

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка модуля «Облік та аналіз гарантійних звернень» інформаційної
системи мережі магазинів побутової техніки

(тема)

Виконав:

здобувач 4 року навчання,

групи ІТУ-21-2

Максим КРИКУН

(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології

управління

(повна назва освітньої програми)

Керівник: доц. каф. ІУС Олена МІХНОВА

(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри ІУС



(підпис)

Костянтин ПЕТРОВ

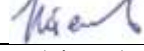
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наукКафедра Інформаційних управляючих системРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)Освітня програма Інформаційні технології управління
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)“ 19 ” травня 2025 р.**ЗАВДАННЯ****НА КВАЛІФІКАЦІЙНУ РОБОТУ**здобувачеві Крикуну Максиму Ігоровичу
(прізвище, ім'я, по батькові)1. Тема роботи Розробка модуля «Облік та аналіз гарантійних звернень» інформаційної системи мережі магазинів побутової технікизатверджена наказом по університету від “ 19 ” травня 2025 р. № 370Ст2. Термін подання здобувачем роботи до екзаменаційної комісії “ 19 ” червня 2025 р.3. Вихідні дані до роботи Вимоги до автоматизації процесів фіксації звернень, контролю термінів їх обробки, ведення журналу змін, формування статистики та рекомендацій щодо закупівель. Функціональні потреби користувачів різних ролей: сервісних працівників, відділу постачання, адміністраторів системи.4. Перелік питань, що потрібно опрацювати у роботі Аналіз предметної області та постановка задачі, розробка архітектури інформаційної системи, проектування бази даних та структури таблиць, реалізація програмного забезпечення для авторизації, обробки звернень, контролю термінів та формування звітності, створення модуля аналітики та рекомендацій, розробка графічного інтерфейсу користувача, проведення ефективності.

КАЛЕНДАРНИЙ ПЛАН


№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та обґрунтування потреби в автоматизації	19.05.2025 - 20.05.2025	Виконано
2	Аналіз аналогічних програмних рішень	21.05.2025 - 23.05.2025	Виконано
3	Формулювання завдання розробки	24.05.2025 - 01.06.2025	Виконано
4	Опис архітектури об'єкта розробки на рівні функцій	02.05.2025 - 20.05.2025	Виконано
5	Розробка та обґрунтування елементів інформаційної забезпечуючої системи	21.05.2025 - 24.05.2025	Виконано
6	Розробка та обґрунтування елементів програмної забезпечуючої системи	25.05.2025 - 27.05.2025	Виконано
7	Оформлення пояснювальної записки	28.05.2025 - 01.06.2025	Виконано
8	Захист кваліфікаційної роботи	19.06.2025	

Дата видачі завдання 19 травня 2025 р.

Здобувач


_____ (підпис)

Керівник роботи


_____ (підпис)

доц. каф. ІУС Олена Міхнова

_____ (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 75 с., 19 рис., 2 табл., 1 дод., 19 джерел.

АНАЛІТИКА, АВТОМАТИЗАЦІЯ, ГАРАНТІЙНЕ ОБСЛУГОВУВАННЯ, ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРФЕЙС КОРИСТУВАЧА, ОБЛІК ЗВЕРНЕНЬ, MYSQL, PYTHON.

У галузі роздрібної торгівлі побутовою технікою важливою складовою якісного обслуговування клієнтів є налагоджений процес гарантійного супроводу. Ефективність такої роботи значною мірою залежить від здатності підприємства оперативно фіксувати звернення, контролювати їх виконання та аналізувати накопичену інформацію для покращення сервісу.

У межах кваліфікаційної роботи аналізується підхід до систематизації звернень клієнтів і пошуку способів їх ефективної обробки. Основною метою є створення програмного модуля, що дозволяє реєструвати звернення, контролювати статуси, здійснювати аналітику та формувати рекомендації для покращення логістики.

Під час реалізації проєкту проведено аналіз предметної області, розроблено архітектуру та базу даних, створено графічний інтерфейс, реалізовано функції перегляду й обробки звернень. Особливу увагу приділено модулю аналітики та рекомендацій.

Програмну частину реалізовано з використанням Python, Tkinter та MySQL. Отримане рішення може бути впроваджене у роботу сервісного підрозділу з подальшим масштабуванням функціоналу.

ABSTRACT

Thesis: 75 pages, 19 figures, 2 tables, 1 appendix, 19 sources.

ANALYTICS, AUTOMATION, CLAIM MANAGEMENT, INFORMATION SYSTEM, MYSQL, PYTHON, USER INTERFACE, WARRANTY SERVICE.

In the field of consumer electronics retail, a well-organized warranty support process is a key component of high-quality customer service. The efficiency of such service largely depends on the enterprise's ability to promptly register claims, monitor their processing, and analyze accumulated data to improve service quality.

This qualification work explores an approach to structuring customer claims and identifying ways to process them effectively. The main objective is to develop a software module that enables claim registration, status tracking, analytics, and the generation of recommendations for optimizing logistics.

During the project, the subject area was analyzed, the system architecture and database were designed, a graphical user interface was developed, and key functions for viewing and processing claims were implemented. Special attention was given to the analytics and recommendations module.

The software component was developed using Python, Tkinter, and MySQL. The resulting solution can be implemented in a service department and further expanded in terms of functionality.

ЗМІСТ

Скорочення та умовні позначки.....	8
Вступ.....	9
1 Аналіз предметної області та обґрунтування потреби в автоматизації....	10
1.1 Аналіз предметної області та обґрунтування потреби в автоматизації.....	10
1.1.1 Опис підприємства.....	10
1.1.2 Структурна схема та функції підрозділів.....	11
1.2 Аналіз проблеми автоматизації обліку гарантійних звернень.....	13
2 Аналіз аналогічних програмних рішень.....	15
2.1 Огляд існуючих рішень для обліку гарантійних звернень	15
2.2 Порівняння програмних продуктів.....	16
2.3 Обґрунтування необхідності розробки власного модуля.....	18
3 Формулювання завдання розробки.....	20
3.1 Опис вимог до об'єкта розробки.....	20
3.1.1 Розробка системних вимог.....	20
3.1.2 Розробка функціональних вимог.....	21
3.1.3 Розробка вимог до інтерфейсу.....	23
3.2 Обґрунтування мети і критеріїв ефективності об'єкта розробки.....	26
4 Опис архітектури об'єкта розробки на рівні функцій.....	28
4.1. Загальна архітектура системи.....	28
4.2 Схема функціональної взаємодії.....	29
5 Розробка та обґрунтування елементів інформаційної забезпечуючої системи.....	33
5.1. Обґрунтування вибору системи управління базами даних (СУБД)....	33
5.2. Логічна й фізична модель даних системи обліку та аналізу гарантійних звернень.....	35

5.3 Створення бази даних системи обліку та аналізу гарантійних звернень на платформі СУБД MySQL.....	38
6 Розробка та обґрунтування елементів програмної забезпечуючої системи.....	41
6.1 Загальний опис програмної системи.....	41
6.2 Архітектура програмної частини.....	42
6.3 Структура програми.....	44
6.4 Екранні форми інтерфейсу користувача (UI).....	54
Висновки.....	63
Перелік джерел посилання.....	64
Додаток А Графічний матеріал.....	66

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІК – інтерфейс користувача

ІС – інформаційна система

ПЗ – програмне забезпечення

СУБД – система управління базами даних

ACID – Atomicity Consistency Isolation Durability

CRM – Customer Relationship Management

DFD – Data Flow Diagram

ER – Entity-Relationship model

GNU GPL – General Public License

HTTPS – HyperText Transfer Protocol Secure

PDF – Portable Document Format

UI – User Interface

UX – User Experience

ВСТУП

У сучасних умовах, коли ринок побутової техніки стрімко зростає, питання якісного післяпродажного обслуговування набуває особливого значення. Гарантійна підтримка клієнтів – це не просто частина сервісу, а й важливий елемент репутації компанії. У багатьох торговельних мережах спостерігаються труднощі з веденням обліку звернень, відстеженням їх статусів, дотриманням строків і формуванням достовірної статистики. Це ускладнює щоденну роботу персоналу та створює проблеми з оцінкою ефективності сервісу на рівні керівництва.

Аналіз практик показав, що в багатьох компаніях звернення фіксуються у розрізних таблицях або застарілих програмах. Відсутність єдиного середовища призводить до збоїв у координації, затримок у ремонтах і втрати важливої інформації. Тому виникла ідея створити спеціалізований модуль інформаційної системи (ІС), який би вирішував ці завдання комплексно.

Метою проекту стало розроблення зручного інструменту для реєстрації звернень, контролю статусів, аналізу даних і формування рекомендацій. У межах реалізації враховано як технічні аспекти, так і специфіку роботи сервісних відділів. Створено архітектуру рішення, інтерфейс користувача та функціональні модулі для обліку й аналітики, включаючи блок рекомендацій.

Запропоноване рішення має практичну користь – воно пришвидшує обслуговування, зменшує кількість помилок і забезпечує контрольованість процесів. Система дозволяє підприємству краще організувати роботу, мати доступ до актуальної інформації й планувати подальші дії на основі фактів. Також система має потенціал для подальшого розвитку – її легко адаптувати до змін вимог компанії або масштабувати під нові завдання в межах інших підрозділів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ПОТРЕБИ В АВТОМАТИЗАЦІЇ

1.1 Опис підприємства, структурна схема та функції підрозділів

1.1.1 Опис підприємства

Об'єктом дослідження кваліфікаційної роботи є мережа магазинів побутової техніки, яка займається роздрібним продажем електроніки, великої та дрібної побутової техніки, а також аксесуарів. Компанія працює як через звичайні магазини, розташовані у різних містах України, так і через власний онлайн-магазин, що дає змогу здійснювати дистанційні покупки.

Основною метою діяльності підприємства є забезпечення споживачів якісною технікою провідних світових брендів, надання професійних консультаційних послуг та забезпечення належного післяпродажного обслуговування, включаючи гарантійний сервіс.

Управління бізнес-процесами на підприємстві здійснюється за допомогою корпоративної ІС, яка допомагає контролювати торгівлю, склад, логістику, фінанси і, що важливо для цієї роботи, сервісне обслуговування клієнтів.

Окремо варто виділити напрямок гарантійного обслуговування, адже швидкість і якість його виконання впливають на думку покупця про компанію загалом. Тому актуальною є задача автоматизації цього напрямку шляхом розробки спеціалізованого модуля ІС «Облік та аналіз гарантійних звернень».

1.1.2 Структурна схема та функції підрозділів

Організаційна структура мережі магазинів побутової техніки складається з таких основних підрозділів:

- адміністрація підприємства виконує стратегічне управління, формує політику розвитку мережі, ухвалює рішення щодо інвестицій, управління персоналом та розширення ринку;

- відділ продажів організовує процес реалізації товарів як у фізичних магазинах, так і через електронну комерційну платформу. До його обов'язків також належить впровадження програм лояльності та підвищення якості обслуговування клієнтів.

- складський відділ та логістика забезпечують приймання, облік і зберігання товарів, а також їх транспортування до роздрібних точок або кінцевого споживача;

- інформаційно-технологічний відділ (ІТ-відділ) відповідає за стабільну роботу ІС підприємства, інтеграцію нових програмних модулів, технічну підтримку користувачів та обслуговування баз даних;

- відділ маркетингу проводить рекламні кампанії, займається просуванням бренду на ринку, а також аналізує конкурентне середовище для адаптації бізнес-стратегії;

- фінансовий відділ та бухгалтерія здійснюють облік господарських операцій, контролюють фінансові потоки, забезпечують формування внутрішньої та зовнішньої звітності;

- відділ по роботі з постачальниками організовує процес закупівлі продукції, веде переговори з партнерами, укладає договори та координує дії щодо гарантійного супроводу техніки;

- відділ сервісного обслуговування здійснює прийом та обробку звернень клієнтів у межах гарантійного і післягарантійного обслуговування. Цей підрозділ координує співпрацю з авторизованими сервісними

центрами, контролює строки виконання ремонтів і веде облік звернень за товарами та брендами.

Організаційну структуру підприємства наведено на рисунку 1.1.

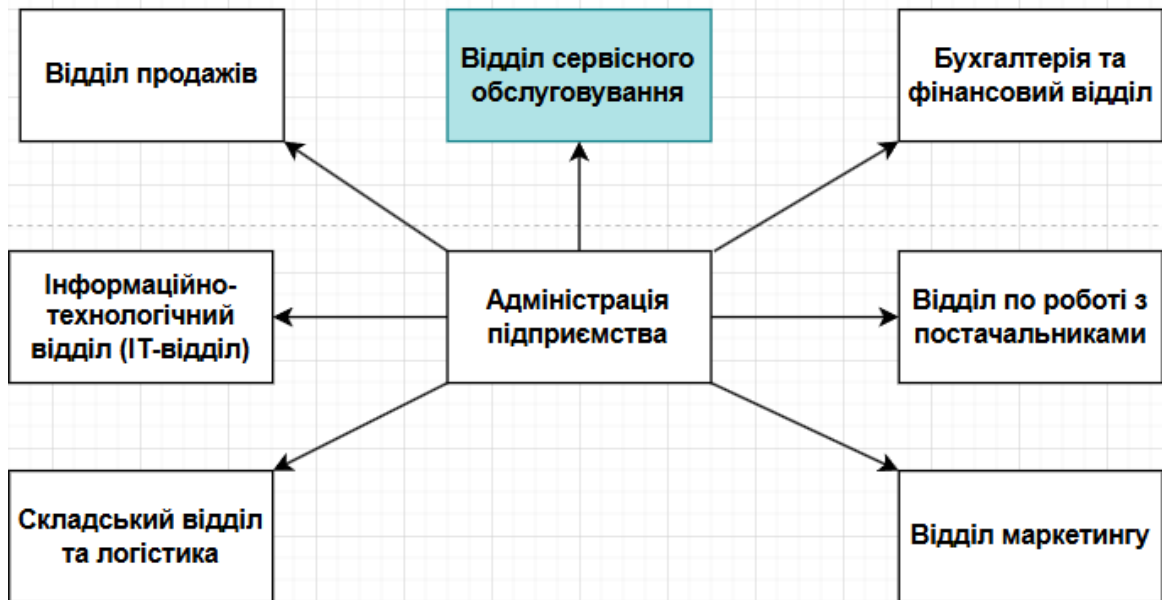


Рисунок 1.1 – Схема організаційної структури мережі магазинів побутової техніки

Саме для відділу сервісного обслуговування передбачена розробка модуля «Облік та аналіз гарантійних звернень», який забезпечить автоматизацію таких процесів:

- реєстрація нових звернень з фіксацією даних клієнта, товару, дати звернення та опису дефекту;
- відстеження статусу виконання звернень (у роботі, завершено, очікує запчастини тощо);
- контроль термінів ремонту відповідно до гарантійних умов;
- формування звітності про кількість звернень по брендах, товарах, регіонах;
- аналіз ефективності роботи сервісних центрів та рекомендації для покращення обслуговування.

1.2 Аналіз проблеми автоматизації обліку гарантійних звернень

У сучасних умовах, коли конкуренція на ринку побутової техніки постійно зростає, на перший план виходить не лише якість товару, а й рівень сервісу, який надає компанія. Покупці дедалі частіше звертають увагу на те, як швидко та зручно можна вирішити питання, пов'язані з гарантійним обслуговуванням. Якщо клієнт не отримує належної підтримки після покупки, це може вплинути на його подальший вибір магазину або бренду.

Проте в багатьох компаніях, зокрема й у мережах побутової техніки, процес обліку таких звернень досі залишається неавтоматизованим або реалізованим дуже частково. Наприклад, записи ведуться вручну – у блокнотах, таблицях Excel або інших непристосованих для цього програмних рішеннях. Такий підхід може видаватися зручним на початкових етапах, коли обсяг звернень є незначним і проблеми залишаються малопомітними. Однак зі збільшенням кількості звернень виникають труднощі з оперативним пошуком необхідної інформації, зростає ризик втрати або ненавмисного видалення важливих даних.

Крім того, у компанії немає єдиного електронного реєстру для гарантійних випадків. Через це частина даних дублюється або губиться, а пошук потрібної інформації займає надто багато часу. Аналіз проблемних товарів або частих поломок також проводиться дуже рідко, бо для цього потрібно вручну збирати дані з кількох джерел. Як наслідок – закупівельники не мають повної картини і можуть знову замовити техніку, яка вже викликала нарікання.

Ще одна проблема – відсутність зручного контролю термінів обробки звернень. Часто працівники відстежують строки вручну, тож не дивно, що трапляються затримки з ремонтом або відповідями клієнтам.

Усе це свідчить про потребу у створенні спеціального модуля, який би дозволяв автоматично реєструвати звернення, вести їх облік, формувати

звіти й аналітику. Як зазначається у сучасних наукових дослідженнях, цифрові технології, зокрема підходи прогнозування на основі даних, стають основою трансформації систем aftersales та гарантійного обслуговування, забезпечуючи їх ефективність і здатність до адаптації [1]. Завдяки такому рішенню сервісний відділ зможе працювати ефективніше, а керівництво – швидше приймати рішення на основі актуальної статистики. Крім того, підвищення якості обслуговування сприятиме зміцненню позитивного іміджу компанії та зростанню рівня довіри з боку клієнтів.

2 АНАЛІЗ АНАЛОГІЧНИХ ПРОГРАМНИХ РІШЕНЬ

2.1 Огляд існуючих рішень для обліку гарантійних звернень

На сучасному ринку інформаційних технологій представлено низку готових програмних рішень, придатних для автоматизації обліку клієнтських звернень та сервісного обслуговування. Ці системи розрізняються за своїм функціоналом, вартістю впровадження, складністю налаштування та інтеграції з іншими системами підприємства.

До найпоширеніших систем, які можуть бути використані для обліку гарантійних звернень, належать:

– Zoho CRM дозволяє реєструвати звернення та здійснювати базовий аналіз по клієнтській базі [2]. Водночас для повноцінного обліку гарантійних звернень і аналітики потрібна додаткова інтеграція з модулями аналітики Zoho Reports, що ускладнює впровадження і підвищує вартість;

– Freshdesk сервіс-деск платформа, орієнтована на облік звернень до служби підтримки [3]. Має функції для контролю статусів звернень та формування звітів, однак не забезпечує інтеграції з обліком товарів і не надає рекомендацій для відділу закупівель;

– рішення на базі Excel/Google Таблиць. Велика кількість невеликих компаній використовують табличні редактори для реєстрації гарантійних звернень [4, 5]. Ці інструменти мають низький поріг входу, але не дозволяють ефективно автоматизувати процес аналізу звернень і складати аналітичні звіти в автоматичному режимі. Має обмежені інструменти для глибокої аналітики гарантійних випадків без додаткового налаштування;

– ServiceDesk Plus потужна ITSM-система, яка підтримує повний цикл управління інцидентами [6]. Придатна для середнього і великого бізнесу, але має високу вартість ліцензії та складність в налаштуванні під задачі роздрібною торгівлі.

У роботах [7], [8] наголошується, що готові Customer Relationship Management (CRM) рішення зазвичай орієнтовані на зовнішню взаємодію з клієнтами, але не завжди враховують особливості післяпродажного обслуговування та потреби аналізу гарантійних звернень.

2.2 Порівняння програмних продуктів

Для більш наочного аналізу існуючих рішень представлено таблицю порівняння за ключовими критеріями (таблиця 2.1).

Таблиця 2.1 – Порівняння програмних рішень для обліку гарантійних звернень

Критерій	Zoho CRM	Freshdesk	Google Таблиці	ServiceDesk Plus
1	2	3	4	5
Реєстрація звернень	Так	Так	Так	Так
Візуалізація даних	Так	Так	Ні	Так
Інтеграція з обліком товарів	Ні	Ні	Ні	Так

Кінець таблиці 2.1

1	2	3	4	5
Рекомендації для закупівель	Ні	Ні	Ні	Так
Статистика за брендами/ категоріями	Обмежено	Обмежено	Ні	Так
Складність налаштування	Середня	Низька	Низька	Висока
Вартість впровадження	Середня	Висока	Низька	Висока
Орієнтованість на сервіс після продажу	Ні	Так	Ні	Так

Проведене порівняння дозволяє зробити висновок, що кожна з розглянутих систем орієнтована на задоволення окремих потреб користувачів і має власні функціональні обмеження. Така платформа, як Zoho CRM, демонструє зручність у реєстрації звернень і веденні клієнтської бази, однак її можливості щодо поглибленої аналітики гарантійних випадків, а також інтеграції з обліком товарів є обмеженими.

Система Freshdesk забезпечує ефективну підтримку для обробки клієнтських звернень, але водночас має недостатній рівень взаємодії з іншими бізнес-процесами компанії. Щодо ServiceDesk Plus, то вона вирізняється широким спектром функцій, включаючи аналіз звернень та генерацію рекомендацій для відділу закупівель, проте її висока вартість і

складність у впровадженні можуть стати перешкодою для невеликих або середніх компаній.

Окрему групу рішень становлять інструменти на базі Google Таблиць. Вони є доступними та простими у використанні, але значно поступаються іншим системам у плані автоматизації, аналітичних можливостей та рівня безпеки.

У підсумку, жодне з наявних рішень не забезпечує одночасно високого рівня інтеграції, гнучкої аналітики й помірної вартості. Це свідчить про доцільність розробки власного спеціалізованого програмного модуля, який буде враховувати специфіку бізнес-процесів конкретного підприємства та дозволить оптимізувати обробку гарантійних звернень. Найбільш розвинені компанії звертають увагу не лише на кількість заявок, а й на динаміку витрат. Згідно з аналітичними даними, незважаючи на скорочення кількості звернень, витрати на гарантію зростають – через ускладнення ремонту та підвищення трудових витрат [9]. Це підкреслює необхідність глибшої аналітики та ефективнішого управління гарантійним сервісом.

2.3 Обґрунтування необхідності розробки власного модуля

На основі проведеного аналізу програмних рішень, що вже існують на ринку, можна дійти висновку про обґрунтовану потребу створення власного модуля для обліку та аналітики гарантійних звернень. Подібне рішення дозволить краще адаптувати інструменти до специфіки роботи конкретного підприємства й уникнути проблем, пов'язаних із використанням громіздких або надмірно універсальних систем.

Серед технічних переваг власного модуля можна виділити:

- повна адаптація під конкретні бізнес-процеси компанії, що дозволить уникнути зайвого функціоналу та спростить використання модуля;

- легкість інтеграції з внутрішніми системами обліку продажів, закупівель та складського управління, що забезпечить оперативність та зручність обміну даними;

- можливість подальшого розвитку та масштабування функцій відповідно до змін у бізнес-процесах та вимогах компанії.

Щодо фінансових аспектів, варто зазначити такі переваги:

- можливість реалізації на основі безкоштовних інструментів і мов програмування, таких як Python та MySQL, що значно знижує вартість розробки;

- відсутність постійних витрат на ліцензування комерційного програмного забезпечення (ПЗ), що значно знижує вартість володіння;

- скорочення часу, який витрачається персоналом на обробку звернень вручну, що в підсумку позитивно позначиться на ефективності роботи.

Перелічені переваги дозволяють стверджувати, що створення власного модуля є доцільним як із технічної, так і з економічної точки зору. Це рішення дозволить не лише підвищити рівень автоматизації сервісного обслуговування, а й забезпечить більшу гнучкість і контроль над процесами обробки гарантійних звернень.

Окремо слід підкреслити важливість можливостей аналітики для виявлення потенційних аномалій у зверненнях. Сучасні системи використовують алгоритми машинного навчання для виявлення шахрайських шаблонів у зверненнях на обслуговування [10].

3 ФОРМУЛЮВАННЯ ЗАВДАННЯ РОЗРОБКИ

3.1 Опис вимог до об'єкта розробки

3.1.1 Розробка системних вимог

Першим етапом під час створення ІС є визначення її системних вимог. Цей процес має вирішальне значення, оскільки саме на його основі вибудовується вся архітектура майбутнього рішення. У випадку модуля для обліку та аналізу гарантійних звернень у мережі магазинів побутової техніки системні вимоги дозволяють зорієнтуватися у виборі потрібних технологій і передбачити обмеження, з якими доведеться працювати.

Системні вимоги стосуються переважно зовнішніх умов експлуатації ПЗ. Їхня мета – забезпечити правильне функціонування продукту в середовищі конкретного підприємства, врахувати технічні обмеження, питання безпеки, сумісність із наявною ІТ-інфраструктурою та можливості подальшої підтримки.

Передусім модуль має бути придатним для встановлення на корпоративний сервер або в межах локальної мережі підприємства, забезпечуючи при цьому доступ користувачів через внутрішній веб-інтерфейс. Це вимагає технічної сумісності з типовими компонентами ІТ-інфраструктури – операційними системами на зразок Windows Server чи Linux, веб-серверами типу Apache або Nginx, а також популярними системами управління базами даних, як-от MySQL або PostgreSQL.

Особливу увагу слід приділити питанню надійності системи. Модуль має бути здатним ефективно функціонувати при паралельній роботі кількох користувачів, не допускаючи суттєвих затримок. Обов'язковими є також механізми створення резервних копій і відновлення даних, а також вести журнали подій для цілей аудиту та моніторингу активності.

Не менш важливим є і забезпечення можливості масштабування. Система повинна легко адаптуватися до змін, наприклад, у випадку розширення торгівельної мережі, збільшення обсягу оброблюваної

інформації чи підключення нових співробітників. Це потребує впровадження гнучких технічних рішень, які дозволяють нарощувати ресурси без повної перебудови архітектури.

Окремим блоком вимог виступає безпека. В умовах дії Закону України «Про захист персональних даних» [17] ПЗ зобов'язане забезпечувати як збереження конфіденційності клієнтської інформації, так і прозорість доступу до неї. Це означає: шифрування трафіку через HyperText Transfer Protocol Secure (HTTPS), чітка ієрархія прав доступу, автентифікація, автоматичне завершення неактивних сесій – усе це має бути частиною базової конфігурації.

Також варто врахувати, що більшість середніх підприємств мають обмежені ІТ-ресурси, Тому система має бути невибагливою до апаратного забезпечення: працювати на офісному сервері або віртуальній машині з 4 ГБ оперативної пам'яті, процесором із двома ядрами та не менше 100 ГБ вільного місця на жорсткому диску.

У підсумку, грамотно сформульовані системні вимоги створюють основу для подальшого технічного проектування, впливають на вибір засобів реалізації та гарантують, що розроблений модуль буде відповідати всім необхідним параметрам сумісності, надійності та захисту даних у рамках уже діючої ІТ-інфраструктури.

3.1.2 Розробка функціональних вимог

Функціональні вимоги до модуля, що відповідає за облік та аналітику гарантійних звернень, описують ключові завдання, які має реалізовувати система, а також передбачувані дії користувача та загальні принципи обробки даних.

Цей модуль створюється з метою спростити та автоматизувати процеси реєстрації звернень, їх подальшої обробки, аналізу накопиченої інформації та забезпечення належного контролю за виконанням гарантійних зобов'язань. Все це повинно відбуватися у напівавтоматичному режимі. Інтерфейс системи має бути інтуїтивно зрозумілим: користувач повинен мати змогу швидко внести дані або знайти потрібну інформацію. Важливо також дотримуватись внутрішніх процедур обробки заявок та мати інструменти для оперативного отримання звітів чи статистики з бази.

До найважливіших функціональних характеристик модуля належить підтримка повного циклу роботи зі зверненням. На початковому етапі користувач вносить усі необхідні відомості: інформацію про клієнта, товар, дату покупки та дату звернення, опис несправності й номер гарантії. Система самостійно присвоює зверненню статус очікування й перевіряє, чи не пропущені обов'язкові поля.

У процесі подальшої обробки звернення може змінювати свій статус. При цьому кожне оновлення супроводжується фіксацією часу, коментарем і зазначенням відповідального працівника. Система повинна слідкувати за дотриманням строків виконання, сповіщаючи про порушення. Передбачено зручну систему фільтрації й сортування: за ПШБ, назвою моделі, брендом, гарантійним кодом або поточним статусом.

Окрема підсистема виконує аналітичну обробку даних. Вона дозволяє отримати статистику щодо поширених несправностей, середнього часу до звернення, побудови графіків і діаграм. Ці дані використовуються для формування рекомендацій, насамперед для відділу закупівель. На основі аналізу система виявляє моделі з частими поломками або коротким терміном експлуатації й формує звіти.

Для фіксації всіх змін у зверненнях реалізовано окремий журнал подій. У ньому зберігається інформація про те, хто, коли й що саме оновив. Цей механізм дозволяє бачити історію роботи з кожним запитом.

У програмі передбачено три рівні доступу: адміністратор має повний контроль над системою; працівник сервісного підрозділу реєструє й обробляє звернення; керівник відділу постачання має доступ до статистики та рекомендацій.

Щодо збереження інформації, використовується СУБД MySQL. Вона містить таблиці, в яких зберігаються звернення, клієнти, товари, історія змін. Дані про бренди та моделі нормалізовані й впорядковані через довідники. Вся інформація обробляється з дотриманням вимог чинного законодавства України про захист персональних даних[17].

3.1.3 Розробка вимог до інтерфейсу

Інтерфейс, через який відбувається взаємодія користувача із системою, відіграє одну з ключових ролей у зручності та ефективності роботи з ПЗ. Саме тому для модуля «Облік та аналіз гарантійних звернень» важливо передбачити просту й логічну структуру інтерфейсу, яка забезпечуватиме швидкий доступ до головних функцій, комфортну навігацію та зрозуміле відображення даних.

Інтерфейс повинен бути інтуїтивно зрозумілим, мінімалістичним і відповідати сучасним стандартам User Interface (UX/UI) дизайну. Особлива увага приділяється адаптації до специфіки діяльності працівників сервісного відділу, які не є технічними спеціалістами. Тому система повинна забезпечити швидке навчання нових користувачів без потреби в додаткових інструкціях.

Основні вимоги до інтерфейсу користувача:

– наявність головного меню , з можливістю швидкого переходу до розділів «Звернення», «Звіти», «Аналітика», «Рекомендації», тощо.

– форма для створення нового звернення має містити поля для введення ПІБ клієнта, контакту, моделі товару, серійного номера, дати покупки, дати звернення, опису несправності, а також список вибору для статусу звернення і відповідального працівника.

– можливість редагування звернень із журналюванням змін (хто, коли, що змінив).

– інструменти пошуку та фільтрації: фільтрація за статусом, моделлю, працівником, датою тощо.

– відображення звернень у вигляді таблиці з можливістю сортування та перегляду деталей у модальному вікні або на окремій сторінці.

– вкладка для перегляду звітів з опцією вибору періоду, бренду, категорії.

– графіки (стовпчикові, лінійні, кругові) для візуалізації динаміки звернень, поширених несправностей тощо.

– сторінка із сформованими рекомендаціями для відділу закупівель, із можливістю експорту у форматі Portable Document Format (PDF)/Excel.

Щодо візуального оформлення, інтерфейс повинен бути витриманий у нейтральних тонах, з логічними акцентами на важливих елементах (наприклад, кнопках або індикаторах статусу). Для кращого сприйняття стану звернень бажано візуально виділяти їх за допомогою кольору чи іконок.

Крім того, необхідно врахувати вимоги до доступності: усі елементи повинні бути чітко підписані, мати достатній контраст і підтримувати навігацію за допомогою клавіатури – це робить інтерфейс зручним для ширшого кола користувачів.

Загалом добре продуманий інтерфейс допоможе зменшити навантаження на персонал, мінімізувати кількість помилок і в результаті – зробити роботу сервісного відділу ефективнішою.

Однією з основних вимог є впровадження системи автентифікації користувачів. Кожен користувач повинен мати унікальні облікові дані для

входу до системи. Паролі зберігаються у зашифрованому вигляді. Крім того, доцільним є впровадження політики складності паролів та періодичної їх зміни.

Система має реалізовувати розмежування прав доступу. Користувачі розподіляються на ролі відповідно до їхніх обов'язків. Наприклад, оператор має доступ до створення та редагування звернень, керівник відділу може переглядати звіти та аналітику, тоді як адміністратор має повний доступ до налаштувань системи та управління користувачами.

Щоб унеможливити передавання інформації, рекомендується використовувати протокол HTTPS. Це допомагає уникнути перехоплення даних під час їх передачі між клієнтом і сервером.

Для запобігання несанкціонованому доступу та виявлення підозрілої активності система повинна вести журнал подій. У цьому журналі фіксуються усі дії користувачів, включаючи час входу, зміни у зверненнях, створення та видалення даних. Це дозволить проводити аудит безпеки та аналіз інцидентів у разі їх виникнення.

Особливу увагу слід приділити захисту персональних даних клієнтів відповідно до вимог чинного законодавства України. Дані, що зберігаються у базі (ПІБ, контактна інформація тощо), повинні бути захищені від несанкціонованого доступу як ззовні, так і з боку внутрішніх користувачів, які не мають відповідних прав.

Додатковим заходом є впровадження автоматичного завершення сесії у випадку бездіяльності користувача протягом визначеного часу. Завдяки цьому можливо уникнути ситуацій, коли система залишається відкритою на робочому місці без нагляду.

Комплексна реалізація вказаних вимог забезпечуватиме високий рівень захисту даних, безпечну експлуатацію модуля в умовах підприємства та відповідність сучасним стандартам інформаційної безпеки.

3.2 Обґрунтування мети і критеріїв ефективності об'єкта розробки

Необхідність у створенні модуля для обліку та аналізу гарантійних звернень виникла внаслідок потреби покращити якість обслуговування клієнтів у сервісному підрозділі торговельної мережі побутової техніки. В умовах зростаючої конкуренції на ринку електроніки компанії змушені не лише підтримувати високу якість товару, а й постійно вдосконалювати роботу із зверненнями клієнтів. Швидке реагування, можливість аналітики і прийняття рішень на основі даних стають ключовими чинниками конкурентоспроможності. У цьому контексті розробка окремого програмного модуля виглядає цілком виправданою як з технічної, так і з економічної точки зору.

Головна мета полягає у створенні зручного засобу автоматизації, що дозволить зменшити час обробки звернень, оптимізувати роботу з інформацією, покращити загальну аналітику та, як наслідок, підвищити рівень задоволеності клієнтів. Така автоматизація також повинна допомогти зменшити навантаження на персонал і зробити процеси закупівлі більш обґрунтованими.

Для того щоб реально оцінити, наскільки впровадження модуля буде ефективним, доцільно визначити чіткі показники. Один із найважливіших – це середній час обробки одного звернення до й після впровадження системи. Він дає змогу оцінити, наскільки швидше почав працювати сервісний відділ. Ще один показник – кількість повторних звернень з одного й того ж приводу, яка має зменшитися, якщо система допомагатиме вчасно та якісно реагувати на звернення.

Ефективність роботи співробітників також має зрости: автоматизація рутинних дій дає змогу одному спеціалісту опрацювати більшу кількість звернень протягом зміни. Не менш важливим є й те, що система допоможе

уникати помилок під час заповнення форм – завдяки обов’язковим полям і шаблонам даних, що значно підвищить точність введеної інформації.

Очікується також поліпшення в напрямі формування звітності. Система повинна скоротити час на побудову статистичних графіків і таблиць, що, у свою чергу, дозволить оперативніше реагувати на виявлені проблеми. До критеріїв аналітичної якості можна віднести і здатність точно виявляти моделі або бренди з підвищеним рівнем повернень – це важливо для ухвалення закупівельних рішень.

Окремо варто врахувати враження самих користувачів модуля. Комфортність роботи з інтерфейсом, логіка навігації та загальна зручність користування впливають на бажання працювати із системою та її реальну ефективність. Крім того, критично важливим є надійне зберігання інформації. Система має забезпечувати захист усіх записів, регулярне резервне копіювання та можливість відновлення у разі збоїв.

У підсумку, від впровадження цього модуля очікується низка позитивних ефектів. По-перше, покращиться рівень сервісу, що сприятиме зростанню довіри клієнтів. По-друге, керівництво зможе приймати рішення, спираючись на конкретні аналітичні дані, а не лише на інтуїцію. По-третє, загальна організація процесів стане більш оптимізованою, що дозволить заощадити час і ресурси.

Отже, запропонований модуль є не просто технічним компонентом, а дієвим інструментом управління, що здатен значно підвищити ефективність роботи сервісного відділу й поліпшити результати діяльності підприємства загалом.

4 ОПИС АРХІТЕКТУРИ ОБ'ЄКТА РОЗРОБКИ НА РІВНІ ФУНКЦІЙ

4.1. Загальна архітектура системи

Розроблена система орієнтована на автоматизацію процесів, пов'язаних з обліком та обробкою звернень, що надходять у межах гарантійного обслуговування в мережі магазинів побутової техніки. Архітектура побудована на основі модульного підходу, що значно полегшує обслуговування ПЗ, його масштабування та модернізацію у майбутньому.

Система структурована за трирівневою логікою, що охоплює такі складові:

– презентаційний рівень відповідає за взаємодію з користувачем. Цей рівень реалізований за допомогою бібліотеки Tkinter, яка надає інструменти для створення графічного інтерфейсу. Через візуальні форми користувачі виконують ключові дії: входять у систему, додають звернення, змінюють статуси, переглядають журнали подій або аналітику. Інтерфейс побудований так, щоб кожна дія мала свою окрему логічну форму;

– логічний рівень містить основну логіку, що керує внутрішніми процесами. На цьому рівні відбувається обробка введених даних, перевірка правильності заповнення, оновлення статусів, аналіз звернень, створення аналітичних висновків і формування рекомендацій. Сюди ж входить обрахунок строків виконання, генерування PDF-звітів та ведення історії змін;

– рівень доступу до даних відповідає за роботу з базою даних. Він забезпечує підключення до системи управління базами даних (СУБД) MySQL, виконує SQL-запити та взаємодіє з основними таблицями (claims, users, products, status_logs). Цей рівень повністю відокремлений від користувацького інтерфейсу, що дозволяє в майбутньому легко змінити систему керування базами даних без значного переписування коду.

Короткий опис основних модулів системи:

- модуль авторизації та управління доступом визначає права доступу користувачів згідно з їх ролями;
- модуль реєстрації звернень забезпечує збір необхідної інформації для фіксації нових гарантійних звернень;
- модуль перегляду і оновлення статусів дозволяє контролювати процес обробки звернень, змінювати статуси та додавати необхідні коментарі;
- модуль контролю термінів визначає звернення, які потребують термінового реагування чи прострочені;
- модуль аналітики та звітності генерує статистичні звіти та рекомендації на основі накопичених даних про звернення.

Обраний підхід до побудови архітектури дозволить чітко розділити відповідальність між компонентами, полегшить технічне обслуговування й адаптацію системи під нові задачі, що виникатимуть у процесі її експлуатації.

4.2 Схема функціональної взаємодії

Для кращого розуміння того, як працює модуль «Облік та аналіз гарантійних звернень», була створена схема його функціональної взаємодії. Як засіб візуалізації обрано побудову діаграми потоків даних Data Flow Diagram (DFD) у межах концептуальної моделі TO-BE, яка відображає бажаний або цільовий стан процесів після впровадження системи.

На рисунку 4.1 представлено загальну контекстну діаграму системи. Вона ілюструє модуль «Облік та аналіз гарантійних звернень» як єдиний процес, що взаємодіє з зовнішнім середовищем – працівниками сервісного відділу, керівниками відділу постачання, а також нормативними джерелами. Вхідні потоки включають дані про клієнтів, товари, звернення, нормативні

терміни, політики гарантії. Вихідними результатами є зареєстровані звернення, оновлені статуси, статистичні звіти, аналітика та рекомендації щодо закупівель.

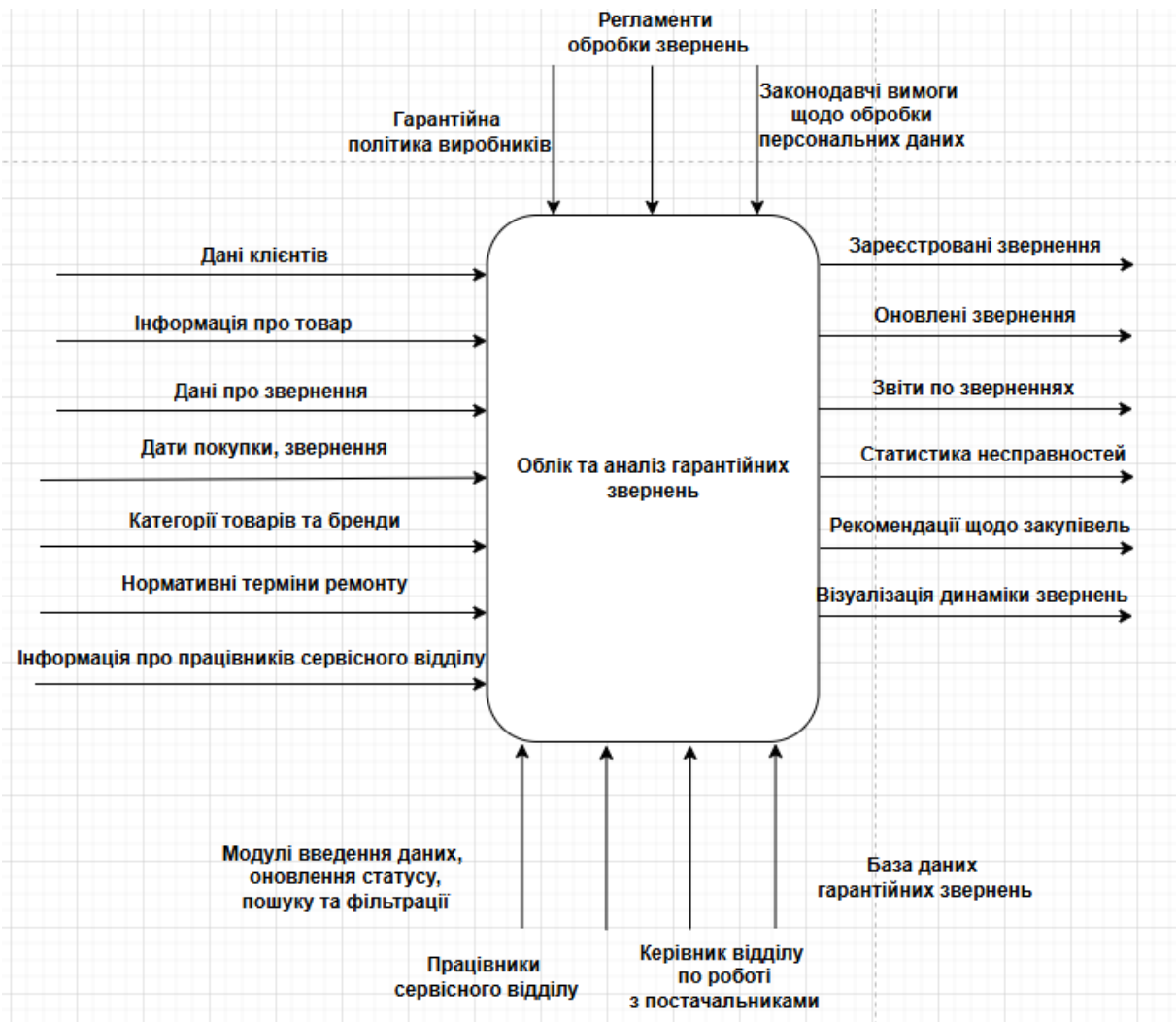


Рисунок 4.1 – Схема процесу «Облік та аналіз гарантійних звернень» (контексна діаграма), модель TO-BE

Основні зовнішні елементи взаємодії:

- працівники сервісного відділу здійснюють введення та оновлення інформації;
- керівник відділу по роботі з постачальниками використовує результати аналітики;

- нормативні джерела регламентують обробку даних та політику обслуговування;
- база даних зберігає інформацію про звернення, клієнтів, продукти тощо.

Для деталізації логіки функціонування було побудовано діаграму першого рівня (рисунок 4.2), яка розкриває внутрішню структуру модуля та взаємодію між підпроцесами. Кожен з них виконує окрему функцію, а потоки даних між ними забезпечують логічну послідовність.

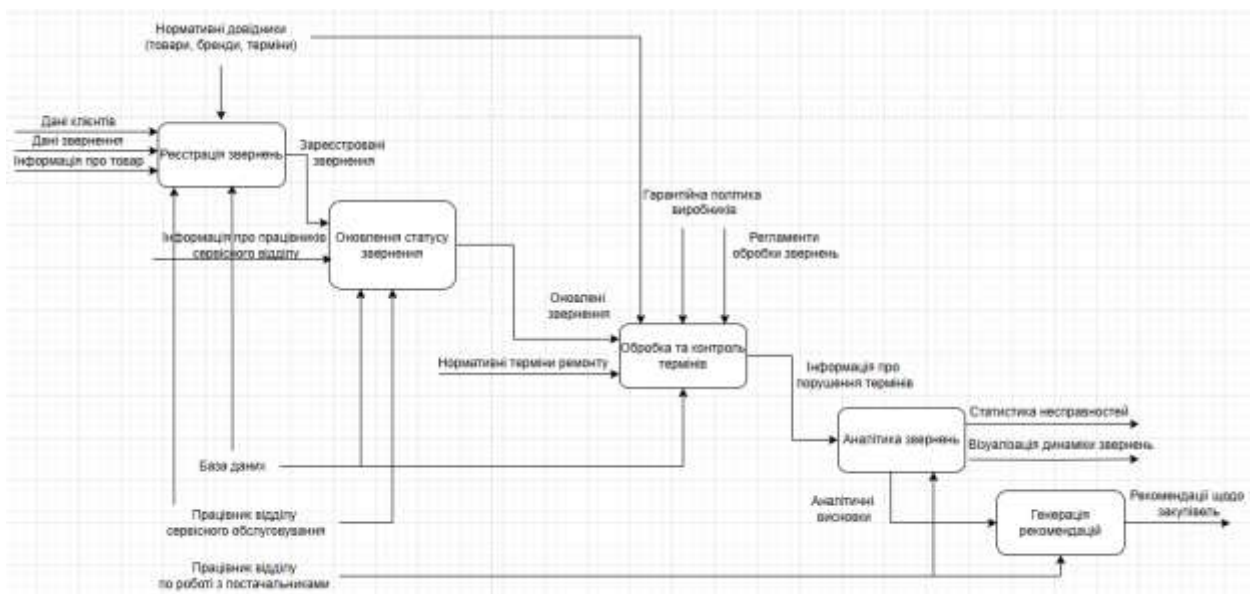


Рисунок 4.2 – Схема процесу «Облік та аналіз гарантійних звернень» (діаграма декомпозиції першого рівня), модель ТО-ВЕ

Деталізація діаграми першого рівня дозволяє глибше зрозуміти внутрішню структуру модуля та побудову логіки взаємодії між його складовими елементами. У межах цієї діаграми виділено основні підпроцеси, які забезпечують повноцінне функціонування системи обліку й аналізу гарантійних звернень.

Усе починається з реєстрації звернення: працівник сервісного відділу вносить до системи дані про клієнта, модель техніки, дату придбання та

звернення, опис несправності. Ця інформація фіксується в базі даних, що дозволяє надалі відслідковувати історію обробки звернення.

Наступним етапом стає оновлення статусу. Після реєстрації звернення отримує відповідний етап обробки — наприклад, його можуть перевести в статус «у роботі» або «завершено». Водночас додаються коментарі та вказується відповідальна особа. Усі ці зміни системно фіксуються разом із часовими позначками.

Паралельно з цим система постійно контролює дотримання встановлених термінів виконання. Вона автоматично перевіряє строки перебування звернень у кожному статусі, і якщо виникають порушення — вони відображаються у відповідній формі.

На основі зібраних звернень формується аналітика: модуль аналізує частоту дефектів, періоди між покупкою та поломкою, а також загальні тенденції. Результати обробки виводяться у формі графіків і звітів, які допомагають краще зрозуміти загальну ситуацію з обслуговуванням техніки.

У фіналі, аналітичні дані трансформуються в практичні рекомендації для менеджменту. Наприклад, система може запропонувати скоригувати структуру замовлень, відмовитися від окремих моделей або посилити контроль за постачальниками. Усі ці висновки оформлюються у звіти, придатні для подальшої роботи з партнерами.

5 РОЗРОБКА ТА ОБҐРУНТУВАННЯ ЕЛЕМЕНТІВ ІНФОРМАЦІЙНОЇ ЗАБЕЗПЕЧУЮЧОЇ СИСТЕМИ

5.1. Обґрунтування вибору системи управління базами даних (СУБД)

Для ефективної роботи ІС, яка має забезпечити автоматизований облік і аналіз гарантійних звернень, необхідно обрати надійний інструмент для зберігання, організації та обробки структурованих даних. У межах цього проєкту одним із ключових етапів стало визначення найбільш доцільної системи управління базами даних, яка б відповідала як технічним, так і експлуатаційним вимогам.

З урахуванням поставлених завдань, характеристик майбутньої системи та середовища розробки, вибір було зупинено на СУБД MySQL. Це рішення ґрунтується на ряді аргументів, які роблять її оптимальним варіантом для реалізації даного програмного продукту.

По-перше, MySQL поширюється за відкритою ліцензією (General Public License, GNU GPL), що дозволяє вільно використовувати її для навчальних і дослідницьких цілей, а також у невеликих комерційних системах без додаткових витрат. Це особливо актуально у випадку розробки в рамках студентського чи внутрішнього проєкту підприємства.

По-друге, система добре інтегрується з мовою Python, що активно використовувалася під час реалізації прикладної логіки. Підтримка через бібліотеки `mysql-connector`, `PyMySQL`, `SQLAlchemy` спрощує роботу з базою, дозволяє легко налаштувати підключення та реалізувати обробку запитів на стороні програми.

Крім того, MySQL відома своєю стабільністю та продуктивністю при роботі з невеликими та середніми обсягами даних, що відповідає масштабу запланованої системи. За потреби можливо реалізувати масштабування без суттєвої перебудови архітектури.

Важливою перевагою також є підтримка транзакцій і забезпечення цілісності даних. Завдяки використанню рушія InnoDB, MySQL дотримується властивостей Atomicity Consistency Isolation Durability (ACID), що дозволяє запобігти втраті або пошкодженню інформації при збої або несподіваному завершенні сеансу.

У процесі вибору також було розглянуто інші популярні СУБД, які потенційно могли бути використані у даному проєкті. Таблиця 5.1 ілюструє порівняння різних СУБД.

Таблиця 5.1 – Порівняння характеристик альтернативних СУБД

Критерій	MySQL	SQLite	PostgreSQL
Тип	Серверна	Вбудована (файлова)	Серверна
Ліцензія	GPL (безкоштовна)	Public Domain (вільна)	Open Source (безкоштовна)
Підтримка транзакцій	Так (InnoDB)	Частково	Так (повна підтримка ACID)
Сумісність з Python	Висока	Висока	Висока
Простота налаштування	Середня	Дуже проста	Складна
Масштабованість	Висока	Обмежена	Дуже висока
Підтримка складних типів	Обмежена	Обмежена	Розширена (JSONB, масиви тощо)
Відповідність проєкту	Так	лише для локальних проєктів	надлишкова функціональність для цієї задачі

На основі порівняння стає зрозуміло, що MySQL в повній мірі відповідає потребам розроблюваної системи: вона підтримує необхідний

функціонал, легко інтегрується з мовою програмування Python, працює стабільно та не потребує зайвих ресурсів для налаштування й супроводу. Це робить MySQL практичним і виваженим вибором для реалізації проєкту.

5.2. Логічна й фізична модель даних системи обліку та аналізу гарантійних звернень

З метою структурування інформації, що обробляється системою, було здійснено логічне та фізичне моделювання бази даних. Це дозволить на етапі проєктування чітко окреслити, які саме об'єкти предметної області підлягають зберіганню, якими характеристиками вони володіють і як пов'язані між собою.

На рисунку 5.1 логічну модель побудовано з урахуванням специфіки

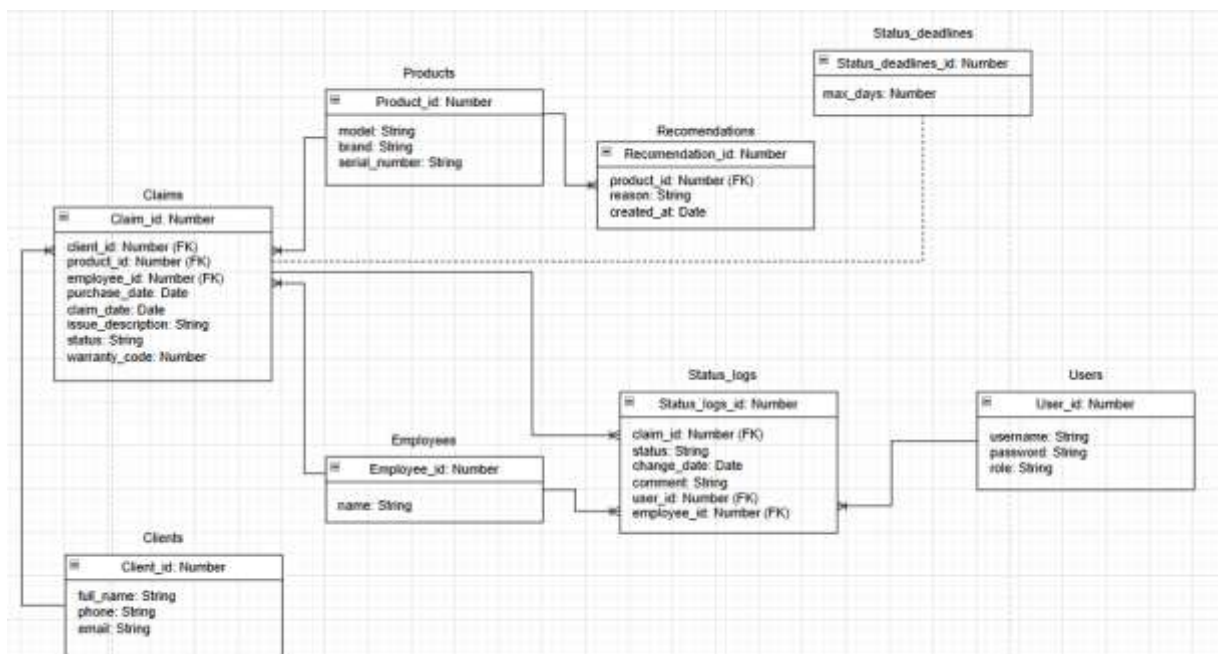


Рисунок 5.1 – Логічна модель даних системи обліку та аналізу гарантійних звернень у вигляді ER-діаграми

функціонування сервісного відділу мережі побутової техніки. Для візуалізації структури даних застосовано нотацію IDEF1X, яка дає змогу наочно представити сутності, їх атрибути та логічні зв'язки між ними.

Логічна модель даних охоплює ключові сутності системи: користувачі, клієнти, звернення, статуси, товари, працівники, журнал змін, нормативні терміни й рекомендації у вигляді Entity-Relationship model (ER) діаграми. Модель передбачає використання зовнішніх ключів для реалізації зв'язків типу «один до багатьох» між сутностями.

Логічна модель даних описує структуру ІС обліку та аналізу гарантійних звернень, у якій кожна таблиця виконує функцію певної сутності в межах бізнес-процесів сервісного обслуговування. Центральне місце займає таблиця claims, що акумулює інформацію про самі звернення клієнтів. Вона є зв'язуючою ланкою між користувачами системи, клієнтами, товарами, працівниками сервісного відділу та журналом змін.

Кожне звернення прив'язане до конкретного клієнта, дані якого зберігаються в таблиці clients. У ній фіксується повне ім'я, контактний номер та електронна пошта – все необхідне для зворотного зв'язку. Також звернення пов'язане з конкретним товаром, що дозволяє відстежити проблеми за моделями, брендами чи серійними номерами. Ця інформація надалі використовується для аналітики й генерації рекомендацій щодо закупівель.

У структурі звернення передбачено поле для відповідального працівника, що вказує на таблицю employees. Такий підхід дає змогу закріплювати звернення за конкретними виконавцями, формувати звітність за персоналом і відстежувати навантаження між членами команди.

Процес зміни статусу звернення супроводжується автоматичним фіксуванням усіх дій у таблиці status_logs. Тут зберігаються відомості про новий статус, дату зміни, коментар, а також ідентифікаційні дані користувача (users), що вніс зміни, та працівника, який їх реалізував. Це забезпечує повний контроль за перебігом обробки кожного окремого

звернення.

Контроль дотримання строків обробки звернень реалізується через таблицю `status_deadlines`, де задано граничні терміни перебування у кожному статусі. Система, орієнтуючись на ці параметри, виявляє звернення, що потребують негайної реакції або перевищили допустимі часові межі.

Ключовим елементом після накопичення звернень є аналітичний блок, який представлений таблицею `recommendations`. На основі аналізу проблемних товарів або повторюваних звернень система формує висновки – наприклад, про доцільність обмеження закупівель певної моделі. Ці рекомендації прив'язані до конкретних товарів і можуть бути використані для прийняття стратегічних управлінських рішень.

Загалом логічна структура бази даних повністю охоплює життєвий цикл гарантійного звернення – від його первинної реєстрації до фінального аналізу та управлінських дій. Це забезпечує не лише цілісність і повноту збереження даних, а й сприяє ефективності внутрішніх процесів сервісного обслуговування.

Фізична модель сформована на основі логічної й відображає реалізацію таблиць та атрибутів у СУБД MySQL. Усі поля містять уточнені типи даних, обмеження цілісності та довжини, що оптимізовані для практичного використання. Фізична модель представлена на рисунку 5.2.

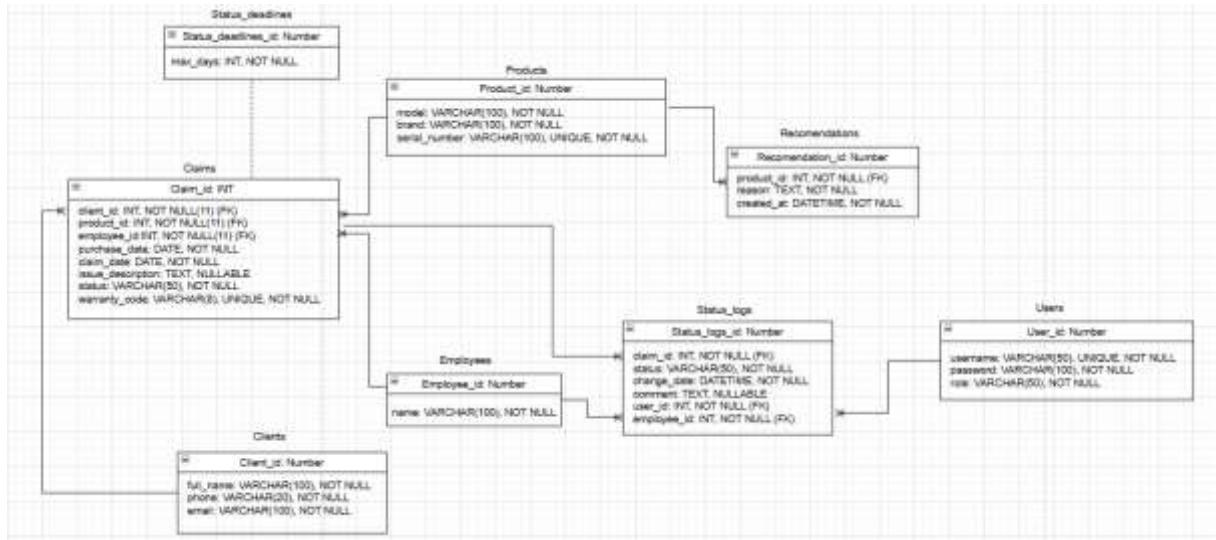


Рисунок 5.2 – Фізична модель даних системи обліку та аналізу гарантійних звернень у вигляді ER-діаграми

5.3 Створення бази даних системи обліку та аналізу гарантійних звернень на платформі СУБД MySQL

Для реалізації системи обліку та аналізу гарантійних звернень було розроблено реляційну базу даних, створену на платформі СУБД MySQL [13]. У якості інструменту моделювання структури бази даних використовувався графічний редактор ER-схем у середовищі phpMyAdmin, що дозволило створити наочну реалізацію логічної моделі у вигляді фізичної структури з чіткими обмеженнями цілісності.

На рисунку 5.3 наведено реалізовану EER-схему, що включає всі основні сутності системи, їх атрибути, типи даних, а також зовнішні ключі для зв'язку між таблицями.

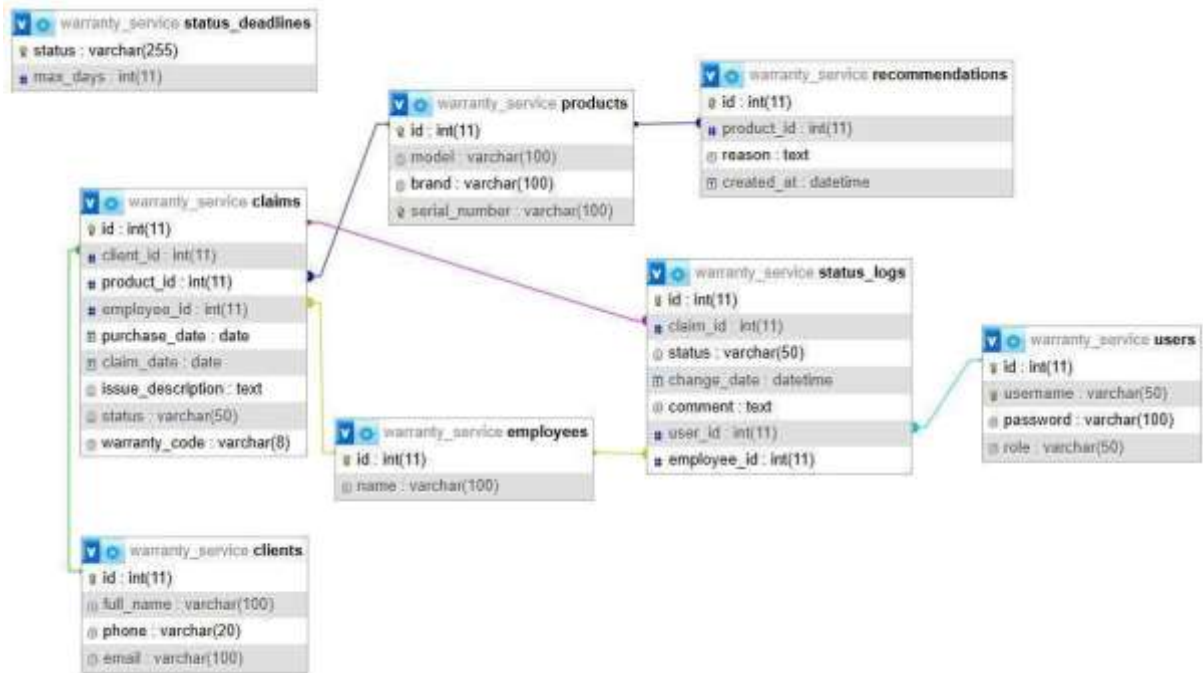


Рисунок 5.3 – Реалізована структура бази даних у вигляді ER-схеми

Створення бази даних передбачає використання зовнішніх ключів, які забезпечують дотримання посилальної цілісності та логічну пов'язаність між сутностями. Основні зв'язки між таблицями реалізовані наступним чином:

- `claims` пов'язана з таблицями `clients`, `products` та `employees` через зовнішні ключі `client_id`, `product_id` та `employee_id` відповідно. Це дозволяє для кожного звернення зберігати інформацію про клієнта, об'єкт обслуговування та відповідального виконавця;

- `status_logs` має зовнішні ключі на `claims`, `users` та `employees`. Така побудова дозволяє зберігати повну історію змін статусу звернення, вказуючи як на користувача, що ініціював зміну, так і на виконавця, який реалізував дію;

- `status_deadlines` не має явних зовнішніх ключів, однак логічно пов'язується з полем `status` у таблиці `claims`, забезпечуючи контроль строків обробки згідно з нормативами;

- `recommendations` пов'язана з таблицею `products` через поле `product_id`, що дозволяє системі генерувати рекомендації на основі історії

звернень по конкретних моделях.

Під час роботи над базою даних вирішено автоматизувати створення первинних ключів застосувавши механізм автоінкременту. Це значно спростило введення нових записів і зняло потребу вручну прописувати унікальні значення. Для деяких полів, зокрема імені користувача, гарантійного коду та серійного номера, встановлено перевірку на унікальність – щоб уникнути повторів. Типи даних підібрані залежно від того, яку саме інформацію потрібно зберігати: дати – у форматі DATE, текст – як VARCHAR або TEXT, а числа – через INT. Також обмеження NOT NULL для того, щоб важливі поля не залишались порожніми.

Така структура дозволяє ефективно вирішувати ключові завдання системи: реєстрацію звернень, контроль термінів обробки, фіксацію історії змін, авторизацію користувачів та формування аналітичних звітів. Крім того, структура бази даних є масштабованою та може бути адаптована для розширення функціональності в майбутньому (наприклад, додавання нових ролей, типів звернень або інтеграції з постачальниками).

Завдяки використанню СУБД MySQL та інструментів phpMyAdmin розроблена база даних є не лише функціонально повною, але й зручною для подальшого адміністрування, резервного копіювання, а також візуального контролю структури.

6 РОЗРОБКА ТА ОБҐРУНТУВАННЯ ЕЛЕМЕНТІВ ПРОГРАМНОЇ ЗАБЕЗПЕЧУЮЧОЇ СИСТЕМИ

6.1 Загальний опис програмної системи

Розроблена ІС призначена для автоматизації процесу обліку та аналізу гарантійних звернень у галузі роздрібно́ї торгівлі технікою. Її впровадження спрямоване на оптимізацію фіксації звернень, контроль за їх розглядом, ведення історії змін статусів, аналітичну обробку накопичених даних, а також підтримку прийняття рішень щодо постачання продукції.

Система охоплює весь життєвий цикл гарантійного звернення – від його реєстрації до формування висновків на основі статистичних спостережень. Для зручності роботи користувачів інтерфейс реалізовано у простій та зрозумілій формі, що дозволяє працювати з системою навіть без спеціальної технічної підготовки.

Функціонал системи охоплює такі основні можливості:

- авторизація користувача для обмеження доступу до функціоналу;
- реєстрація звернень клієнтів із зазначенням проблеми та деталей товару;
- оновлення статусів звернень із фіксацією змін у журналі;
- перевірка строків обробки звернень відповідно до нормативів;
- аналітика типових проблем, товарів і частоти звернень;
- формування рекомендацій щодо оптимізації закупівель на основі статистики.

Програмне рішення передбачене для роботи в умовах сервісного центру або офісу компанії. Інтерфейс, побудований із використанням бібліотеки Tkinter, дозволяє співробітникам швидко виконувати всі необхідні дії без втручання ІТ-спеціалістів.

Зберігання та обробка даних реалізовані за допомогою реляційної СУБД MySQL, що забезпечує структуровану організацію інформації та

цілісність між пов'язаними таблицями. Для доступу до бази даних використовується бібліотека `mysql.connector` [15], що дозволяє виконувати запити безпосередньо з програмного коду на Python.

Систему реалізовано у середовищі Windows 10+, з використанням Python 3.11. Графічна частина програми не потребує зовнішніх залежностей, оскільки Tkinter є стандартною бібліотекою мови Python. База даних функціонує на сервері з розгорнутою інстанцією MySQL 8.0.

Загальна архітектура програмної системи включає три ключові рівні:

- клієнтська частина (інтерфейс) реалізована через Tkinter для взаємодії користувача з функціоналом;
- прикладна логіка (модулі) набір `.py` файлів, кожен з яких відповідає за окремий функціональний блок;
- серверна частина (СУБД) база даних MySQL із пов'язаними таблицями `claims`, `clients`, `products`, `employees`, `status_logs`, `status_deadlines` та іншими.

Така побудова дозволяє підтримувати зрозумілу та впорядковану структуру програмного коду, забезпечує стабільність роботи, а також відкриває можливість подальшої модернізації системи – зокрема, шляхом переходу на веб-інтерфейс або розширення функціоналу за допомогою мобільного додатку..

6.2 Архітектура програмної частини

Архітектура програмної частини розробленої ІС побудована на модульному принципі, за якого кожен файл відповідає за окрему функцію. Такий підхід забезпечує логічну структурування коду, полегшує супровід програми, спрощує тестування та дозволяє легко розширювати функціональність у разі потреби.

Структура програмного коду представлена наступними модулями:

- `auth_login.py` – відповідає за процес авторизації користувача. При успішному введенні логіну та пароля відкриває головне вікно системи;
- `register_claim.py` – реалізує механізм реєстрації нових звернень, дозволяючи вказувати дані клієнта, товару та опис проблеми;
- `view_claims.py` – забезпечує перегляд усіх зареєстрованих звернень у табличному вигляді з можливістю фільтрації;
- `update_status.py` – надає інтерфейс для зміни статусу звернення, збереження коментарів та фіксації змін у журналі;
- `view_logs.py` – реалізує перегляд історії змін статусів із зазначенням дати, користувача та коментарів;
- `analytics.py` – виконує обробку збережених даних для формування зведеної статистики звернень за періодами, категоріями товарів тощо;
- `check_deadlines.py` – автоматично перевіряє, чи не перевищено терміни перебування звернень у поточному статусі згідно з нормативами, і виводить відповідні сповіщення;
- `recomendations.py` – на основі аналітичної інформації формує управлінські рекомендації, що стосуються доцільності закупівлі певних товарів.

Файли програми взаємодіють між собою через обробку подій, виклик функцій або спільну роботу з базою даних. Наприклад, модуль `update_status.py` після внесення змін записує подію до таблиці журналу, а `analytics.py` зчитує цю інформацію для візуалізації тенденцій.

Приклад взаємодії між модулями:

- користувач вводить логін у вікні авторизації (`auth_login.py`);
- після авторизації відкривається інтерфейс з кнопками, які дають змогу створити нове звернення (`register_claim.py`) або переглянути існуючі (`view_claims.py`);
- користувач змінює статус звернення за допомогою модуля `update_status.py`, зміна фіксується в `status_logs`;

- періодично виконується автоматична перевірка термінів обробки звернень (`check_deadlines.py`);
- дані з таблиць `claims`, `status_logs`, `products` обробляються у `analytics.py`, який формує графіки та таблиці;
- `recomendations.py` використовує результати аналітики для формування списку товарів, які рекомендовано виключити або додати до наступної закупівлі.

Запропонована архітектура дозволяє гнучко масштабувати систему, додаючи нові функціональні модулі (наприклад, генерацію актів виконаних робіт, надсилання електронних повідомлень або підтримку веб-інтерфейсу), без суттєвого перегляду вже реалізованої логіки. Такий підхід забезпечує надійну основу для довготривалої експлуатації та адаптації системи під змінні вимоги бізнесу.

У межах такого розширення можливе подальше впровадження модулів прогнозування кількості звернень, що базуються на математичних моделях. Зокрема, неоднорідний пуассонівський процес доцільно використовувати для моделювання часової нерівномірності звернень, що дозволяє підвищити точність планування ресурсів та логістики обслуговування [11].

6.3 Структура програми

Для забезпечення логічної організації програмного коду розробленої системи реалізовано чітку структуру директорій та модулів. Кожен функціональний блок винесено в окремий файл, що сприяє спрощенню супроводу, тестування й масштабування системи.

На рисунку 6.1 наведено структуру розроблюваного проекту.

```

project_root/
|
├─ auth_login.py      # авторизація користувачів
├─ register_claim.py  # створення нового звернення
├─ view_claims.py     # перегляд звернень
├─ update_status.py   # зміна статусів звернень
├─ view_logs.py       # перегляд журналу змін
├─ check_deadlines.py # автоматичне оновлення термінів
├─ analytics.py       # аналітика звернень
├─ recomendations.py  # генерація рекомендацій

```

Рисунок 6.1 – Структура проєкту

Основними використовуваними бібліотеками у системі є:

- tkinter [14] для побудови графічного інтерфейсу;
- mysql.connector [15] – для підключення до СУБД;
- fpdf [16] для створення PDF-звітів;
- datetime для обробка дати і часу;
- os, ttk, messagebox допоміжні засоби інтерфейсу.

Основні приклади реалізації функціональних можливостей програмної системи наведено у лістингах 6.1 – 6.6. Вони ілюструють базові принципи взаємодії користувача з інтерфейсом, збереження звернень до бази даних, оновлення статусу та виконання аналітики.

Перед початком роботи користувач повинен пройти перевірку автентичності. Для цього реалізовано функцію перевірки логіну та пароля. Вона порівнює введені дані з базою користувачів та відкриває основне меню у разі успішного входу.

Лістинг 6.1 – Функція перевірки логіну та пароля

```

def login():
    username = entry_username.get()
    password = entry_password.get()
    cursor.execute("SELECT * FROM users WHERE username=%s AND
password=%s", (username, password))

```

```

result = cursor.fetchone()
if result:
    root.destroy()
    open_main_menu()
else:
    messagebox.showerror("Помилка", "Невірний логін або
пароль")

```

Далі реалізовано механізм реєстрації нового звернення. Користувач вводить основну інформацію про клієнта, товар, дату покупки та суть проблеми. Введені дані проходять перевірку та записуються в кілька таблиць бази даних.

Лістинг 6.2 – Збереження нового звернення у базу даних

```

def save_claim():
    full_name = entry_name.get()
    phone = entry_phone.get()
    email = entry_email.get()
    model = entry_model.get()
    brand = entry_brand.get()
    serial = entry_serial.get()
    purchase_date = entry_purchase.get()
    claim_date = entry_claim.get()
    warranty_code = entry_warranty_code.get()
    description = entry_description.get("1.0", tk.END).strip()

    if not all([full_name, model, brand, serial,
purchase_date, claim_date, warranty_code]):
        messagebox.showerror("Помилка", "Будь ласка, заповніть
усі обов'язкові поля.")
        return

    if not warranty_code.isdigit() or len(warranty_code) != 8:
        messagebox.showerror("Помилка", "Гарантійний номер

```

```

повинен складатися з 8 цифр.")
    return

try:
    conn = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="warranty_service"
    )
    cursor = conn.cursor()

    cursor.execute("INSERT INTO clients (full_name, phone,
email) VALUES (%s, %s, %s)", (full_name, phone, email))
    client_id = cursor.lastrowid

    cursor.execute("INSERT INTO products (model, brand,
serial_number) VALUES (%s, %s, %s)", (model, brand, serial))
    product_id = cursor.lastrowid

    initial_status = "Очікує призначення"
    cursor.execute("""
        INSERT INTO claims (client_id, product_id,
claim_date, purchase_date, issue_description, status,
employee_id, warranty_code)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
        """, (client_id, product_id, claim_date,
purchase_date, description, initial_status, None,
warranty_code))

    conn.commit()
    messagebox.showinfo("Успіх", "Звернення успішно
збережено.")
    window.destroy()

except mysql.connector.Error as err:

```

```

        messagebox.showerror("Помилка БД", f"Сталася помилка
при збереженні: {err}")

```

```

finally:
    if conn.is_connected():
        cursor.close()
        conn.close()

```

У наведеному нижче прикладі реалізовано функцію `show_claims()`, яка відповідає за виведення зареєстрованих звернень у графічному інтерфейсі у форматі таблиці. Таблиця створюється у новому вікні за допомогою елемента `tk.Treeview`, що дозволяє користувачеві зручно переглядати наявні записи з бази даних. Функція `show_claims()` створює окреме вікно, у якому за допомогою `tk.Treeview` відображаються звернення з бази даних. Передбачено пошук за ключовими полями (ПІБ, модель, бренд), а також фільтрація за статусом. Додатково реалізовано можливість видалення запису через подвійне натискання та підтвердження. Така реалізація забезпечує гнучкий інструмент для менеджера сервісного центру.

Лістинг 6.3 – Функція відображення таблиці звернень із фільтрацією та обробкою подій

```

def show_claims():
    def sort_by_column(col_name):
        global current_sort_column, current_sort_reverse
        index = columns.index(col_name)
        if current_sort_column == index:
            current_sort_reverse = not current_sort_reverse
        else:
            current_sort_reverse = False
            current_sort_column = index
        filter_claims(tree, status_filter, search_entry,
sort_column=index, reverse=current_sort_reverse)

```

```

def on_double_click(event):
    selected = tree.selection()
    if selected:
        values = tree.item(selected[0], "values")
        claim_id = values[0]
        confirm = messagebox.askyesno("Підтвердження",
f"Ви дійсно бажаєте видалити звернення ID {claim_id}?")
        if confirm:
            delete_claim(claim_id)
            filter_claims(tree, status_filter,
search_entry, current_sort_column, current_sort_reverse)

view_window = tk.Tk()
view_window.title("Перегляд звернень")
view_window.geometry("1250x550")
view_window.configure(bg="#f0f0f0")

filter_frame = tk.Frame(view_window, bg="#f0f0f0")
filter_frame.pack(fill="x", padx=10, pady=10)

tk.Label(filter_frame, text="Статус:", bg="#f0f0f0",
font=("Arial", 10)).pack(side="left")
status_filter = ttk.Combobox(filter_frame, values=[
    "Усі", "Очікує призначення", "Чекає діагностики", "У
роботі", "Чекає запчастини", "Готовий до видачі"
], width=25)
status_filter.set("Усі")
status_filter.pack(side="left", padx=5)

tk.Label(filter_frame, text="Пошук (ПІВ / модель /
бренд):", bg="#f0f0f0", font=("Arial", 10)).pack(side="left",
padx=(20, 5))
search_entry = tk.Entry(filter_frame, width=30)
search_entry.pack(side="left", padx=5)

apply_btn = tk.Button(filter_frame, text="Застосувати",

```

```

bg="#007acc", fg="white", font=("Arial", 10, "bold"),
                                command=lambda: filter_claims(tree,
status_filter, search_entry, current_sort_column,
current_sort_reverse))
    apply_btn.pack(side="left", padx=10)

    global columns
    columns = ["ID", "ПІВ", "Модель", "Бренд", "Серійний
номер", "Дата покупки", "Дата звернення", "Опис", "Статус",
"Гарантійний номер"]
    tree = ttk.Treeview(view_window, columns=columns,
show="headings")

    for col in columns:
        tree.heading(col, text=col, command=lambda _col=col:
sort_by_column(_col))
        tree.column(col, width=120, anchor="center")

    style = ttk.Style()
    style.configure("Treeview.Heading", font=("Arial", 10,
"bold"))
    style.configure("Treeview", font=("Arial", 10),
rowheight=25)

    tree.pack(fill="both", expand=True, padx=10, pady=10)
    tree.bind("<Double-1>", on_double_click)

    filter_claims(tree, status_filter, search_entry)
    view_window.mainloop()

show_claims()

```

У наступному прикладі наведено реалізацію оновлення статусу звернення та призначення відповідального працівника. Функція також фіксує зміни у журналі статусів.

Лістинг 6.4 – Оновлення статусу звернення

```

def save_status():
    new_status = status_var.get()
    comment = comment_text.get("1.0", tk.END).strip()
    employee_name = employee_entry.get().strip()

    if not new_status or not employee_name:
        messagebox.showwarning("Увага", "Оберіть статус і
введіть відповідального співробітника.")
        return

    try:
        with open("user_session.txt", "r") as f:
            username = f.read().strip()

            conn = mysql.connector.connect(**db_config)
            cursor = conn.cursor()

            cursor.execute("SELECT id FROM users WHERE
username=%s", (username,))
            user_result = cursor.fetchone()
            user_id = user_result[0] if user_result else None

            cursor.execute("SELECT id FROM employees WHERE
name=%s", (employee_name,))
            emp_result = cursor.fetchone()
            if emp_result:
                employee_id = emp_result[0]
            else:
                cursor.execute("INSERT INTO employees (name)
VALUES (%s)", (employee_name,))
                employee_id = cursor.lastrowid

            cursor.execute("UPDATE claims SET status=%s,
employee_id=%s WHERE id=%s",

```

```

        (new_status, employee_id, claim_id))

    cursor.execute("""
        INSERT INTO status_logs (claim_id, status,
change_date, comment, user_id)
        VALUES (%s, %s, NOW(), %s, %s)
    """, (claim_id, new_status, comment, user_id))

    conn.commit()
    conn.close()

    messagebox.showinfo("Успіх", "Статус і відповідальний
оновлені.")
    edit_win.destroy()
    load_claims(current_sort_column, current_sort_reverse)
except Exception as e:
    messagebox.showerror("Помилка", str(e))

```

Для підтримки управлінських рішень система включає аналітичний модуль. Він будує графіки динаміки звернень, частоти за брендами і моделями. Як зазначається в [12], аналітика гарантійних звернень дає змогу ідентифікувати проблемні вузли продукції, які потребують доопрацювання.

Лістинг 6.5 – Побудова аналітики за зверненнями

```

def run_analytics():
    conn = mysql.connector.connect(**db_config)
    query = """
        SELECT products.brand, products.model,
claims.claim_date
        FROM claims
        JOIN products ON claims.product_id = products.id
    """
    df = pd.read_sql(query, conn)
    conn.close()

```

```

df["claim_date"] = pd.to_datetime(df["claim_date"],
errors="coerce")
df.dropna(subset=["claim_date"], inplace=True)

brand_counts = df["brand"].value_counts().head(10)
model_counts = df.groupby(["brand",
"model"]).size().sort_values(ascending=False).head(10)
monthly_counts = df.resample("M", on="claim_date").size()

plt.figure(figsize=(10, 5))
sns.barplot(x=brand_counts.index, y=brand_counts.values)
plt.title("ТОП 10 брендів за кількістю звернень")
plt.xlabel("Бренд")
plt.ylabel("Кількість звернень")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

Останній приклад демонструє формування рекомендацій щодо закупівель. Він здійснює агрегацію звернень за останні місяці, обраховує середній час до поломки та генерує PDF-звіт.

Лістинг 6.6 – Генерація рекомендацій щодо закупівель

```

def generate_procurement_recommendations(threshold=10):
    conn = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="warranty_service"
    )

    query = """
        SELECT products.brand, products.model,

```

```

        COUNT(*) AS claim_count,
        AVG(DATEDIFF(claims.claim_date,
claims.purchase_date)) AS avg_days_to_claim
    FROM claims
    JOIN products ON claims.product_id = products.id
    GROUP BY products.brand, products.model
    ORDER BY claim_count DESC
"""

df = pd.read_sql(query, conn)
conn.close()

problematic_models = df[df["claim_count"] >=
threshold].copy()

if problematic_models.empty:
    print("Усі моделі мають прийнятний рівень звернень.")
else:
    print(problematic_models.head())

```

6.4 Екранні форми інтерфейсу користувача (UI)

Інтерфейс користувача в розробленій системі реалізований на основі бібліотеки Tkinter [14] та відповідає принципам доступності, логічної структурованості та зручності використання. Кожне вікно оформлено у відповідному стилі з урахуванням взаємозв'язку між елементами, акцентами на важливі дії та дотриманням єдиної кольорової гами.

На рисунку 6.2 наведено блок-схему переходів між формами.



Рисунок 6.2 – Блок-схема переходів між формами

На рисунках 6.3 – 6.12 наведено основні форми інтерфейсу користувача.

The image shows a user login window titled 'Авторизація' (Authorization). The window contains the following elements:

- Window title: Авторизація
- Window controls: Minimize, Maximize, Close buttons.
- Section header: **Вхід до системи** (System login)
- Input fields: 'Логін' (Login) and 'Пароль' (Password).
- Submit button: 'Увійти' (Login).

Рисунок 6.3 – Вікно авторизації користувача

Ця форма дозволяє користувачу ввести ім'я та пароль для доступу до функціоналу системи. У разі помилкових даних відображається

повідомлення про помилку. Кнопка «Увійти» активує перевірку введених даних у базі.

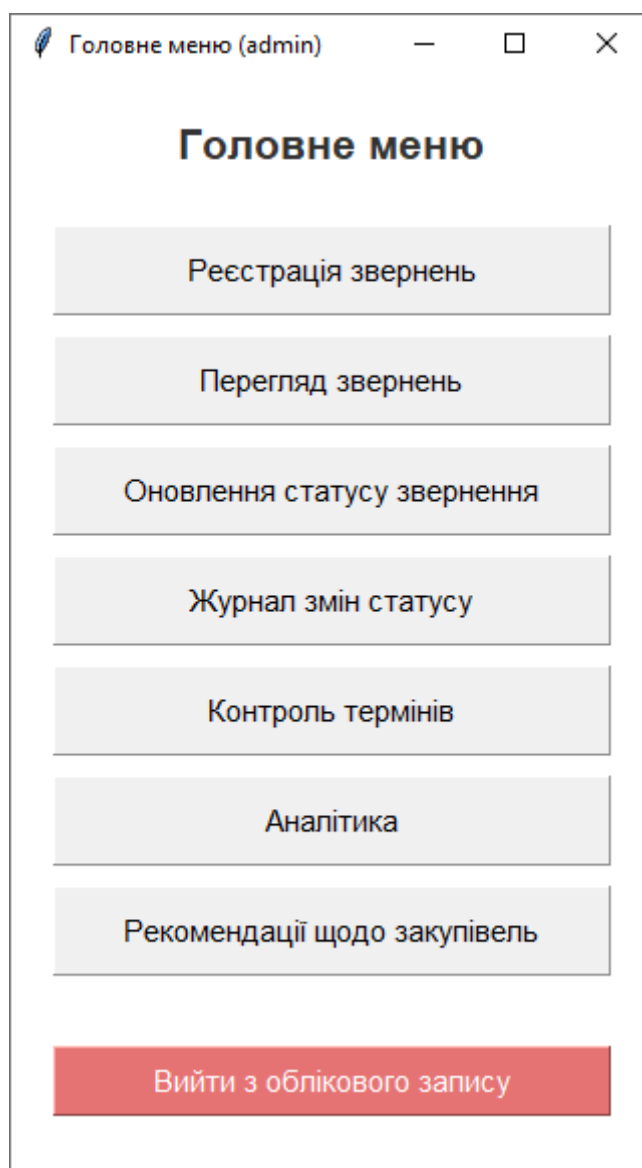


Рисунок 6.4 – Головне меню програми

Після успішної авторизації користувач потрапляє до головного меню, з якого здійснюється навігація до основних функцій програми. У вікні розміщено кнопки для реєстрації нового звернення, перегляду наявних звернень, зміни статусів, аналізу даних, формування PDF-звіту, перегляду журналу змін та виходу з програми.

Реєстрація звернення

ПІБ клієнта *	Іванов Іван Іванович
Контактний телефон	0968979231
Email	ivanov@gmail.com
Модель товару *	A-31
Бренд *	Samsung
Серійний номер *	a324-43
Дата покупки * (рррр-мм-дд)	2024-03-24
Дата звернення * (рррр-мм-дд)	2025-01-02
Гарантійний номер * (8 цифр)	12345678

Опис несправності

Мерехтить екран

Зберегти звернення

Рисунок 6.5 – Форма реєстрації нового звернення

Форма містить поля для введення ПІБ клієнта, моделі та бренду техніки, серійного номера, дати покупки, короткого опису проблеми. Поля є обов'язковими для заповнення, що унеможливорює збереження неповного запису.

Перегляд звернень

Статус: Усі Пошук (ПІБ / модель / бренд): Застосувати

ID	ПІБ	Модель	Бренд	Серійний номер	Дата покупки	Дата зверненн	Опис	Статус	Гарантія
957	Іванов Іван Іванович	A-31	Samsung	a324-43	2024-03-24	2025-01-02	Мерехтять екран	Очікує призначенн	12345678
956	Аніта Адаменко	Galaxy Note	Samsung	wt461824fW	2024-02-29	2024-11-12	Телефон не зард	Чекає діагностики	69752507
955	пан Орхон Гузій	iPhone 12	Apple	XK381646BS	2023-06-09	2024-10-05	Швидко розряд	Чекає запчастини	84911114
954	Михайлина Журавел	Legion 5	Lenovo	0417891Ne	2025-03-24	2025-05-04	Не запускається	Чекає запчастини	16206230
953	Аніта Заруба	Elmu x360	HP	hB463282DN	2023-11-08	2024-02-26	Екран мерехтит	Очікує призначенн	97313501
952	Мирослав Гаєвський	iPhone 12	Apple	PH855608nx	2024-09-15	2025-05-01	Проблеми з під	Чекає діагностики	67440561
951	Марта Чабан	Mi 11 Lite	Xiaomi	nY196387bq	2024-03-03	2024-11-29	Проблеми з GP	Чекає запчастини	16946787
950	Роксолана Колісник	Redmi Note	Xiaomi	Fp350455XJ	2023-07-13	2024-12-08	Мимовільні пер	Чекає діагностики	00936556
949	Людмила Козаченк	iPhone SE	Apple	ur716156mf	2024-07-22	2024-11-20	Проблеми з під	У роботі	79788853
948	Єлизавета Кибіцало	Mi 11 Lite	Xiaomi	xO475344fh	2024-03-03	2024-05-24	Проблеми з GP	Чекає запчастини	23726552
947	Валентина Перебийні	Galaxy Note	Samsung	KJ615648xD	2024-07-07	2024-10-24	Проблеми зі зву	Очікує призначенн	33478158
946	добродій Йосип Біл	iPhone 13	Apple	cF180018md	2024-05-17	2024-07-05	Проблеми з під	Чекає діагностики	06722533
945	Миколай Петренко	Poco X3	Xiaomi	GY934635GS	2024-01-29	2025-03-21	Проблеми з GP	Чекає запчастини	40604775
944	Тетяна Чабан	Legion 5	Lenovo	PZ477980hr	2024-02-09	2024-05-28	Синій екран см	Очікує призначенн	33963168
943	Ажаква Приска Євген	Redmi Note	Xiaomi	zV909296Ke	2024-07-21	2025-01-20	Сенсор не реаг	Очікує призначенн	09365898
942	панна Данка Цюпа	iPhone 13	Apple	qf1384833qk	2024-02-14	2025-05-13	Швидко розряд	Очікує призначенн	81961497
941	Оксеня Юхименко	Galaxy A52	Samsung	oO173184yp	2023-10-15	2023-12-05	Проблеми зі зву	Чекає запчастини	29778403
940	Варвара Чаленко	Gram 16	LG	gj596893Jm	2024-05-19	2025-02-03	Телефон не вм	Чекає запчастини	86096564

Рисунок 6.6 – Вікно перегляду звернень

У вікні реалізовано сортування, фільтрацію за статусом, а також пошук за ключовими словами. Подвійний клік по рядку дозволяє видалити звернення після підтвердження. Візуалізація реалізована з використанням Treeview для зручного табличного подання даних.

Журнал змін статусів

Журнал змін статусів звернень

ID звернення	Користувач	Статус	Дата зміни	Коментар
957	admin	У роботі	2025-06-04 07:07:01	
791	admin	У роботі	2025-05-30 08:59:54	іфва

Рисунок 6.7 – Журнал змін статусів звернень

Цей екран відображає історію змін статусів, включно з датою, попереднім і новим статусом, ПІБ відповідального та коментарем. Журнал оновлюється автоматично при кожній зміні статусу.

На рисунку 6.8 зображено вікно програмного модуля, який

призначений для контролю строків обробки звернень. У цьому вікні виводиться список записів, у яких зафіксовано перевищення допустимого терміну перебування звернення в конкретному статусі. У табличній формі відображено основну інформацію: номер звернення, дані клієнта, модель пристрою, чинний статус, кількість днів, що минули з моменту останньої зміни, та граничне значення, визначене згідно з відповідним нормативом із таблиці `status_deadlines`.



ID	ПІБ	Модель	Статус	Днів минуло	Ліміт
458	Гаврило Якимчук	iPhone 13	Чекає запчастини	457	7
459	Елисей Матяш	Legion 5	Чекає діагностики	82	2
460	Василина Вдовенко	IdeaPad 3	У роботі	521	7
461	Розалія Носенко	ThinkPad X1	У роботі	133	7
464	Наталія Щербань	ThinkPad X1	Чекає запчастини	343	7
466	Альбіна Дем'янюк	Redmi Note 11	Чекає запчастини	11	7
467	Софія Рябовіл	Legion 5	Чекає діагностики	193	2
468	Юстим Кириленко	Galaxy S21	У роботі	37	7
469	Роксолана Заруба	ThinkPad X1	Чекає запчастини	108	7
470	Яків Назаренко	Envy x360	Очікує призначення	19	1
471	Марина Ванченко	Legion 5	Чекає запчастини	9	7
474	Лесь Лазаренко	UltraGear	Очікує призначення	348	1
479	Стефан Деряжний	Velvet	У роботі	177	7
480	Омелян Перебийніс	Velvet	Чекає запчастини	11	7
481	Христина Вернигора	IdeaPad 3	Очікує призначення	132	1
482	Анастасія Артимішин	Gram 16	Чекає діагностики	23	2
483	Адаменко Демид Віталійо	OMEN 17	Очікує призначення	332	1
484	Аліна Яремко	iPhone 12	Чекає запчастини	42	7

Рисунок 6.8 – Вікно контролю термінів виконання

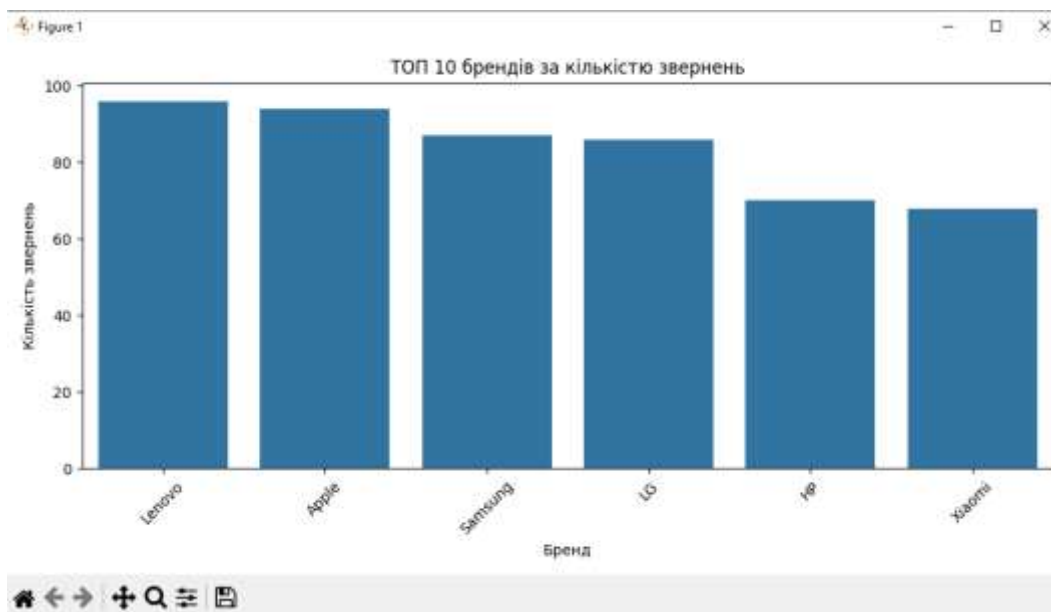


Рисунок 6.9 – Перше вікно аналітики

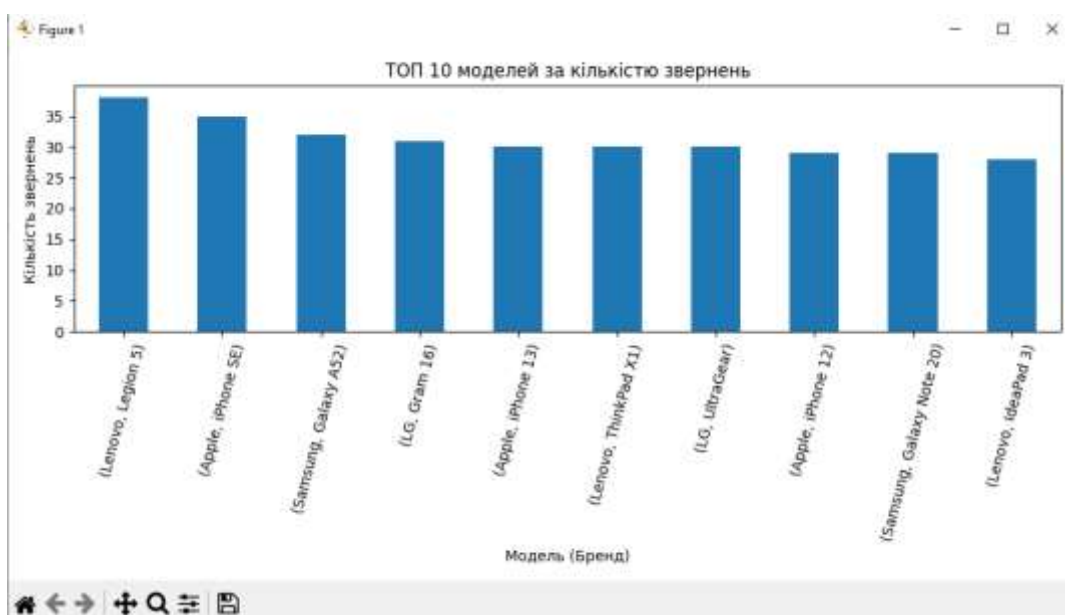


Рисунок 6.10 – Друге вікно аналітики

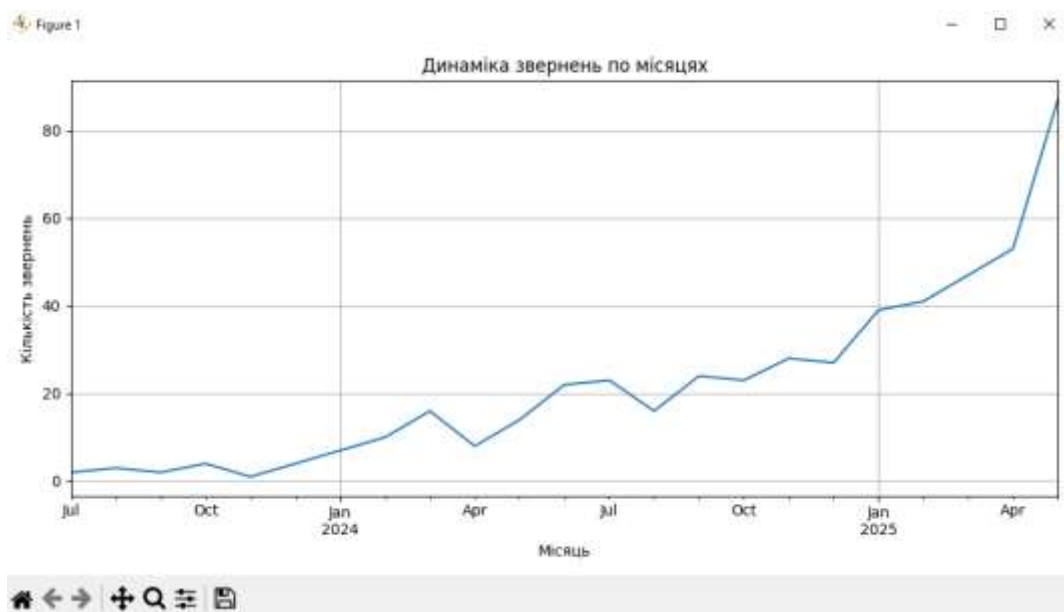


Рисунок 6.11 – Трете вікно аналітики

Модуль аналітики надає зведені графіки частоти звернень за категоріями товарів, брендами, датами звернень та статусами. Усі графіки динамічно формуються на основі поточних даних із бази. Додатково відображаються кількісні значення під кожною діаграмою для зручності аналізу.

Рекомендації щодо закупівель

Аналіз моделей з високою кількістю звернень

Бренд	Модель	К-сть звернень	Середній час до звернення (днів)	Перше звернення	Останнє звернення	Звернення за останні 3 міс
Lenovo	Legion 5	38	176.6	2023-08-15	2025-05-30	21.0
Apple	iPhone SE	35	212.7	2023-08-01	2025-05-29	10.0
Samsung	Galaxy A52	32	161.5	2023-12-05	2025-05-21	10.0
LG	Gram 16	31	181.4	2023-07-20	2025-05-29	11.0
Apple	iPhone 13	30	242.4	2024-02-17	2025-05-27	13.0
Lenovo	ThinkPad X1	30	206.6	2023-11-16	2025-05-29	11.0

Аналітичний висновок

ТОП-5 моделей з найбільшою кількістю гарантійних звернень:

- Lenovo Legion 5**
 - Звернень: 38
 - Середній час до звернення: 176.6 днів
 - Звернень за останні 3 міс: 21.0
 - Найчастіші проблеми: Проблеми з клавіатурою (17), Не запускається після сну (12), Синій екран смерті (9)
- Apple iPhone SE**
 - Звернень: 35
 - Середній час до звернення: 212.7 днів
 - Звернень за останні 3 міс: 10.0
 - Найчастіші проблеми: Швидко розряджається батарея (17), Не працює Face ID (10), Проблеми з підключенням до Wi-Fi (8)
- Samsung Galaxy A52**
 - Звернень: 32
 - Середній час до звернення: 161.5 днів
 - Звернень за останні 3 міс: 10.0
 - Найчастіші проблеми: Телефон не заряджається (11), Проблеми зі звуком при розмові (11), Тріщина на дисплеї після падіння (10)
- LG Gram 16**
 - Звернень: 31
 - Середній час до звернення: 181.4 днів
 - Звернень за останні 3 міс: 11.0
 - Найчастіші проблеми: Зображення слізоване або мерехтить (12), Телефон не вмикається (10), Немає зображення на екрані (9)
- Apple iPhone 13**
 - Звернень: 30
 - Середній час до звернення: 242.4 днів
 - Звернень за останні 3 міс: 13.0
 - Найчастіші проблеми: Проблеми з підключенням до Wi-Fi (13), Не працює Face ID (12), Швидко розряджається батарея (5)

Рекомендація
 Уважно переглянути доцільність подальших закупівель моделей із ТОПу.
 Найбільші ризики — ті, що мають високий рівень звернень протягом перших 30 днів після покупки та повторені типи несправності. Рекомендується обмежити закупівлі або вимагати кращих гарантій від постачальників.

Зберегти звіт у PDF

Рисунок 6.12 – Рекомендації щодо закупівель

Цей модуль надає рекомендації на основі аналізу моделей з високою кількістю звернень. Після натискання відповідної кнопки формується документ PDF із усіма поточними зверненнями. У звіті враховано застосовані фільтри та сортування, що забезпечує актуальність даних на момент генерації. Цей документ можна зберегти для подальшого друку та передачі відділу по роботі з постачальниками.

ВИСНОВКИ

У процесі підготовки кваліфікаційної роботи було розроблено рішення для обліку й аналізу гарантійних звернень, адаптоване під потреби торгівельної мережі з продажу побутової техніки. Система створена як відповідь на виклики сервісного обслуговування, де затримки й помилки можуть призвести до втрат. Основна мета – не лише автоматизація рутинних дій, а й надання інструментів для аналітичної підтримки управлінських рішень.

Перед реалізацією було вивчено типові сценарії обробки звернень і сформовано вимоги. Архітектуру спроектовано за модульним принципом де кожен компонент має чітке призначення, що спрощує супровід і дає змогу розширювати систему в майбутньому.

Для зберігання й обробки даних використано СУБД MySQL із логічною та фізичною моделлю. Клієнтська частина написана мовою Python, а інтерфейс реалізовано через бібліотеку Tkinter – це лаконічно та достатньо функціонально. Програма охоплює модулі для реєстрації звернень, зміни їх статусів, аналітики, формування рекомендацій і контролю термінів.

Впроваджене рішення дозволяє зменшити адміністративне навантаження, знизити ризики, уникнути дублювання інформації й покращити звітність. За потреби систему можна адаптувати під інші компанії або доповнити мобільним доступом і електронним документообігом.

Результати опубліковано у вигляді тез доповідей «Крикун М.І. Автоматизація обробки гарантійних звернень у мережі роздрібної торгівлі технікою. 11th International scientific and practical conference “Current trends in scientific research development” (June 5–7, 2025). BoScience Publisher, 2025. С. 265–267».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Sankaran S., Dhamija A., Tripathi V. Digital transformation in aftersales and warranty management. *Algorithms*, Vol. 18, No. 4, 2023. P. 231.
2. Zoho CRM. Cloud Software Suite for Businesses URL: <https://www.zoho.com/> (дата звернення: 9.05.2025).
3. Freshdesk. Agentic AI platform for modern customer service. URL: <https://freshdesk.com> (дата звернення: 9.05.2025).
4. Google Таблиці. Довідкові матеріали. URL: <https://support.google.com/docs/answer/6000292?hl=uk> (дата звернення: 9.05.2025).
5. Google Таблиці замість CRM? Безплатне та зручне рішення для малого та середнього бізнесу. URL: <https://speka.media/google-tablici-zamist-crm-bezplatne-ta-zrucne-risennya-dlya-malogo-ta-serednyogo-biznesu-vm65d9> (дата звернення: 9.05.2025).
6. ServiceDesk Plus. ManageEngine ServiceDesk Plus Store. URL: <https://store.manageengine.com/service-desk/> (дата звернення: 9.05.2025).
7. Laudon K. C., Laudon J. P. *Management information systems: Managing the digital firm*. London: Pearson Education, 2021. 624 p.
8. Buttle F., Maklan S. *Customer relationship management: Concepts and technologies*. London: Routledge, 2019. 442 p.
9. Global Automotive Warranty Survey Report. URL: https://www.bearingpoint.com/files/autowarrantyreport_final_web.pdf (дата звернення: 9.05.2025).
10. Dmello S. Transforming CRM warranty management with advanced analytics and AI. *Algorithms*, Vol. 18, No. 4, 2023. P. 231.
11. Shanbhag D., Farias V. F., McCormick T. S. Warranty claims modelling. Cornell ORIE Technical Report, No. 2024-01, 2014. P. 1–26.

12. Detering Consulting. What is Warranty Analytics & Why Do You Need It? URL: <https://www.deteringconsulting.com/blog/what-is-warranty-analytics-why-do-you-need-it> (дата звернення: 10.05.2025).

13. MySQL : Developer Zone URL: <https://dev.mysql.com> (дата звернення: 10.05.2025).

14. Python interface to Tcl/Tk – Python 3.13.5 documentation. URL: <https://docs.python.org/uk/3.13/library/tkinter.html> (дата звернення: 12.05.2025).

15. MySQL Connector/Python Developer Guide. URL: <https://dev.mysql.com/doc/connector-python/en/> (дата звернення: 12.05.2025).

16. FPDF2. URL: <https://pyfpdf.github.io/fpdf2/> (дата звернення: 13.05.2025).

17. Закон України «Про захист персональних даних» № 2297-VI від 1 червня 2010 р. URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 1.06.2025).

18. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. К.: ДП «УкрНДНЦ», 2016. 20 с.

19. ДСТУ 3008:2015. Державний стандарт України. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення. К.: ДП «УкрНДНЦ», 2016. 31 с.