного забезпечення для обліку та контролю руху обладнання

База даних у програмному забезпеченні для обліку та контролю руху обладнання формує цілісну модель предметної області, у якій зберігається інформація про матеріальні цінності, їх місцезнаходження, переміщення, постачальників та класифікаційні ознаки обладнання. Усі сутності взаємопов'язані між собою через зовнішні ключі, завдяки чому забезпечується безпомилкове простежування походження кожного об'єкта, його характеристик і історії операцій.

Схему бази даних програмного забезпечення для обліку та контролю руху обладнання наведено на рис. 3.3.

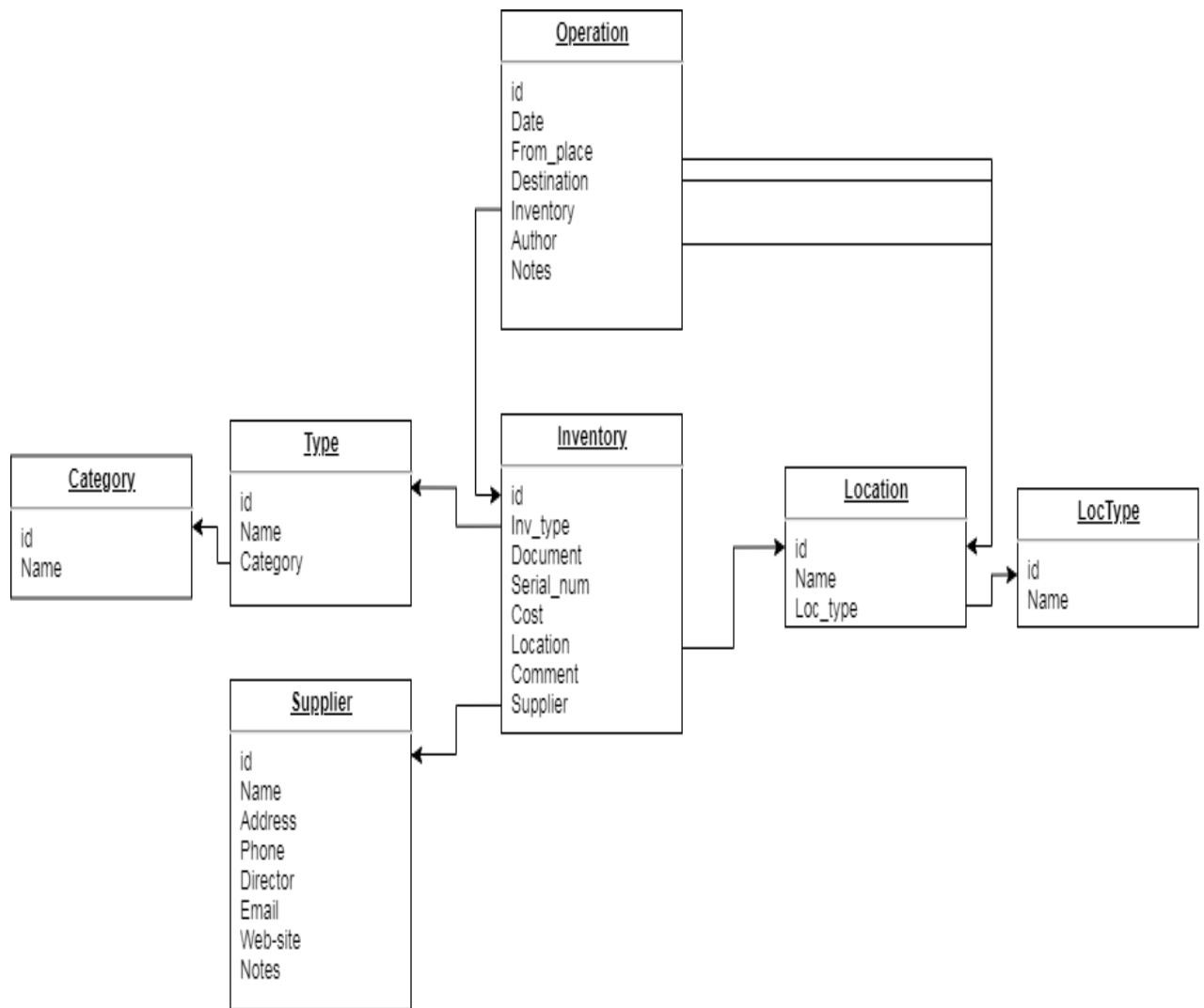


Рисунок 3.3 – Схема бази даних програмного забезпечення для обліку та контролю руху обладнання

У центрі структури перебуває сутність Inventory, яка представляє кожен одиницю обладнання окремим записом. Цій сутності відводиться збереження серійних номерів, вартості, файлів документів та коментарів, що відображають деталі придбання чи експлуатації. Через посилання на зовнішні таблиці в ній фіксується тип обладнання, його поточний постачальник та місце розташування, що дозволяє визначати приналежність конкретного інвентарного об'єкта до певної категорії, простежувати його походження та аналізувати належність до відповідних матеріальних груп.

Окрему роль відіграє сутність Operation, яка фіксує кожну подію, пов'язану з переміщенням чи зміною статусу обладнання. Завдяки зберіганню

часу здійснення операції, початкового та кінцевого місцезнаходження, а також відповідальної особи, забезпечується повна історія руху матеріальних цінностей. Така логіка таблиці дозволяє установлювати, коли саме відбулося перевезення, де зараз знаходиться об'єкт і хто підтвердив таку дію.

Сутність Location представляє фізичне місце перебування обладнання. Зберігаються назви приміщень, підрозділів чи осіб, у яких може знаходитися інвентар. Її тип визначається за допомогою таблиці LocType, де класифікуються такі місця за певними ознаками, наприклад як склади, робочі кабінети, тимчасові зони зберігання або працівники, відповідальні за конкретний об'єкт.

Визначення технічних характеристик обладнання забезпечується через сутність Type, яка деталізує назви та категорії інвентарних одиниць. Вона пов'язується з таблицею Category, де наведено узагальнені класи матеріальних ресурсів. Такий підхід гарантує коректну систематизацію великої кількості різноманітного обладнання, надаючи можливість групувати його за призначенням, класом або сферою використання.

Окремою складовою структури є сутність Supplier, де зберігається повний профіль постачальників обладнання. Контактна інформація, адреси, вебресурси та примітки створюють можливість не лише ідентифікувати джерело придбання, а й формувати звітність за виробниками та контрагентами, що постачають елементи облікуваних ресурсів.

Так сформована база даних працює узгоджено, забезпечуючи безперервність зв'язків між об'єктами, операціями, точками зберігання та зовнішніми джерелами постачання. Її схема дозволяє здійснювати пошук за широким набором параметрів, автоматизувати створення звітів і підтримувати точний контроль над рухом обладнання в межах будь-якої організації. Така модель гарантує, що жодна одиниця матеріальних цінностей не буде втрачена з обліку, а історія її використання залишиться повною, достовірною та доступною для аналізу.

3.4 Проектування інтерфейсу програмного забезпечення для обліку та контролю руху обладнання

Проектування інтерфейсу взаємодії користувача з програмним забезпеченням для обліку та контролю руху обладнання виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для обліку та контролю руху обладнання враховані функціональні вимоги до програми:

- підтримка можливості відображення та редагування інформації про обладнання на підприємстві;
- підтримка можливості авторизації користувачів;
- можливість формування журналу операцій стосовно переміщень обладнання;
- можливість формування звітів;
- підтримка можливості експортування звітів у формат pdf;
- підтримка можливості відображення та редагування інформації про постачальників обладнання.

Інтерфейс головної форми програмного забезпечення для обліку та контролю руху обладнання наведено на рис. 3.4.

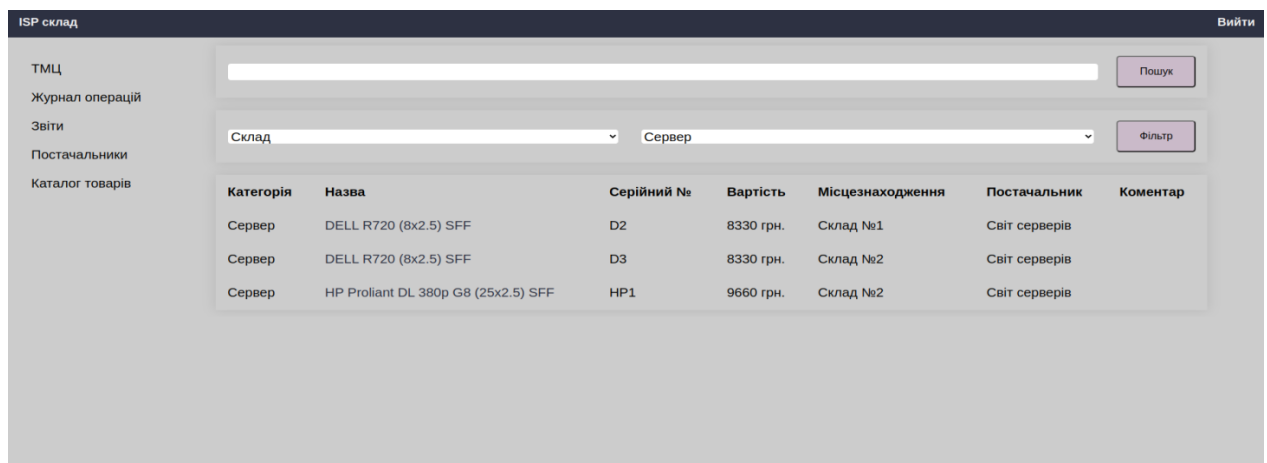


Рисунок 3.4 – Інтерфейс головної форми програмного забезпечення для обліку та контролю руху обладнання

Інтерфейс прототипу сторінки авторизації наведено на рис. 3.5.

ISP склад

Авторизація

Логін

Пароль

Увійти

Рисунок 3.5 – Інтерфейс прототипу сторінки авторизації

Сторінку з прикладом інформації про обладнання наведено на рис. 3.6.

ISP склад

Вийти

ТМЦ

Журнал операцій

Звіти

Постачальники

Каталог товарів

Найменування	DELL R720 (8x2.5) SFF
Документ	---
Серійний №	D2
Вартість	8330 грн.
Місцезнаходження	Склад №1
Коментар	
Постачальник	Світ серверів

Choose File No file chosen

Завантажити документ

Редагувати

№	Дата	Звідки	Куди	Автор	Примітки
5	28.05.2025	вул. Вишнева, 15	Склад №1	Петренко П. П.	

Рисунок 3.6 – Сторінку з прикладом інформації про обладнання

Сторінку журналу операцій з обладнанням наведено на рис. 3.7.

ISP склад

Вийти

ТМЦ

Журнал операцій

Звіти

Постачальники

Каталог товарів

Провести операцію

Звідки: вул. Вишнева, 15

Номер:

Куди: Склад №1

Автор: Петренко П. П.

Тип: DELL R720 (8x2.5) SFF

Фільтр

№	ID	Дата	Звідки	Куди	Найменування	Автор	Примітки
1	5	28.05.2025	вул. Вишнева, 15	Склад №1	DELL R720 (8x2.5) SFF	Петренко П. П.	

Рисунок 3.7 – Сторінка журналу операцій з обладнанням

Приклад звіту про обладнання наведено на рис. 3.8.

ISP склад

Склад

станом та 04.05.2026

Місцезнаходження	ТМЦ	Серійний номер	Вартість
Склад №1			8330 грн.
	DELL R720 (8x2.5) SFF	№D2	8330 грн.
Склад №2			17990 грн.
	HP Proliant DL 380p G8 (25x2.5) SFF	№HP1	9660 грн.
	DELL R720 (8x2.5) SFF	№D3	8330 грн.

Рисунок 3.8 – Приклад звіту про обладнання

Приклад сторінки з інформацією про постачальника наведено на рис. 3.9.

ISP склад		Вийти
ТМЦ	Найменування	TechnoShop
Журнал операцій	Адреса	м. Дніпро, вул. Барвінкова, 2
Звіти	Телефон	+38 (222) 222-22-22
Постачальники	Керівник	Іваненко І.І.
Каталог товарів	E-mail	technoshop@mail.com
	Веб-сайт	technoshop.com
	Примітки	Можна замовити обладнання

Рисунок 3.9 – Приклад сторінки з інформацією про постачальника

Сторінку каталогу товарів наведено на рис. 3.10.

ISP склад		Вийти
ТМЦ	UPS	
Журнал операцій	APC Smart-UPS SRT 1500VA Li-Ion	
Звіти	FSP Winner 700VA WINNER700	
Постачальники	Додати вид ТМЦ	
Каталог товарів	Маршрутизатор	
	LINKSYS E2500V4	
	MIKROTIK RB450	
	MIKROTIK RB450G	
	Додати вид ТМЦ	
	Сервер	
	DELL R720 (8x2.5) SFF	
	HP Proliant DL 380p G8 (25x2.5) SFF	
	Додати вид ТМЦ	

Рисунок 3.10 – Сторінка каталогу товарів

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для обліку та контролю руху обладнання.

Запропоновано структуру програмного забезпечення для обліку та контролю руху обладнання. Визначено, що основу програмного забезпечення

для обліку та контролю руху обладнання становлять підсистема інтерфейсу, підсистема програмної логіки та підсистема даних, тобто база даних, у якій зберігаються всі відомості про обладнання, його характеристики, переміщення та стан. Взаємодія цих частин забезпечує можливість введення, обробки й подальшого використання інформації у процесах управління матеріальними ресурсами. Підсистема логіки складається з модуля додавання нових даних до бази даних, модуля редагування даних у базі даних, модуля виведення даних з бази даних, модуля авторизації, модуля експорту даних у файл.

Описано функціонування програмного забезпечення для обліку та контролю руху обладнання. Функціонування програмного забезпечення для обліку та контролю руху обладнання ґрунтується на послідовній взаємодії кількох внутрішніх механізмів, кожний з яких виконує окрему роль у роботі з інформацією. У процесі створення програмної системи було сформовано набір процедур, що забезпечують внесення, перегляд, коригування, захист та передачу даних поза межі програми.

Розроблено базу даних програмного забезпечення для обліку та контролю руху обладнання, у якій зберігається інформація про матеріальні цінності, їх місцезнаходження, переміщення, постачальників та класифікаційні ознаки обладнання. Усі сутності взаємопов'язані між собою через зовнішні ключі, завдяки чому забезпечується безпомилкове простежування походження кожного об'єкта, його характеристик і історії операцій.

Описано особливості реалізації програмного забезпечення для обліку та контролю руху обладнання. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для обліку та контролю руху обладнання.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для обліку та контролю руху обладнання. Програма написана на мові Python, яка забезпечує простоту розробки, широкий вибір бібліотек, можливості гнучкої інтеграції з аналітичними інструментами та низьким витратам на підтримку. В якості середовища розробки для створення програмного забезпечення для обліку та контролю руху обладнання обрано PyCharm, що забезпечує розробнику розгорнуті можливості роботи з файловою системою, керування версіями, підключення віртуальних середовищ та конфігурація інтерпретаторів забезпечують повноцінний цикл розробки без додаткових налаштувань.

Програмне забезпечення для обліку та контролю руху обладнання забезпечує виконання таких функцій:

- підтримка можливості відображення та редагування інформації про обладнання на підприємстві;
- підтримка можливості авторизації користувачів;
- можливість формування журналу операцій стосовно переміщень обладнання;
- можливість формування звітів;
- підтримка можливості експортування звітів у формат pdf;
- підтримка можливості відображення та редагування інформації про постачальників обладнання.

Для роботи програмного забезпечення для обліку та контролю руху обладнання потрібні такі технічні та програмні ресурси: персональний комп'ютер або ноутбук з процесором, здатним забезпечувати стабільну обробку запитів до бази даних, тому доцільно використовувати центральний процесор із тактовою частотою не нижче ніж 2,0 ГГц або еквівалентним за продуктивністю багатоядерним рішенням. Для повноцінної роботи інтерфейсу

та обробки одночасних операцій з даними оптимальним вважається обсяг оперативної пам'яті не менше 4 ГБ, що дає змогу забезпечувати швидкість опрацювання інформації без істотних затримок; у випадку великої кількості записів бажано використовувати 8 ГБ.

Для збереження компонентів програми, бази даних і службових файлів рекомендується наявність вільного простору на накопичувачі обсягом від 1 ГБ. Бажаною є робота на твердотільному диску, оскільки швидкість читання та запису позитивно впливає на час обробки запитів при роботі з великим масивом інформації. Відеокарта високої продуктивності не є обов'язковою, достатньо базового графічного адаптера, що підтримує стандартний інтерфейс відображення даних.

ПЗ може функціонувати на операційних системах сімейства Windows, Linux або macOS за умови підтримки інтерпретатора Python та доступності середовища розробки або виконання, сумісного зі створеними модулями програми. Необхідним є встановлення бібліотек, що забезпечують роботу з базами даних, формування звітів і роботу з файлами різних форматів. За наявності доступу до мережі передбачається можливість оновлення компонентів та взаємодії зі сторонніми сервісами, проте автономний режим також залишається доступним.

Функціональні характеристики розробленого програмного забезпечення для обліку та контролю руху обладнання наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для обліку та контролю руху обладнання наведено у додатку Б.

4.2 Характеристики програми для обліку та контролю руху обладнання

Програмне забезпечення для обліку та контролю руху обладнання характеризується спрямованістю на систематизацію інформації про матеріальні цінності та точне відстеження їх місцезнаходження. Його робота

орієнтована на автоматизоване внесення, зберігання та коригування даних, що забезпечує високу достовірність відомостей і мінімізує вплив людського фактора. Інструментарій програми дозволяє підтримувати актуальний стан записів та оперативно реагувати на зміни у русі обладнання, запобігаючи втратам чи неточностям у документації.

У процесі функціонування програма для обліку та контролю руху обладнання демонструє стійку продуктивність та стабільність, що проявляється у швидкій обробці запитів та безперервності роботи під час інтенсивного використання. Передбачено механізми сортування, фільтрації та пошуку даних, що прискорюють отримання потрібної інформації незалежно від обсягів бази даних. Така властивість програми для обліку та контролю руху обладнання є ключовою при роботі з великою кількістю об'єктів і складним рухом між різними місцями зберігання.

Особливістю програмного продукту є підтримка захисту даних, що виражається у наявності авторизації та надання доступу до інформації лише ідентифікованим користувачам. Таким чином забезпечується персональна відповідальність за виконані дії та запобігається несанкціонованому втручанню. Завдяки цьому система не лише зберігає інформацію, а й гарантує її безпечність.

Програмне забезпечення для обліку та контролю руху обладнання також відзначається гнучкістю, оскільки здатна працювати із різними типами обладнання та місцезнаходжень, підтримуючи різні категорії об'єктів і постачальників. Можливість формування звітів та експорту даних у поширені формати забезпечує інтеграцію із зовнішніми системами документообігу та дає змогу використовувати результати роботи для аналізу та планування. Усе це формує програмний продукт, здатний адаптуватися до особливостей конкретного підприємства і відповідати його практичним потребам у зручному керуванні ресурсами.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для обліку та контролю руху обладнання на серверній частини необхідно запустити відповідний файл. Для звертання до клієнтської сторони необхідно перейти на відповідну вебсторінку.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для обліку та контролю руху обладнання є відомості про наявне обладнання, його характеристики, серійні номери, вартість, а також дані про місце зберігання, постачальників та зміни у місцезнаходженні. У процесі оформлення операцій переміщення користувач заносить дані про відправну та кінцеву локацію, відповідального працівника та інші супровідні примітки. При авторизації вводяться реєстраційні дані, що дозволяють ідентифікувати користувача та надати йому відповідний рівень доступу.

Вихідними даними програмного забезпечення для обліку та контролю руху обладнання є сформовані та оброблені системою відомості, які відображаються на екрані або експортуються у вигляді звітів. Серед них можуть бути списки обладнання, його поточне місцезнаходження, історія переміщень, інформація про постачальників та сформовані аналітичні вибірки згідно з умовами фільтрації.

4.3.3 Повідомлення

Користувач програмного забезпечення для обліку та контролю руху обладнання може отримати такі повідомлення: повідомлення про успішне внесення, редагування або видалення записів, повідомлення про генерацію

звітів. У разі некоректного введення даних виводиться попередження про необхідність уточнення або виправлення інформації. При невдалій авторизації користувач отримує повідомлення про відмову в доступі.

4.4 Виконання програмного забезпечення для обліку та контролю руху обладнання

Після запуску програмного забезпечення для обліку та контролю руху обладнання користувачу необхідно авторизуватися, ввівши свій логін та пароль (рис. 4.1).

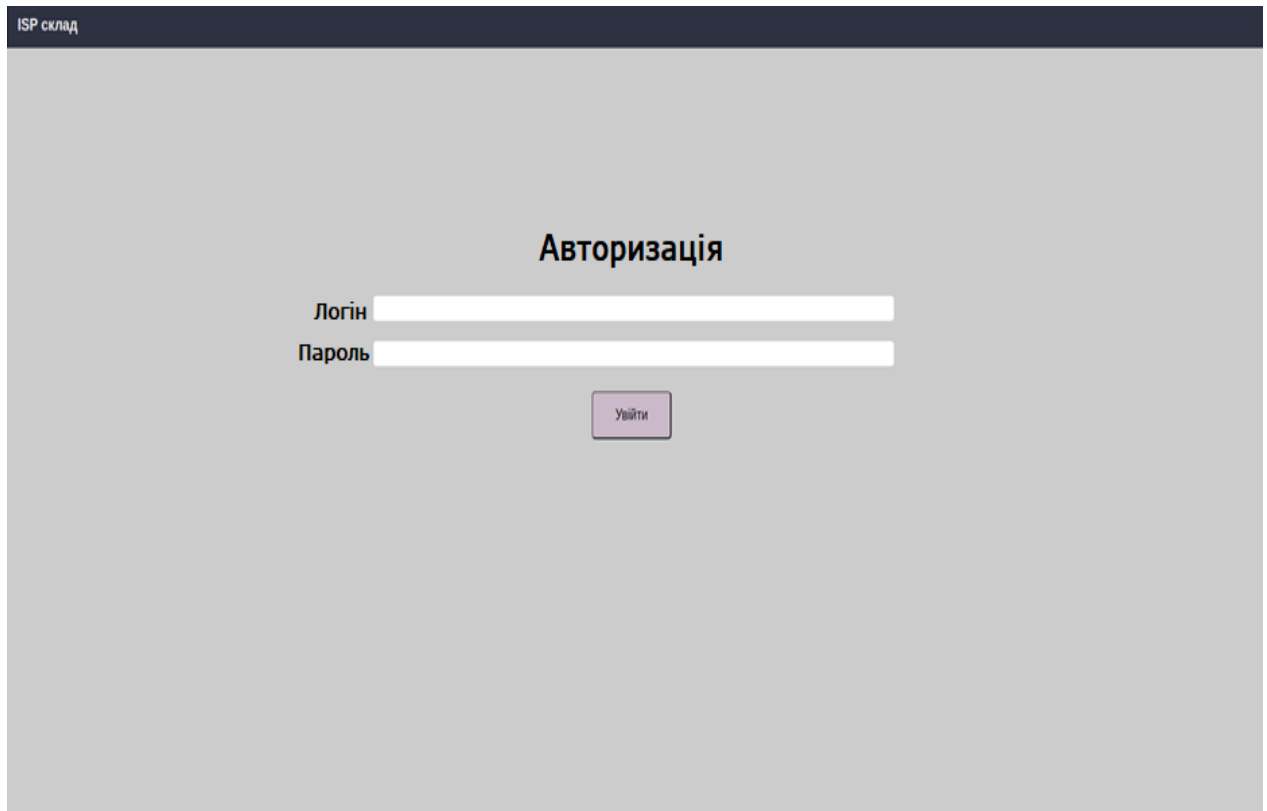


Рисунок 4.1 – Інтерфейс сторінки авторизації програмного забезпечення для обліку та контролю руху обладнання

Після авторизації відображається головна сторінка (рис. 4.2), де наводиться таблиця з обладнанням.

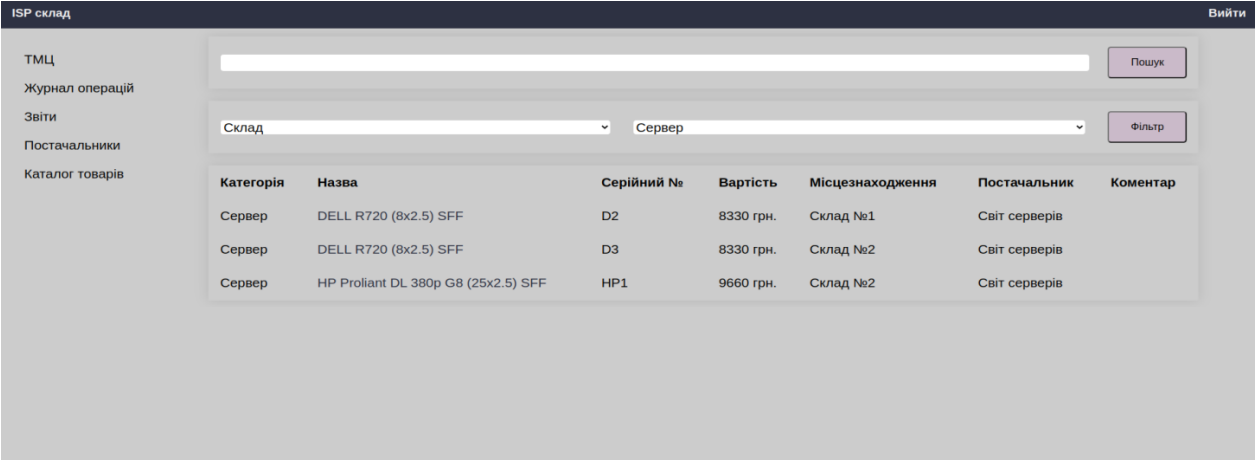


Рисунок 4.2 – Головна сторінка програмного забезпечення для обліку та контролю руху обладнання

При виборі конкретного найменування обладнання у таблиці відбувається перехід до сторінки з описом цього обладнання (рис. 4.3), де наводиться його найменування, серійний номер, вартість, місцезнаходження та інша інформація. Також існує можливість завантаження документів, що стосуються цього обладнання.

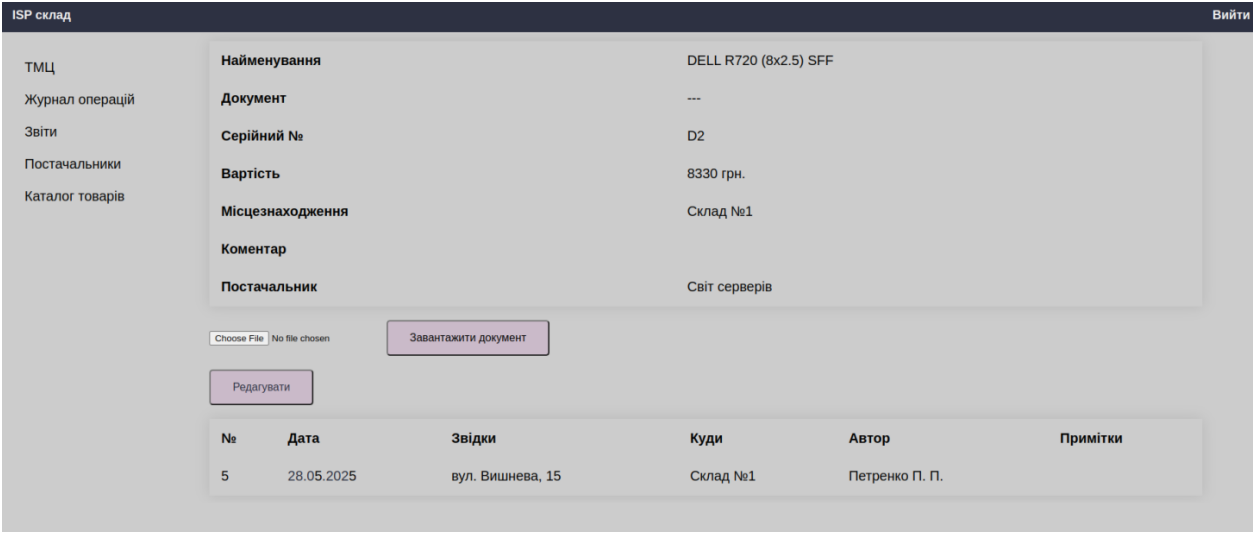


Рисунок 4.3 – Інформація про обладнання

При натисканні кнопки Редагувати відбувається перехід до форми (рис. 4.4), де за необхідністю потрібно ввести нові характеристики обладнання (або залишити існуючі).

ISP склад Вийти

ТМЦ

Журнал операцій

Звіти

Постачальники

Каталог товарів

Тип: DELL R720 (8x2.5) SFF

Серійний номер: D2

Вартість (грн.): 8330

Місцезнаходження: Склад №1

Постачальник: Світ серверів

Примітки:

Зберегти

Рисунок 4.4 – Редагування інформації про одиницю обладнання

При виборі пункту меню Журнал операцій (рис. 4.5) відображається перелік переміщень обладнання, зокрема, надається інформація про місця, звідки та куди було зроблено переміщення, а також інформація про відповідальну за переміщення особу.

ISP склад Вийти

ТМЦ

Журнал операцій

Звіти

Постачальники

Каталог товарів

Провести операцію

Звідки: вул. Вишнева, 15

Куди: Склад №1

Тип: DELL R720 (8x2.5) SFF

Номер:

Автор: Петренко П. П.

Фільтр

№	ID	Дата	Звідки	Куди	Найменування	Автор	Примітки
1	5	28.05.2025	вул. Вишнева, 15	Склад №1	DELL R720 (8x2.5) SFF	Петренко П. П.	

Рисунок 4.5 – Журнал операцій з обладнанням

При виборі пункту меню Звіти (рис. 4.6) необхідно ввести параметри, за якими формується звіт, а також обрати формат звіту для експорту.

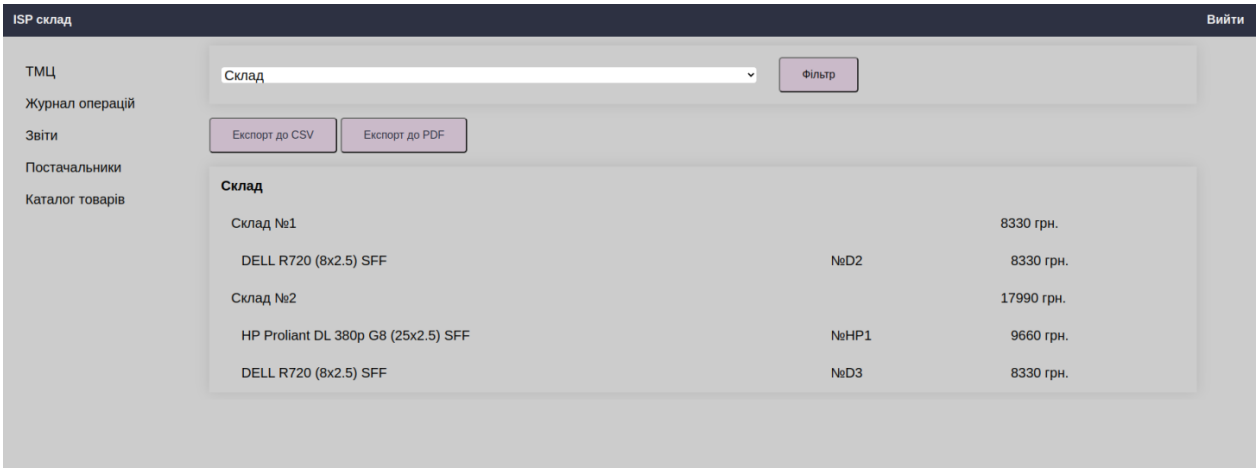


Рисунок 4.6 – Формування звіту

Приклад експортованого звіту у формат pdf наведено на рис. 4.7.

ISP склад

Склад
станом та 04.05.2026

Місцезнаходження	ТМЦ	Серійний номер	Вартість
Склад №1			8330 грн.
	DELL R720 (8x2.5) SFF	№D2	8330 грн.
Склад №2			17990 грн.
	HP Proliant DL 380p G8 (25x2.5) SFF	№HP1	9660 грн.
	DELL R720 (8x2.5) SFF	№D3	8330 грн.

Рисунок 4.7 – Приклад звіту

На сторінці Постачальники (рис. 4.8) міститься інформація про постачальників обладнання.

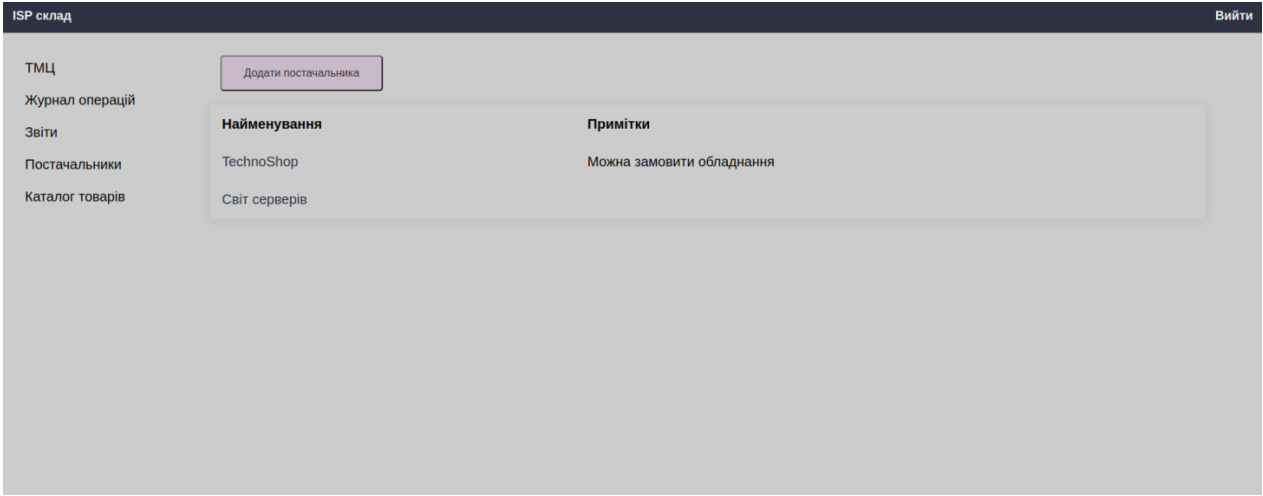


Рисунок 4.8 – Сторінка постачальників

При виборі конкретного постачальника відображається інформація про нього (рис. 4.9), зокрема, його адреса, вебсайт, телефон та ін.

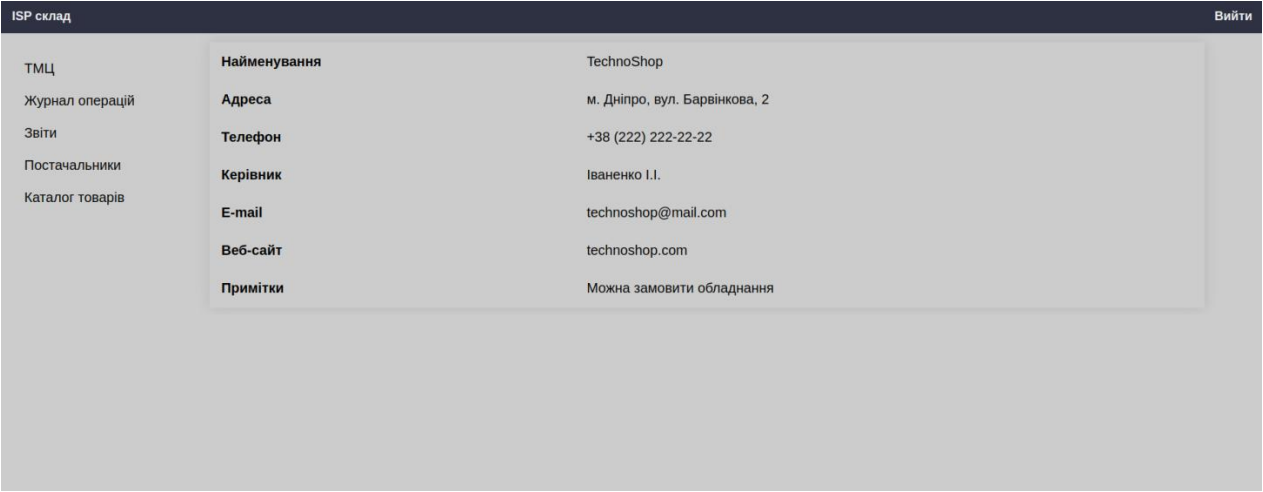


Рисунок 4.9 – Інформація про постачальника

При натисканні на відповідній сторінці кнопки Додати постачальника відображається форма (рис. 4.10), де користувач вводить інформацію про нового постачальника.

ISP склад

Вийти

ТМЦ
Журнал операцій
Звіти
Постачальники
Каталог товарів

ПІБ:

Адреса:

Телефон:

Директор:

Ел. пошта:

Сайт:

Примітки:

Додати

Рисунок 4.10 – Додавання нового постачальника

При переході за посиланням Каталог товарів (рис. 4.11) відображається перелік видів обладнання за категоріями.

ISP склад

Вийти

ТМЦ
Журнал операцій
Звіти
Постачальники
Каталог товарів

UPS
APC Smart-UPS SRT 1500VA Li-Ion
FSP Winner 700VA WINNER700
Додати вид ТМЦ

Маршрутизатор
LINKSYS E2500V4
MIKROTIK RB450
MIKROTIK RB450G
Додати вид ТМЦ

Сервер
DELL R720 (8x2.5) SFF
HP Proliant DL 380p G8 (25x2.5) SFF
Додати вид ТМЦ

Рисунок 4.11 – Каталог товарів

При виборі конкретного виду обладнання існує можливість додати одиницю обраного товару (рис. 4.12).

Рисунок 4.12 – Додавання обладнання

4.5 Тестування програмного забезпечення для обліку та контролю руху обладнання

Виконано тестування програмного забезпечення для обліку та контролю руху обладнання.

Виявлено, що програма для обліку та контролю руху обладнання функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу управління рухом обладнання.

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

4.6 Висновки за розділом 4

Описано програмне забезпечення для обліку та контролю руху обладнання. Виконано тестування розробленого програмного забезпечення для обліку та контролю руху обладнання. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління рухом обладнання.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для обліку та контролю руху обладнання.

Визначено, що програмне забезпечення для обліку та контролю руху обладнання являє собою спеціалізовану систему, призначену для автоматизації процесів обліку, відстеження та управління обладнанням у різних організаціях. Воно дозволяє централізовано зберігати інформацію про наявність, місцезнаходження, стан та історію використання технічних засобів, що суттєво спрощує управління матеріальними ресурсами та зменшує ризик втрат чи помилок.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для обліку та контролю руху обладнання. Проте важливою перешкодою під час використання таких програмних систем може стати потреба в додаткових витратах на придбання ліцензій, обслуговування програмного продукту, технічну підтримку та навчання персоналу. Складність інтеграції зі сторонніми програмними системами також може знижувати ефективність програмного забезпечення. Не всі продукти забезпечують взаємодію з бухгалтерськими сервісами, системами логістики, закупівель або виробничого планування, що призводить до фрагментації інформаційного простору підприємства. У результаті дані залишаються розділеними між різними платформами, і використання автоматизації не досягає очікуваного рівня. Тому актуальною є розробка програмного забезпечення для обліку та контролю руху обладнання.

Сформульовано функціональні вимоги до програмного забезпечення для обліку та контролю руху обладнання.

Для реалізації програмного забезпечення для обліку та контролю руху обладнання обрано мову програмування Python, яка забезпечує простоту

розробки, широкий вибір бібліотек, можливості гнучкої інтеграції з аналітичними інструментами та низьким витратам на підтримку.

Для створення програмного забезпечення для обліку та контролю руху обладнання обрано середовище розробки PyCharm, що забезпечує розробнику розгорнуті можливості роботи з файловою системою, керування версіями, підключення віртуальних середовищ та конфігурація інтерпретаторів забезпечують повноцінний цикл розробки без додаткових налаштувань.

Розроблено програмне забезпечення для обліку та контролю руху обладнання.

Запропоновано структуру програмного забезпечення для обліку та контролю руху обладнання. Визначено, що основу програмного забезпечення для обліку та контролю руху обладнання становлять підсистема інтерфейсу, підсистема програмної логіки та підсистема даних, тобто база даних, у якій зберігаються всі відомості про обладнання, його характеристики, переміщення та стан. Взаємодія цих частин забезпечує можливість введення, обробки й подальшого використання інформації у процесах управління матеріальними ресурсами. Підсистема логіки складається з модуля додавання нових даних до бази даних, модуля редагування даних у базі даних, модуля виведення даних з бази даних, модуля авторизації, модуля експорту даних у файл.

Описано функціонування програмного забезпечення для обліку та контролю руху обладнання. Функціонування програмного забезпечення для обліку та контролю руху обладнання ґрунтується на послідовній взаємодії кількох внутрішніх механізмів, кожний з яких виконує окрему роль у роботі з інформацією. У процесі створення програмної системи було сформовано набір процедур, що забезпечують внесення, перегляд, коригування, захист та передачу даних поза межі програми.

Розроблено базу даних програмного забезпечення для обліку та контролю руху обладнання, у якій зберігається інформація про матеріальні цінності, їх місцезнаходження, переміщення, постачальників та класифікаційні ознаки обладнання. Усі сутності взаємопов'язані між собою через зовнішні

ключі, завдяки чому забезпечується безпомилкове простежування походження кожного об'єкта, його характеристик і історії операцій.

Описано особливості реалізації програмного забезпечення для обліку та контролю руху обладнання. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для обліку та контролю руху обладнання.

Виконано тестування розробленого програмного забезпечення для обліку та контролю руху обладнання. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу управління рухом обладнання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is Inventory Management Software? Analysis of Features, Types, Benefits and Pricing. URL: <https://financesonline.com/inventory-management-software-analysis-features-types-benefits-pricing/> (date of access: 19.04.2026).
2. Understanding Inventory Management Software – A Step-by-Step Guide. URL: <https://smallbiztrends.com/what-is-inventory-management-software/> (date of access: 19.04.2026).
3. What Is IT Inventory Management & Why Is It Important. URL: <https://growrk.com/blog/it-inventory-management> (date of access: 19.04.2026).
4. Automated Inventory Management. Key Benefits And Features. URL: <https://www.softwaresuggest.com/blog/automated-inventory-management/> (date of access: 19.04.2026).
5. What is Inventory Software. URL: <https://www.zoho.com/inventory/academy/inventory-management/what-is-inventory-software.html> (date of access: 19.04.2026).
6. A Complete Guide on Warehouse Inventory Management Software. URL: <https://www.deskera.com/blog/warehouse-inventory-management-software/> (date of access: 19.04.2026).
7. Best Warehouse Management Software. URL: <https://www.softwareadvice.com/scm/warehouse-management-system-comparison/> (date of access: 19.04.2026).
8. Top 10 Warehouse Optimization Software Tools. URL: <https://www.linnworks.com/blog/warehouse-optimization-software/> (date of access: 19.04.2026).
9. Best Warehouse Management Software for Small Business. URL: <https://www.ecommerceceo.com/wms-software-for-small-business/> (date of access: 19.04.2026).

10. The Best Warehouse Management Software. URL: <https://softwareconnect.com/roundups/best-warehouse-management-software/> (date of access: 19.04.2026).

11. Best Warehouse Management Software. URL: <https://www.capterra.com/warehouse-management-software/> (date of access: 19.04.2026).

12. Fishbowl. URL: <https://www.fishbowlinventory.com/fishbowl-inventory> (date of access: 19.04.2026).

13. Striven. URL: <https://www.striven.com/solutions/features/inventory-management-software> (date of access: 19.04.2026).

14. 3PL Warehouse Manager. URL: <https://www.extensiv.com/extensiv-3pl-warehouse-manager> (date of access: 19.04.2026).

15. How to Use Python: Your First Steps. URL: <https://realpython.com/python-first-steps/> (date of access: 19.04.2026).

16. The Python Tutorial. URL: <https://docs.python.org/3/tutorial/index.html> (date of access: 19.04.2026).

17. Getting Started With PyCharm for Python GUI Development. URL: <https://www.pythonguis.com/tutorials/getting-started-pycharm/> (date of access: 19.04.2026).

18. PyCharm The only Python IDE you need. URL: <https://www.jetbrains.com/pycharm/> (date of access: 19.04.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для обліку та контролю руху обладнання.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для обліку та контролю руху обладнання», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 07 квітня 2026 р.

A.2 Призначення розробки

Програмний продукт призначений для обліку та контролю руху обладнання.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для обліку та контролю руху обладнання забезпечує виконання таких функцій:

- підтримка можливості відображення та редагування інформації про обладнання на підприємстві;
- підтримка можливості авторизації користувачів;
- можливість формування журналу операцій стосовно переміщень обладнання;
- можливість формування звітів;
- підтримка можливості експортування звітів у формат pdf;

– підтримка можливості відображення та редагування інформації про постачальників обладнання.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для обліку та контролю руху обладнання повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для обліку та контролю руху обладнання повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для обліку та контролю руху обладнання необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для обліку та контролю руху обладнання потрібні такі технічні та програмні ресурси: персональний комп'ютер або ноутбук з процесором, здатним забезпечувати стабільну обробку запитів до бази даних, тому доцільно використовувати центральний процесор із тактовою частотою не нижче ніж 2,0 ГГц або еквівалентним за продуктивністю багатоядерним рішенням. Для повноцінної роботи інтерфейсу та обробки одночасних операцій з даними оптимальним вважається обсяг

оперативної пам'яті не менше 4 ГБ, що дає змогу забезпечувати швидкість опрацювання інформації без істотних затримок; у випадку великої кількості записів бажано використовувати 8 ГБ.

Для збереження компонентів програми, бази даних і службових файлів рекомендується наявність вільного простору на накопичувачі обсягом від 1 ГБ. Бажаною є робота на твердотільному диску, оскільки швидкість читання та запису позитивно впливає на час обробки запитів при роботі з великим масивом інформації. Відеокарта високої продуктивності не є обов'язковою, достатньо базового графічного адаптера, що підтримує стандартний інтерфейс відображення даних.

ПЗ може функціонувати на операційних системах сімейства Windows, Linux або macOS за умови підтримки інтерпретатора Python та доступності середовища розробки або виконання, сумісного зі створеними модулями програми. Необхідним є встановлення бібліотек, що забезпечують роботу з базами даних, формування звітів і роботу з файлами різних форматів. За наявності доступу до мережі передбачається можливість оновлення компонентів та взаємодії зі сторонніми сервісами, проте автономний режим також залишається доступним.

A.3.6 Вимоги до маркування і пакування

Програма для обліку та контролю руху обладнання може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для обліку та контролю руху обладнання».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для обліку та контролю руху обладнання є відомості про наявне обладнання, його характеристики,

серійні номери, вартість, а також дані про місце зберігання, постачальників та зміни у місцезнаходженні. У процесі оформлення операцій переміщення користувач заносить дані про відправну та кінцеву локацію, відповідального працівника та інші супровідні примітки. При авторизації вводяться реєстраційні дані, що дозволяють ідентифікувати користувача та надати йому відповідний рівень доступу.

Вихідними даними програмного забезпечення для обліку та контролю руху обладнання є сформовані та оброблені системою відомості, які відображаються на екрані або експортуються у вигляді звітів. Серед них можуть бути списки обладнання, його поточне місцезнаходження, історія переміщень, інформація про постачальників та сформовані аналітичні вибірки згідно з умовами фільтрації.

ДОДАТОК Б
Фрагмент тексту програми

```

class Invnt(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    nm = db.Column(db.String(100), nullable=False)
    tp = db.Column(db.String(50), nullable=False)
    sl_num = db.Column(db.String(50), nullable=True)
    cst = db.Column(db.Integer, nullable=False)
    loc = db.Column(db.String(100), nullable=False)
    cmnt = db.Column(db.String(200), nullable=True)
    spr = db.Column(db.String(100), nullable=False)

class Oprt(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    dt = db.Column(db.String(50), nullable=False)
    frm_loc = db.Column(db.String(100), nullable=False)
    to_loc = db.Column(db.String(100), nullable=False)
    invnt_id = db.Column(db.Integer, db.ForeignKey('invnt.id'),
nullable=False)
    authr = db.Column(db.String(100), nullable=False)
    nts = db.Column(db.String(200), nullable=True)

def add_invnt():
    if request.method == "POST":
        nm = request.form['nm']
        tp = request.form['tp']
        sl_num = request.form['sl_num']
        cst = request.form['cst']
        loc = request.form['loc']
        cmnt = request.form['cmnt']
        spr = request.form['spr']
        nw_invnt = Invnt(nm=nm, tp=tp, sl_num=sl_num, cst=cst,
loc=loc, cmnt=cmnt, spr=spr)
        db.session.add(nw_invnt)
        db.session.commit()
        return redirect(url_for("rts.hme"))
    return render_template("add_inventory.html")

def view_invnt(id):
    invnt = Invnt.query.get_or_404(id)
    oprts = Oprt.query.filter_by(invnt_id=id).all()
    return render_template("view_inventory.html", invnt=invnt,
oprts=oprts)

def add_oprt(id):
    invnt = Invnt.query.get_or_404(id)
    if request.method == "POST":
        dt = request.form['dt']
        frm_loc = request.form['frm_loc']

```

```

        to_loc = request.form['to_loc']
        authr = request.form['authr']
        nts = request.form['nts']
        nw_oprt = Oprt(dt=dt, frm_loc=frm_loc, to_loc=to_loc,
invnt_id=id, authr=authr, nts=nts)
        db.session.add(nw_oprt)
        db.session.commit()
        return redirect(url_for("rts.view_invnt", id=id))
    return render_template("add_inventory.html", invnt=invnt)

def cr_usr(nm, psw):
    from models_module import Usr
    hshd_psw = generate_password_hash(psw)
    nw_usr = Usr(nm=nm, psw=hshd_psw)
    db.session.add(nw_usr)
    db.session.commit()
    return nw_usr

def chk_usr_lgn(nm, psw):
    from models_module import Usr
    usr = Usr.query.filter_by(nm=nm).first()
    if usr and check_password_hash(usr.psw, psw):
        session['usr_id'] = usr.id
        return True
    return False

def lgt_usr():
    session.pop('usr_id', None)
    return True

def srch_invnt(by_nm=None, by_tp=None, by_loc=None):
    q = Invnt.query
    if by_nm:
        q = q.filter(Invnt.nm.contains(by_nm))
    if by_tp:
        q = q.filter(Invnt.tp.contains(by_tp))
    if by_loc:
        q = q.filter(Invnt.loc.contains(by_loc))
    return q.all()

def fltr_oprt(by_dt=None, by_frm_loc=None, by_to_loc=None):
    q = Oprt.query
    if by_dt:
        q = q.filter(Oprt.dt.contains(by_dt))
    if by_frm_loc:
        q = q.filter(Oprt.frm_loc.contains(by_frm_loc))
    if by_to_loc:
        q = q.filter(Oprt.to_loc.contains(by_to_loc))

```

```

return q.all()

def del_invnt(invnt_id):
    invnt = Invnt.query.get(invnt_id)
    if invnt:
        oprts = Oprt.query.filter_by(invnt_id=invnt_id).all()
        for oprt in oprts:
            db.session.delete(oprt)
        db.session.delete(invnt)
        db.session.commit()
        return True
    return False

def upd_invnt(invnt_id, nm=None, tp=None, sl_num=None, cst=None,
loc=None, cmnt=None, spr=None):
    invnt = Invnt.query.get(invnt_id)
    if not invnt:
        return False
    if nm:
        invnt.nm = nm
    if tp:
        invnt.tp = tp
    if sl_num:
        invnt.sl_num = sl_num
    if cst:
        invnt.cst = cst
    if loc:
        invnt.loc = loc
    if cmnt:
        invnt.cmnt = cmnt
    if spr:
        invnt.spr = spr
    db.session.commit()
    return True

def add_oprt_log(invnt_id, dt, frm_loc, to_loc, authr, nts=None):
    nw_oprt = Oprt(dt=dt, frm_loc=frm_loc, to_loc=to_loc,
invnt_id=invnt_id, authr=authr, nts=nts)
    db.session.add(nw_oprt)
    db.session.commit()
    return nw_oprt

def get_invnt_history(invnt_id):
    return
Oprt.query.filter_by(invnt_id=invnt_id).order_by(Oprt.dt.desc()).
all()

def gen_rpt(by_loc=None):

```

```

q = Invnt.query
if by_loc:
    q = q.filter(Invnt.loc.contains(by_loc))
rpt_data = []
for invnt in q.all():
    history = get_invnt_history(invnt.id)
    rpt_data.append({
        'invnt': invnt,
        'history': history
    })
return rpt_data

def bulk_add_invnts(invnts_list):
    for itm in invnts_list:
        nw_invnt = Invnt(
            nm=itm.get('nm'),
            tp=itm.get('tp'),
            sl_num=itm.get('sl_num'),
            cst=itm.get('cst'),
            loc=itm.get('loc'),
            cmnt=itm.get('cmnt'),
            spr=itm.get('spr')
        )
        db.session.add(nw_invnt)
    db.session.commit()
    return True

def chk_invnt_avlbl(invnt_id):
    invnt = Invnt.query.get(invnt_id)
    if not invnt:
        return False
    oprt_history =
Oprt.query.filter_by(invnt_id=invnt_id).order_by(Oprt.dt.desc()).
first()
    if oprt_history and oprt_history.to_loc.lower() == "is":
        return True
    return False

def trnsf_invnt(invnt_id, new_loc, authr, nts=None):
    invnt = Invnt.query.get(invnt_id)
    if not invnt:
        return False
    nw_oprt = Oprt(dt="now", frm_loc=invnt.loc, to_loc=new_loc,
invnt_id=invnt_id, authr=authr, nts=nts)
    invnt.loc = new_loc
    db.session.add(nw_oprt)
    db.session.commit()
    return True

```

```

def get_invnt_by_tp(tp):
    return Invnt.query.filter_by(tp=tp).all()

def gen_csv_rpt(by_loc=None):
    output = io.StringIO()
    writer = csv.writer(output)
    writer.writerow(['ID', 'Name', 'Type', 'Serial', 'Cost',
'Location', 'Supplier', 'Comment'])
    q = Invnt.query
    if by_loc:
        q = q.filter(Invnt.loc.contains(by_loc))
    for invnt in q.all():
        writer.writerow([invnt.id, invnt.nm, invnt.tp,
invnt.sl_num, invnt.cst, invnt.loc, invnt.spr, invnt.cmnt])
    output.seek(0)
    return send_file(io.BytesIO(output.getvalue().encode()),
attachment_filename="inventory_report.csv", as_attachment=True)

def calc_totl_cst(by_loc=None):
    q = Invnt.query
    if by_loc:
        q = q.filter(Invnt.loc.contains(by_loc))
    totl = sum([invnt.cst for invnt in q.all()])
    return totl

def get_invnts_by_spr(spr):
    return Invnt.query.filter_by(spr=spr).all()

def bulk_trnsf(invnts_ids, new_loc, authr):
    for invnt_id in invnts_ids:
        trnsf_invnt(invnt_id, new_loc, authr)
    return True

def merge_oprt_data(oprt_list):
    merged = {}
    for oprt in oprt_list:
        if oprt.invnt_id not in merged:
            merged[oprt.invnt_id] = []
        merged[oprt.invnt_id].append({'dt': oprt.dt, 'frm_loc':
oprt.frm_loc, 'to_loc': oprt.to_loc, 'authr': oprt.authr})
    return merged

def get_oprts_by_dt_range(start_dt, end_dt):
    return Oprt.query.filter(Oprt.dt >= start_dt, Oprt.dt <=
end_dt).all()

def backup_db():

```

```

    invnts = Invnt.query.all()
    oprts = Oprt.query.all()
    backup_data = {
        'invnts': [{c.name: getattr(i, c.name) for c in
i.__table__.columns} for i in invnts],
        'oprts': [{c.name: getattr(o, c.name) for c in
o.__table__.columns} for o in oprts]
    }
    return backup_data

def restore_db(backup_data):
    for invnt_data in backup_data.get('invnts', []):
        invnt = Invnt(**invnt_data)
        db.session.add(invnt)
    for oprt_data in backup_data.get('oprts', []):
        oprt = Oprt(**oprt_data)
        db.session.add(oprt)
    db.session.commit()
    return True

def stats_by_loc():
    locs = {}
    for invnt in Invnt.query.all():
        if invnt.loc not in locs:
            locs[invnt.loc] = 0
        locs[invnt.loc] += 1
    return locs

def stats_by_tp():
    tps = {}
    for invnt in Invnt.query.all():
        if invnt.tp not in tps:
            tps[invnt.tp] = 0
        tps[invnt.tp] += 1
    return tps

def reset_all_invnts(loc="на складі"):
    invnts = Invnt.query.all()
    for invnt in invnts:
        trnsf_invnt(invnt.id, loc, "syst", "srt")
    return True

def filter_invnt_complex(nm=None, tp=None, loc=None, spr=None,
cst_min=None, cst_max=None):
    q = Invnt.query
    if nm:
        q = q.filter(Invnt.nm.contains(nm))
    if tp:

```

```
        q = q.filter(Invnt.tp.contains(tp))
    if loc:
        q = q.filter(Invnt.loc.contains(loc))
    if spr:
        q = q.filter(Invnt.spr.contains(spr))
    if cst_min is not None:
        q = q.filter(Invnt.cst >= cst_min)
    if cst_max is not None:
        q = q.filter(Invnt.cst <= cst_max)
    return q.all()
```

ДОДАТОК В
Слайди презентації



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

Програмне забезпечення для обліку та контролю руху обладнання

Виконав: Данило ЄРМАКОВ

ст. групи КНТ-122

Керівник: Тетяна КОЛПАКОВА

к.т.н., доцент

Рисунок В.1 – Слайд 1



Об'єкт, предмет та мета роботи

- Об'єкт дослідження – процес розробки програмного забезпечення для обліку та контролю руху обладнання.
- Предмет дослідження – програмні засоби для обліку та контролю руху обладнання.
- Метою роботи є розробка програмного забезпечення для обліку та контролю руху обладнання.

2

Рисунок В.2 – Слайд 2

Завдання роботи

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

- виконати аналіз предметної області та програмних засобів для обліку та контролю руху обладнання;
- здійснити проектування програмного забезпечення для обліку та контролю руху обладнання;
- створити програмне забезпечення для обліку та контролю руху обладнання;
- виконати тестування розробленого програмного забезпечення для обліку та контролю руху обладнання.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Fishbowl	Striven	3PL Warehouse Manager
Автоматизація складських процесів	+—	+—	+
Легкість освоєння функцій	+	+—	—
Інтеграція з бухгалтерськими системами	+	+	+—
Сканування штрих-кодів	+	+—	+
Хмарна доступність	+—	+	+
Інтеграція з транспортними службами	+	+—	+

4

Рисунок В.4 – Слайд 4

Постановка завдання

При розробці програмного забезпечення для обліку та контролю руху обладнання необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення та редагування інформації про обладнання на підприємстві;
- підтримка можливості авторизації користувачів;
- можливість формування журналу операцій стосовно переміщень обладнання;
- можливість формування звітів;
- підтримка можливості експортування звітів у формат **pdf**;
- підтримка можливості відображення та редагування інформації про постачальників обладнання.

5

Рисунок В.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	PHP
Простота та швидкість розробки	+	+—	+—
Читабельність коду та легкість супроводу	+	+	—
Гнучкість масштабування та розширення	+	+—	+
Зручність для створення ПЗ для обліку та контролю руху обладнання	+	—	—
Платформна незалежність	+	+—	+

Рисунок В.6 – Слайд 6

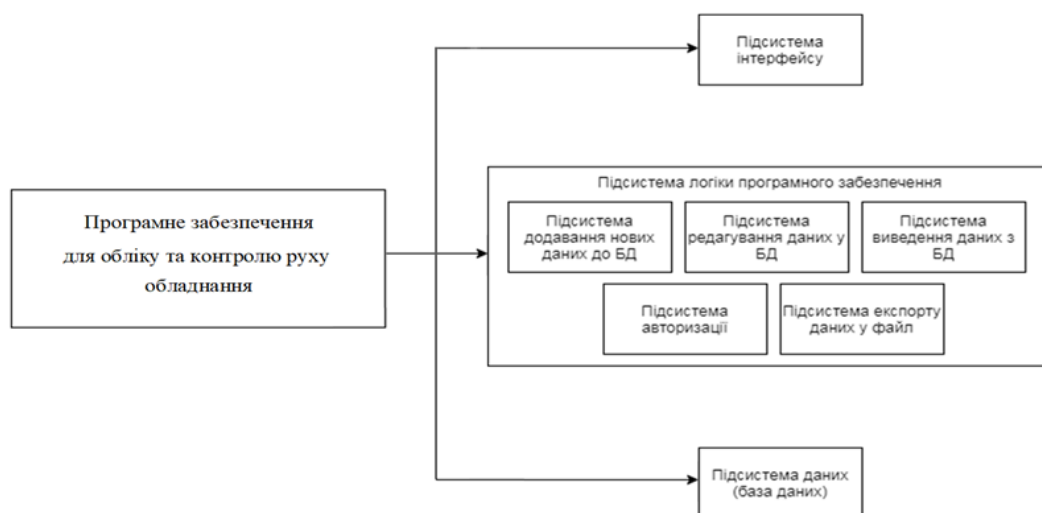
Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Spyder	Sublime Text
Організація структури проєктів	+	—	—
Автоматизація тестування та налагодження	+	+—	+—
Інтеграція зі системами контролю версій	+	—	—
Інтелектуальні підказки та автодоповнення	+	+—	+—
Зручність для роботи з аналітикою	+	+	+—

7

Рисунок В.7 – Слайд 7

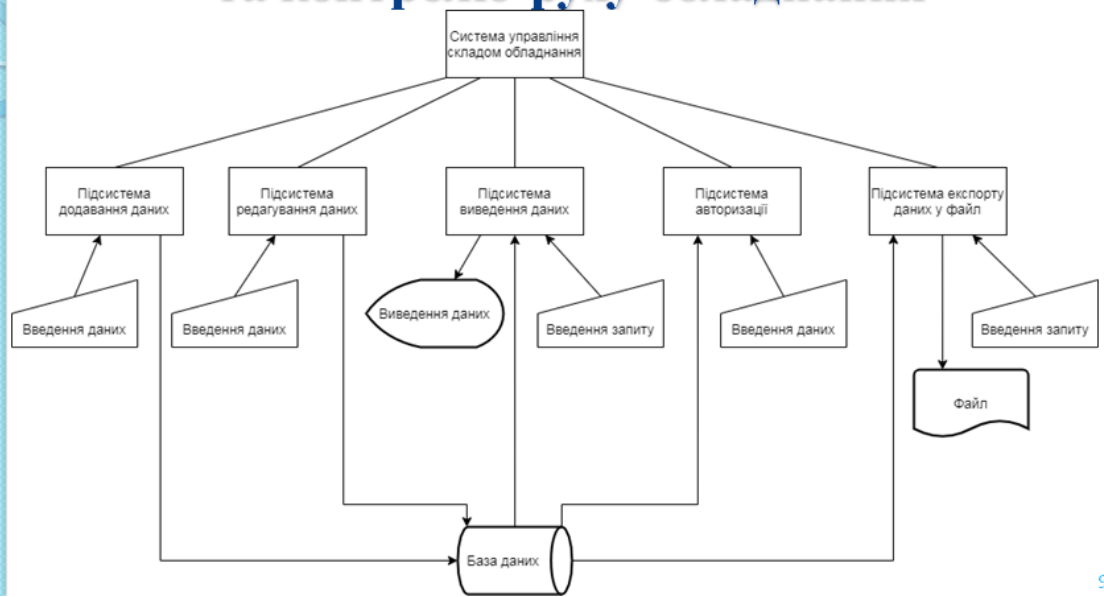
Структура програми та контролю руху обладнання



8

Рисунок В.8 – Слайд 8

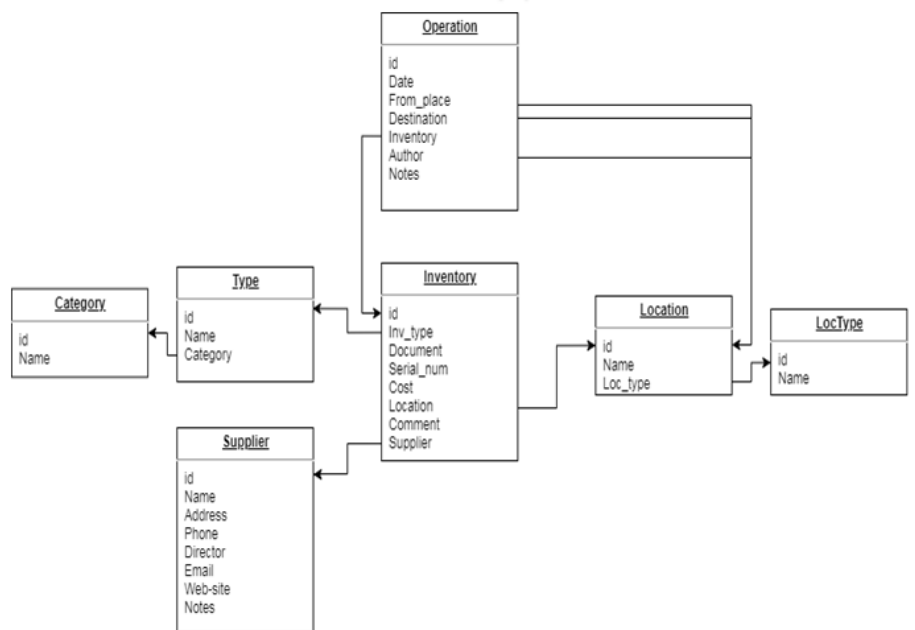
Схема функціонування програми для обліку та контролю руху обладнання



9

Рисунок В.9 – Слайд 9

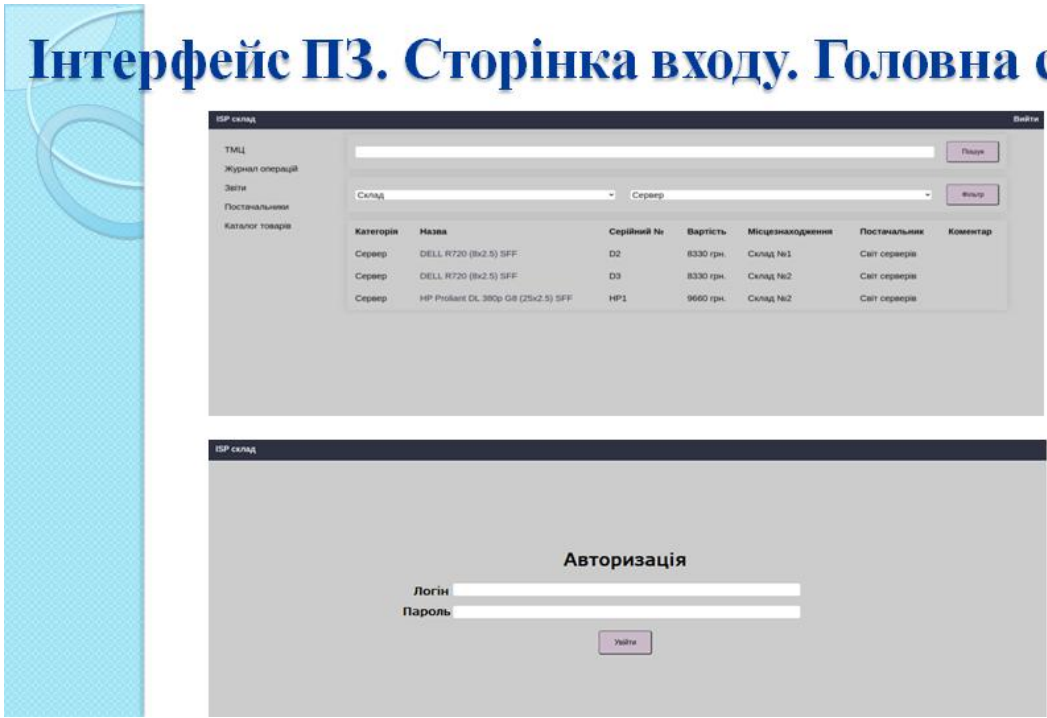
Схема бази даних



10

Рисунок В.10 – Слайд 10

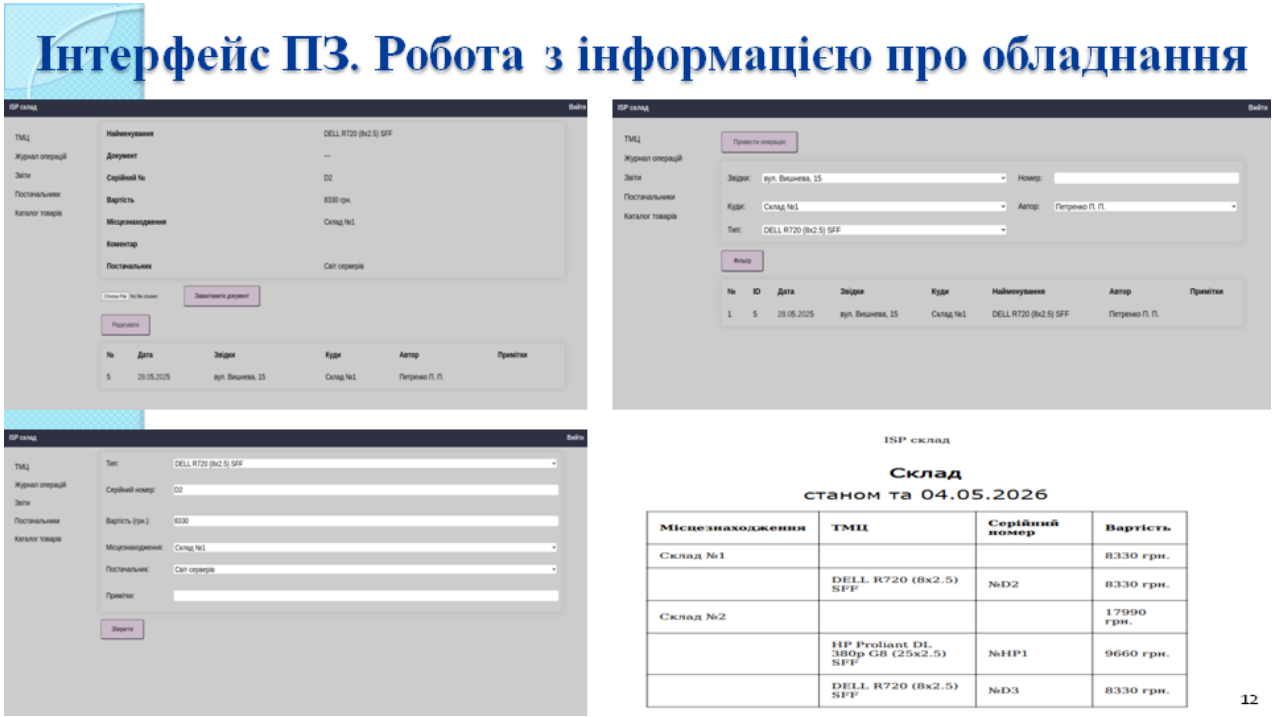
Інтерфейс ПЗ. Сторінка входу. Головна сторінка



11

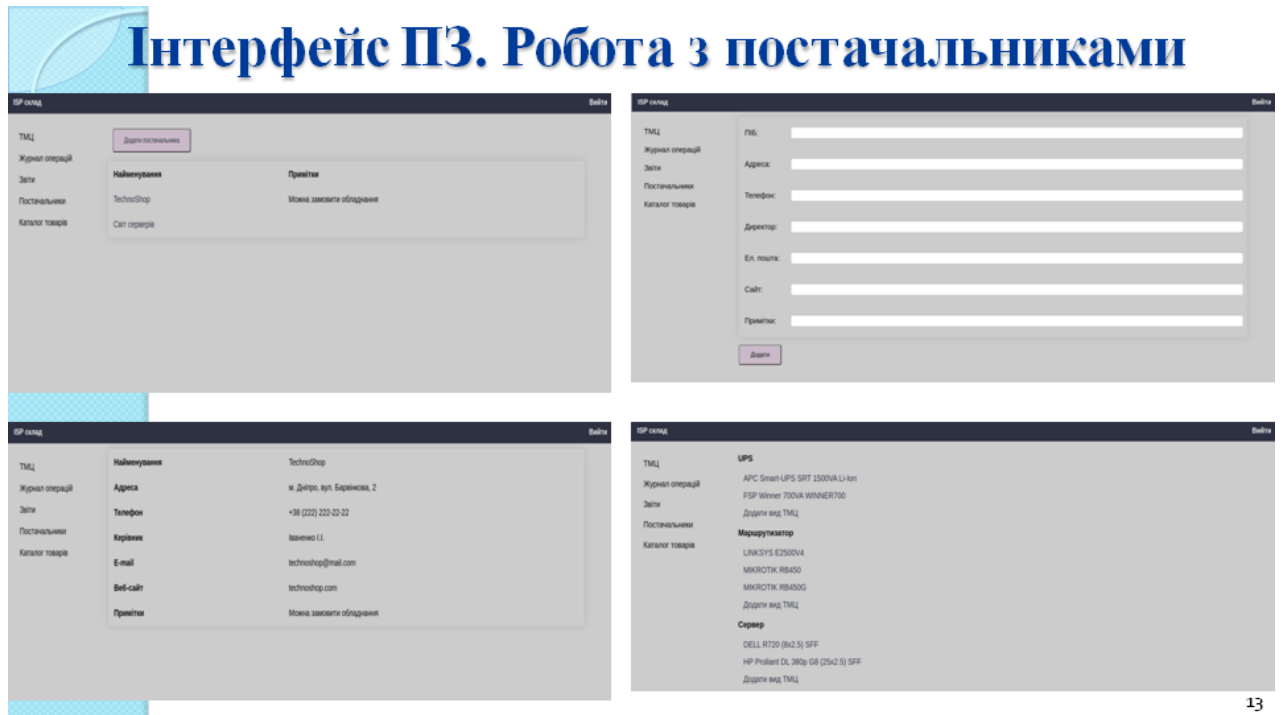
Рисунок В.11 – Слайд 11

Інтерфейс ПЗ. Робота з інформацією про обладнання



12

Рисунок В.12 – Слайд 12



13

Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для обліку та контролю руху обладнання.

Сформульовано функціональні вимоги до програмного забезпечення для обліку та контролю руху обладнання. Для створення програмного забезпечення для обліку та контролю руху обладнання обрано мову програмування Python та середовище розробки PyCharm. Розроблено програмне забезпечення для обліку та контролю руху обладнання. Описано функціонування програми. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для обліку та контролю руху обладнання.

Виконано тестування розробленого програмного забезпечення для обліку та контролю руху обладнання. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу обліку обладнання.

14

Рисунок В.14 – Слайд 14