

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

## **Бакалаврська дипломна робота**

на тему: «Програмна система обліку поставок запасних частин до мережі  
ремонтних ділянок»

Виконав: студент 4 курсу  
групи ІПІ-18б  
спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Овод Д. В.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Хошаба О.М.

(прізвище та ініціали)

Рецензент: д.т.н., проф. КН Васілевський О.М.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедр \_\_\_\_\_

«  » \_\_\_\_\_ 2022 р.

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра програмного забезпечення  
Рівень вищої освіти перший бакалаврський  
Галузь знань 12 - Інформаційні технології  
Спеціальність 121 – Інженерія програмного забезпечення  
Освітньо-професійна програма - Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ  
Завідувач кафедри ПЗ  
Романюк О. Н.  
25 березня 2022 року

## **З А В Д А Н Н Я НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Овод Дмитро Васильович

1. Тема роботи - «Програмна система обліку поставок запасних частин до мережі ремонтних ділянок»

Керівник роботи: Хошаба Олександр Мирославович, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від 24 березня 2022 р. № 66

2. Строк подання студентом роботи 13 червня 2022 р.

3. Вихідні дані до роботи: середовище розробки – Visual Studio Code, Мови розробки – Java, операційна система – Windows 7/8/10, Linux Centos/Debian.

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз та постановка задачі; порівняльний аналіз аналогів; проектування модулів програмної системи; розробка алгоритму роботи відділу постачання запасних частин, модуля обслуговування заявок на ремонтних ділянках та інтерфейсу програмної системи обліку поставок запасних частин; тестування модуля заповнення замовлень на поставку запасних частин, frontend та backend частини програмної системи, висновки; перелік посилань, додатки.

5. Перелік графічного матеріалу: мета та задачі роботи, порівняльна характеристика аналогів, блок-схема алгоритмів роботи програмної системи, тестування програмної системи, висновки.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Хошаба О.М., к.т.н., доцент кафедри ПЗ		

7. Дата видачі завдання 25 березня 2022 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз проблеми, обґрунтування актуальності розробки системи та постановка задач	26.03.2022- 10.04.2022	Вик.
2	Проектування модулів системи обліку поставок запасних частин	11.04.2022- 26.04.2022	Вик.
3	Вибір середовища та розробка алгоритму роботи відділу постачання запасних частин	27.04.2022- 4.05.2022	Вик.
4	Розробка frontend та backend частин програмної системи	5.05.2022- 24.05.2022	Вик.
5	Тестування роботи та впровадження програмної системи	25.05.2022- 27.05.2022	Вик.
6	Оформлення матеріалів до захисту БДР	27.05.2022- 10.06.2022	Вик.

Студент

\_\_\_\_\_

(підпис)

Овод Д.В.

(прізвище та ініціали)

Керівник бакалаврської дипломної роботи

\_\_\_\_\_

(підпис)

Хошаба О.М.

(прізвище та ініціали)

## АНОТАЦІЯ

Бакалаврська дипломна робота складається з 75 сторінок формату А4, на яких є 13 рисунків, 1 таблиці, список використаних джерел містить 23 найменувань.

В роботі проведено аналіз моделей управління поставками запасних частин у мережу ремонтних ділянок та порівняльний аналіз аналогів. Детально виконано аналіз розвитку сучасних систем обліку поставок запасних частин до мережі ремонтних ділянок. На основі аналізу створена постановка задачі, де визначено основні етапи розробки, тестування та впровадження програмної системи. В ході роботи виконано проектування, де визначені модулі системи обліку поставок запасних частин для розділу приватних постачальників, власника бізнесу та технічної підтримки. На основі цього, розроблені модулі обслуговування заявок на ремонтних ділянках та інтерфейс програмної системи обліку поставок запасних частин та розроблено алгоритм роботи відділу постачання запасних частин на ремонтній ділянці. На основі створеного програмного коду виконано тестування модуля заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок. При цьому, виконано тестування frontend та backend частин програмної системи обліку поставок запасних частин до мережі ремонтних ділянок.

## ANNOTATION

The bachelor's thesis consists of 75 A4 pages, which contain 13 figures, 1 tables, a list of sources used contains 23 items.

The analysis of spare parts supply management models in the network of repair sites and comparative analysis of analogues is carried out in the work. The analysis of development of modern systems of the account of deliveries of spare parts to a network of repair sites is in detail executed. On the basis of the analysis the statement of the problem is created, where the main stages of development, testing and implementation of the software system are determined. In the course of the work the design was performed, where the modules of the spare parts supply accounting system for the section of private suppliers, business owner and technical support were determined. Based on this, the service modules for repair sites and the interface of the software system for accounting for the supply of spare parts were developed and the algorithm of the spare parts supply department for the repair sites was developed. On the basis of the created program code the testing of the module of filling of orders for delivery of spare parts to a network of repair sites is executed. At the same time, testing of frontend and backend parts of the software system of accounting for the supply of spare parts to the network of repair sites was performed.

## ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ	13
1.1 Моделі управління поставками запасних частин	13
1.2 Порівняльний аналіз аналогів	19
1.3 Аналіз розвитку сучасних систем обліку поставок запасних частин	25
1.4 Постановка задачі	32
1.4 Висновки	33
2 ПРОЕКТУВАННЯ МОДУЛІВ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ПОСТАВОК ЗАПАСНИХ ЧАСТИН	35
2.1 Проектування модулів системи для розділу приватних постачальників	35
2.2 Проектування модулів системи для розділу власника бізнесу	38
2.3 Проектування модулів системи для розділу технічної підтримки	42
2.4 Висновки	45
3 РОЗРОБКА МОДУЛІВ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ПОСТАВОК ЗАПАСНИХ ЧАСТИН	46
3.1 Розробка модуля обслуговування заявок на ремонтних ділянках	46
3.2 Розробка алгоритму роботи відділу постачання запасних частин	49
3.3 Розробка інтерфейсу програмної системи обліку поставок запасних частин	51
3.4 Висновки	56
4 ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ МОДУЛІВ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ПОСТАВОК ЗАПАСНИХ ЧАСТИН	58
4.1 Тестування модуля заповнення замовлень на поставку запасних частин	58
4.2 Тестування frontend частини програмної системи	62
4.3 Тестування backend частини програмної системи	67
4.4 Висновки	70
ВИСНОВКИ	72
ПЕРЕЛІК ПОСИЛАНЬ	76

ДОДАТОК А	79
ДОДАТОК Б	83
ДОДАТОК В	84
ДОДАТОК Г	100
ДОДАТОК Д	105

## ВСТУП

**Обґрунтування вибору теми дослідження.** На сьогоднішній день організація обліку автозапчастин до мережі ремонтних ділянок є дуже важливим та актуальним завданням. Важливість завдання полягає в тому, що та сама запасна частина підходить до кількох моделей, марок авто, або навпаки, жорстко прив'язана до моделі двигуна або кузова. Тому, складність вирішення цього завдання без використання сучасних інформаційних технологій приречено на невдачу у розвитку середнього та великого бізнесу у сфері організації обліку автозапчастин.

Складність даного завдання полягає в тому, що для організації обліку автозапчастин необхідно враховувати такі важливі параметри як одиничні продажі, велику кількість постачальників, що пропонують ідентичні автозапчастини від різних виробників з різним маркуванням, наявність різних довідників номенклатури, що налічує сотні позицій, необхідність пошуку аналогів, наявність певних запчастин від різних виробників на складі або робота «на замовлення», облік клієнтів та багато іншого.

У той же час, використання сучасних інформаційних технологій та програмних засобів дозволяє вирішити безліч проблем. Наприклад найактуальнішим завданням у час є проведення одночасного обліку на кількох складах, де окрема роздрібна точка має окремий віртуальний склад, але фізично автозапчастини перебувають у одній території зберігання. При цьому віртуальних складів необхідно мати практично необмежену кількість.

Також, незалежно від кількості складів, програмний засіб повинен проводити єдиний облік приходу, наявності, витрати автозапчастин, а також створювати віртуальні ремкомплекти, включаючи необхідні деталі. Ця функція надає можливість продавцеві задати пошук, наприклад, щодо «ремонт стабілізаторів поперечної стійкості» та отримати повний перелік необхідного.

Для вирішення завдань управління програмна система обліку постачання запасних частин у мережу ремонтних ділянок повинна мати розвинені функції аналітики. При цьому важливими питаннями є облік поставки товарів,



формування звітів з продажу, аналіз руху товарів за групами, виробниками, аналітична робота з продавцями та постачальниками, відстеження повернень товарів, надходження кредитних або безготівкових коштів. Також важливими є вирішення завдань повністю або частково оплачених замовлень.

Функція управління запасними частинами має вирішальне значення з погляду операційної діяльності, особливо у галузях, як автомобільна промисловість. На процес обліку запасних частин впливають ціла низка важливих факторів. До таких факторів можна віднести попит, що змінюється, і пропозиція на поставку запасних частин, непередбачуваність розвитку ринку, зміни в заміщенні деталей і жорсткий контроль за запасами запасних частин у поєднанні з високими очікуваннями рівня обслуговування від клієнтів. Тому, невідкладність у проведенні бізнес-процесів управління поставками запасних частин у виробничих операціях не може бути недооцінена.

В той же час, істотну роль у управлінні запасами запасних частин грає аналіз ABC. Аналіз ABC надає метод для ідентифікації елементів, які значно впливають на загальну вартість інвентарю. При цьому забезпечується механізм визначення різних категорій запасів, які вимагають різного управління і контролю. Аналіз ABC передбачає, що інвентаризація всередині організації постачання деталей мають різні значення з їх обслуговування. Таким чином, інвентар який поставляється від різних виробників згрупований за трьома категоріями (A, B і C) у порядку їх оцінного значення.

Насамперед інвентар запасних частин, як правило, аналізується за допомогою ефектів режиму відмови та аналізу критичності, який спрямований на визначення та зменшення проблеми, пов'язаних з поставками. Це означає, що аналізуються поставки лише найважливіших запасних частин, які приносять найбільший прибуток.

Після цього виконується аналіз ABC відповідно до принципу Парето для найважливіших запасних частин із метою класифікації різного інвентарю. Далі, переходять до наступного скорочення виробів, що поставляються, тобто розглядаються тільки запасні частини, які відносяться до найбільш критичного класу.

Таким чином, аналіз ABC, який прийнятий на озброєння багатьма світовими підприємствами, класифікує різні компоненти виробів поставок на три класи критичності: дуже важливий (клас А), важливий (В-клас) і менш важливий (клас С). У цьому, першому етапі лише компоненти виробів, що належать класу А, вимагають уваги керівництва. Далі послідовно виконуються інші етапи у відповідності з наявними ресурсами організації.

При цьому, функції управління матеріальними потоками та закупівлями запасних частин часто пов'язані з обслуговуванням, обладнанням, технікою та операціями щодо великої кількості людей.

Тому актуальними питаннями є розробка програмної системи обліку поставок запасних частин до мережі ремонтних ділянок.

**Зв'язок роботи з науковими програмами, планами, темами.** Робота виконувалась згідно плану виконання наукових досліджень на кафедрі програмного забезпечення

**Мета та завдання.** Метою роботи є підвищення ефективності поставок запасних частин до мережі ремонтних ділянок. За рахунок впровадження програмної системи на основі моделі управління вдається покращити виробничий процес, підвищити прибуток працівникам центрального офісу та ремонтних ділянок.

Основними завданнями розробляємої програмної системи є вирішення основних задач, що пов'язані з процесом отримання та виконання замовлень з ремонту автомобільного транспорту до центрального офісу.

**Об'єкт дослідження** – процес поставок запасних частин до мережі ремонтних ділянок.

**Предмет дослідження** – методи та програмні засоби, що дозволяють покращити виробничий процес.

**Методи дослідження.** У процесі досліджень використовувались методи дослідження:

- методи розробки графічного інтерфейсу;
- методи менеджменту великої кількості поставок до мережі ремонтних ділянок;

- методи розробки клієнт серверних систем;
- методи авторизації користувача;

**Наукова новизна отриманих результатів.** Створена програмна система на основі моделі управління поставками запасних частин у мережу ремонтних ділянок враховуються кращі застосування в обслуговуванні та ремонті автомобільного транспорту, що виконує оптимальні постачання запасних частин. Впровадження отриманих результатів значно покращує зручність роботи з великими мережами ремонтних ділянок та ефективно використовує наявні матеріальні та людські ресурси.

**Практична цінність отриманих результатів.** Практичне значення полягає у розробці, тестуванні та впровадженні програмної системи, що на основі моделі управління поставками запасних частин у мережу ремонтних ділянок враховуються кращі практики в обслуговуванні та ремонті автомобільного транспорту, схеми класифікації за методами ABC.

Відповідно до поставленої мети виконані наступні задачі:

- провести аналіз моделей управління поставками запасних частин у мережу ремонтних ділянок та порівняльний аналіз аналогів;
- виконати аналіз розвитку сучасних систем обліку поставок запасних частин до мережі ремонтних ділянок;
- створити постановку задачі;
- виконати проектування модулів системи обліку поставок запасних частин для розділу приватних постачальників, власника бізнесу та технічної підтримки;
- розробити модулі обслуговування заявок на ремонтних ділянках та інтерфейс програмної системи обліку поставок запасних частин;
- розробити алгоритм роботи відділу постачання запасних частин на ремонтній ділянці;
- виконати тестування модуля заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок;
- виконати тестування frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок;

- виконати тестування backend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок.

**Особистий внесок здобувача.** Усі наукові результати, викладені у бакалаврській дипломній роботі, отримані автором особисто. Автору належать такі результати: архітектура програмної системи; розроблені алгоритми; модулі клієнтської частини; модулі серверної частини.

**Публікації.** Публікацій по даній роботі немає.

## 1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Моделі управління поставками запасних частин

Моделі управління постачанням запасних частин у мережу ремонтних ділянок мають велике значення у розвитку економіки України з точки зору операційної діяльності, особливо в таких галузях знаходження як автомобільні підприємства, нафтопереробні заводи, хімічні заводи, паперові фабрики, верстатобудування та підприємства харчування.

Для побудови моделей управління поставками запасних частин у мережу ремонтних ділянок необхідно враховувати цілу низку таких факторів як попит, непередбачуваність у прогнозуванні, заміщення деталей та жорсткий контроль за запасами запасних частин, які знаходяться на складах у поєднанні з високими очікуваннями рівня обслуговування. При цьому фактор часу в поєднанні невідкладності управління запасними частинами у виробничих операціях не може бути недооцінена.

У більшості організацій постійний контроль здійснюється за зберіганням, замовленням та обслуговуванням за допомогою постачання запасних частин, які необхідні для обслуговування автотранспорту. Такі функції у моделях управління матеріальними потоками та закупівель запасних частин часто пов'язані з обслуговуванням, обладнанням, технікою та різними операціями; з урахуванням різної кількості та типів одиниць зберігання, таким видом продукції як капітальні та критичні запасні частини, застарілі запаси, інвестиції в інвентар, обіг та функціонування деталей.

При цьому, у моделях управління поставками запасних частин у мережу ремонтних ділянок необхідно враховувати кращі практики в обслуговуванні та ремонті автомобільного транспорту, схеми класифікації за методами ABC, де описуються ролі персоналу технічного обслуговування як постачальник послуг. Розглянемо такі моделі докладніше (рис. 1.1).



Рисунок 1.1 - Моделі управління поставками запасних частин у мережу ремонтних ділянок

У більшості організацій постійний контроль за поставками запасних частин забезпечується за допомогою зберігання, замовлень та обслуговування продукції, необхідних для обслуговування та операцій ремонту автомобільної техніки. Тому, у моделях функції управління матеріальними потоками та закупівель часто пов'язані з обслуговуванням та операціями складського поповнення продукції; обліку різної кількості та типів одиниць продукції, класифікації капітальних та критичних видів запасних частин, облік застарілих запасів, інвестиції у постачання запасних частин, обіг та експлуатація програмного забезпечення.

При цьому повсюдне впровадження досягнень нових інформаційних технологій потребує вдосконалення систем підтримки прийняття рішень у галузі управління промисловими та автомобільними підприємствами, що спираються на сучасну апаратно-програмну платформу.

Принцип самофінансування більшості підприємств, що діє в умовах ринкової економіки України, вимагає забезпечити від них зацікавленість у реалістичній оцінці витрат на транспортування комплектуючих у виробничій

мережі ремонтних підприємств. У зв'язку з цим структура транспортної ланки мережі ремонтних підприємств має бути достатньо гнучкою, щоб гарантувати частку та цілодобову доставку вантажів з метою підтримки ритмічності виробничого циклу підприємств.

Для планомірного та збалансованого розвитку будь-якого підприємства, тим більше для великих ремонтних підприємств, що забезпечують перевезення у масштабах регіону, необхідно впровадження нових технологічних та технічних рішень, інформаційних та телекомунікаційних технологій, нових моделей управлінських рішень та методів оперативного збору та аналізу результатів обстежень у потоці замовлень на комплектуючі автомобілів.

Тому, метою розробки моделі планування та управління поставками запасних частин у мережу ремонтних ділянок є підвищення ефективності планування виробничої діяльності транспортної ланки мережі ремонтних підприємств за рахунок використання математичних методів та моделювання процесів управління поставками комплектуючих.

Для розробки такої моделі планування та управління постачанням запасних частин необхідно послідовно вирішувати такі завдання:

- системного аналізу методів та моделей поставок комплектуючих для мережі ремонтних підприємств;
- формальне представлення бізнес-процесів, що реалізують функції управління та планування рухом запчастин та комплектуючих;
- розробку моделей постачання запчастин та комплектуючих в умовах ймовірнісної невизначеності у розвитку економічних факторів;
- розробку методик планування експерименту та аналітичної обробки результатів моделювання експериментів;
- розробку алгоритмів вибору рухомого складу у системі управління постачанням комплектуючих деталей;
- розробку структурно-функціональної моделі системи управління постачанням для мережі ремонтних підприємств.

Тому розробка моделей управління поставками запасних частин у мережу ремонтних ділянок може бути заснована на системному аналізі, теорії управління

запасами, теорії випадкових процесів, методів імітаційного моделювання, статистичних методів та методів розвідувального аналізу даних, мереж масового обслуговування та інших методів багатовимірної аналізу.

Розглянемо наступну модель управління постачанням запасними частинами з урахуванням використання логістичних принципів.

Очевидно, що якісна транспортна послуга може бути надана з використанням надійного рухомого складу, підтримка якого у працездатному стані потребує вдосконалення роботи не лише служб, зайнятих технічним обслуговуванням та ремонтом, а й системи постачання.

Парадоксальність сьогоденного стану більшості АТП полягає в тому, що, незважаючи на ліквідацію дефіциту запасних частин у товаропровідній мережі, простої автомобілів через їхню відсутність не зменшилися порівняно з тим періодом, коли в умовах планової економіки діяв лімітно-розподільний механізм постачання.

Основна причина такої ситуації полягає у високих цінах на запасні частини та агрегати, зростання яких у кілька разів перевищило збільшення тарифів на перевезення.

При цьому ціни на запчастини інтегрально відображають ті зміни, що відбулися в розвитку економіки автомобільного сегмента: збільшення кількості посередників від заводу-виробника до споживача; зростання кількості АТП із невеликою кількістю автомобілів (від 1 до 10 одиниць); збільшення різного типу автомобільного парку, зокрема за рахунок автомобілів іноземного виробництва; старіння рухомого складу та інші фактори.

Все вищезгадане вимагає розвитку нових підходів до управління постачанням запасними частинами, зокрема, використання логістичних принципів, згідно з якими, як і раніше, актуальними для товаропровідної мережі залишаються два питання: визначення номенклатури та кількісна оцінка запасів для кожної позиції номенклатури.

Для розробки моделі управління постачанням запасних частин на основі використання логістичних принципів необхідно вирішити наступні завдання:



- оцінити вплив системи постачання запасними частинами на показники роботи АТП;
- дати кількісну оцінку чинників, визначальних витрата запасних частей;
- проаналізувати способи визначення номенклатурних груп та методи розрахунку запасних частин;
- розробити аналітичну методику поділу номенклатури запасних частин на групи;
- удосконалити методику розрахунку запасних частин для мінімальних вибірок;
- розробити загальну блок-схему розрахунку запасних частин.

Для більш ефективної розробки моделі управління постачанням запасними частинами на основі використання логістичних принципів застосовуються такі підходи щодо:

- визначення кількісної оцінки впливу різних факторів на витрату запасних частин в АТП та виділення основних факторів;
- виконання класифікації підходів та визначення номенклатурних груп за такими відмітними ознаками: об'єкт логістичного ланцюга, інформаційна база, спосіб поділу;
- розроблення аналітичної методики поділу номенклатури на групи А, В та С, що базуються на єдиному вартісному показнику витрат мережі ремонтних ділянок;
- розроблення методики розрахунку запасних частин для малих АТП;
- формуванню загальної блок-схеми розрахунку продажу запасних частин, об'єднаних в один програмний комплекс.

Така модель управління постачанням запасними частинами на основі використання логістичних принципів сприятиме підвищенню ефективності роботи АТП, оскільки вона спрямована на зниження витрат на запасні частини та покращення організації системи постачання.

Розглянемо ще одну модель автоматизації забезпечення складів запасних частин, що є найважливішою частиною мережі технічного обслуговування підприємства-виробника автомобілів.

Відомо, що високоефективне постачання можливе нині лише за умови автоматизації статистики, аналізу, прогнозу, обробки документації, що дозволяє як оптимізувати запаси, знизити витрати на зберігання запасних частин, а й значно прискорити обслуговування споживачів.

За відсутності ефективної інформаційної системи, що забезпечує збирання та обробку статистичної інформації, організувати конкурентоспроможну мережу забезпечення поточного парку автомобілів запасними частинами практично неможливо.

Тому, своєчасний науково обґрунтований і точний прогноз потреби у запасних частинах дозволяє приймати правильні управлінські рішення про номенклатуру та кількість деталей, що поставляються на склад, поліпшити економічні показники діяльності підприємства.

Важливість моделі автоматизації забезпечення складів запасних частин визначає можливість оптимізації номенклатури, що зберігається в базах даних відповідно до поточних та прогнозованих потреб ринку автомобільних запасних частин за рахунок автоматизації процесів обробки інформації, підвищення якості та оперативності прогнозів на основі обґрунтованих формалізованих моделей.

Метою використання моделі автоматизації забезпечення складів запасних частин є автоматизація постачання запасних частин мережі автомобільних підприємств з виробництва та обслуговування автотранспортної техніки за рахунок розробки відповідних моделей, методів, алгоритмів та програм.

Для цього необхідно послідовно вирішуються такі завдання:

- аналіз сучасних методів організації технологічних процесів та прогнозування потреби на базі статистичної інформації про збут автомобільних запасних частин;

- розробка моделей та методів автоматизації найбільш трудомістких етапів збору та обробки інформації з метою скорочення термінів та вартості підготовки прогнозів;

- використання запропонованих підходів для вирішення завдання розробки структури баз даних автоматизованих систем управління та постачання запасними частинами регіонального складу;

- використання науково обґрунтованих методів та алгоритмів при проектуванні автоматизованих систем управління та постачання запасних частин;
- налагодження на моделі та практичне впровадження розробленої автоматизованих систем управління та постачання автомобільними запасними частинами.

Таким чином, у даному підрозділі описані моделі управління постачанням запасних частин у мережу ремонтних ділянок.

## 1.2 Порівняльний аналіз аналогів

Порівняльна характеристика аналогів проводилась серед десктопних (Автоділер) та хмарних платформ (LiveSkлад) програмно-технічних середовищ, які існують в Україні.

Зведені дані, які використовуються з метою порівняння цих платформ наведені в таблиці (табл. 1.1).

Розглянемо ці аналогі більш ретельно. Під час проведення порівняльної характеристики аналогів поширення запасних частин до мережі ремонтних ділянок звертає на себе увагу велика кількість десктопних додатків. Це зумовлено тим, що такі програмні засоби почали раніше розповсюджуватись у порівнянні з хмарними.

Також, у десктопних додатків більш розвинуті механізми комплексної автоматизації оперативного обліку запасних частин на підприємствах, що займаються оптово-роздрібною торгівлею автомобільними запчастинами та шинами. Завдяки цьому, під час експлуатації таких програмних засобів існують численні центри впровадження, які впливають на гнучку подальшу розробку автоматизованих систем. Так, в таких системах найбільш розвинута облікова торгова діяльність у конфігураціях, що реалізована за допомогою функціональних можливостей компоненти «Оперативний облік» і передбачає реєстрацію всіх операцій із замовлення, купівлі, зберігання, продажу запасних частин і ведення що з цими операціями взаєморозрахунків з покупцями і постачальниками.

Порівняльна характеристика аналогів поширення запасних частин до мережі ремонтних ділянок

Функціонал	Десктопні додатки (на прикладі Автоділер)	Хмарні платформи (на прикладі LiveSkлад)
Оперативний облік	Розвинутий	Не досить розвинутий
Експортно-імпортні операції	Розвинуті	Не досить розвинуті
Довідник "Постачальники"	Добре зібрана	Відмінно зібрана
Довідник "Номенклатура"	Добре зібрана	Відмінно зібрана
Нарахування заробітних плат	Досить розвинуті операції	Не досить розвинуті операції
Касові операції	Досить розвинуті операції	Не досить розвинуті операції
Аналіз бухгалтерської та економічної діяльності	Розвинутий	Не досить розвинутий
Технічна підтримка	Не досить ефективна	Ефективна
Засоби безпеки	Не досить ефективні	Ефективні

Як правило, в таких конфігураціях проводиться досить ефективний облік, який підтримує автоматичне списання собівартості товарів за одним із заданих методів сортування: "за середньою вартістю", FIFO або LIFO.

Також, у програмі Автоділер досить розвинутими є механізми інтеграції. Сенс використання таких механізмів полягає в тому, що користувачі цієї програми можуть вивантажити дані по запчастинах зі свого складу до будь-яких систем в різних текстових форматах із зазначенням ціни та кількості що є у наявності. Це досить зручно для потенційних покупців які можуть знайти різні відомості про товар та купити їх у ремонтну мережу. Причому витратити час на налаштування конфігурації вивантаження інформації з бази даних потрібно буде лише раз, а система буде продовжувати робити це автоматично у разі необхідності.

Описаний механізм інтеграції є корисним та важливим завдяки тому, що такі результати експорту даних можливо використовувати для багатьох популярних інтернет-магазинів з поширення автомобільних запасних частин таких як: Exist, Autodoc, Autopiter, Berg, Автостелс, Армтек, Шате-М, JAparts, IXORA, Rossko, Форум-авто та інші.

Також, в цьому випадку може бути передбачена можливість підбору запчастин через різні сайти. Для цього тільки потрібна реєстрація (логін та пароль) на сайті потрібного інтернет-магазину. Пошук запчастин буде проводитися лише за номерами (оригінальний номер та номер виробника).

Результатом такого пошуку буде перелік номенклатурних позицій із зазначенням кількості товару, що доступна на складі, та переліку цін на цю номенклатуру. Знайдені артикули можуть підсвічуються різними кольорами, залежно від їх наявності у довіднику «Номенклатура» і/або складі у програмі. На підставі зіставлення номенклатури є можливість створення різноманітних документів. Наприклад, замовлення клієнта, замовлення постачальнику, тощо.

Досить важливим функціоналом в програмній платформі Автоділер є окремий довідник, який формується виходячи з загальних даних бази "Співробітники". Цими даними особливо користуються менеджери центрів з поширення запасних частин до мережі ремонтних ділянок. При цьому, через менеджерів проходить необхідна інформація про потреби клієнта. Тому, наскільки правильно та повно будуть заповнені всі документи – залежить від менеджера. При виникненні певних питань, що пов'язані з конкретним документом, інші працівники центрів відразу звертаються до менеджера, який оформляє та відповідає за надані відомості про постачальників та клієнтів. Таким чином, цей функціонал допомагає більш оперативно вирішити питання, що виникає на етапі постачання автомобільних запасних частин.

Також, важливими є інформація про постачальників внаслідок таких положень. Під час постановки ці на облік якоїсь запасної частини необхідно враховувати, хто був постачальником. Тому, у довіднику "Постачальників" необхідно мати таку можливість, щоб отримати необхідну інформацію.

Для повторного придбання запасних частин, також можна в довіднику "Номенклатура" подивитися, хто був постачальником її минулого разу, а не згадувати чи шукати через інтернет, де можна її придбати.

Тому, система повинна мати можливість генерувати відомості по постачальникам, для отримання кількісного і вартісного аналізу по кожному постачальнику за певний період часу з детальною розкладкою по ньому. Це також необхідно для повернення товару внаслідок браку. Для цього, існує можливість відстежити всю історію взаємин із постачальниками.

Важливою є інформація про суму і дату купування товару, можливість передоплати. Таким чином, досить зручно вести всю історію взаємовідносин в одній базі, а не в кількох зошитах.

Досить прогресивним за напрямком розвитку програмних засобів з поширення запасних частин до мережі ремонтних ділянок є використання хмарних технологій. Розглянемо цей напрямок на прикладі CRM LiveSkлад.

Отже, CRM LiveSkлад – це хмарна платформа, яка в першу чергу розрахована на збільшення показників поширення запасних частин до мережі ремонтних ділянок та дозволяє ефективно керувати різними механізмами з обліку, такими як пересортування товарів на складі, впорядкування бухгалтерії, підвищення рівня та швидкості обслуговування клієнтів.

Програма для обліку запчастин LiveSkлад є актуальною для малого бізнесу, до яких належать в першу чергу сервісні центри. Функціонал та можливості програми обліку запчастин від LiveSkлад включає наступні переваги.

Функціонал обробки замовлень, що підвищує швидкість обробки заявок та включає в себе такі основні дії як визначення втрати замовлення, стеження за дотриманням термінів виконання поставки запасних частин, реалізація можливості індивідуального підходу до кожного клієнта завдяки збереженій історії взаємин від першого дзвінка.

Функціонал нарахування зарплат призначений для здійснення розрахунків та нарахувань заробітних плат для кожного працівника окремо. Він зберігає реєстр виплат та відображає їх у зарплатній відомості.

Функціонал онлайн-каса, що відповідає вимогам законів України щодо роботи з контрольно-касовою технікою інтегрується з більшістю моделей як онлайн-кас, так й інших кас які пропонує ринок поширення запасних частин.

Функціонал відстеження реклами допомагає якісно оцінити співвідношення витрати/ефективність, тобто зрозуміти, скільки грошей йде на рекламу і наскільки вона продуктивна.

Функціонал взаємовідносини з постачальниками визначає в режимі реального часу (онлайн) інформацію про наявність товару, вартість, терміни відвантаження продукції більш ніж у 200 найбільших постачальників різних країн СНД. В цьому ж функціоналі існує можливість оперативного додавання до реєстру нового партнера. Для цього потрібно оформити заявку за допомогою якої він буде внесений до оновленого списку.

Функціонал з реалізації продажів надає можливість врахування прибутку під час продаж. Також, швидкість обробки заявок значно зростає завдяки використанню сканеру штрих-кодів.

Функціонал з підтримки бухгалтерії та економічного відділу надає можливість обліку реалізації товарів на сучасному рівні, де за допомогою вбудованих функцій 1С бухгалтерії вдається ефективно систематизувати взаєморозрахунки з постачальниками, здійснює контроль товарно-матеріальних цінностей, реалізує касові транзакції, щодо доходів/витрат, відстежувати заборгованості, контролювати фінансові потоки.

Функціонал з використання статистичних даних призначений для того, щоб працівники центру могли вчасно отримувати статистичні дані з усіх напрямків бізнесу, що забезпечує досягнення запланованого фінансового результату.

Функціонал з обліку по складу оптимізує та спрощує складські операції, починаючи від прийому продукції та закінчуючи списанням витратних матеріалів.

Функціонал з управління повідомленнями необхідний для завчасного сповіщення клієнтів через SMS канали зв'язку. Також, досить ефективним в обміні повідомленнями є донесення інформації до працівників через Telegram.

Функціонал з роздруківки документації надає можливість більш ефективно управляти поширенням запасних частин завдяки наявності шаблонів. Це дозволяє

роздрукувати будь-який документ, у тому числі суворої звітності: товарні чеки, квитанції про приймання продукції та інше.

Функціонал з менеджером задач дозволяє співробітникам центру своєчасно ставити конкретні цілі співробітникам та контролювати їх виконання згідно з встановленими заздалегідь термінами.

Функціонал з номенклатурним обліком запасних частин у вигляді довідника. При цьому, довідник призначений для зберігання переліку товарів та продукції, що використовуються у центрі поширення запасних частин для ведення торговельно-закупівельної та виробничої діяльності. Ієрархічна структура допомагає розділити товари на зручні групи. Для кожного товару вводиться ідентифікаційна інформація, таку як код товару, оригінальний номер виробника, штрих-код та інше. Крім цього, можна встановити мінімальний та максимальний запас на складі. Для візуалізації запасної частини можна прикріпити її фотографії. Також для кожної номенклатурної одиниці вказується її аналоги та взаємозамінні деталі.

Досить важливим функціоналом, який зустрічається у десктопних та і хмарних платформ є формування прайс-листів. Тобто, існує можливість ведення різних прайс-листів. Наприклад, запчастини для сервісу коштуватимуть дешевше, ніж продаж у магазині (у системі може бути представлено шість видів цін для кожного прайсу).

Для цього, співробітникам не потрібно постійно змінювати ціну в документах. Все, що потрібно, це просто вибрати необхідний прайс-лист і заповнювати документи. Система сама перерахує вартість та надасть вже підсумкову суму з урахуванням усіх нюансів. Також можна роздрукувати прайс-лист і викласти, наприклад, на стенді у торговому залі. Додатково передбачено друк цінників на товар.

Функціонал з управління складськими приміщеннями. Цей функціонал є важливим так як навіть мінімальний запас товарів потребує складського обліку. А якщо у склад налічує сотні найменувань товарів, то тут вже необхідна досить потужна система. Тому, у програмі є можливість обліку товарів на складах. Склад може бути як торговим, і не торговим. У кожному з видів складу можливо вести



необмежену кількість розділів, які можуть відображати як її фізичну (ангар, теплий бокс), так і логічну структуру (наприклад, за видами товару). Таким чином, за допомогою цього функціоналу можливо легко зорієнтуватися, де саме знаходиться необхідний товар, а не оминати весь склад у його пошуках.

Також, існують певні переваги хмарної програми для обліку запчастин відносно наступних положень:

- зручний, простий та інтуїтивно зрозумілий інтерфейс;
- безпека, що доступ до клієнтської бази лише у власника компанії, де інформація шифрується, копіюється та зберігається у захищеному Data-центрі;
- покрокова інструкція з ефективного використання програмного забезпечення, відеоуроків, технічної підтримки в режимі 24/7 (доступ у хмару), експертна консультація в телефонному режимі, чаті, по електронній пошті;
- постійний контроль за бізнес-процесами, що здійснюються віддалено – зі смартфона, планшета;
- однаково ефективна підтримка користувачів в корпоративній мережі та одиночному сервісі;
- можливість застосувати програму безкоштовно протягом 7 днів – тест-драйв.

Проте головна перевага програми обліку запчастин – автоматизація та оптимізація всіх бізнес-процесів, виключення людського фактора. У сукупності це призводить до збільшення продажів і, як наслідок, зростання прибутку – мети, яку має кожен власник бізнесу, погоджуючись підключити той чи інший пакет CRM-системи LiveSklad.

### 1.3 Аналіз розвитку сучасних систем обліку поставок запасних частин

В даний час у системах обліку поставок запасних частин у мережу ремонтних ділянок, включаючи підсистему технічної експлуатації, відбуваються суттєві кількісні та якісні зміни в галузі інформаційного забезпечення виробничих процесів, які через кілька років призведуть до наступних змін.

1. Завершиться комп'ютеризація на рівні вирішення традиційних обліково-аналітичних, планових та управлінських завдань, автоматизації документообігу, ведення бухгалтерського обліку та вирішення інших завдань.

2. Нові інформаційні технології поширяться не лише на великі, а й на середні та малі транспортні, ремонтні та сервісні ремонтні ділянки. Такі ремонтні ділянки нині що неспроможні дозволити собі великих накладних витрат, які виживання визначається оперативністю реакцію змінюються умови роботи країни загалом. У таких умовах на невеликих ремонтних ділянках раціонально експлуатувати не складні та швидко модифіковані програмні комплекси, створені на основі загальнодоступних офісних додатків, наприклад, таких як Excel або Access.

3. Найважливішою тенденцією у сфері розвитку ремонтних ділянок стане перехід від застосування комп'ютерів на вирішення важливих, але часто ізольованих завдань до створення комплексних інформаційних систем підприємства. Це дозволить:

- скоротити витрати на програмне забезпечення та експлуатацію інформаційних комплексів;
- уніфікувати та значно скоротити кількість вторинних документів;
- повністю виключити дублювання інформації у первинних документах;
- забезпечити контроль виконання прийнятих рішень та отримання оперативних даних про відхилення системи від прийнятих показників ефективності її функціонування.

4. Розширяться традиційне коло завдань, які вирішуються з використанням інформаційних технологій. Щодо роботи ремонтних ділянок виконуватиметься розробка та застосування на практиці системи цільових нормативів, що використовуються при управлінні ефективністю роботи, яка дозволить здійснювати наступні можливості:

- індивідуалізація нормативів до рівня конкретних об'єктів та виконавців;
- створення надійної інформаційної бази, що дозволяє реально управляти виробничими процесами лише на рівні підприємства, цеху, ділянки, постів.

5. Відбудеться вдосконалення та зміна методів та механізмів прийняття управлінських рішень. Наявність оперативно діючих інформаційних систем дозволить реально використовувати економіко-математичні методи на рівні ремонтних ділянок, у тому числі:

- при використанні сучасних методів управління виробництвом та прийняття рішень;
- розроблення та коригування нормативів технічної експлуатації;
- оцінки та управління віковою структурою парку автомобілів та технічною ремонтною базою;
- визначення раціональних та ефективних критеріїв заміни запасних частин в автомобілях, доцільності використання лізингу;
- найбільш оптимальному підборі до ремонтних робіт автомобілів з урахуванням особливостей умов експлуатації;
- розподіл обмежених ресурсів за різними підсистемами ремонтних ділянок.

З'явиться реальна можливість застосування експертних систем після прийняття управлінських рішень.

Експертна система - це програмний комплекс, що включає базу знань (набір взаємопов'язаних правил, формалізує досвід фахівців у деякій галузі) і механізм виведення, що дозволяє на основі правил і факторів, що надаються користувачем, розпізнати ситуацію і дати рекомендації для вибору подальших дій.

На відміну від традиційного програмного забезпечення, яке видає користувачам інформацію про стан об'єкта, експертні системи забезпечують вироблення оптимального рішення щодо управління об'єктом на основі даних про його стан (наприклад, ставлять діагноз про несправність автомобіля, формують набір технічних впливів на основі даних про стан елементів двигуна ). Експертна система включає два елементи:

- база даних - набір факторів, що характеризують поточний стан об'єкта управління;
- база знань - набір правил, що визначають алгоритми пошуку оптимального рішення.

З використанням експертних систем вирішуватимуться завдання діагностування та пошуку несправностей у складних системах двигунів, розстановки автомобілів на посади поточного ремонту, формування оптимальної послідовності виконання технологічних операцій технічного обслуговування оперативного управління витратами та вирішення інших завдань.

Робота експертної системи базується на двох основних класифікаторах:

- причин погіршення показників роботи рухомого складу (незадовільний технічний стан автомобілів, низька якість технічного обслуговування, недостатня кваліфікація водіїв, важкі умови експлуатації, неякісні експлуатаційні матеріали та інші завдання);

- заходів (технічних, організаційних, адміністративних), вкладених у усунення названих причин.

Ці відомості формуються кваліфікованим екпертом та заносяться в основу знань експертної системи. Крім того, обов'язково повинні бути присутніми три підсистеми:

- врахування фактичних показників роботи рухомого складу (облік витрати палива, запасних частин, шин, виконання технічного обслуговування, ремонтів, пробігу та інших завдань);

- розрахунок нормативних показників роботи рухомого складу ремонтних ділянок;

- аналіз роботи автомобілів, водіїв та підрозділів ремонтних ділянок.

Внаслідок роботи цих елементів експертної системи персонал ремонтних ділянок отримує таку інформацію:

- перелік об'єктів, що мають відхилення від нормативних показників роботи (автомобілі з підвищеною витратою палива, підрозділи з високими показниками за простоями тощо);

- перелік винуватців наднормативних витрат (водії, підрозділи, бригади, автомобілі тощо);

- перелік заходів, спрямованих на усунення причин відхилення показників роботи персоналу та автомобілів від нормативів.

З використанням цього підходу можна управляти, наприклад, витратою палива, ресурсом шин, простоями на ремонтних ділянках.

6. Розпочнеться перехід до мережевих комп'ютерних технологій, територіально-розподілених мереж, які забезпечують ремонтним ділянкам та їх філіям оперативний обмін інформацією, доступ до центральної бази даних, ресурсів галузевої, національної та глобальної мереж. Всі ці можливості надають інтранет- та інтернет-технології.

Інтранет (Intranet) – це внутрішня регіональна мережа підприємства. Вона може бути повністю ізольованою від «зовнішнього світу», або мати вихід у глобальну мережу. Інтрамережі базуються на технології «клієнт-сервер». При цьому віддаленість філій ремонтних ділянок від сховища даних не має принципового значення, оскільки як канали зв'язку використовуються сучасні засоби передачі даних.

Використовуючи інтрамережі, працівники ремонтних ділянок можуть занести свою інформацію в центральне сховище даних, так і витягти зі сховища будь-яку інформацію, на доступ до якої вони мають відповідні права. Насамперед ці технології знайдуть застосування в організаціях автосервісу, великих регіональних складах запасних частин, у компаніях, що займаються міжнародними та міжміськими перевезеннями вантажів та пасажирів.

Прикладом інтранет-системи може бути велика сервісна служба, що займається технічним забезпеченням та ремонтом автомобілів. Вона має регіональні склади запасних частин, розподілені територією країни, та мережу ремонтних майстерень. Єдиним джерелом інформації є загальний банк даних, з яким мають комп'ютерний зв'язок усі склади, що набувають можливість отримувати оперативну інформацію про наявність запасних частин у будь-якому з регіональних складів, поточні ціни на запчастини та матеріали. Майстерні немає необхідності мати власні складські приміщення, не потрібно заморожувати кошти у власному незнижуваному запасі. Можна оперативно сформулювати замовлення на комплект запасних частин під будь-якого клієнта, який звернувся до майстерні, виходячи з мінімуму витрат на придбання та доставку, і відразу виставити рахунок клієнту за виконання послуг з ремонту та обслуговування автомобіля.

Використання подібних систем дозволяє працівникам ремонтних майстерень оперативно орієнтуватися у великих масивах інформації та мати економію внаслідок скорочення часу та коштів на придбання та доставку запасних частин, матеріалів та обладнання, що особливо важливо при зміні конструкції автомобіля, агрегатів та комплектуючих виробів.

Інтернет – це відкрита для загального доступу світова комп'ютерна мережа. Через цю мережу здійснюється обмін електронними листами, ведеться торгівля рухомим складом, запасними частинами, паливом, матеріалами, технологічним обладнанням, поширюється інформація щодо надання сервісних послуг (технічне обслуговування та ремонт автомобілів), укладаються договори та контракти, виконуються платежі, здійснюється розповсюдження нормативних та законодавчих актів, відстежується переміщення транспортованих вантажів та інше.

В останні роки з'явилося багато невеликих ремонтних майстерень. Їхнє виживання та конкурентоспроможність залежатимуть від кількості залучених клієнтів, швидкої орієнтації у цінах на послуги, запасні частини та матеріали, ефективності реклами своєї діяльності. Підвищенню ефективності їхньої роботи все більше сприятиме розвиток інтернету.

7. Почнеться перехід підприємств на нові сучасні програмно-технічні комплекси. Це з появою потужніших обчислювальних машин, швидким поширенням прогресивних Windows-технологій та інших, напівпромислових і промислових систем управління базами даних. Застосування таких комплексів забезпечує суттєве підвищення надійності та продуктивності інформаційних систем при значному зниженні трудовитрат на їх розробку та експлуатацію.

8. При створенні інформаційних систем відбудеться перехід від «кустарних» способів вирішення проблем до послуг спеціалізованих підприємств та консалтингових фірм, які здійснюють проектування, монтаж, налагодження мереж, супровід системного та прикладного програмного забезпечення. Це пояснюється тим, що створення комплексних інформаційних систем потребує значних витрат часу та інтелектуальної праці.

9. Масово пошириться використання бортових комп'ютерів автомобілів для збору інформації про стан найважливіших систем та агрегатів з подальшою передачею цих даних до інформаційної системи підприємства для формування рекомендацій щодо тактики обслуговування та ремонту автомобілів.

10. Відповідно до застосовуваних інформаційних систем підвищиться кваліфікація персоналу. Технічний персонал повинен мати навички роботи з готовими комп'ютерними системами. Інженерний персонал повинен вміти грамотно формулювати та ставити завдання програмістам, виконувати аналіз даних за допомогою комп'ютерної техніки та програм загального призначення (MS Office), вносити пропозиції щодо розвитку та удосконалення діючих на підприємстві інформаційних систем. Керівний персонал повинен розуміти тенденції розвитку інформаційних технологій, знати їхні можливості та бачити перспективи їх застосування на своїх підприємствах.

Таким чином, застосування нових інформаційних технологій є найважливішою передумовою розвитку та вдосконалення систем управління якістю технічної експлуатації та сервісу на ремонтних майстернях. Стане можливим перехід від окремих фрагментів таких систем (облік напрацювання на відмову, простоїв у ремонті, витрати запасних частин та матеріалів) або фіксації рівня якості та її невідповідності наявним вимогам щодо запобігання неякісному виконання послуг.

Для ремонтних ділянок автотранспортних та сервісних підприємств управління якістю технічного обслуговування та ремонту забезпечуватиметься:

- застосуванням науково обґрунтованих методів та процедур прийняття та реалізації керуючих рішень;
- розробкою та реалізацією багаторівневої системи цільових нормативів, виконання яких може бути узагальнюючим показником якості функціонування конкретних виконавців, бригад, дільниць, цехів та ІТС у цілому;
- визначенням поняття якості роботи конкретного виконавця, яке має характеризуватись, насамперед, дотриманням технології, виконанням призначеної роботи точно в строк та забезпечувати задані цільові нормативи простою

автомобілів на технічне обслуговування та у ремонті та гарантоване напрацювання після технічного обслуговування та ремонту;

- доступністю оперативної індивідуальної (автомобіль, виконавець) інформації та її використанням при управлінні виробництвом та оцінці його ефективності на ремонтних ділянках;

- розробкою та застосуванням об'єктивних нормативів технічного обслуговування та ремонту, їх коригуванням з урахуванням умов експлуатації автомобілів на ремонтних ділянках;

- раціональним технологічним та нормативним забезпеченням виробничих процесів, що включають забезпечення робочого місця на ремонтних ділянках;

- підвищенням кваліфікації персоналу, удосконаленням методів його відбору, підвищенням престижу професії працівників ремонтних ділянок;

- удосконаленням методів контролю якості (використовуються матеріали та запасні частини, поточний виробничий та вихідний контроль) на ремонтних ділянках;

- матеріальної та моральної зацікавленістю персоналу як виконувани роботи, відповідальністю за кінцеві результати (відмови, простої, рекламации, втрата клієнтури) на ремонтних ділянках.

#### 1.4 Постановка задачі

Для розробки та впровадження програмної системи обліку поставок запасних частин до мережі ремонтних ділянок необхідно виконати наступні завдання:

- провести аналіз моделей управління поставками запасних частин у мережу ремонтних ділянок та порівняльний аналіз аналогів;

- виконати аналіз розвитку сучасних систем обліку поставок запасних частин до мережі ремонтних ділянок;

- створити постановку задачі;

- виконати проектування модулів системи обліку поставок запасних частин для розділу приватних постачальників;



- виконати проектування модулів системи обліку поставок запасних частин для розділу власника бізнесу;
- виконати проектування модулів системи обліку поставок запасних частин для розділу технічної підтримки;
- розробити модуль обслуговування заявок на ремонтних ділянках;
- розробити алгоритм роботи відділу постачання запасних частин на ремонтні ділянки;
- розробити інтерфейс програмної системи обліку поставок запасних частин;
- виконати тестування модуля заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок;
- виконати тестування frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок;
- виконати тестування backend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок.

#### 1.4 Висновки

В першому розділі виконано аналіз предметної області, де на основі моделі управління поставками запасних частин у мережу ремонтних ділянок враховуються кращі практики в обслуговуванні та ремонті автомобільного транспорту, схеми класифікації за методами ABC. Ретельно розглянуті за моделі та методи ABC, де описуються ролі персоналу технічного обслуговування як постачальник послуг.

Для розробки моделі управління постачанням запасних частин на основі використання логістичних принципів вирішені важливі завдання. Для більш ефективної розробки моделі управління постачанням запасними частинами на основі використання логістичних принципів застосовувались різні підходи.

Виконувалась порівняльна характеристика аналогів проводилась серед десктопних (Автоділер) та хмарних платформ (LiveSkлад) програмно-технічних середовищ, які існують в Україні, де звертала на себе увагу велика кількість десктопних додатків.

Завдяки цьому, під час експлуатації таких програмних засобів існують численні центри впровадження, які впливають на гнучку подальшу розробку автоматизованих систем.

Проведений аналіз розвитку сучасних систем обліку поставок запасних частин до мережі ремонтних ділянок показав, що в даний час у системах обліку поставок запасних частин у мережу ремонтних ділянок, включаючи підсистему технічної експлуатації, відбуваються суттєві кількісні та якісні зміни в галузі інформаційного забезпечення виробничих процесів, які через кілька років призведуть до особливих змін.

Виконана постановка задачі, де були визначені основні кроки щодо розробки та впровадження програмна система обліку поставок запасних частин до мережі ремонтних ділянок.

## 2 ПРОЕКТУВАННЯ МОДУЛІВ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ПОСТАВОК ЗАПАСНИХ ЧАСТИН

Проектування та створення модулів системи обліку поставок запасних частин до мережі поділяється на три великі частини, які пов'язані роботою постачальників (рис. 2.1).

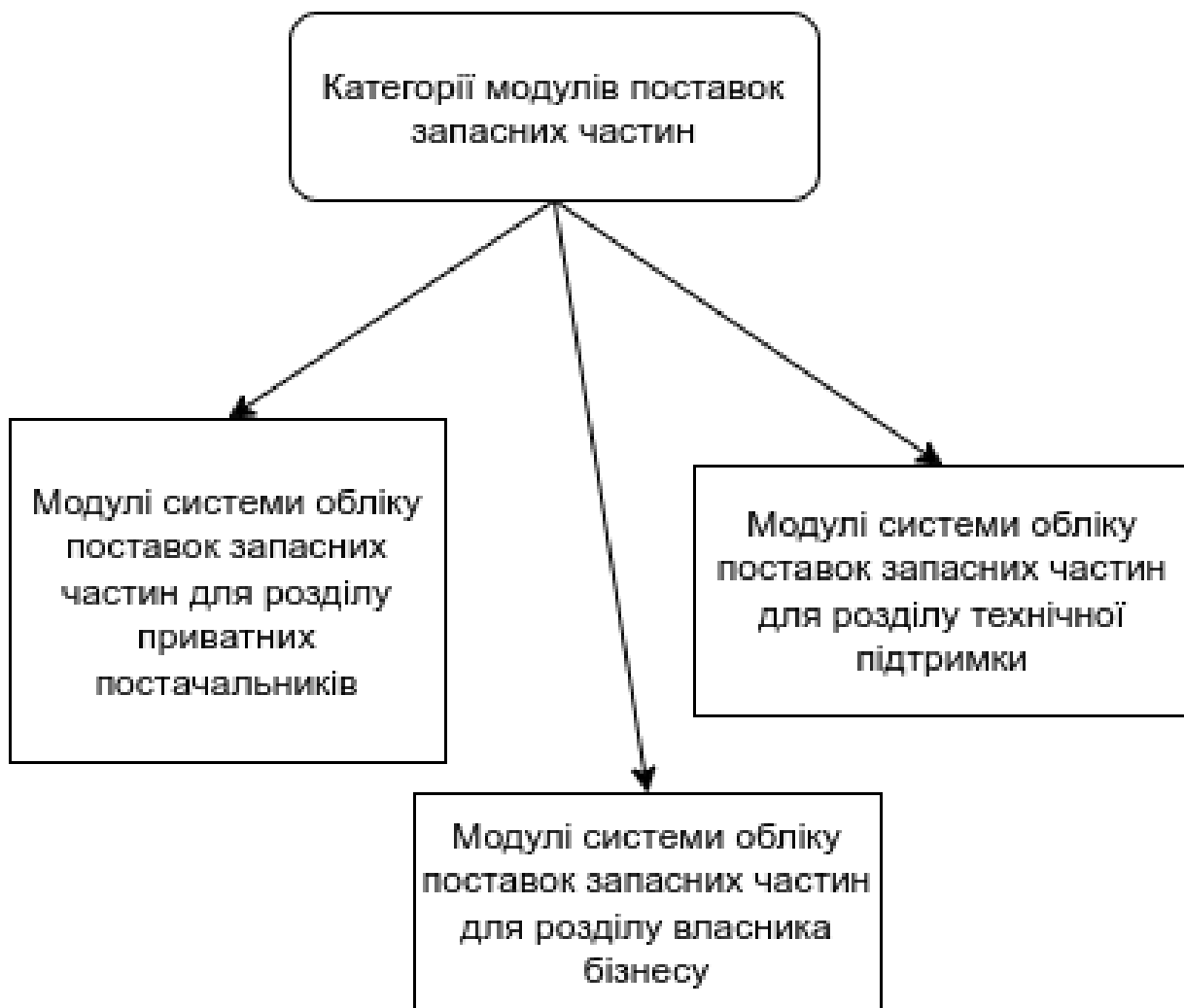


Рисунок 2.1 – Категорії модулів поставок запасних частин

Перш, за все, такі умовні категорії поділяються на певні розділи до яких відноситься робота з приватними постачальниками, бізнес-структурами та технічної підтримки з супроводу системи.

### 2.1 Проектування модулів системи для розділу приватних постачальників

Проектування модулів системи обліку поставок запасних частин до мережі ремонтних ділянок є важливим етапом розробки програмного засобу та

виконується на основі вимог. Опишемо такі вимоги та функціональні призначення кожного модуля (рис. 2.2).

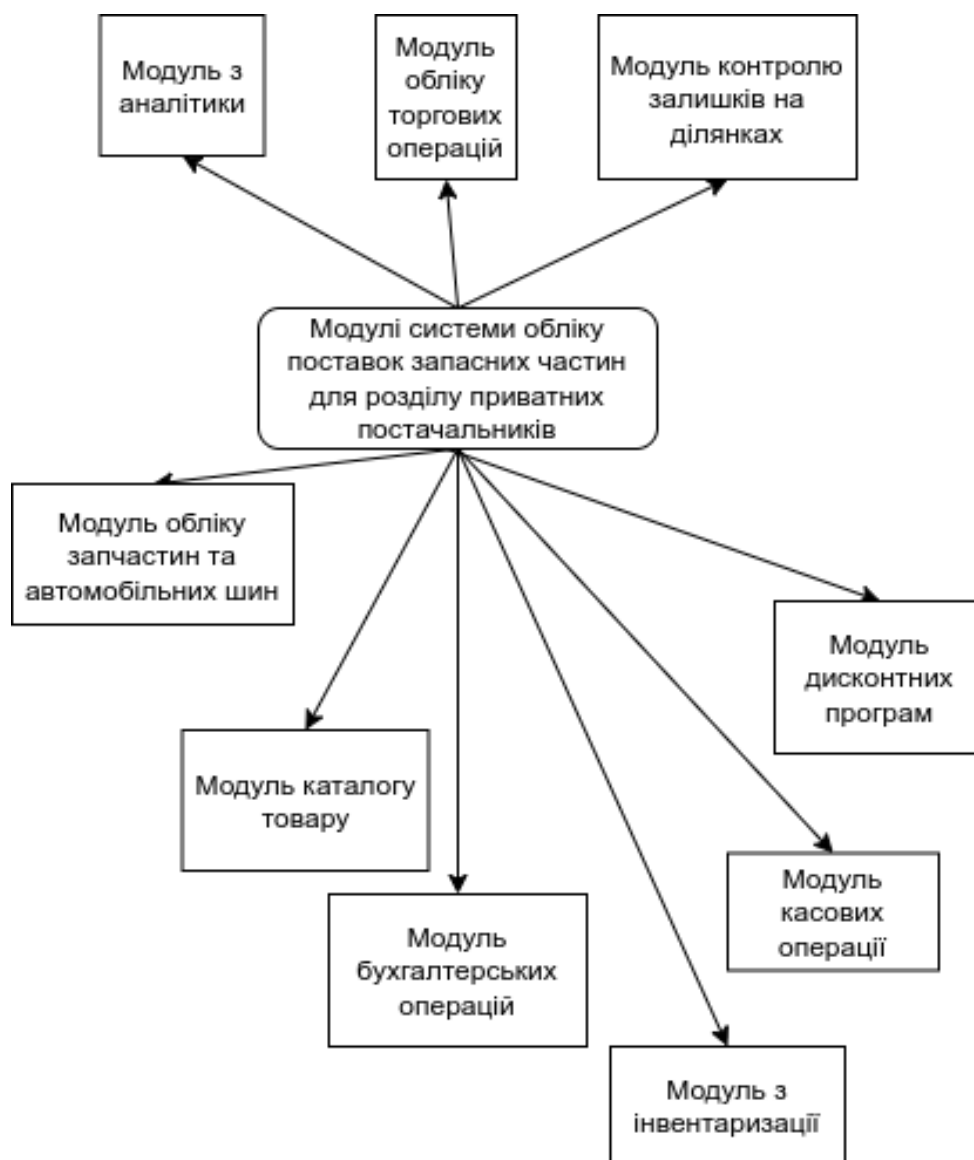


Рисунок 2.2 – Основні модулі системи обліку поставок запасних частин для розділу приватних постачальників

Модуль обліку запчастин та автомобільних шин має певну специфіку так як деякі запчастини можуть використовуватись лише в автомобілях певної моделі, інші – у кількох моделях. Цю інформацію необхідно фіксувати у системі.

Модуль каталог товару складає повну номенклатура товару які є в наявності та у замовленнях, категорії. Інформація про товар може містити довільну кількість полів, яка може додаватись при необхідності. До таких полів відноситься

собівартість продукції, націнка, дата приходу, термін придатності, мінімальний залишок тощо.

Для кожної запасної частини довідник "Номенклатура" вводиться окрема картка. Картка містить інформацію про виробника та країни походження запасної частини, про моделі автомобілів, в яких вона може бути використана, про аналогічні деталі, які можуть замінити штатну запасну частину у разі її відсутності на складі, про мінімальний складський залишок та інші дані. Для автомобільних шин додатково вказується інформація про типорозміри, тобто (радіус, ширина, висота). Також вводяться й інші поля, такі як профілі (радіальний, діагональний і т.д.), сезонність, індекси навантаження та швидкості. У конфігурації довідника також передбачено також облік номерних агрегатів.

Під час роботи, менеджер може призначати номенклатурним одиницям, і навіть контрагентам і підрозділам ще додаткові довільні ознаки. Так, наприклад, для контрагентів, наприклад, можна ввести ознаки "Регіон" та "Платоспроможність".

Модуль бухгалтерських операцій призначений для роботи із накладними. Переважно, робота складається з заповнень прибуткових, видаткових та зворотних накладних. Також, до модуля входить робота з актами переоцінки (автоматичні, ручні), відстрочена переоцінка.

Модуль з інвентаризації складається з заповнень електронних документів по наявним технічним засобам, аналізу підсумків перевірок, локальних інвентаризацій ремонтних участків.

Модуль касових операцій складається з електронних документів обліку, комбінованих способів оплати постачальників (користувачів системи): готівка, картка, сертифікат, безготівка, бонуси, борг.

Модуль з аналітики складається з електронних документів контролю каси, взаємозаліків із постачальниками, інші витрати, зарплата. Розрахунок бонусів працівникам, прибуток продавців.

Модуль дисконтних програм має проводити облік операцій з картками, накопичувальними знижками, сезонними та тимчасовими знижками на товари та різні категорії, наявність продажу у борг.

Модуль обліку торгових операцій має свої особливості. Так, існує особливість торгівлі автомобільними запчастинами, яка полягає у широкій номенклатурі деталей і, як наслідок, у відсутності складських запасів з більшості позицій номенклатури. Тому, у цьому випадку основним способом роботи є торгівля на замовлення.

Модуль контролю залишків на ділянках. До цього модуля відноситься формування гнучких звітів щодо товару, різних видів накладних, які дозволяють швидко та без зусиль провести будь-яку операцію з товаром. Модуль замовлень полягає у відображенні поточної інформації щодо замовлень покупців та постачальників. Так, для відображення в обліку замовлень у конфігурації використовуються такі документи як "Замовлення покупця" та "Замовлення постачальнику". У документі "Замовлення покупця" вводиться інформація про замовника, замовлені товари, категорію цін, підрозділ, у якому оформлено замовлення, та інше.

У разі необхідності існує можливість вказання знижки на замовлення. При цьому, існує певна інформація про менеджера, який оформив замовлення або відповідального за його виконання, а також джерело інформації, за допомогою якого замовник дізнався про дану компанію. Дані про джерело інформації надалі можуть бути використані для аналізу ефективності проведення реклами.

Для розділу "Замовлення покупця" можна запровадити документ "Замовлення постачальнику", і навіть різні документи реалізації товару. У конфігурації може проводитись облік нерозподілених замовлень, тобто таких замовлень, товари за якими були замовлені власне постачальникам.

Крім торгівлі на замовлення у конфігурації модуля міститься необхідні документи для організації оптової та роздрібною торгівлі, а також для торгівлі товарами, прийнятими або переданими на комісію.

## 2.2 Проектування модулів системи для розділу власника бізнесу

До системи обліку поставок запасних частин для власника бізнесу також відносяться важливі модулі, що використовуються у повсякденні. До таких модулів відносяться наступні (рис 2.3).

Модуль складського обліку, що ведеться з певною деталізацією, де вказана інформація яка поділяється за розділами власними бізнесу, підрозділами (складами), постачальниками, номенклатурою, її видами та властивостями. Також, в цьому ж модулі передбачено інвентаризацію та переоцінку як власних товарів, так і товарів, отриманих та переданих на реалізацію. Для реєстрації цих операцій застосовуються типові документи.

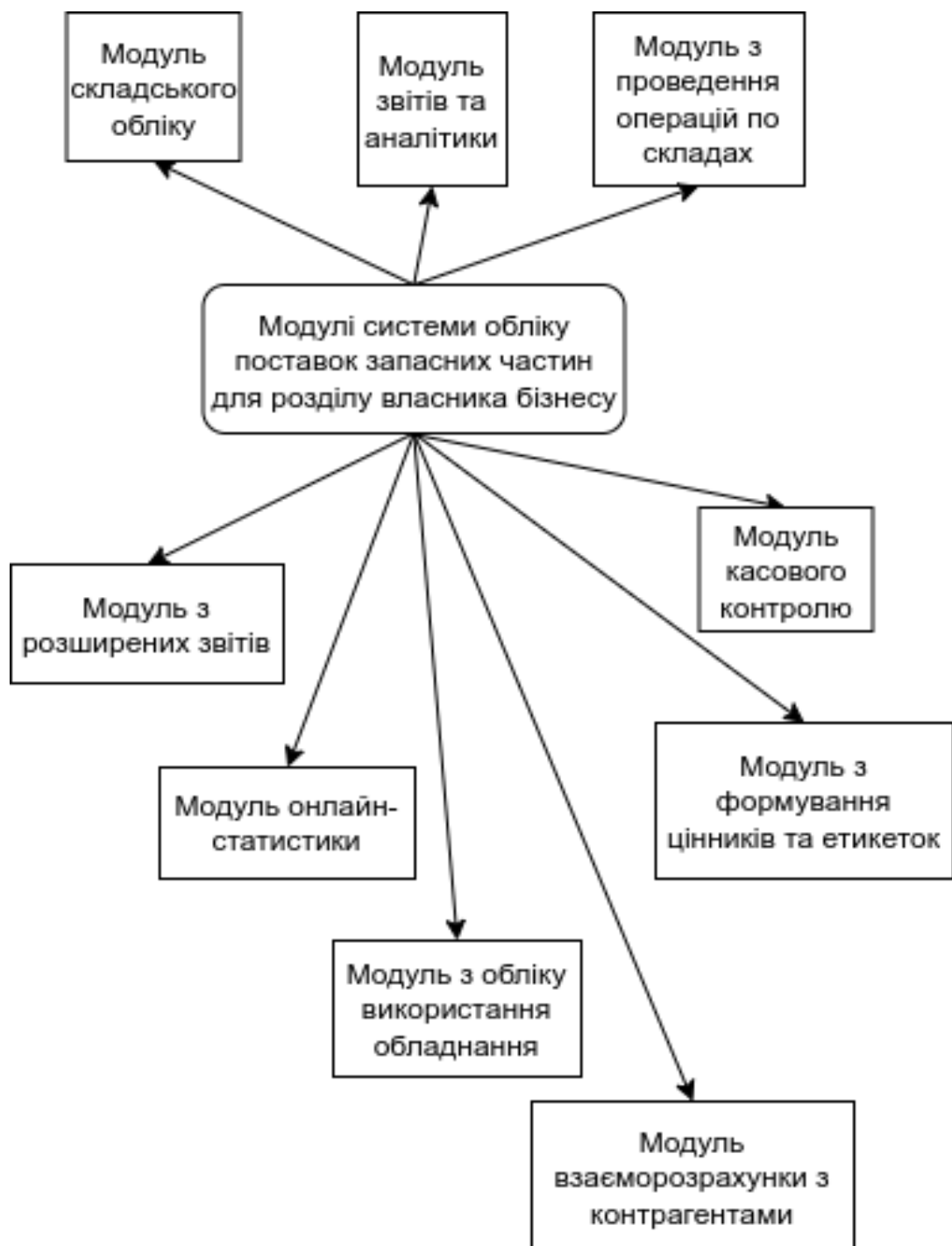


Рисунок 2.3 – Основні модулі системи обліку поставок запасних частин для розділу власника бізнесу

В цьому ж модулі присутнє заповнення цілої низки важливих документів. Так, за допомогою документа "Акт комплектації" відображається операція зі створення комплектів запасних частин.

Під час реєстрації документа зі складу автоматично списуються комплектуючі та надходять готові комплекти. Для реєстрації операції розбирання комплектів використовується документ "Акт розукомплектації". Ведеться облік додаткових витрат, що супроводжують операції комплектації, розукомплектації та переміщення партії необхідних деталей.

Модуль звітів та аналітики здійснює підтримку ведення електронних документів у розрізі звітів з продажу запасних частин та чекам за будь-який період. Такі звіти формуються з продажу та прибутку в розрізі категорій товарів та постачальників, суми залишків за категоріями, постачальниками, детальна інформація по артикулу.

Особливо важливими є звіти зі строків придатності товарів, звіт про залишки з граничним терміном придатності.

Модуль з проведення операцій по складах здійснює проводки документів відповідно у розрізі складів та ще виконує функції з формування звітів зі списання комплектуючих. У цьому ж модулі існує можливість формування звітів по всіх типах накладних та пошуку необхідної накладної.

За необхідністю може формуватись детальний звіт про повернення та скасування позицій, контроль скасування часу для синхронізації з відеоспостереженням, звіти з продажу фіскальних товарів.

До того ж може додаватись розширений звіт щодо застосування дисконтів та акцій, додаткових касових звіт, звітів з інших витрат, інкасацій та інше.

В цьому ж модулі існують досить розширені звіти щодо руху товару або групи товарів, звіти щодо лідерів/аутсайдерів продажів, сезонних коливань попиту на запасні частини, тощо. При цьому, існує можливість перегляду складу на будь-яку дату.



Модуль з розширених звітів щодо дій користувачів виконує прив'язку всіх операцій у програмі користувача та надає можливість створення додаткових звітів без залучення виробника програмного забезпечення.

Модуль онлайн-статистики, який дозволяє переглядати певні дозволені звіти з мобільного телефону як у WEB версії для IOS та Android, так і з окремої програми для Android.

Модуль з обліку використання обладнання. Так як для організації роздрібною торгівлі у конфігурації підтримується широкий спектр торговельного обладнання - сканери штрих-кодів, фіскальні реєстратори, контрольно-касові апарати (ККМ), то існує необхідність у створення даного модуля.

Даний модуль використовується з метою підтримки широкого спектру торговельного обладнання: сканерів та принтери штрих-кодів, принтерів чеків, фіскальні реєстратори, POS термінали, ваги з печаткою етикетки та багато іншого. В цьому ж модулі існує фіксація про відправлення даного обладнання у ремонт та списання.

Модуль касового контролю надає можливість отримати поточну статистику про прибуток участків продажу запасних частин. В цьому ж модулі надається інформація про взаєморозрахунки з постачальниками, дані про поточну касу місця та безготівкові розрахунки.

В цьому ж модулі реалізовано багатовалютний облік коштів, для оформлення операцій з руху коштів у касі та на розрахунковому рахунку передбачено набір відповідних документів. Передбачається ведення взаєморозрахунків із контрагентами, підзвітними особами та філіями.

Модуль з формування цінників та етикеток. Цей модуль забезпечує графічні форми щодо друку цінників та етикеток.

Модуль звітів з управлінського балансу підприємства. В цьому модулі реалізовано формування звіту "Активи і пасиви", що є управлінський баланс підприємства з фірмам і підрозділам, які входять у організацію. У цьому звіті представлена сукупна інформація про залишки та рух грошових коштів, товарів на складах та переданих на реалізацію, а також про борги контрагентів та зміну сум боргів за вибраний період.

Велика група звітів призначена для отримання та аналізу інформації як по компанії в цілому, так і по кожній фірмі та підрозділу.

За допомогою звіту "Доходи та витрати" можна отримати інформацію про фінансовий результат кожної господарської операції. У звіті "Рух грошових коштів" представлені дані про залишки та обороти грошових коштів по касі, розрахунковому рахунку та про цінні папери у грошовому еквіваленті.

Модуль взаєморозрахунки з контрагентами використовується для аналізу стану взаєморозрахунків. Для звірки взаєморозрахунків між фірмою та контрагентом застосовується звіт "Акт звіряння взаєморозрахунків". Для отримання інформації про замовлення клієнтів та замовлення постачальникам, резервування, реалізації, касі у конфігурації є також відповідні звіти. В цьому ж модулі досить важливим є аналіз ефективності реклами та оцінка діяльності менеджерів також виробляються за допомогою звітів.

### 2.3 Проектування модулів системи для розділу технічної підтримки

До системи обліку поставок запасних частин для технічної підтримки також відносяться важливі модулі, що використовуються у повсякденні. До таких модулів відносяться наступні (рис. 2.4).

Модуль по роботі програм з базами даних, що дозволяє зберігати дані користувачів на локальному комп'ютері, в мережі, на віддаленому сервері в будь-якій країні світу. Для цього впроваджені засоби захисту від несанкціонованого доступу до баз даних навіть у разі вилучення обладнання так як використовуються механізми криптування.

Інший модуль являє собою Front End, що використовує модулі з інших дрібних підрозділів, таких як касовий модуль, модуль продажу. Програмні засоби, що підтримують технології Front End можуть працювати на будь-яких пристроях: планшеті, мобільному телефоні, смартфоні, комп'ютері. В цьому ж модулі існують зручні засоби управління віддаленими записами на сервері, що пристосовані для виїзної торгівлі, кафе, точок продажу з обмеженим простором, підтримкою мобільних торгових представників компанії. Також, досить вдалим рішенням для впровадження даного модуля є можливість віддаленого друку чеків на будь-якому

принтері та підтримка інших пристроїв локальної мережі, що працює на різних платформах та операційних системах.

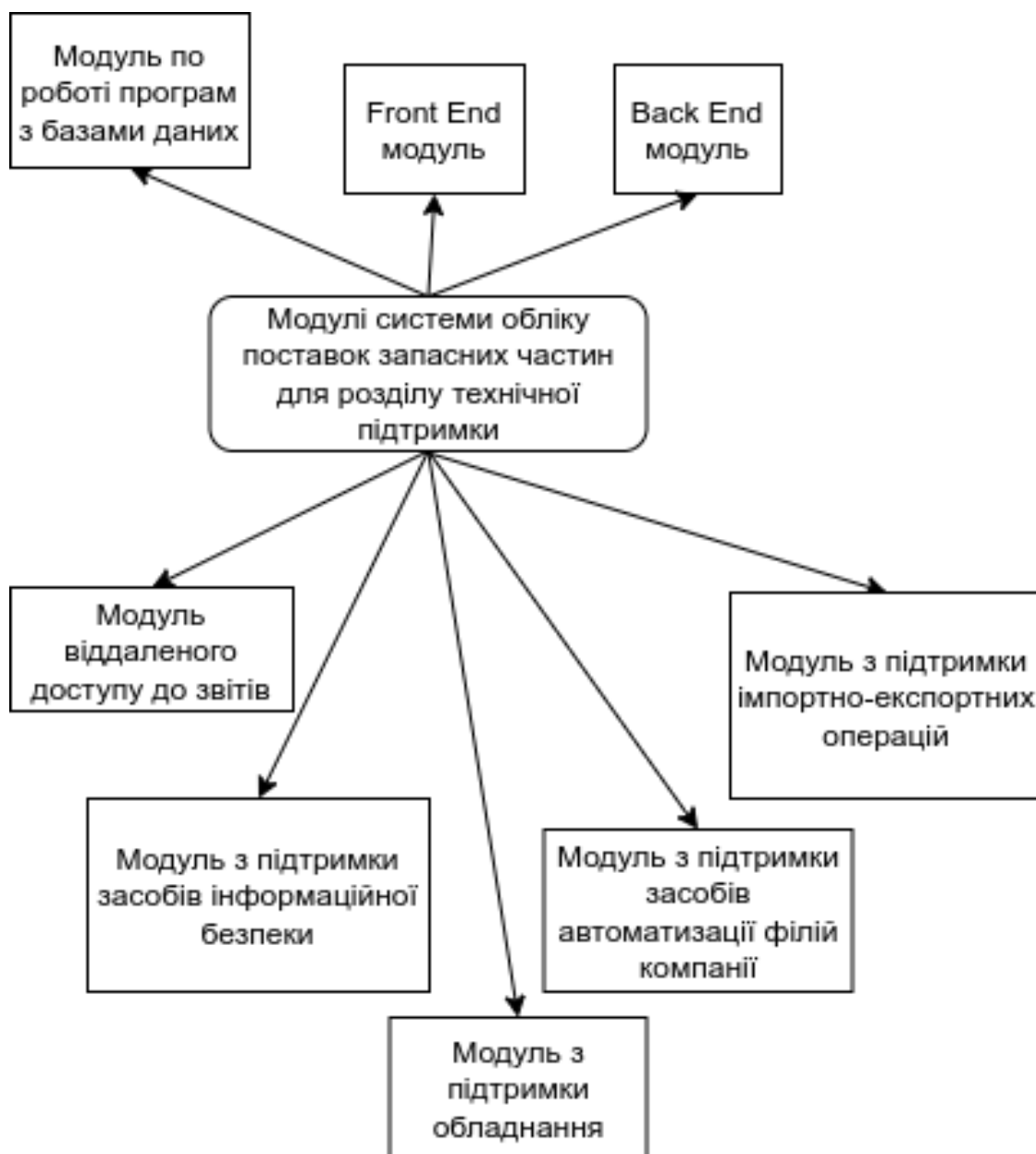


Рисунок 2.4 – Основні модулі системи обліку поставок запасних частин для розділу технічної підтримки

Модуль Back End, що реалізований на операційній системі Linux підтримує різні запити, що надходять з різних операційних систем та платформ. За спеціальною розробкою можливе встановлення даного модуля з обмеженим функціоналом в операційній системі MS Windows. Даний модуль використовує ліцензовані програмні засоби з OpenSource та мови програмування. Він є

відкритим та має можливість доробки різних компонент на мовах програмування Java та Python.

Також, в цьому модулі реалізовано функціонал у конфігурації "Альфа-Авто: Автозапчастини + Автошини", що надає можливість вивантаження даних у типову конфігурацію "1С:Бухгалтерія" за різними документами.

Модуль віддаленого доступу до звітів дозволяє контролювати бізнес операції з будь-якої точки світу, з будь-якого пристрою, без встановлення ПЗ, за будь-якої швидкості інтернету.

Модуль з підтримки імпортно-експортних операцій виконує імпорт-експорт даних у більшість популярні формати (xls, csv, html, mdb), мається можливість інтеграції з іншими програмними додатками.

Модуль з підтримки засобів інформаційної безпеки здійснює підтримку та моніторинг за розмежуванням доступу до інформаційних ресурсів корпоративної мережі. В цьому ж модулі здійснюється облік доступу співробітників за допомогою встановлених політик безпеки, де описуються лише ті функції, які їм потрібні під час роботи з системою. Також, усі дії користувачів фіксуються у логах (звітах) які зберігаються в системі. За особливими потребами передбачена можливість хмарного зберігання такої бази даних. Це надає можливість додаткового захисту інформації від несанкціонованого доступу, навіть у разі вилучення обладнання.

Модуль з підтримки засобів автоматизації філій компанії: магазину, складу та супермаркету здійснюють програмну підтримку зі встановленню зв'язку між цими філіями та призначений для обміну даними, що виконуються переважно в запланованому режимі. В цьому ж модулі реалізовано підтримка обміном накладними, електронними документами з центральним складом, одночасну роботу з кількома торговими точками з одного робочого місця.

Переважно такий обмін здійснюється даними з програмного додатку "Автокаталог" фірми "Автософт", де виконується електронна обробка різних версій каталогів запасних частин з найпоширеніших автомобілів і двигунів, що існують на ринку України. Це дозволяє, наприклад, завантажити із програми "Автокаталог" перелік деталей конкретних моделей ВАЗ із номерами за каталогом

заводу-виробника. Після завантаження даних можна переглянути зображення деталі, що мають на складах компанії або філій, вибравши її у довіднику "Номенклатура".

Модуль з підтримки обладнання виконує діагностичні операції з встановлення працеспроможності апаратних засобів, що мають у корпоративній мережі. До таких апаратних засобів відносяться: сканери штрих-коду (будь-який тип підключення), принтери зі штрих-кодом, фіскальні реєстратори (Datecs, Exelio, Марія), зчитувачі дисконтних карток та інше обладнання.

Такі операції виконуються внаслідок того, що багато з апаратного обладнання має IP адреса. Тому, за допомогою системних утиліт таких як ping та інші вдається визначати працеспроможність апаратних засобів. Результати таких операцій заносяться до системи моніторингу, де технічний персонал має можливість переглядати стан апаратних засобів в будь-який момент часу на різних філіалах компанії.

## 2.4 Висновки

В другому розділі було виконано проектування та створення модулів програмної системи обліку поставок запасних частин до мережі, яка поділялась на три великі частини, що були пов'язані з роботою постачальників. Перш, за все, такі умовні категорії поділяються на певні розділи до яких відноситься робота з приватними постачальниками, бізнес-структурами та технічної підтримки з супроводу системи.

Проектування модулів системи обліку поставок запасних частин до мережі ремонтних ділянок був важливим етапом розробки програмного засобу, що виконувався на основі певних вимог. Тому, в цьому розділі були описані такі вимоги та функціональні призначення кожного модуля.

## 3 РОЗРОБКА МОДУЛІВ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ПОСТАВОК ЗАПАСНИХ ЧАСТИН

### 3.1 Розробка модуля обслуговування заявок на ремонтних ділянках

Для успішного функціонування сучасної мережі ремонтних ділянок є автоматизація бізнес-процесів. Автоматизація роботи мережі ремонтних ділянок при роздрібній торгівлі запасними частинами автомобілів була пов'язана з контролем роботи торгових точок з офісу. Для максимально ефективної роботи мережі ремонтних ділянок необхідно частково або повністю (якщо це можливо) автоматизувати їхню діяльність.

Сучасні розробки в галузі інформаційних технологій дозволяють автоматизувати бізнес-процеси на всіх рівнях мережі ремонтних ділянок. Тому основним завданням, яке необхідно вирішити керівництву мережі ремонтних ділянок, є автоматизація їхньої діяльності.

Програмна система обліку поставок запасних частин до мережі ремонтних ділянок повинна мати такі властивості:

- бути динамічною та легко змінюванюватись;
- бути інтуїтивно зрозуміло;
- мати високу захищеність.

Також, програмна система обліку поставок запасних частин до мережі ремонтних ділянок, що розробляється, повинна забезпечувати автоматизований облік і контроль замовлень від користувачів. Для цього вона повинна:

- забезпечувати введення, обробку та зберігання даних на замовлення;
- проводити планування та контроль, а також розрахунок фінансових ресурсів компанії;
- вести облік та коригування заборгованості, проведення взаємозаліків, списання заборгованості;
- створювати документи та звіти для обліку замовлень у системі;
- забезпечувати облік ціноутворення;
- виконувати роботу в системі яка має бути організована з різними правами доступу.

Розглянемо алгоритм (рис. 3.1) формування та обліку замовлення придбання запасних частин у відділі головного офісу, при якому простежуються такі основні етапи його роботи:

- прийом заявки клієнта менеджером, уточнення та виявлення суттєвих особливостей у роботі;
- формування та пересилання заявки менеджером начальнику відділу розробки та супроводу заявки;
- підтвердження спеціалістом про отримання заявки;
- відстеження менеджером термінів та умов виконання замовлення на розробку/доопрацювання/супровід заявки;
- моніторинг керівником відділу розробки та супроводу заявки та умов, термінів та якості виконання замовлення;
- оповіщення менеджера про готовність виконання замовлення, у разі неготовності замовлення із зазначенням пояснень;
- оповіщення клієнта про статус замовлення, а також результат його виконання;
- при незадовільному узгодженні результатів виконання замовлення вирішення проблем менеджером;
- формування та надсилання клієнту менеджером документів на оплату замовлення;
- контроль менеджером факту оплати замовлення клієнтом;
- надання бухгалтерії менеджером первинних документів;
- складання звітів на замовлення.

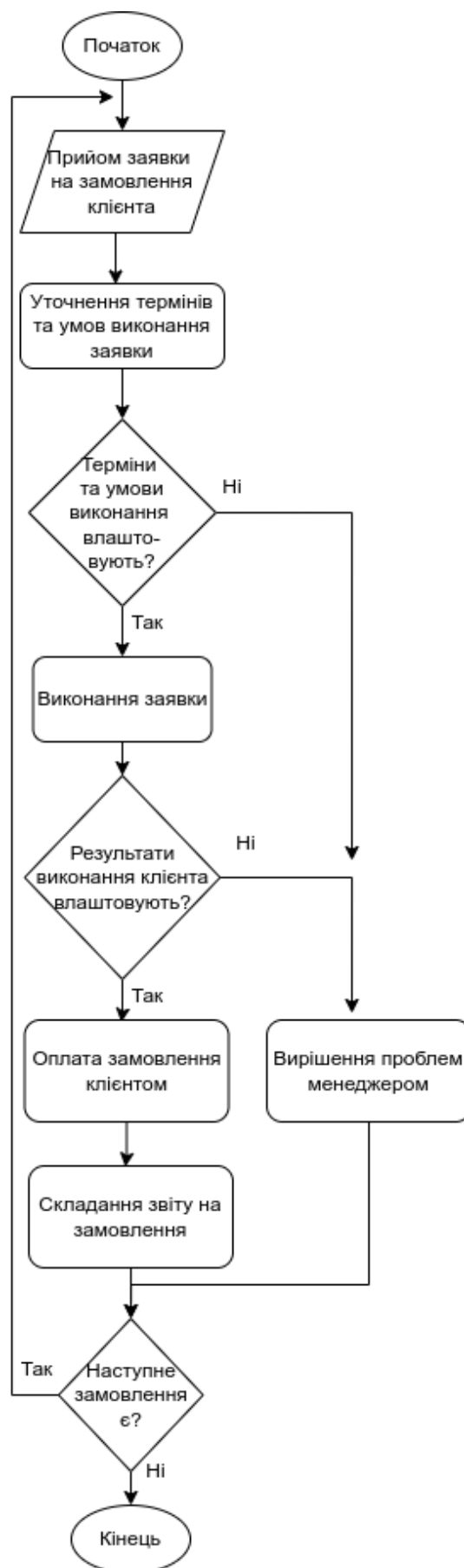


Рисунок 3.1 - Алгоритм формування та обліку замовлення придбання запасних частин у відділі головного офісу



### 3.2 Розробка алгоритму роботи відділу постачання запасних частин

Програмна система обліку поставок запасних частин до мережі ремонтних ділянок, що розробляється, повинна мати змогу розв'язувати завдання які будуть реалізовані відповідним програмним забезпеченням та алгоритму роботи (рис. 3.2). Більшість сучасних програмних розробок спрямовані насамперед те що, щоб значною мірою спростити людству життя. У даній програмній системі обліку поставок запасних частин до мережі ремонтних ділянок на всіх формах існують засоби введення, редагування та пошуку даних будуть реалізовані однотипним способом. Для введення та редагування інформації використані екранні форми, на яких розташовані всі необхідні поля введення, а також керуючі елементи, навігаційні кнопки та пояснення. Екранні форми також використовуються і для пошуку цікавих для даних, що дозволяють задавати різні параметри. Також існують доступні всі необхідні звітні форми для формування та друку потрібних документів.

Основними завданнями автоматизації мережі ремонтних ділянок є:

- облік та зберігання даних про клієнтів та постачальників;
- облік та ведення замовлень на всіх етапах виконання;
- облік розроблених/доопрацьованих інформаційних систем (програм);
- ведення звітності за замовленнями та клієнтами;
- розробка інформаційних систем (програм).

Таким чином, автоматизація роботи на ремонтних ділянках надає можливість оперативніше проводити складні пошуки інформації за різними параметрами, а також швидше оформляти замовлення клієнтів, таким чином продуктивність персоналу підвищиться з допомогою те, що трудомісткість обробки інформаційних потоків вийде оптимальний тимчасової рівень.



Рисунок 3.2 - Алгоритм роботи відділу постачання запасних частин на ремонтні ділянки

### 3.3 Розробка інтерфейсу програмної системи обліку поставок запасних частин

Важливою вимогою у розробці інтерфейсу програмної системи обліку поставок запасних частин у технічне забезпечення. Саме від нього залежить успіх впровадження програмної системи обліку поставок запасних частин. Тому, перед розробкою необхідно визначити загальний парк комп'ютерної техніки, на яких буде працювати програмна система.

У загальному випадку, під технічним забезпеченням розуміється сукупність взаємозалежних технічних засобів, і навіть засобів передачі та комп'ютерної електронної техніки, необхідні використання автоматизованої системи управління інформаційними ресурсами.

По даній дипломній роботі передбачається, що програмна система переважно працюватиме у відділах бухгалтерії, аналітики та ремонтних ділянках, де апаратне забезпечення приблизно матиме таку конфігурацію:

- процесор Intel Core 2 Duo 6320 2,8 ГГц;
- відеоадаптер NVIDIA GeForce 7300 LE (1024 Мб);
- ОЗУ 8 Гб; дискові накопичувачі Seagate ST3160815AS ATA Device (250 Гб, 7200 RPM, SATA-II);
- монітор Samsung SyncMaster 1040N; принтер HP LaserJet 2110.

При цьому, під час розробка інтерфейсу програмної системи обліку поставок запасних частин використовується системне та прикладне програмне забезпечення, тобто комплекси програм, які забезпечують керування основними компонентами операційної системи та комп'ютерної системи, такими як процесор, оперативна пам'ять, пристрої вводу-виводу, мережеве обладнання, виступаючи як «міжшаровий інтерфейс», з одного боку якого апаратура, а з іншого - додатки користувача.

На відміну від прикладного програмного забезпечення, системне не вирішує конкретні практичні завдання, лише забезпечує роботу інших програм, надаючи їм сервісні функції, абстрагуючі деталі апаратної та мікропрограмної реалізації обчислювальної системи, керує апаратними ресурсами обчислювальної системи.

Водночас, системне та прикладне програмні засоби забезпечують роботу інтерфейсу користувача в технічних системах, що є важливим аспектом в управлінні бізнес-процесу мережі ремонтних майстерень.

Розглянемо інтерфейс користувача основного модуля програмного засобу обліку поставок запасних частин до мережі ремонтних ділянок. Основний модуль інтерфейсу програмної системи обліку поставок запасних частин складається з каталогу модулів, що мають свою, кожен окрему функціональність (рис. 3.3).

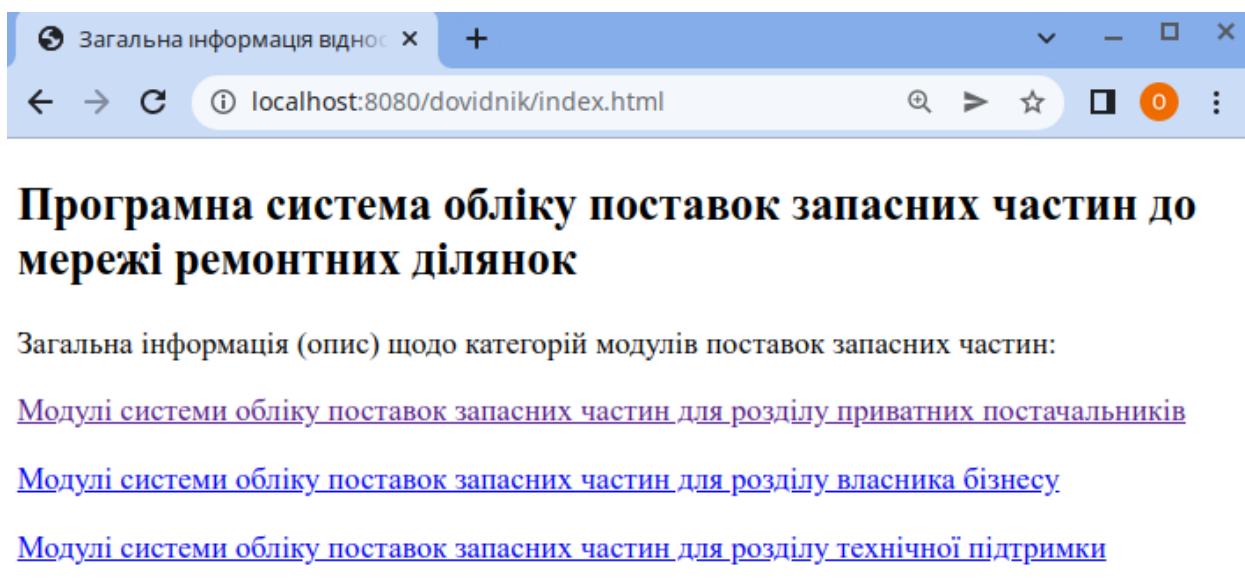


Рисунок 3.3 - Інтерфейс основного модуля програмної системи обліку поставок запасних частин

При цьому, від основного модуля існує можливість переходу до додаткових модулів (рис. 3.4), що складаються з певних розділів та необхідні під час роботи співробітників мережі поставок запасних частин.

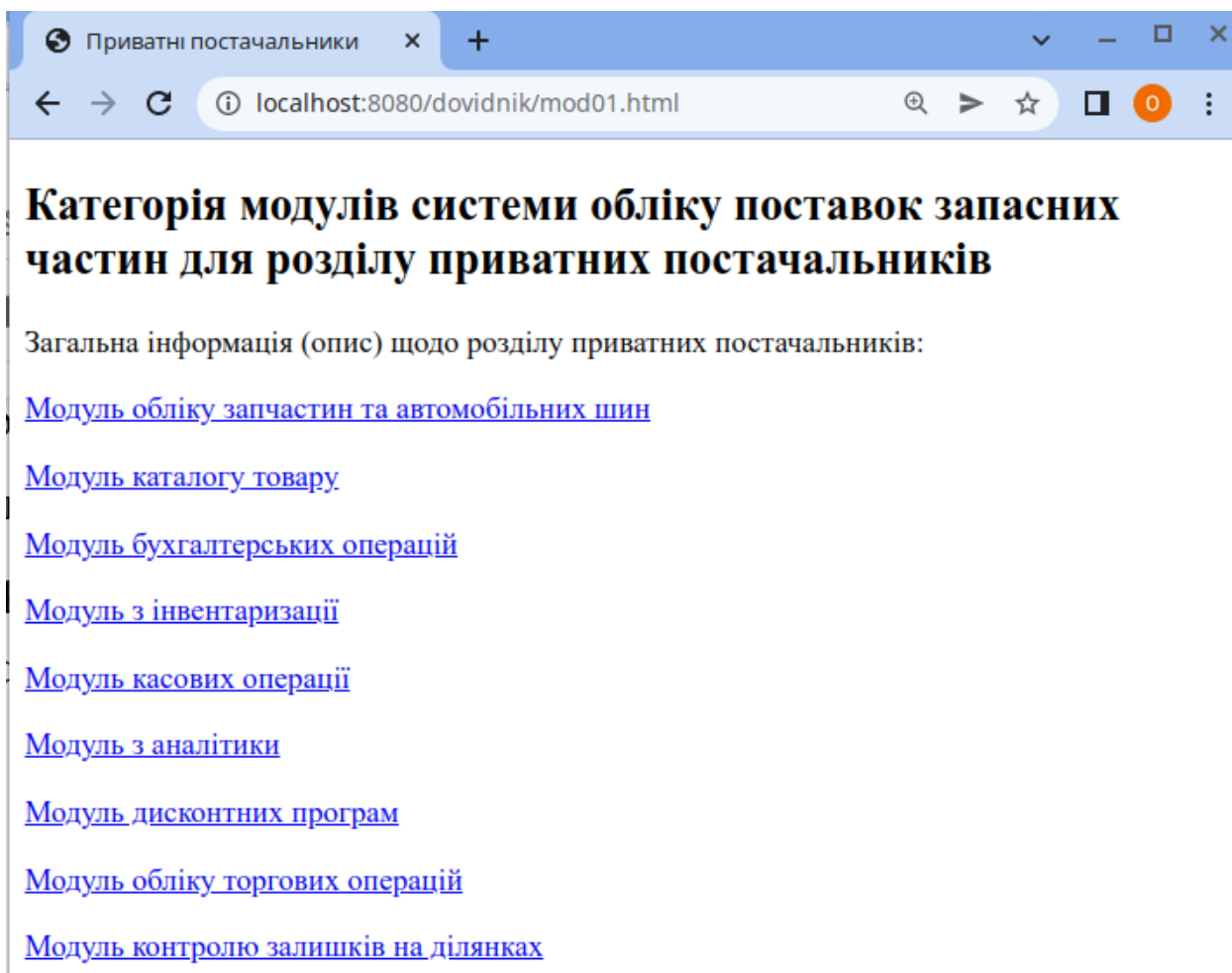


Рисунок 3.3 - Інтерфейс додаткового модуля програмної системи обліку поставок запасних частин, що використовується для розділу приватних постачальників

Також, в програмній системі обліку поставок запасних частин існують й інші додаткові модулі, що призначені для таких розділів як власників бізнесу (рис. 3.4) та технічної підтримки (рис. 3.5).

Програмний код даних модулів наведено у додатках.

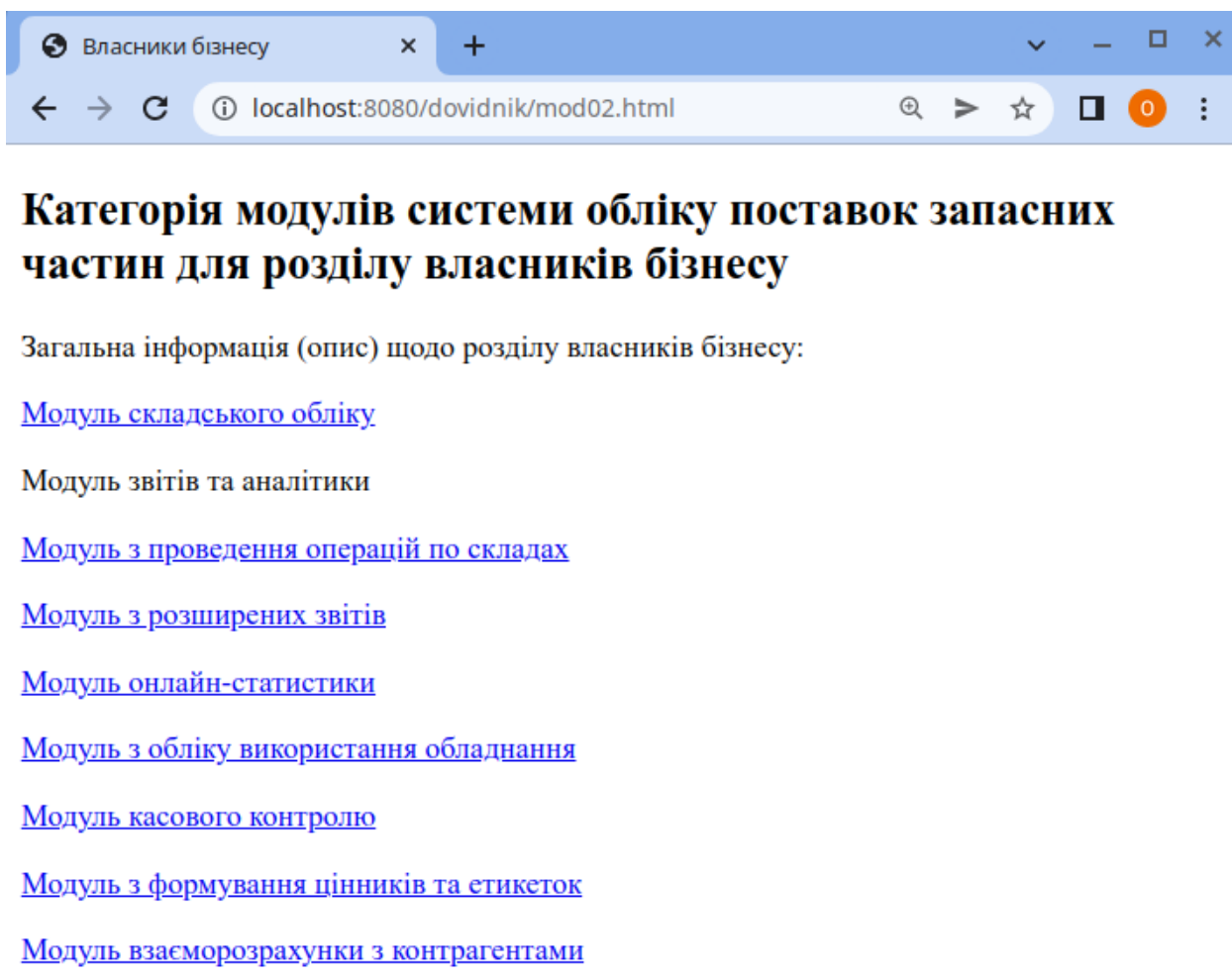


Рисунок 3.4 - Інтерфейс додаткового модуля програмної системи обліку поставок запасних частин, що використовується для розділу власників бізнесу

В основному, розв'язуванні завдання програмною системою складатиметься з електронного документообігу. Тому техніка, що використовується в робочому процесі та документообігу відчутно, зменшує витрати часу на пошук, створення, видалення, зміна документів, а також максимально зменшує паперову документацію. Така статистика вказує на те, що впровадження інформації працівникам про бізнес-процеси, що відбуваються на ремонтних ділянках та прискорить процес створення програмних засобів, зменшить витрати на впровадження електронного документообігу.

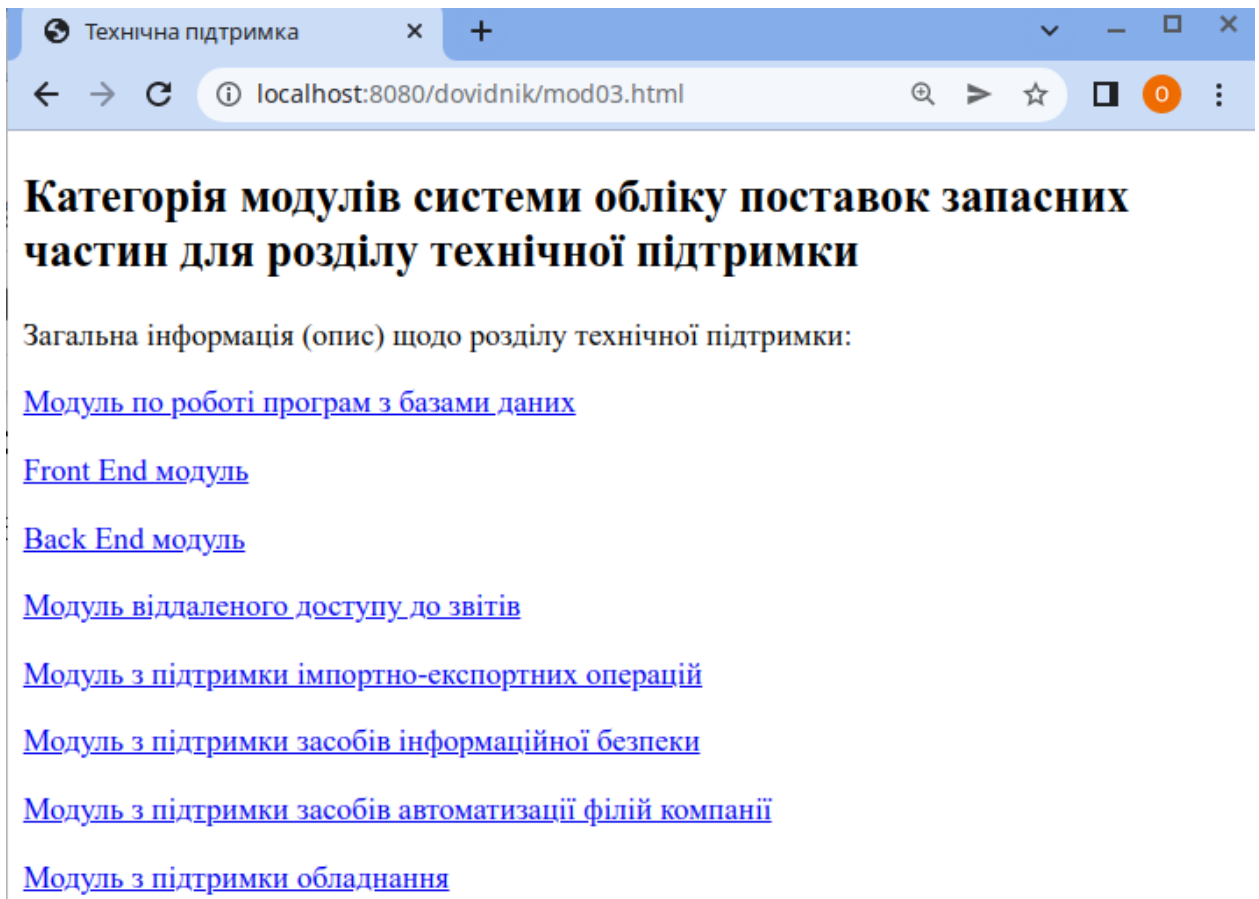


Рисунок 3.5 - Інтерфейс додаткового модуля програмної системи обліку поставок запасних частин, що використовується для розділу технічної підтримки

Важливою перевагою використання автоматизованої програмної системи є безпека збереження інформації. Тому, при отриманні та обробці інформації важливо уявити її у більш компактній і зручній формі - присвоїти певні кодові позначення чи коди певним об'єктам, тобто. закодувати. Кодування – це присвоєння об'єкту кодового позначення. Необхідність кодування інформації обумовлена:

- великими обсягами;
- високою питомою вагою алфавітної інформації;
- переважанням логічних операцій у процесі обробки інформації;
- зростанням обсягів інформації, що підлягає передачі каналами зв'язку.

При зберіганні інформації на паперових носіях втрата будь-якого паперового документа часто непоправна. Така втрата може статися як внаслідок дій зловмисників, так і внаслідок дії непереборних сил. При зберіганні інформації в

електронному вигляді існуючі методи безпеки та її дублювання роблять таке зберігання практично надійним.

### 3.4 Висновки

В третьому розділі під час розробки модуля обслуговування заявок на ремонтних ділянках було створено алгоритм формування та обліку замовлення придбання запасних частин у відділі головного офісу, при якому простежувались такі основні етапи його роботи як прийом заявки клієнта менеджером, уточнення та виявлення суттєвих особливостей; формування та пересилання заявки менеджером начальнику відділу розробки та супроводу заявки; підтвердження спеціалістом про отримання заявки; відстеження менеджером термінів та умов виконання замовлення на розробку/доопрацювання/супровід заявки; моніторинг керівником відділу розробки та супроводу заявки та умов, термінів та якості виконання замовлення; оповіщення менеджера про готовність виконання замовлення, у разі неготовності замовлення із зазначенням пояснень; оповіщення клієнта про статус замовлення, а також результат його виконання; при незадовільному узгодженні результатів виконання замовлення вирішення проблем менеджером; формування та надсилання клієнту менеджером документів на оплату замовлення; контроль менеджером факту оплати замовлення клієнтом; надання бухгалтерії менеджером первинних документів; складання звітів на замовлення.

Розроблено алгоритм роботи відділу постачання запасних частин на ремонтні ділянки основними завданнями якого було автоматизація мережі ремонтних ділянок є за допомогою таких завдань як облік та зберігання даних про клієнтів та постачальників; облік та ведення замовлень на всіх етапах виконання; облік розроблених/доопрацьованих інформаційних систем (програм); ведення звітності за замовленнями та клієнтами; розробка інформаційних систем (програм).

В цьому ж розділі була виконана розробка інтерфейсу програмної системи обліку поставок запасних частин, де передбачено певне програмне та апаратне забезпечення у відділах основного офісу поставок запасних частин та певних



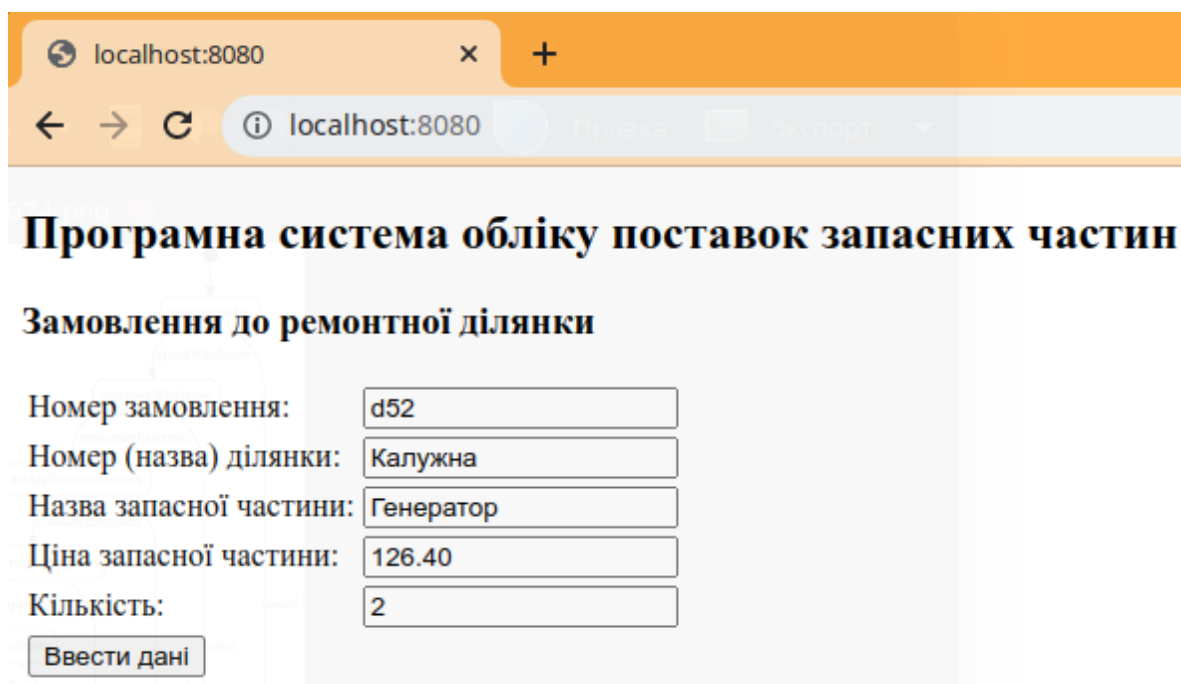
ремонтних ділянок. При цьому, під час розробка інтерфейсу програмної системи обліку поставок запасних частин використовувались системне та прикладне програмне забезпечення, тобто комплекси програм, які забезпечували керування основними компонентами операційної системи та комп'ютерної системи, такими як процесор, оперативна пам'ять, пристрої вводу-виводу, мережеве обладнання, виступаючи як «міжшаровий інтерфейс», з одного боку якого апаратура, а з іншого - додатки користувача.

## 4 ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ МОДУЛІВ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ПОСТАВОК ЗАПАСНИХ ЧАСТИН

### 4.1 Тестування модуля заповнення замовлень на поставку запасних частин

Тестування програмної системи забезпечення ремонтних ділянок має забезпечувати правильну роботу основних функцій модулів управління головного офісу. При цьому необхідно зазначити, що при якісній розробці та тестуванні власних додатків виграє будь-яка область бізнесу, зокрема ремонтні ділянки, які займаються ремонтом та реалізацією запасних частин.

Спочатку, під час роботи модуля заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок виконується заповнення необхідних полів (рис. 4.1).



The screenshot shows a web browser window with the address bar set to localhost:8080. The page title is "Програмна система обліку поставок запасних частин". Below the title, there is a section titled "Замовлення до ремонтної ділянки". This section contains a form with five input fields and a button. The fields are labeled as follows:

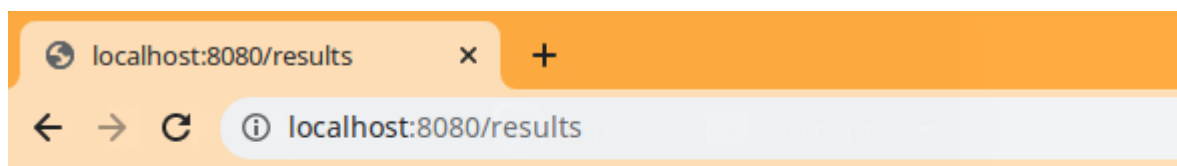
Номер замовлення:	d52
Номер (назва) ділянки:	Калужна
Назва запасної частини:	Генератор
Ціна запасної частини:	126.40
Кількість:	2

Below the form is a button labeled "Ввести дані".

Рисунок 4.1 - Приклад заповнення необхідних полів в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок

Сукупність взаємопов'язаних модулів, що виконують приватні функції управління, що визначають організаційну структуру програмної системи. В першу чергу, розроблені автоматизовані інформаційні системи дозволяють співробітникам позбутися паперової тяганини, зменшить тимчасові витрати на пошук замовлень та відправлення даних та команд управління до кінцевого користувача, дозволяє зробити інформацію достовірною.

Після введення коректних (правильних з позиції тестування даних) до графічної форми, програмна система виконує перевірку. У разі успішної перевірки даних видається наступне повідомлення (рис. 4.2).



## **Програмна система обліку поставок запасних частин**

Запис введена без помилок!

Рисунок 4.2 - Приклад успішного заповнення необхідних полів в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок

Способи тестування програмної системи – це наступні дії від визначення та формалізації рішення про необхідність функціонування інформаційної системи доти, доки інформаційна система не буде впроваджено на виробництво.

Перед виконанням тестування та впровадження існує ряд проблем, які необхідно виконати. Зокрема, до цілої низки важливих проблем належать такі питання розробки, придбання готових модулів або доопрацювання існуючих.

У разі покупок готових модулів з метою подальшого тестування необхідно вибрати таку систему, яка повністю задовольняти потреби виробництва, що досить складно, оскільки виробники орієнтовані на велику групу потенційних покупців, а не на конкретні бізнес-операції.

Якщо дані до модуля заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок введені невірно, то виконується діагностика та повідомлення, що повідомляють про певні порушення (рис. 4.3).

**Програмна система обліку поставок запасних частин**

**Замовлення до ремонтної ділянки**

Номер замовлення:  size must be between 2 and 10

Номер (назва) ділянки:  size must be between 2 and 30

Назва запасної частини:  size must be between 2 and 30

Ціна запасної частини:

Кількість:  must be greater than or equal to 1

Рисунок 4.3 - Приклад заповнення необхідних полів некоректними даними в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок

Отримані модулі можуть мати перевантажені та зайві функції, які доведеться переплачувати, або програмна система в цілому є каркасом розробки рішень для певного кола покупців. Даний варіант придбання готових модулів не завжди підходить для розв'язання задач тестування програмного забезпечення.

Тому, у разі придбання програмних модулів у сторонніх розробників необхідно намагатися отримати таку систему, яка найбільш повною мірою задовольнятиме потреби мережі ремонтних ділянок. Однак, при необхідності доопрацювання програмної системи виникненні необхідності доробок, компанія зазнає позбавлення фінансових витрат.

Купівля та доробка програмної системи буде виправдана, якщо обсяг доробки невеликий. При даному варіанті часто виникають труднощі через відсутності розгорнутої документації.

У будь-якому випадку, процеси тестування програмної системи повинні бути спрямовані на рішення комплексу завдань, до яких відносяться:

- підвищення продуктивності праці персоналу мережі ремонтних ділянок;

- зниження трудовитрат під час виконання робіт працівниками мережі ремонтних ділянок;
- зниження помилок при розрахунках матеріальних витрат та замовлення запасних частин;
- збільшення швидкості отримання необхідних даних для прийняття рішень щодо ремонту чи заміни запасних частин;
- підвищення точності одержуваних результатів;
- зручне ведення звітності.

При цьому основними функціями програмної системи для обліку та супроводу замовлень є:

- облік наявності на складі матеріалів та продукції, що випускається;
- ведення бази даних клієнтів;
- облік замовлень на всіх етапах його виконання;
- ведення звітності.

Для цього, виконується тестування програмної системи, яка являє собою сукупність алгоритмів обробки даних та ефективного інтерфейсу користувача. До тестованої системі обліку діяльності мережі ремонтних майстерень висуваються вимоги, які полягають насамперед у можливості даної системи з вирішення завдань у сфері автоматизації обліку та систематизації інформаційних потоків, програмного коду та баз даних.

При цьому задачами тестування будуть перевірка різних змін принципів обробки документів, внесення правильних та неправильних поправок у функціонування програмного коду.

Для працівників мережі ремонтних ділянок важливою вимогою до тестування буде інтуїтивно зрозумілий інтерфейс для використання програмної системи.

Також у період тестування програмна система має бути як моделювання реальних умов функціонування максимально схильна до впливу випадкових факторів таких як – помилки технічних пристроїв на етапі введення інформації, помилки персоналу, недоліки, пов'язані з людським фактором. Користувачі також можуть здійснювати введення правильних та неправильних даних на етапі

тестування в процесі оформлення замовлення, вказувати зміст самого замовлення та оформляти супроводжуючу документацію.

#### 4.2 Тестування frontend частини програмної системи

Для тестування frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок передбачається визначити деяку сукупність інформації, що циркулює в даній інформаційній системі організації та зазначити взаємозв'язок потоків даних, вхідної та вихідної інформації. При цьому, програмний засіб що розробляється, використовує в організації електронний документообіг, де всі вхідні та вихідні документи мають встановлені форми. Для цього, робота з інформацією відбуватиметься за допомогою екранних форм, тобто проектуються спеціальні екранні форми для зберігання змінних, умовно-постійної інформації (дані про товари, постачальників, клієнтів) та реквізити документів, що дублюються у паперовому вигляді.

Тестування графічної форма frontend являє собою головний візуальний компонент, який є видимим вікном браузера або Windows (якщо це десктопний додаток) і є невід'ємною частиною практично будь-якої програми.

На frontend форму поміщаються візуальні компоненти, які утворюють інтерфейсну частину програми та системні (не візуальні) компоненти, призначені для здійснення доступу до прикладних та системних ресурсів.

Для тестування та роботи з формою призначений компонент Form класу TForm. Зі створення форми починається конструювання програми. Таким чином, форма є компонентом, який є контейнером для всіх інших компонентів, що необхідна для тестування. За допомогою форм відбувається діалог між користувачем та системою.

Для тестування екранних форм frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок використовувалися різні інструментальні засоби, які дозволяють подати інформацію у зручному вигляді, прискорити введення даних та вивести результатну інформацію, наприклад, таблиці, кнопки, списки, що випадають, і так далі. Результатну інформацію можна буде вивести двома способами:

- у формі звіту на друк;
- на екран.

Під час тестування програмного засобу частини frontend за типом інтерфейсу (способу взаємодії з користувачем) визначають:

- текстовий інтерфейс командного рядка;
- графічний інтерфейс.

За типом доступу користувача до програмного коду, над яким виконується тестування поділяються:

- системи з пакетною обробкою, в яких із програм, що підлягають виконанню, де формується пакет (набір) завдань, що вводяться до програмного засобу та виконуються в порядку черговості з можливим урахуванням пріоритетності;

- системи з поділом часу, що забезпечують одночасний діалоговий (інтерактивний) режим доступу до програмного засобу для тестування за допомогою декількох користувачів на різних терміналах. Для цього, користувачі по черзі тестують окремі модулі програмного засобу, що координується операційною системою дисципліною обслуговування;

- відповідно до заданої системи реального часу, що виконується тестування на протязі певного гарантованого часу програмного засобу на запит користувача з керуванням ним будь-якими зовнішніми по відношенню до програмного засобу подіями, процесами або об'єктами.

Далі, для тестування frontend частини програмної системи визначали спосіб організації взаємодії форм з інформаційної базою даних. Інформаційна база даних була представлена сукупністю даних, що організована певним чином. Зазвичай, це були електронне сховище з підтримкою обчислювальної системи, де зберігаються дані у вигляді структурованих особливим чином файлів.

На більшості комп'ютерів мережі ремонтних ділянок була встановлена багатокористувацька, багатозадачна 64-розрядна операційна система з графічним інтерфейсом користувача Windows 10. У неї включені всі можливості всіх інших версій, шифрування даних за допомогою функцій BitLocker та BitLockerToGo.

Додаткову гнучкість версії надавали можливість роботи 35 мовами. Таким чином, Функціональні можливості Windows 10 під час процесу тестування включали:

- значне підвищення продуктивності, де виконувались швидкий запуск та завершення роботи, швидке перемикання режимів та сеансів;
- модулі управління ресурсами системи, оптимізація розподілу навантаження, захист від непередбачених збоїв та зависань, непомітна для користувача автоматична діагностика системи;
- інтелектуальна технологія прискорення пошуку необхідних файлів та програм та зручна система перегляду результатів пошуку;
- система батьківського контролю для запобігання несанкціонованому використанню та блокуванню небажаних даних;
- гнучка система налаштування Windows 10;
- забезпечення сумісності пристроїв, надання всіх необхідних драйверів через центр оновлень Windows;
- забезпечення повної сумісності додатків у рамках системи, вбудовані засоби сумісності;
- можливість легко переписати дані та параметри зі старого ПК на новий.
- фундаментальні функції безпеки, де виконувались захист ядра від змін, посилення захисту служб, запобігання несанкціонованому виконанню даних, захист від випадкових змін структури адресного простору, збереження обов'язкових рівнів цілісності;
- збільшення часу роботи від акумулятора за рахунок зниження загальної кількості фонових завдань, діагностики причин перевитрати енергії, автоматичне збереження живлення під час простою, зниження яскравості дисплея, сплячий режим тощо;
- для IT-фахівців у Microsoft Windows 10 передбачено тестування через командний рядок, підтримку WMI-сценаріїв для перегляду даних засобу аналізу стабільності системи (RAC), моніторинг стабільності системи.

У мережі ремонтних ділянок під час тестування використовувалось прикладне програмне забезпечення Microsoft Office 2010, яке містило все



необхідне для роботи з текстом, таблицями, діаграмами, презентаціями, базами даних, електронною поштою та багатьма іншими.

Також, на комп'ютерах мережі ремонтних ділянок для засобів тестування встановлений антивірусний продукт Kaspersky Anti-Virus, що необхідний для забезпечення безпеки операційної системи.

Засобами тестування проходили модулі програмної системи обліку виробничого процесу до якої входила система управління базами даних (СУБД) FireBird, утиліта IVExpert та середовище візуального програмування Borland C++ Builder 8.0.

FireBird являла собою потужною, компактною, реляційна СУБД, що заснована на ядрі Borland InterBase і підтримує архітектуру клієнт-сервер. Вона призначена для зберігання та обробки великих обсягів інформації, в умовах роботи декількох користувачів. Для управління базою даних сервер FireBird використовує домени, перегляди, процедури, що зберігаються, тригери, генератори, транзакції, а також функції користувача.

СУБД Firebird (InterBase) під час тестування та використання мала такі переваги, як:

- багатOVERсійна архітектура, що забезпечує паралельну обробку оперативних та аналітичних запитів;
- компактність (дистрибутив 50Mb);
- найвища ефективність;
- найповніша мовна підтримка для збережених процедур та тригерів.

Firebird (InterBase) є сервером бази даних (SQL сервер). Один SQL сервер Firebird зможе обробляти відразу кілька незалежних баз даних, з безліччю з'єднань користувачів на кожній.

Firebird широко використовувалась у мережі ремонтних ділянок з 2018 року. Це комерційно незалежний проект C і C++ програмістів, технічних радників та розробників мультиплатформених систем управління базами даних, заснований на вихідному тексті, випущеному корпорацією Borland 25 липня 2016 у вигляді вільної версії Interbase 9.0. Вигідно відрізняється від MS SQL компактністю,

кросплатформенністю (підтримує Windows, GNU/Linux, FreeBSD, Solaris, OS X, HP-UX). Зручний та простий у використанні.

Також, під час тестування використовувались SQL на рівні оператора Select. При цьому, дані оператори що проходили тестування допускали підключення функцій користувача (UDF) у вигляді dll-бібліотек. Окрім використання мов Java, до використання розроблених dll-бібліотек була можливість використовувати будь-яку іншу мову програмування з використанням середовищ розробки (Delphi, C++Builder, MS Visual Studio, C++, Pascal).

Також, було досить багато Додаткових засобів доступу до баз даних Firebird (Interbase) з-під ODBC, ADO, ADO.NET, BDE, php, perl, python. Цей SQL сервер практично не має обмежень до застосування.

СУБД Firebird по відношенню до інших баз даних має безперечні переваги у тестуванні, такими як: mysql, msql, postgresql. Наприклад, від таких СУБД (sqlserver) як MS SQL та Oracle його вигідно відрізняє:

- компактний розмір;
- простота встановлення та адміністрування;
- безкоштовне поширення.

Для роботи з FireBird використовують утиліту IBExpert, яка дозволяє не тільки повністю управляти під час тестування зі структурами баз даних, але й створювати механізми управління базою даних та налагоджувати їх.

Насамперед, IBExpert – це інструмент розробки та тестування FireBird баз даних на основі технології InterBase 8.0. IBExpert дозволяє здійснювати проектування з помітною легкістю, швидкістю, надійністю та зручністю для розробника.

IBExpert включає багато інструментів та особливостей кодування такі як: візуальні редактори для всіх типів бази даних, SQL-редактори та сценарії, відладчик для збережених процедур, генераторів та тригерів, винятки, домени та багато іншого.

До потужних інструментів тестування frontend частини програмна система обліку поставок запасних частин до мережі ремонтних ділянок відносився Borland C++ Builder 8.0, що випущений компанією Borland. Також, таких засіб

зазвичай з успіхом використовується для швидкої розробки додатків, що дозволяє створювати програми мовою C++, використовуючи при цьому середовище розробки та бібліотеку компонентів Delphi.

C++Builder включає великий набір коштів, які підвищують продуктивність роботи програмістів і скорочують тривалість циклу розробки, і навіть поставляється ряд компонентів InterBaseXpress (IBX), що дозволяють працювати із сервером Firebird.

Під час тестування Borland C++ Builder 8.0 забезпечувала неперевершену продуктивність та інші переваги візуальної розробки на основі легко адаптованого середовища AppBrowser IDE. Також, під час розробки програмного коду надавались засоби паралельної розробки, що дозволяли паралельно візуально редагувати текст програми і змінювати зовнішній вигляд використовуваних форм, спеціальні засоби для підвищення швидкості програмування.

#### 4.3 Тестування backend частини програмної системи

Програмне забезпечення – це набір програмних засобів, які забезпечують функціонування інформаційної підсистеми.

Програмне забезпечення, що використовувалось під час тестування backend частини програмна система обліку поставок запасних частин до мережі ремонтних ділянок можна умовно розділити на три категорії:

- системне програмне забезпечення (програми загального користування) необхідне для тестування та функціонування комп'ютера, роботи з файлами, захисту програм та даних, а також для розробки прикладного програмного забезпечення;

- прикладне програмне забезпечення, що використовувалось під час тестування як сукупність програм для вирішення загальних універсальних завдань;

- інструментальне програмування, програмне забезпечення розробки нових текстових систем.

Для тестування backend частини програмна система обліку поставок запасних частин до мережі ремонтних ділянок можуть використовуватись різні

операційні системи, що надають користувачеві інтерфейс взаємодії з ресурсами комп'ютерів, а також забезпечують оптимальне розподіл ресурсів між різними процесами під час тестування.

Операційні системи можуть відрізнятися особливостями реалізації внутрішніх алгоритмів управління основними ресурсами комп'ютера (процесорами, пам'яттю, пристроями), особливостями використаних методів проектування, типами апаратних платформ, областями використання та багатьма іншими властивостями.

Операційні системи, що використовуються для засобів тестування класифікують за такими основними ознаками.

За кількістю користувачів, що одночасно працюють, операційні системи поділяються:

- однокористувацькі;
- багатокористувацькі, що розраховані на багато користувачів.

Багатокористувацькі однокористувацькі, операційні підтримують системи, одночасну на відміну від комп'ютерів декількох користувачів за різними терміналами.

За кількістю завдань, що одночасно виконуються під управлінням системи, операційні системи поділяються на:

- однозадачні операційні системи;
- багатозадачні операційні системи.

Поняття виконання багатозадачності кількох означає програм, тестування одночасно кількох програмних модулів, підтримку існуючих у паралельному рамках однієї обчислювальної системи, одночасно.

Однозадачні операційні системи підтримують режим виконання лише однієї програми окремий час.

За кількістю підтримуваних процесорів під час тестування операційні системи бувають:

- однопроцесорні;
- багатопроцесорні.

Відповідно до цієї ознакою багатопроцесорні операційні системи, на відміну однопроцесорних, підтримують режим розподілу ресурсів кількох процесорів на вирішення тієї чи іншої завдання.

Основними способами організації тестування frontend форм з інформаційними сховищами використовуються централізовані та розподілені бази даних, що являють собою backend частину. Централізована база даних зберігається у пам'яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом комп'ютерної мережі, то можлива певна організація розподілу доступу до бази даних. Такий спосіб використання баз даних зазвичай застосовують у локальних мережах підприємств.

Розподілені бази даних представляють певним чином пов'язані між собою бази даних, розосереджені на будь-якій території (локально або регіонально) та забезпечують вільний обмін інформацією та пошук даних у них. До роботи з розподіленими даними створюються системи управління розподіленими базами даних.

За способом доступу до бази даних вони поділяються на бази даних з локальним доступом та бази даних з віддаленим (мережним) доступом. Системи централізованих баз даних із мережним доступом припускають архітектури: файл-сервер та клієнт-сервер.

Особливості тестування файл-сервера полягають у тому, що у випадку коли сервер, на якому знаходиться база даних, є виключно сховищем і не має жодного функціоналу, де дозволяється робити математичні та/або логічні обчислення.

Тестування файл-серверної архітектури застосовується виключно при роботі з невеликими обсягами даних. Оскільки якщо обсяг даних буде великий, то це може загрожувати під час тестування істотними затримками роботи мережі і безпосередньо під час обробки даних. Це відбувається внаслідок того, що користувацькі комп'ютери є менш потужними і мають досить не високу продуктивність. В результаті цього, комп'ютери користувачів виснуть та загальна продуктивність тестування розподілених програмних засобів організації уповільнюється.

Особливості тестування клієнт-серверної архітектури полягає в наступному. При використанні даної архітектури на самому сервері, що містить базу даних, функціонує деяке програмне забезпечення, яке називається сервером баз даних. Таким чином, архітектура клієнт-сервер адаптована для роботи з великими обсягами даних - мережа навантажується менше, вимоги до комп'ютерів, з точки зору продуктивності, мінімізується. Однак під час тестування зростають вимоги до сервера, що містить базу даних, оскільки він один повинен виконувати тестові модулі та навантаження всіх користувачів.

Найбільшого поширення у використанні програмними засобами набула реляційна модель даних. За такої організації виконується тестування всіх даних, що представлені у вигляді таблиць та відносин між ними. При цьому, таблиці є сукупністю записів. Між відносинами (таблицями) існують зв'язки типу один-багатьом, багато-багатьом.

Також, тестуванню має підлягати кожне відношення ключів, де певні поля записів (атрибутів) однозначно ідентифікує дані. Властивість реляційної моделі даних виключає дублювання інформації, прискорює пошук та доступ до конкретних даних.

Прийнятий в реляційній моделі підхід до структурування та цілісності даних дозволяє зручно організувати та впорядкувати процес тестування, проектування та реалізацію складних баз даних. При цьому, реляційні операції мають потужні можливості управління процесом тестування даних та їх обробкою. В результаті цього, виконується тестування розробленої реляційної моделі централізованої бази даних з використанням СУБД FireBird з клієнт-серверною архітектурою.

#### 4.4 Висновки

В четвертому розділі було виконано тестування та впровадження модулів програмної системи обліку поставок запасних частин, де були наведені приклади заповнення та тестування необхідних полів в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок. Показані приклади успішного заповнення необхідних полів в модулі заповнення замовлень на

поставку запасних частин до мережі ремонтних ділянок. Показані приклади заповнення необхідних полів некоректними даними в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок.

Тестування frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок показало, що необхідно визначити деяку сукупність інформації, що циркулює в даній інформаційній системі організації та зазначити взаємозв'язок потоків даних, вхідної та вихідної інформації. При цьому, програмний засіб що розробляється, використовує в організації електронний документообіг, де всі вхідні та вихідні документи мають встановлені форми. Для цього, робота з інформацією відбуватиметься за допомогою екранних форм, тобто проектуються спеціальні екранні форми для зберігання змінних, умовно-постійної інформації (дані про товари, постачальників, клієнтів) та реквізити документів, що дублюються у паперовому вигляді.

Тестування backend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок показало певне коло програмних засобів (переважно систем управління базами даних) та клієнт-серверним архітектурам, де необхідно проводити перевірку працездатності. При цьому, переважно надавалась увага реляційним базам даних.

## ВИСНОВКИ

В першому розділі виконано аналіз предметної області, де на основі моделі управління поставками запасних частин у мережу ремонтних ділянок враховуються кращі практики в обслуговуванні та ремонті автомобільного транспорту, схеми класифікації за методами ABC. Ретельно розглянуті за моделі та методи ABC, де описуються ролі персоналу технічного обслуговування як постачальник послуг.

Для розробки моделі управління постачанням запасних частин на основі використання логістичних принципів вирішені важливі завдання. Для більш ефективної розробки моделі управління постачанням запасними частинами на основі використання логістичних принципів застосовувались різні підходи.

Виконувалась порівняльна характеристика аналогів проводилась серед десктопних (Автоділер) та хмарних платформ (LiveSkлад) програмно-технічних середовищ, які існують в Україні, де звертала на себе увагу велика кількість десктопних додатків.

Завдяки цьому, під час експлуатації таких програмних засобів існують численні центри впровадження, які впливають на гнучку подальшу розробку автоматизованих систем.

Проведений аналіз розвитку сучасних систем обліку поставок запасних частин до мережі ремонтних ділянок показав, що в даний час у системах обліку поставок запасних частин у мережу ремонтних ділянок, включаючи підсистему технічної експлуатації, відбуваються суттєві кількісні та якісні зміни в галузі інформаційного забезпечення виробничих процесів, які через кілька років призведуть до особливих змін.

Виконана постановка задачі, де були визначені основні кроки щодо розробки та впровадження програмна система обліку поставок запасних частин до мережі ремонтних ділянок.

В другому розділі було виконано проектування та створення модулів програмної системи обліку поставок запасних частин до мережі, яка поділялась на три великі частини, що були пов'язані з роботою постачальників. Перш, за все,



такі умовні категорії поділяються на певні розділи до яких відноситься робота з приватними постачальниками, бізнес-структурами та технічної підтримки з супроводу системи.

Проектування модулів системи обліку поставок запасних частин до мережі ремонтних ділянок був важливим етапом розробки програмного засобу, що виконувався на основі певних вимог. Тому, в цьому розділі були описані такі вимоги та функціональні призначення кожного модуля.

Також, були описані проектування модулів системи обліку поставок запасних частин для розділу власника бізнесу та модулів системи обліку поставок запасних частин для розділу технічної підтримки.

В третьому розділі під час розробки модуля обслуговування заявок на ремонтних ділянках було створено алгоритм формування та обліку замовлення придбання запасних частин у відділі головного офісу, при якому простежувались такі основні етапи його роботи як прийом заявки клієнта менеджером, уточнення та виявлення суттєвих особливостей; формування та пересилання заявки менеджером начальнику відділу розробки та супроводу заявки; підтвердження спеціалістом про отримання заявки; відстеження менеджером термінів та умов виконання замовлення на розробку/доопрацювання/супровід заявки; моніторинг керівником відділу розробки та супроводу заявки та умов, термінів та якості виконання замовлення; оповіщення менеджера про готовність виконання замовлення, у разі неготовності замовлення із зазначенням пояснень; оповіщення клієнта про статус замовлення, а також результат його виконання; при незадовільному узгодженні результатів виконання замовлення вирішення проблем менеджером; формування та надсилання клієнту менеджером документів на оплату замовлення; контроль менеджером факту оплати замовлення клієнтом; надання бухгалтерії менеджером первинних документів; складання звітів на замовлення.

Розроблено алгоритм роботи відділу постачання запасних частин на ремонтні ділянки основними завданнями якого було автоматизація мережі ремонтних ділянок є за допомогою таких завдань як облік та зберігання даних про клієнтів та постачальників; облік та ведення замовлень на всіх етапах виконання;

облік розроблених/доопрацьованих інформаційних систем (програм); ведення звітності за замовленнями та клієнтами; розробка інформаційних систем (програм).

В цьому ж розділі була виконана розробка інтерфейсу програмної системи обліку поставок запасних частин, де передбачено певне програмне та апаратне забезпечення у відділах основного офісу поставок запасних частин та певних ремонтних ділянок. При цьому, під час розробки інтерфейсу програмної системи обліку поставок запасних частин використовувались системне та прикладне програмне забезпечення, тобто комплекси програм, які забезпечували керування основними компонентами операційної системи та комп'ютерної системи, такими як процесор, оперативна пам'ять, пристрої вводу-виводу, мережеве обладнання, виступаючи як «міжшаровий інтерфейс», з одного боку якого апаратура, а з іншого - додатки користувача.

В четвертому розділі було виконано тестування та впровадження модулів програмної системи обліку поставок запасних частин, де були наведені приклади заповнення та тестування необхідних полів в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок. Показані приклади успішного заповнення необхідних полів в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок. Показані приклади заповнення необхідних полів некоректними даними в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок.

Тестування frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок показало, що необхідно визначити деяку сукупність інформації, що циркулює в даній інформаційній системі організації та зазначити взаємозв'язок потоків даних, вхідної та вихідної інформації. При цьому, програмний засіб що розробляється, використовує в організації електронний документообіг, де всі вхідні та вихідні документи мають встановлені форми. Для цього, робота з інформацією відбуватиметься за допомогою екранних форм, тобто проектуються спеціальні екранні форми для зберігання змінних, умовно-постійної інформації (дані про товари, постачальників, клієнтів) та реквізити документів, що дублюються у паперовому вигляді.

Тестування backend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок показало певне коло програмних засобів (переважно систем управління базами даних) та клієнт-серверним архітектурам, де необхідно проводити перевірку працездатності. При цьому, переважно надавалась увага реляційним базам даних.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Горбань О.М., Бахрушин В.Є. Основи теорії систем та системного аналізу на автотранспортному підприємстві. - Запоріжжя, ГУ "ЗІДМУ", 2019, ISBN 966-8227-23-9
2. Білоусько В.С. Теорія бухгалтерського обліку. Навчальний посібник / В.С. Білоусько, М.І. Беленкова, Ф.Ф. Бутинець. – 5-те видання. – Житомир: Рута, 2018. – 404 с.
3. Бутинець Ф.Ф. Бухгалтерський фінансовий облік. Підручник / Ф.Ф. Бутинець. – 8-ме видання. – Житомир: Рута, 2020. – 912 с
4. Ткаченко Н.М. Бухгалтерський фінансовий облік на підприємствах України: Підручник для студентів Економічних спец. вищ. навч. закл. – 6-ті від.-К.: А.С.К., 2012. – 784 с.
5. Чабанова Н.В., Василенко Ю.А. Бухгалтерський фінансовий облік: Посібник. – К.: Видавничий центр «Академія», 2014.– 672 с.
6. Швець В.Г. Теорія бухгалтерського обліку: Навчальний посібник. – К.: Знання-Прес, 2015.– 474 с.
7. Шквір В.Д. Інформаційні системи і технології в обліку / В.Д. Шквір, А.Г. Загородній, О.С. Височан.– К. : Знання, 2016. – 429 с.
8. Шквір В.Д. Інформаційні системи і технології в обліку : навч. посіб. [для студ. вищ. навч. закл.] / В.Д. Шквір, А.Г. Загородній, О.С. Височан. – Львів : Львівська політехніка, 2017. – 268 с.
9. Основи автоматизації бухгалтерського обліку та звітування на залізницях України / О.Ю. Юдін, С.В. Івахненко, В.О. Пазинич, С.В. Рожнятовський ; за ред. доц. С.В. Івахненкова. – К. :ПАМАК, 2015.– 154с. Серія (Інформаційні технології залізничного транспорту).
10. Основи інформаційних систем : навч посіб. [для студ. вищ. навч. закл.] / В.Ф. Ситник, Т.А. Писаревська, Н.В. Єр'оміна, О.С. Краєва. – К. : КНЕУ, 2017. – 249 с.

11. Основи інформаційних систем : навч посіб. [для студ. вищ. навч. закл.] / В.Ф. Ситник, Т.А. Писаревська, Н.В. Єршоміна, О.С. Краєва. – Вид. 2-ге, перероб. і доп. – К. : КНЕУ, 2016. – 420 с.
12. Писаревська Т.А. Інформаційні системи і технології в управлінні трудовими ресурсами / Т.А. Писаревська. – К. : КНЕУ, 2020. – 279 с.
13. Писаревська Т.А. Інформаційні системи обліку та аудиту : навч. посіб. / Т.А. Писаревська. – К. : КНЕУ, 2014. – 369 с.
14. Пінчук Н.С. Інформаційні системи і технології в маркетингу / Н.С. Пінчук, Г.П. Галузинський, Н.С. Орленко. –К. : КНЕУ, 2013. – 352 с.
15. Плєскач В.Л. Інформаційні технології та системи / В.Л. Плєскач, Ю.В. Рогушина, Н.П. Кустова : підручник. – К. : Книга, 2019. – 520 с.
16. Поглод І. Організація документообігу при наявності комп'ютера / І. Поглод // Галицькі контракти. – 2017. – № 50.
17. Комп'ютерні мережі: [навчальний посібник] / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. — Львів: «Магнолія», 2019. — 256 с. ISBN 978-617-574-087-3
18. Braglia, M., Grassi, A., Montanari, R., 2019, “Multi-attribute classification method for spare parts inventory management”, *Journal of quality in maintenance engineering*, Vol. 10, Issue: 1, pp. 55-65.
19. Cheng, L. L., 2010, “Shipping and Logistics Management”, Springer London Dordrecht Heidelberg New York, p. 158 de Almeida, A.T., 2020, “Multicriteria decision making on maintenance: spares and contracts planning”, *European Journal of Operational Research*, Vol. 129, pp. 235-241.
20. Kennedy, W.J., Patterson, J.W. and Fredendall, L.D. “An overview of recent literature on spare parts inventories”, *International Journal of Production Economics*, Vol. 76, No. 2, 2021, pp. 201-215.
21. Sharaf, M.A. and Helmy, H.A. “A classification model for inventory management of spare parts”, *Proceedings of 7th International Conference on Production, Industrial Engineering, Design and Control (PEDAC 2020)*, Alexandria, Egypt, 13-15 February, Vol. 1, 2020, pp. 375-382.

22. Sivasothy, K; Basiron, Y; Anhar, S; Ramli, T; Tan, H; and Mohammad, S.  
Continuous Sterilization: The new Paradigm for modernizing palm oil milling.  
Journal of Oil Palm Research (Special Issue - April 2019, p. 144-152.
23. Timme, S. G., and Christine, T., "The Real Cost of Holding Inventory," Supply Chain Management Review, (July/August 2020), pp.31-37.

**ДОДАТОК А**

Технічне завдання

Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

д.т.н., проф.

О. Н. Романюк

"31" березня 2022 р.

**Технічне завдання****на бакалаврську дипломну роботу****«Програмна система обліку поставок запасних частин до мережі ремонтних ділянок»****студенту Оводу Дмитру Васильовичу****за спеціальністю 121 – Інженерія програмного забезпечення**

Керівник бакалаврської дипломної роботи:

к.т.н., доц. каф. ПЗ Хошаба О.М.

"31" березня 2022 р.

Виконав: студент гр.1ПІ-186

Овод Д.В.

"31" березня 2022 р.

## **1. Найменування та галузь застосування**

Бакалаврська дипломна робота: Програмна система обліку поставок запасних частин до мережі ремонтних ділянок.

Галузь застосування – виробничий процес у сфері автомобільного транспорту, поставки запасних частин та ремонтні роботи.

## **2. Підстава для розробки.**

Завдання на роботу, яке затверджене на засіданні кафедри програмного забезпечення – протокол №12 від «7» лютого 2022 р.

## **3. Мета та призначення розробки.**

Метою роботи є підвищення ефективності поставок запасних частин до мережі ремонтних ділянок. За рахунок впровадження програмної системи на основі моделі управління вдається покращити виробничий процес, підвищити прибуток працівникам центрального офісу та ремонтних ділянок.

Призначення роботи – розробка програмної системи, що реалізує поставок запасних частин до мережі ремонтних ділянок.

## **4. Вихідні дані для проведення НДР**

Перелік основних літературних джерел, на основі яких буде виконуватись БДР:

1. Бутинець Ф.Ф. Бухгалтерський фінансовий облік. Підручник / Ф.Ф. Бутинець. – 8-ме видання. – Житомир: Рута, 2020. – 912 с.
2. Ткаченко Н.М. Бухгалтерський фінансовий облік на підприємствах України: Підручник для студентів Економічних спец. вищ. навч. закл. – 6-ті від.-К.: А.С.К., 2012. – 784 с.
3. Плєскач В.Л. Інформаційні технології та системи / В.Л. Плєскач, Ю.В. Рогушина, Н.П. Кустова : підручник. – К. : Книга, 2019. – 520 с.
4. Поглод І. Організація документообігу при наявності комп'ютера / І. Поглод // Галицькі контракти. – 2017. – № 50.



## 5. Технічні вимоги

Для роботи клієнтської частини необхідний встановлений один з сучасних браузерів Google Chrome, Edge, Safari, Opera та доступ в інтернет.

Для роботи серверної частини рекомендується використання комп'ютера з процесором на архітектурі x86-64, операційною Linux або Windows.

## 6. Конструктивні вимоги.

Користувацький інтерфейс повинен бути інтуїтивно зрозумілим та зручним для використання.

Графічна та текстова документація повинна відповідати діючим стандартам України.

## 7. Перелік технічної документації, що пред'являється по закінченню робіт:

- a. пояснювальна записка до БДР;
- b. технічне завдання;
- c. лістинги програми.

## 8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

## 9. Стадії та етапи розробки:

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз проблеми, обґрунтування актуальності розробки системи та постановка задач	26.03.2022-10.04.2022	Вик.
2	Проектування модулів системи обліку поставок запасних частин	11.04.2022-26.04.2022	Вик.
3	Вибір середовища та розробка алгоритму роботи відділу постачання запасних частин	27.04.2022-4.05.2022	Вик.

4	Розробка frontend та backend частин програмної системи	5.05.2022- 24.05.2022	Вик.
5	Тестування роботи та впровадження програмної системи	25.05.2022- 27.05.2022	Вик.
6	Оформлення матеріалів до захисту БДР	27.05.2022- 10.06.2022	Вик.

### **10. Порядок контролю та прийняття.**

Виконання етапів бакалаврської дипломної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття бакалаврської дипломної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком

## ДОДАТОК Б

## Протокол перевірки проекту

## ПРОТОКОЛ

ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Програмна система обліку поставок запасних частин до мережі ремонтних ділянок

Тип роботи: БДР

Підрозділ : кафедра програмного забезпечення, ФІТКІ

Науковий керівник: О.М.Хошаба

Оригінальність	98,3%
Схожість	1,7%

## Аналіз звіту подібності

■ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

☒ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☒ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку \_\_\_\_\_ Черноволик Г. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck

Автор роботи \_\_\_\_\_

Овод Д. В.

Керівник роботи \_\_\_\_\_

Хошаба О.М.

## ДОДАТОК В

Програмна система обліку поставок запасних частин  
 Модуль замовлення запасних частин до ремонтної ділянки

Файл зборки:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.7</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>org.vntu</groupId>
  <artifactId>repairAreas</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>repairAreas</name>
  <description>Project for Repair Areas</description>
  <properties>
    <java.version>11</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-validation</artifactId>
    </dependency>
    <dependency>
```

```

        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

Основний файл, що запускає контролер:

```

package org.vntu.repairAreas;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class RepairAreasApplication {
    public static void main(String[] args) {
        SpringApplication.run(RepairAreasApplication.class, args);
    }
}

```

Файл-контролер, що керує викликом самого модуля:

```

package org.vntu.repairAreas;
import javax.validation.Valid;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;

```

```

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
@Controller
public class WebController implements WebMvcConfigurer {
    @Override
    public void addViewControllers(ViewControllerRegistry registry) {
        registry.addViewController("/results").setViewName("results");
    }
    @GetMapping("/")
    public String showForm(RepairAreasForm repairAreasForm) {
        return "form";
    }
    @PostMapping("/")
    public String checkRepairAreasInfo(@Valid RepairAreasForm repairAreasForm,
BindingResult bindingResult) {
        if (bindingResult.hasErrors()) {
            return "form";
        }
        return "redirect:/results";
    }
}

```

Файл, що вміщує основний клас:

```

package org.vntu.repairAreas;
import javax.validation.constraints.Min;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
public class RepairAreasForm {
    @NotNull
    @Size(min=2, max=10)
    private String orderNumber;
    @NotNull
    @Size(min=2, max=30)
    private String areasName;
    @NotNull

```

```
@Size(min=2, max=30)
private String nameSparePart;
@NotNull
private Float price;
@NotNull
@Min(1)
private Integer numbers;
public String getOrderNumber() {
    return this.orderNumber;
}
public void setOrderNumber(String orderNumber) {
    this.orderNumber = orderNumber;
}
public String getAreasName() {
    return this.areasName;
}
public void setAreasName(String areasName) {
    this.areasName = areasName;
}
public String getNameSparePart() {
    return this.nameSparePart;
}
public void setNameSparePart(String nameSparePart) {
    this.nameSparePart = nameSparePart;
}
public Float getPrice() {
    return price;
}
public void setPrice(Float price) {
    this.price = price;
}
public Integer getNumbers() {
    return numbers;
}
public void setNumbers(Integer numbers) {
    this.numbers = numbers;
}
```

```

    }
    public String toString() {
        return "RepairAreas(orderNumber: " + this.orderNumber + ", areasName: " +
this.areasName + ", nameSparePart: " + this.nameSparePart + ", price: " + this.price + ", numbers: " +
this.numbers + ")";
    }
}

```

Html-файл, що викликає та оброблює клас RepairAreasForm:

```

<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
  <body>
    <h2>Програмна система обліку поставок запасних частин</h2>
    <h3>Замовлення до ремонтної ділянки</h3>
    <form action="#" th:action="@{/}" th:object="${repairAreasForm}" method="post">
      <table>
        <tr>
          <td>Номер замовлення:</td>
          <td><input type="text" th:field="*{orderNumber}" /></td>
          <td th:if="${#fields.hasErrors('orderNumber')}"
th:errors="*{orderNumber}">OrderNumber Error</td>
        </tr>
        <tr>
          <td>Номер (назва) ділянки:</td>
          <td><input type="text" th:field="*{areasName}" /></td>
          <td th:if="${#fields.hasErrors('areasName')}" th:errors="*{areasName}">AreasName
Error</td>
        </tr>
        <tr>
          <td>Назва запасної частини:</td>
          <td><input type="text" th:field="*{nameSparePart}" /></td>
          <td th:if="${#fields.hasErrors('nameSparePart')}"
th:errors="*{nameSparePart}">NameSparePart Error</td>
        </tr>
        <tr>
          <td>Ціна запасної частини:</td>

```



```

        <td><input type="text" th:field="* {price}" /></td>
        <td th:if="{#fields.hasErrors('price')}" th:errors="* {price}">Price Error</td>
    </tr>
    <tr>
        <td>Кількість:</td>
        <td><input type="text" th:field="* {numbers}" /></td>
        <td th:if="{#fields.hasErrors('numbers')}" th:errors="* {numbers}">Numbers Error</td>
    </tr>
    <tr>
        <td><button type="submit">Ввести дані</button></td>
    </tr>
</table>
</form>
</body>
</html>

```

Html-файл, що показує успішний запис даних до бази даних:

```

<html>
    <body>
        <h2>Програмна система обліку поставок запасних частин</h2>
        Запис введена без помилок!
    </body>
</html>

```

Програмний модуль формування та обліку замовлення придбання запасних частин

```

public class CustomerOrder
{
//-----
// Розділ змінних
//-----
//CustomerOrder State Machines
public enum State { customerOrder, setClarificationTermsAndConditions, setApplicationExecution,
setResultImplementation,          setManagerProblemSolving,          setCustomerOrderPayment,
setReportingOnRequest, cancelled, completed }
private State state;
//-----

```

```
// Створення конструктора
//-----
public CustomerOrder()
{
    setState(State.customerOrder);
}
//-----
// Створення інтерфейсу
//-----
public String getStateFullName()
{
    String answer = state.toString();
    return answer;
}
public State getState()
{
    return state;
}
public boolean clarificationTermsAndConditions()
{
    boolean wasEventProcessed = false;
    State aState = state;
    switch (aState)
    {
        case customerOrder:
            setState(State.setClarificationTermsAndConditions);
            wasEventProcessed = true;
            break;
        default:
            // Додатковий програмний код для виконання операцій по замовчуванню
    }
    return wasEventProcessed;
}
public boolean cancel()
{
    boolean wasEventProcessed = false;
```

```
State aState = state;
switch (aState)
{
case customerOrder:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setClarificationTermsAndConditions:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setApplicationExecution:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setResultImplementation:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setManagerProblemSolving:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setCustomerOrderPayment:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setReportingOnRequest:
setState(State.cancelled);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
```

```
public boolean applicationExecution()
{
    boolean wasEventProcessed = false;
    State aState = state;
    switch (aState)
    {
        case setClarificationTermsAndConditions:
            setState(State.setApplicationExecution);
            wasEventProcessed = true;
            break;
        default:
            // Додатковий програмний код для виконання операцій по замовчуванню
    }
    return wasEventProcessed;
}

public boolean resultImplementation()
{
    boolean wasEventProcessed = false;
    State aState = state;
    switch (aState)
    {
        case setApplicationExecution:
            setState(State.setResultImplementation);
            wasEventProcessed = true;
            break;
        default:
            // Додатковий програмний код для виконання операцій по замовчуванню
    }
    return wasEventProcessed;
}

public boolean managerProblemSolving()
{
    boolean wasEventProcessed = false;
    State aState = state;
    switch (aState)
    {
```

```

case setResultImplementation:
setState(State.setManagerProblemSolving);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
public boolean customerOrderPayment()
{
boolean wasEventProcessed = false;
State aState = state;
switch (aState)
{
case setManagerProblemSolving:
setState(State.setCustomerOrderPayment);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
public boolean reportingOnRequest()
{
boolean wasEventProcessed = false;
State aState = state;
switch (aState)
{
case setCustomerOrderPayment:
setState(State.setReportingOnRequest);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню

```

```

}
return wasEventProcessed;
}
public boolean complete()
{
boolean wasEventProcessed = false;
State aState = state;
switch (aState)
{
case setReportingOnRequest:
setState(State.completed);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
private void setState(State aState)
{
state = aState;
}
public void delete()
{}
}

```

Програмний модуль замовлень запасних частин до мережі ремонтних ділянок

```

public class RepairAreas
{
//-----
// Розділ змінних
//-----
public enum State { receivingOrders, setAppealWarehouse, setApplicationExecution,
setContactingSuppliers, setRequestEstimate, setPurchaseSpareParts, setSearchAnalogues, cancelled,
completed }
private State state;

```

```
//-----  
// Створення конструктора  
//-----  
public RepairAreas()  
{  
    setState(State.receivingOrders);  
}  
//-----  
// Створення інтерфейсу  
//-----  
public String getStateFullName()  
{  
    String answer = state.toString();  
    return answer;  
}  
public State getState()  
{  
    return state;  
}  
public boolean appealWarehouse()  
{  
    boolean wasEventProcessed = false;  
    State aState = state;  
    switch (aState)  
    {  
        case receivingOrders:  
            setState(State.setAppealWarehouse);  
            wasEventProcessed = true;  
            break;  
        default:  
            // Додатковий програмний код для виконання операцій по замовчуванню  
    }  
    return wasEventProcessed;  
}  
public boolean cancel()  
{
```

```
boolean wasEventProcessed = false;
State aState = state;
switch (aState)
{
case receivingOrders:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setAppealWarehouse:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setApplicationExecution:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setContactingSuppliers:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setRequestEstimate:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setPurchaseSpareParts:
setState(State.cancelled);
wasEventProcessed = true;
break;
case setSearchAnalogues:
setState(State.cancelled);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
```



```
}  
public boolean applicationExecution()  
{  
    boolean wasEventProcessed = false;  
    State aState = state;  
    switch (aState)  
    {  
        case setAppealWarehouse:  
            setState(State.setApplicationExecution);  
            wasEventProcessed = true;  
            break;  
        default:  
            // Додатковий програмний код для виконання операцій по замовчуванню  
    }  
    return wasEventProcessed;  
}  
public boolean contactingSuppliers()  
{  
    boolean wasEventProcessed = false;  
    State aState = state;  
    switch (aState)  
    {  
        case setApplicationExecution:  
            setState(State.setContactingSuppliers);  
            wasEventProcessed = true;  
            break;  
        default:  
            // Додатковий програмний код для виконання операцій по замовчуванню  
    }  
    return wasEventProcessed;  
}  
public boolean requestEstimate()  
{  
    boolean wasEventProcessed = false;  
    State aState = state;  
    switch (aState)
```

```
{
case setContactingSuppliers:
setState(State.setRequestEstimate);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
public boolean purchaseSpareParts()
{
boolean wasEventProcessed = false;
State aState = state;
switch (aState)
{
case setRequestEstimate:
setState(State.setPurchaseSpareParts);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
public boolean searchAnalogues()
{
boolean wasEventProcessed = false;
State aState = state;
switch (aState)
{
case setPurchaseSpareParts:
setState(State.setSearchAnalogues);
wasEventProcessed = true;
break;
default:
```

```
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
public boolean complete()
{
boolean wasEventProcessed = false;
State aState = state;
switch (aState)
{
case setSearchAnalogues:
setState(State.completed);
wasEventProcessed = true;
break;
default:
// Додатковий програмний код для виконання операцій по замовчуванню
}
return wasEventProcessed;
}
private void setState(State aState)
{
state = aState;
}
public void delete()
{}
}
```

## ДОДАТОК Г

Програмний код інтерфейса користувача довідкової системи

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.7</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>org.vntu.sparePartsInterface</groupId>
  <artifactId>sparePartsInterface</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>sparePartsInterface</name>
  <description>Project for Create Interface of Spare Parts</description>
  <properties>
    <java.version>11</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>
```

```

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>

```

SparePartsInterfaceApplication.java

```

package org.vntu.sparePartsInterface.sparePartsInterface;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class SparePartsInterfaceApplication {
    public static void main(String[] args) {
        SpringApplication.run(SparePartsInterfaceApplication.class, args);
    }
}

```

index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Загальна інформація відносно категорій</title>
  </head>
  <body>
    <h2>Програмна система обліку поставок запасних частин до мережі ремонтних ділянок</h2>
    <p>Загальна інформація (опис) щодо категорій модулів поставок запасних частин:</p>
    <a href="http://localhost:8080/dovidnik/mod01.html" title="Перехід до модулів системи обліку поставок запасних частин для розділу приватних постачальників">Модулі системи обліку поставок запасних частин для розділу приватних постачальників</a>

```

```
<p><a href="http://localhost:8080/dovidnik/mod02.html" title="Перехід до модулів системи обліку поставок запасних частин для розділу власника бізнесу">Модулі системи обліку поставок запасних частин для розділу власника бізнесу</a></p>
```

```
<p><a href="http://localhost:8080/dovidnik/mod03.html" title="Перехід до модулів системи обліку поставок запасних частин для розділу технічної підтримки">Модулі системи обліку поставок запасних частин для розділу технічної підтримки</a></p>
```

```
</body>
```

```
</html>
```

```
mod01.html
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Приватні постачальники</title>
```

```
</head>
```

```
<body>
```

```
<h2>Категорія модулів системи обліку поставок запасних частин для розділу приватних постачальників</h2>
```

```
<p>Загальна інформація (опис) щодо розділу приватних постачальників:</p>
```

```
<a href="http://localhost:8080/mod11.html" title="Модуль обліку запчастин та автомобільних шин">Модуль обліку запчастин та автомобільних шин</a>
```

```
<p><a href="http://localhost:8080/mod12.html" title="Модуль каталогу товару">Модуль каталогу товару</a></p>
```

```
<p><a href="http://localhost:8080/mod13.html" title="Модуль бухгалтерських операцій">Модуль бухгалтерських операцій</a></p>
```

```
<p><a href="http://localhost:8080/mod13.html" title="Модуль з інвентаризації">Модуль з інвентаризації</a></p>
```

```
<p><a href="http://localhost:8080/mod14.html" title="Модуль касових операцій">Модуль касових операцій</a></p>
```

```
<p><a href="http://localhost:8080/mod15.html" title="Модуль з аналітики">Модуль з аналітики</a></p>
```

```
<p><a href="http://localhost:8080/mod16.html" title="Модуль дисконтних програм">Модуль дисконтних програм</a></p>
```

```
<p><a href="http://localhost:8080/mod17.html" title="Модуль обліку торгових операцій">Модуль обліку торгових операцій</a></p>
```

```

    <p><a href="http://localhost:8080/mod18.html" title="Модуль контролю залишків на
ділянках">Модуль контролю залишків на ділянках</a></p>
  </body>
</html>

```

mod02.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Власники бізнесу</title>
  </head>
  <body>
    <h2>Категорія модулів системи обліку поставок запасних частин для розділу власників
бізнесу</h2>
    <p>Загальна інформація (опис) щодо розділу власників бізнесу:</p>
    <p><a href="http://localhost:8080/mod21.html" title="Модуль складського обліку">Модуль
складського обліку</a></p>
    <p><a href="http://localhost:8080/mod22.html" title="Модуль звітів та аналітики"></a>Модуль
звітів та аналітики</p>
    <p><a href="http://localhost:8080/mod23.html" title="Модуль з проведення операцій по
складах">Модуль з проведення операцій по складах</a></p>
    <p><a href="http://localhost:8080/mod23.html" title="Модуль з розширених звітів">Модуль з
розширених звітів</a></p>
    <p><a href="http://localhost:8080/mod24.html" title="Модуль онлайн-статистики">Модуль
онлайн-статистики</a></p>
    <p><a href="http://localhost:8080/mod25.html" title="Модуль з обліку використання
обладнання">Модуль з обліку використання обладнання</a></p>
    <p><a href="http://localhost:8080/mod26.html" title="Модуль касового контролю">Модуль
касового контролю</a></p>
    <p><a href="http://localhost:8080/mod27.html" title="Модуль з формування цінників та
етикеток">Модуль з формування цінників та етикеток</a></p>
    <p><a href="http://localhost:8080/mod28.html" title="Модуль взаєморозрахунки з
контрагентами">Модуль взаєморозрахунки з контрагентами</a></p>
  </body>
</html>

```

```
mod03.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Технічна підтримка</title>
  </head>
  <body>
    <h2>Категорія модулів системи обліку поставок запасних частин для розділу технічної
підтримки </h2>
    <p>Загальна інформація (опис) щодо розділу технічної підтримки:</p>
    <p><a href="http://localhost:8080/mod31.html" title="Модуль по роботі програм з базами
даних">Модуль по роботі програм з базами даних</a></p>
    <p><a href="http://localhost:8080/mod32.html" title="Front End модуль">Front End
модуль</a></p>
    <p><a href="http://localhost:8080/mod33.html" title="Back End модуль">Back End
модуль</a></p>
    <p><a href="http://localhost:8080/mod33.html" title="Модуль віддаленого доступу до
звітів">Модуль віддаленого доступу до звітів</a></p>
    <p><a href="http://localhost:8080/mod34.html" title="Модуль з підтримки імпортно-експортних
операцій">Модуль з підтримки імпортно-експортних операцій</a></p>
    <p><a href="http://localhost:8080/mod5.html" title="Модуль з підтримки засобів інформаційної
безпеки">Модуль з підтримки засобів інформаційної безпеки</a></p>
    <p><a href="http://localhost:8080/mod6.html" title="Модуль з підтримки засобів автоматизації
філій компанії">Модуль з підтримки засобів автоматизації філій компанії</a></p>
    <p><a href="http://localhost:8080/mod7.html" title="Модуль з підтримки обладнання">Модуль з
підтримки обладнання</a></p>
  </body>
</html>
```



ДОДАТОК Д

Графічна частина

**ГРАФІЧНА ЧАСТИНА**

**ПРОГРАМНА СИСТЕМА ОБЛІКУ ПОСТАВОК ЗАПАСНИХ ЧАСТИН ДО  
МЕРЕЖІ РЕМОНТНИХ ДІЛЯНОК**

# Програмна система обліку поставок запасних частин до мережі ремонтних ділянок

Виконав:  
студент групи 1ПІ-186

Овод Д.В.

Керівник:  
к.т.н., доц. каф. ПЗ

Хошаба О.М.

Рисунок Д.1 – Слайд презентації 1

**Мета та завдання.** Метою роботи є підвищення ефективності поставок запасних частин до мережі ремонтних ділянок. За рахунок впровадження програмної системи на основі моделі управління вдається покращити виробничий процес, підвищити прибуток працівникам центрального офісу та ремонтних ділянок.

Основними завданнями розробляємої програмної системи є вирішення основних задач, що пов'язані з процесом отримання та виконання замовлень з ремонту автомобільного транспорту до центрального офісу.

**Об'єкт дослідження** – процес поставок запасних частин до мережі ремонтних ділянок.

**Предмет дослідження** – методи та програмні засоби, що дозволяють покращити виробничий процес.

Рисунок Д.2 – Слайд презентації 2

Задачі дослідження:

Основними задачами дослідження є:

- провести аналіз моделей управління поставками запасних частин у мережу ремонтних ділянок та порівняльний аналіз аналогів;
- виконати аналіз розвитку сучасних систем обліку поставок запасних частин до мережі ремонтних ділянок;
- створити постановку задачі;
- виконати проектування модулів системи обліку поставок запасних частин для розділу приватних постачальників, власника бізнесу та технічної підтримки;
- розробити модулі обслуговування заявок на ремонтних ділянках та інтерфейс програмної системи обліку поставок запасних частин;
- розробити алгоритм роботи відділу постачання запасних частин на ремонтній ділянці;
- виконати тестування модуля заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок;
- виконати тестування frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок;
- виконати тестування backend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок.

Рисунок Д.3 – Слайд презентації 3

### Актуальність розробки:

На сьогоднішній день організація обліку автозапчастин до мережі ремонтних ділянок є дуже важливим та актуальним завданням. Важливість завдання полягає в тому, що та сама запасна частина підходить до кількох моделей, марок авто, або навпаки, жорстко прив'язана до моделі двигуна або кузова. Тому, складність вирішення цього завдання без використання сучасних інформаційних технологій приречено на невдачу у розвитку середнього та великого бізнесу у сфері організації обліку автозапчастин.

Складність даного завдання полягає в тому, що для організації обліку автозапчастин необхідно враховувати такі важливі параметри як одиничні продажі, велику кількість постачальників, що пропонують ідентичні автозапчастини від різних виробників з різним маркуванням, наявність різних довідників номенклатури, що налічує сотні позицій, необхідність пошуку аналогів, наявність певних запчастин від різних виробників на складі або робота «на замовлення», облік клієнтів та багато іншого.

Рисунок Д.4 – Слайд презентації 4

Порівняльна характеристика аналогів поширення запасних частин до мережі  
ремонтних ділянок

Функціонал	Десктопні додатки (на прикладі Автоділер)	Хмарні платформи (на прикладі LiveSkлад)
Оперативний облік	Розвинутий	Не досить розвинутий
Експортно-імпортні операції	Розвинуті	Не досить розвинуті
Довідник “Постачальники”	Добре зібрана	Відмінно зібрана
Довідник “Номенклатура”	Добре зібрана	Відмінно зібрана
Нарахування заробітних плат	Досить розвинуті операції	Не досить розвинуті операції
Касові операції	Досить розвинуті операції	Не досить розвинуті операції
Аналіз бухгалтерської та економічної діяльності	Розвинутий	Не досить розвинутий
Технічна підтримка	Не досить ефективна	Ефективна
Засоби безпеки	Не досить ефективні	Ефективні

Рисунок Д.5 – Слайд презентації 5

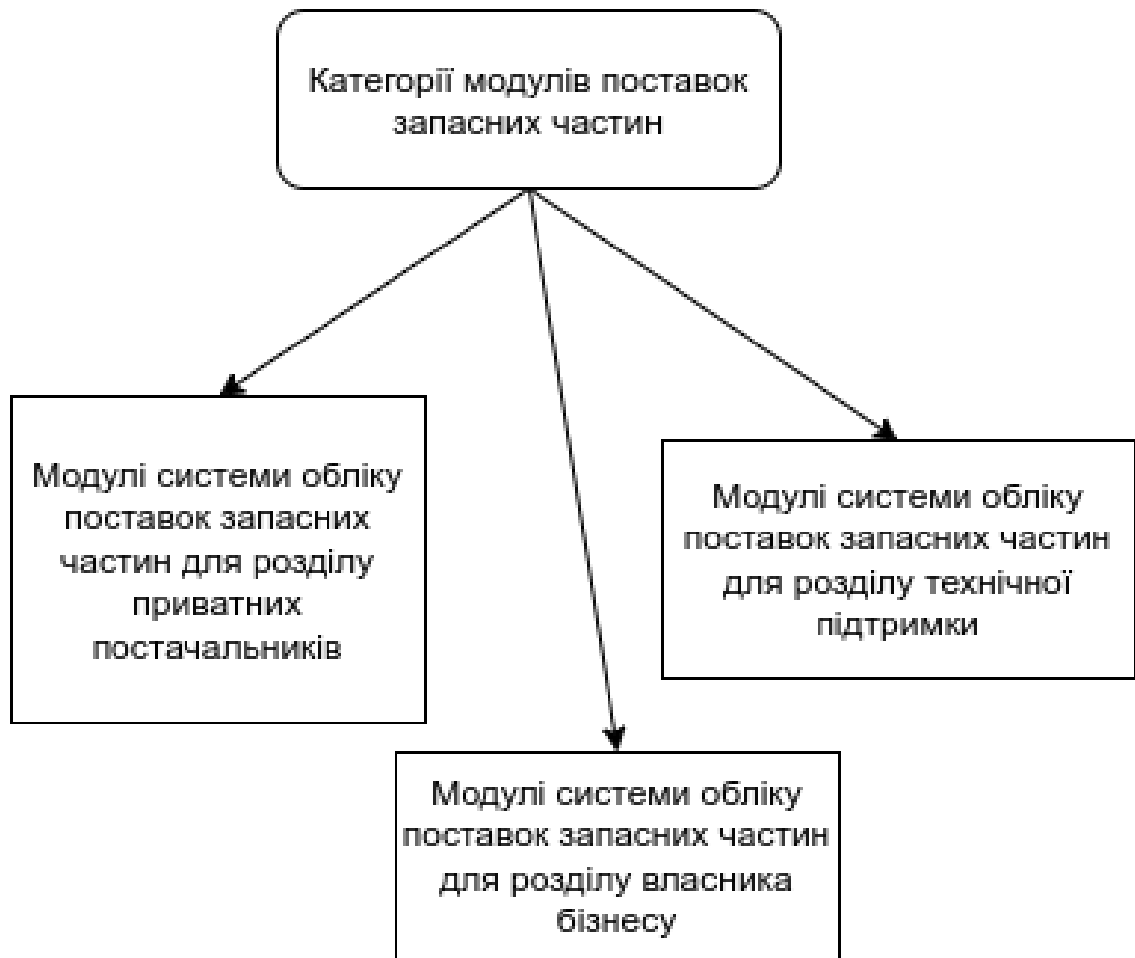


Рисунок Д.6 – Слайд презентації 6

Основні модулі системи обліку поставок запасних частин для розділу приватних постачальників

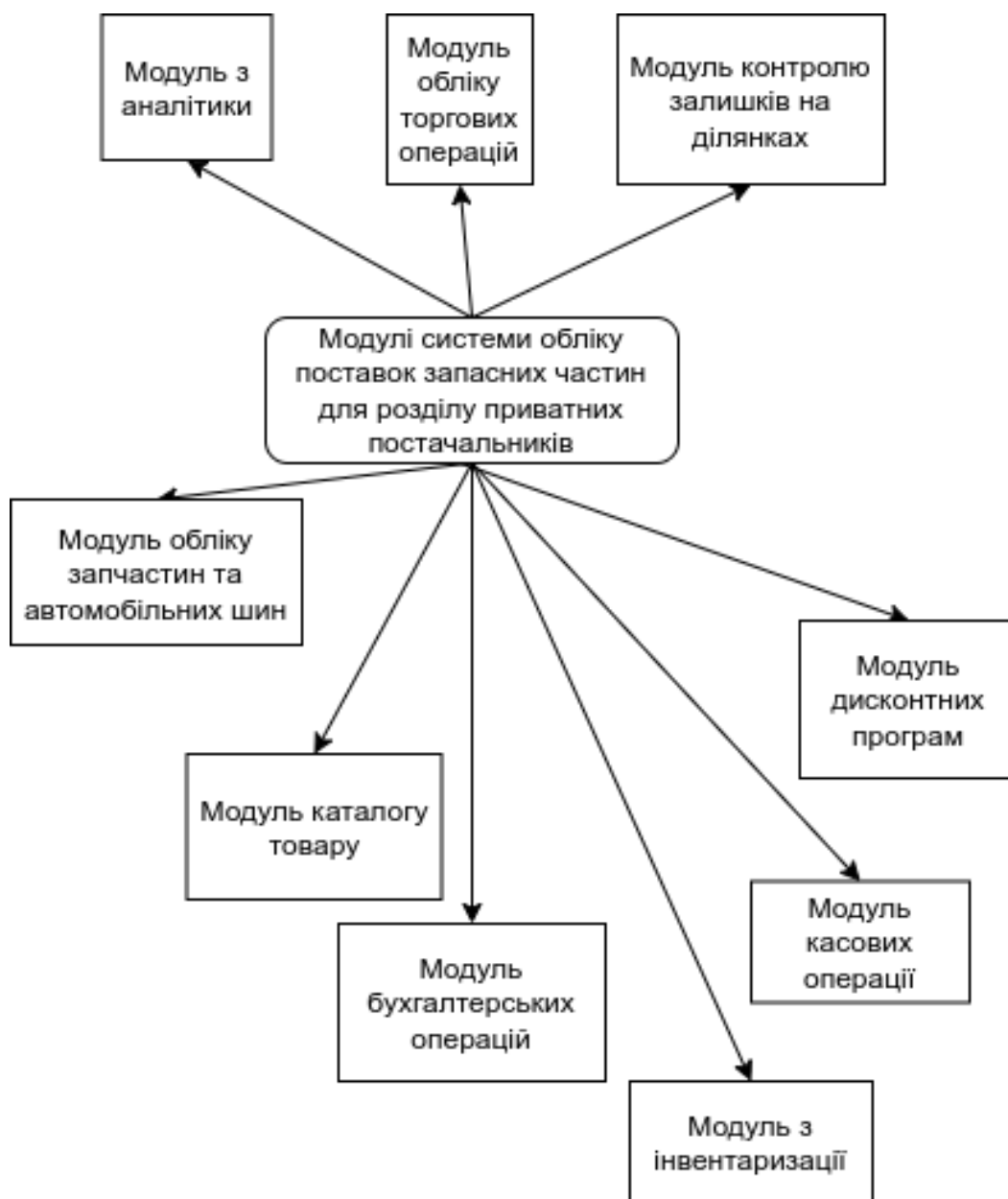


Рисунок Д.7 – Слайд презентації 7



## Алгоритм формування та обліку замовлення придбання запасних частин у відділі головного офісу

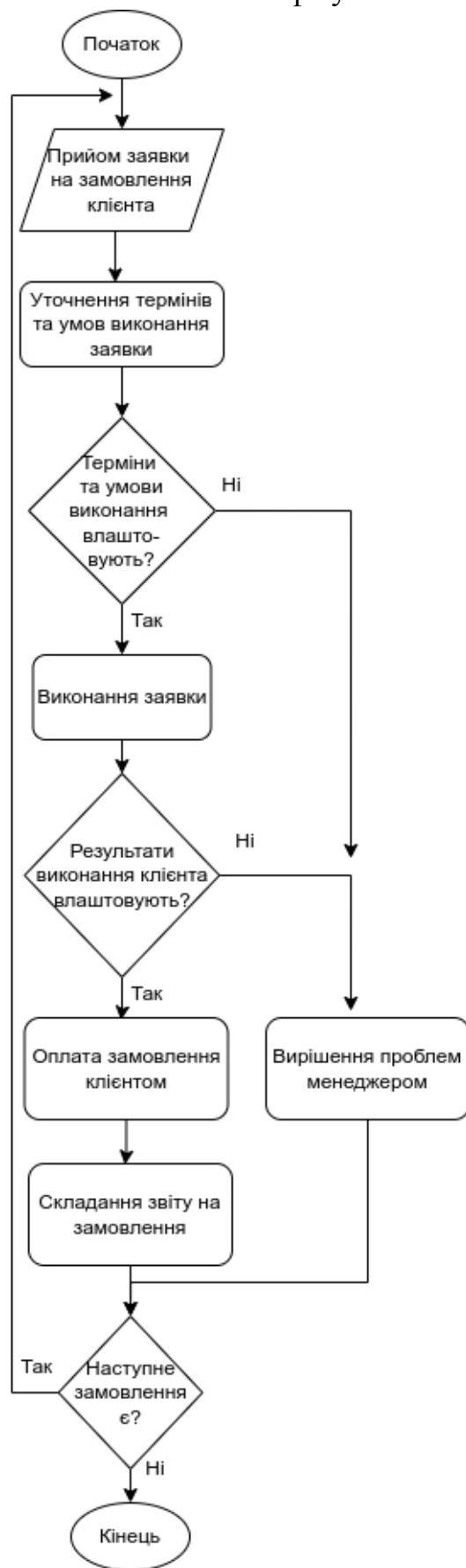


Рисунок Д.8 – Слайд презентації 8

## Алгоритм роботи відділу постачання запасних частин на ремонтні ділянки



Рисунок Д.9 – Слайд презентації 9

Інтерфейс основного модуля програмної системи обліку поставок запасних частин

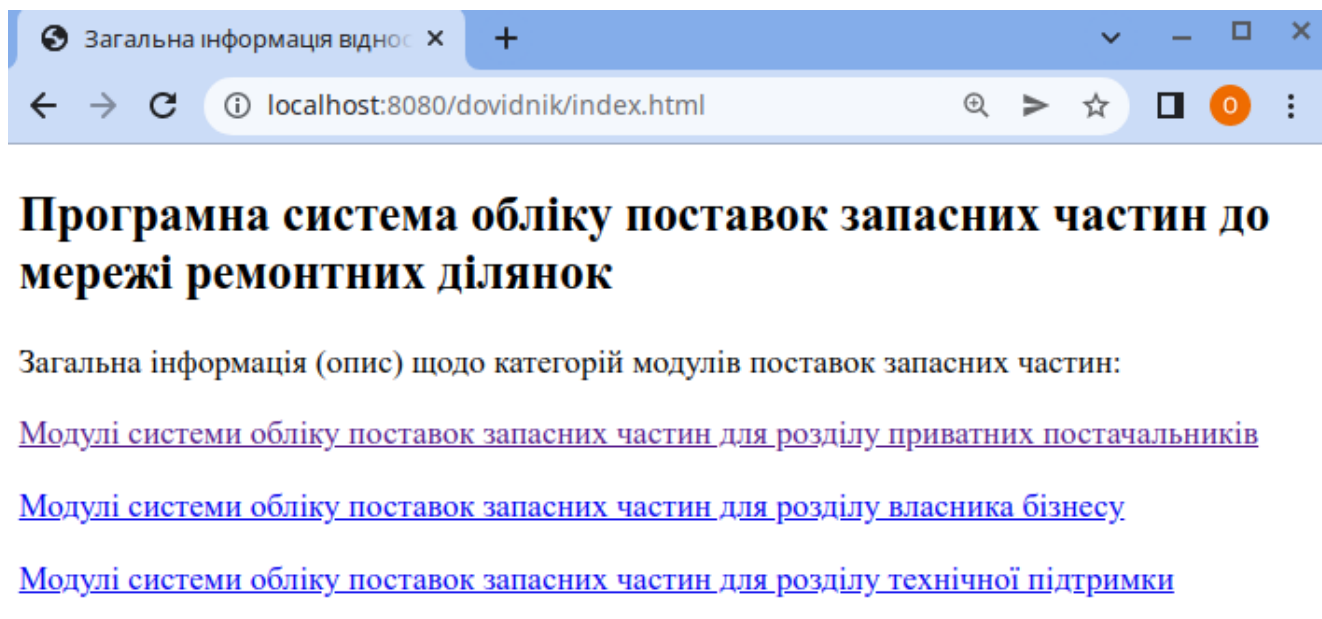


Рисунок Д.10 – Слайд презентації 10

Інтерфейс додаткового модуля програмної системи обліку поставок запасних частин, що використовується для розділу власників бізнесу

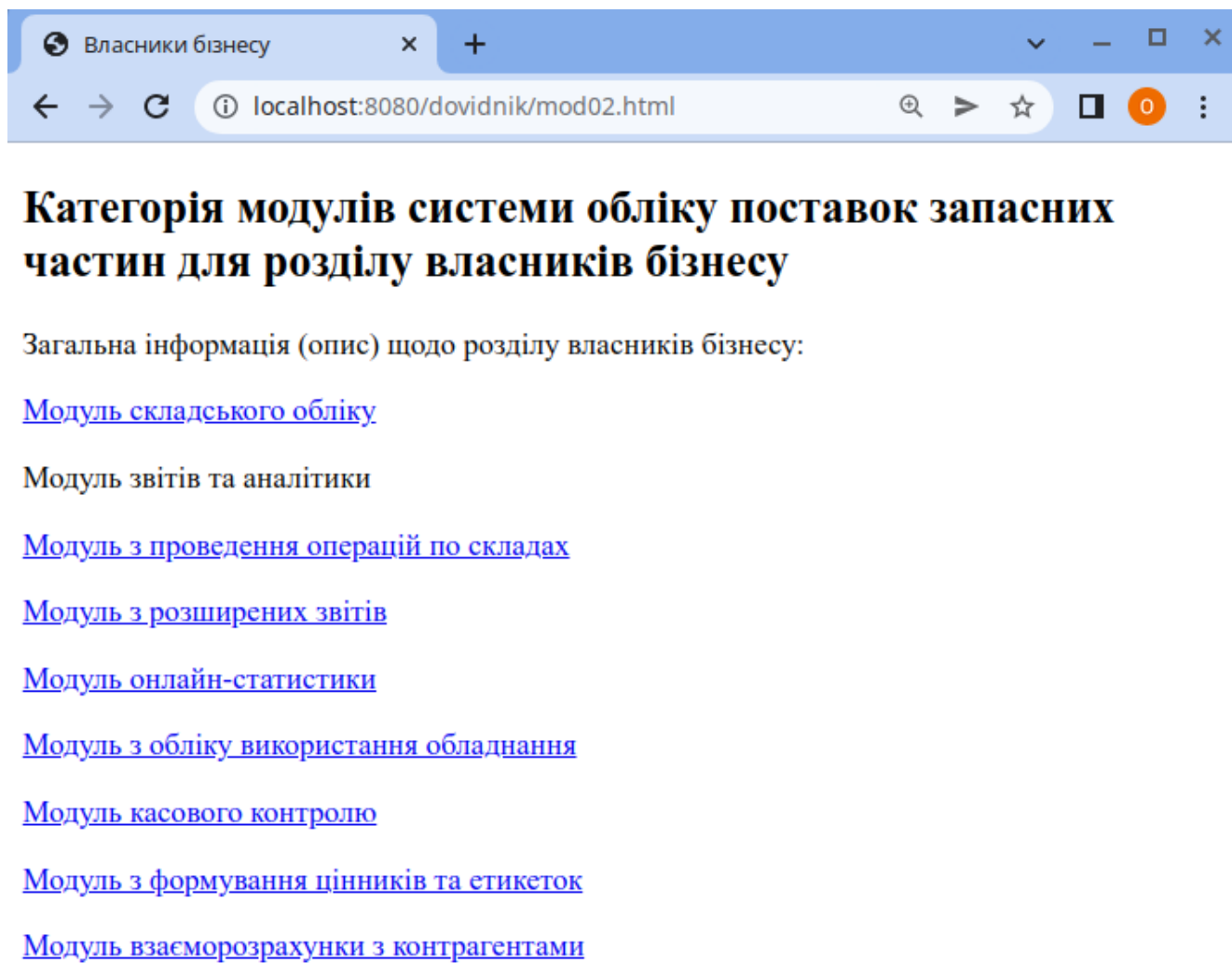
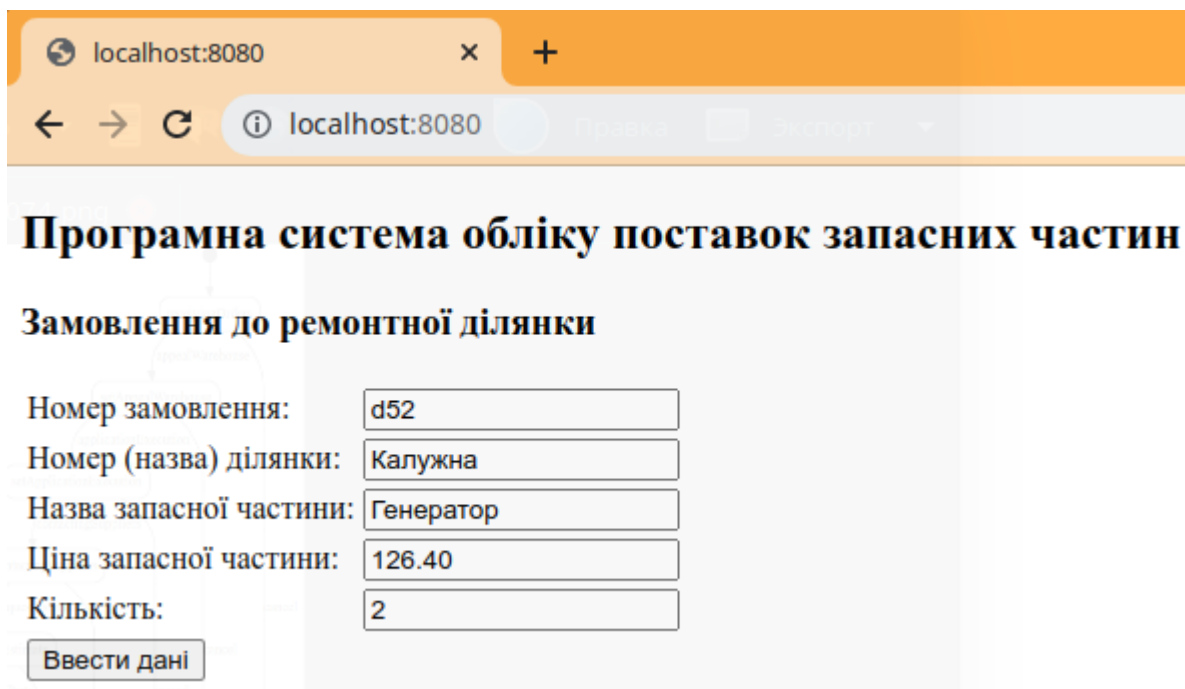


Рисунок Д.11 – Слайд презентації 11

Приклад заповнення необхідних полів в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок



The image shows a web browser window with the address bar set to localhost:8080. The page title is "Програмна система обліку поставок запасних частин". Below the title, there is a section titled "Замовлення до ремонтної ділянки". This section contains a form with five input fields and a button. The fields are labeled as follows:

Номер замовлення:	d52
Номер (назва) ділянки:	Калужна
Назва запасної частини:	Генератор
Ціна запасної частини:	126.40
Кількість:	2

Below the form is a button labeled "Ввести дані".

Рисунок Д.12 – Слайд презентації 12

Приклад успішного заповнення необхідних полів в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок

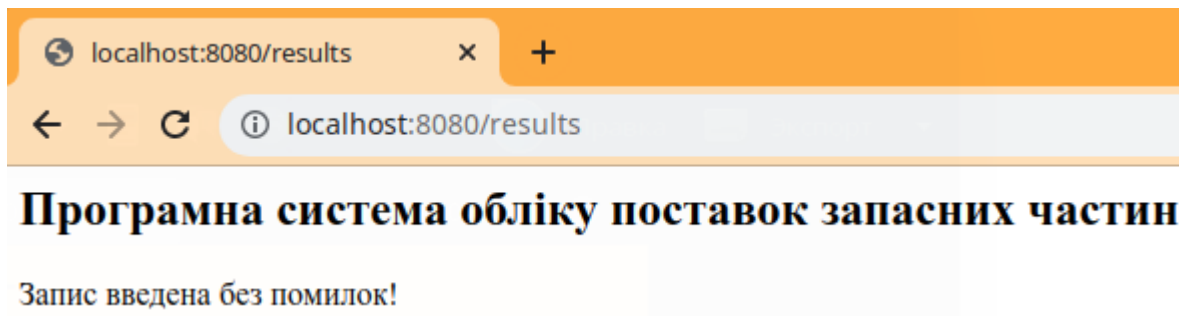
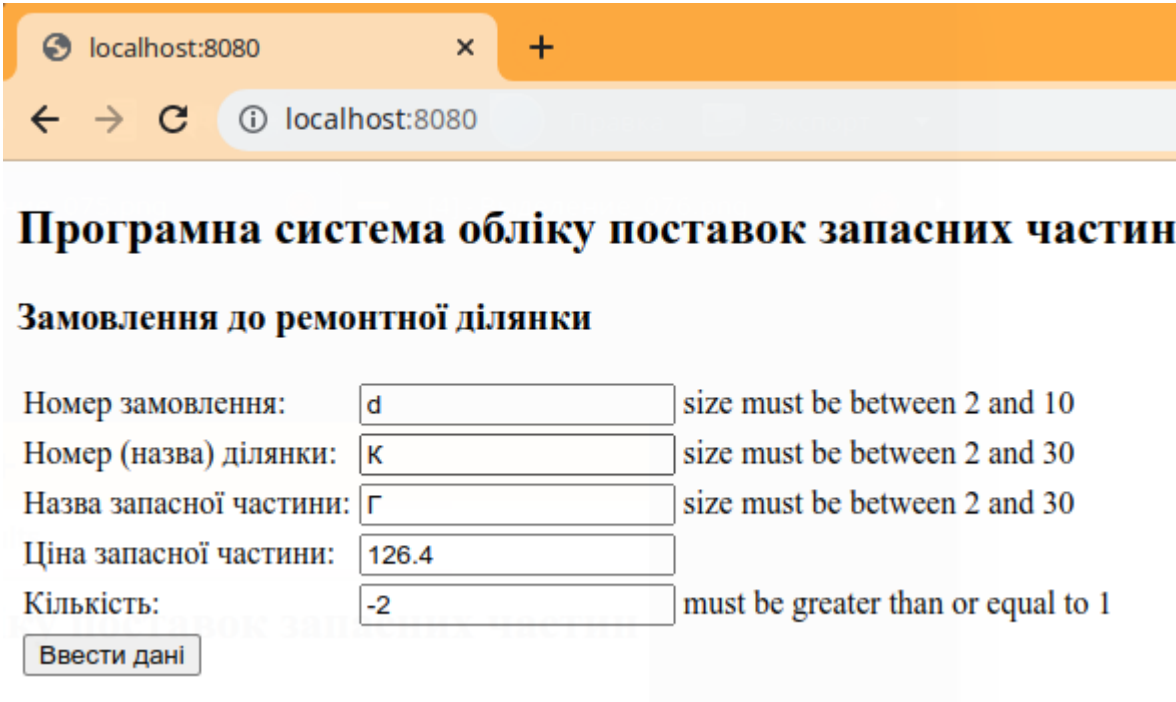


Рисунок Д.13 – Слайд презентації 13

Приклад заповнення необхідних полів некоректними даними в модулі заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок



The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The page title is 'Програмна система обліку поставок запасних частин'. Below the title, the section is 'Замовлення до ремонтної ділянки'. There are five input fields with validation messages:

Номер замовлення:	<input type="text" value="d"/>	size must be between 2 and 10
Номер (назва) ділянки:	<input type="text" value="К"/>	size must be between 2 and 30
Назва запасної частини:	<input type="text" value="Г"/>	size must be between 2 and 30
Ціна запасної частини:	<input type="text" value="126.4"/>	
Кількість:	<input type="text" value="-2"/>	must be greater than or equal to 1

At the bottom left, there is a button labeled 'Ввести дані'.

Рисунок Д.14 – Слайд презентації 14

## Висновки:

У бакалаврській дипломній роботі:

- 1) проведено аналіз моделей управління поставками запасних частин у мережу ремонтних ділянок та порівняльний аналіз аналогів;
- 2) виконано аналіз розвитку сучасних систем обліку поставок запасних частин до мережі ремонтних ділянок;
- 3) створена постановка задачі;
- 4) виконано проектування модулів системи обліку поставок запасних частин для розділу приватних постачальників, власника бізнесу та технічної підтримки;
- 5) розроблені модулі обслуговування заявок на ремонтних ділянках та інтерфейс програмної системи обліку поставок запасних частин;
- 6) розроблено алгоритм роботи відділу постачання запасних частин на ремонтній ділянці;
- 7) виконано тестування модуля заповнення замовлень на поставку запасних частин до мережі ремонтних ділянок;
- 8) виконано тестування frontend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок;
- 9) виконано тестування backend частини програмної системи обліку поставок запасних частин до мережі ремонтних ділянок.

Рисунок Д.15 – Слайд презентації 15