

НАСТІЛЬНИЙ ДОДАТОК КОНТРОЛЮ ТА ОБЛІКУ ЗАПИТІВ НА РЕМОТ ОБЛАДНАННЯ

Коваленко Є. В.

Коледж інформаційних технологій та землевпорядкування НАУ, м. Київ

ВСТУП

Актуальність і постановка проблеми. В умовах цифровізації економіки ефективне управління технічним обслуговуванням обладнання набуває стратегічного значення для підприємств будь-якого масштабу. Ручні методи реєстрації ремонтних запитів призводять до втрати інформації, затримок виконання та відсутності аналітичної бази для прийняття управлінських рішень. Вирішення цієї проблеми можливе шляхом впровадження настільного програмного забезпечення, що забезпечує централізований облік запитів на ремонт обладнання.

Мета дослідження. Дослідити предметну область обліку ремонтних запитів, проаналізувати прикладну реалізацію настільного додатку на платформі Windows Forms і SQL Server та обґрунтувати напрями його модернізації до рівня промислово придатної системи.

Аналіз останніх досліджень і публікацій. Сучасні системи управління технічним обслуговуванням (CMMS) – такі як ALLDATA, Mitchell 1, Autodata та Dealer Management Systems – є потужними, але дорогими та складними для малих підприємств [1; 2]. Дослідження у сфері автоматизації технічного сервісу [3] підкреслюють потребу в компактних настільних рішеннях, що не вимагають складної серверної інфраструктури. Використання платформи Windows Forms (.NET) у поєднанні з ADO.NET є обґрунтованим вибором для такого класу систем [4].

Короткий опис дослідження, його методів і засобів.

Об'єктом дослідження є репозиторій InventoryManagement – настільний додаток на C# / Windows Forms із базою даних SQL Server LocalDB. Система охоплює 31 клас, розподілений між чотирма шарами: 14 класів інтерфейсного шару (Forms, 45%), 7 класів доступу до даних (DAL, 23%), 7 таблиць бази даних (23%) та 3 допоміжні утиліти (9%), як показано на рис. 1.

Методи дослідження: структурний аналіз вихідного коду, функціональне та негативне тестування, аналіз схеми бази даних, порівняльний аналіз із промисловими CMMS-системами. Для систематизації результатів застосовано табличне представлення функціональних характеристик та напрямів розвитку системи.

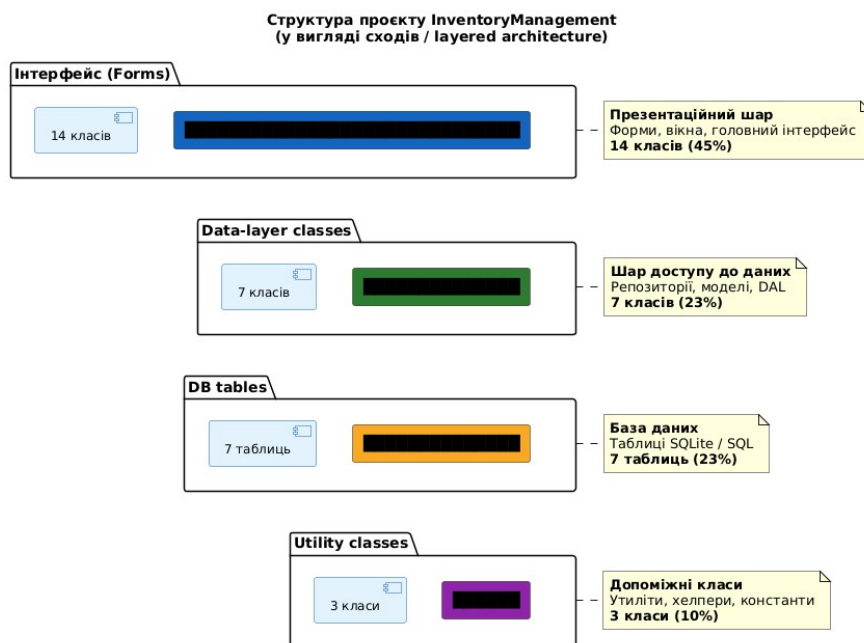


Рис. 1. Структура проєкту InventoryManagement (шарова архітектура)

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Аналіз архітектури показав, що система реалізує повний цикл облікових операцій через взаємопов'язані модулі, наведені у табл. 1.

Таблиця 1

Основні модулі розробленої системи та їх функції

№	Модуль	Функція	Технологія
1	UserAuth + AccountManager	Автентифікація та реєстрація користувачів у системі	WinForms, ADO.NET
2	Product + ProductManager	Повний CRUD обліку запитів, пошук за назвою та категорією	SQL Server LocalDB
3	Category + CategoryManager	Довідник категорій обладнання, класифікація заявок	WinForms, SQL
4	AddStock + History	Реєстрація дій, накопичення журналу змін	SqlCommand, DataGridView
5	Cart + Sale + Quantity	Управління поточними операціями, контроль кількості	WinForms, ADO.NET
6	Checkout + PointOfSale	Оформлення, розрахунок вартості, перевірка оплати	TransactionManager

7	Transaction Details +	Перегляд виконаних операцій та деталізація журналу	DataGridView, SQL
---	-----------------------	--	-------------------

Тестування системи за позитивними та негативними сценаріями підтвердило коректне виконання CRUD-операцій, контролю залишків, формування та збереження транзакцій, перегляду журналу змін. Водночас виявлено структурні обмеження: паролі зберігаються у відкритому вигляді (без хешування); фінансові поля в таблицях Transaction та Orders мають тип VARCHAR замість decimal, що унеможлиблює коректну агрегацію; фільтрація за ідентифікатором користувача (SessionUID) реалізована непослідовно — метод GetTotalPrice() рахує суму по всій таблиці Cart без прив'язки до поточного користувача [5]. Відсутній окремий сервісний шар та транзакційна логіка на рівні SQL Server, що знижує надійність при збоях.

На основі виявлених обмежень сформовано пріоритетні напрями модернізації (табл. 2).

Таблиця 2
Пріоритетні напрями розвитку системи

Напрямок модернізації	Суть змін	Очікуваний ефект
Нормалізація БД	Перехід від текстових полів і VARCHAR-сум до зовнішніх ключів та numeric-типів	Підвищення цілісності даних і спрощення аналітики
Статусна модель заявки	Додавання полів статусу, виконавця, дедлайну, пріоритету	Повноцінний контроль життєвого циклу запиту
Безпека облікових даних	Хешування паролів, рольова модель, аудит критичних дій	Зниження ризиків несанкціонованого доступу
Транзакційність	Обгортання критичних SQL-операцій у транзакції з відкатом	Гарантія узгодженості даних при збоях
Аналітична панель	Наповнення Dashboard реальними КРІ та показниками	Підтримка управлінських рішень на основі даних
Масштабування	Міграція з LocalDB до серверної СУБД, багатопарова архітектура	Готовність до багатокористувацького корпоративного використання

ВИСНОВКИ

У результаті дослідження проаналізовано предметну область обліку запитів на ремонт обладнання та функціональну реалізацію настільного додатку InventoryManagement. Система демонструє повний цикл облікових операцій і є обґрунтованою основою для подальшого розвитку. Визначено шість ключових напрямів модернізації: нормалізація бази даних, впровадження статусної моделі заявки, криптографічний захист облікових даних, реалізація транзакційної логіки, наповнення аналітичної панелі та масштабування до серверної СУБД. Реалізація запропонованих удосконалень дозволить перевести систему з рівня навчального прототипу на рівень повноцінного корпоративного CMMS-рішення.

ДЖЕРЕЛА

1. Mitchell1. ProDemand: Professional-grade repair information [Electronic resource]. URL: <https://mitchell1.com> (дата звернення: 04.05.2026).
2. Autodata Group. Autodata Online – technical information platform [Electronic resource]. URL: <https://autodata-group.com> (дата звернення: 04.05.2026).
3. Баранов О.А. Інформаційні системи управління технічним сервісом. Київ: НАУ, 2021. 210 с.
4. Microsoft. Windows Forms documentation [Electronic resource]. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/> (дата звернення: 04.05.2026).
5. Microsoft. ADO.NET Overview [Electronic resource]. URL: <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview> (дата звернення: 04.05.2026).