

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

ВІДДІЛЕННЯ
«ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА КОМП'ЮТЕРНІ НАУКИ»

ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ
«КОМП'ЮТЕРНІ НАУКИ»

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня фаховий молодший бакалавр
за спеціальністю 122 Комп'ютерні науки

на тему:

**«Інформаційна система обліку та технічного
обслуговування сільськогосподарської техніки»**

Виконав здобувач освіти групи П-43стн
Ярмоченко Руслан Іванович

Керівник роботи:
Студзінський Володимир Вікторович

Рецензент:

Зміст

Вступ	6
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	9
1.1 Основні задачі обліку техніки	9
1.2 Огляд існуючих рішень	10
1.3 Функціональні вимоги до програми	12
1.4 Вибір технологій розроблення	14
1.5 Постановка задачі	16
Висновки до розділу 1	17
Розділ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ	19
2.1 Модель бази даних	19
2.2 Список основних форм та їх призначення	32
2.3 Архітектура програмного засобу	40
Висновки до розділу 2	46
Розділ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ	47
3.1 Мінімальні вимоги до системи	47
3.2 Інструкція для роботи з програмою	48
Висновки до розділу 3	54
ВИСНОВКИ	55
Перелік літературних джерел	57
Додаток А. Код для роботи з базою даних	60

					КРФМБ.122.043стн.020.2026.ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Ярмоченко Р І			Інформаційна система обліку та технічного обслуговування сільськогосподарської техніки	Літ.	Арк.
Перевірів		Студзінський В В					3
Рецензент		.				ЖАТФК	
Н. Контр.							
Затвердив.		Гнатюк О.Ф.					
						Аркушів	61

РЕФЕРАТ

Обсяг роботи: 61 стор., 17 рис., 22 табл., 1 додаток, 24 джерел.

Об'єкт дослідження — процеси обліку та технічного обслуговування сільськогосподарської техніки на аграрних підприємствах.

Предмет дослідження — методи та засоби автоматизації зазначених процесів із застосуванням сучасних технологій розроблення настільного програмного забезпечення.

Мета роботи — проектування та реалізація інформаційної системи обліку та технічного обслуговування сільськогосподарської техніки як настільного застосунку для ОС Windows.

У роботі проведено аналіз предметної галузі, виявлено недоліки існуючих програмних рішень та сформульовано функціональні вимоги. Спроектовано реляційну модель бази даних SQLite у складі чотирьох таблиць. Розроблено архітектуру системи за шаблоном MVVM з чотирма логічними шарами та механізмом впровадження залежностей. Реалізовано CRUD-функціонал управління реєстром техніки та журналом ТО, автоматичний контроль строків обслуговування, підсистему формування звітів у форматах PDF та CSV, розмежування прав доступу і журналювання дій користувачів.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ, СІЛЬСЬКОГОСПОДАРСЬКА ТЕХНІКА, WPF, MVVM, C#, .NET 9, SQLITE, ENTITY FRAMEWORK CORE, PDF-ЗВІТИ.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	База даних
CMMS	Computerized Maintenance Management System — комп'ютеризована система управління технічним обслуговуванням
CRUD	Create, Read, Update, Delete — базові операції з даними (створення, читання, оновлення, видалення)
CSV	Comma-Separated Values — текстовий формат даних з роздільниками
DI	Dependency Injection — впровадження залежностей
EAM	Enterprise Asset Management — управління активами підприємства
EF Core	Entity Framework Core — об'єктно-реляційний маппер
FK	Foreign Key — зовнішній ключ
MTP	Машинно-тракторний парк
MVVM	Model-View-ViewModel — архітектурний шаблон
ORM	Object-Relational Mapping — об'єктно-реляційне відображення
PDF	Portable Document Format — формат переносних документів
PK	Primary Key — первинний ключ
СУБД	Система управління базами даних
ТО	Технічне обслуговування
UI	User Interface — інтерфейс користувача
ViewModel	Клас логіки представлення у шаблоні MVVM
WPF	Windows Presentation Foundation — технологія побудови інтерфейсу
XAML	Extensible Application Markup Language — мова розмітки інтерфейсу WPF
ЗНФ	Третя нормальна форма реляційної БД

ВСТУП

Сільське господарство є однією з ключових галузей національної економіки України, що забезпечує продовольчу безпеку країни та формує значну частку експортного потенціалу. Ефективне функціонування аграрних підприємств безпосередньо залежить від технічного стану машинно-тракторного парку, оскільки сільськогосподарська техніка є основним засобом виробництва у рослинництві та інших підгалузях агропромислового комплексу. Незадовільний технічний стан машин призводить до зриву агротехнічних строків, зростання витрат на непланові ремонти та зниження загальної ефективності виробничого процесу.

Одним із головних чинників підтримання техніки у належному стані є своєчасне проведення технічного обслуговування (ТО) відповідно до регламентів заводів-виробників. Проте на більшості вітчизняних аграрних підприємств облік техніки та планування ТО здійснюються із застосуванням паперової документації або засобів загального призначення — електронних таблиць, — що не забезпечує належного рівня автоматизації, оперативності та достовірності інформації. Відсутність спеціалізованого програмного забезпечення унеможливорює централізоване зберігання даних про техніку, формування нагадувань про наближення строків обслуговування та оперативне отримання аналітичної звітності.

Зазначені обставини зумовлюють актуальність розроблення інформаційної системи, яка автоматизує процеси обліку сільськогосподарської техніки та управління її технічним обслуговуванням.

Метою кваліфікаційної роботи є проєктування та реалізація інформаційної системи обліку та технічного обслуговування сільськогосподарської техніки, що забезпечує ведення реєстру машин, планування та реєстрацію проведених робіт з ТО, автоматичне формування нагадувань і звітів.

Для досягнення поставленої мети визначено такі **завдання**:

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

- провести аналіз предметної області та виявити функціональні вимоги до системи;
- спроектувати структуру бази даних та архітектуру програмного забезпечення;
- реалізувати модулі обліку техніки та журналу технічного обслуговування;
- розробити підсистему автоматичного контролю строків ТО з механізмом сповіщення;
- забезпечити формування звітів у форматах PDF та CSV;
- реалізувати розмежування прав доступу користувачів системи.

Об'єктом дослідження є процеси обліку та технічного обслуговування сільськогосподарської техніки на аграрних підприємствах.

Предметом дослідження є методи та засоби автоматизації зазначених процесів із застосуванням сучасних технологій розроблення настільного програмного забезпечення.

У роботі застосовано такі методи дослідження: аналіз і синтез при формуванні вимог до системи; об'єктно-орієнтоване проєктування при розробленні архітектури; метод прецедентів при моделюванні функціональності; емпіричне тестування при перевірці коректності роботи програмного продукту.

Практичне значення одержаних результатів полягає в тому, що розроблена інформаційна система може бути впроваджена на аграрних підприємствах будь-якого масштабу для підвищення ефективності управління машинно-тракторним парком. Автоматизація контролю строків технічного обслуговування дозволяє мінімізувати ризик виходу техніки з ладу в критичні агротехнічні періоди та скоротити витрати на позапланові ремонти.

Система реалізована з використанням платформи .NET 9 та мови програмування C#. Графічний інтерфейс користувача побудовано на основі технології Windows Presentation Foundation (WPF) із застосуванням шаблону проєктування MVVM (Model-View-ViewModel), що забезпечує чітке

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

розмежування рівнів представлення та бізнес-логіки. Для зберігання даних використовується вбудована реляційна СУБД SQLite у поєднанні з ORM-бібліотекою Entity Framework Core 9, що унеможливило необхідність встановлення окремого сервера баз даних і спростило розгортання системи в умовах підприємства.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі виконано аналіз предметної області та розглянуто існуючі програмні рішення. Другий розділ присвячено проєктуванню системи: визначенню архітектури, структури бази даних та інтерфейсу користувача. У третьому розділі описано реалізацію програмного продукту та результати його тестування.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні задачі обліку техніки

Сільськогосподарська техніка є найбільш капіталомістким виробничим ресурсом аграрного підприємства. Трактори, комбайни, сівалки, обприскувачі та інші машини функціонують в умовах інтенсивних навантажень у жорстко обмежені агротехнічні строки, що висуває підвищені вимоги до організації їх технічного обслуговування та своєчасного обліку технічного стану.

Управління машинно-тракторним парком підприємства охоплює два взаємопов'язані напрями: облік техніки та управління технічним обслуговуванням.

Облік техніки передбачає вирішення таких задач:

- ведення реєстру одиниць техніки з повним набором технічних та ідентифікаційних характеристик (найменування, модель, рік випуску, серійний номер);
- відстеження поточного статусу кожної машини (активна в експлуатації, перебуває на ремонті, списана);
- фіксування дат введення техніки в експлуатацію та проведення останнього технічного обслуговування;
- зберігання службових приміток щодо особливостей експлуатації окремих машин.

Управління технічним обслуговуванням включає:

- реєстрацію виконаних робіт з технічного обслуговування та ремонту із зазначенням дати, типу, змісту робіт, їх вартості та виконавця;
- планування наступних ТО на підставі встановленого для кожної машини міжсервісного інтервалу;
- автоматичний контроль наближення та прострочення строків обслуговування;
- формування нагадувань для відповідальних осіб щодо необхідності проведення ТО;

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

- накопичення статистики витрат на обслуговування в розрізі окремих машин і типів робіт.

Крім зазначеного, ефективна інформаційна система має забезпечувати розмежування прав доступу між категоріями користувачів — адміністраторами та операторами, — а також надавати можливість формування звітної документації для прийняття управлінських рішень.

Відсутність автоматизованого обліку на практиці призводить до таких наслідків: пропуск планових ТО через відсутність нагадувань; дублювання або втрата записів при паперовому документообігу; неможливість оперативного аналізу стану парку техніки; труднощі у формуванні звітності для керівництва та контрольних органів. Ці проблеми зумовлюють необхідність розроблення спеціалізованого програмного забезпечення.

1.2 Огляд існуючих рішень

На ринку програмного забезпечення для управління технічним обслуговуванням техніки представлено рішення різного рівня складності та орієнтації. Проведений аналіз охоплює як спеціалізовані аграрні системи, так і універсальні платформи управління активами (EAM/CMMS).

1.2.1 Комп'ютеризовані системи управління технічним обслуговуванням (CMMS)

eMaint CMMS — хмарна платформа американського розробника Fluke Reliability, орієнтована на управління технічним обслуговуванням промислового обладнання. Система надає широкий функціонал: планування ТО, облік запчастин, мобільний доступ, інтеграція з IoT-пристроями. Однак висока вартість ліцензування (від 69 USD на місяць за одного користувача), необхідність постійного підключення до мережі Інтернет та відсутність локалізації для українського ринку роблять цей продукт недоступним для більшості вітчизняних аграрних підприємств.

Fiix — хмарна CMMS канадського розробника, що пропонує зручний інтерфейс і мобільні застосунки. Аналогічно до eMaint, орієнтована на промисловість, не адаптована до специфіки сільськогосподарського

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

виробництва, потребує хмарного розгортання та передбачає значні витрати на ліцензування.

Proteus MMX — настільне рішення для управління технічним обслуговуванням з підтримкою локального розгортання. Система містить розширений модуль управління запасами, що ускладнює впровадження на малих підприємствах, де подібний функціонал є надлишковим.

1.2.2 Спеціалізовані аграрні програми

АгроПлан — вітчизняна веб-система для планування польових робіт та обліку ресурсів, орієнтована на рослинництво. Попри наявність модуля техніки, система не містить повноцінного журналу ТО та механізму автоматичних нагадувань. Хмарна архітектура не є прийнятною для підприємств з обмеженим доступом до мережі.

John Deere Operations Center та **CNH AFS Connect** — фірмові платформи провідних виробників сільськогосподарської техніки, інтегровані з телематичними системами машин. Суттєвим обмеженням є прив'язаність до техніки конкретного виробника: підприємства з різномірним парком машин не можуть повноцінно використовувати ці рішення для управління всім парком.

1.2.3 Засоби загального призначення

Значна частина вітчизняних аграрних підприємств веде облік техніки у **Microsoft Excel**. Цей підхід не вимагає витрат на ліцензування спеціалізованого ПЗ, однак має суттєві недоліки: відсутність автоматичного контролю строків ТО, відсутність розмежування прав доступу, ризик втрати або пошкодження файлів, неможливість одночасної роботи кількох користувачів у режимі реального часу.

1.2.4 Порівняльний аналіз

Аналіз показав, що жодне з розглянутих рішень повною мірою не відповідає вимогам малих і середніх аграрних підприємств України одночасно за критеріями вартості, можливості локального розгортання, підтримки державної мови та наявності повного функціоналу управління технічним

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

обслуговуванням. Зазначений розрив обґрунтовує доцільність розроблення власної інформаційної системи.

Узагальнені результати аналізу наведено в таблиці 1.2.4.1

Таблиця 1.2.4.1 — Порівняльний аналіз існуючих рішень

Критерій	eMaint	Fiix	АгроПлан	MS Excel	Розроблювана система
Локальне розгортання	—	—	—	✓	✓
Безкоштовне використання	—	—	—	—	✓
Специфіка агро	—	—	✓	—	✓
Журнал ТО	✓	✓	частково	—	✓
Автонагадування	✓	✓	—	—	✓
Звіти PDF/CSV	✓	✓	✓	частково	✓
Розмежування доступу	✓	✓	✓	—	✓
Українська мова	—	—	✓	—	✓

1.3 Функціональні вимоги до програми

На підставі аналізу предметної галузі та недоліків існуючих рішень сформульовано функціональні вимоги до розроблюваної інформаційної системи.

1.3.1 Управління обліковими записами користувачів

- Система має підтримувати дві ролі: адміністратор і оператор.
- Адміністратор має право на повний доступ до всіх функцій системи, включаючи додавання, редагування та видалення записів.
- Оператор має доступ до перегляду даних та реєстрації записів обслуговування.
- Автентифікація здійснюється за логіном і паролем; паролі зберігаються у хешованому вигляді.
- Система має фіксувати дату та час останнього входу кожного користувача.

1.3.2 Облік сільськогосподарської техніки

- Система має забезпечувати зберігання таких атрибутів одиниці техніки: найменування, модель, рік випуску, серійний номер, статус, дата останнього ТО, міжсервісний інтервал (у місяцях), примітки.

- Підтримуються три значення статусу техніки: «Активна», «В ремонті», «Списана».
- Система автоматично обчислює дату наступного планового ТО на підставі дати останнього ТО та встановленого інтервалу.
- Серійний номер техніки має бути унікальним у межах системи.
- Підтримуються операції перегляду, додавання, редагування та видалення записів техніки.
- Видалення одиниці техніки тягне за собою каскадне видалення всіх пов'язаних записів обслуговування.

1.3.3 Журнал технічного обслуговування

- Система має забезпечувати реєстрацію записів ТО з атрибутами: одиниця техніки, дата проведення, тип (планове, ремонт, позапланове), опис виконаних робіт, вартість, виконавець.
- Після додавання запису ТО система автоматично оновлює дату останнього обслуговування відповідної одиниці техніки.
- Підтримуються фільтрація записів за технікою, типом обслуговування та діапазоном дат.
- Система розраховує та відображає загальну вартість обслуговування в розрізі обраного фільтра.

1.3.4 Контроль строків технічного обслуговування

- При запуску системи автоматично перевіряється наявність техніки, строк ТО якої настає протягом наступних 30 днів або вже прострочений.
- При виявленні такої техніки користувачеві відображається сповіщення з переліком відповідних машин та кількістю днів до наступного ТО (або кількістю днів прострочення).
- У переліку техніки відповідні записи виділяються кольором для візуального акцентування.

1.3.5 Пошук і фільтрація

- Система має підтримувати пошук техніки за найменуванням, моделлю, серійним номером, статусом і датою ТО.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

- Журнал обслуговування має підтримувати фільтрацію за технікою, типом обслуговування та діапазоном дат.

1.3.6 Звітність і експорт даних

- Система має формувати звіти у форматі **PDF**: зведений реєстр техніки; журнал обслуговування за обраний період; перелік техніки, що потребує ТО.
- Система має підтримувати експорт даних у форматі **CSV** для подальшого опрацювання в зовнішніх програмах.
- Звіти мають містити дату формування та підсумкові показники (загальна кількість, вартість тощо).

1.3.7 Журналювання дій

- Система має автоматично фіксувати дії користувачів (входи, додавання, редагування, видалення, експорт) із зазначенням логіну, дії, об'єкта та часу.

1.4 Вибір технологій розроблення

Вибір технологій здійснювався відповідно до функціональних вимог, необхідності локального розгортання на підприємстві без залежності від хмарної інфраструктури, а також з урахуванням поширеності та перспективності відповідних платформ.

1.4.1 Платформа та мова програмування

Для реалізації системи обрано платформу **.NET 9** та мову програмування **C#**. Зазначений вибір обґрунтовується такими чинниками:

- **Продуктивність і зрілість платформи.** .NET 9 є актуальною версією платформи Microsoft з підтримкою до 2026 року. Платформа забезпечує високу продуктивність виконання коду та широку екосистему бібліотек.
- **Строга типізація та об'єктна орієнтованість C#** сприяють написанню надійного, підтримуваного коду, зменшуючи ймовірність помилок під час розроблення.
- **Широке поширення у корпоративному секторі та наявність великої**

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

спільноти розробників забезпечують доступність документації та підтримки.

1.4.2 Технологія побудови інтерфейсу користувача

Для реалізації графічного інтерфейсу обрано **Windows Presentation Foundation (WPF)** — технологію побудови настільних застосунків для ОС Windows, що входить до складу платформи .NET.

Переваги WPF порівняно з альтернативами:

- **XAML-розмітка** забезпечує чітке розмежування між описом інтерфейсу та логікою застосунку, що спрощує підтримку та модифікацію UI.
- **Потужна система прив'язки даних (Data Binding)** дозволяє реалізовувати шаблон MVVM без надлишкового коду.
- **Векторна графіка та стилізація** надають можливість створення сучасного, візуально привабливого інтерфейсу.
- Рідне виконання на ОС Windows без необхідності встановлення браузера або веб-сервера.

Як архітектурний шаблон обрано **MVVM (Model-View-ViewModel)**, що забезпечує розмежування рівнів відображення, бізнес-логіки та даних, спрощує тестування та подальший розвиток системи.

1.4.3 Система управління базами даних

Для зберігання даних обрано **SQLite** — вбудовану реляційну СУБД, що не потребує окремого серверного процесу. Обґрунтування вибору:

- **Безсерверна архітектура** — база даних зберігається у єдиному файлі на диску, що суттєво спрощує розгортання системи: не потрібна конфігурація СУБД-сервера (Microsoft SQL Server, PostgreSQL тощо).
- **Повна підтримка SQL** та транзакційна цілісність даних відповідно до властивостей ACID.
- **Висока продуктивність** для однокористувацьких і малогрупових застосунків; достатня для зберігання даних підприємства з парком до кількох сотень одиниць техніки.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

- **Відкритий вихідний код** і відсутність ліцензійних обмежень.

Взаємодія застосунку з базою даних реалізована через **Entity Framework Core 9** — об'єктно-реляційний маппер (ORM), що дозволяє працювати з даними у вигляді об'єктів C# без написання SQL-запитів вручну. EF Core забезпечує автоматичне створення схеми бази даних та спрощує реалізацію репозиторіїв.

1.4.4 Додаткові бібліотеки

- **QuestPDF 2024.3** — бібліотека для генерації PDF-документів з відкритим вихідним кодом (Community License). Надає зручний Fluent API для програмного опису структури документу.
- **CsvHelper 33.0** — бібліотека для серіалізації та десеріалізації даних у форматі CSV з підтримкою локалізованих роздільників.
- **BCrypt.Net-Next 4.0** — реалізація алгоритму хешування паролів BCrypt, що забезпечує криптографічно стійке зберігання облікових даних користувачів.
- **Microsoft.Extensions.DependencyInjection 9.0** — контейнер впровадження залежностей, що забезпечує слабку зв'язаність компонентів системи.

1.5 Постановка задачі

На підставі проведеного аналізу предметної галузі, огляду існуючих рішень та сформульованих вимог визначено таку постановку задачі.

Розробити інформаційну систему обліку та технічного обслуговування сільськогосподарської техніки у вигляді настільного застосунку для ОС Windows, що реалізує такі функції:

1. **Автентифікацію** користувачів двох ролей — адміністратора та оператора — з хешуванням паролів за алгоритмом BCrypt.
2. **Облік техніки** — ведення реєстру машин з повним набором характеристик, підтримкою статусів, автоматичним обчисленням дати наступного ТО та пошуком за множинними критеріями.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

3. **Журнал технічного обслуговування** — реєстрацію записів про проведені планові та позапланові ТО і ремонти з можливістю фільтрації за технікою, типом і діапазоном дат.
4. **Автоматичний контроль строків ТО** — при запуску системи перевірку та відображення переліку техніки, строк обслуговування якої настає або вже прострочений.
5. **Звітність** — формування звітів у форматах PDF і CSV: реєстр техніки, журнал обслуговування за період, нагадування про ТО.
6. **Журналювання** — автоматичну фіксацію дій користувачів у системному журналі.

Система має функціонувати локально без необхідності підключення до мережі Інтернет, розгортатися без встановлення окремого СУБД-сервера та мати повністю україномовний інтерфейс.

Висновки до розділу 1

У першому розділі виконано аналіз предметної галузі та сформульовано постановку задачі кваліфікаційної роботи.

Встановлено, що управління машинно-тракторним парком аграрного підприємства охоплює дві взаємопов'язані задачі — облік техніки та управління технічним обслуговуванням, — автоматизація яких є необхідною умовою ефективного функціонування господарства.

Проведений огляд існуючих програмних рішень — CMMS-платформ (eMaint, Fiix), спеціалізованих аграрних систем (АгроПлан, фірмові платформи виробників) та засобів загального призначення (Microsoft Excel) — показав, що жодне з них не задовольняє одночасно вимогам доступності, локального розгортання, підтримки державної мови та повноти функціоналу для умов вітчизняних аграрних підприємств.

Сформульовано детальні функціональні вимоги до розроблюваної системи, які охоплюють управління обліковими записами, облік техніки, ведення журналу ТО, контроль строків обслуговування, пошук і фільтрацію,

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

формування звітності та журналювання дій.

Обґрунтовано вибір технологічного стеку: платформа .NET 9 і мова C# як основа розроблення; WPF з шаблоном MVVM як технологія побудови інтерфейсу; SQLite з Entity Framework Core 9 як рівень зберігання даних; QuestPDF, CsvHelper та BCrypt.Net як допоміжні бібліотеки. Сукупність зазначених технологій забезпечує реалізацію всіх висунутих вимог при мінімальній складності розгортання системи на підприємстві.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

2.1 Модель бази даних

2.1.1 Опис предметної галузі та основних сутностей

Проектування моделі бази даних є ключовим етапом розроблення інформаційної системи, що визначає спосіб організації, зберігання та взаємозв'язку даних. Коректно спроектована модель забезпечує цілісність даних, унеможливорює їх дублювання та створює основу для ефективної реалізації бізнес-логіки застосунку.

Предметна галузь розроблюваної системи охоплює процеси обліку сільськогосподарської техніки та управління її технічним обслуговуванням на аграрному підприємстві. На підставі аналізу предметної галузі, проведеного у розділі 1, визначено чотири основні сутності, що підлягають зберігання в реляційній базі даних: техніка (Techniky), обслуговування (Obslugovuvannya), користувачі (Koristuvachi) та журнал дій (ActionLogs).

Сутність «Техніка» (Techniky)

Сутність «Техніка» є центральною в моделі та описує одиницю сільськогосподарської машини, що перебуває на обліку підприємства.

Таблиця 2.1.1.1 — Атрибути сутності «Техніка»

Атрибут	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AUTOINCREMENT	Унікальний числовий ідентифікатор запису
Nazva	VARCHAR(200)	NOT NULL	Найменування одиниці техніки
Model	VARCHAR(200)	NOT NULL	Повна назва моделі машини
RikVypusku	INTEGER	NOT NULL	Рік виготовлення
SerijnyjNomer	VARCHAR(100)	NOT NULL, UNIQUE	Серійний (заводський) номер машини
Status	INTEGER	NOT NULL, DEFAULT 0	Поточний статус: 0 — Активна, 1 — В ремонті, 2 — Списана
DataOstannogTO	DATE	NULL	Дата проведення останнього технічного обслуговування
TOIntervalMisyatsiv	INTEGER	NOT NULL, DEFAULT 3	Міжсервісний інтервал у місяцях

Prymitky	VARCHAR(500)	NULL	Службові примітки щодо особливостей експлуатації
DataDodavannya	DATETIME	NOT NULL	Дата та час внесення запису до системи

Первинний ключ: атрибут Id — сурогатний числовий ключ із автоінкрементом.

Унікальний індекс: атрибут SerijnyjNomer — унікальність серійного номера забезпечується індексом на рівні СУБД, що унеможливлює дублювання машин.

Обчислювані атрибути (реалізовані на рівні застосунку, не зберігаються в БД):

- DataNastvnoгТО — дата наступного планового ТО, обчислюється як DataOstannogTO + TOIntervalMisyatsiv місяців;
- PotrebuyeTO — логічна ознака того, що машина потребує обслуговування (дата наступного ТО настала або настане протягом 30 днів).

Класифікація атрибутів:

- *Прості однозначні:* Id, RikVypusku, TOIntervalMisyatsiv, DataOstannogTO, DataDodavannya — зберігають єдине неподільне значення;
- *Прості однозначні з доменним обмеженням:* Status — приймає одне з трьох фіксованих числових значень перелічуваного типу;
- *Текстові однозначні:* Nazva, Model, SerijnyjNomer, Prymitky — рядки з обмеженням максимальної довжини;
- *Похідні (обчислювані):* DataNastvnoгТО, PotrebuyeTO — не зберігаються в БД.

Сутність «Обслуговування» (Obslugovuvannya)

Сутність «Обслуговування» фіксує кожен окремий факт проведення технічного обслуговування або ремонту для конкретної одиниці техніки. Є залежною від сутності «Техніка».

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Таблиця 2.1.1.2 — Атрибути сутності «Обслуговування»

Атрибут	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AUTOINCREMENT	Унікальний числовий ідентифікатор запису
TechnikaId	INTEGER	NOT NULL, FK → Techniky.Id	Зовнішній ключ до сутності «Техніка»
DataTO	DATE	NOT NULL	Дата фактичного проведення обслуговування
Тип	INTEGER	NOT NULL	Тип: 0 — Планове, 1 — Ремонт, 2 — Позапланове
Opys	VARCHAR(1000)	NOT NULL	Детальний опис виконаних робіт
Vartist	DECIMAL	NOT NULL, DEFAULT 0	Вартість робіт у гривнях
Vykonavets	VARCHAR(200)	NULL	Виконавець (ПІБ або назва організації)
DataDodavannya	DATETIME	NOT NULL	Дата та час внесення запису до системи

Первинний ключ: атрибут Id.

Зовнішній ключ: атрибут TechnikaId — посиляється на Id сутності «Техніка». При видаленні запису техніки всі пов'язані записи обслуговування видаляються каскадно (ON DELETE CASCADE).

Класифікація атрибутів:

- *Прості однозначні:* Id, TechnikaId, DataTO, Vartist, DataDodavannya;
- *Прості однозначні з доменним обмеженням:* Тип;
- *Текстові однозначні:* Opys, Vykonavets.

Сутність «Користувач» (Koristuvachi)

Сутність «Користувач» зберігає облікові дані осіб, яким надано доступ до системи, а також інформацію про їхню роль та активність.

Таблиця 2.1.1.3 — Атрибути сутності «Користувач»

Атрибут	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AUTOINCREMENT	Унікальний числовий ідентифікатор
Login	VARCHAR(100)	NOT NULL, UNIQUE	Логін для входу до системи
PasswordHash	TEXT	NOT NULL	BCrypt-хеш пароля користувача
PovneImya	VARCHAR(200)	NULL	Повне ім'я (прізвище, ім'я, по батькові)

Rol	INTEGER	NOT NULL	Роль: 0 — Адміністратор, 1 — Оператор
Aktivnyj	BOOLEAN	NOT NULL, DEFAULT 1	Ознака активності облікового запису
DataReyestratsiyi	DATETIME	NOT NULL	Дата та час створення облікового запису
OstannijVxid	DATETIME	NULL	Дата та час останньої успішної автентифікації

Первинний ключ: атрибут Id.

Унікальний індекс: атрибут Login — унікальність логіну забезпечується індексом на рівні СУБД.

Примітка щодо безпеки: атрибут PasswordHash ніколи не містить відкритого пароля. При автентифікації введений пароль порівнюється з хешем засобами бібліотеки BCrypt.Net, що унеможливорює компрометацію паролів навіть у разі несанкціонованого доступу до файлу бази даних.

Класифікація атрибутів:

- *Прості однозначні:* Id, Rol, Aktivnyj, DataReyestratsiyi, OstannijVxid;
- *Прості однозначні з доменним обмеженням:* Rol;
- *Текстові однозначні:* Login, PasswordHash, PovneImya.

Сутність «Журнал дій» (ActionLogs)

Сутність «Журнал дій» забезпечує аудит подій у системі: фіксує кожну значущу дію користувача з зазначенням часу, об'єкта дії та деталей.

Таблиця 2.1.1.4 — Атрибути сутності «Журнал дій»

Атрибут	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AUTOINCREMENT	Унікальний числовий ідентифікатор запису
KoristuvachId	INTEGER	NULL, FK → Koristuvachi.Id	Зовнішній ключ до сутності «Користувач»
KoristuvachLogin	VARCHAR(100)	NULL	Логін користувача на момент дії (денормалізація)
Diia	VARCHAR(500)	NOT NULL	Назва виконаної дії (наприклад, «Додавання», «Видалення»)
Sutnist	VARCHAR(100)	NULL	Об'єкт дії (наприклад, «Техніка», «Обслуговування»)
Chas	DATETIME	NOT NULL	Дата та час виконання дії
Detali	VARCHAR(1000)	NULL	Додаткові деталі (наприклад, назва запису, шлях до файлу)

Первинний ключ: атрибут Id.

Зовнішній ключ: атрибут KoristuvachId — допускає значення NULL для можливості фіксації системних подій, не пов'язаних із конкретним користувачем.

Денормалізований атрибут: KoristuvachLogin зберігається окремо від зовнішнього ключа, що дозволяє зберегти читабельність записів журналу навіть у разі видалення облікового запису користувача в майбутньому.

2.1.2 ER-діаграма (Entity-Relationship Diagram)

ER-діаграма є графічним засобом формалізованого опису структури бази даних, що відображає сутності предметної галузі, їх атрибути та зв'язки між ними. Для розроблюваної системи побудовано ER-діаграму, що включає чотири сутності та два явні зв'язки між ними.

Перелік сутностей:

1. **Techniky** — реєстр одиниць сільськогосподарської техніки;
2. **Obslugovuvannya** — журнал технічного обслуговування та ремонтів;
3. **Koristuvachi** — облікові записи користувачів системи;
4. **ActionLogs** — системний журнал аудиту дій користувачів.

Зв'язки між сутностями та їх кардинальність:

Зв'язок 1: Techniky → Obslugovuvannya (1 : N)

Одна одиниця техніки може мати нуль або більше записів обслуговування; кожен запис обслуговування належить рівно одній одиниці техніки. Тип зв'язку — «один до багатьох» (1:N). Зв'язок реалізовано через зовнішній ключ TechnikaId у таблиці Obslugovuvannya. При видаленні запису техніки всі пов'язані записи обслуговування видаляються автоматично (ON DELETE CASCADE), що забезпечує цілісність даних.

- Мінімальна кардинальність з боку «Техніка»: 1 (кожен запис обслуговування зобов'язаний посилатися на існуючу техніку);
- Мінімальна кардинальність з боку «Обслуговування»: 0 (техніка може не мати жодного запису ТО).

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Зв'язок 2: Koristuvachi → ActionLogs (1 : N, необов'язковий)

Один користувач може бути пов'язаний із нулем або більше записів журналу дій; кожен запис журналу може посилатися на одного користувача або не мати такого посилання (NULL). Тип зв'язку — «один до багатьох» (1:N), необов'язковий з боку «Журнал дій». Реалізовано через nullable зовнішній ключ KoristuvachId у таблиці ActionLogs.

- Зовнішній ключ допускає NULL для фіксації системних подій (наприклад, автоматична перевірка ТО при запуску);
- Логін користувача додатково зберігається в атрибуті KoristuvachLogin для забезпечення читабельності журналу після можливого видалення облікового запису.

Зв'язки між сутностями «Техніка» та «Користувач» не є прямими: користувачі взаємодіють з технікою виключно через інтерфейс застосунку, а факт цієї взаємодії фіксується в журналі дій.

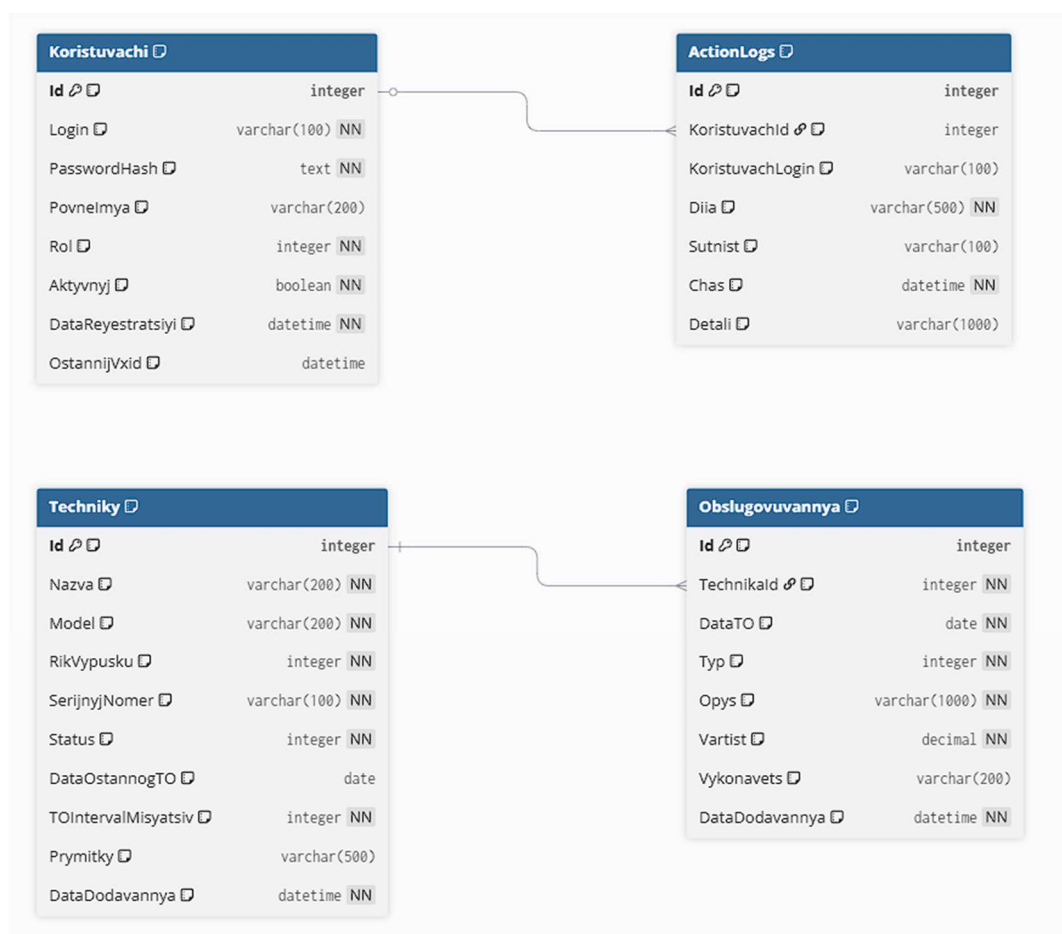


Рисунок 2.1.1. – ER-діаграма

Зв'язок «Техніка — Обслуговування» (1:N, обов'язковий)

Параметр	Значення
Сутності	Techniky (батьківська), Obslugovuvannya (дочірня)
Тип	Один до багатьох (1:N)
Назва	«має записи обслуговування»
Реалізація	FK: Obslugovuvannya.TechnikaId → Techniky.Id
Атрибути зв'язку	— (зв'язок типу 1:N не потребує окремих атрибутів)
Поведінка при видаленні	ON DELETE CASCADE
Обов'язковість	З боку «Обслуговування» — обов'язковий (NOT NULL FK); з боку «Техніка» — необов'язковий (техніка може не мати записів ТО)

Семантика зв'язку: одна машина (Technika) може пройти необмежену кількість обслуговувань (Obslugovuvannya), однак кожен запис обслуговування обов'язково стосується конкретної машини. Каскадне видалення забезпечує узгодженість даних: при списанні машини з реєстру її повна історія обслуговувань видаляється автоматично, без ризику появи «осиротілих» записів (orphaned records).

Зв'язок «Користувач — Журнал дій» (1:N, необов'язковий)

Параметр	Значення
Сутності	Koristuvachi (батьківська), ActionLogs (дочірня)
Тип	Один до багатьох (1:N)
Назва	«виконав дії»
Реалізація	FK: ActionLogs.KoristuvachId → Koristuvachi.Id (nullable)
Атрибути зв'язку	—
Поведінка при видаленні	Без каскаду (SET NULL або RESTRICT на розсуд СУБД)
Обов'язковість	Необов'язковий з обох боків

Семантика зв'язку: один користувач може бути автором нуля або більше записів у журналі; запис журналу може не мати прив'язки до конкретного користувача (наприклад, системні події). Збереження KoristuvachLogin безпосередньо в таблиці ActionLogs є свідомою денормалізацією, що забезпечує читабельність журналу після можливого видалення або перейменування облікового запису, зберігаючи при цьому цілісний аудитний слід.

Забезпечення цілісності даних та уникнення надмірності

Розроблена модель забезпечує цілісність даних і мінімізацію надмірності через такі механізми:

1. Обмеження цілісності на рівні СУБД:

- Обмеження NOT NULL на ключових атрибутах унеможливлюють появу неповних записів;
- Унікальні індекси на SerijnyjNomer та Login виключають дублювання машин і облікових записів;
- Зовнішні ключі з каскадним видаленням (ON DELETE CASCADE) гарантують відсутність «осиротілих» записів обслуговування;
- Обмеження доменів (перелічувані значення для Status, Typ, Rol) забезпечують семантичну коректність даних.

2. Нормалізація до третьої нормальної форми (3НФ):

- Кожна таблиця містить дані виключно про одну сутність, без транзитивних залежностей;
- Дані про техніку та дані про її обслуговування зберігаються окремо, що унеможливлює дублювання характеристик машини в кожному записі ТО;
- Дані користувачів зберігаються в окремій таблиці та пов'язуються з журналом дій через зовнішній ключ, а не дублюються в кожному рядку журналу.

3. Свідома денормалізація з обґрунтуванням:

- Атрибут KoristuvachLogin у таблиці ActionLogs є єдиним свідомим відступом від строгої нормалізації. Він реалізований для збереження цілісності аудитного журналу: у разі видалення або зміни логіну користувача архівні записи журналу зберігають оригінальну ідентифікаційну інформацію, що є обов'язковою вимогою до систем аудиту.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

4. Обчислювані атрибути поза базою даних:

- Атрибути DataNastvnogTO та PotrebuyeTO сутності «Техніка» обчислюються на рівні доменної моделі застосунку (клас Technika у C#, атрибут [NotMapped]), а не зберігаються в БД. Це виключає проблему розсинхронізації обчислених значень із вихідними даними та гарантує їх актуальність на момент звернення.

2.1.3 Фізична модель бази даних

Фізична модель бази даних описує конкретну реалізацію логічної структури у термінах обраної СУБД — SQLite. На відміну від концептуальної моделі, фізична модель фіксує точні типи даних, обмеження цілісності, індекси та синтаксис визначення таблиць. SQLite використовує систему динамічної типізації зі спорідненістю типів (type affinity), тому всі текстові та дата-атрибути зберігаються як TEXT, числові — як INTEGER або REAL.

Фізична модель розроблюваної системи містить чотири таблиці: Techniku, Obslugovuvannya, Koristuvachi та ActionLogs.

Таблиця 2.1.3.1 — Фізична структура таблиці «Techniku»

№	Поле	Тип даних SQLite	Обмеження	Опис
1	Id	INTEGER	PK, AUTOINCREMENT, NOT NULL	Сурогатний первинний ключ
2	Nazva	TEXT	NOT NULL	Найменування техніки
3	Model	TEXT	NOT NULL	Модель машини
4	RikVypusku	INTEGER	NOT NULL	Рік виготовлення
5	SerijnyjNomer	TEXT	NOT NULL, UNIQUE	Серійний номер (унікальний індекс)
6	Status	INTEGER	NOT NULL	0=Активна, 1=В ремонті, 2=Списана
7	DataOstannogTO	TEXT	NULL	Дата останнього ТО (ISO 8601)
8	TOIntervalMisyatsiv	INTEGER	NOT NULL	Міжсервісний інтервал (місяців)
9	Prymitky	TEXT	NOT NULL	Службові примітки
10	DataDodavannya	TEXT	NOT NULL	Дата додавання запису (ISO 8601)

Індекси таблиці Techniky:

- PK_Techniky — кластерний первинний ключ по полю Id;
- IX_Techniky_SerijnyjNomer — унікальний індекс по полю SerijnyjNomer.

DDL-визначення:

```
CREATE TABLE Techniky (  
    Id INTEGER NOT NULL  
        CONSTRAINT PK_Techniky PRIMARY KEY AUTOINCREMENT,  
    Nazva TEXT NOT NULL,  
    Model TEXT NOT NULL,  
    RikVypusku INTEGER NOT NULL,  
    SerijnyjNomer TEXT NOT NULL,  
    Status INTEGER NOT NULL,  
    DataOstannogTO TEXT,  
    TOIntervalMisyatsiv INTEGER NOT NULL,  
    Prymitky TEXT NOT NULL,  
    DataDodavannya TEXT NOT NULL  
);  
CREATE UNIQUE INDEX IX_Techniky_SerijnyjNomer  
ON Techniky (SerijnyjNomer);
```

Таблиця 2.1.3.2 — Фізична структура таблиці «Obslugovuvannya»

№	Поле	Тип даних SQLite	Обмеження	Опис
1	Id	INTEGER	PK, AUTOINCREMENT, NOT NULL	Сурогатний первинний ключ
2	TechnikaId	INTEGER	NOT NULL, FK → Techniky.Id	Зовнішній ключ до таблиці Techniky
3	DataTO	TEXT	NOT NULL	Дата проведення ТО (ISO 8601)
4	Typ	INTEGER	NOT NULL	0=Планове, 1=Ремонт, 2=Позапланове
5	Opys	TEXT	NOT NULL	Опис виконаних робіт
6	Vartist	TEXT	NOT NULL	Вартість робіт (зберігається як TEXT для збереження точності decimal)
7	Vykonavets	TEXT	NOT NULL	Виконавець (ПІБ або організація)
8	DataDodavannya	TEXT	NOT NULL	Дата додавання запису (ISO 8601)

Індекси таблиці Obslugovuvannya:

- PK_Obslugovuvannya — кластерний первинний ключ по полю Id;
- IX_Obslugovuvannya_TechnikaId — некластерний індекс по зовнішньому ключу TechnikaId (створюється автоматично Entity Framework Core для прискорення виконання JOIN-запитів).

Обмеження зовнішнього ключа:

- FK_Obslugovuvannya_Techniky_TechnikaId — посиляється на Techniky(Id), поведінка при видаленні батьківського запису: ON DELETE CASCADE.

DDL-визначення:

```
CREATE TABLE Obslugovuvannya (  
    Id INTEGER NOT NULL  
        CONSTRAINT PK_Obslugovuvannya PRIMARY KEY AUTOINCREMENT,  
    TechnikaId INTEGER NOT NULL,  
    DataTO TEXT NOT NULL,  
    Typ INTEGER NOT NULL,  
    Opys TEXT NOT NULL,  
    Vartist TEXT NOT NULL,  
    Vykonavets TEXT NOT NULL,  
    DataDodavannya TEXT NOT NULL,  
    CONSTRAINT FK_Obslugovuvannya_Techniky_TechnikaId  
        FOREIGN KEY (TechnikaId)  
        REFERENCES Techniky (Id) ON DELETE CASCADE  
);
```

Таблиця 2.1.3.3 — Фізична структура таблиці «Koristuvachi»

№	Поле	Тип даних SQLite	Обмеження	Опис
1	Id	INTEGER	PK, AUTOINCREMENT, NOT NULL	Сурогатний первинний ключ
2	Login	TEXT	NOT NULL, UNIQUE	Логін для входу (унікальний індекс)
3	PasswordHash	TEXT	NOT NULL	BCrypt-хеш пароля
4	PovneImya	TEXT	NOT NULL	Повне ім'я (ПІБ)
5	Rol	INTEGER	NOT NULL	0=Адміністратор, 1=Оператор
6	Aktivnyj	INTEGER	NOT NULL	1=Активний, 0=Деактивований
7	DataReyestratsiyi	TEXT	NOT NULL	Дата реєстрації облікового запису
8	OstannijVxid	TEXT	NULL	Дата останнього успішного входу

Індекси таблиці Koristuvachi:

- PK_Koristuvachi — кластерний первинний ключ по полю Id;
- IX_Koristuvachi_Login — унікальний індекс по полю Login.

DDL-визначення:

```
CREATE TABLE Koristuvachi (  
    Id INTEGER NOT NULL  
        CONSTRAINT PK_Koristuvachi PRIMARY KEY AUTOINCREMENT,  
    Login TEXT NOT NULL,  
    PasswordHash TEXT NOT NULL,  
    PovneImya TEXT NOT NULL,  
    Rol INTEGER NOT NULL,  
    Aktyvnyj INTEGER NOT NULL,  
    DataReyestratsiyi TEXT NOT NULL,  
    OstannijVxid TEXT  
);  
CREATE UNIQUE INDEX IX_Koristuvachi_Login  
ON Koristuvachi (Login);
```

Таблиця 2.1.3.4 — Фізична структура таблиці «ActionLogs»

№	Поле	Тип даних SQLite	Обмеження	Опис
1	Id	INTEGER	PK, AUTOINCREMENT, NOT NULL	Сурогатний первинний ключ
2	KoristuvachId	INTEGER	NULL, FK → Koristuvachi.Id	Зовнішній ключ (nullable)
3	KoristuvachLogin	TEXT	NOT NULL	Логін користувача (денормалізація)
4	Diia	TEXT	NOT NULL	Назва виконаної дії
5	Sutnist	TEXT	NOT NULL	Об'єкт, над яким виконано дію
6	Chas	TEXT	NOT NULL	Дата та час дії (ISO 8601)
7	Detali	TEXT	NOT NULL	Додаткові деталі події

Індекси таблиці ActionLogs:

- PK_ActionLogs — кластерний первинний ключ по полю Id.

Примітка: зовнішній ключ KoristuvachId є необов'язковим (NULL) для можливості фіксації системних подій. На відміну від таблиці Obslugovuvannya, каскадне видалення не застосовується: записи журналу зберігаються незалежно від стану облікового запису користувача з метою збереження повноти аудитного сліду.

DDL-визначення:

```
CREATE TABLE ActionLogs (  
    Id INTEGER NOT NULL  
        CONSTRAINT PK_ActionLogs PRIMARY KEY AUTOINCREMENT,  
    KoristuvachId INTEGER,  
    KoristuvachLogin TEXT NOT NULL,  
    Diia TEXT NOT NULL,  
    Sutnist TEXT NOT NULL,  
    Chas TEXT NOT NULL,  
    Detali TEXT NOT NULL  
);
```

Таблиця 2.1.3.5 — Структура таблиць бази даних

Таблиця	Кількість полів	PK	FK	Унікальні індекси	CASCADE
Techniky	10	Id	—	SerijnyjNomer	—
Obslugovuvannya	8	Id	TechnikaId → Techniky.Id	—	ON DELETE CASCADE
Koristuvachi	8	Id	—	Login	—
ActionLogs	7	Id	KoristuvachId → Koristuvachi.Id (nullable)	—	—
Разом	33	4	2	2	1

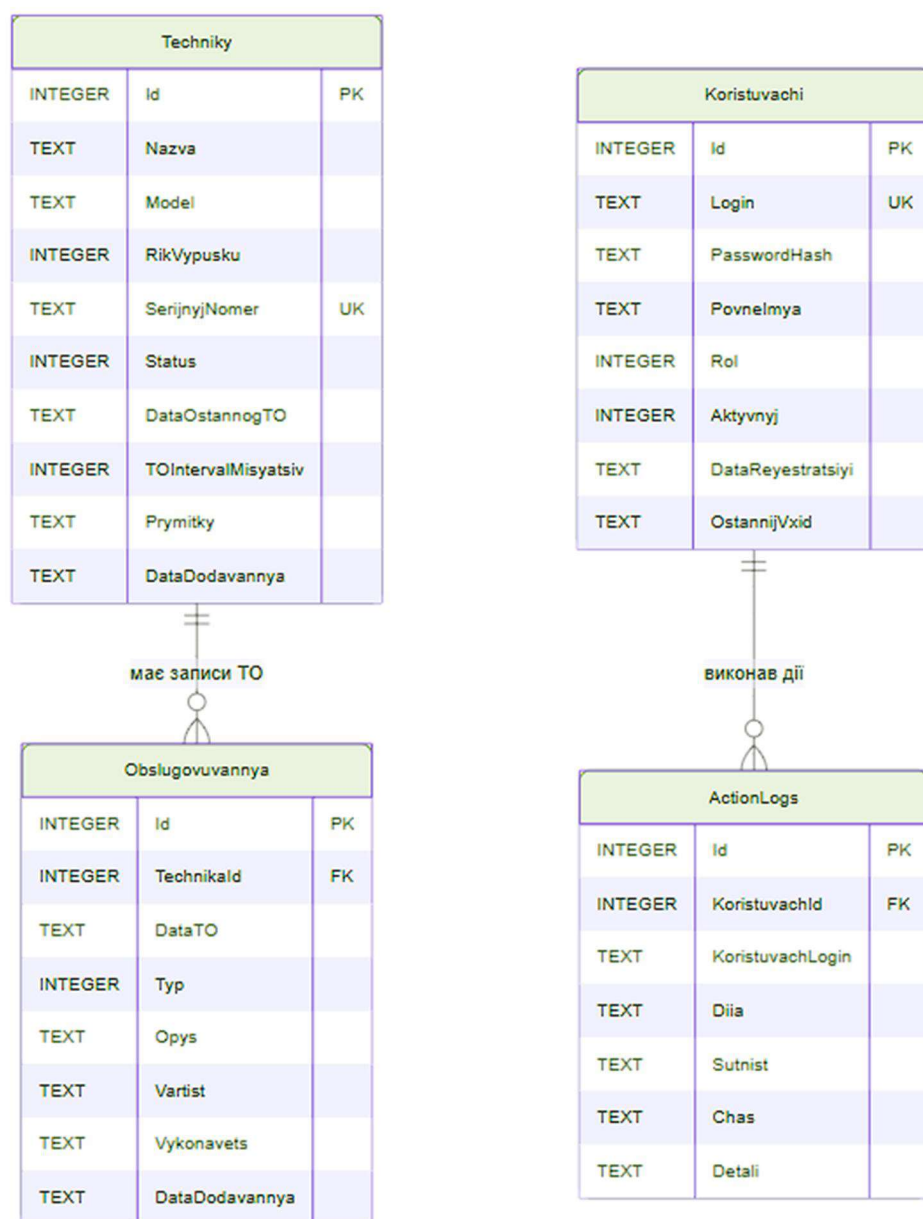


Рисунок 2.1.2. – Схема бази даних

2.2 Список основних форм та їх призначення

Графічний інтерфейс розроблюваної системи складається з семи форм (вікон та діалогів), кожна з яких відповідає за окремий функціональний блок. Усі форми реалізовані засобами WPF з дотриманням єдиного стилю відповідно до ресурсного словника Styles.xaml.

Таблиця 2.2.1 — Перелік форм системи

№	Клас форми	Тип	Призначення
1	LoginWindow	Window	Автентифікація користувача
2	MainWindow	Window	Головне вікно з навігацією
3	TechnikaView	UserControl	Облік техніки
4	ObslugovuvannyaView	UserControl	Журнал обслуговування
5	ReportsView	UserControl	Формування звітів
6	AddEditTechnikaDialog	Window (Dialog)	Додавання / редагування техніки
7	AddEditObslugovuvannyaDialog	Window (Dialog)	Додавання / редагування запису ТО

2.2.1 Вікно автентифікації

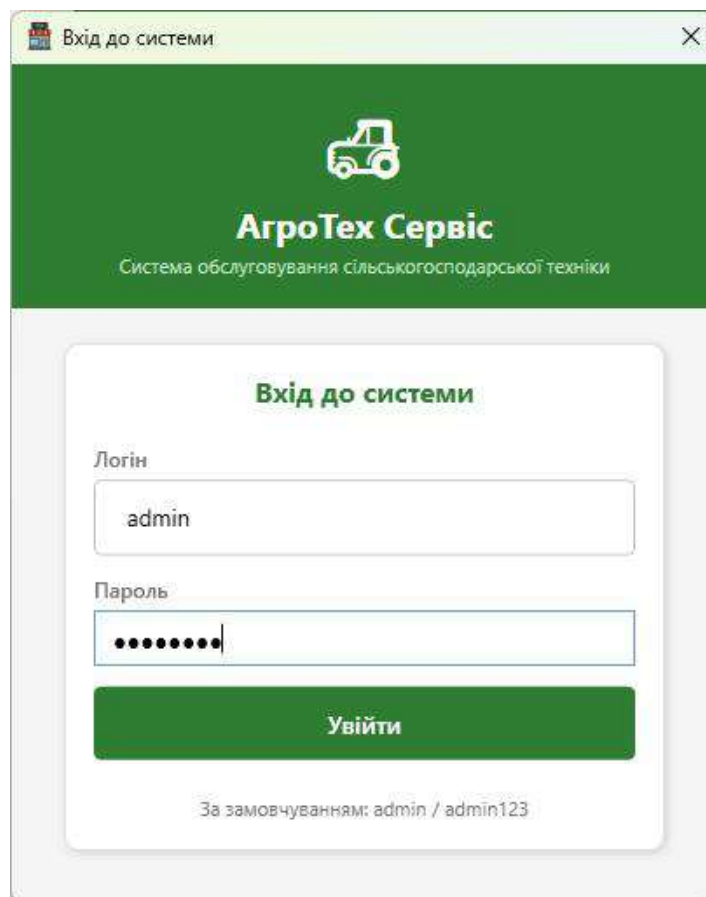


Рисунок 2.2.1. – Вікно автентифікації (LoginWindow)

LoginWindow є першою формою, що відображається при запуску застосунку. Вона реалізує процес автентифікації користувача та забезпечує контрольований вхід до системи. Верхня частина вікна містить брендований заголовок із логотипом та найменуванням системи, нижня — форму введення облікових даних.

Форма містить такі елементи: поле введення логіну (TextBox), поле введення пароля (PasswordBox — стандартний елемент WPF, що маскує введений текст), кнопку «Увійти» та блок відображення помилки автентифікації. Блок помилки є прихованим за замовчуванням і стає видимим лише при невдалій спробі входу — через прив'язку Visibility до boolean-властивості HasErrorMessage у LoginViewModel. При успішній автентифікації вікно закривається та відкривається MainWindow; перевірка пароля здійснюється функцією BCrypt.Verify() без передачі відкритого пароля до рівня даних.

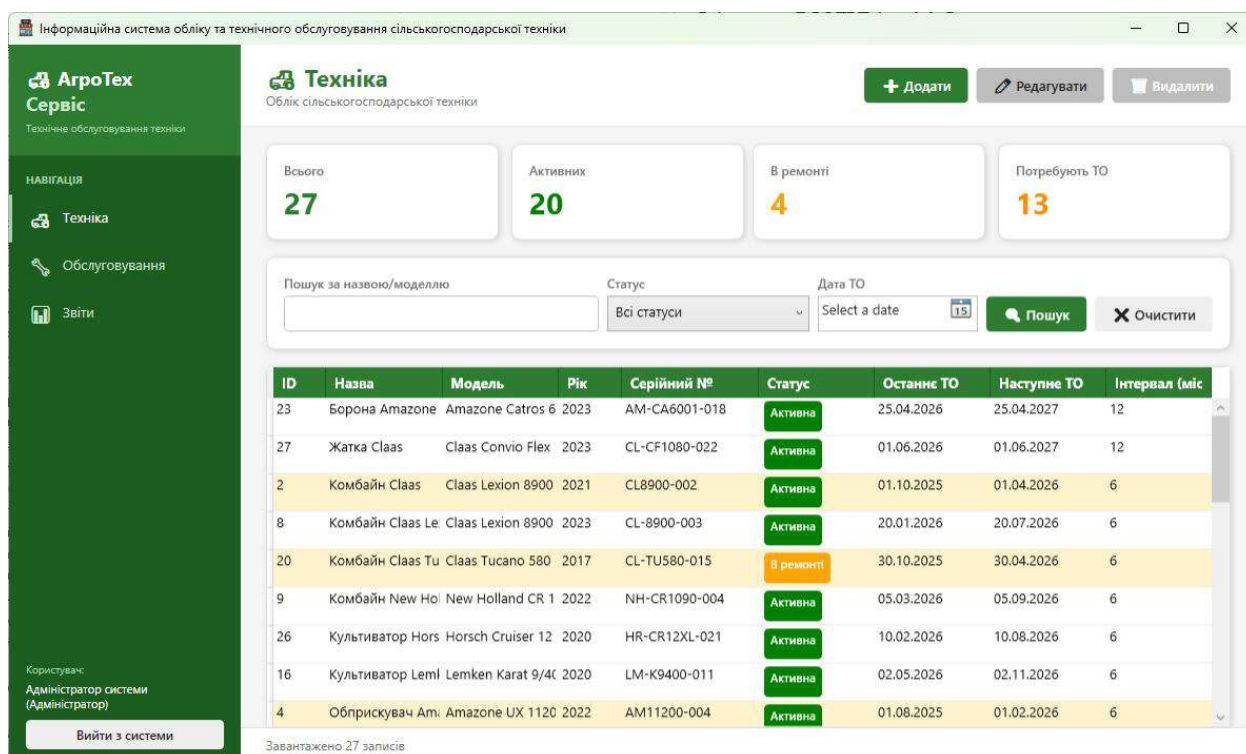


Рисунок 2.2.2. – Головне вікно системи з активним розділом «Техніка»

MainWindow є контейнерним вікном застосунку, що відображається після успішної автентифікації та залишається відкритим протягом усього сеансу роботи. Вікно реалізує дволічинну компоновання: ліворуч розташована

панель навігації фіксованої ширини (220 px), праворуч — область основного контенту, яка динамічно перемикається між UserControl-компонентами (TechnikaView, ObslugovuvannyaView, ReportsView) залежно від обраного розділу.

Панель навігації містить логотип системи, кнопки переходу між розділами та блок інформації про поточного користувача з кнопкою виходу. Активний пункт меню виділяється кольоровою бічною смугою. Перемикання між розділами реалізовано через команду `NavigateCommand` у `MainViewModel`, яка змінює булеві властивості видимості (`IsTechnikaVisible`, `IsObslugovuvannyaVisible`, `IsReportsVisible`) та ініціює завантаження даних відповідного UserControl. При запуску форми автоматично викликається `InitializeAsync()`, що завантажує дані про техніку та ініціює перевірку строків ТО.

2.2.2 Перегляд та управління технікою

`TechnikaView` є основним робочим розділом для управління реєстром сільськогосподарської техніки. Форма реалізована як UserControl і відображається в правій частині `MainWindow`. Компонування форми є вертикальним і включає п'ять горизонтальних зон.

Перша зона — **заголовок** — містить назву розділу та блок кнопок управління: «Додати», «Редагувати», «Видалити». Кнопки «Редагувати» та «Видалити» доступні лише за умови вибору рядка в таблиці (прив'язка `IsEnabled` до властивості `HasSelection`).

Друга зона — **статистичні картки** — відображає чотири показники в режимі реального часу: загальну кількість одиниць техніки, кількість активних, кількість машин у стані ремонту та кількість машин, що потребують ТО. Значення оновлюються автоматично після кожної операції зміни даних.

Третя зона — **панель фільтрації** — надає можливість пошуку за текстовим рядком (назва або модель), статусом (випадаючий список) та датою останнього ТО (календарний вибір). Активується кнопкою «Пошук»; скидання фільтрів — кнопкою «Очистити».

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

Четверта зона — **таблиця даних (DataGrid)** — відображає відфільтровані записи. Рядки таблиці кольорово кодуються: машини, що потребують ТО, виділяються жовтуватим фоном через конвертер `TOWarningColorConverter`. Виділений рядок підсвічується зеленим.

П'ята зона — **рядок статусу** — відображає поточне повідомлення (`StatusMessage`): кількість завантажених записів, результат пошуку або підтвердження операції.

2.2.3 Форма журналу технічного обслуговування

Інформаційна система обліку та технічного обслуговування сільськогосподарської техніки

Обслуговування
Журнал технічного обслуговування та ремонтів

+ Додати Редагувати Видалити

Всього записів: **30** Планових ТО: **22** Ремонтів: **5** Загальна вартість: **498,400.00 грн**

Техніка: — Вся техніка — Тип обслуговування: Всі типи Дата від: 04.01.2026 Дата до: 04.04.2026 Пошук Очистити

ID	Техніка	Модель	Дата ТО	Тип	Опис	Вартість (грн)	Виконавець
30	Трактор New Hc	New Holland T7	10.06.2026	Планове	Переджнивне ТО комбайна: пер	19,500.00	Бондар П.П.
29	Трактор Deutz-f	Deutz-Fahr Agro	01.06.2026	Планове	Підготовка жатки до жнив: замін	18,600.00	Сервісний цент
26	Трактор Case IH	Case IH Magnum	15.05.2026	Планове	ТО прес-підбирача: заміна моту:	8,900.00	Лисенко М.В.
28	Сівалка Horsch	Horsch Maestro	10.05.2026	Позапланове	Позапланова перевірка через пк	9,800.00	Петренко В.С.
27	Трактор Fendt 2	Fendt 724 Vario	02.05.2026	Планове	Сезонне ТО культиватора: замін	5,400.00	Коваленко О.І.
22	Трактор Fendt 2	Fendt 724 Vario	25.04.2026	Планове	Підготовка борони: заміна знош	16,800.00	Коваленко О.І.
23	Трактор New Hc	New Holland T7	25.04.2026	Планове	ТО борони Amazone: заміна різа	14,200.00	Мельник Т.М.
24	Комбайн Claas	Claas Lexion 890	20.04.2026	Ремонт	Ремонт двигуна: заміна паливни:	31,000.00	Сервісний цент
21	Трактор John De	John Deere 8R 3	10.04.2026	Планове	Передсезонне ТО обприскувача	13,500.00	Бондар П.П.

Завантажено 30 записів

Рисунок 2.2.3. — Форма журналу технічного обслуговування

`ObslugovuvannyaView` реалізує повний функціонал ведення журналу технічного обслуговування та ремонтів. Структурно форма є аналогічною до `TechnikaView`, проте адаптована до специфіки даних про сервісні роботи.

Блок статистичних карток відображає чотири показники: загальну кількість записів, кількість планових ТО, кількість ремонтів та загальну вартість усіх робіт за поточним фільтром. Показник загальної вартості є ключовим управлінським індикатором і обчислюється через `Sum()` колекції в пам'яті після кожного перезавантаження. Панель фільтрації передбачає вибір техніки з випадаючого списку (при значенні «— Вся техніка —» фільтрація за

цим критерієм вимикається), типу обслуговування та діапазону дат. Таблиця даних відображає: техніку та її модель, дату ТО, тип (кольорові бейджі: зелений — планове, червоний — ремонт, синій — позапланове), опис робіт, вартість та виконавця. Рядки чергуються за кольором фону для покращення читабельності.

2.2.4 Формування звітів

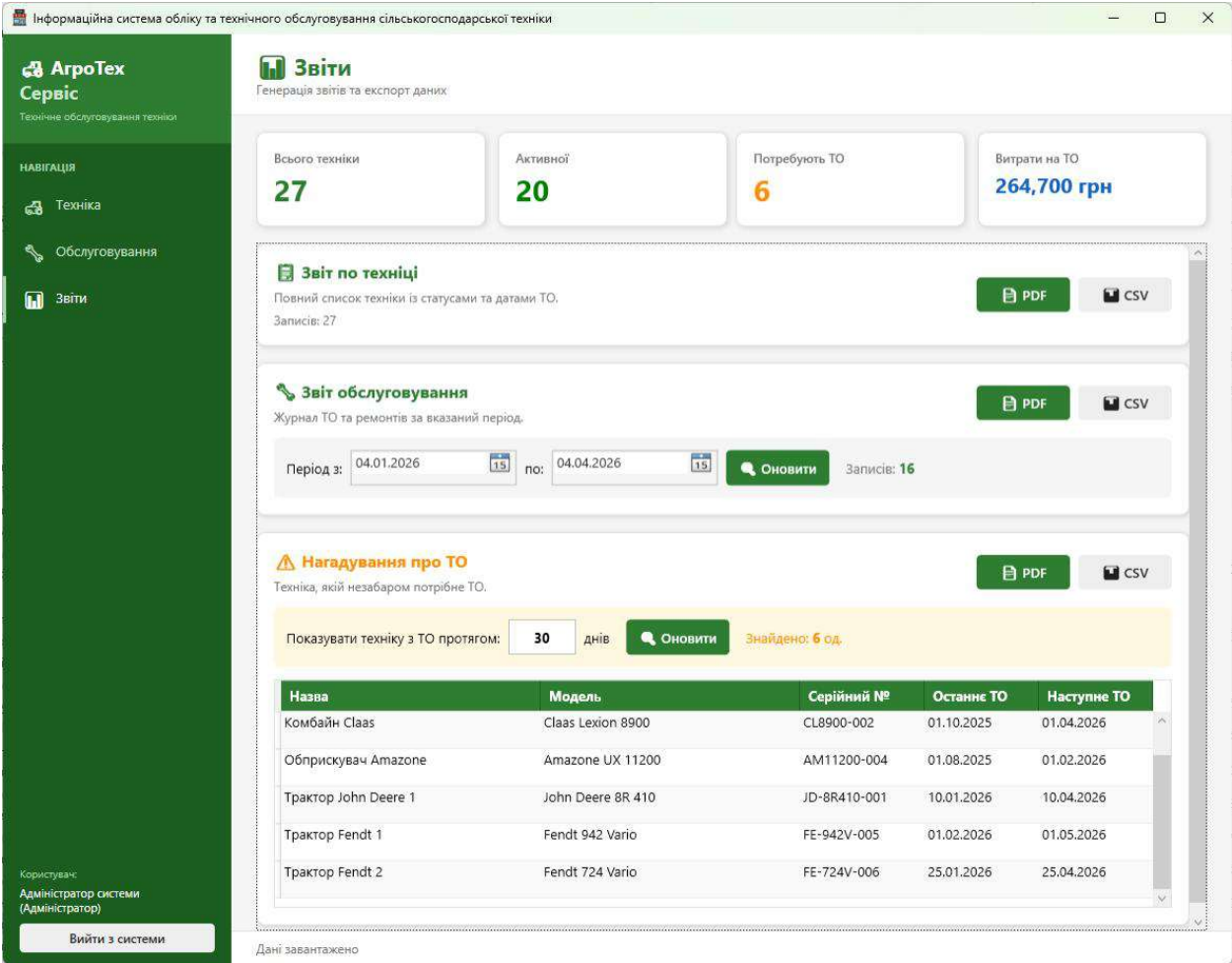


Рисунок 2.2.4. – Форма створення звітів

ReportsView є аналітичним розділом системи, що надає можливість формування та експорту трьох типів звітів. Форма організована як вертикально прокручуваний список з трьома картками — по одній на кожен тип звіту.

Верхній блок містить чотири підсумкові картки: загальна кількість техніки, кількість активної техніки, кількість машин що потребують ТО та загальна вартість обслуговування. Ці показники формуються при кожному завантаженні розділу незалежно від фільтрів.

Картка «Звіт по техніці» не потребує додаткових фільтрів і відразу надає кнопки експорту у форматах PDF та CSV для повного реєстру техніки. Картка «Звіт обслуговування» містить панель вибору діапазону дат та кнопку «Оновити», що перезапитує базу даних і оновлює лічильник ObsCount. Картка «Нагадування про ТО» дозволяє задати горизонт попередження (кількість днів) та відображає попередній перегляд у вбудованій таблиці до виконання експорту. Всі операції формування PDF та CSV виконуються асинхронно у фоновому потоці (Task.Run()), щоб не блокувати UI.

2.2.5 Діалог додавання / редагування техніки

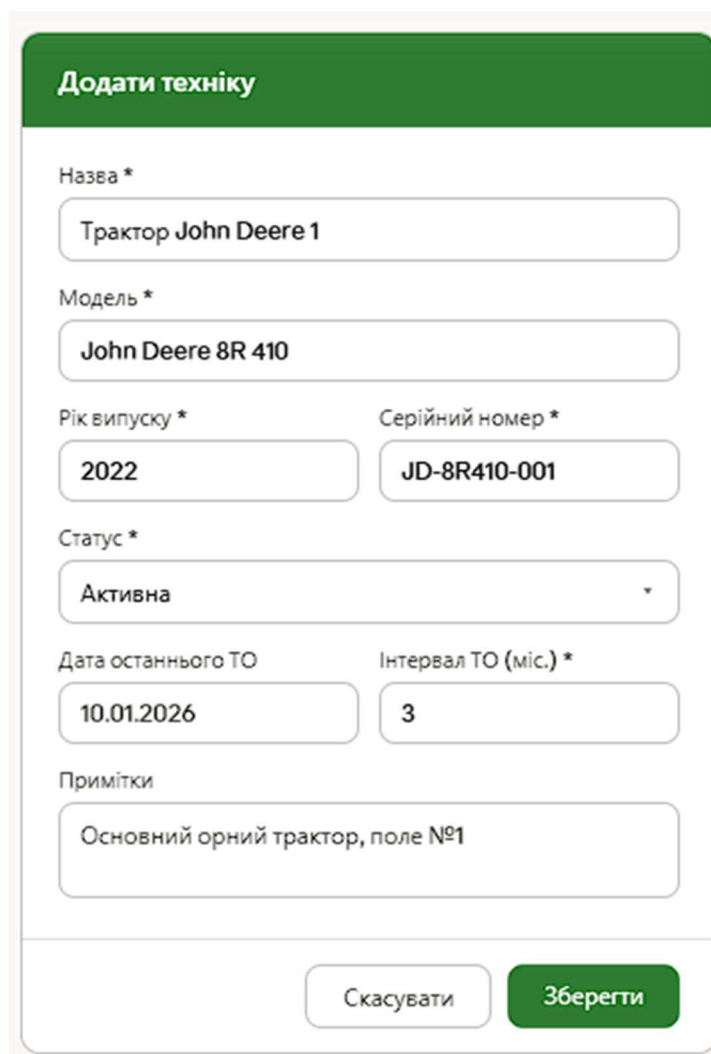


Рисунок 2.2.5. – Діалог додавання техніки, коректне заповнення

AddEditTechnikaDialog є модальним діалоговим вікном (тип Window, що відкривається через ShowDialog()), призначеним для введення або редагування повного набору характеристик одиниці техніки. Форма є

спільною для двох сценаріїв: при відкритті без параметра конструктора — режим додавання нового запису; при передачі об'єкта Technika — режим редагування з попередньо заповненими полями.

Рисунок 2.2.6. – Діалог з помилками валідації

Форма містить такі поля введення: «Назва» та «Модель» (TextBox), «Рік випуску» та «Серійний номер» (TextBox, розміщені в двоколонному Grid), «Статус» (ComboBox з трьома значеннями: «Активна», «В ремонті», «Списана»), «Дата останнього ТО» (DatePicker), «Інтервал ТО у місяцях» (TextBox), «Примітки» (багаторядковий TextBox).

Перед збереженням форма виконує клієнтську валідацію: перевіряє обов'язковість заповнення полів «Назва», «Модель», «Серійний номер»; коректність року (від 1900 до поточного); числовість та діапазон інтервалу ТО (від 1 до 120 місяців). При виявленні помилок блок з їх переліком стає видимим, а збереження блокується. Результат форми передається через

публічну властивість Result типу Technika; значення DialogResult = true встановлюється лише при успішній валідації та натисканні «Зберегти».

2.2.6 Діалог додавання / редагування запису ТО

Додати запис обслуговування

Техніка *

Трактор John Deere 1 (JD-8R410-001)

Дата ТО *

10.06.2026

Тип обслуговування *

Планове

Опис виконаних робіт *

Заміна моторного масла 5W-40, масляного та повітряного фільтрів.
Перевірка рівня охолоджувальної рідини. Діагностика електронних систем.

Вартість (грн)

8500.00

Виконавець

Петренко В.С.

* — обов'язкові поля

Скасувати

Зберегти

Рисунок 2.2.7. – Діалог додавання запису обслуговування

AddEditObslugovuvannyaDialog є модальним діалоговим вікном для введення або редагування запису журналу технічного обслуговування. Форма отримує через конструктор список усієї наявної техніки, що підставляється у випадаючий список для вибору.

Поле «Техніка» є обов'язковим і відображає список усіх одиниць техніки у форматі «Назва (Серійний номер)», що унеможлиблює вибір неіснуючої машини та забезпечує референтну цілісність. «Дата ТО» реалізована через DatePicker і заповнюється поточною датою при відкритті діалогу додавання. «Тип обслуговування» (ComboBox) надає три варіанти: «Планове», «Ремонт», «Позапланове». Поле «Опис виконаних робіт» є багаторядковим текстовим полем і є обов'язковим. Поля «Вартість» та «Виконавець» є необов'язковими; вартість при порожньому полі прирівнюється до нуля.

Важливою поведінковою особливістю є автоматичне оновлення дати останнього ТО в батьківській сутності «Техніка»: при додаванні запису репозиторій `ObslugovuvannyaRepository` перевіряє, чи є нова дата ТО пізнішою за вже збережену `DataOstannogTO`, і у такому разі оновлює її. Це забезпечує актуальність обчислюваної дати наступного ТО без жодного ручного втручання.

2.3 Архітектура програмного засобу

2.3.1 Загальна архітектурна концепція

Архітектура розроблюваної інформаційної системи побудована за принципом чіткого розмежування відповідальностей між компонентами на основі шаблону проєктування MVVM (Model-View-ViewModel). Застосунок організовано у чотири логічні шари: рівень представлення (Views), рівень логіки представлення (ViewModels), рівень бізнес-логіки та доступу до даних (Repositories, Services) та рівень моделі даних (Models). Взаємодія між шарами здійснюється виключно через чітко визначені інтерфейси та механізм впровадження залежностей (Dependency Injection).

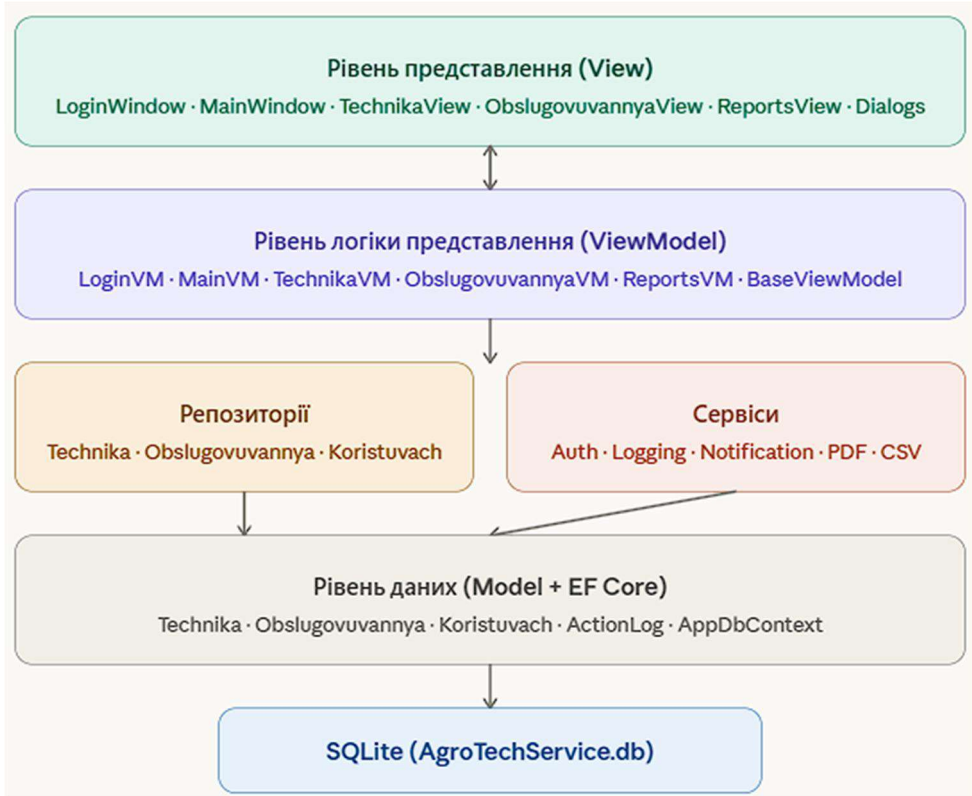


Рисунок 2.3.1. – Шарова архітектура інформаційної системи

2.3.2 Шаблон проектування MVVM

Шаблон MVVM є архітектурною основою усього застосунку і реалізує принцип розмежування відповідальностей (Separation of Concerns) між трьома складниками.

Model — доменні об'єкти (Technika, Obslugovuvannya, Koristuvach, ActionLog), що описують сутності предметної галузі. Моделі є «чистими» об'єктами даних (POCO) без залежностей від UI або бізнес-логіки. Обчислювані властивості (DataNastvnogTO, PotrebuyeTO) реалізовані безпосередньо в класах моделей та позначені атрибутом [NotMapped], що виключає їх зі схеми бази даних.

View — XAML-компоненти, що відповідають виключно за відображення даних і реакцію на дії користувача. View не містить жодної бізнес-логіки: обробники подій у code-behind обмежені передачею пароля з PasswordBox (специфічна обмеженість WPF) та відкриттям дочірніх вікон. Усі решта взаємодії здійснюються через механізм прив'язки даних XAML Binding до властивостей і команд ViewModel.

ViewModel — посередник між View та Model, що реалізує логіку представлення. Кожен ViewModel успадковує клас BaseViewModel, який забезпечує реалізацію інтерфейсів INotifyPropertyChanged (для сповіщення View про зміни властивостей) та INotifyDataErrorInfo (для валідації форм). Команди реалізовані через класи RelayCommand і AsyncRelayCommand, що дозволяє прив'язувати асинхронні операції до кнопок без блокування UI-потoku.

Ключовою перевагою MVVM у контексті WPF є те, що View та ViewModel не мають прямих посилань одне на одного — їх зв'язок встановлюється виключно через DataContext та систему прив'язки. Це спрощує тестування ViewModel без запуску UI.

2.3.3 Рівень доступу до даних: шаблон Repository

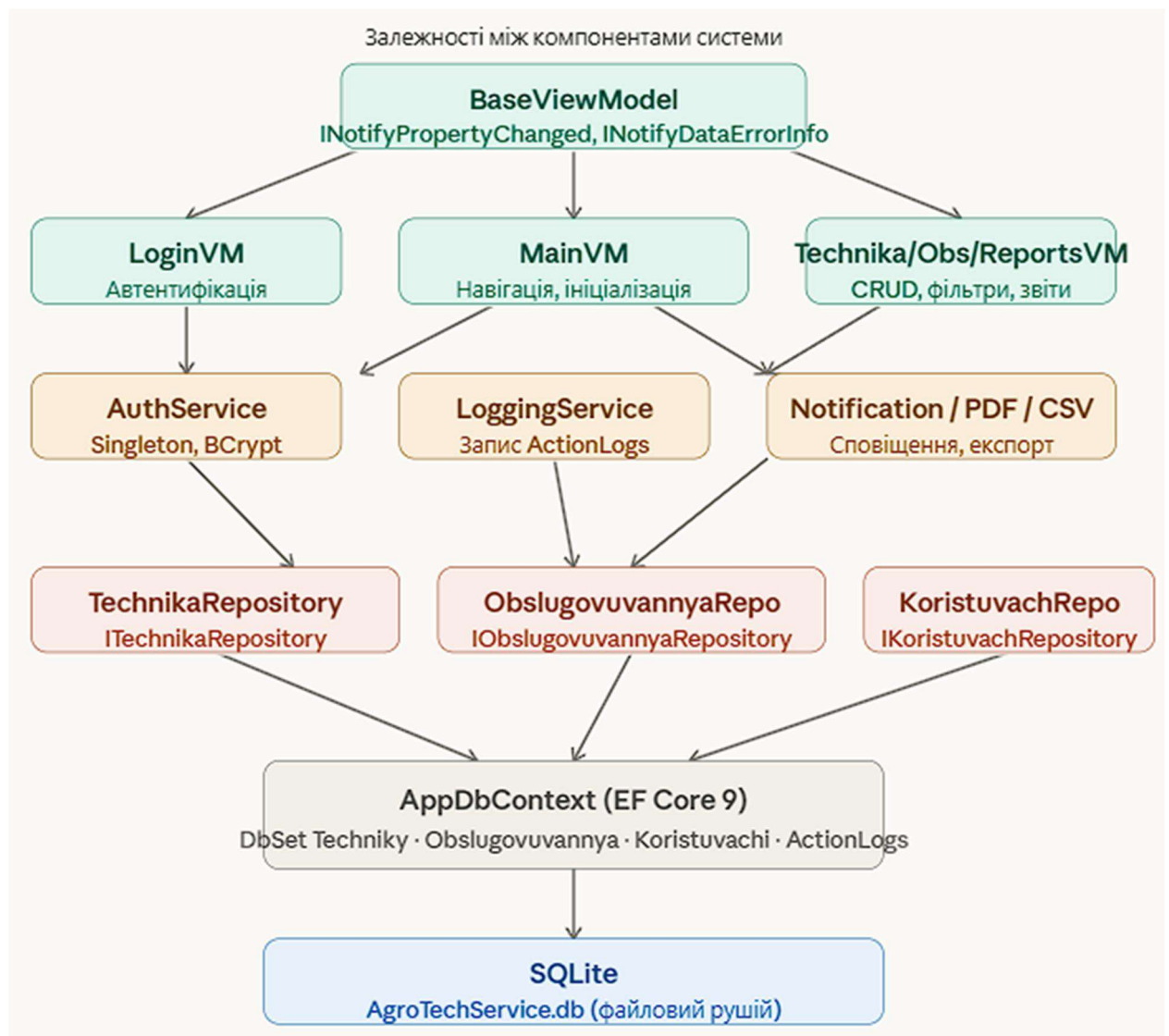


Рисунок 2.3.2. – Структура компонентів та їх залежності

Доступ до бази даних організовано за шаблоном «Репозиторій» (Repository Pattern), що інкапсулює логіку запитів до БД і надає ViewModel абстрактний інтерфейс для роботи з даними незалежно від СУБД. Кожен репозиторій реалізує відповідний інтерфейс:

- ITechnikaRepository / TechnikaRepository — CRUD-операції, пошук за критеріями, отримання техніки з наближеним строком ТО;
- IObslugovuvannyaRepository / ObslugovuvannyaRepository — CRUD-операції, фільтрація за технікою/типом/датами, підрахунок загальної вартості; після додавання запису автоматично оновлює DataOstannogTO батьківської техніки;

- `IKoristuvachRepository` / `KoristuvachRepository` — пошук за логіном, CRUD.

Усі методи репозиторіїв є асинхронними (`async/await`), що забезпечує неблокуючий доступ до БД з UI-потoku. Взаємодія з SQLite відбувається через `AppDbContext` (Entity Framework Core 9), що надає LINQ-інтерфейс для формування запитів.

2.3.4 Рівень сервісів

Бізнес-логіка, що виходить за межі простих операцій CRUD, інкапсульована в окремих сервісних класах:

`AuthService` реєструється як **Singleton** у контейнері DI, оскільки зберігає стан поточного сеансу (об'єкт `PorochnyjKoristuvach`). Для доступу до `IKoristuvachRepository` (Scoped) використовує `IServiceProvider` та щоразу створює новий scope, уникаючи антипатерну «Captive Dependency».

`LoggingService` фіксує кожен значущу дію у таблиці `ActionLogs`. Метод `LogAsync` виконується у блоці `try/catch` без перебросу виняткових ситуацій — збій журналювання не повинен переривати основну операцію користувача.

`NotificationService` при запуску застосунку запитує список техніки з наближенням ТО через `GetPotrebuuyutTOAsync(30)` та відображає `MessageBox` із переліком машин, що потребують уваги.

`PdfReportService` використовує бібліотеку `QuestPDF` з `Fluent API` для програмного опису структури документа. Генерація PDF виконується у фоновому потоці (`Task.Run()`) для уникнення блокування UI.

`CsvExportService` використовує бібліотеку `CsvHelper` з роздільником «;» та кодуванням UTF-8 BOM для коректного відображення кирилиці у `Microsoft Excel`.

2.3.5 Механізм впровадження залежностей

Управління залежностями між компонентами реалізовано через контейнер `Microsoft.Extensions.DependencyInjection`. Конфігурація контейнера зосереджена у методі `OnStartup` класу `App`. Компоненти реєструються з різними часами життя залежно від їх природи:

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.3.1 — Реєстрація компонентів у DI-контейнері

Компонент	Час життя	Обґрунтування
AppDbContext	Scoped	Окремий контекст на область видимості; EF Core не є потоково-безпечним
ITechnikaRepository	Scoped	Залежить від AppDbContext; має той самий час життя
IObslugovuvannyaRepository	Scoped	Аналогічно
IKoristuvachRepository	Scoped	Аналогічно
AuthService	Singleton	Зберігає стан поточного сеансу між вікнами
LoggingService	Transient	Не зберігає стану; новий екземпляр на кожне звернення
NotificationService	Transient	Аналогічно
PdfReportService	Transient	Stateless-сервіс генерації
CsvExportService	Transient	Stateless-сервіс генерації
LoginViewModel	Transient	Новий екземпляр при кожному відкритті вікна
MainViewModel	Transient	Аналогічно
TechnikaViewModel	Transient	Аналогічно
ObslugovuvannyaViewModel	Transient	Аналогічно
ReportsViewModel	Transient	Аналогічно
LoginWindow	Transient	Форми є Transient — кожне відкриття створює новий екземпляр
MainWindow	Transient	Аналогічно

Статичний доступ до контейнера надається через властивість `App.ServiceProvider`, що дозволяє розв'язувати залежності у точках, де конструкторна ін'єкція неможлива (наприклад, при відкритті `MainWindow` з обробника події `LoginSuccess`).

2.3.6 Потік даних при типовій операції

Послідовність взаємодії при додаванні запису обслуговування охоплює сім кроків. По-перше, користувач натискає кнопку «Додати» у `ObslugovuvannyaView`, що через механізм `ICommand` викликає метод `AddAsync()` у `ObslugovuvannyaViewModel`. По-друге, `ViewModel` відкриває `AddEditObslugovuvannyaDialog` з переліком техніки. По-третє, після заповнення форми та натискання «Зберегти» `ViewModel` отримує заповнений об'єкт `Obslugovuvannya` через властивість `Dialog.Result`. По-четверте, `ViewModel` викликає `AddAsync(obs)` на репозиторії. По-п'яте, `ObslugovuvannyaRepository` виконує `INSERT` нового запису та оновлює `DataOstannogTO` батьківської техніки в межах одного контексту EF Core. По-

шосте, виклик `SaveChangesAsync()` атомарно зберігає обидві зміни в SQLite. По-сьоме, `ViewModel` оновлює `ObservableCollection`, викликає `LoggingService.LogAsync()` та оновлює статистику. `View` реагує на зміни колекції та властивостей автоматично через `Data Binding`.

Рис. 2.3.3 ілюструє послідовність взаємодії компонентів при виконанні операції додавання нового запису обслуговування.

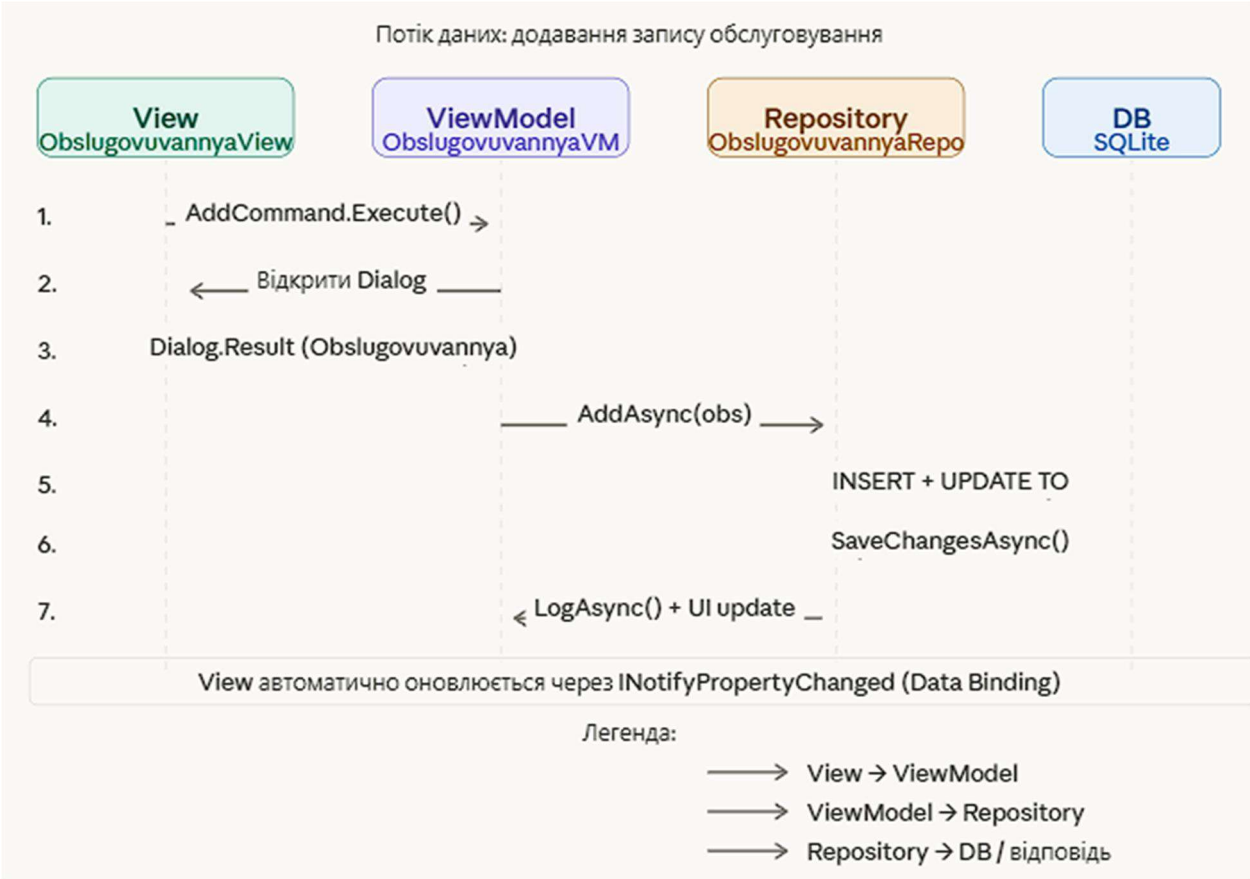


Рисунок 2.3.3. – Потік даних при додаванні запису обслуговування

2.3.7 Структура файлів проєкту

Таблиця 2.3.2 — Структура папок проєкту та їх призначення

Папка	Вміст	Призначення
Models/	Technika.cs, Obslugovuvannya.cs, Koristuvach.cs, ActionLog.cs	Доменні моделі, enum-типи
Data/	AppDbContext.cs, DatabaseInitializer.cs	EF Core контекст, ініціалізація БД
Repositories/Interfaces/	ITechnikaRepository.cs та ін.	Контракти репозиторіїв
Repositories/	TechnikaRepository.cs та ін.	Реалізації репозиторіїв
Services/	AuthService.cs, LoggingService.cs та ін.	Бізнес-сервіси
Helpers/	RelayCommand.cs, Converters.cs, PasswordHasher.cs	Допоміжні класи та конвертери

ViewModels/	BaseViewModel.cs, LoginViewModel.cs та ін.	ViewModel-класи
Views/	LoginWindow.xaml, MainWindow.xaml та ін.	XAML-форми
Views/Dialogs/	AddEditTechnikaDialog.xaml та ін.	Модальні діалоги
Resources/	Styles.xaml	Глобальні стилі та ресурси

Висновки до розділу 2

У другому розділі виконано повне проектування програмного засобу, що охоплює модель бази даних, перелік форм та архітектуру системи.

Розроблено фізичну модель бази даних SQLite, що складається з чотирьох таблиць (33 поля загалом), двох зовнішніх ключів та двох унікальних індексів. Модель приведено до третьої нормальної форми; єдина свідома денормалізація (KoristuvachLogin в ActionLogs) обґрунтована вимогою довгострокової цілісності аудитного журналу.

Описано сім форм графічного інтерфейсу: два вікна (LoginWindow, MainWindow), три UserControl-компоненти (TechnikaView, ObslugovuvannyaView, ReportsView) та два модальних діалоги. Для кожної форми визначено функціональне призначення, склад елементів керування та логіку взаємодії з ViewModel.

Архітектура системи реалізована за шаблоном MVVM з чотирма логічними шарами: представлення, логіка представлення, репозиторії та сервіси, дані. Управління залежностями між 16 компонентами здійснюється через контейнер DI з обґрунтованим вибором часу життя (Singleton, Scoped, Transient). Описано потік даних при типовій операції CRUD, що підтверджує коректність реалізації шаблону MVVM та узгодженість компонентів.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

3.1 Мінімальні вимоги до системи

Для коректного встановлення та функціонування інформаційної системи обліку та технічного обслуговування сільськогосподарської техніки комп'ютер користувача має відповідати наведеним нижче технічним вимогам.

Таблиця 3.1.1 — Мінімальні апаратні вимоги

Компонент	Мінімальна вимога	Рекомендована конфігурація
Процесор	Intel Core i3 / AMD Ryzen 3, 1.5 ГГц, 2 ядра	Intel Core i5 / AMD Ryzen 5, 2.5 ГГц, 4 ядра
Оперативна пам'ять	4 ГБ RAM	8 ГБ RAM
Місце на диску	200 МБ (застосунок + БД)	500 МБ (з урахуванням зростання БД)
Роздільна здатність екрана	1280 × 720 пікселів	1920 × 1080 пікселів
Підключення до мережі	Не потрібне	—

Таблиця 3.1.2 — Програмні вимоги

Компонент	Вимога
Операційна система	Windows 10 (версія 1903 або новіша) / Windows 11
Розрядність ОС	64-розрядна (x64)
.NET Runtime	.NET 9 Desktop Runtime для Windows (x64)
Microsoft Visual C++ Redistributable	2015–2022 (x64) — входить до .NET 9 Runtime
Права користувача	Стандартні (без прав адміністратора ОС)
Вільне місце у папці профілю	Мінімум 50 МБ (для файлу бази даних SQLite)

Система не потребує встановлення жодного стороннього програмного забезпечення баз даних (СУБД-сервера), оскільки використовує вбудований рушій SQLite. Файл бази даних AgroTechService.db автоматично створюється у робочій папці застосунку при першому запуску.

Таблиця 3.1.3 — Залежності, що постачаються разом із застосунком

Бібліотека	Версія	Призначення
Entity Framework Core	9.0.0	ORM для доступу до SQLite
Microsoft.Data.Sqlite	9.0.0	SQLite-провайдер для EF Core
QuestPDF	2024.3.4	Генерація PDF-звітів
CsvHelper	33.0.1	Експорт даних у форматі CSV
BCrypt.Net-Next	4.0.3	Хешування паролів

Усі зазначені бібліотеки постачаються разом з дистрибутивом застосунку і не потребують окремого встановлення.

Порядок встановлення:

1. Завантажити та встановити **Microsoft .NET 9 Desktop Runtime** з офіційного сайту Microsoft

(<https://dotnet.microsoft.com/download/dotnet/9.0>), обравши варіант «Desktop Runtime» для Windows x64.

2. Розпакувати архів дистрибутиву застосунку AgroTechService.zip у будь-яку зручну папку на диску комп'ютера (наприклад, C:\AgroTechService\).
3. Запустити файл AgroTechService.exe. При першому запуску система автоматично ініціалізує базу даних та створить облікові записи за замовчуванням.

Облікові записи за замовчуванням:

Роль	Логін	Пароль
Адміністратор	admin	admin123
Оператор	operator	operator123

Після першого входу рекомендується змінити паролі облікових записів за замовчуванням.

3.2 Інструкція для роботи з програмою

3.2.1 Запуск системи та автентифікація

Рисунок 3.2.1. – Вхід до системи

При запуску застосунку відкривається вікно автентифікації. Після введення коректних облікових даних та натискання кнопки «Увійти» система перевіряє пароль із використанням алгоритму BCrypt і, у разі успіху, відкриває головне вікно. Одночасно виконується автоматична перевірка строків

технічного обслуговування: якщо є техніка, ТО якої настає протягом наступних 30 днів або вже прострочене, відображається відповідне сповіщення.

При введенні невірного логіну або пароля під полем пароля з'являється повідомлення «Невірний логін або пароль». Кількість спроб входу не обмежена.

3.2.2 Навігація у головному вікні

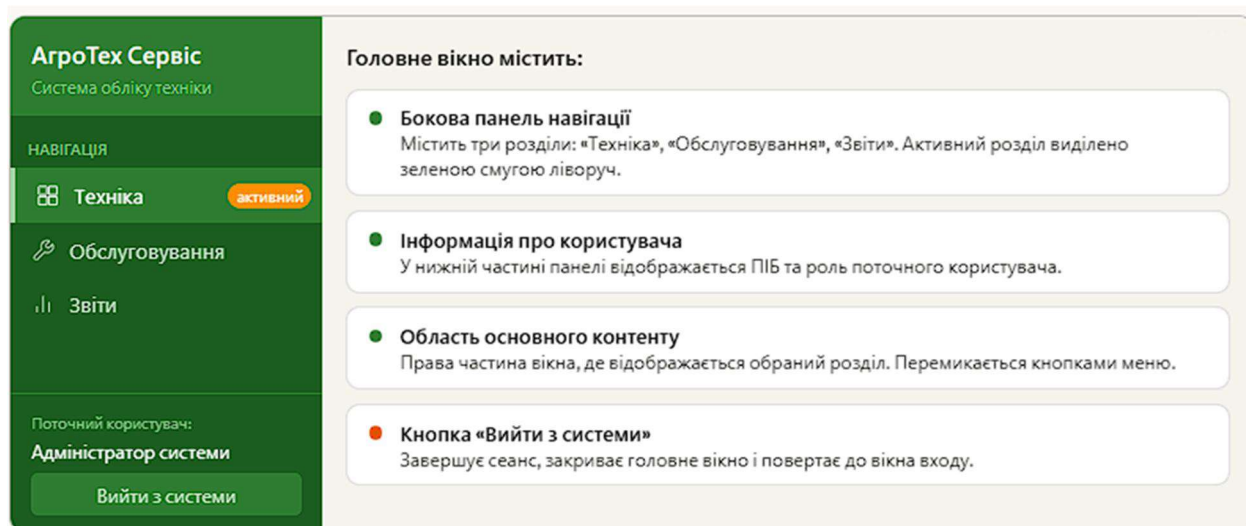


Рисунок 3.2.2. – Головне вікно з описом елементів навігації

Після успішного входу відкривається головне вікно системи, що складається з двох зон: бокової панелі навігації (ліворуч) та області основного контенту (праворуч). Перехід між розділами здійснюється натисканням відповідних кнопок на панелі навігації. При кожному переході виконується автоматичне завантаження актуальних даних обраного розділу. Для завершення сеансу роботи необхідно натиснути кнопку «Вийти з системи» у нижній частині панелі навігації.

3.2.3 Робота з реєстром техніки

Техніка
Облік сільськогосподарської техніки

+ Додати Редагувати Видалити

Всього **22** Активних **18** В ремонті **2** Потребують ТО **5**

Пошук за назвою/моделлю: John Deere Статус: Всі статуси Дата ТО: 01.01.2026 Пошук Очистити

Назва	Модель	Рік	Статус	Останнє ТО	Наступне ТО
Трактор John Deere 1	JD 8R 410	2022	Активна	10.01.2026	10.04.2026
Трактор John Deere 2	JD 8R 370	2021	Активна	15.02.2026	15.05.2026

Рядок з жовтим фоном — техніка потребує ТО найближчим часом

Рисунок 3.2.3. – Головне вікно з описом елементів навігації

Розділ «Техніка» є основним робочим розділом для управління реєстром машин. При вході до розділу автоматично завантажується повний перелік техніки та відображаються зведені показники у картках статистики.

Додавання нової техніки. Натисніть кнопку «+ Додати» у правій частині заголовка. У вікні, що відкриється, заповніть обов'язкові поля (позначені символом «*»): «Назва», «Модель», «Рік випуску», «Серійний номер», «Статус» та «Інтервал ТО». Поле «Дата останнього ТО» є необов'язковим — якщо техніка щойно придбана і ТО ще не проводилося, залиште його порожнім. Натисніть «Зберегти». Якщо введені дані містять помилки (наприклад, некоректний рік або дублюючий серійний номер), система відобразить перелік помилок і заблокує збереження до їх виправлення.

Редагування техніки. Оберіть рядок у таблиці одним натисканням лівої кнопки миші — рядок виділиться зеленим кольором. Натисніть кнопку «Редагувати». У вікні редагування всі поля будуть попередньо заповнені поточними даними. Внесіть потрібні зміни та натисніть «Зберегти».

Видалення техніки. Оберіть рядок у таблиці та натисніть кнопку «Видалити». Система відобразить діалог підтвердження з попередженням про

те, що разом з технікою будуть видалені всі пов'язані записи обслуговування. Натисніть «Так» для підтвердження або «Ні» для скасування.

Пошук і фільтрація. Введіть частину назви або моделі у поле пошуку, оберіть статус зі списку та/або вкажіть дату ТО. Натисніть «Пошук». Для повернення до повного переліку натисніть «Очистити».

Кольорове кодування таблиці:

Виділення	Значення
Жовтуватий фон рядка	Техніка потребує ТО протягом наступних 30 днів або ТО вже прострочене
Зелений бейдж статусу	Техніка активна і в експлуатації
Помаранчевий бейдж	Техніка перебуває на ремонті
Сірий бейдж	Техніку списано
Червона дата «Наступне ТО»	Строк обслуговування минув або настає найближчим часом

3.2.4 Ведення журналу технічного обслуговування

Додати запис обслуговування

Техніка *
Трактор John Deere 1 (JD-8R410-001)

Дата ТО *
10.06.2026

Тип *
Планове

Опис виконаних робіт *
Заміна моторного масла, масляного та повітряного фільтрів. Перевірка гальмівної системи.

Вартість (грн)
8500.00

Виконавець
Петренко В.С.

Скасувати

Зберегти

Порядок додавання запису:

- Оберіть техніку зі списку — відображається назва та серійний номер
- Вкажіть дату проведення ТО через календар
- Оберіть тип: Планове Ремонт Позапланове
- Введіть детальний опис виконаних робіт
- Вкажіть вартість (необов'язково) та виконавця
- Натисніть «Зберегти». Дата останнього ТО техніки оновиться автоматично

Автоматичне оновлення

Після збереження запису дата «Останнє ТО» відповідної техніки оновлюється автоматично, якщо нова дата є пізнішою за попередню.

Рисунок 3.2.4. – Діалог додавання запису обслуговування

Для переходу до журналу обслуговування натисніть «Обслуговування» на панелі навігації. Розділ відображає всі записи ТО та ремонтів у хронологічному порядку (найновіші — першими).

Додавання запису обслуговування. Натисніть «+ Додати». У діалоговому вікні послідовно заповніть поля: оберіть техніку зі списку, вкажіть дату проведення ТО через вбудований календар (за замовчуванням

підставляється поточна дата), оберіть тип обслуговування («Планове», «Ремонт» або «Позапланове»), введіть опис виконаних робіт. Поля «Вартість» та «Виконавець» є необов'язковими. Натисніть «Зберегти». Дата «Останнє ТО» відповідної техніки в реєстрі оновиться автоматично.

Фільтрація записів. У панелі фільтрів оберіть конкретну техніку зі списку (або залиште «— Вся техніка —» для перегляду всіх записів), оберіть тип обслуговування та вкажіть діапазон дат «від» і «до». Натисніть «Пошук». Загальна вартість у картці статистики перераховується автоматично відповідно до застосованого фільтра.

3.2.5 Формування звітів та експорт даних

Рисунок 3.2.5. – Розділ «Звіти»

Розділ «Звіти» надає можливість формування трьох типів аналітичних документів у форматах PDF та CSV.

Звіт по техніці. Містить повний реєстр усіх одиниць техніки з колонками: назва, модель, рік, серійний номер, статус, дата останнього та наступного ТО. Для формування натисніть «PDF» або «CSV» у відповідній картці. Збереження відбувається у діалозі вибору шляху — за замовчуванням пропонується Робочий стіл. Після збереження система запропонує відкрити файл.

Звіт обслуговування. Формується за вказаний діапазон дат. Перед експортом:

1. Вкажіть початкову та кінцеву дати у відповідних полях;
2. Натисніть «Оновити» — лічильник записів відображатиме кількість знайдених записів за обраний період;
3. Натисніть «PDF» або «CSV» для збереження звіту.

PDF-звіт обслуговування містить підсумковий рядок із загальною вартістю всіх робіт за обраний період.

Нагадування про ТО. Формується для техніки, строк обслуговування якої настане протягом вказаної кількості днів:

1. У полі «Показувати протягом» введіть потрібну кількість днів (наприклад, 30);
2. Натисніть «Оновити» — у таблиці попереднього перегляду відобразиться перелік відповідної техніки;
3. Натисніть «PDF» або «CSV» для збереження звіту.

Таблиця 3.2.1 — Формати та вміст звітів

Тип звіту	PDF	CSV	Вміст
Реєстр техніки	Таблиця A4 landscape, дата формування	Рядки з усіма полями, роздільник «;»	22 колонки (дата наст. ТО включно)
Журнал ТО	Таблиця A4 landscape, підсумок вартості	Рядки з описом та виконавцем	Записи за обраний період
Нагадування ТО	Картки A4 portrait, кожна — одна машина	Таблиця аналогічна реєстру	Лише техніка з наближеним ТО

3.2.6 Типові помилки та способи їх усунення

Таблиця 3.2.2 — Типові ситуації та рекомендовані дії

Ситуація	Причина	Рекомендована дія
При запуску з'являється помилка «.NET Runtime not found»	Не встановлено .NET 9 Desktop Runtime	Встановіть .NET 9 Desktop Runtime з сайту Microsoft
Кнопка «Зберегти» недоступна у діалозі	Не заповнено обов'язкові поля або є помилки валідації	Перегляньте перелік помилок у нижній частині діалогу та виправте їх
При додаванні техніки: «Серійний номер вже існує»	Інша одиниця техніки вже має такий серійний номер	Перевірте реєстр і введіть коректний унікальний номер
Кнопки «Редагувати» та «Видалити» неактивні	Жоден рядок у таблиці не вибраний	Клацніть лівою кнопкою миші на потрібному рядку

Після збереження запису ТО дата не оновилася	Нова дата ТО є раннішою за вже збережену	Система зберігає лише пізнішу дату; поточне значення правильне
Звіт PDF відкривається з помилкою	Стороння програма блокує файл	Закрийте файл PDF та повторіть збереження з іншим ім'ям
Файл CSV відображається некоректно у Excel	Неправильне кодування або роздільник	Відкрийте Excel → «Дані» → «З тексту/CSV», оберіть кодування UTF-8, роздільник «;»

Висновки до розділу 3

У третьому розділі сформульовано вимоги до середовища функціонування системи та розроблено повне керівництво користувача.

Визначено мінімальні апаратні та програмні вимоги: операційна система Windows 10/11 (64-розрядна), .NET 9 Desktop Runtime, не менше 4 ГБ оперативної пам'яті та 200 МБ вільного місця на диску. Система не потребує встановлення СУБД-сервера, окремих мережових з'єднань або прав адміністратора операційної системи, що суттєво спрощує її розгортання на підприємстві.

Наведено покрокові інструкції для всіх функціональних можливостей системи: автентифікації, навігації головним вікном, управління реєстром техніки (додавання, редагування, видалення, пошук), ведення журналу технічного обслуговування, формування та збереження звітів трьох типів у форматах PDF і CSV. Усі інструкції проілюстровано макетами екранів системи (рис. 3.2.1 — 3.2.5).

Наведено таблицю типових помилок та рекомендованих дій для їх усунення, що дозволяє користувачеві самостійно вирішувати поширені ситуації без залучення технічної підтримки.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-практичне завдання — розроблено інформаційну систему обліку та технічного обслуговування сільськогосподарської техніки для аграрних підприємств України.

У першому розділі проаналізовано предметну галузь та встановлено, що відсутність спеціалізованого програмного забезпечення призводить до пропуску планових ТО, дублювання записів і неможливості оперативного аналізу стану машинно-тракторного парку. Проведений огляд існуючих рішень — платформ eMaint, Fiix, АгроПлан та засобів загального призначення — показав, що жодне з них не задовольняє одночасно вимогам доступності, локального розгортання, підтримки державної мови та повноти функціоналу управління ТО. На підставі аналізу сформульовано сім груп функціональних вимог, які стали основою технічного завдання на розроблення системи. Обґрунтовано вибір технологічного стеку: платформа .NET 9, мова C#, технологія WPF з архітектурним шаблоном MVVM, СУБД SQLite та ORM Entity Framework Core 9.

У другому розділі спроектовано реляційну модель бази даних у складі чотирьох таблиць (33 поля, 2 зовнішніх ключі, 2 унікальних індекси), приведену до третьої нормальної форми. Для кожної таблиці розроблено фізичну модель із точними типами даних SQLite та DDL-визначеннями, підготовлено DBML-код для відтворення ER-діаграми на сервісі dbdiagram.io. Описано сім форм графічного інтерфейсу та обґрунтовано їх компонування відповідно до принципів зручності використання. Архітектуру системи побудовано за шаблоном MVVM з чотирма логічними шарами та управлінням залежностями через DI-контейнер з обґрунтованим розподілом часів життя компонентів.

У третьому розділі сформульовано мінімальні апаратні та програмні вимоги до системи та розроблено покрокове керівництво користувача з ілюстраціями. Реалізовано повний CRUD-функціонал для управління

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

реєстром техніки та журналом технічного обслуговування з асинхронним доступом до БД. Впроваджено автоматичний контроль строків ТО з відображенням сповіщення при запуску системи та кольоровим виділенням проблемних записів у таблиці. Реалізовано підсистему формування звітів у форматах PDF (з використанням бібліотеки QuestPDF) та CSV (з використанням CsvHelper) із підтримкою трьох типів звітів. Забезпечено безпеку автентифікації через хешування паролів алгоритмом BCrypt та розмежування прав доступу між ролями адміністратора й оператора. Реалізовано журналювання дій користувачів у таблиці ActionLogs для забезпечення аудитного сліду. Застосування шаблону Repository забезпечує незалежність бізнес-логіки від конкретної СУБД і спрощує потенційне перенесення системи на інший рушій баз даних. Система не потребує встановлення СУБД-сервера, мережевого підключення або прав адміністратора ОС, що суттєво спрощує її розгортання в умовах аграрного підприємства.

Розроблений програмний засіб у повному обсязі відповідає сформульованим функціональним вимогам і може бути впроваджений на аграрних підприємствах для підвищення ефективності управління машинно-тракторним парком та своєчасного проведення технічного обслуговування. Подальший розвиток системи може передбачати реалізацію багатокористувацького доступу, хмарного резервного копіювання бази даних, інтеграцію з телематичними системами машин та мобільного клієнта для операторів у полі.

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Берус В. В., Кухаренко О. М. Інформаційні системи в агропромисловому комплексі : навч. посіб. Київ : Аграрна наука, 2021. 284 с.
2. Буч Г., Рамбо Дж., Якобсон А. Мова моделювання UML. Керівництво користувача / пер. з англ. Харків : Вільямс, 2020. 496 с.
3. Мартін Р. К. Чистий код: створення, аналіз і рефакторинг / пер. з англ. Харків : Вільямс, 2019. 464 с.
4. Поляков Д. В. Проектування баз даних : навч. посіб. Харків : ХНАДУ, 2022. 312 с.
5. Степаненко О. С., Левченко Ю. М. Управління технічним обслуговуванням сільськогосподарської техніки : монографія. Київ : ННЦ «ІАЕ», 2020. 198 с.
6. Фаулер М. Архітектура корпоративних програмних застосунків / пер. з англ. Харків : Вільямс, 2021. 560 с.
7. Хоровський О. Б., Яремчук С. С. Системи підтримки прийняття рішень в агровиробництві. Вісник ЖНАЕУ. 2022. № 2. С. 112–121.
8. Шевченко Д. С. Технічна надійність машинно-тракторного парку : навч. посіб. Харків : ХНТУСГ, 2021. 240 с.
9. Microsoft. .NET 9 documentation. URL: <https://learn.microsoft.com/en-us/dotnet/> (дата перегляду: 15.01.2026).
10. Microsoft. Windows Presentation Foundation (WPF) overview. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/> (дата перегляду: 18.01.2026).
11. Microsoft. Entity Framework Core documentation. URL: <https://learn.microsoft.com/en-us/ef/core/> (дата перегляду: 22.01.2026).
12. SQLite Consortium. SQLite documentation. URL: <https://www.sqlite.org/docs.html> (дата перегляду: 25.01.2026).
13. QuestPDF. QuestPDF documentation. URL: <https://www.questpdf.com/documentation.html> (дата перегляду: 03.02.2026).

					КРФМБ.122.043стн.020.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

14. CsvHelper. CsvHelper library documentation. URL:
<https://joshclose.github.io/CsvHelper/> (дата перегляду: 07.02.2026).
15. Provos N., Mazieres D. A future-adaptable password scheme. Proceedings of the USENIX Annual Technical Conference. 1999. P. 81–92. URL:
<https://www.usenix.org/legacy/events/usenix99/provos/provos.pdf> (дата перегляду: 10.02.2026).
16. Microsoft. Dependency injection in .NET. URL:
<https://learn.microsoft.com/en-us/dotnet/core/extensions/dependency-injection> (дата перегляду: 14.02.2026).
17. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley, 2002. 533 p.
18. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.
19. BCrypt.Net-Next. BCrypt.Net-Next NuGet package. URL:
<https://www.nuget.org/packages/BCrypt.Net-Next> (дата перегляду: 20.02.2026).
20. ДСТУ 4063:2001. Технічне обслуговування і ремонт техніки. Терміни та визначення. Київ : Держстандарт України, 2001. 28 с.
21. Yalcin M., Ustun V. Development of CMMS for agricultural machinery maintenance management. Computers and Electronics in Agriculture. 2021. Vol. 185. P. 1–12. DOI: <https://doi.org/10.1016/j.compag.2021.106162> (дата перегляду: 05.03.2026).
22. Smith J. Entity Framework Core in Action. 2nd ed. Shelter Island : Manning Publications, 2021. 600 p.
23. Staatsen B. WPF MVVM in Depth. Leanpub, 2022. 310 p. URL:
<https://leanpub.com/wpfmvvm> (дата перегляду: 20.03.2026).
24. Troelsen A., Japikse P. Pro C# 10 with .NET 6. 10th ed. New York : Apress, 2022. 1320 p.