

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка
до кваліфікаційної роботи бакалавра
за спеціальністю 123 «Комп'ютерна інженерія»

на тему: КАЛЕНДАР-ДОВІДНИК ГАРАНТІЙНОГО ТА ТЕХНІЧНОГО
ОБСЛУГОВУВАННЯ ЕЛЕКТРОННИХ ТА ПОБУТОВИХ ПРИСТРОЇВ

Проектував	_____	Б.Є. Басов
Керівник роботи	_____	С. В. Сьомочкина
Нормоконтроль	_____	Д. І. Кузнєцов
Завідувач кафедри	_____	А. І. Купін

Кривий Ріг
2025

Криворізький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних систем та мереж

Ступінь вищої освіти
Спеціальність

бакалавр
123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри, голова циклової комісії

_____ А. І. Купін

“ ____ ” _____ 20__ року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Басов Богдан Євгенович

(прізвище, ім'я, по батькові)

1. Тема роботи Календар-довідник гарантійного та технічного
обслуговування електронних та побутових пристроїв

керівник роботи к.т.н., доц. Сьомочкина Світлана Володимирівна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ ____ ” _____ 20__ року № ____

2. Строк подання студентом роботи 05.06.2025

3. Вихідні дані до роботи є інформація, яку вводить користувач: назва пристрою, бренд, дата покупки, гарантійний термін і файл чека. Система автоматично визначає статус пристрою (активний, зламаний або прострочений) на основі розрахованої дати завершення гарантії. Для брендів використовується довідник сервісних центрів. Усі дані зберігаються в базі PostgreSQL і застосовуються для нагадувань, аналітики та експорту.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) У записці розглянуто етапи створення програмного забезпечення для обліку гарантійного та технічного обслуговування пристроїв. Розроблено структуру бази даних, описано логіку роботи користувацького інтерфейсу, реалізовано взаємодію з PostgreSQL через Npgsql, а також функції нагадування, фільтрації та експорту. Окрему увагу приділено побудові статистики з використанням графіків, перевірці коректності даних та логіки обробки гарантійних випадків.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Представлено графічні матеріали, що ілюструють структуру бази даних, роботу користувацького інтерфейсу та аналітичні можливості програми. Включено схему таблиць, знімки вікон (додавання, перегляд, сортування), повідомлення системи та графіки з даними про поломки та середню тривалість гарантії.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Збір та аналіз інформації, формулювання мети та завдань	22.04 – 26.04.2025	Теор. блок
2	Аналіз ринку аналогічних програм, визначення вимог до ПЗ	27.04 – 30.04.2025	Розділ 1
3	Розробка структури БД, проектування основних сутностей	01.05 – 04.05.2025	Початок розділу 2
4	Розробка основної логіки: додавання пристрою, база даних, облік	05.05 – 10.05.2025	Реалізація
5	Створення інтерфейсу користувача, робота з чеками	11.05 – 14.05.2025	Реалізація UI
6	Додавання фільтрації, статусів, зламаних пристроїв	15.05 – 18.05.2025	Логіка
7	Реалізація статистики, аналітики та експорту в Excel	19.05 – 22.05.2025	Завершення функцій
8	Тестування застосунку, відлагодження, перевірка БД	23.05 – 25.05.2025	Тестування
9	Написання висновків, оформлення звітності, графіків	26.05 – 29.05.2025	Документація
10	Перевірка повного тексту, правки, підготовка до захисту	30.05 – 04.06.2025	Фінальний етап

Студент _____
(підпис) (прізвище та ініціали)Керівник роботи _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 76 сторінок, 35 рисунків, 4 таблиць, 20 використаних джерел.

Проект складається з трьох розділів.

У першому розділі проаналізовано проблеми з веденням обліку гарантій на пристрої та обґрунтовано потребу у створенні окремого десктопного застосунку для зручного контролю гарантій, чеків і сервісного обслуговування.

Другий розділ описує реалізацію програми: інтерфейси введення, перегляду, аналітики, збереження до бази PostgreSQL, обробку подій і використання графіків та експорту до Excel.

У третьому розділі проведено тестування. Підтверджено, що всі функції працюють коректно: зміна статусів, фільтрація, сповіщення, аналітика — все реалізовано з урахуванням зручності для користувача.

ГАРАНТІЯ, ПРИСТРІЙ, БАЗА ДАНИХ, СЕРВІСНИЙ ЦЕНТР, ФІЛЬТРАЦІЯ, C#, POSTGRESQL, WINDOWS FORMS, СТАТИСТИКА, EXCEL.

					КНУ.РБ.123.25.01.Р			
Змн.	Арк.	№ документа	Підпис	Дата	РЕФЕРАТ	Літера	Аркуш	Аркушів
Розробив	Басов							
Перевірив	Купін							
Н.контроль	Кузнецов					KI-21		
Затвердив	Купін							

Explanatory note: 76 pages, 35 figures, 4 tables, 20 sources used.

The project consists of three sections.

The first section analyzes the problems with maintaining warranty records on devices and justifies the need to create a separate desktop application for convenient control of warranties, checks and service.

The second section describes the implementation of the program: input, viewing, analytics interfaces, saving to the PostgreSQL database, event processing and use of graphs and export to Excel.

The third section tests. It is confirmed that all functions work correctly: changing statuses, filtering, notifications, analytics - everything is implemented taking into account user convenience.

WARRANTY, DEVICE, DATABASE, SERVICE CENTER, FILTERING, C#, POSTGRESQL, WINDOWS FORMS, STATISTICS, EXCEL.

					КНУ.РБ.123.25.01. Р	Арк.
	Арк.	№ документа	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	7
ВСТУП	8
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ОБЛІКУ ГАРАНТІЙНОГО ТА ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ ЕЛЕКТРОННИХ ТА ПОБУТОВИХ ПРИСТРОЇВ	10
1.1 Проблематика обліку гарантійних термінів та технічного обслуговування пристроїв	10
1.2 Аналіз існуючих програмних рішень	12
1.3 Огляд існуючих мобільних додатків та сервісів для гарантійного обліку.....	16
1.4 Вимоги до ефективного інструменту обліку гарантійного обслуговування.....	22
Висновки.....	27
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	29
2.1 Загальна архітектура системи	29
2.2 Функціональні та нефункціональні вимоги	30
2.3. Структура бази даних:	32
2.4 Опис графічного інтерфейсу користувача Form1.cs	35
2.5 Опис графічного інтерфейсу користувача FormDevicesList.cs	38
2.6 Опис графічного інтерфейсу користувача FormAnalytics.cs.....	43
2.7 Алгоритми роботи системи по фрагментам коду	49
Висновки.....	61
3. ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА МОЖЛИВОСТІ РОЗВИТКУ СИСТЕМИ	63
3.1 Результати тестування	63
Висновки.....	73
ВИСНОВКИ.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	75

					КНУ.РБ.123.25.01.3			
Змн.	Арк.	№ документа	Підпис	Дата	ЗМІСТ			
Розробив	Басов							
Перевірів	Купін							
Н.контроль	Кузнецов							
Затвердив	Купін				KI-21			

ПЕРЕЛІК СКОРОЧЕНЬ

JSON – JavaScript Object Notation (формат обміну даними)

CSV – Comma-Separated Values (формат збереження табличних даних)

SQL – Structured Query Language (структурована мова запитів)

PostgreSQL – система керування реляційними базами даних з відкритим кодом

C# – мова програмування, що використовується в розробці застосунку

WinForms – Windows Forms (технологія створення графічного інтерфейсу у Windows)

Excel – табличний редактор, у який експортуються дані

UX/UI – загальна аббревіатура для дизайну користувацького досвіду та інтерфейсу

ВСТУП

У сучасному світі електронні та побутові пристрої займають важливе місце в житті кожної людини. Їх кількість у домогосподарствах і підприємствах постійно зростає, що створює нові виклики в організації обліку їх гарантійного та технічного обслуговування. Більшість користувачів зіштовхуються з проблемою втрати важливої інформації про дату придбання пристрою, тривалість гарантії або умови технічної підтримки. Це часто призводить до ситуацій, коли можливість безкоштовного ремонту або заміни пристрою в межах гарантії втрачається через несвоєчасне звернення або брак необхідних документів.

Попри наявність окремих рішень на ринку, більшість із них орієнтовані на корпоративних користувачів або мають обмежений функціонал, який не враховує потреби пересічного споживача. У більшості випадків такі системи є занадто складними у використанні або не підтримують автоматичного нагадування про закінчення гарантії та відсутність інтеграції з інформацією про сервісні центри. Тому існує потреба у створенні інтуїтивно зрозумілого інструменту, що дозволить користувачам ефективно контролювати стан своїх пристроїв.

Темою даної кваліфікаційної роботи є "Календар-довідник гарантійного та технічного обслуговування електронних та побутових пристроїв".

Мета роботи полягає у розробці простого та зручного програмного забезпечення для ведення обліку пристроїв, автоматичного контролю термінів гарантійного обслуговування та надання рекомендацій щодо можливих дій у разі виявлення несправностей.

Для досягнення поставленої мети у межах роботи передбачається проаналізувати існуючі рішення у цій галузі, визначити їх недоліки та сформулювати вимоги до власного продукту. На основі проведеного аналізу буде спроектовано інформаційну систему, яка дозволить користувачеві додавати дані про пристрої, зберігати інформацію про дату покупки, гарантійний термін та наявність чека, а також своєчасно отримувати сповіщення про наближення завершення гарантії.

					КНУ.РБ.123.25.01.ВС			
Змн.	Арк.	№ документа	Підпис	Дата	ВСТУП	Літера	Аркуш	Аркушів
Розробив	Басов							
Перевірив	Купін							
Н.контроль	Кузнецов					КІ-21		
Затвердив	Купін							

Особливістю запропонованого рішення стане також можливість фіксації стану пристрою, зокрема відмітка про його несправність з перевіркою чинності гарантії. Пристрій у разі поломки буде супроводжуватися рекомендаціями щодо звернення до відповідного сервісного центру, що значно полегшить процес технічного обслуговування для користувача.

Крім базового обліку, система передбачає створення аналітичних звітів у вигляді графіків, які дозволять оцінити надійність різних брендів на основі накопичених даних про пристрої та їх поломки. Додатково передбачається можливість експорту статистики у популярні формати файлів для подальшої обробки чи архівації.

Отже, розробка такого додатку дозволить користувачам своєчасно реагувати на закінчення гарантійних зобов'язань, мінімізувати ризики втрати права на безкоштовний ремонт і підвищити загальний рівень організації інформації про свої електронні та побутові пристрої.

					КНУ.РБ.123.25.01. ВС	Арк.
	Арк.	№ документа	Підпис	Дата		

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ОБЛІКУ ГАРАНТІЙНОГО ТА ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ ЕЛЕКТРОННИХ ТА ПОБУТОВИХ ПРИСТРОЇВ

1.1 Проблематика обліку гарантійних термінів та технічного обслуговування пристроїв

Сучасні реалії технологічного розвитку призвели до того, що у середньостатистичного користувача накопичується велика кількість електронних та побутових пристроїв: смартфони, ноутбуки, телевізори, пральні машини, кухонна техніка тощо. Практично кожен з цих пристроїв має гарантійний термін, протягом якого виробник або продавець зобов'язується безкоштовно усунути несправності, що виникли з вини виробника.

Однак на практиці облік гарантійного обслуговування часто залишається поза увагою користувача. Після покупки пристрою більшість споживачів зберігають чеки та гарантійні талони без систематизації або зовсім їх втрачають. При виникненні поломок часто виявляється, що знайти необхідні документи неможливо, або користувач дізнається про закінчення гарантії лише тоді, коли ремонт вже стає платним.

Крім того, складність полягає не тільки у зберіганні документів, але й у запам'ятовуванні строків дії гарантії, що особливо проблематично при великій кількості пристроїв у користуванні. У випадку бізнес-структур проблема ще більше загострюється, оскільки йдеться про десятки або сотні одиниць обладнання, кожне з яких потребує ретельного контролю стану гарантійних зобов'язань.

Системи ручного обліку, такі як записи у блокнотах або електронних таблицях, часто виявляються неефективними через відсутність нагадувань про закінчення гарантії та необхідності постійного оновлення інформації вручну. Відсутність автоматизації призводить до збільшення кількості пропущених термінів звернення до сервісних центрів, що несе додаткові фінансові витрати.

Ще одним важливим аспектом є питання своєчасної діагностики та ремонту пристроїв до закінчення гарантійного терміну. Без відповідної системи нагадувань користувачі можуть пропустити критичний момент, коли несправність можна було б усунути безкоштовно.

					КНУ.РБ.123.25.01.AIPCOГТ							
Змн.	Арк.	№ документа	Підпис	Дата	АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У СФЕРІ ОБЛІКУ ГАРАНТІЙНОГО ТА ТЕХНІЧНОГО				Літера	Аркуш	Аркушів	
Розробив	Басов											
Перевірив	Купін											
Н.контроль	Кузнецов											
Затвердив	Купін								KI-21			

Таким чином, проблема обліку гарантійного та технічного обслуговування пристроїв є актуальною як для окремих користувачів, так і для організацій. Її вирішення потребує створення спеціалізованого інструменту, який дозволяв би:

- централізовано зберігати інформацію про кожен пристрій,
- автоматично розраховувати дату завершення гарантії,
- своєчасно повідомляти користувача про необхідність звернення до сервісу,
- сприяти збереженню прав на безкоштовне обслуговування.

Розробка такого інструменту дозволить мінімізувати втрати, пов'язані з несвоєчасним ремонтом, покращити організацію обліку пристроїв та підвищити загальну ефективність технічного обслуговування в побуті та бізнесі.

Для кращого розуміння актуальності проблеми обліку гарантійних термінів, було проаналізовано статистичні дані з відкритих джерел. Згідно з дослідженням омніканального бренду Фокстрот, 28% покупців вважають наявність офіційної гарантії важливим чинником при виборі техніки

. За даними Державної служби України з питань безпечності харчових продуктів та захисту споживачів, лише 54% скарг на побутову техніку задовольняються через наявність відповідних документів

. Крім того, 75% інтернет-магазинів не видають фіскальні чеки при продажу складної побутової техніки, що значно ускладнює процес гарантійного обслуговування

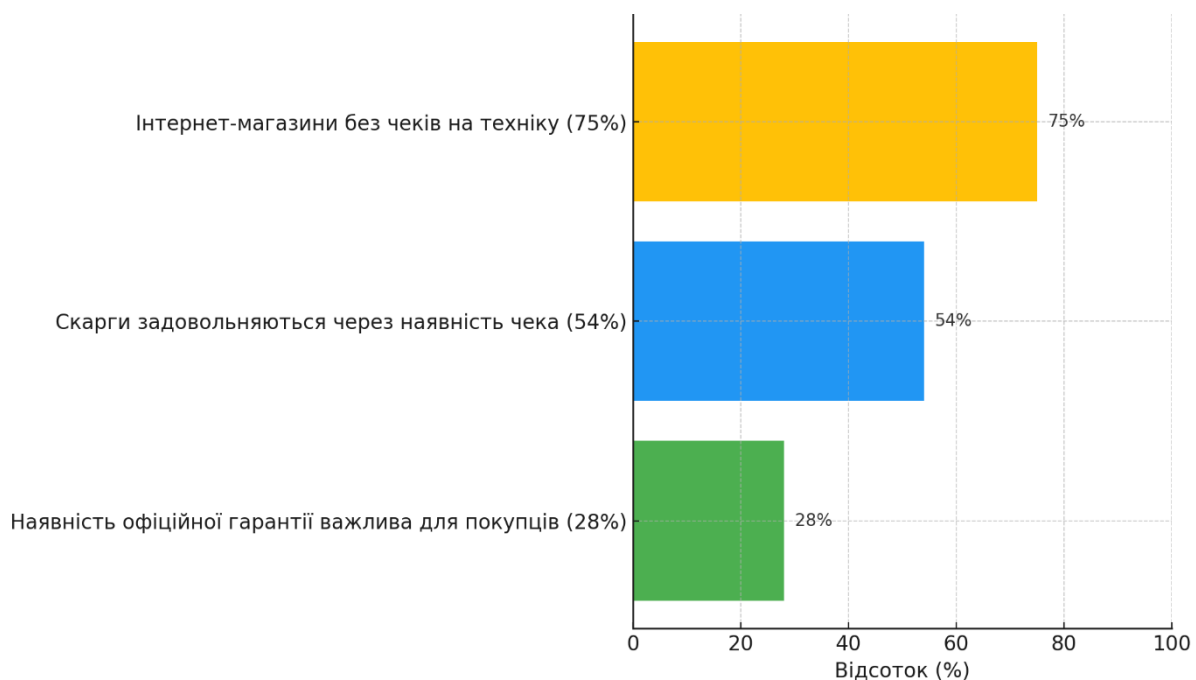


Рисунок 1.1 – Статистика обліку гарантійних обслуговування пристроїв

На рисунку наведено порівняння трьох основних показників: частки покупців, для яких важлива офіційна гарантія, відсотка задоволених скарг завдяки наявності фіскальних чеків та частки інтернет-магазинів, що не видають чеки.

Дані ілюструють актуальність проблеми обліку гарантійних документів: без централізованого контролю користувачі часто втрачають право на безкоштовне обслуговування, що призводить до додаткових фінансових витрат.

1.2 Аналіз існуючих програмних рішень

У сучасному цифровому середовищі для ведення обліку гарантійного та технічного обслуговування пристроїв користувачі мають змогу використовувати різноманітні програмні продукти. Залежно від потреби в автоматизації процесів, рівня деталізації інформації та вимог до інтеграції з іншими сервісами, обираються різні категорії програмного забезпечення. Основними рішеннями, що заслуговують на увагу, є календарні додатки, системи нагадувань та CRM-системи. Окремі користувачі також розглядають можливість використання спеціалізованих сервісів обліку гарантій або створення власних рішень на основі таблиць і баз даних.

Календарні додатки

Календарі стали одним із базових інструментів цифрової організації часу. Популярні рішення, такі як Google Calendar, Outlook Calendar, Apple Calendar, а також численні локальні календарі на Android і iOS, забезпечують базову функціональність планування подій та нагадувань.

- Стандартні можливості календарних сервісів включають:
- Створення подій із фіксованою датою та часом.
- Налаштування повторення подій (щодня, щотижня, щомісяця).
- Додавання сповіщень заздалегідь (за годину, день, тиждень тощо).
- Спільне використання календаря між кількома користувачами.
- Синхронізацію між пристроями через хмарні сервіси.

Календарні додатки мають значні переваги з точки зору доступності та простоти використання. Користувачі можуть швидко встановити нагадування про закінчення гарантії певного пристрою, не потребуючи спеціального навчання або тривалого налаштування системи.

Однак їхнє використання для комплексного обліку пристроїв має суттєві обмеження:

- Відсутність полів для введення розширеної інформації про пристрій (серійний номер, модель, виробник).
- Неможливість прикріплення супутніх документів (чеків, гарантійних талонів) без використання сторонніх сервісів.
- Відсутність механізмів для аналітики, сортування або групування пристроїв.
- Ризик втрати інформації у разі зміни пристрою без належної синхронізації.

Таким чином, календарні додатки більше підходять для одиничного обліку важливих дат, ніж для системного ведення гарантійної інформації.

Стандартні додатки нагадувань

Додатки нагадувань, інтегровані в сучасні операційні системи, такі як Google Tasks (Android), Apple Reminders (iOS), Microsoft To Do (Windows), пропонують дещо інший підхід до організації справ.

- Основні функціональні можливості цих сервісів:
- Створення списків завдань із кінцевими термінами виконання.
- Розбиття завдань на підзавдання.
- Пріоритезація справ.
- Встановлення нагадувань на певний час.
- Синхронізація із календарями та іншими програмами.

Перевагами додатків нагадувань є простота у використанні, мінімалістичний інтерфейс, а також можливість швидкого створення нагадування в один клік. У контексті обліку гарантійного обслуговування вони можуть бути використані для встановлення завдань, наприклад: "Перевірити гарантію холодильника Samsung до 15.06.2025".

Однак ці сервіси мають обмеження:

- Відсутність детальної інформаційної структури для кожного пристрою.
- Обмежена можливість додавання документів або додаткових атрибутів до завдань.
- Неможливість автоматизації процесів обліку (наприклад, автоматичний розрахунок закінчення гарантії від дати покупки).

Через це додатки нагадувань також більше підходять для обліку окремих випадків, а не для ведення повноцінної бази даних пристроїв.

CRM-системи

					КНУ.РБ.123.25.01. AIRCOGT	Арк.
	Арк.	№ документа	Підпис	Дата		

CRM-системи (Customer Relationship Management) першочергово призначені для управління відносинами з клієнтами та організації бізнес-процесів. Такі системи, як Salesforce, HubSpot, Zoho CRM, мають потужний функціонал, який дозволяє адаптувати їх під найрізноманітніші потреби, у тому числі для обліку пристроїв.

Базові можливості CRM-систем:

- Створення повноцінних профілів об'єктів з безліччю атрибутів (назва, серійний номер, дата купівлі, виробник, дата завершення гарантії).
- Прикріплення файлів (чеки, інструкції, гарантійні талони) до профілю пристрою.
- Налаштування автоматичних сповіщень про закінчення гарантії.
- Можливість створення кастомних звітів та аналітики (наприклад, які бренди найчастіше виходять із ладу).
- Масштабованість для великої кількості записів.

Незважаючи на свої переваги, CRM-системи мають ряд недоліків для домашнього або малокомерційного використання:

- Висока складність первинного налаштування.
- Потреба у постійному адмініструванні та оновленні даних.
- Часто CRM-системи є платними або мають обмеження у безкоштовних версіях.
- Потреба в навчанні користувачів для ефективної роботи з системою.

Таким чином, CRM-системи виправдані при великій кількості пристроїв або якщо потрібно вести історію їх обслуговування в організованому вигляді, однак для побутових завдань часто є надмірним рішенням.

Інші підходи

Окрім вищезазначених категорій, окремі користувачі створюють власні системи обліку на базі:

Google Sheets або Excel — створення таблиць із даними про пристрої.

Notion, Evernote — використання структурованих нотаток з можливістю прикріплення файлів.

Спеціалізовані додатки для гарантій — менш поширені, але існують програми, які прямо орієнтовані на облік гарантій, наприклад, Warranty Keeper, Bliggit Warranty Tracker.

Такі рішення дають змогу більш гнучко організувати облік, але вимагають від користувача самостійного налаштування системи і ведення даних.

Таблиця 1.1 - Порівняння інструментів для гарантійного обліку

Критерій	Google Calendar	Стандартні нагадування	CRM-системи	Розроблюваний додаток для обліку гарантій
Структурований облік пристроїв	×	×	✓	✓
Збереження копій чеків	×	×	✓	✓
Автоматичний розрахунок терміну гарантії	×	×	✓	✓
Простота використання	✓	✓	×	✓
Сповіщення про закінчення гарантії	×	✓	✓	✓
Можливість позначити пристрій як зламаний	×	×	✓	✓
Пошук сервісних центрів	×	×	×	✓
Аналітика брендів (середній термін роботи, частота поломок)	×	×	×	✓
Експорт статистики у файл	×	×	✓	✓

1. Структурований облік пристроїв

Google Calendar та стандартні нагадування не мають можливості систематизувати дані про пристрої (тільки прості події або нагадування).

CRM-системи мають потужні можливості для обліку, але часто перевантажені зайвими функціями для обслуговування клієнтів, що не потрібно для гарантій.

Розроблюваний додаток чітко орієнтований на облік саме пристроїв: назва, бренд, дата купівлі, термін гарантії.

2. Збереження копій чеків

У Google Calendar та нагадуваннях немає такої можливості.

CRM може зберігати файли, але налаштування складне.

У розроблюваній програмі передбачена легка можливість завантажити чек при додаванні пристрою і швидкий доступ до нього.

3. Автоматичний розрахунок терміну гарантії

У загальних календарях та нагадуваннях все треба робити вручну.

CRM може обробляти дати через спеціальні модулі.

Програма самостійно рахувати дату закінчення гарантії після введення дати покупки і тривалості.

4. Простота використання

Google Calendar і стандартні нагадування прості, але функціонал обмежений.

CRM зазвичай вимагають навчання через велику кількість зайвих функцій. Розроблюваний додаток буде мати інтуїтивний інтерфейс: проста форма додавання пристрою, автоматичні розрахунки, зрозуміла таблиця.

5. Сповіщення про закінчення гарантії

Google Calendar сам по собі не аналізує гарантію — тільки фіксоване нагадування.

У стандартних нагадуваннях можна поставити нагадування, але без автоматичного аналізу.

CRM та розроблюваний програма — автоматичне виявлення пристроїв із закінченням гарантії (у тебе ще й повідомлення про це буде).

6. Позначення пристрою як зламаного

Така функція є тільки в деяких розширених CRM.

У розроблюваному додатку можна буде вручну змінити статус пристрою на "Зламаний" і програма автоматично перевіряє чи є ще гарантія.

7. Пошук сервісних центрів

Google Calendar, нагадування і CRM такої функції не мають.

Програма пропонує список сервісних центрів по бренду або повідомляє, що найближчі відсутні — це значно спрощує ремонт пристрою.

8. Аналітика брендів

Жоден із стандартних варіантів не пропонує аналіз поломок по брендам.

У програмі буде аналітика: середній термін роботи, частота поломок за брендом.

9. Експорт статистики у файл

Google Calendar і нагадування не мають експорту пристроїв і гарантій.

CRM може експортувати дані, але складно налаштувати потрібні звіти.

У програмі користувач може експортувати список пристроїв та аналітику у CSV або Excel буквально в 1 клік.

1.3 Огляд існуючих мобільних додатків та сервісів для гарантійного обліку

У процесі вирішення завдання ефективного обліку гарантійного обслуговування електронних та побутових пристроїв користувачі звертають увагу на різні типи мобільних додатків та онлайн-сервісів. Серед них виділяються як універсальні інструменти для організації особистої інформації, так і спеціалізовані застосунки, що орієнтовані на облік покупок і гарантійних термінів.

Одним із найпоширеніших підходів є використання додатків для створення нотаток, таких як Evernote, Google Keep та Microsoft OneNote. Ці сервіси дозволяють користувачам створювати текстові записи, прикріплювати зображення чеків, гарантійних талонів, договорів на обслуговування тощо. Наявність функції встановлення нагадувань дозволяє певним чином організувати контроль над строками дії гарантії.

Однак такі універсальні рішення мають низку обмежень:

Вони не пропонують спеціалізованих полів для вводу даних про пристрій (назва, модель, бренд, дата покупки, тривалість гарантії).

Немає автоматичного розрахунку дати закінчення гарантійного терміну.

Відсутні механізми групування пристроїв за статусом гарантії або поломки.

Немає можливості швидкого отримання статистичних даних або формування звітів.

Ще однією категорією програмного забезпечення є мобільні застосунки спеціалізованого характеру, орієнтовані саме на облік гарантійних термінів. Їх приклади включають:

Warranty Keeper

Warranty Keeper — це популярний мобільний додаток, розроблений спеціально для збереження гарантійних даних. Основні можливості застосунку:

Створення записів про нові пристрої із зазначенням дати покупки та тривалості гарантії.

Додавання фотографій чеків та гарантійних талонів.

Надсилання автоматичних нагадувань про наближення дати закінчення гарантії.

Проте, попри зручність базового функціоналу, додаток має обмежену аналітичну складову: відсутня можливість аналізу середнього терміну роботи брендів, немає фільтрів за категоріями пристроїв чи їх статусом.

Bbyd

Bbyd — менш відомий додаток, який орієнтується на керування підписками та фінансовими зобов'язаннями. Частково його можна використовувати для контролю гарантій, встановлюючи персоналізовані нагадування про закінчення терміну обслуговування. Однак він не призначений для повноцінного обліку характеристик пристроїв, гарантійних строків і їхнього технічного стану.

My Warranty

					КНУ.РБ.123.25.01. AIRCOGT	Арк.
Арк.	№ документа	Підпис	Дата			

My Warranty — застосунок, орієнтований на збереження базової інформації про пристрої та їх гарантії. Програма дозволяє:

- Створювати профілі пристроїв.
- Вказувати дату покупки та тривалість гарантії.
- Зберігати фотографії чеків.

Водночас функціонал My Warranty обмежується простим нагадуванням і не підтримує глибокої інтеграції з сервісними центрами, аналітики за брендами чи експорту даних у зовнішні формати.

Попри наявність певної кількості мобільних застосунків для гарантійного обліку, аналіз існуючих рішень демонструє системні недоліки, що обмежують їх ефективність у реальних сценаріях використання.

Перш за все, більшість таких програм фокусуються на збереженні мінімальної інформації про покупку без можливості комплексного управління життєвим циклом пристрою. Користувач отримує нагадування про завершення гарантії, однак залишається без підтримки при подальших діях: пошуку сервісного центру, оформленні заявки на ремонт або проведенні аналізу надійності виробника.

Іншою суттєвою проблемою є відсутність стандартизованої структури збереження даних. У багатьох застосунках користувачу пропонується заповнювати довільні поля без чітких категорій і формату. Це ускладнює подальший пошук пристроїв, унеможлиблює ефективне сортування і фільтрацію за різними критеріями — наприклад, за брендом, датою покупки або поточним статусом гарантії.

Важливим аспектом, який практично не враховується більшістю наявних рішень, є потреба в інтеграції із зовнішніми сервісами. Сучасний користувач очікує, що додаток зможе автоматично надати перелік найближчих авторизованих сервісних центрів для конкретного бренду або запропонувати типові інструкції дій при поломці пристрою в межах гарантії.

Крім того, можливості візуалізації та аналітики даних у існуючих рішеннях залишають бажати кращого. Для користувачів, які мають декілька або десятки пристроїв, було б корисним мати доступ до статистики — наприклад, інформації про те, які бренди частіше виходять з ладу або які пристрої потребують обслуговування найчастіше. Проте подібна функціональність відсутня у більшості існуючих додатків.

Ще однією перешкодою для широкого використання спеціалізованих рішень є питання безпеки збережених даних. Чеки, серійні номери пристроїв, персональні дані можуть містити конфіденційну інформацію, і багато програм

не приділяють належної уваги захисту даних — наприклад, не використовують шифрування або багатофакторну аутентифікацію.

Нарешті, варто зазначити обмежену гнучкість існуючих систем у плані експорту даних. Більшість мобільних додатків дозволяють лише базовий перегляд інформації в інтерфейсі додатку без можливості вивантаження даних у форматах, придатних для збереження або аналізу (наприклад, CSV або Excel).

Таким чином, проведений огляд існуючих мобільних додатків та сервісів показує, що на сучасному ринку відчувається суттєвий дефіцит інструментів, які б повністю задовольняли потреби користувачів в обліку гарантійного обслуговування електронних і побутових пристроїв.

Більшість існуючих рішень обмежуються лише базовими функціями — збереженням дати покупки та встановленням простого нагадування про завершення гарантійного терміну. Проте реальні потреби користувачів набагато ширші. У процесі життєвого циклу пристрою виникають додаткові запити, такі як відстеження стану пристрою (наприклад, чи є він у робочому стані, чи потребує ремонту), потреба в пошуку авторизованих сервісних центрів, надання рекомендацій щодо дій при поломці в межах гарантії.

Іншими словами, існуючі програми на сьогоднішній день переважно орієнтовані на пасивний облік інформації, тоді як користувачам потрібен активний інструмент управління життєвим циклом пристрою. Відсутність адаптивності до змін стану пристрою, нестача гнучких систем фільтрації (наприклад, фільтрація за терміном завершення гарантії або за ознакою поломки) значно обмежує можливості користувача у прийнятті обґрунтованих рішень.

Крім функціональних недоліків, існують також серйозні питання щодо безпеки персональних даних у таких системах. Оскільки інформація про пристрої часто включає серійні номери, вартість, документи на покупку та персональні дані власника, їх витік або втрата можуть призвести до фінансових та репутаційних ризиків. На жаль, більшість існуючих додатків не надають достатнього рівня шифрування даних, можливостей резервного копіювання або багатофакторної аутентифікації користувачів.

Ще одним важливим аспектом є недостатня масштабованість існуючих рішень. При обліку невеликої кількості пристроїв, базовий функціонал може бути достатнім. Проте при зростанні кількості пристроїв навіть до кількох десятків виникають складнощі з пошуком необхідної інформації, її аналізом та оновленням. Це особливо критично для малого бізнесу, де облік гарантій і сервісного обслуговування обладнання має прямий вплив на безперебійність роботи та витрати компанії.

Крім технічних обмежень, слід також зазначити слабку локалізацію багатьох існуючих додатків. Частина з них не підтримує українську мову, не адаптована до локальних сервісних центрів або не враховує особливості гарантійних умов на території України. Це створює додаткові незручності при спробах використовувати такі додатки на практиці.

Таким чином, можна стверджувати, що наразі ринок мобільних рішень у сфері обліку гарантій є недостатньо розвиненим, і потребує появи нового продукту, який би відповідав сучасним вимогам користувачів. Такий продукт повинен включати:

- централізовану базу даних пристроїв із чіткою структурою полів,
- автоматизовану систему сповіщень про важливі події,
- можливості сортування та фільтрації пристроїв за різними критеріями,
- аналітичні модулі для візуалізації стану парку пристроїв,
- інтеграцію з базами сервісних центрів,
- захист персональних даних на високому рівні,
- підтримку багатомовності та локалізації під специфіку українського ринку.

У наступному підрозділі буде докладніше визначено конкретні функціональні та нефункціональні вимоги до розроблюваної системи, що дозволить забезпечити повне управління процесом обліку гарантійного та технічного обслуговування пристроїв.

Таблиця 1.2 - Порівняння можливостей популярних додатків для обліку гарантій

Назва додатку	Збереження чеку	Автоматичне нагадування	Статус пристрою	Аналітика	Інтеграція з сервісами
Warranty Keeper	Так	Так	Частково	Ні	Ні
Bbyd	Ні	Так	Ні	Ні	Ні
MyWarranty	Так	Так	Ні	Ні	Ні

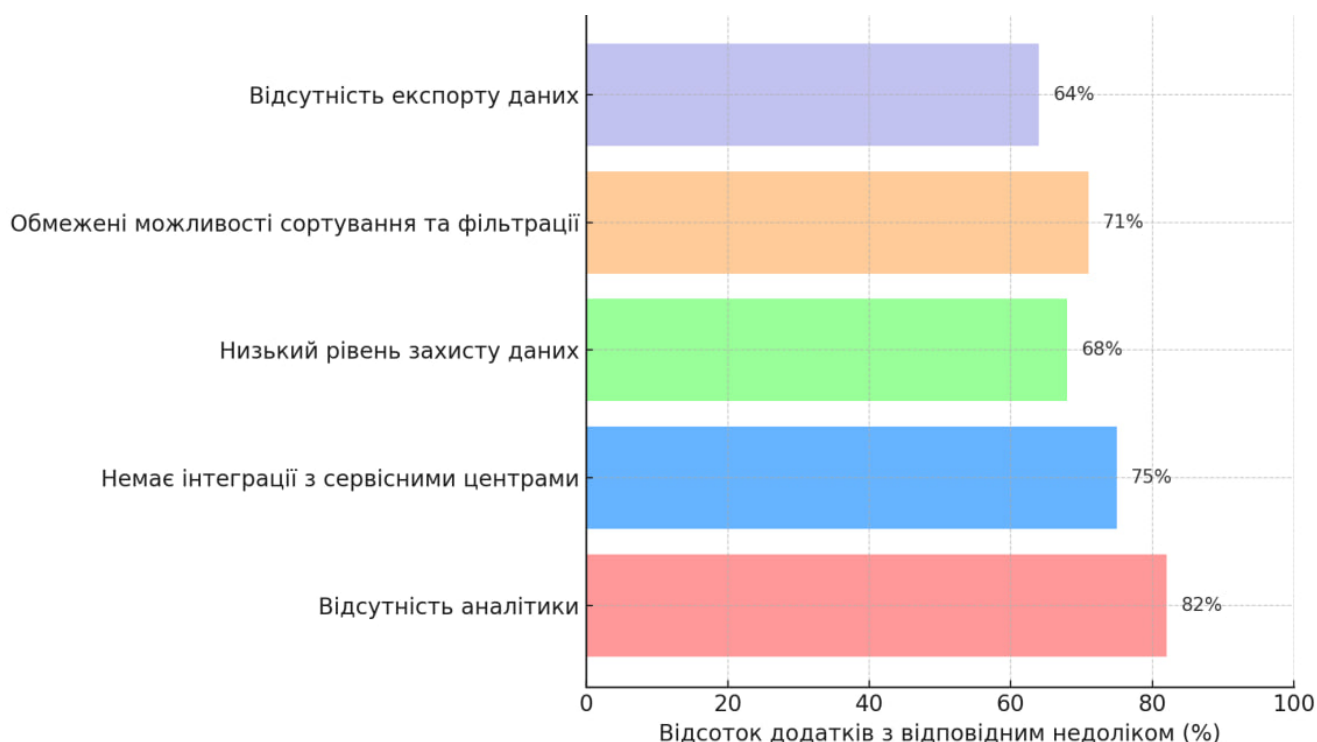


Рисунок 1.2 – Основні недоліки існуючих мобільних додатків для обліку гарантій

Проведений аналіз наявних мобільних застосунків для обліку гарантійного обслуговування пристроїв показав, що більшість із них мають суттєві обмеження у функціональності. Найчастіше відсутні такі важливі можливості, як аналітика, інтеграція із сервісними центрами, ефективне сортування інформації, захист персональних даних та експорт даних у зручних форматах.

Зважаючи на виявлені недоліки, виникла необхідність розробити спеціалізований застосунок, який би усував ці проблеми та максимально задовольняв реальні потреби користувачів у сфері обліку гарантійного обслуговування.

Основні функціональні можливості розроблюваної системи включатимуть:

- централізоване збереження інформації про пристрої (назва, бренд, дата покупки, тривалість гарантії);
- можливість додавання копій чеків для кожного пристрою;
- автоматичний розрахунок дати завершення гарантійного терміну;
- повідомлення користувача про наближення закінчення гарантії;
- можливість позначати пристрої як зламані та отримувати рекомендації щодо сервісного обслуговування;
- формування аналітичних графіків (наприклад, частота поломок по брендах);
- експорт даних у формати CSV або Excel;

забезпечення захисту персональних даних користувачів за допомогою сучасних методів шифрування та аутентифікації.

Таким чином, майбутній програмний продукт не тільки компенсує недоліки існуючих рішень, а й надасть користувачам повноцінний інструмент для управління життєвим циклом їхніх електронних та побутових пристроїв.

1.4 Вимоги до ефективного інструменту обліку гарантійного обслуговування

Результати аналізу існуючих програмних рішень, представлених у попередніх підрозділах, дозволяють сформулювати загальні вимоги до інформаційної системи, призначеної для обліку гарантійного та технічного обслуговування електронних і побутових пристроїв. Така система повинна бути не лише зручною у використанні, а й функціонально повною, безпечною, масштабованою та адаптованою до потреб звичайного користувача.

У контексті сучасного цифрового середовища важливо розглядати цю систему не просто як «базу даних з пристроями», а як інтерактивний інструмент, що допомагає контролювати гарантійні терміни, аналізувати тенденції поломок, приймати рішення про ремонт чи заміну, і при цьому мінімізує ризики, пов'язані з людським фактором — забутими датами, втраченими чеками, простроченими зверненнями до сервісу.

Оскільки подібні задачі потребують комплексного підходу, доцільно поділити вимоги до майбутньої системи на кілька груп: функціональні, нефункціональні, інтерфейсні, аналітичні та вимоги до захисту даних.

До основного функціоналу системи у ідеалі повинні входити :

- можливість додавання пристроїв з ключовими параметрами: назва, бренд, дата покупки, тривалість гарантії;
- прикріплення зображення чеку або гарантійного талону;
- автоматичний розрахунок дати завершення гарантії на основі дати покупки та її тривалості;
- формування централізованого списку пристроїв з можливістю пошуку та фільтрації;
- реалізація статусів пристроїв: активний, гарантія закінчується, гарантія завершена, зламаний;
- встановлення мітки "зламаний" з можливістю додати дату поломки;
- перевірка чинності гарантії при позначенні пристрою як зламаного;
- генерація сповіщення про гарантію, що скоро завершиться;
- рекомендації щодо дій при поломці — включно з відображенням контактів сервісних центрів або інформації про їхню відсутність.

Нефункціональні вимоги описують характеристики системи, які не визначають конкретні функції, але мають критичне значення для стабільної та ефективної роботи, зручності використання, безпеки, масштабування та адаптивності програмного забезпечення. Саме вони формують загальне враження користувача від системи, впливають на її популярність і здатність задовольняти потреби в реальних умовах експлуатації.

Зручність використання (Usability)

Інтерфейс користувача повинен бути інтуїтивно зрозумілим і орієнтованим на широке коло користувачів — від технічно підкованих фахівців до звичайних користувачів, які не мають спеціалізованих знань. Всі основні дії, такі як додавання нового пристрою, перегляд гарантійної інформації, фільтрація пристроїв за статусом, позначення пристрою як зламаного або перегляд аналітики, повинні бути реалізовані у вигляді простих зрозумілих кнопок, форм або випадаючих меню.

Окрему увагу варто приділити організації візуального простору інтерфейсу. Дані про пристрої мають відображатися у вигляді таблиці з можливістю сортування та фільтрації. Інформація повинна бути представлена у компактній формі, з можливістю швидкого доступу до розширених деталей. Колірні індикатори (наприклад, жовтий — гарантія закінчується, червоний — гарантія завершена) можуть значно полегшити візуальне сприйняття інформації.

Крім цього, система має забезпечувати мінімальну кількість кліків для виконання повсякденних задач. Це особливо важливо для користувачів, які мають великий обсяг записів і цінують економію часу.

Продуктивність та швидкодія

Швидкість відгуку інтерфейсу напряму впливає на задоволеність користувача. При переході між розділами, відкритті таблиці пристроїв, застосуванні фільтрів або натисканні кнопок очікуваний час реакції системи повинен складати не більше 1–2 секунд. У випадках, коли запити стосуються великої кількості пристроїв (100+), час відповіді не повинен перевищувати 3–4 секунд навіть при складній аналітичній обробці.

Для цього система повинна ефективно використовувати кешування, асинхронну обробку даних та оптимізовані запити до бази даних. У випадку вебреалізації рекомендується попереднє завантаження основних блоків інтерфейсу та відкладене завантаження другорядного контенту.

Портативність та доступність

Сучасні користувачі очікують мати доступ до своїх даних із будь-якого пристрою, незалежно від місця перебування. Саме тому система повинна бути реалізована як адаптивний веб-застосунок або мати окрему мобільну версію. Інтерфейс повинен динамічно підлаштовуватись під розмір екрана: смартфон, планшет або десктоп.

Також важливо передбачити синхронізацію даних між пристроями — наприклад, при роботі в обліковому записі через браузер і мобільний застосунок. Резервне копіювання або експорт бази у файл (наприклад, JSON або CSV) має бути доступним в один клік, аби користувач міг відновити дані в разі втрати пристрою.

Масштабованість

Програма повинна однаково ефективно працювати як із декількома пристроями, так і зі значними обсягами даних — десятками або сотнями записів. Це означає, що на рівні архітектури має бути закладена підтримка масштабування — як по кількості записів, так і по модулях (наприклад, можливість у майбутньому додати нові функції: онлайн-авторизацію, спільну роботу, API тощо).

Система повинна залишатися стабільною незалежно від кількості користувачів, які взаємодіють з нею одночасно (у випадку хмарної реалізації), або кількості пристроїв у локальній базі.

Аспект	Функціональні Вимоги	Нефункціональні Вимоги
Визначення	Визначає, що повинна робити система (дії та завдання).	Описує, наскільки добре система виконує свої функції (атрибути якості).
Мета	Забезпечити виконання системою необхідних функцій для користувачів.	Забезпечити відповідність системи стандартам якості щодо продуктивності, надійності та зручності.
Фокус	Орієнтований на функції та можливості системи.	Орієнтований на якість системи та досвід користувача.
Приклад	Вхід користувача, отримання даних, обробка платежів.	Швидкодія, доступність системи, протоколи безпеки.
Метод тестування	Функціональне тестування (для перевірки поведінки функцій).	Тестування продуктивності, безпеки та зручності.
Вимірювання	Вимірюється через конкретні функції або дії.	Вимірюється атрибутами якості (наприклад, часом відгуку, часом безвідмовної роботи).
Вплив на розробку	Визначається на ранньому етапі для формування основної функціональності системи.	Часто уточнюється протягом розробки для забезпечення стабільності та масштабованості.
Ціль	Виконання потреб користувача та бізнесу щодо можливостей системи.	Поліпшення користувацького досвіду та забезпечення надійності системи.

Рисунок 1.3 – Основні категорії нефункціональних вимог

Інтерфейс користувача відіграє критичну роль у будь-якому програмному забезпеченні, орієнтованому на збереження, перегляд і керування структурованими даними. Особливо це стосується систем, у яких користувач взаємодіє з численними об'єктами, такими як пристрої, що перебувають на гарантійному обслуговуванні. Від зручності інтерфейсу наряду залежить ефективність виконання повсякденних дій, кількість помилок з боку користувача, швидкість доступу до важливої інформації та загальний досвід роботи із застосунком.

Оскільки кінцевими користувачами подібних рішень часто є не технічні спеціалісти, інтерфейс має бути максимально простим, логічно побудованим і передбачуваним. Візуальна структура повинна сприяти швидкому орієнтуванню в основних можливостях системи, виключати плутанину між діями, а також мінімізувати кількість кроків для досягнення цільової задачі.

Важливим критерієм є також консистентність інтерфейсу — його зовнішній вигляд і логіка навігації мають бути однаковими на всіх етапах взаємодії, незалежно від конкретного сценарію використання. Це дає змогу зменшити когнітивне навантаження на користувача та уникнути ситуацій, у яких функціональність втрачається через неочевидне розміщення елементів або непрозоре керування.

Окрему увагу слід приділити тому, як інтерфейс реагує на зміну вмісту — оновлення інформації, появу нових записів або зміну статусу об'єкта. Всі подібні зміни мають бути відображені чітко, з можливістю швидко зрозуміти, що саме відбулося. Важливо, щоб динамічність інтерфейсу не шкодила його стабільності: виведення повідомлень, підтверджень чи попереджень повинно бути поміркованим і не заважати основному процесу роботи.

Інтерфейс має бути достатньо гнучким, щоб забезпечити комфортну роботу як у випадку з невеликою кількістю записів, так і при значному їх обсязі. Це вимагає логічної організації інформації, її компактного представлення та можливості легко переходити від загального перегляду до перегляду окремих деталей.

Таким чином, вимоги до інтерфейсу користувача в системах обліку гарантій мають ґрунтуватися на принципах простоти, послідовності, швидкого доступу до даних та зрозумілої взаємодії. Саме ці властивості визначають ефективність взаємодії людини з програмним забезпеченням і, як наслідок, — практичну цінність самого продукту.

Аналітична складова в програмних системах подібного типу відіграє роль інструмента вищого рівня, який дозволяє не лише фіксувати інформацію, а й отримувати з неї узагальнені висновки для прийняття рішень. У випадку обліку гарантійного обслуговування електронних та побутових пристроїв, аналітичні можливості надають користувачу змогу оцінити загальний стан своєї техніки, виявити закономірності щодо частоти поломок, переглянути статистику по брендам тощо.

Успішна реалізація такого блоку дозволяє підняти рівень корисності системи: з простої бази записів вона перетворюється на інструмент контролю та прогнозування. Наприклад, користувач може своєчасно помітити, що певна марка пристроїв частіше виходить з ладу до завершення гарантії, і відповідно ухвалити обґрунтоване рішення про майбутні покупки або звернення до сервісу.

Аналітична підсистема повинна ґрунтуватися на накопичених даних користувача — таких як дати покупок, тривалість гарантії, частота поломок, строки служби окремих брендів або типів пристроїв. Візуальне представлення цієї інформації у вигляді діаграм, графіків або звітів значно полегшує її сприйняття, особливо при роботі з великою кількістю записів.

Окреме значення має надання зведеної інформації про пристрої, гарантія на які закінчується у найближчий час, або вже завершилася. Така інформація повинна бути доступною у вигляді коротких звітів або дашбордів, що дозволяє користувачу оперативно реагувати на потенційні ризики.

Предбачення аналітичного блоку в архітектурі системи також дає змогу у майбутньому реалізувати розширену статистику — наприклад, за типами пристроїв, моделями, роками придбання, тривалістю служби тощо. Усе це підвищує гнучкість і перспективність подібних програмних рішень.

Безпека персональних даних та цілісність інформації є одним із ключових аспектів, що визначають довіру користувача до програмного забезпечення. У випадку систем, пов'язаних з обліком пристроїв та їх гарантійним обслуговуванням, особливої важливості набуває захист від несанкціонованого доступу, втрати або пошкодження даних.

Типовий набір інформації, що зберігається у таких системах, включає найменування пристроїв, дати покупок, документи на гарантію, зображення чеків та інші чутливі дані, які потенційно можуть бути використані зловмисниками. Окрім цього, для багатьох користувачів ці записи мають практичну цінність у випадках звернення до сервісних центрів чи ведення фінансової звітності.

Тому інформаційна система повинна бути розроблена з урахуванням базових принципів інформаційної безпеки:

захист конфіденційності, тобто забезпечення доступу до даних лише авторизованим користувачам;

цілісність інформації, що передбачає запобігання її несанкціонованому змінненню або пошкодженню;

доступність даних, зокрема можливість їх резервного копіювання та відновлення у випадку втрати пристрою чи збою системи.

Також варто враховувати технічні аспекти безпеки: збереження файлів чеку або інших вкладених документів не повинно піддавати систему ризикам, пов'язаним з використанням зовнішніх зловмисних об'єктів.

Окремо важливо передбачити можливість експорту або локального збереження даних користувача у відкритому форматі. Це забезпечує контроль над власною інформацією навіть без активного підключення до інтернету або в автономному режимі.

Незалежно від форми реалізації — локальна програма чи веб-застосунок — система повинна бути побудована із розумінням того, що навіть прості на перший погляд рішення зберігають потенційно чутливу інформацію. Це зобов'язує розробника відповідально підходити до аспектів безпеки з самого початку проектування.

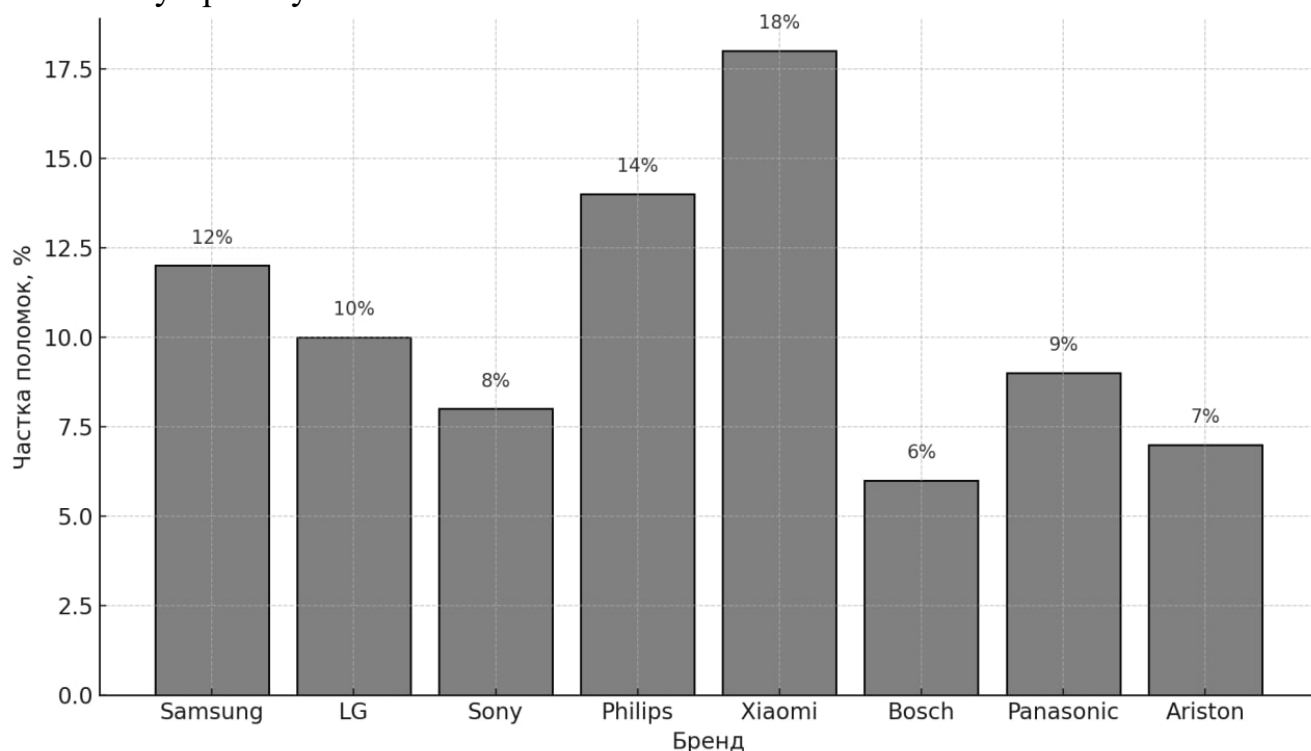


Рисунок 1.4 – Приблизна частота поломок пристроїв за брендами

Висновки

Було здійснено всебічний аналіз сучасних підходів до обліку гарантійного та технічного обслуговування електронних і побутових пристроїв. Розглянуто ключові проблеми, з якими стикаються користувачі в побуті та бізнесі: це, зокрема, втрата супровідних документів (чеків,

гарантійних талонів), забуті або неконтрольовані строки дії гарантії, а також відсутність зручного централізованого механізму для ведення обліку пристроїв.

У ході аналізу програмних рішень було розглянуто як універсальні засоби, зокрема календарні сервіси та системи нагадувань, так і спеціалізовані CRM-системи. Незважаючи на певні функціональні переваги, жоден із зазначених інструментів не дозволяє реалізувати повноцінний облік гарантійного обслуговування. Основні недоліки наявних рішень полягають у складності використання, відсутності автоматичного розрахунку термінів гарантії, слабкій підтримці аналітики, обмежених можливостях сортування пристроїв за статусами та відсутності зв'язку із сервісними центрами.

Окрему увагу було приділено аналізу мобільних застосунків для збереження гарантій. Дослідження продемонструвало, що навіть спеціалізовані рішення, такі як Warranty Keeper чи My Warranty, не враховують багатьох важливих потреб користувача. Зокрема, йдеться про відсутність гнучкої структури даних, аналітичних інструментів, можливості формування звітів, належного захисту персональної інформації та адаптації до локальних умов (мови, сервісні центри, регіональні особливості гарантій).

З урахуванням виявлених проблем і функціональних прогалин було сформульовано бачення майбутнього інструменту, який дозволить автоматизувати облік гарантійного і технічного обслуговування. Розроблювана система має включати централізовану базу даних пристроїв з можливістю зберігання супровідної інформації, функціонал нагадувань, інтеграцію з сервісними структурами, механізми фільтрації, аналітичні модулі, засоби експорту даних та забезпечення належного рівня інформаційної безпеки.

Таким чином, результати аналізу чітко демонструють актуальність створення нової програмної системи, здатної забезпечити ефективний, надійний та зручний облік гарантійного обслуговування в побуті й у професійному середовищі.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Загальна архітектура системи

Інформаційна система "Календар-довідник гарантійного та технічного обслуговування електронних та побутових пристроїв" призначена для зручного ведення обліку пристроїв, відстеження терміну гарантії, фіксації поломок, а також отримання аналітики та експорту статистики.

Загальна архітектура системи базується на модульному підході й охоплює такі основні компоненти:

- Інтерфейс користувача — реалізований у вигляді форм Windows Forms (WinForms), де користувач може додавати пристрої, переглядати список, отримувати повідомлення, фільтрувати записи, переглядати статистику.
- Логічний модуль — обробляє введені дані, обчислює дату завершення гарантії, перевіряє її дійсність, формує повідомлення, здійснює фільтрацію.
- База даних — зберігає всю інформацію про пристрої, їхній статус, чеки, бренди, дати покупки, терміни гарантії, поломки, а також список сервісних центрів.
- Модуль сповіщень — відповідає за автоматичну генерацію попереджень про наближення дати завершення гарантії (наприклад, за 3 дні).
- Модуль аналітики — формує графіки та статистику за зібраними даними (середній термін служби брендів, частота поломок).
- Модуль експорту — забезпечує вивантаження статистичних даних у CSV або Excel для подальшого аналізу або архівування.

Компоненти системи взаємодіють між собою за спрощеною схемою:

→ Користувач працює з інтерфейсом → дані передаються у логіку → логіка записує або зчитує дані з бази → модулі сповіщень, аналітики та експорту працюють з цією інформацією.

Система реалізується як локальний застосунок на платформі Windows, що не вимагає підключення до інтернету та зберігає всі дані локально у внутрішній базі (наприклад, SQLite або PostgreSQL при потребі).

					КНУ.РБ.123.25.01.ПІС			
Змн.	Арк.	№ документа	Підпис	Дата	ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ			
Розробив	Басов							
Перевірив	Купін							
Н.контроль	Кузнецов							
Затвердив	Купін				Літера Аркуш Аркушів KI-21			

Архітектура побудована з урахуванням простоти використання, стабільності та можливості масштабування функціоналу в майбутньому (наприклад, додавання авторизації або хмарної синхронізації).

2.2 Функціональні та нефункціональні вимоги

Функціональні вимоги

Система повинна виконувати такі основні функції:

1. Додавання нового пристрою

- Користувач вводить назву пристрою, бренд, дату покупки, тривалість гарантії.
- За бажанням додається зображення чека.
- Програма автоматично розраховує дату завершення гарантії та зберігає дані в базу.

2. Перегляд списку пристроїв

- Відображення таблиці з усіма доданими пристроями: назва, бренд, дата покупки, дата завершення гарантії, статус.
- Можливість фільтрації за статусами: “Усі”, “Гарантія закінчується”, “Зламани”.

3. Нагадування про завершення гарантії

- Програма щоденно перевіряє, чи залишилося 3 дні до завершення гарантії.
- Якщо так — відображає відповідне повідомлення.

4. Позначення пристрою як зламаного

- Користувач вибирає пристрій, вказує дату поломки.
- Програма перевіряє, чи гарантія ще діє, та видає повідомлення.
- Якщо гарантія активна — система пропонує список сервісних центрів або повідомлення про їх відсутність.

5. Аналітика та статистика

- Формування графіків: середній термін служби брендів, частота поломок.
- Дані оновлюються на основі введених пристроїв.

6. Експорт даних

Можливість експортувати список пристроїв та аналітичну статистику у форматах CSV або Excel.

Нефункціональні вимоги:

1. Зручність користування (Usability)

Інтерфейс має бути простим та зрозумілим, з чіткими назвами кнопок, форм та підказками до полів. Основні дії — додавання пристрою, фільтрація, експорт — виконуються в кілька кліків.

(Опціонально) можливе кольорове виділення рядків у таблиці: наприклад, жовтий — гарантія скоро завершиться, червоний — завершена. Це покращує візуальне сприйняття, але не є обов'язковим елементом.

2. Швидкодія та стабільність роботи (Performance)

Програма повинна працювати стабільно при звичайному обсязі даних (до 100 пристроїв). Очікується, що час обробки запитів (наприклад, фільтрації чи додавання запису) не перевищує 1–2 секунди.

(Опціонально) можуть реалізовуватись прості перевірки на помилки введення — наприклад, якщо не заповнено обов'язкове поле або введено некоректну дату.

3. Безпека та збереження даних (Security)

Дані зберігаються локально, доступ до них має лише користувач ПК.

(Опціонально) передбачена можливість резервного копіювання шляхом експорту даних у файл. Додані зображення чеків зберігаються лише за згодою користувача і відкриваються вручну.

4. Масштабованість (Scalability)

Система має бути побудована з урахуванням можливого розширення: додавання нових полів, аналітики чи зовнішньої синхронізації в майбутньому.

Базова структура має підтримувати роботу з великою кількістю пристроїв без втрати швидкодії.

5. Портативність (Portability)

Програма працює як десктопний застосунок під Windows без потреби встановлення серверів або СУБД.

(Опціонально) допускається запуск з флеш-накопичувача — наприклад, у вигляді portable-версії.

6. Локалізація (Localization)

Інтерфейс програми реалізується українською мовою. Усі повідомлення, дати та підписи адаптовано під локальні стандарти.

(Опціонально) можливе розширення на інші мови у майбутньому шляхом винесення текстів у ресурсні файли.

2.3. Структура бази даних:

Під час розробки програмного забезпечення було прийнято рішення використати систему керування базами даних PostgreSQL, оскільки вона є потужною, безкоштовною, кросплатформною, підтримує всі основні типи даних і дозволяє зручно працювати як з простими, так і з складними запитами. Також PostgreSQL забезпечує хорошу продуктивність, надійність, підтримує транзакції, зовнішні ключі та індекси, що робить її ідеальним вибором для створення прикладного ПЗ з фокусом на збереження облікової інформації про пристрої та сервісні центри.

У проєкті створено дві основні таблиці: `devices` та `service_centers`. Вони пов'язані між собою логічно, за значенням поля `brand` — хоча зв'язок не реалізовано у вигляді зовнішнього ключа, він підтримується на рівні логіки програми.

1) Таблиця `devices` — облік електронних пристроїв:

Ця таблиця є центральною у базі даних і призначена для зберігання усіх пристроїв, які користувач вводить у систему для подальшого обліку гарантії, фіксації поломок та аналітики. Вона включає такі поля:

Таблиця 2.1 - Таблиця `devices` — облік електронних пристроїв

<code>id</code>	integer (PK)	Унікальний ідентифікатор запису. Автоматично зростає.
<code>name</code>	text	Назва пристрою, яку вводить
<code>brand</code>	text	Бренд пристрою (Lenovo, Asus, HP тощо). Обирається зі списку в інтерфейсі.
<code>purchase_date</code>	date	Дата покупки пристрою.
<code>warranty_months</code>	integer	Тривалість гарантії у місяцях (вказується вручну).
<code>warranty_end_date</code>	date	Обчислюється автоматично на основі дати покупки + гарантія.
<code>receipt_path</code>	text	Шлях до зображення або PDF-файлу чека, який додається користувачем.
<code>status</code>	text	Поточний статус: Активний, Прострочений, Зламаний. Змінюється автоматично або вручну.
<code>fault_date</code>	date	Якщо пристрій зламався — вказується дата поломки. Інакше залишається null.

Призначення таблиці — дозволити вести облік пристроїв, слідкувати за станом гарантії, нагадувати користувачу про її завершення, та вести історію поломок. Кожен запис є унікальним, а поле `status` дозволяє швидко фільтрувати пристрої за актуальністю або технічним станом.

2. Таблиця `service_centers` — довідник сервісних центрів:

Дана таблиця реалізована як довідкова структура, яка містить заздалегідь заповнені дані про сервісні центри, що обслуговують конкретні бренди електронної техніки. Вона не змінюється користувачем у процесі експлуатації програми, що забезпечує стабільність її роботи та цілісність даних. У разі необхідності таблиця може бути оновлена адміністратором або розробником через адміністративний інтерфейс або безпосередньо в базі даних.

Така таблиця дозволяє автоматизувати процес пошуку контактної інформації у випадку виходу пристрою з ладу. Наприклад, якщо користувач зазначає, що його ноутбук бренду *Asus* зламався, програма автоматично звертається до таблиці `service_centers`, знаходить відповідний запис з брендом *Asus* і надає користувачу контактні дані найближчого сервісного центру — включно з його назвою, адресою, містом та номером телефону.

Таким чином, таблиця виконує функцію інтелектуального підказувача, зменшуючи навантаження на користувача, усуваючи необхідність самостійного пошуку сервісів, а також підвищує зручність і оперативність реагування на технічні несправності пристроїв.

Таблиця 2.2 - Таблиця `service_centers` — довідник сервісних центрів

Назва поля	Тип даних	Призначення
<code>id</code>	<code>integer (PK)</code>	Унікальний ідентифікатор центру
<code>brand</code>	<code>text</code>	Назва бренду, який обслуговує цей центр (наприклад, <i>Lenovo</i> , <i>Asus</i> тощо).
<code>city</code>	<code>text</code>	Місто розташування сервісного центру.
<code>name</code>	<code>text</code>	Назва сервісного центру (наприклад: “ <i>Asus Центр</i> ”).
<code>phone</code>	<code>text</code>	Контактний номер телефону.
<code>address</code>	<code>text</code>	Повна поштова адреса центру.

Коли користувач позначає пристрій як зламаний, програма перевіряє, чи існує відповідний сервісний центр для бренду цього пристрою. Якщо такий є — програма виводить інформацію про нього. Якщо ж центр відсутній у таблиці — користувач отримує повідомлення, що сервіс для цього бренду недоступний.

Зв'язок між таблицями

Між таблицями `devices` та `service_centers` встановлено логічний зв'язок за полем `brand`. Тобто, бренд у таблиці пристроїв використовується для пошуку відповідного запису у довіднику сервісних центрів.

Цей підхід був обраний через простоту реалізації та відсутність необхідності в жорсткому зовнішньому ключі. У майбутньому, якщо система буде розширюватись — можна реалізувати справжній зв'язок типу `foreign key` на рівні бази, із окремою таблицею брендів і `brand_id` як ключем.

Обґрунтування вибору PostgreSQL

1. Ця реалізації реляційної бази даних у проєкті було обрано систему управління базами даних PostgreSQL, яка є сучасною, функціональною, надійною та відкритою (open-source). Нижче наведено основні аргументи, які обґрунтовують це рішення:
2. Відкритий код та безкоштовна ліцензія. PostgreSQL є безкоштовним продуктом з відкритим вихідним кодом, що дозволяє використовувати його без додаткових ліцензійних витрат як у навчальних, так і в комерційних проєктах. Це знижує загальну вартість розробки програмного забезпечення.
3. Надійність, стабільність і перевіреність. PostgreSQL існує вже понад 30 років і вважається однією з найстабільніших СУБД, яка активно використовується у великих промислових і наукових системах. Її механізми транзакцій, журналювання змін (WAL), підтримка відновлення після збоїв і контроль цілісності даних роблять її надзвичайно надійною для відповідального зберігання інформації.
4. Гнучка підтримка типів даних. PostgreSQL дозволяє використовувати всі необхідні типи даних для реалізації таблиць у проєкті: `text`, `date`, `integer`, `null` та інші. Це дозволяє створити точну модель предметної області без обмежень, які іноді зустрічаються в інших СУБД.
5. Масштабованість та продуктивність. У разі необхідності система може бути розгорнута на сервері з великою кількістю підключень і великою

базою записів. PostgreSQL чудово масштабується як вертикально (під час збільшення навантаження на один сервер), так і горизонтально (розподіленням бази по кількох вузлах).

6. Зручна інтеграція з .NET та C#. Завдяки підтримці бібліотеки Npgsql для C#, з'єднання з PostgreSQL встановлюється швидко та стабільно. Це дозволяє безпосередньо виконувати SQL-запити, додавати або зчитувати інформацію з таблиць бази, оновлювати дані або виконувати фільтрацію на рівні запитів.
7. Активна спільнота та документація. PostgreSQL має активну спільноту користувачів, регулярні оновлення безпеки, повну документацію англійською мовою і підтримку великої кількості сторонніх інструментів та візуальних оболонок (pgAdmin, DBeaver тощо).
8. Завдяки переліченим факторам PostgreSQL було визнано найбільш відповідною системою для реалізації зберігання та обробки даних у межах цього проєкту.

2.4 Опис графічного інтерфейсу користувача Form1.cs

Рисунок 2.1 – Зовнішній вигляд вікна вводу пристрою користувачем

Елементи інтерфейсу: опис та логіка взаємодії

Форма складається з наступних основних компонентів:

- Поле "Назва пристрою" — текстове поле (TextBox), яке приймає назву пристрою, що додається до системи. Тут вводиться будь-яке ідентифікаційне ім'я моделі, яке користувач вважає зручним: "Lenovo ThinkPad X1 Carbon", "Samsung Galaxy S20+", "Ноутбук для навчання", тощо. Це поле є обов'язковим для заповнення.
- Випадаючий список брендів (ComboBox) — містить фіксований перелік найбільш розповсюджених брендів техніки (Asus, Acer, Lenovo, HP, Dell, Apple, Samsung, MSI, Xiaomi, Huawei). Даний список не допускає вільного вводу тексту, тим самим унеможливорюючи помилки у назвах та дозволяючи однозначно зіставити пристрій з відповідним сервісним центром у подальшій логіці програми.
- Поле вибору дати покупки (DateTimePicker) — дає можливість швидко та зручно вказати точну дату, коли було придбано пристрій. Завдяки зручному календарному віджету, користувач не вводить дату вручну, що усуває потенційні помилки формату.
- Поле введення гарантії у місяцях (NumericUpDown) — приймає числове значення у межах від 1 до 60, що відповідає допустимим гарантійним строкам більшості електронної техніки. Значення обмежується на рівні елемента, тому введення недопустимих даних (наприклад, "0" або "1000") повністю виключається.
- Кнопка "Додати чек" — реалізована для завантаження підтверджувального документа (фото, скан або PDF-файл), що засвідчує факт покупки пристрою. Після вибору файлу, його шлях зберігається у змінну receiptPath, яка передається до БД. Кнопка змінює підпис на "Чек додано ✓", що надає зворотний зв'язок користувачу.
- Кнопка "Розрахувати гарантію" — після натискання автоматично розраховується дата завершення гарантії на основі дати покупки та обраної тривалості. Результат відображається нижче у вигляді інформативного підпису, наприклад: "Гарантія діє до: 18.12.2025".
- Кнопка "Зберегти пристрій" — здійснює валідацію усіх вхідних полів. Якщо хоча б одне з них не заповнене — користувач отримає попередження. У разі успішного заповнення дані формуються у вигляді об'єкта Device і відправляються до бази даних за допомогою SQL-запиту

через бібліотеку Npgsql.

- Кнопка "Переглянути пристрої" — відкриває інше вікно з таблицею усіх збережених пристроїв, включно з інформацією про статус, гарантію та можливість перегляду прикріпленого чеку.

Механізми перевірки та інформування користувача

Перед збереженням реалізовано низку додаткових перевірок:

- Якщо гарантія вже завершена — програма не блокує збереження, але інформує користувача, пропонуючи підтвердити дію або відмовитися від додавання пристрою.
- Якщо до завершення гарантії залишилося 30 днів або менше, програма повідомляє про це через вікно з інформацією, даючи користувачу змогу заздалегідь звернутися до сервісного центру.
- Якщо все заповнено коректно — дані записуються в таблицю devices, при цьому форма очищується для подальшого введення нових пристроїв.

Візуальний стиль форми

Зовнішній вигляд форми сформований відповідно до принципів мінімалізму та юзер-френдлі дизайну:

- Основний фон — світло-сірий (WhiteSmoke), який не втомлює очі.
- Всі текстові елементи — у стандартному шрифті Segoe UI, розмір 10 pt, що забезпечує гарну читабельність.
- Кнопки виконані у кольоровій диференціації:
 - Синій — для розрахунків і переходів.
 - Зелений — для підтвердження (збереження).
 - Світло-блакитний — для нейтральних функцій (розрахунок, додавання чека).

Таке оформлення сприяє швидкому візуальному орієнтуванню у функціоналі навіть для недосвідченого користувача.

Таким чином, вікно Form1 є центральним елементом введення інформації у програмі, поєднує зручний інтерфейс, захист від помилок та автоматичні підказки. Воно забезпечує логічний початок для подальшої роботи системи, закладаючи базові дані для усіх наступних обробок — від гарантійного моніторингу до аналітики сервісів.

2.5 Опис графічного інтерфейсу користувача FormDevicesList.cs

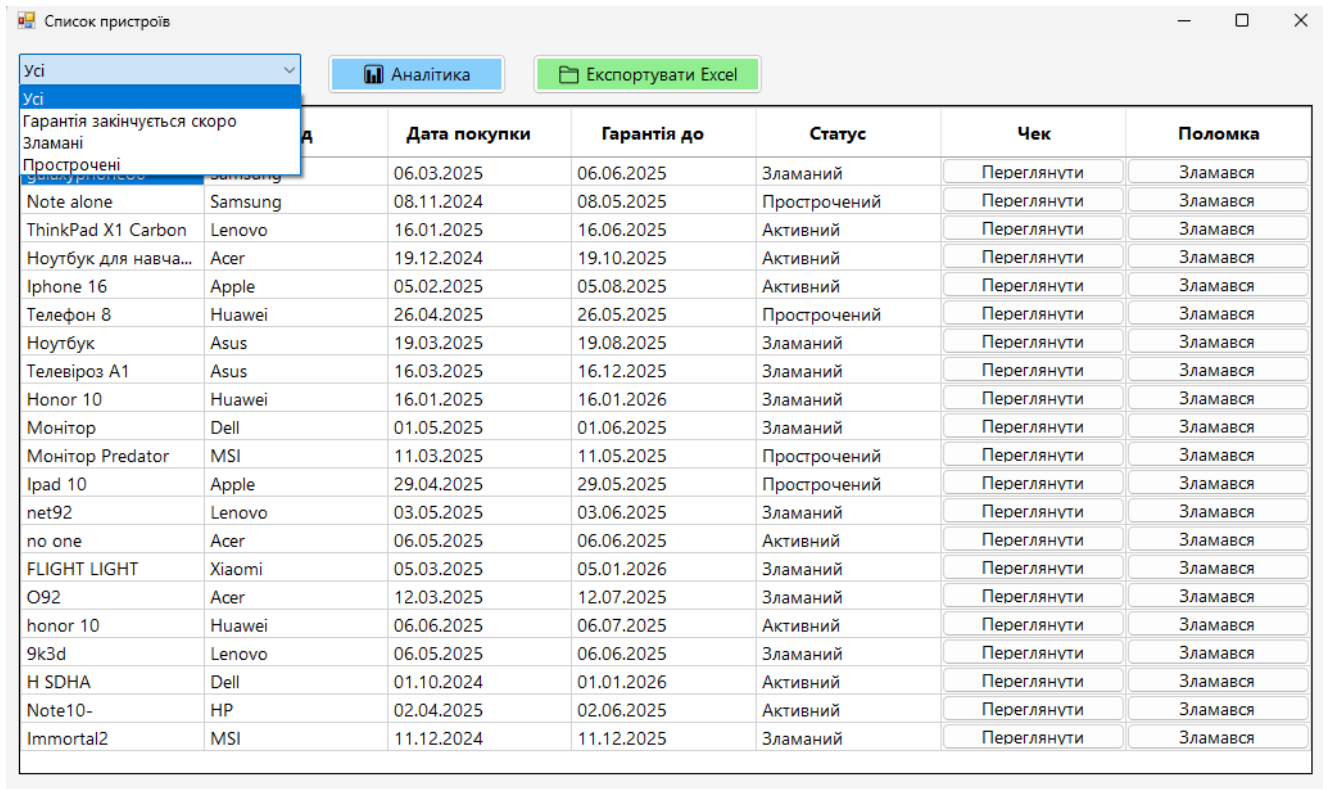


Рисунок 2.2 – Зовнішній вигляд загального перегляду та зміни даних про пристрої.

Після додавання даних до системи через головну форму введення, наступним логічним етапом у роботі користувача є перегляд і керування всіма пристроями, що зберігаються у базі даних. Для цього у застосунку реалізоване окреме вікно перегляду, код якого зосереджений у файлі FormDevicesList.cs.

Це вікно виконує функцію основного інтерфейсу адміністрування даних, надаючи змогу в зручному табличному форматі ознайомитися з переліком зареєстрованих об'єктів, фільтрувати їх за певними критеріями, переглядати прикріплені чеки, а також змінювати статус пристрою у разі виявлення поломки. Вся логіка реалізована з використанням компонента DataGridView та інтеграції з базою даних PostgreSQL через бібліотеку Npgsql.

Форма створена з урахуванням вимог до юзабіліті: передбачено логічне групування елементів, візуальне виділення кнопок, кольорові маркери для взаємодії, та адаптивність під зміну розмірів вікна.

Основні елементи інтерфейсу та їх функції

У верхній частині вікна розташовані елементи керування: комбінований список для вибору фільтрації, кнопка для перегляду статистики та ще одна — для експорту даних до Excel. Вони розташовані горизонтально та мають різні кольори, що відповідають їх функціональній важливості: аналітика — світло-блакитна, експорт — зелена.

Нижче знаходиться таблиця DataGridView, яка є центральним елементом цього вікна. Вона автоматично масштабується разом із вікном, що дає змогу комфортно переглядати навіть велику кількість записів. Всі стовпці мають чітке форматування, наприклад, заголовки вирівняні по центру і виділені напівжирним шрифтом для кращого сприйняття.

- Таблиця складається з наступних колонок:
- Назва пристрою
- Бренд
- Дата покупки
- Гарантія до (кінцева дата дії гарантії)
- Статус (поточний стан пристрою)
- Кнопка «Переглянути» — відкриває вікно з прикріпленим чеком
- Кнопка «Зламався» — дозволяє позначити пристрій як поламаний, із вказанням дати

Усі дані завантажуються автоматично при відкритті форми або зміні фільтра. Зчитування відбувається через виклик методу GetAllDevicesFromDb() із головної форми, що забезпечує централізований підхід до роботи з базою.

Система фільтрації пристроїв

Однією з важливих складових функціоналу форми перегляду є реалізована система фільтрації. Вона дозволяє не лише переглядати повний перелік пристроїв, але й швидко знаходити ті, які потребують негайної уваги або мають особливий статус. Завдяки зручному випадаючому списку, користувач може обрати відповідну категорію та миттєво побачити лише релевантні записи.

Фільтрація реалізована програмно: при зміні значення у списку, викликається відповідна логіка, що фільтрує повний перелік пристроїв згідно з обраним критерієм. Це значно покращує юзабіліті, дозволяючи уникнути перевантаження таблиці зайвою інформацією.

Опис доступних фільтрів:

- Усі — цей варіант відображає повний список пристроїв, що наявні в базі даних, незалежно від їхнього поточного статусу. Це зручно для початкового огляду всієї інформації або у випадках, коли користувач бажає побачити повну картину без будь-яких обмежень.
- Гарантія закінчується скоро — даний фільтр є превентивним інструментом, який автоматично вибирає пристрої, в яких до завершення гарантійного терміну залишилося 10 або менше днів. Це дає змогу вчасно звернути увагу на такі пристрої, зв'язатися із сервісним центром або вжити інших заходів до закінчення терміну. Такий підхід дозволяє підвищити ефективність управління гарантійним обслуговуванням, зменшити ризик пропуску термінів і знизити потенційні витрати.
- Зламани — цей варіант виводить тільки ті пристрої, які були позначені як поламани. Статус «Зламаний» присвоюється вручну користувачем після виявлення поломки. Фільтрація за цим критерієм дозволяє швидко ідентифікувати техніку, що потребує ремонту або вже недоступна для використання. Це також дає можливість у подальшому проводити аналіз за частотою поломок для окремих брендів, моделей або періодів.
- Прострочені — у цю категорію потрапляють пристрої, для яких термін гарантії вже завершився, але користувач не позначив їх як зламани. Такий статус встановлюється автоматично на основі дати завершення гарантії. Наявність даного фільтра дозволяє швидко знайти пристрої, які вже не покриваються сервісом, і, за необхідності, вжити заходів — наприклад, продовжити гарантію, оновити пристрій або списати його.

Завдяки подібній фільтрації, реалізованій у зрозумілому інтерфейсі, навіть користувач без спеціальної технічної підготовки може легко орієнтуватися в даних, приймати виважені рішення щодо техніки та забезпечувати її своєчасне обслуговування.

Позначення пристрою як зламаного

Однією з ключових функцій, що забезпечує гнучкість та практичну користь застосунок, є можливість вручну фіксувати факт поломки пристрою. Для цього передбачена окрема кнопка під назвою «Зламався», яка інтегрована безпосередньо в таблицю списку пристроїв. Вона розташована у відповідному стовпчику та відображається навпроти кожного пристрою, що перебуває у статусі «Активний».

Після натискання на кнопку користувачу відкривається спеціальне модальне вікно, що містить зручний елемент вибору дати (календар). Це дозволяє користувачу вказати точну дату, коли було виявлено несправність. Такий підхід унеможливорює помилки, які могли б виникнути при ручному вводі дати, а також покращує UX за рахунок наочності.

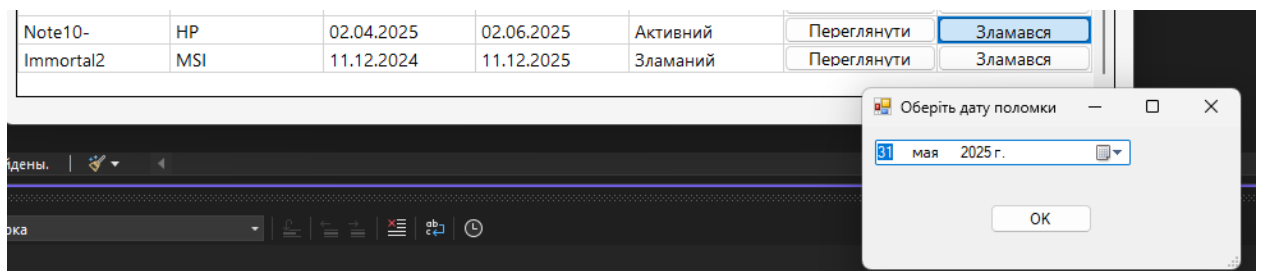


Рисунок 2.3 – Вигляд календаря обрання поломки при натискання кнопки «Зламався»

Після обрання дати програма виконує автоматичну перевірку — чи входить вказана дата поломки до періоду дії гарантії. В залежності від результату цієї перевірки, користувачу надається відповідне повідомлення:

1. Якщо поломка сталася в межах гарантійного терміну, застосунок переходить до наступного кроку — перевірки наявності в базі даних сервісного центру, що обслуговує відповідний бренд пристрою. Якщо такий запис існує, програма виводить повідомлення з детальною інформацією, зокрема назвою сервісного центру, контактним телефоном і фізичною адресою. Це дозволяє користувачу негайно звернутись за допомогою, без необхідності здійснювати додатковий пошук вручну.
2. Якщо сервісний центр не знайдено, система делікатно інформує користувача, що для обраного бренду немає зареєстрованих центрів обслуговування в базі. Така ситуація теж є інформативною, оскільки може слугувати сигналом для подальшого розширення довідника сервісних центрів у системі.
3. Якщо ж гарантія вже завершилася, користувач отримує повідомлення із зазначенням факту завершення гарантії та точної дати, коли вона сплинула. Це дозволяє уникнути непорозумінь та

дає змогу прийняти обґрунтоване рішення — наприклад, звернутись до незалежного сервісного центру або оновити пристрій.

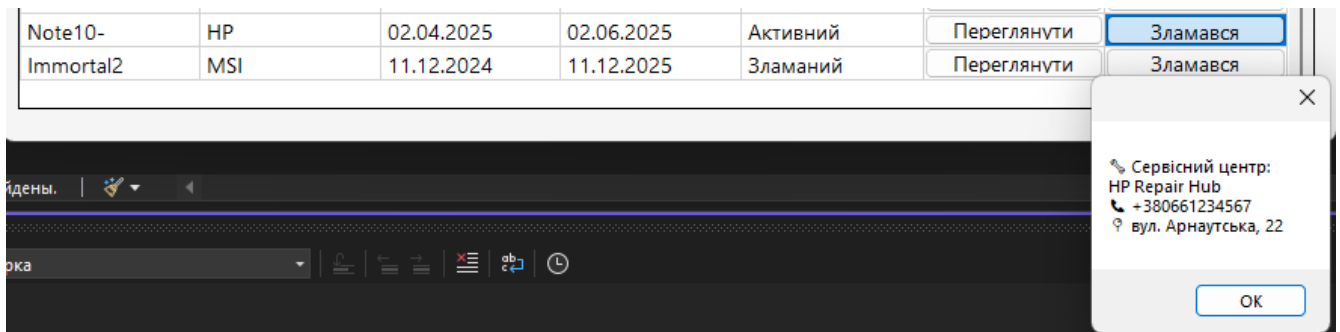


Рисунок 2.4 – Приклад повідомлення якщо гарантійний строк ще діє

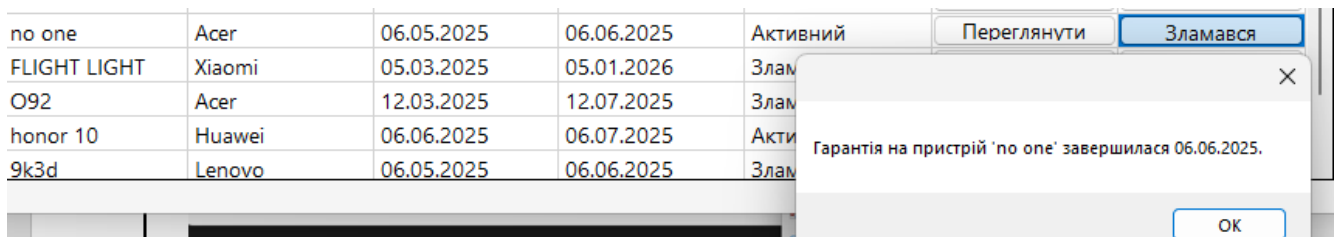


Рисунок 2.5 – Приклад повідомлення якщо гарантійний строк вже закінчився

Усі зміни, пов'язані з фактом поломки, автоматично зберігаються у базі даних. Зокрема:

Поле status у таблиці devices змінюється з «Активний» на «Зламаний».

Поле fault_date отримує значення вибраної користувачем дати.

Це оновлення є критично важливим для побудови подальшої аналітики по поломках — зокрема, для виведення статистики по брендах, середнього терміну безвідмовної роботи техніки, частоті звернень тощо. Таким чином, дана функція виконує не лише практичну роль у роботі користувача, а й служить основою для збору та аналізу інформації на стратегічному рівні.

Перегляд прикріпленого чеку

У межах реалізації користувацького інтерфейсу важливою функцією є можливість перегляду прикріпленого зображення чеку до кожного пристрою. Для цього у таблиці реалізовано окрему кнопку «Переглянути», яка з'являється навпроти кожного запису. Після натискання на неї відкривається нове вікно, де відображається завантажене зображення у повному масштабі. Передбачено також можливість масштабування для комфортного перегляду деталей.

Така реалізація дозволяє не лише оперативно перевіряти ключову інформацію про покупку — дату, серійний номер тощо — а й мати надійне візуальне підтвердження під час гарантійних суперечок або звернень до сервісного центру. Оцифрований чек зберігається у програмі, що знижує ризик його фізичної втрати. Якщо шлях до зображення не вказаний або сам файл відсутній, користувач бачить відповідне повідомлення — це забезпечує зворотний зв'язок і підвищує надійність системи.

Експорт до Excel

Для покращення практичного використання програми реалізовано функцію експорту даних у формат Excel. Вона активується через спеціальну кнопку «Експортувати Excel», що розміщена у верхній частині інтерфейсу. Після її натискання відкривається стандартне діалогове вікно збереження файлу, де користувач обирає місце для збереження звіту.

Програма автоматично формує Excel-документ, у якому всі записи з таблиці конвертуються у структурований вигляд. Дані розподіляються по стовпцях (назва, бренд, дати покупки та завершення гарантії, статус), і кожен рядок відповідає окремому пристрою. Для реалізації використовується бібліотека ClosedXML, яка забезпечує стабільну генерацію файлу з правильним форматуванням.

Це дозволяє не лише зберігати резервні копії, але й створювати звіти, ділитися інформацією у зручному для аналізу форматі, а також використовувати програму в офісному або корпоративному середовищі.

Форма перегляду пристроїв об'єднує ключові функції управління в одному інтерфейсі. Вона дозволяє швидко фільтрувати пристрої за статусом, контролювати строки дії гарантій, позначати поломки та отримувати сервісну інформацію. Користувач має змогу переглядати прикріплені чеки, а також створювати зовнішні копії даних у форматі Excel — усе це в рамках зручної й інтуїтивної панелі керування.

З технічної точки зору, реалізація демонструє ефективну взаємодію між інтерфейсом, логікою програми та базою даних. Автоматизація обробки статусів, перевірок файлів і експорту робить цю систему цілісною та професійною, здатною відповідати реальним потребам користувача.

2.6 Опис графічного інтерфейсу користувача FormAnalytics.cs

Графічне вікно з аналітикою — це окрема функціональна частина програмного забезпечення, що виконує ключову роль у наданні користувачу зведеної візуальної інформації щодо усіх пристроїв, зареєстрованих у системі. Його призначення полягає не лише у відображенні фактів, а й у наданні можливості для глибшого розуміння тенденцій, закономірностей та проблем, що виникають під час експлуатації електронних пристроїв.

Цей розділ є логічним завершенням ланцюга роботи із даними: після збору й збереження інформації на етапі введення, користувач отримує змогу зручно переглядати аналітичні зведення в інтерактивному та зрозумілому вигляді. Завдяки графікам, вся складна інформація сприймається значно легше, особливо при роботі з великими обсягами даних.

Інтерфейс реалізований у вигляді елемента TabControl, що дозволяє впорядкувати кілька типів діаграм по вкладках. Такий підхід є доцільним з

					КНУ.РБ.123.25.01. ПІС	Арк.
Арк.	№ документа	Підпис	Дата			

точки зору зручності та інтуїтивної навігації. Користувач має змогу у будь-який момент перемикатися між окремими розділами, не втрачаючи загального контексту. Всі візуальні елементи виконані у гармонійному стилі, з єдиними шрифтами, відступами та кольоровою палітрою, що створює цілісне враження про інтерфейс.

Перша вкладка: "Зламани пристрої"

Одна з найважливіших аналітичних метрик — це оцінка надійності пристроїв за брендами. Саме для цього призначена перша вкладка, яка демонструє розподіл зламаних пристроїв у вигляді стовпчикової діаграми. Така форма подачі дає змогу миттєво побачити, які виробники частіше потрапляють у категорію несправних, і зробити відповідні висновки щодо їх якості або потреби у частішому сервісному обслуговуванні.

Візуально діаграма побудована так, щоб бути максимально зрозумілою навіть для користувачів без технічної освіти. Основні особливості:

- Кожен стовпчик має окреме кольорове оформлення, найчастіше у відтінках синього — це не тільки додає контрасту, а й дозволяє легко візуально розрізнити бренди.
- Над кожним стовпцем виводиться цифровий підпис з точним значенням кількості зламаних пристроїв, що зменшує потребу в додатковому аналізі.
- Горизонтальна вісь (X) відображає назви брендів, а вертикальна (Y) — кількість зафіксованих поломок.
- Шрифт підписів діаграми підібраний таким чином, щоб залишався читабельним навіть на менших розширеннях екрану.

Важливо зазначити, що дані для цієї діаграми генеруються динамічно на основі вмісту бази даних. Це означає, що кожне додавання нового пристрою, зміна його статусу або оновлення інформації миттєво враховується при побудові графіка. Така поведінка гарантує актуальність аналітики і дозволяє ефективно контролювати загальний технічний стан усього парку зареєстрованих пристроїв.

Даний підхід особливо корисний для:

- визначення проблемних брендів, які найчастіше виходять з ладу;
- аналізу тенденцій по гарантійному обслуговуванню; прийняття рішень щодо закупівель нової техніки або налагодження зв'язків із сервісними

					КНУ.РБ.123.25.01. ПС	Арк.
Арк.	№ документа	Підпис	Дата			

центрами.

- Ця вкладка — потужний інструмент моніторингу, який поєднує в собі простоту візуалізації з глибокою практичною цінністю для кінцевого користувача.

Друга частина вікна аналітики надає користувачу глибше розуміння загальних характеристик техніки, зокрема середньої тривалості гарантії, яку мають пристрої різних брендів. Це дозволяє оцінити, наскільки виробники пропонують конкурентні умови та наскільки довгострокову відповідальність вони несуть за свою продукцію.

Головною метою цієї діаграми є порівняння брендів між собою за тривалістю гарантійного покриття, що може бути важливим фактором при виборі нової техніки або у формуванні рекомендацій для користувача. Чим довша середня гарантія — тим більше довіри до бренду може сформуватися, оскільки це часто свідчить про впевненість у якості свого продукту.

Інтерфейс побудовано аналогічно до першої вкладки, що забезпечує послідовність та зручність сприйняття. Графік реалізовано у вигляді вертикальних стовпців, які представляють усереднені значення гарантійних строків (у місяцях) для кожного бренду. Така візуалізація дозволяє легко помітити навіть невеликі відмінності між показниками.

Відображення передбачає:

- Кольорове виділення кожного стовпчика (відтінки зеленого), що підкреслює позитивний характер показника — тривалість гарантії розглядається як перевага.
- Цифрове маркування на кожній колонці, яке показує точне значення (наприклад, 11.5 місяців), завдяки чому користувач не потребує візуальних оцінок або ручного підрахунку.
- Однаковий масштаб осей і стилі підписів, що забезпечує зручне перемикання між вкладками без втрати орієнтації в інтерфейсі.

Усі дані, як і в попередньому графіку, формуються автоматично на основі записів у базі даних. Система обчислює середнє арифметичне значення для кожного бренду, беручи до уваги фактичні дати покупки та закінчення гарантії. Це дозволяє не тільки побачити, які бренди надають довші гарантії, а й відслідковувати зміни цієї тенденції у часі, якщо база оновлюється регулярно.

Практична користь цієї інформації очевидна: керівник компанії або ІТ-відділ, який закуповує техніку, може опиратись на цю статистику, щоб обрати більш відповідальних постачальників. Крім того, це створює підґрунтя для побудови довгострокових партнерських стратегій із виробниками або сервісними мережами.

Цей аналітичний блок значно розширює можливості системи — від звичайного обліку пристроїв до системного прийняття рішень, підкріпленого реальними цифрами. Завдяки простоті візуалізації та гнучкості у поданні інформації, користувач має доступ до потужного інструменту, що допомагає не просто «бачити», а й аналізувати та прогнозувати технічний стан і цінність брендів.

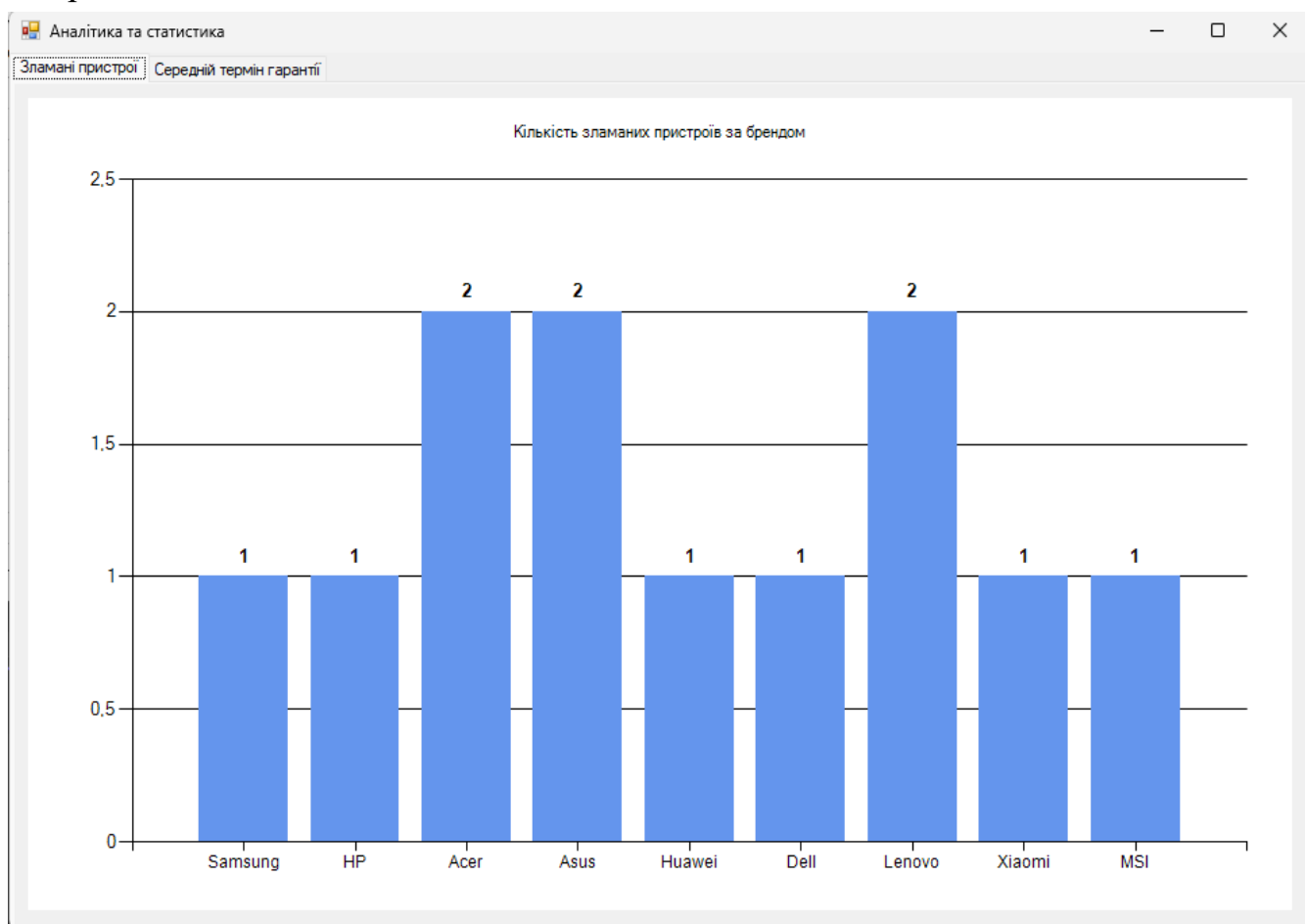


Рисунок 2.6 – Візуальний вигляд вікна статистики у програмі щодо «зламаних пристроях» по брендам

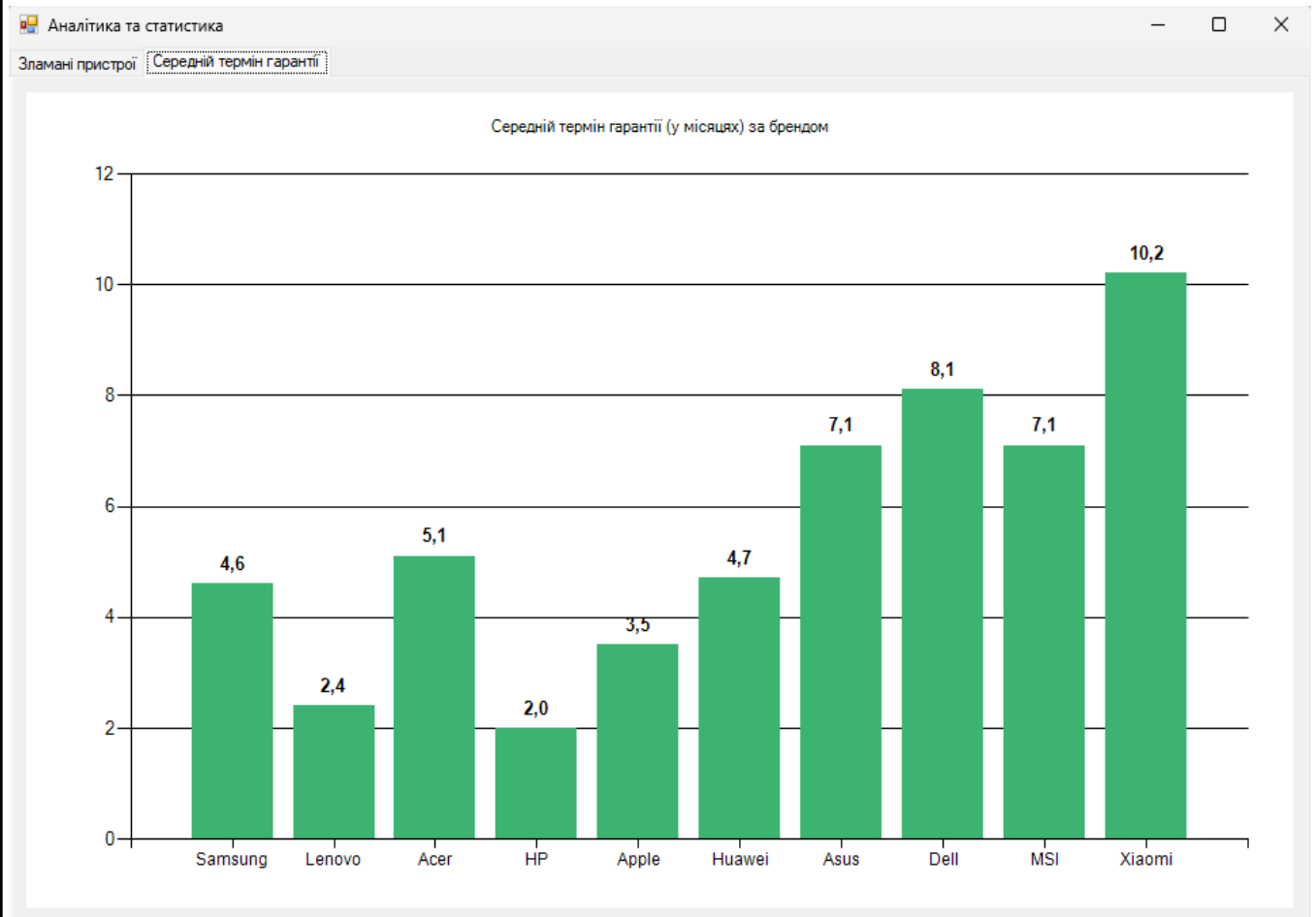


Рисунок 2.7 – Візуальний вигляд вікна статистики у програмі щодо «середнього терміну гарантії» по брендам

Загальна структура та враження

Аналітичне вікно програми розроблене з урахуванням принципів сучасного UI-дизайну — воно не лише естетично привабливе, але й функціонально зручне у використанні. З перших секунд взаємодії інтерфейс створює позитивне враження завдяки структурованості, логічному розміщенню елементів і спокійній кольоровій палітрі, що не перевантажує зір.

В основі вікна лежить компонент TabControl, який дозволяє користувачеві інтуїтивно перемикатися між вкладками. Цей підхід значно покращує ергономіку інтерфейсу — інформація поділена за змістовими блоками, що виключає необхідність гортати одне перевантажене вікно. Кожна вкладка має чітку назву, що відображає її зміст, а простір усередині вкладок використовує екран максимально ефективно, не залишаючи зайвих пустих зон.

Усі графічні елементи масштабуються відповідно до розмірів вікна. При зміні розміру форми діаграми автоматично підлаштовуються під нові параметри — це особливо зручно для користувачів, які працюють на різних моніторах чи у віконному режимі. Завдяки цьому забезпечується адаптивність і доступність візуального контенту незалежно від роздільної здатності екрана.

Шрифти підібрані таким чином, щоб залишатися читабельними на будь-якому фоні, з перевагою для середнього розміру — що знижує навантаження на очі під час тривалого перегляду. Кольорова гама графіків, підписів і фону є стриманою, але водночас контрастною достатньо для швидкого візуального сприйняття.

Особливу увагу було приділено повній автоматизації аналітики. Користувач не вводить жодних параметрів вручну — усі розрахунки здійснюються автоматично на основі поточних записів у базі даних. Це означає, що будь-які зміни в інформації (наприклад, додавання зламаного пристрою чи зміна статусу) миттєво враховуються в діаграмах без потреби перезапуску застосунку або додаткових дій з боку користувача.

Такий підхід робить аналітичне вікно не просто статичним відображенням, а динамічним інструментом, який завжди показує актуальний стан системи. Завдяки цьому користувач отримує точну картину технічного парку, не вдаючись до зовнішніх засобів обробки даних.

У підсумку, це вікно є не лише допоміжним інструментом, а повноцінним аналітичним центром у межах застосунку, який надає зрозумілу, швидко і візуально зручну інформацію для прийняття обґрунтованих рішень щодо техніки та її обслуговування.

Вибір технології Windows Forms

Як можна було побачити у попередніх підрозділах, графічний інтерфейс користувача реалізовано з використанням технології Windows Forms (WinForms). Це одне з найдавніших та найбільш стабільних рішень у складі платформи .NET, яке на практиці продемонструвало свою ефективність у створенні настільних додатків з класичною структурою вікон, форм і інтерактивних елементів керування.

Основною причиною вибору саме WinForms стала оптимальна збалансованість між простотою реалізації та достатньою функціональністю для потреб цього проєкту. В рамках розробки додатку, який не потребує складної графіки або анімації, але вимагає стабільного і надійного відображення елементів (текстових полів, кнопок, таблиць, графіків), WinForms виявився ідеальним варіантом. Його архітектура дозволяє швидко створювати форми, додавати компоненти та реалізовувати обробку подій без надмірної кількості коду.

Крім того, WinForms має вбудовану підтримку масштабування вікон, гнучке налаштування візуальних елементів, зручну систему подій та добру інтеграцію з базами даних. У цьому застосунку реалізовано кілька форм (додавання пристроїв, перегляд таблиці, аналітика), які не лише мають різну структуру, а й динамічно працюють з базою PostgreSQL. Завдяки бібліотеці

Npgsql, WinForms чудово поєднується з цією СУБД — дані легко передаються, читаються, оновлюються, що спрощує реалізацію всієї бізнес-логіки.

Не менш важливою перевагою є наявність багаторічного досвіду використання WinForms у навчальних та комерційних проектах. У порівнянні з іншими рішеннями на зразок WPF, UWP або сторонніх кросплатформових бібліотек (наприклад, Qt або Electron), WinForms не потребує складної конфігурації середовища та дозволяє зосередитись безпосередньо на функціоналі.

Також слід зазначити, що усі ключові елементи взаємодії в даному застосунку реалізовані із врахуванням UX-принципів, а WinForms надає повний контроль над розміщенням та поведінкою компонентів. Користувач отримує інтуїтивний, знайомий інтерфейс, який швидко освоюється навіть без попередньої підготовки.

Таким чином, використання Windows Forms у даному проекті є цілком виправданим, оскільки ця технологія повністю покриває функціональні та візуальні вимоги програми, забезпечує швидкість розробки, стабільність виконання та просту підтримку коду в майбутньому.

2.7 Алгоритми роботи системи по фрагментам коду

Розрахунок дати завершення гарантії

Одним із ключових алгоритмів, реалізованих у програмі, є обчислення дати завершення гарантії пристрою на основі початкової дати покупки та тривалості гарантії, заданої користувачем. Такий підхід дозволяє точно визначити період обслуговування й надалі орієнтувати всі інші функціональні блоки на ці дані — включаючи фільтрацію, статуси пристроїв, попередження та аналітику.

Інтерфейсна логіка

У графічному вікні введення пристрою (Form1.cs) користувач бачить два поля, які впливають на гарантійний термін

Дата покупки — обирається через DateTimePicker і є початковою точкою розрахунку.

Гарантія (у місяцях) — задається у NumericUpDown, значення якої може варіюватися від 1 до 60.

Після введення даних користувач натискає кнопку «Розрахувати гарантію», після чого програма миттєво обчислює дату завершення гарантії й виводить її у зручному для читання вигляді.

Алгоритмічна частина

					КНУ.РБ.123.25.01. ПІС	Арк.
Арк.	№ документа	Підпис	Дата			

Розрахунок виконується в методі BtnCalculate_Click, який прив'язаний до події натискання відповідної кнопки. Його код виглядає наступним чином:

```
private void BtnCalculate_Click(object sender,
EventArgs e)
{
    calculatedWarrantyEnd =
    dtpPurchaseDate.Value.AddMonths((int)numWarranty.Value)
;
    lblWarrantyEnd.Text = $"Гарантія діє до:
{calculatedWarrantyEnd.ToShortDateString()}";
}
```

Цей фрагмент можна розкласти на два основні логічні кроки:

1. Визначення дати завершення гарантії:

```
calculatedWarrantyEnd =
dtpPurchaseDate.Value.AddMonths((int)numWarranty.Value)
;
```

dtpPurchaseDate.Value — повертає дату, яку обрав користувач як дату покупки.

numWarranty.Value — кількість місяців гарантії, введена користувачем.

Метод AddMonths() додає до обраної дати відповідну кількість місяців.

Результат зберігається у змінній calculatedWarrantyEnd, яка пізніше передається в базу.

2. Відображення результату на інтерфейсі:

```
lblWarrantyEnd.Text = $"Гарантія діє до:
{calculatedWarrantyEnd.ToShortDateString()}";
```

Після обчислення результат одразу виводиться на форму у вигляді зрозумілого повідомлення, наприклад:

“Гарантія діє до: 24.10.2025”

Значення цього алгоритму для всієї системи

Ця дата є основним контрольним параметром, на основі якого:

- Система визначає статус пристрою (активний, прострочений, зламаний).
- Генеруються автоматичні попередження про завершення гарантії.
- Визначається, чи можливо звернення до сервісного центру.
- Ведеться статистика середньої тривалості гарантій по брендам.

Таким чином, на перший погляд простий обчислювальний алгоритм має ключову роль у формуванні логіки та функціональності всієї програми.

Перевірка пристрою на прострочення гарантії

Однією з базових функцій інформаційної системи є автоматичне визначення, чи вийшов пристрій за межі гарантійного періоду. Цей процес відбувається без участі користувача та реалізований у модулі перегляду пристроїв — `FormDevicesList.cs`, у межах функції `LoadData()`.

Алгоритм перевірки виконується щоразу, коли користувач відкриває вікно з пристроями або змінює фільтр на формі. Це дозволяє гарантувати актуальність інформації про статус кожного пристрою, навіть якщо минуло кілька днів з моменту його додавання в базу.

Основна логіка перевірки

У межах циклу, який проходиться по всіх пристроях (фрагмент нижче), відбувається наступне:

```
foreach (var d in filtered)
{
    if (d.Status == "Активний" && d.WarrantyEndDate
        < DateTime.Now)
    {
        UpdateStatusToExpired(d);
        d.Status = "Прострочений";
    }

    int rowIndex = dgv.Rows.Add(d.Name, d.Brand,
        d.PurchaseDate.ToShortDateString(),
        d.WarrantyEndDate.ToShortDateString(), d.Status);
    dgv.Rows[rowIndex].Tag = d;
}
```

Тобто система порівнює поточну дату (`DateTime.Now`) з датою завершення гарантії пристрою. Якщо гарантія вже вичерпана, а статус пристрою все ще "Активний", то такий запис автоматично оновлюється і набуває статусу "Прострочений".

Цей процес забезпечує повну синхронізацію стану кожного запису з реальним часом, без потреби ручного втручання з боку користувача.

Збереження змін у базу даних

Після визначення, що гарантія завершена, зміни одразу фіксуються в таблиці `devices` через наступний виклик:

```

private void UpdateStatusToExpired(Device device)
{
    using var conn = new
NpgsqlConnection("Host=localhost;...");
    conn.Open();

    using var cmd = new NpgsqlCommand("UPDATE
devices SET status='Прострочений' WHERE name=@name AND
purchase_date=@pd", conn);
    cmd.Parameters.AddWithValue("name",
device.Name);
    cmd.Parameters.AddWithValue("pd",
device.PurchaseDate);
    cmd.ExecuteNonQuery();
}

```

Така конструкція гарантує, що при повторному відкритті програми або оновленні списку пристроїв, система покаже коректний статус без додаткової обробки.

Переваги реалізації:

- Повна автоматизація — користувачу не потрібно вручну оновлювати статуси.
- Актуальність — інформація відповідає реальному часу.
- Надійність — жорстка прив'язка до дати покупки, яка разом з гарантійним терміном формує чіткий контрольний дедлайн.
- Безперервність обробки — зміна статусу виконується в тлі, разом із побудовою таблиці.

Таким чином, функція перевірки прострочених гарантій стала важливою частиною логіки системи, що забезпечує точність, актуальність і зменшує навантаження на користувача. Цей механізм тісно інтегрується з іншими модулями, зокрема аналітикою, фільтрами й обробкою полумок.

Як система визначає, що гарантія скоро закінчується

Ця логіка реалізована у файлі FormDevicesList.cs, зокрема в методі LoadData(). Під час завантаження даних з бази та відображення їх у таблиці (DataGridView), система проводить перевірку дати завершення гарантії для кожного пристрою.

Умова перевірки:

```

selected switch
{
    "Гарантія закінчується скоро" =>
allDevices.Where(d =>
    d.Status == "Активний" &&
    d.WarrantyEndDate >= DateTime.Now &&
    (d.WarrantyEndDate -
DateTime.Now).TotalDays <= 10
    ).ToList(),

```

Пояснення логіки:

- `d.Status` — це властивість класу `Device`, яка зберігає поточний стан пристрою (наприклад: "Активний", "Зламаний", "Прострочений"). Ці значення приходять із бази даних (таблиці `devices`) при виконанні методу `GetAllDevicesFromDb()` у `Form1.cs`.
- `d.WarrantyEndDate` — це дата, до якої діє гарантія. Також приходить з БД.
- `(d.WarrantyEndDate - DateTime.Now).TotalDays <= 10` — система розраховує, скільки днів залишилося до кінця гарантії. Якщо менше або дорівнює 10, то вважається, що гарантія "закінчується скоро".
- `d.WarrantyEndDate >= DateTime.Now` — гарантія ще не завершена, тобто пристрій не прострочений.
- `d.Status == "Активний"` — гарантія діє, пристрій не зламаний і не прострочений.

Таким чином, пристрої, що відповідають цим умовам, потрапляють у список "Гарантія закінчується скоро".

Прив'язка до елементів інтерфейсу:

Ця перевірка активується автоматично при виборі відповідного фільтра в комбобоксі `cbFilter.SelectedItem == "Гарантія закінчується скоро"`.

Після цього список `filtered` наповнюється лише тими пристроями, в яких залишилося ≤ 10 днів до кінця гарантії. Далі цей список передається в `DataGridView (dgv.Rows.Add(...))`.

Як зберігаються та обробляються дати поломки

Загальна ідея

Функціональність обробки дати поломки реалізована через кнопку "Зламався", що додається до таблиці пристроїв (`DataGridView`). Коли користувач натискає цю кнопку, відкривається діалогове вікно для вибору

					КНУ.РБ.123.25.01. ППС	Арк.
Арк.	№ документа	Підпис	Дата			

дати поломки. Після цього дата записується у базу даних, а статус пристрою змінюється на "Зламаний".

Візуальний елемент у DataGridView

У методі LoadData() формується таблиця з пристроями. В ній додається колонка-кнопка:

```
var btnFault = new DataGridViewButtonColumn()
{
    Name = "btnMarkBroken",
    HeaderText = "Поломка",
    Text = "Зламався",
    UseColumnTextForButtonValue = true,
    Width = 120
};
dgv.Columns.Add(btnFault);
```

Це означає, що для кожного рядка (пристрою) в таблиці буде своя кнопка "Зламався", яка реагуватиме індивідуально на відповідний пристрій.

Обробка натискання на кнопку "Зламався"

Основна логіка реалізована в методі:

```
private void Dgv_CellContentClick(object sender,
DataGridViewCellEventArgs e)
```

Коли користувач натискає кнопку, відбувається таке:

1. Визначається пристрій, до якого належить натиснутий рядок:

```
var row = dgv.Rows[e.RowIndex];
var device = (Device)row.Tag;
```

Поле .Tag використовується для зберігання об'єкта Device, прив'язаного до конкретного рядка таблиці. Сам клас Device описано у Form1.cs і він містить усі поля, зокрема Name, Brand, PurchaseDate, WarrantyEndDate, FaultDate, Status тощо.

2. Перевірка поточного статусу пристрою:

```
if (device.Status == "Зламаний")
{
    MessageBox.Show("Цей пристрій вже позначений як зламаний.");
    return;
}
```

Ця умова запобігає повторному позначенню вже зламаного пристрою.

Далі, додатково перевіряється статус "Прострочений", щоб уникнути марного запису для пристроїв із завершеною гарантією:

```
if (device.Status == "Прострочений")
{
    MessageBox.Show("Цей пристрій прострочений, гарантія завершилась.");
    return;
}
```

Вибір дати поломки

Користувачу відкривається невелика форма, де можна обрати дату поломки:

```
DateTimePicker picker = new DateTimePicker { Value
= DateTime.Now };
Form prompt = new Form { Width = 300, Height = 150,
Text = "Оберіть дату поломки" };
```

Цей підхід надає гнучкість — користувач сам вирішує, коли саме стався збій у роботі пристрою.

Перевірка гарантії на момент поломки

Коли дата вибрана, код перевіряє, чи пристрій ще перебував на гарантії:

```
string result = faultDate <= device.WarrantyEndDate
? FindServiceCenter(device.Brand) ?? $"Ваш
пристрій '{device.Name}' ще на гарантії! Але сервісного
центру не знайдено."
```

```
: $"Гарантія на пристрій '{device.Name}'
завершилася
{device.WarrantyEndDate.ToShortDateString()}";
```

Метод `FindServiceCenter(string brand)` виконує SQL-запит до таблиці `service_centers` і підбирає відповідний центр для бренду. Якщо дані не знайдені — виводиться базове повідомлення.

Запис дати у базу даних

Оновлення даних у таблиці `devices` виконується методом `UpdateDeviceAsBroken()`:

```
private void UpdateDeviceAsBroken(Device device,
DateTime faultDate)
{
    using var conn = new NpgsqlConnection("...");
    conn.Open();

    using var cmd = new NpgsqlCommand("UPDATE
devices SET status='Зламаний', fault_date=@fault WHERE
name=@name AND purchase_date=@pd", conn);
    cmd.Parameters.AddWithValue("fault",
faultDate);
    cmd.Parameters.AddWithValue("name",
device.Name);
    cmd.Parameters.AddWithValue("pd",
device.PurchaseDate);
    cmd.ExecuteNonQuery();
}
```

В якості ідентифікатора рядка в базі використовується назва пристрою + дата покупки, що вважається достатньо унікальною комбінацією для

невеликого застосунку.

Оновлення в інтерфейсі

Після завершення операції викликається метод:

```
LoadData();
```

Це оновлює таблицю пристроїв, перезавантажуючи всі дані з бази, і тепер статус пристрою буде відображено як "Зламаний", а кнопка "Зламався" вже стане неактивною для цього запису.

Підсумок

- Користувач вручну позначає дату поломки.
- Програма перевіряє дію гарантії.
- Статус пристрою змінюється на "Зламаний".
- У базу вноситься дата (fault_date) та новий статус.
- У таблиці пристроїв візуально змінюється відображення.

Така реалізація дозволяє контролювати гарантійні випадки й вести облік поломок з урахуванням реального часу та статусу пристроїв.

Реалізація фільтрації, повідомлень та нагадувань

Загальна ідея

Усі основні механізми фільтрації, сповіщень та візуального нагадування реалізовані на рівні двох основних форм: Form1.cs (вікно додавання пристрою) та FormDevicesList.cs (вікно перегляду списку пристроїв). Ці форми взаємодіють через спільну базу даних і логіку, вбудовану в методи обробки подій.

Фільтрація в FormDevicesList.cs — деталі логіки

При відкритті вікна FormDevicesList викликається метод LoadData():

```
public FormDevicesList()
{
    InitializeComponent();
    this.Text = "Список пристроїв";
    this.Size = new Size(900, 500);
    this.BackColor = Color.WhiteSmoke;
    InitControls();
    LoadData();
}
```

Цей метод:

1. Створює екземпляр Form1, викликає з нього метод GetAllDevicesFromDb() для завантаження всіх пристроїв з БД.

2. Після цього виконує фільтрацію списку `allDevices` згідно з поточним вибором `ComboBox cbFilter`.

```
private void LoadData()
{
    using (var tempForm = new Form1())
    {
        allDevices =
tempForm.GetAllDevicesFromDb();
    }

    string selected =
cbFilter.SelectedItem.ToString();
    var filtered = selected switch
    {
        "Гарантія закінчується скоро" =>
allDevices.Where(d => d.Status == "Активний" &&
d.WarrantyEndDate >= DateTime.Now && (d.WarrantyEndDate
- DateTime.Now).TotalDays <= 10).ToList(),
        "Зламани" => allDevices.Where(d => d.Status
== "Зламаний").ToList(),
        "Прострочені" => allDevices.Where(d =>
d.Status == "Прострочений").ToList(),
        _ => allDevices
    };
};
```

Фільтрація працює через LINQ, і дозволяє швидко виділити пристрої за умовами:

- Залишилось ≤ 10 днів до кінця гарантії.
- Статус — «Зламаний» або «Прострочений».
- Інакше — всі.

Далі сформований список `filtered` відображається в `DataGridView dgv` — табличному компоненті. Кожен рядок даних додається вручну:

```
int rowIndex = dgv.Rows.Add(d.Name, d.Brand,
d.PurchaseDate.ToShortDateString(),
d.WarrantyEndDate.ToShortDateString(), d.Status);
dgv.Rows[rowIndex].Tag = d;
```

Таким чином, при кліку по кнопці в таблиці (наприклад, «Зламався»), програма точно знає, до якого саме об'єкта `Device` належить цей рядок.

Нагадування при додаванні нового пристрою — `Form1.cs`

У `Form1`, після того як користувач вводить дані і натискає «Зберегти пристрій», спрацьовує метод `BtnSave_Click`, який виконує такі кроки:

1. Перевіряє правильність введення (наприклад, чи розраховано гарантію).

2. Створює об'єкт Device.

3. Якщо гарантія вже завершена — питає у користувача підтвердження.

4. Якщо все добре — додає пристрій до БД (InsertDeviceToDb()).

5. І найважливіше — викликає CheckWarrantyAlerts(device):

```
private void BtnSave_Click(object sender, EventArgs
e)
{
    if (txtName.Text == "" || cbBrand.SelectedItem
== null || calculatedWarrantyEnd == DateTime.MinValue)
    {
        MessageBox.Show("Будь ласка, заповніть усі
поля та розрахуйте гарантію.");
        return;
    }

    Device device = new Device
    {
        Name = txtName.Text,
        Brand = cbBrand.SelectedItem.ToString(),
        PurchaseDate = dtpPurchaseDate.Value,
        WarrantyMonths = (int)numWarranty.Value,
        WarrantyEndDate = calculatedWarrantyEnd,
        ReceiptPath = receiptPath
    };

    if (device.WarrantyEndDate < DateTime.Now)
    {
        var result = MessageBox.Show(
            $"Увага! Зазначена гарантія вже
завершилась
({device.WarrantyEndDate.ToShortDateString()}). Додати
пристрій до бази все одно?",
            "Прострочена гарантія",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Warning
        );

        if (result == DialogResult.No)
        {
            MessageBox.Show("Додавання пристрою
скасовано.");
            return;
        }
    }
}
```

```

try
{
    InsertDeviceToDb(device);
    MessageBox.Show("Пристрій додано до бази
    ✓");
}
catch (Exception ex)
{
    MessageBox.Show($"Помилка при додаванні:
    {ex.Message}");
}

```

CheckWarrantyAlerts(device);

```

txtName.Text = "";
cbBrand.SelectedIndex = 0;
lblWarrantyEnd.Text = "Гарантія діє до: -";
btnSelectReceipt.Text = "Додати чек";
receiptPath = "";

```

}

Цей метод розраховує кількість днів до завершення гарантії:

```

private void CheckWarrantyAlerts(Device device)
{

```

```

    if (device.Status != "Активний") return;

```

```

    double daysLeft = (device.WarrantyEndDate -
    DateTime.Now).TotalDays;

```

```

    int roundedDays = (int)Math.Ceiling(daysLeft);

```

.....

І залежно від результату показує користувачу одне з двох повідомлень:

1. Якщо до завершення гарантії залишилось ≤ 30 днів — виводиться попередження.

2. Якщо більше — просто інформується про дату завершення.

Зв'язок між формами

Найважливіше — всі форми працюють з однією базою даних через Npgsql. Тобто, дані, збережені у Form1, автоматично з'являються у FormDevicesList. Зміна статусу пристрою (наприклад, після натискання «Зламався») також оновлює БД, і при повторному відкритті FormAnalytics або FormDevicesList зміни вже враховано.

Функціональні кнопки та їхня взаємодія з фільтрацією

У межах таблиці пристроїв особливу роль відіграють дві кнопки — «Зламався» та «Переглянути чек». Перша з них запускає механізм виявлення дати поломки: користувачу відкривається вікно з календарем, де він обирає дату несправності. Якщо обрана дата ще потрапляє в гарантійний період,

					КНУ.РБ.123.25.01. ПС	Арк.
Арк.	№ документа	Підпис	Дата			

система автоматично оновлює статус пристрою на «Зламаний», здійснює пошук відповідного сервісного центру за брендом і надає детальну інформацію у вигляді сповіщення. Таким чином, користувач отримує не лише фіксацію факту поломки, а й одразу дізнається, куди можна звернутись для ремонту.

Кнопка «Переглянути чек» дає змогу швидко перевірити документ, прикріплений до пристрою під час його реєстрації. Програма відкриває нове вікно з відображенням зображення чека, зчитаного з вказаного шляху до файлу, що зберігається у полі `ReceiptPath`. Це зручно для підтвердження покупки, гарантійних умов або даних про модель.

Додатковий аналіз логіки `FormAnalytics.cs` – вікна статистики

Хоч вікно аналітики не є інтерактивним у сенсі введення даних, але воно активно використовує наявну інформацію з інших модулів, тому важливо розкрити, як саме працює логіка побудови графіків.

1. Джерело даних: `List<Device> devices`

Основним джерелом для побудови діаграм є список об'єктів `devices`, який передається у конструктор форми `FormAnalytics` з головного вікна `FormDevicesList`. Цей список є повною копією актуального стану бази пристроїв, який уже був завантажений при ініціалізації таблиці у `LoadData()`.

Таким чином, усі зміни, які були зроблені в інших частинах програми, включаючи оновлення статусу, дати поломки, або додавання нових пристроїв — **відразу відображаються в графіках без додаткових дій користувача.**

2. Побудова діаграми зламаних пристроїв

У першій вкладці відбувається групування пристроїв за брендами, де статус пристрою — "Зламаний":

```
var brokenData = devices
    .Where(d => d.Status == "Зламаний")
    .GroupBy(d => d.Brand)
    .ToDictionary(g => g.Key, g => g.Count());
```

Цей код використовує LINQ-фільтрацію та групування, які аналогічно використовуються у `FormDevicesList` під час фільтрації пристроїв у `ComboBox`.

Тобто, аналітична форма повторно застосовує логіку фільтрації статусів, але вже з метою побудови статистики.

3. Побудова графіку середнього гарантійного терміну

У другій вкладці відбувається обчислення середнього значення гарантії по кожному бренду:

```
var avgData = devices
    .GroupBy(d => d.Brand)
    .ToDictionary(
        g => g.Key,
```

```

g => Math.Round(g.Average(d =>
(d.WarrantyEndDate - d.PurchaseDate).TotalDays / 30.0),
1)
);

```

Ця частина є прямим результатом логіки, закладеної в Form1.cs, де розраховується WarrantyEndDate шляхом додавання місяців до дати покупки. Таким чином, використовується раніше розраховане поле, а не повторна формула.

4. Динамічні підписи та оформлення

Кожна точка на діаграмі отримує підпис, колір і шрифт, щоб зробити інтерфейс читабельним і зручним для користувача:

```

p.Label = p.YValues[0].ToString("0.0");
p.Font = new Font("Arial", 10, FontStyle.Bold);
p.LabelForeColor = Color.Black;

```

Це узгоджується з підходом у всій системі — інформація завжди надається у зручному вигляді, без потреби додаткового кліку чи наведення.

Таким чином, форма аналітики — це не просто додаткова візуалізація, а повністю інтегрована частина логіки системи, яка:

- автоматично працює з оновленими даними;
- повторно використовує логіку обчислення гарантії та статусів;
- базується на структурі Device, спільній для всіх форм;
- доповнює загальну концепцію автоматизації обліку й аналізу технічних пристроїв.

Хоч ця форма не змінює дані напряду, вона надає зворотній зв'язок, що дуже важливо для прийняття рішень і візуального контролю.

Висновки

У другому розділі детально розкрито процес побудови інформаційної системи «Календар-довідник гарантійного та технічного обслуговування електронних та побутових пристроїв». Основна увага приділялася формуванню архітектури, яка охоплює візуальні інтерфейси, логіку взаємодії та підключення до бази даних. Усі елементи системи реалізовано у вигляді взаємозалежних модулів, що забезпечує гнучкість та масштабованість.

Розроблено набір функціональних можливостей, які охоплюють повний цикл взаємодії з користувачем: додавання пристроїв, розрахунок гарантійного терміну, прикріплення чеків, фільтрація та візуалізація стану техніки. Система автоматично перевіряє актуальність гарантії, сповіщає про її завершення, а також надає рекомендації щодо сервісного обслуговування. Особливу увагу

приділено простоті взаємодії: всі дії супроводжуються інтуїтивно зрозумілими елементами та повідомленнями.

Аналітичний модуль реалізовано у вигляді окремого вікна з двома вкладками, що відображають кількість поломок по брендах та середній гарантійний термін. Такий підхід дозволяє користувачу швидко оцінювати надійність техніки, формувати звіти та приймати зважені рішення щодо майбутніх покупок.

Використання технології Windows Forms обґрунтовано потребою в швидкому створенні локального графічного інтерфейсу, що не залежить від веб-сервісів. Такий вибір дозволив реалізувати зручний, автономний і водночас функціонально багатий застосунок, орієнтований на звичайного користувача.

Усі етапи проєктування супроводжувалися врахуванням стабільності, безпеки та збереження персональних даних. Передбачено локальне зберігання всієї інформації, з можливістю експорту даних у форматі Excel для архівації або аналітики. Дані зберігаються у централізованій базі PostgreSQL, структура якої побудована з урахуванням надійності, гнучкості та майбутнього розширення.

Таким чином, сформована система забезпечує не лише базову реєстрацію техніки, а й комплексну підтримку користувача протягом усього життєвого циклу пристрою — від моменту покупки до завершення гарантії. Продумана структура, автоматизовані алгоритми та наочна подача інформації свідчать про високий рівень реалізації програмного продукту, який готовий до подальшого використання, масштабування та інтеграції в більш складні системи.

3. ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА МОЖЛИВОСТІ РОЗВИТКУ СИСТЕМИ

3.1 Результати тестування

Додати пристрій

Назва пристрою:

Бренд:

Дата покупки:

Гарантія (місяців):

Додати чек

Розрахувати гарантію

Гарантія діє до: 09.06.2025

Зберегти пристрій

Переглянути пристрої

Рисунок 3.1 – Скріншот вводу інформації бренду ASUS та зазначення розрахунку гарантії

Додати пристрій

Назва пристрою:

Бренд:

Дата покупки:

Гарантія (місяців):

Чек додано ✓

Розрахувати гарантію

Гарантія діє до: 09.06.2025

Зберегти пристрій

Переглянути пристрої

					КНУ.РБ.123.25.01.ТВМРС			
Змн.	Арк.	№ документа	Підпис	Дата	ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА МОЖЛИВОСТІ РОЗВИТКУ СИСТЕМИ	Літера	Аркуш	Аркушів
Розробив	Басов							
Перевірив	Купін							
Н.контроль	Кузнецов							
Затвердив	Купін					KI-21		

Рисунок 3.2 – Скріншот вигляд кнопки у функції вже доданого чеку від користувача

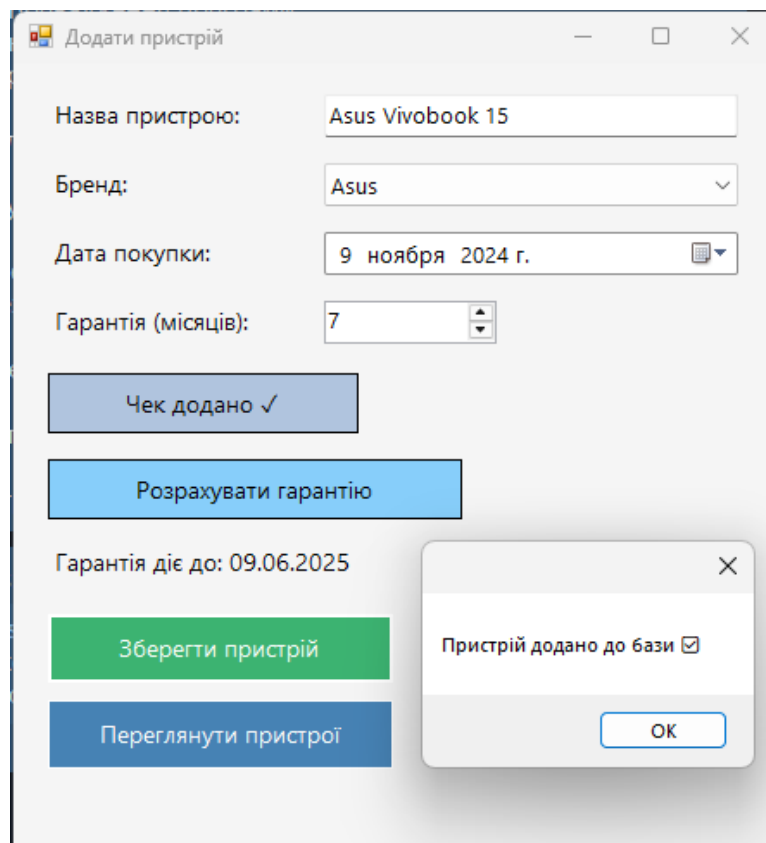


Рисунок 3.3 – Скріншот повідомлення яке програма видає після натискання кнопки «Зберегти пристрій»

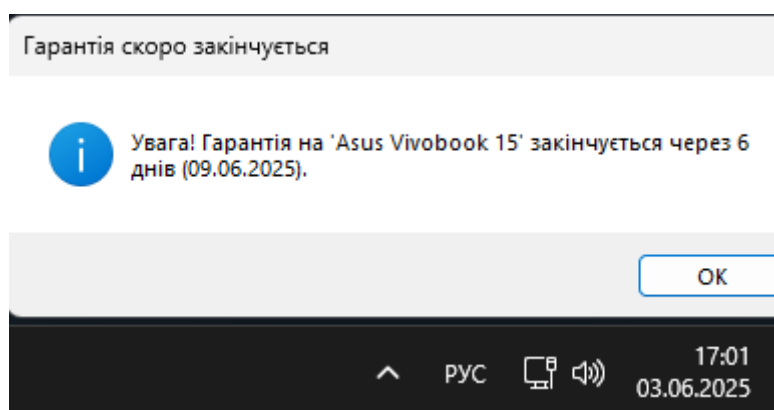


Рисунок 3.4 – Скріншот повідомлення після додавання до бази даних

Як можна побачити на скріні зазначається, що з попередженням про завершення гарантії, це по'язано з логікою програми, як було зазначено раніше вона попереджає таким чином коли до кінця гарантія менше-дорівнює 10 днів.

Список пристроїв

Гарантія закінчується скоро

Аналітика

Експортувати Excel

Назва	Бренд	Дата покупки	Гарантія до	Статус	Чек	Поломка
Asus Vivobook 15	Asus	09.11.2024	09.06.2025	Активний	Переглянути	Зламався

Рисунок 3.5 – Скріншот доданого пристрою вище який закономірно попав у таблицю з поміткою «гарантія закінчується скоро»

Додати пристрій

Назва пристрою: Huawei MateBook D

Бренд: Asus

Дата покупки:

Гарантія (місяців):

Додати чек

Розрахувати гарантію

Гарантія діє до: -

Зберегти пристрій

Переглянути пристрій

Рисунок 3.6 – Скріншот процесу додавання нового пристрою користувачем, наглядний приклад обрання бренду з ComboBox

Додати пристрій

Назва пристрою: Huawei MateBook D

Бренд: Huawei

Дата покупки: 27 ноября 2024 г.

Гарантія (місяців): 6

Чек додано ✓

Розрахувати гарантію

Гарантія діє до: 27.05.2025

Зберегти пристрій

Переглянути пристрій

Прострочена гарантія

Увага! Зазначена гарантія вже завершилась (27.05.2025). Додати пристрій до бази все одно?

Да Нет

Рисунок 3.7 – Скріншот повідомлення після додавання минуло пристрою, по даті тестування та гарантії видно, що пристрій прострочений що і зазначає програма

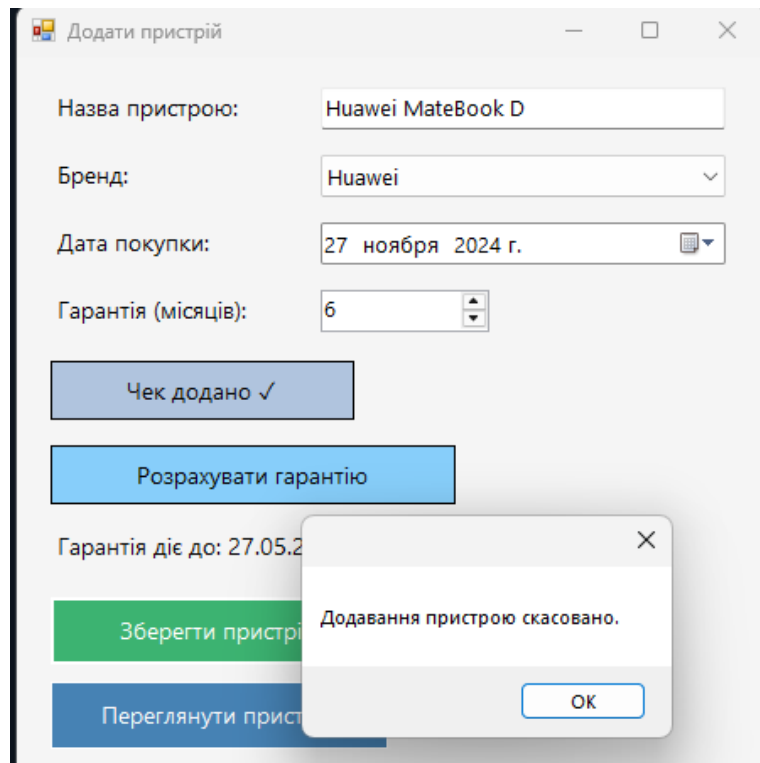


Рисунок 3.8 – Скріншот ситуації якщо користувач натиснув би кнопку «Нет» у минулому кроці

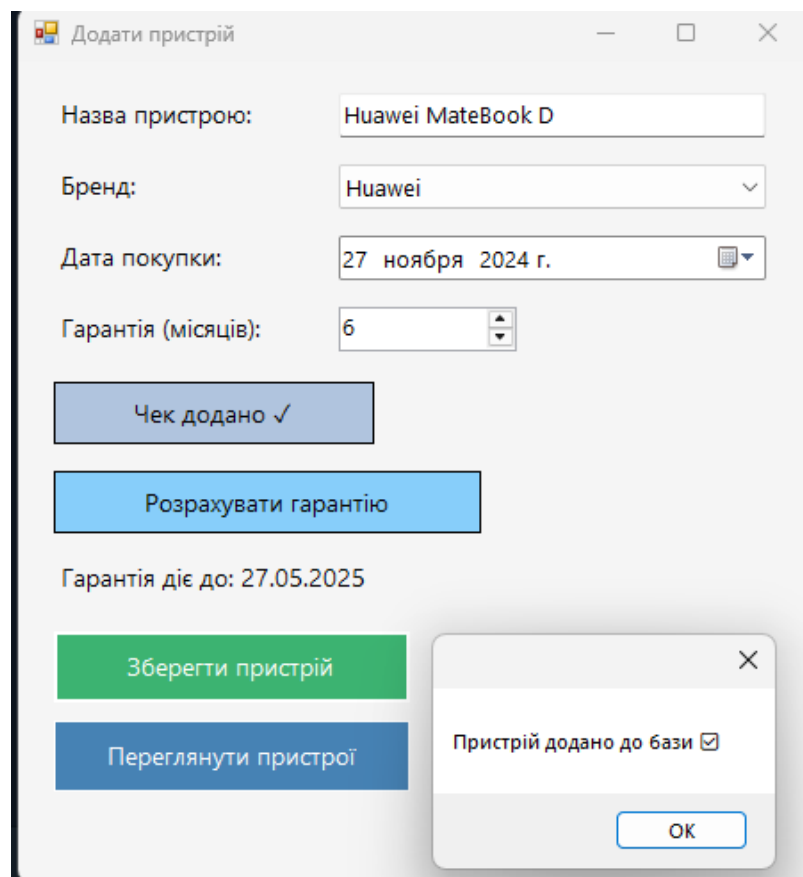


Рисунок 3.9 – Скріншот ситуації якщо користувач натиснув би кнопку «Да» у минулому кроці

Список пристроїв

Прострочені

Аналітика

Експортувати Excel

Назва	Бренд	Дата покупки	Гарантія до	Статус	Чек	Поломка
Ipad 10	Apple	29.04.2025	29.05.2025	Прострочений	Переглянути	Зламався
Huawei MateBook D	Huawei	27.11.2024	27.05.2025	Прострочений	Переглянути	Зламався
Телефон 8	Huawei	26.04.2025	26.05.2025	Прострочений	Переглянути	Зламався
Монітор Predator	MSI	11.03.2025	11.05.2025	Прострочений	Переглянути	Зламався
Note alone	Samsung	08.11.2024	08.05.2025	Прострочений	Переглянути	Зламався

Рисунок 3.10 – Скріншот де видно відсортований пристрій доданий раніше у колонці прострочений

Список пристроїв

Прострочені

Аналітика

Експортувати Excel

Назва	Бренд	Дата покупки	Гарантія до	Статус	Чек	Поломка
Ipad 10	Apple	29.04.2025	29.05.2025	Прострочений	Переглянути	Зламався
Huawei MateBook D	Huawei	27.11.2024	27.05.2025	Прострочений	Переглянути	Зламався
Телефон 8	Huawei	26.04.2025	26.05.2025	Прострочений	Переглянути	Зламався
Монітор Predator	MSI	11.03.2025	11.05.2025	Прострочений	Переглянути	Зламався
Note alone	Samsung	08.11.2024	08.05.2025	Прострочений	Переглянути	Зламався

Чек

ООО «КОМПАНИЯ»
ДОБРО ПОЖАЛОВАТЬ!

КАССОВЫЙ ЧЕК/ПРИХОД

НОМЕР СМЕНИ 000008 ЧЕК 000002

СНО ОСН

ПОЛНЫЙ РАСЧЕТ (4)

№0003 Краска масляная арт. 0345

1 20.000x 300.00 =6000.00B

ИТОГ =6000.00

В: СУММА НДС 18% =915.25

НАЛИЧНЫМИ =3000.00

ЭЛЕКТРОННЫМИ =0.00

09/06/2017 11:42 Кассир 1

ФИО

ЗН ККТ 0535860004001003

РН ККТ 0000000015018840

ИНН 7721306333

ФН 9999078900002273

ФД 0000000043 ФП 0159733624

СПАСИБО!

Рисунок 3.11 – Скріншот вигляду доданого чеку біля будь якого пристрою (тестове фото)

Рисунок 3.12 – Скріншот вводу нового пристрою користувачем, де видно зміну повідомлення на більш м'який варіант

Тут можна побачити як змінився тон повідомлення на відміну від тих пристроїв де гарантія вже була близька до завершення приклад яких був наведений раніше, так само пишеться кількість днів +- до завершення гарантію, враховуючи поточну дату.

Назва	Бренд	Дата покупки	Гарантія до	Статус	Чек	Поломка
Телевізор A1	Asus	16.03.2025	16.12.2025	Зламаний	Переглянути	Зламався
Dell Vostro	Dell	18.09.2024	18.08.2025	Активний	Переглянути	Зламався
Монітор	Dell	01.05.2025	01.06.2025	Зламаний	Переглянути	Зламався
H SDHA	Dell	01.10.2024	01.01.2026	Активний	Переглянути	Зламався
Note10-	HP	02.04.2025	02.06.2025	Зламаний	Переглянути	Зламався
Телефон 8	Huawei	26.04.2025	26.05.2025	Прострочений	Переглянути	Зламався
Huawei MateBo...	Huawei	27.11.2024	27.05.2025	Прострочений	Переглянути	Зламався
Honor 10	Huawei	16.01.2025	16.01.2026	Зламаний	Переглянути	Зламався
honor 10	Huawei	06.06.2025	06.07.2025	Активний	Переглянути	Зламався
Lenovo ThinkBo...	Lenovo	04.04.2025	04.12.2025	Активний	Переглянути	Зламався
		2025	16.06.2025	Активний	Переглянути	Зламався
		2025	03.06.2025	Зламаний	Переглянути	Зламався
		2025	06.06.2025	Зламаний	Переглянути	Зламався
		2025	11.05.2025	Прострочений	Переглянути	Зламався
		2024	11.12.2025	Зламаний	Переглянути	Зламався
		2025	06.06.2025	Зламаний	Переглянути	Зламався
Note alone	Samsung	08.11.2024	08.05.2025	Прострочений	Переглянути	Зламався
FLIGHT LIGHT	Xiaomi	05.03.2025	05.01.2026	Зламаний	Переглянути	Зламався

Рисунок 3.13 – Скріншот приклад вікна при натисканні кнопки «Зламався» у таблиці бази даних

ThinkPad X1 Ca...	Lenovo	16.01.2025	16.06.2025	Активний	Переглянути	Зламався
net92	Lenovo		06.06.2025	Зламаний	Переглянути	Зламався
9k3d	Lenovo		06.06.2025	Зламаний	Переглянути	Зламався
Монітор Predat...	MSI		05.05.2025	Прострочений	Переглянути	Зламався
Immortal2	MSI		12.12.2025	Зламаний	Переглянути	Зламався
galaxyphone80	Samsung		06.06.2025	Зламаний	Переглянути	Зламався
Note alone	Samsung		05.05.2025	Прострочений	Переглянути	Зламався
FLIGHT LIGHT	Xiaomi		01.01.2026	Зламаний	Переглянути	Зламався

Рисунок 3.14 – Скріншот демонстрації вікна якщо була введена дата поломки яка входить до гарантії (така як на минулому скрині)

Список пристроїв

Зламани

Аналітика

Експортувати Excel

Назва	Бренд	Дата покупки	Гарантія до	Статус	Чек	Поломка
galaxyphone80	Samsung	06.03.2025	06.06.2025	Зламаний	Переглянути	Зламався
Note10-	HP	02.04.2025	02.06.2025	Зламаний	Переглянути	Зламався
no one	Acer	06.05.2025	06.06.2025	Зламаний	Переглянути	Зламався
Ноутбук	Asus	19.03.2025	19.08.2025	Зламаний	Переглянути	Зламався
Телевізор A1	Asus	16.03.2025	16.12.2025	Зламаний	Переглянути	Зламався
Honor 10	Huawei	16.01.2025	16.01.2026	Зламаний	Переглянути	Зламався
Монітор	Dell	01.05.2025	01.06.2025	Зламаний	Переглянути	Зламався
net92	Lenovo	03.05.2025	03.06.2025	Зламаний	Переглянути	Зламався
FLIGHT LIGHT	Xiaomi	05.03.2025	05.01.2026	Зламаний	Переглянути	Зламався
O92	Acer	12.03.2025	12.07.2025	Зламаний	Переглянути	Зламався
9k3d	Lenovo	06.05.2025	06.06.2025	Зламаний	Переглянути	Зламався
Immortal2	MSI	11.12.2024	11.12.2025	Зламаний	Переглянути	Зламався
ThinkPad X1 Car...	Lenovo	16.01.2025	16.06.2025	Зламаний	Переглянути	Зламався

Рисунок 3.15 – Скріншот демонструючий перенесання та зміну статусу минулого пристрою з «Активний» до «Зламаний»

Оберіть дату поломки...

16.01.2025

OK

		04.04.2025	04.12.2025	Активний	Переглянути	Зламався
		06.05.2025	06.06.2025	Зламаний	Переглянути	Зламався
		16.01.2025	16.06.2025	Зламаний	Переглянути	Зламався
		03.05.2025	03.06.2025	Зламаний	Переглянути	Зламався
		11.03.2025	11.05.2025	Прострочений	Переглянути	Зламався
		11.12.2024	11.12.2025	Зламаний	Переглянути	Зламався
		12.11.2024	12.09.2025	Активний	Переглянути	Зламався
Katana 17 B13VFK	MSI					
galaxyphone80	Samsung	06.03.2025	06.06.2025	Зламаний	Переглянути	Зламався
Note alone	Samsung	08.11.2024	08.05.2025	Прострочений	Переглянути	Зламався
FLIGHT LIGHT	Xiaomi	05.03.2025	05.01.2026	Зламаний	Переглянути	Зламався

Рисунок 3.16 – Скріншот вибору користувачем дату поза гарантії приладу та натискання кнопки «Зламаний»

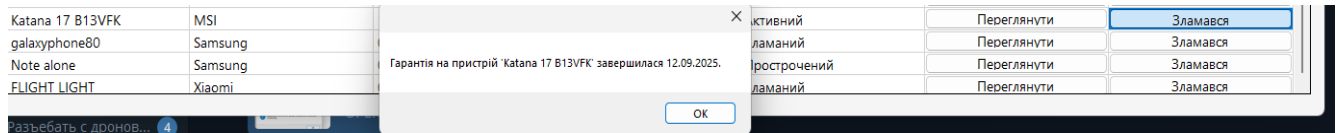


Рисунок 3.17 – Скріншот демонстрації вікна якщо була введена дата поломки яка НЕ входить до гарантії. (така як на минулому скрині)

Список пристроїв

Усі Аналітика Експортувати Excel

Назва	Бренд	Дата покупки	Гарантія до	Статус	Чек	Поломка
O92	Acer	12.03.2025	12.07.2025	Зламаний	Переглянути	Зламався
no one	Acer	06.05.2025	06.06.2025	Зламаний	Переглянути	Зламався
Ноутбук для навчання	Acer	19.12.2024	19.10.2025	Активний	Переглянути	Зламався
Ipad 10	Apple	29.04.2025	29.05.2025	Прострочений	Переглянути	Зламався
Iphone 16	Apple	05.02.2025	05.08.2025	Активний	Переглянути	Зламався
Asus Vivobook 15	Asus	09.11.2024	09.06.2025	Активний	Переглянути	Зламався
Телевізор A1	Asus	16.03.2025	16.12.2025	Зламаний	Переглянути	Зламався
Ноутбук	Asus	19.03.2025	19.08.2025	Зламаний	Переглянути	Зламався
H SDHA	Dell	01.10.2024	01.01.2026	Активний	Переглянути	Зламався
Dell Vostro	Dell	18.09.2024	18.08.2025	Активний	Переглянути	Зламався
Монітор	Dell	01.05.2025	01.06.2025	Зламаний	Переглянути	Зламався
Note10-	HP	02.04.2025	02.06.2025	Зламаний	Переглянути	Зламався
Honor 10	Huawei	16.01.2025	16.01.2026	Зламаний	Переглянути	Зламався
honor 10	Huawei	06.06.2025	06.07.2025	Активний	Переглянути	Зламався
Huawei MateBook D	Huawei	27.11.2024	27.05.2025	Прострочений	Переглянути	Зламався
Телефон 8	Huawei	26.04.2025	26.05.2025	Прострочений	Переглянути	Зламався
net92	Lenovo	03.05.2025	03.06.2025	Зламаний	Переглянути	Зламався
ThinkPad X1 Carbon	Lenovo	16.01.2025	16.06.2025	Зламаний	Переглянути	Зламався
Lenovo ThinkBook	Lenovo	04.04.2025	04.12.2025	Активний	Переглянути	Зламався
9k3d	Lenovo	06.05.2025	06.06.2025	Зламаний	Переглянути	Зламався
Immortal2	MSI	11.12.2024	11.12.2025	Зламаний	Переглянути	Зламався
Монітор Predator	MSI	11.03.2025	11.05.2025	Прострочений	Переглянути	Зламався
Katana 17 B13VFK	MSI	12.11.2024	12.09.2025	Зламаний	Переглянути	Зламався
galaxyphone80	Samsung	06.03.2025	06.06.2025	Зламаний	Переглянути	Зламався
Note alone	Samsung	08.11.2024	08.05.2025	Прострочений	Переглянути	Зламався
FLIGHT LIGHT	Xiaomi	05.03.2025	05.01.2026	Зламаний	Переглянути	Зламався

Рисунок 3.18 – Скріншот можливості сортування по будь якій категорії, тут приклад сортуються по брендам, ця інформація буде застосована у всій аналітиці далі.

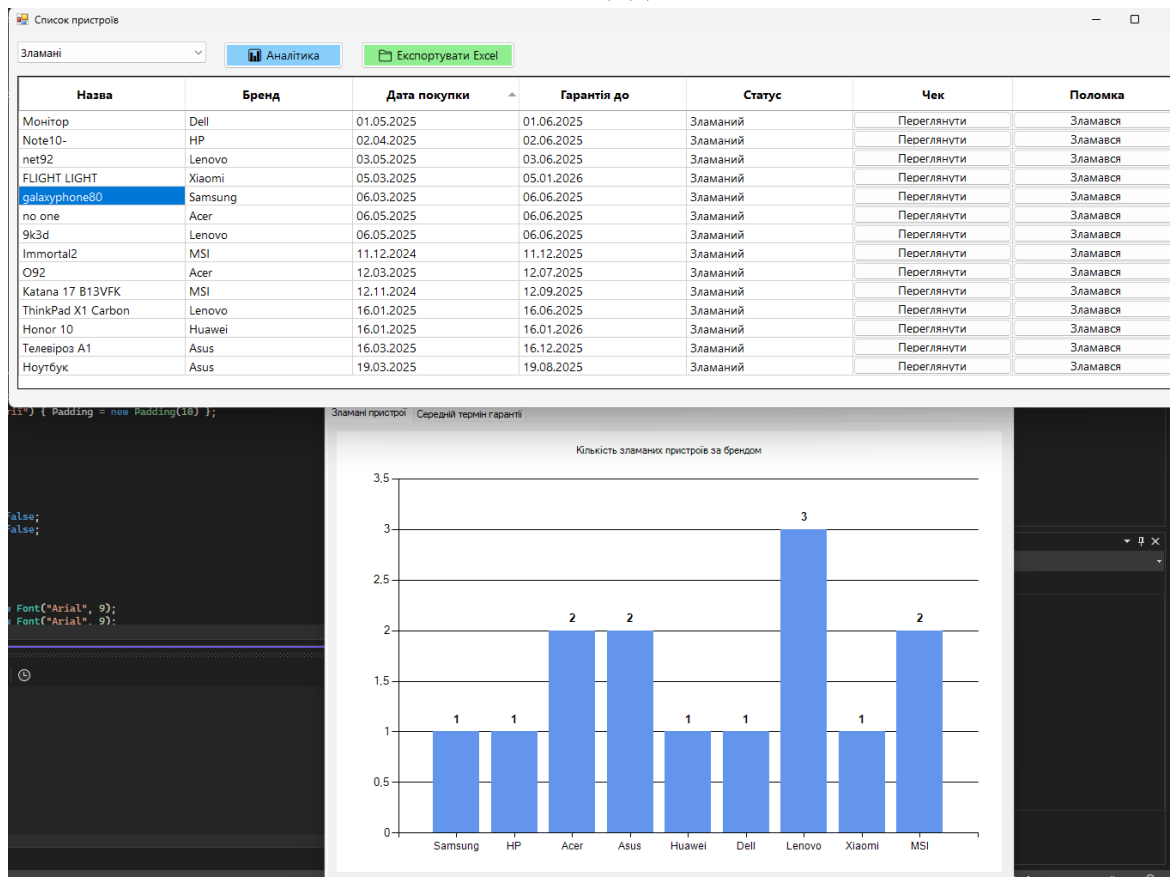


Рисунок 3.19 – Скріншот реалізація та демонстрації статистики за критеріями по кількості зламаних пристроїв за брендами

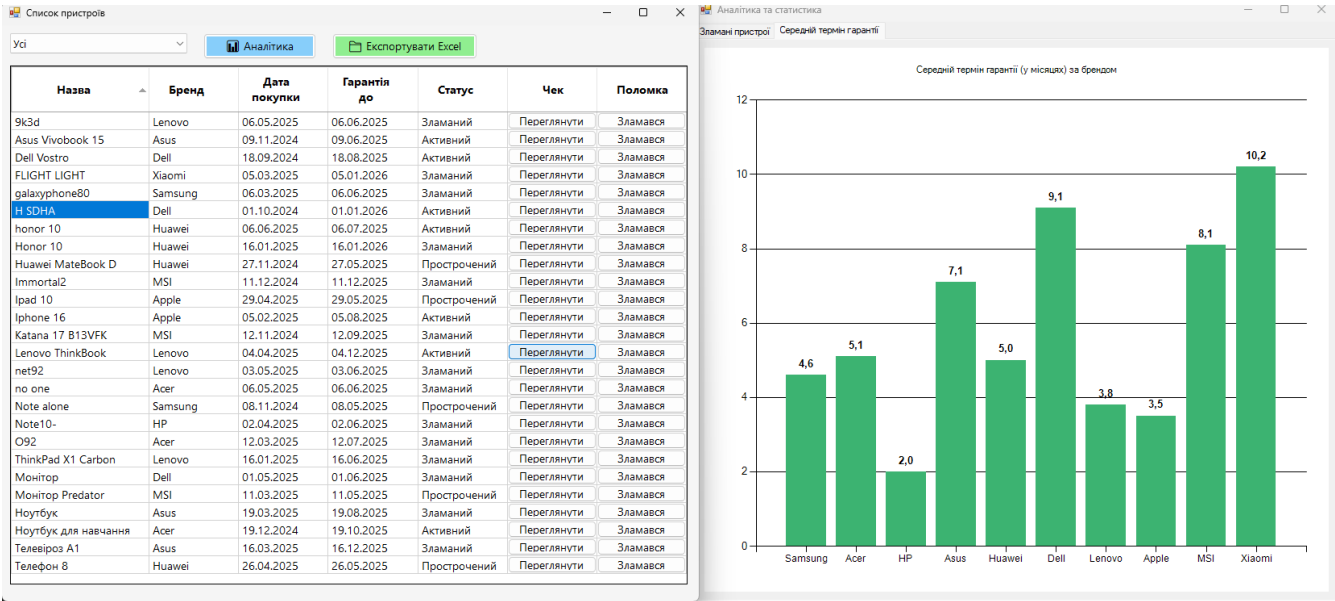


Рисунок 3.20 – Скріншот реалізація та демонстрації статистики за критеріями по кількості середніх термінів гарантії у пристроїв

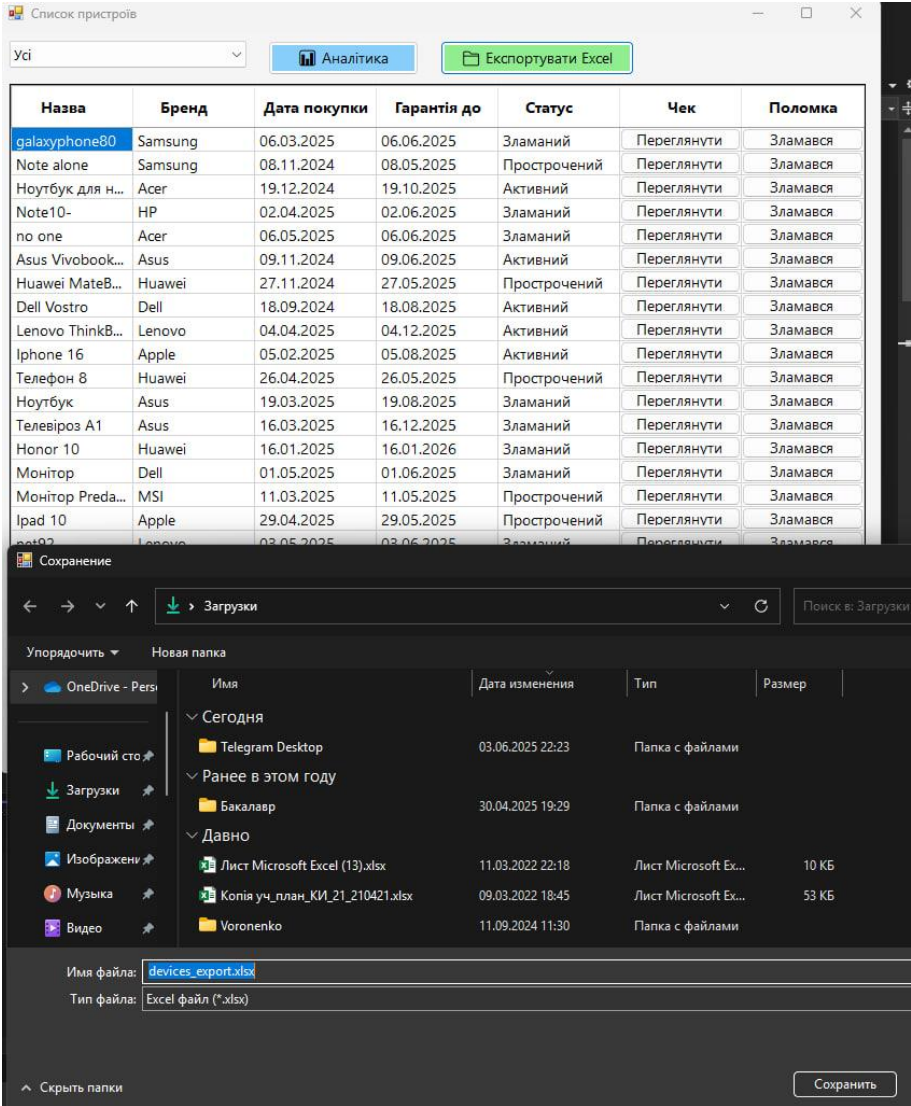


Рисунок 3.21 – Скріншот демонстрація вікна збереження яке з'являється при натисканні кнопки «Експортувати Excel»

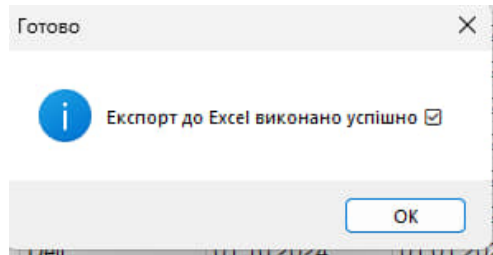


Рисунок 3.22 – Скріншот повідомлення яке спливає після успішного збереження

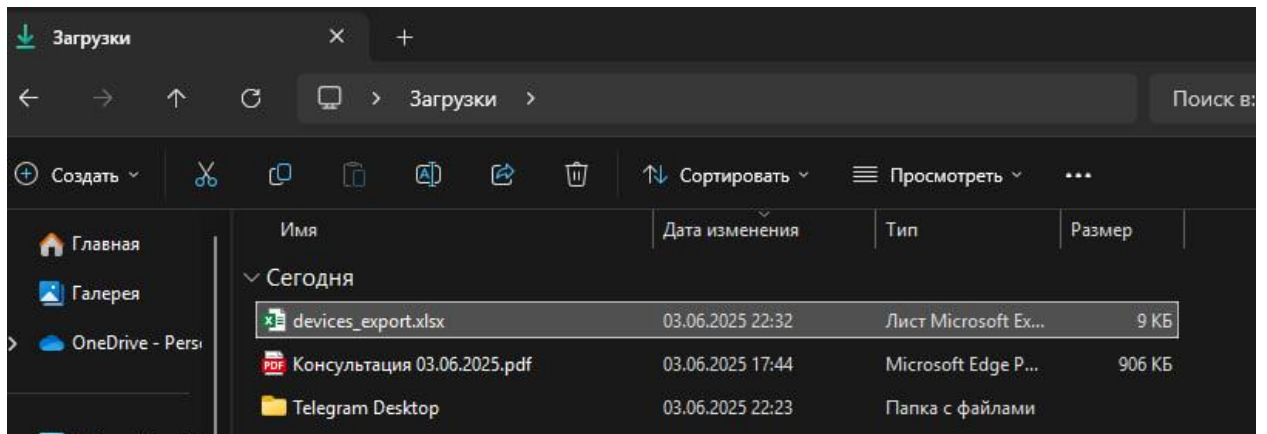


Рисунок 3.23 – Скріншот перевірки наявності збереження таблиці

devices_export.xlsx - Excel										
basovb6@gmail.com										
Файл Главная Вставка Разметка стл Формулы Данные Рецензиров Вид Справка Acrobat Помощь										
Вставить Шрифт Выравнивание Число Условное форматирование Форматировать как таблицу Стили Ячейки Редактирование Создать PDF Adobe Acrobat										
Буфер обмена fx Назва										
	A	B	C	D	E	F	G	H		
1	Назва	Бренд	Дата покупки	Гарантия до	Статус					
2	galaxyphone80	Samsung	06.03.2025	06.06.2025	Зламаний					
3	Note alone	Samsung	08.11.2024	08.05.2025	Прострочений					
4	Ноутбук для навчання	Acer	19.12.2024	19.10.2025	Активний					
5	Note10-	HP	02.04.2025	02.06.2025	Зламаний					
6	no one	Acer	06.05.2025	06.06.2025	Зламаний					
7	Asus Vivobook 15	Asus	09.11.2024	09.06.2025	Активний					
8	Huawei MateBook D	Huawei	27.11.2024	27.05.2025	Прострочений					
9	Dell Vostro	Dell	18.09.2024	18.08.2025	Активний					
10	Lenovo ThinkBook	Lenovo	04.04.2025	04.12.2025	Активний					
11	Iphone 16	Apple	05.02.2025	05.08.2025	Активний					
12	Телефон 8	Huawei	26.04.2025	26.05.2025	Прострочений					
13	Ноутбук	Asus	19.03.2025	19.08.2025	Зламаний					
14	Телевізор A1	Asus	16.03.2025	16.12.2025	Зламаний					
15	Honor 10	Huawei	16.01.2025	16.01.2026	Зламаний					
16	Монітор	Dell	01.05.2025	01.06.2025	Зламаний					
17	Монітор Predator	MSI	11.03.2025	11.05.2025	Прострочений					
18	Ipad 10	Apple	29.04.2025	29.05.2025	Прострочений					
19	net92	Lenovo	03.05.2025	03.06.2025	Зламаний					
20	FLIGHT LIGHT	Xiaomi	05.03.2025	05.01.2026	Зламаний					
21	O92	Acer	12.03.2025	12.07.2025	Зламаний					
22	honor 10	Huawei	06.06.2025	06.07.2025	Активний					
23	9k3d	Lenovo	06.05.2025	06.06.2025	Зламаний					
24	H SDHA	Dell	01.10.2024	01.01.2026	Активний					
25	Immortal2	MSI	11.12.2024	11.12.2025	Зламаний					
26	ThinkPad X1 Carbon	Lenovo	16.01.2025	16.06.2025	Зламаний					
27	Katana 17 B13VFK	MSI	12.11.2024	12.09.2025	Зламаний					

Рисунок 3.24 – Скріншот точного збереження інформації з бази даних до excel файлу

Висновки

У ході третього розділу було здійснено повноцінне тестування розробленого застосунку, що дало змогу впевнено оцінити його стабільність, функціональність та відповідність поставленим цілям. Завдяки покроковій демонстрації всіх можливостей програми з використанням реальних прикладів вдалося перевірити коректність роботи кожного елемента інтерфейсу, внутрішньої логіки та обробки даних. Такий підхід дозволив охопити увесь спектр основних сценаріїв взаємодії користувача із системою.

Програма показала здатність ефективно працювати з базою даних, правильно реагуючи на введені значення, обробляючи дати та змінюючи статуси пристроїв відповідно до реального стану гарантії. Всі ключові процеси, включаючи розрахунок дати завершення гарантії, фільтрацію пристроїв за критеріями, відображення чеку, фіксацію поломки та побудову аналітичних графіків — були протестовані вручну в умовах, наближених до реального використання, та продемонстрували стабільну роботу без збоїв і логічних помилок.

Особливу увагу було приділено перевірці взаємозв'язку між графічним інтерфейсом і базою даних PostgreSQL. Як показали тести, програма коректно зчитує, зберігає та оновлює записи в таблицях. Окремо було протестовано експорт до Excel, який дав змогу переконатись у зручності подальшого використання даних поза межами застосунку. Усі збережені таблиці містили необхідну структуру, формулювання полів та значення, що відповідають візуальному відображенню на формі.

Користувацький інтерфейс також пройшов перевірку з точки зору ергономіки та інтуїтивності. Розташування елементів, кольорова гама, повідомлення та підказки виявилися достатньо інформативними, що забезпечує комфортне користування навіть для осіб без технічного досвіду. Система не дозволяє виконати критичні дії без попереднього підтвердження, що гарантує захист від випадкових помилок.

У результаті тестування встановлено, що застосунок повністю відповідає функціональним вимогам, визначеним на етапі проектування. Програма не лише реалізує усі заплановані функції, а й має потенціал до подальшого розвитку. Завдяки модульності коду та розділенню логіки між формами, можливе легке розширення системи або її адаптація до інших сценаріїв використання.

Таким чином, завершене тестування підтвердило життєздатність програмного продукту як локального засобу обліку електронних та побутових пристроїв з контролем термінів гарантії та обслуговування. Отримані результати свідчать про готовність програми до впровадження у реальних умовах, забезпечуючи користувачам надійний інструмент для контролю гарантійних зобов'язань.

ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було повністю реалізовано інформаційну систему «Календар-довідник гарантійного та технічного обслуговування електронних та побутових пристроїв». Вона поєднує інтуїтивно зрозумілий графічний інтерфейс, гнучку архітектуру, централізовану базу даних та автоматизовану логіку, що забезпечує високий рівень зручності та надійності для кінцевого користувача.

У першому розділі проведено глибокий аналіз предметної області та виявлено проблеми, пов'язані з обліком і контролем гарантійних строків у повсякденному використанні пристроїв. Показано недоліки типових рішень, які не дають змоги централізовано й ефективно керувати інформацією про гарантії, чеки, поломки й сервісні центри. Цей аналіз обґрунтував доцільність створення нової системи саме у вигляді окремого десктопного застосунку.

Другий розділ присвячений технічному проектуванню інформаційної системи. Детально описано реалізацію вікон введення, перегляду та аналітики, показано логіку взаємодії між формами, обробку подій, підключення до бази даних PostgreSQL та використання бібліотек для візуалізації та експорту. Вибір технології Windows Forms обґрунтовано простотою у реалізації, достатньою функціональністю та можливістю локального збереження даних без залежності від зовнішніх сервісів.

У третьому розділі здійснено ручне тестування всіх основних функцій системи. Було підтверджено, що система коректно працює з введенням пристроїв, обчислює термін гарантії, зберігає дані до бази, автоматично змінює статуси (наприклад, «Прострочений» або «Зламаний»), надає повідомлення, фільтрує пристрої, дозволяє переглядати чеки та експортувати дані. Особливу увагу приділено інтегрованій аналітиці, яка без додаткових дій з боку користувача формує графіки для прийняття обґрунтованих рішень.

Розроблена система є функціонально завершеним продуктом, який може бути впроваджений на будь-якому персональному комп'ютері під керуванням Windows. Завдяки використанню відкритих технологій та гнучкої структури, її легко адаптувати або розширювати в майбутньому — наприклад, додати хмарне збереження, підтримку мобільного доступу чи інтеграцію з онлайн-базами сервісних центрів.

Отже, розроблена інформаційна система повністю відповідає меті дипломного проєкту — вона автоматизує облік гарантійного обслуговування, спрощує взаємодію з пристроями, допомагає уникнути втрати важливої інформації та створює надійну основу для цифрової організації технічного обліку

					КНУ.РБ.123.25.01.В			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробив	Басов				ВИСНОВКИ			
Перевірив	Купін							
Н.контроль	Кузнецов							
Затвердив	Купін							
						Літера	Аркуш	Аркушів
						KI-21		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Оліфер В. Г., Оліфер Н. А. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов. 5-е изд. СПб. : Питер, 2016. 992 с.
2. Семенов А. Б. Проектирование и расчет структурированных кабельных систем и их компонентов. М. : ДМК Пресс, 2016. 416 с.
3. Заміховська О. Л. Комп'ютерні мережі та телекомунікації : навч. посіб. Івано-Франківськ : ІФНТУНГ, 2013. 177 с.
4. Буров Є. В. Комп'ютерні мережі : підручник. Львів : Манголія, 2010. 262 с.
5. Мельник І. В., Лутовинський А. О. Проектування та дослідження комп'ютерних мереж. К. : Університет Україна, 2010. 362 с.
6. Воробієнко П. П., Нікітюк Л. А., Резнієнко П. І. Телекомунікаційні та інформаційні мережі : підручник. К. : САММІТ-Книга, 2010. 640 с.
7. Таненбаум Э., Узердол Л. Компьютерные сети. СПб. : Питер, 2016. 960 с.
8. Яворський В. Л. Бази даних. Теорія та практика використання : навч. посібник. К. : Ліра-К, 2020. 312 с.
9. Постгрес: Офіційна документація PostgreSQL. URL: <https://www.postgresql.org/docs/> (дата звернення: 02.06.2025).
10. Microsoft. Створення додатків Windows Forms у Visual Studio. URL: <https://learn.microsoft.com/> (дата звернення: 20.05.2025).
11. Глушаков І. Ю. Технології розробки програмного забезпечення : підручник. К. : Каравела, 2019. 384 с.
12. ISO/IEC 12207. Software life cycle processes – International Standard. Geneva : ISO, 2017.
13. Галкін М. О. Програмна інженерія : курс лекцій. Х. : ХНУРЕ, 2018. 156 с.
14. Романенко В. І. Стандартизація та сертифікація в ІТ. К. : Ліра-К, 2017. 248 с.
15. Онлайн-сервіс для роботи з гарантійними пристроями: аналіз існуючих рішень. URL: <https://www.warranty.co.uk/> (дата звернення: 01.06.2025).
16. API документація Npgsql – офіційний драйвер PostgreSQL для .NET. URL: <https://www.npgsql.org/doc/> (дата звернення: 04.06.2025).
17. ClosedXML. Простий API для створення Excel-файлів у C#. URL: <https://github.com/ClosedXML/ClosedXML> (дата звернення: 24.05.2025).
18. Електроніка для розробників. URL: <https://electronics.stackexchange.com/> (дата звернення: 03.06.2025).

					КНУ.РБ.123.25.01.СВЖ			
Змн.	Арк.	№ документа	Підпис	Дата	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ			
Розробив	Басов							
Перевірів	Купін							
Н.контроль	Кузнецов							
Затвердив	Купін							
						Літера	Аркуш	Аркушів
						KI-21		

19. Stack Overflow — технічна підтримка та спільнота. URL: <https://stackoverflow.com> (дата звернення: 28.05.2025).

20. Офіційний сайт компанії Cisco. URL: <https://www.cisco.com> (дата звернення: 30.05.2025).

					КНУ.РБ.123.25.01. СВЖ	Арк.
	Арк.	№ документа	Підпис	Дата		