

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 автоматизації бізнес-
процесів компанії з оренди автомобілів»**

Виконав здобувач освіти групи П-42
Ракович Дмитро Володимирович

Керівник роботи:
Устименко Людмила Миколаївна

Рецензент:

Зміст

Вступ	6
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	9
1.1 Основні задачі	9
1.2 Огляд існуючих рішень	11
1.3 Функціональні вимоги до програми	14
1.4 Вибір технології	17
1.5 Постановка задачі	19
Висновки до розділу 1	20
Розділ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ	22
2.1 Модель бази даних	22
2.2 Список основних форм та їх призначення	29
2.3 Архітектура програмного засобу	34
2.4 Програмна реалізація	37
2.5 Тестування програми	41
Висновки до розділу 2	43
Розділ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ	45
3.1 Мінімальні вимоги до системи	45
3.2 Інструкція для роботи з програмою	46
Висновки до розділу 3	48
ВИСНОВКИ	50
Перелік літературних джерел	52
Додаток А. Код для роботи з базою даних	55
Додаток Б. Програмний код модулів	56

					КРФМБ.122.042.033.2026.ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Ракович Д.В.			Інформаційна система автоматизації бізнес-процесів компанії з оренди автомобілів	Літ.	Арк.
Перевірів		Устименко Л.М.					3
Рецензент		.				ЖАТФК	
Н. Контр.							
Затвердив.		Гнатюк О.Ф.					
						Аркуші	60

РЕФЕРАТ

Пояснювальна записка: 60 с., 9 рис., 4 табл., 2 додатки, 24 джерела.

Об'єкт дослідження – бізнес-процеси підприємства, що надає послуги з прокату автомобілів.

Мета роботи – розробка надійної локальної інформаційної системи для автоматизації управління автопарком, обліку клієнтської бази та оформлення орендних угод.

Методи дослідження: системний та об'єктно-орієнтований аналіз, архітектурне моделювання програмного забезпечення, проектування реляційних баз даних.

У кваліфікаційній роботі проаналізовано специфіку автопрокатного бізнесу та обґрунтовано необхідність відмови від ручних методів обліку. Спроектовано логічну та фізичну структуру реляційної бази даних. Програмно реалізовано десктопний застосунок на платформі .NET 9 мовою C# із використанням фреймворку WPF та архітектурного шаблону MVVM. У ролі сховища транзакційних даних застосовано вбудовану СУБД SQLite у зв'язці з Entity Framework Core.

Розроблена інформаційна система забезпечує ведення каталогів транспортних засобів (із механізмом локального збереження зображень) та реєстру клієнтів, повністю автоматизує процес розрахунку фінансової вартості оренди, здійснює перевірку доступності авто і генерує звітну документацію у форматі PDF. Практичне значення полягає у тому, що створений програмний продукт є функціонально завершеним та готовим до впровадження для підвищення операційної ефективності реальної компанії.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, ОRENDA АВТОМОБІЛІВ, АВТОМАТИЗАЦІЯ, БАЗА ДАНИХ, C#, WPF, SQLITE, MVVM.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ACID (Atomicity, Consistency, Isolation, Durability) — атомарність, узгодженість, ізольованість, довговічність (вимоги до транзакційної системи)

API (Application Programming Interface) — інтерфейс програмування застосунків

CRUD (Create, Read, Update, Delete) — створення, читання, оновлення, видалення (базові функції управління даними)

DI (Dependency Injection) — впровадження залежностей

EF Core (Entity Framework Core) — технологія доступу до даних (об'єктно-реляційний перетворювач)

ER-діаграма (Entity-Relationship diagram) — діаграма «сутність-зв'язок»

FIFO (First In, First Out) — перший прийшов — перший пішов (алгоритм організації та маніпулювання даними / метод списання запасів)

GUI (Graphical User Interface) — графічний інтерфейс користувача

GUID (Globally Unique Identifier) — глобально унікальний ідентифікатор

MVVM (Model-View-ViewModel) — архітектурний шаблон проектування «Модель-Відображення-Модель відображення»

ORM (Object-Relational Mapping) — об'єктно-реляційне відображення

PDF (Portable Document Format) — формат портативного документа

POCO (Plain Old CLR Object) — простий об'єкт середовища CLR (об'єкт, не обтяжений успадкуванням або атрибутами фреймворку)

SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation, Dependency inversion) — п'ять базових принципів об'єктно-орієнтованого програмування та проектування

SQL (Structured Query Language) — мова структурованих запитів

WPF (Windows Presentation Foundation) — графічна підсистема у складі .NET для побудови інтерфейсів користувача

XAML (Extensible Application Markup Language) — розширювана мова розмітки застосунків

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ВСТУП

У сучасних умовах стрімкого розвитку економіки та підвищення рівня мобільності населення, послуги з оренди автомобілів набувають дедалі більшої популярності. Зростання попиту на короткостроковий та довгостроковий прокат транспортних засобів вимагає від компаній, що працюють у цій сфері, високої швидкості обслуговування, бездоганної точності обліку та ефективного управління наявним автопарком. Традиційні методи ведення бізнесу, що базуються на паперовому документообігу, розрізнених електронних таблицях або застарілих системах обліку, є нерелевантними вимогам сьогодення. Вони неминуче призводять до виникнення помилок через людський фактор, випадків подвійного бронювання, втрати часу на оформлення договорів оренди та суттєво ускладнюють процес аналітики фінансових показників.

Відтак, комплексна автоматизація бізнес-процесів є критично важливою умовою для забезпечення конкурентоспроможності, масштабованості та рентабельності компанії з оренди автомобілів. Впровадження спеціалізованої інформаційної системи дозволяє централізувати зберігання даних, оптимізувати цикли бронювання, видачі та повернення автомобілів, а також автоматизувати складні розрахунки вартості послуг з урахуванням різних критеріїв.

Окремої уваги заслуговує вибір технологічного стека для реалізації програмного продукту. Для підприємств малого та середнього бізнесу часто оптимальним рішенням є надійні десктопні додатки, які забезпечують швидку роботу, високий рівень безпеки даних, що зберігаються локально, та незалежність від постійного підключення до глобальної мережі. Використання новітньої платформи .NET 9 у поєднанні з об'єктно-орієнтованою мовою програмування C# гарантує високу продуктивність та раціональне використання системних ресурсів комп'ютера. Застосування технології Windows Presentation Foundation (WPF) дозволяє створити сучасний, інтуїтивно зрозумілий та ергономічний графічний інтерфейс користувача з

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

використанням стилів та графічних анімацій. Водночас інтеграція легкої реляційної системи управління базами даних SQLite забезпечує надійне збереження та швидке оброблення транзакційних даних без необхідності розгортання та адміністрування громіздких серверних СУБД. З огляду на це, розробка запропонованої інформаційної системи є актуальним науково-практичним завданням.

Метою кваліфікаційної роботи є проектування та розробка надійної інформаційної системи автоматизації бізнес-процесів компанії з оренди автомобілів для підвищення ефективності управління автопарком, покращення якості обслуговування клієнтів та забезпечення точного обліку орендних операцій.

Для досягнення поставленої мети було сформульовано та вирішено такі **завдання дослідження**:

- проаналізувати предметну область, специфіку бізнес-процесів автопрокатних підприємств та виявити функціональні недоліки існуючих підходів до управління;
- сформулювати вимоги до інформаційної системи та спроектувати її програмну архітектуру з використанням патерну проектування MVVM (Model-View-ViewModel);
- розробити концептуальну, логічну та фізичну моделі бази даних для зберігання інформації про транспортні засоби, клієнтську базу та історію орендних угод, реалізувавши їх засобами СУБД SQLite;
- реалізувати програмний продукт мовою C# на базі платформи .NET 9;
- створити зручний графічний інтерфейс користувача засобами WPF, який включає інтерактивні каталоги, панелі адміністрування (CRUD-операції) та візуалізацію даних;
- імплементувати алгоритми автоматичного підрахунку вартості оренди на основі заданих періодів, а також логіку ведення статистики вільних, зайнятих автомобілів і розрахунку загального прибутку;

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

- провести комплексне тестування розробленого програмного забезпечення для підтвердження його коректності та безперебійності роботи.

Об'єктом дослідження є бізнес-процеси компанії, що надає послуги з оренди автомобілів, зокрема процеси управління автопарком, взаємодії з клієнтами та фінансового моніторингу.

Предметом дослідження є методи, алгоритми та програмно-інструментальні засоби (C#, .NET 9, WPF, SQLite) розробки інформаційної системи для автоматизації процесів автопрокату підприємства.

Методи дослідження. Методологічною та теоретичною основою кваліфікаційної роботи є загальнонаукові та спеціальні методи: метод системного аналізу (для вивчення структури предметної області), методи об'єктно-орієнтованого аналізу та проектування (для побудови архітектури програмного забезпечення), методи проектування реляційних баз даних (для розробки структур збереження інформації), а також емпіричні методи тестування та зневадження програмного коду.

Практичне значення отриманих результатів полягає у тому, що розроблена інформаційна система є повністю функціональним програмним продуктом, готовим до експлуатації та впровадження у повсякденну діяльність підприємств з прокату автомобілів. Її застосування дозволить суттєво скоротити часові витрати персоналу на обробку замовлень, мінімізувати ризики людських помилок, забезпечити прозорий облік транспортних засобів, а також надати керівництву дієвий інструмент для відстеження прибутковості та прийняття обґрунтованих управлінських рішень.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні задачі

Діяльність підприємств, що надають послуги з оренди транспортних засобів, супроводжується необхідністю безперервної обробки значних масивів інформації. Ефективне функціонування такого бізнесу залежить від чіткої координації між різними структурними підрозділами або процесами: від первинної реєстрації клієнта до фінального повернення автомобіля та формування фінансової звітності. Враховуючи високу динаміку цих процесів, використання застарілих методів обліку (паперових журналів або непов'язаних електронних таблиць) стає критичним фактором стримування розвитку компанії.

З огляду на специфіку предметної області, можна виокремити ключові бізнес-процеси, які потребують невідкладної оптимізації. Відповідно, головною метою впровадження розроблюваної інформаційної системи є вирішення комплексу взаємопов'язаних задач, які можна об'єднати у такі логічні блоки:

1.1.1 Автоматизація обліку та управління.

Автомобільний парк є основним активом прокатної компанії. Основна задача полягає у забезпеченні точного та своєчасного обліку кожної одиниці техніки. Інформаційна система повинна надавати функціонал для:

- додавання нових автомобілів із зазначенням ключових характеристик (марка, модель, клас комфортності, базова вартість оренди за добу, фотографічне зображення);
- редагування та актуалізації даних про транспортні засоби;
- вилучення з обліку (видалення) автомобілів, що виведені з експлуатації;
- моніторингу поточного статусу транспортного засобу в режимі реального часу («Доступний» або «В оренді») для уникнення логістичних колізій та подвійного бронювання.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

1.1.2 Систематизація роботи з клієнтською базою.

Швидкість та якість обслуговування клієнтів є визначальними факторами конкурентоспроможності. Задача системи полягає у створенні надійного та структурованого сховища клієнтських даних. Це передбачає:

- реєстрацію нових орендарів із фіксацією їхніх ідентифікаційних даних (прізвище, ім'я, по батькові, контактний номер телефону, серія та номер паспортного документа);
- забезпечення механізмів швидкого пошуку, редагування та видалення записів про клієнтів;
- створення підґрунтя для подальшого аналізу клієнтської активності (історії оренд).

1.1.3 Оптимізація процесу оформлення орендних угод.

Це центральна транзакційна задача інформаційної системи. Вона покликана мінімізувати вплив людського фактора та прискорити процес укладання договору оренди. Дана задача охоплює:

- встановлення зв'язку між конкретним клієнтом та обраним транспортним засобом на визначений період часу (дати початку та закінчення оренди);
- автоматичний підрахунок загальної вартості послуги на основі тривалості періоду оренди та добового тарифу конкретного автомобіля (з урахуванням його класу);
- автоматичну зміну статусу автомобіля під час відкриття та закриття орендного періоду.

1.1.4 Збір статистики та фінансова аналітика.

Для прийняття ефективних управлінських рішень керівництву компанії необхідна достовірна інформація про результати комерційної діяльності. Задача аналітичного модуля полягає у:

- агрегації даних щодо загального прибутку підприємства за всіма виконаними орендними угодами;

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

- наданні кількісної статистики щодо поточного стану автопарку (відсоткове або кількісне співвідношення вільних та зайнятих автомобілів);
- формуванні історії поїздок для виявлення найбільш рентабельних транспортних засобів.

Вирішення вищезазначених задач за допомогою розробки десктопного програмного застосунку дозволить сформувати єдиний інформаційний простір компанії. Це, у свою чергу, суттєво підвищить прозорість операційної діяльності, зменшить адміністративні витрати часу та забезпечить високий рівень достовірності даних, необхідних для стабільного функціонування автопрокатного підприємства.

1.2 Огляд існуючих рішень

Ринок інформаційних технологій пропонує широкий спектр програмних продуктів, орієнтованих на автоматизацію процесів автопрокатного бізнесу. Існуючі рішення можна умовно поділити на три основні категорії: спеціалізовані хмарні сервіси (SaaS-рішення), універсальні системи управління взаємовідносинами з клієнтами (CRM) та базові офісні програми (електронні таблиці). Для обґрунтування доцільності розробки власної інформаційної системи необхідно провести порівняльний аналіз найбільш популярних аналогів, виявити їхні переваги та критичні недоліки.

Одним із найвідоміших рішень на світовому ринку є хмарна платформа RentSyst. Це спеціалізоване програмне забезпечення, яке пропонує повний цикл управління автопарком. Система інтегрується з GPS-трекерами, дозволяє автоматизувати виставлення рахунків та підтримує онлайн-бронювання.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

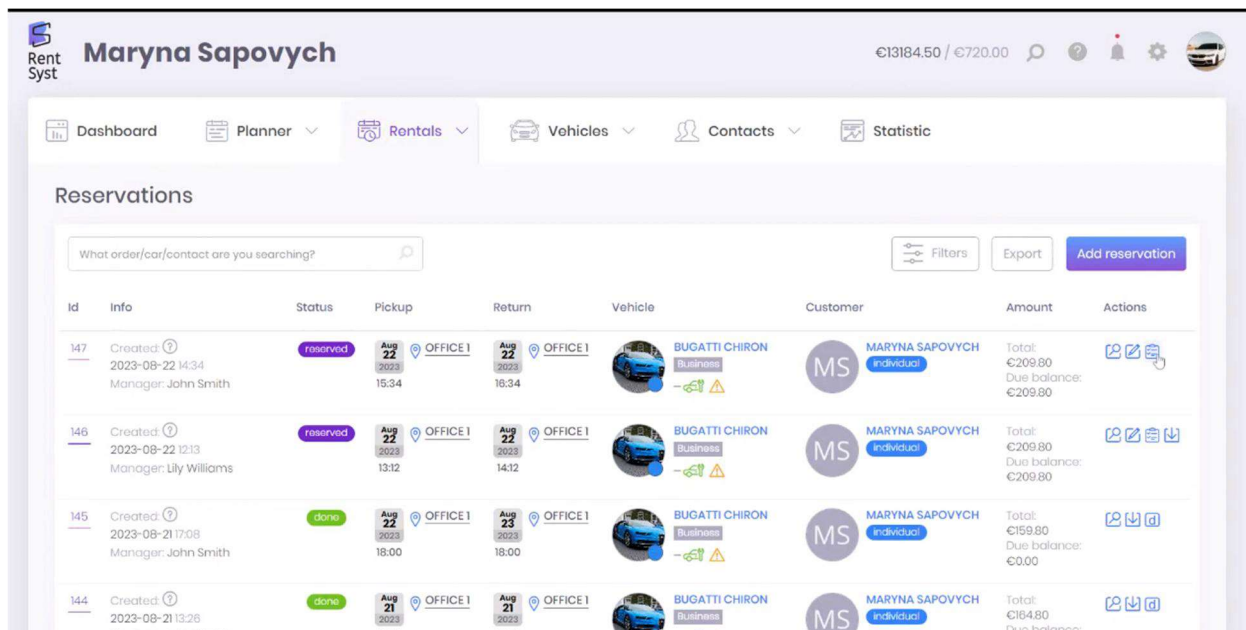


Рисунок 1.2.1. – Графічний інтерфейс користувача RentSyst

Перевагами системи RentSyst є високий рівень деталізації аналітики та можливість доступу з будь-якого пристрою через веб-браузер. Проте система має низку суттєвих недоліків для малого та середнього бізнесу. По-перше, це модель монетизації на основі щомісячної абонентської плати, яка розраховується за кожен зареєстрований автомобіль. По-друге, залежність від постійного підключення до мережі Інтернет: у разі технічних збоїв у провайдера робота пункту прокату повністю паралізується. Крім того, зберігання конфіденційних даних клієнтів на сторонніх серверах може порушувати внутрішні політики безпеки деяких компаній.

Іншим потужним спеціалізованим комплексом є HQ Rental Software. Дана система орієнтована на великі корпоративні мережі автопрокату і пропонує широкі можливості кастомізації, підтримку франчайзингових мереж та інтеграцію з платіжними шлюзами.

HQ Ries

Marguerite Roberts - RA #60 Assign Vehicle Print Agreement Email Agreement Email Templates

1 Dates **2** Cars 3 Features 4 Customer 5 Confirm 6 Payments 7 Pickup 8 Contract 9 Return 10 Changes

Selected Vehicle Class

Premium
Automatic Transmission
Airconditioning
Radio/CD Player
Power Steering
4-doors

\$42.86 /Day
\$300.00 total for 7 Days
Excl. taxes & insurance(s)
Cars Available: 3 (75%)

Rent this Car

Select Other Vehicle Class

Economic Manual
Manual Transmission
Airconditioning
Radio/CD Player
Power Steering
4-doors

\$25.00 /Day
\$175.00 total for 7 Days
Excl. taxes & insurance(s)
Cars Available: 2 (33.3%)

Rent this Car

Economic Automatic
Automatic Transmission
Airconditioning

Recommended

Summary

The Security Deposit hasn't been paid yet.

Pickup
Thursday, Nov 17, 2022 @ 11:12 PM
Location: Office

Return
Thursday, Nov 24, 2022 @ 11:12 PM
Location: Office

Premium		
1x	\$300.00	\$300.00
Weeks:		\$300.00
Protections & Coverages		
CDW:		\$77.00
Miscellaneous		
Sales Tax (5%):		\$18.85
Payments		
Paid Commission Partner:		\$300.00
Total		\$395.85
Outstanding Balance		

Рисунок 1.2.2. – Панель адміністрування HQ Rental Software

До сильних сторін HQ Rental Software належить потужний функціонал для динамічного ціноутворення. Водночас основними недоліками цього рішення є його висока вартість, надмірна перевантаженість інтерфейсу надлишковим функціоналом (який не використовується локальними компаніями) та складність процесу впровадження, що вимагає тривалого навчання персоналу.

На противагу спеціалізованим системам, значна частина локальних підприємств на початкових етапах діяльності використовує універсальні офісні програми, такі як Microsoft Excel або Google Sheets.

Використання електронних таблиць не потребує додаткових фінансових витрат на ліцензування та є звичним для більшості користувачів. Однак такий підхід повністю виключає поняття автоматизації. Відсутність реляційної цілісності даних призводить до дублювання інформації. Крім того, неможливість автоматичного підрахунку вартості оренди з урахуванням різних тарифних сіток, високий ризик помилок через людський фактор та відсутність багатоклієнтського доступу із розмежуванням прав роблять цей метод абсолютно неефективним при масштабуванні автопарку.

Проведений аналіз існуючих рішень демонструє наявність поляризації на ринку: з одного боку пропонуються дорогі, перевантажені функціями хмарні системи, а з іншого — примітивні ручні методи обліку. Для компаній малого та середнього розміру оптимальним вибором є створення власного десктопного застосунку (на базі платформи .NET та технології WPF).

Розробка власної інформаційної системи із застосуванням локальної СУБД SQLite дозволить:

1. Уникнути регулярних витрат на підписку (SaaS) або придбання дорогих ліцензій.
2. Забезпечити повну незалежність від якості інтернет-з'єднання.
3. Гарантувати безпеку та конфіденційність персональних даних клієнтів шляхом їх локального зберігання.
4. Створити мінімалістичний та адаптований суто під бізнес-процеси конкретного підприємства інтерфейс користувача без надлишкових функцій.

Розробка запропонованої у кваліфікаційній роботі системи є технічно та економічно обґрунтованим рішенням, що дозволить задовольнити потреби компанії в автоматизації без залучення надмірних ресурсів.

1.3 Функціональні вимоги до програми

На основі проведеного аналізу предметної галузі та визначених цілей розробки, було сформовано комплекс функціональних вимог, які регламентують поведінку, можливості та програмні реакції розроблюваної інформаційної системи. Функціональні вимоги визначають перелік операцій, які система повинна виконувати для забезпечення автоматизації бізнес-процесів компанії з оренди автомобілів. Для структурування вимог їх розподілено за основними логічними модулями програмного продукту.

1. Вимоги до модуля управління автопарком:

- Система повинна забезпечувати можливість додавання нового транспортного засобу до бази даних із обов'язковим заповненням

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

атрибутів: марка, модель, клас автомобіля (наприклад, економ, бізнес, преміум), вартість оренди за одну добу, а також графічне зображення автомобіля.

- Програмний додаток має автоматично копіювати обрані користувачем файли зображень до внутрішньої локальної директорії проекту з генерацією унікальних ідентифікаторів імен (GUID), що гарантує збереження файлів та їхню автономну доступність.
- Система повинна надавати функціонал для редагування параметрів існуючих автомобілів та заміни їхніх фотографій.
- Повинна бути реалізована можливість видалення запису про автомобіль з бази даних (за умови відсутності активних пов'язаних орендних угод).
- Система має автоматично відстежувати та відображати поточний експлуатаційний статус транспортного засобу («Доступний» або «В оренді»).

2. Вимоги до модуля управління клієнтською базою:

- Програма повинна дозволяти реєстрацію нового клієнта із введенням його персональних ідентифікаційних даних: прізвище, ім'я, по батькові (ПІБ), контактний номер телефону та реквізити документа, що посвідчує особу (серія та номер паспорта).
- Система має підтримувати базові CRUD-операції (читання, оновлення, видалення) для записів таблиці клієнтів, забезпечуючи швидкий пошук та навігацію у вигляді табличного представлення (DataGrid).

3. Вимоги до модуля управління процесом оренди:

- Система повинна надавати інтерфейс для оформлення нової орендної угоди, який вимагає вибору конкретного автомобіля (з переліку доступних) та існуючого клієнта.
- Програма має забезпечувати введення календарних дат початку та прогнозованого закінчення оренди.
- На основі обраних часових меж та встановленого добового тарифу конкретного автомобіля, система повинна автоматично розраховувати

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

загальну вартість орендної послуги до моменту фінального підтвердження операції.

- При підтвердженні оформлення оренди система зобов'язана автоматично ініціювати транзакцію у базі даних: створити активний запис в історії оренд та змінити статус відповідного автомобіля на «В оренді».
- Система повинна містити механізм завершення оренди (повернення автомобіля), який переводить статус угоди в архівний (неактивний), а статус транспортного засобу — у вільний.

4. Вимоги до аналітично-статистичного модуля:

- Інформаційна система має агрегувати дані та в режимі реального часу відображати статистичне зведення: загальну кількість автомобілів у парку, кількість вільних транспортних засобів та кількість одиниць техніки, що перебувають в оренді.
- Система повинна автоматично підраховувати загальний фінансовий прибуток компанії за всіма історичними (архівними) та поточними угодами.
- Повинна бути реалізована функція перегляду індивідуальної статистики по кожному окремо обраному автомобілю (кількість здійснених поїздок та сумарний згенерований дохід).

5. Загальносистемні та інтерфейсні вимоги:

- Графічний інтерфейс користувача повинен бути реалізований із застосуванням технології WPF на базі архітектурного патерну MVVM, що забезпечує чітке розділення візуального представлення та бізнес-логіки.
- Навігація між підсистемами (автопарк, оренда, клієнти) має здійснюватися за допомогою вкладок (TabControl) для мінімізації кількості відкритих вікон.
- Програма повинна реалізовувати валідацію введених даних на стороні клієнта (наприклад, блокування кнопок збереження у разі наявності

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

порожніх обов'язкових текстових полів або від'ємного значення вартості).

- Оновлення даних в інтерфейсі (реактивність) має відбуватися синхронно зі змінами у базі даних SQLite без необхідності примусового перезавантаження всього застосунку.

1.4 Вибір технології

Вибір інструментальних засобів та технологій розробки є одним із найважливіших етапів проектування інформаційної системи. Обраний технологічний стек безпосередньо визначає продуктивність, масштабованість, рівень безпеки даних та загальну вартість подальшого супроводу програмного продукту. Враховуючи необхідність створення надійного десктопного застосунку для управління процесами оренди автомобілів, було обрано комплекс технологій, що базується на екосистемі Microsoft та сучасних стандартах програмної інженерії.

Базовою мовою програмування обрано C# — сучасну, об'єктно-орієнтовану мову зі строгою типізацією. Її використання забезпечує високу надійність коду, безпеку типів та ефективне управління пам'яттю за рахунок механізму автоматичного збирання сміття (Garbage Collection). Програмною платформою виступає .NET 9 — новітня версія фреймворку, яка пропонує оптимізовану швидкість виконання базових операцій, розширені можливості бібліотек класів (BCL) та підтримку передових синтаксичних конструкцій. Застосування платформи .NET 9 гарантує довгострокову підтримку програмного продукту та його відповідність найсучаснішим стандартам розробки програмного забезпечення.

Для створення графічного інтерфейсу користувача (GUI) застосовано технологію WPF (Windows Presentation Foundation). На відміну від застарілих бібліотек (зокрема Windows Forms), WPF використовує апаратне прискорення графічного процесора для рендерингу візуальних елементів, що забезпечує плавність роботи інтерфейсу та можливість імплементації складних графічних

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

ефектів і анімацій. Фундаментальною перевагою WPF є використання декларативної мови розмітки XAML (Extensible Application Markup Language), яка дозволяє фізично та логічно розмежувати візуальне представлення від обчислювальної логіки. Потужна система прив'язки даних (Data Binding), підтримка шаблонів відображення (Data Templates) та гнучка система стилізації роблять WPF оптимальним інструментом для розробки корпоративних систем з високими вимогами до ергономіки та користувацького досвіду (UX).

Використання технології WPF нерозривно пов'язане із застосуванням архітектурного шаблону MVVM (Model-View-ViewModel). Цей патерн забезпечує відокремлення моделі даних (Model) та користувацького інтерфейсу (View) за допомогою проміжного компонента — моделі представлення (ViewModel). Застосування підходу MVVM підвищує модульність програмного коду, усуває дублювання логіки, суттєво спрощує процес модульного тестування (Unit Testing) та дозволяє незалежно модифікувати графічний інтерфейс без необхідності внесення змін у базові алгоритми програми.

Для організації зберігання, пошуку та модифікації транзакційних даних (інформації про автомобілі, клієнтів та договори оренди) обрано реляційну систему управління базами даних SQLite. Дана СУБД є вбудованою (embedded), що означає відсутність потреби у розгортанні, конфігуруванні та адмініструванні окремого сервера баз даних. Уся база даних фізично зберігається у вигляді єдиного локального файлу, що кардинально спрощує процес розповсюдження та інсталяції інформаційної системи на робочі станції підприємства. Попри свою компактність, SQLite повною мірою підтримує стандарти мови SQL, забезпечує виконання транзакцій за принципами ACID (атомарність, узгодженість, ізольованість, довговічність) та гарантує збереження цілісності даних. Це цілком задовольняє потреби локальної системи обліку компанії малого чи середнього масштабу.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Інтеграція платформи .NET 9, мови програмування C#, візуального фреймворку WPF у поєднанні з архітектурою MVVM та локальною СУБД SQLite утворює збалансовану, надійну та високопродуктивну архітектуру. Визначений технологічний стек повністю відповідає специфіці предметної області та надає всі необхідні інструментальні засоби для реалізації сформульованих функціональних вимог до інформаційної системи.

1.5 Постановка задачі

На основі результатів проведеного аналізу предметної галузі, огляду існуючих програмних рішень та визначеного переліку функціональних вимог, виникає обґрунтована необхідність у практичному створенні власного програмного продукту.

Головною задачею даної кваліфікаційної роботи є проектування та безпосередня програмна реалізація локальної інформаційної системи (десктопного застосунку) для автоматизації процесів обліку та управління підприємством з оренди автомобілів.

Для досягнення поставленої мети необхідно виконати декомпозицію головної задачі на такі взаємопов'язані етапи:

1. Розробити логічну та фізичну структуру реляційної бази даних під управлінням СУБД SQLite. Створити нормалізовані таблиці для надійного зберігання сутностей «Автомобілі» (Cars), «Клієнти» (Clients) та «Орендні угоди» (Rentals), забезпечивши підтримку реляційної цілісності даних за допомогою зовнішніх ключів (Foreign Keys).
2. Використовуючи мову програмування C# та платформу .NET 9, реалізувати алгоритмічну базу застосунку. Це передбачає написання методів для виконання транзакційних CRUD-операцій, алгоритмів перевірки експлуатаційної доступності транспортних засобів, математичного розрахунку підсумкової вартості оренди на основі часових проміжків, а також агрегації статистичних та фінансових показників.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

3. Створити програмний механізм для завантаження, унікального іменування (через генерацію GUID) та безпечного локального копіювання фотографій транспортних засобів до внутрішньої файлової структури проекту.
4. Застосовуючи технологію WPF та архітектурний шаблон MVVM, розробити сучасний, реактивний та інтуїтивно зрозумілий інтерфейс. Реалізувати навігацію на базі системи вкладок (TabControl), розробити ергономічні діалогові вікна для введення даних та забезпечити візуалізацію масивів інформації за допомогою списків і таблиць (DataGrid).
5. Провести комплексне зневадження (дебагінг) розробленого програмного забезпечення, перевірити коректність обробки виключних ситуацій (наприклад, введення некоректних даних користувачем) та оптимізувати взаємодію програми з локальною базою даних.

Очікуваним результатом виконання поставленої задачі є повністю функціональний, автономний та готовий до дослідної експлуатації програмний продукт. Розроблена інформаційна система повинна ефективно вирішувати проблему надлишкового паперового документообігу, мінімізувати вплив людського фактора під час укладання орендних угод та надати керівництву компанії зручний інструмент для моніторингу комерційної діяльності автопарку.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи було проведено комплексний аналіз предметної галузі управління підприємствами, що надають послуги з оренди автомобілів. Дослідження специфіки бізнес-процесів дозволило встановити, що для ефективного функціонування таких компаній критично необхідною є автоматизація процесів обліку автопарку, ведення клієнтської бази та фіксації орендних угод. Використання застарілих методів управління (паперові носії, незв'язані електронні таблиці) визнано неефективним через високий ризик виникнення помилок та відсутність оперативної аналітики.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Огляд існуючих на ринку програмних рішень (SaaS-платформ та універсальних систем) виявив їхню поляризацію: наявні системи є або занадто дороговартісними, перевантаженими надлишковим функціоналом та залежними від підключення до Інтернету, або ж занадто примітивними для забезпечення повноцінної автоматизації. Це економічно та технічно обґрунтувало доцільність розробки власної локальної десктопної інформаційної системи, адаптованої під потреби конкретного підприємства.

На основі виявлених потреб було сформовано чіткий і структурований перелік функціональних вимог до розроблюваного програмного продукту. Система повинна забезпечувати повноцінне виконання CRUD-операцій для довідників автомобілів і клієнтів, автоматизувати процеси оформлення та завершення оренди (із динамічним підрахунком вартості послуг), а також містити аналітичний модуль для агрегації статистичних даних та розрахунку прибутковості підприємства.

Для практичної реалізації сформульованих вимог було обґрунтовано вибір сучасного та надійного технологічного стека:

1. Базовою платформою обрано .NET 9 та об'єктно-орієнтовану мову програмування C#, що гарантує високу продуктивність застосунку.
2. Для проектування візуальної складової обрано технологію WPF, використання якої у поєднанні з архітектурним патерном MVVM дозволить створити інтуїтивно зрозумілий інтерфейс і чітко розмежувати візуальне представлення від обчислювальної логіки.
3. У якості засобу збереження даних обрано вбудовану реляційну СУБД SQLite, яка забезпечує локальну автономність, високу швидкість обробки транзакцій та надійність збереження інформації без необхідності адміністрування серверів.

У першому розділі було повністю визначено концептуальні засади майбутньої інформаційної системи.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

Етап проектування є фундаментальною складовою життєвого циклу розробки програмного забезпечення. У цьому розділі здійснюється концептуальне, логічне та фізичне моделювання архітектури розроблюваної інформаційної системи. Якісне проектування структур збереження даних забезпечує їхню цілісність, швидкодію системи та створює надійне підґрунтя для подальшої програмної реалізації бізнес-логіки.

2.1 Модель бази даних

Для забезпечення надійного зберігання, швидкого пошуку та обробки транзакційних даних компанії розробляється реляційна модель бази даних (БД). Враховуючи вимоги до локальної автономності системи, як засіб реалізації БД обрано вбудовану систему управління базами даних SQLite.

2.1.1 Опис предметної галузі та основних сутностей

Предметна галузь охоплює діяльність підприємства, що надає послуги з прокату транспортних засобів. Основним бізнес-процесом є передача автомобіля у тимчасове користування клієнту на визначений термін за відповідну плату. З огляду на це, інформаційна модель повинна фіксувати три головні інформаційні потоки: облік наявного автопарку, ведення реєстру орендарів (клієнтів) та реєстрацію фактів укладання орендних угод.

Відповідно до результатів системного аналізу предметної галузі, було виокремлено такі основні сутності:

- **Автомобілі (Cars):** сутність, що зберігає інформацію про весь транспортний парк компанії, незалежно від поточного статусу оренди.
- **Клієнти (Clients):** сутність, що містить персональні та контактні дані осіб, які користуються або користувалися послугами компанії.
- **Орендні угоди (Rentals):** транзакційна сутність, що фіксує факт передачі певного автомобіля конкретному клієнту на заданий період.

2.1.2 ER-діаграма

Для візуалізації логічної структури бази даних розроблено ER-діаграму (Entity-Relationship Diagram).

Структура діаграми базується на реляційній парадигмі та складається з трьох нормалізованих таблиць, які відповідають визначеним сутностям. Діаграма ілюструє назви таблиць, перелік атрибутів (полів) із зазначенням їхніх типів даних, первинні ключі для унікальної ідентифікації записів, а також зовнішні ключі, що формують зв'язки між сутностями.

Перелік сутностей:

1. Cars (Автомобілі)
2. Clients (Клієнти)
3. Rentals (Оренди)

Типи зв'язків та їхня кардинальність:

- **Клієнти (Clients) — Оренди (Rentals):** зв'язок типу **1:N** (один до багатьох). Кардинальність: один клієнт може мати нуль, одну або безліч орендних угод за історію обслуговування, але одна конкретна орендна угода завжди належить виключно одному клієнту.
- **Автомобілі (Cars) — Оренди (Rentals):** зв'язок типу **1:N** (один до багатьох). Кардинальність: один автомобіль може здаватися в оренду багато разів (бути пов'язаним із багатьма записами в історії оренд), проте один конкретний запис про оренду стосується лише одного автомобіля.
(Примітка: Зв'язок між клієнтами та автомобілями є зв'язком типу N:M, який на логічному рівні успішно розв'язується через створення проміжної сутності «Rentals», перетворюючись на два зв'язки типу 1:N).

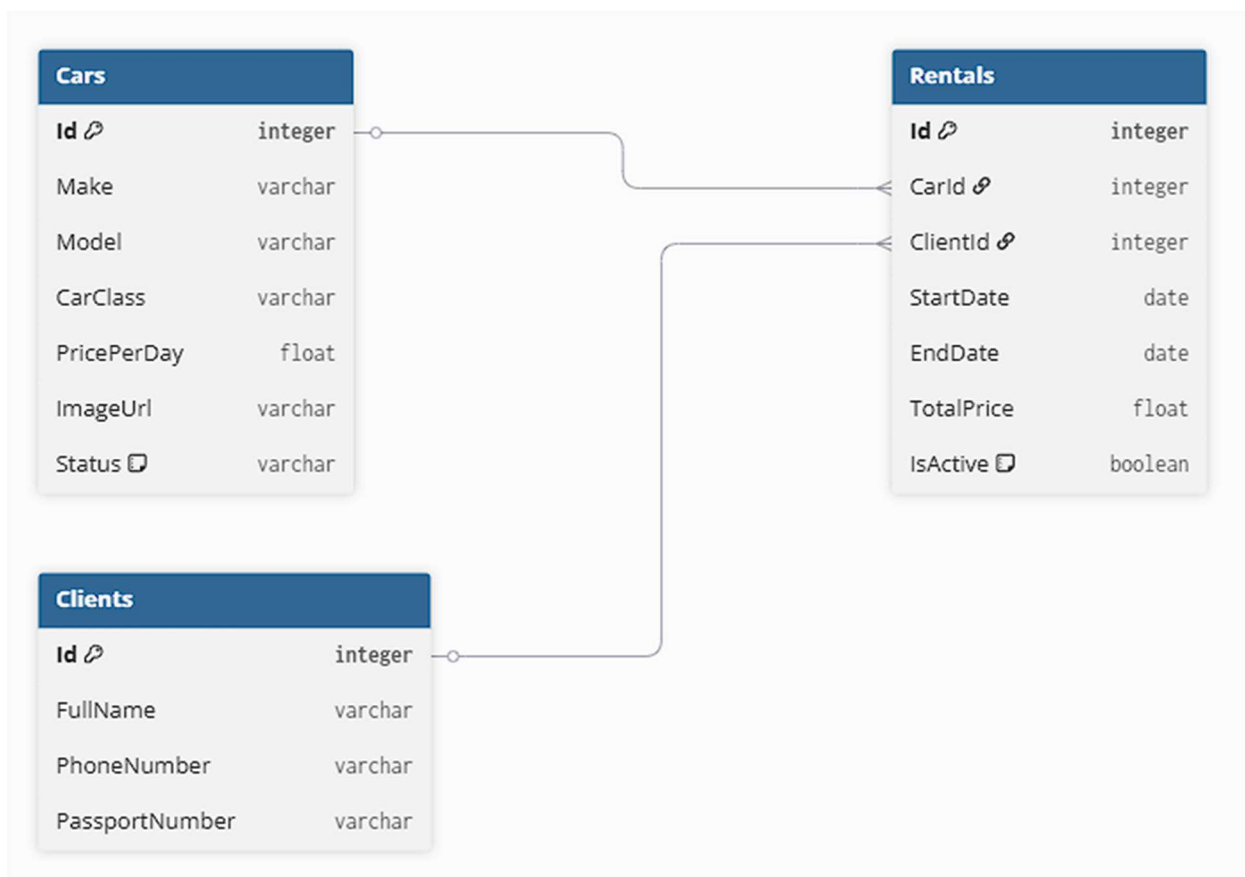


Рисунок 2.1.1. – ER-діаграма

Сутності та їхні атрибути:

1. Назва сутності: Cars (Автомобілі)

- **Первинний ключ:** Id (Унікальний ідентифікатор автомобіля).
- **Основні атрибути:** Make (Марка), Model (Модель), CarClass (Клас комфорту), PricePerDay (Вартість за добу), ImageUrl (Посилання на фото), Status (Поточний експлуатаційний статус).

2. Назва сутності: Clients (Клієнти)

- **Первинний ключ:** Id (Унікальний ідентифікатор клієнта).
- **Основні атрибути:** FullName (ПІБ клієнта), PhoneNumber (Контактний номер телефону), PassportNumber (Серія та номер паспорта).

3. Назва сутності: Rentals (Орендні угоди)

- **Первинний ключ:** Id (Унікальний ідентифікатор угоди).
- **Основні атрибути:** CarId (Зовнішній ключ автомобіля), ClientId (Зовнішній ключ клієнта), StartDate (Дата початку оренди),

EndDate (Дата завершення оренди), TotalPrice (Загальна вартість послуги), IsActive (Статус угоди: активна/завершена).

У спроектованій реляційній моделі (що відповідає Першій нормальній формі – 1NF) усі атрибути є простими (атомарними, тобто такими, що не поділяються на менші незалежні складові в межах даної архітектури) та однозначними (кожен запис містить лише одне значення для конкретного атрибута, без масивів чи списків). Наприклад, атрибут FullName концептуально є складнозначним (Прізвище, Ім'я, По батькові), проте на фізичному рівні він реалізований як простий рядковий атрибут для оптимізації бази даних малого підприємства.

Опис зв'язків:

1. Зв'язок 1:

- **Пов'язані сутності:** Clients та Rentals.
- **Тип зв'язку:** 1:N (Один до багатьох).
- **Назва зв'язку:** «Оформлює» (Клієнт оформлює оренду).

2. Зв'язок 2:

- **Пов'язані сутності:** Cars та Rentals.
- **Тип зв'язку:** 1:N (Один до багатьох).
- **Назва зв'язку:** «Здається в» (Автомобіль здається в оренду).

Оскільки зв'язки реалізовані через 1:N, безпосередніх атрибутів зв'язку (характерних для N:M) немає. Транзакційна сутність Rentals сама по собі виступає резолюцією відношення N:M між клієнтами та авто.

Спроектowana модель бази даних приведена до Третьої нормальної форми (3NF), що гарантує уникнення надмірності даних (аномалій оновлення, вставки та видалення). Наприклад, контактні дані клієнта не дублюються у кожній орендній угоді; замість цього зберігається лише унікальний ідентифікатор клієнта (ClientId). Цілісність даних на рівні бази гарантується використанням механізмів первинних ключів (Primary Keys), що унеможливлюють існування ідентичних записів, та зовнішніх ключів (Foreign Keys), що забезпечують реляційну цілісність. Зовнішні ключі (CarId та ClientId

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

у таблиці Rentals) гарантують, що неможливо створити орендну угоду для неіснуючого клієнта або неіснуючого автомобіля, а також захищають від випадкового видалення клієнтів чи авто, якщо за ними закріплені активні історичні записи у журналі оренд.

2.1.3 Фізична модель бази даних

Фізична модель бази даних відображає безпосередню реалізацію спроектованої логічної моделі засобами конкретної системи управління базами даних (СУБД). Оскільки в якості цільової СУБД обрано SQLite, фізична модель враховує її специфічні особливості. Зокрема, SQLite використовує динамічну типізацію та обмежений набір класів пам'яті. Тому логічні типи даних адаптовано до підтримуваних фізичних типів: рядки та дати зберігаються у форматі TEXT, дійсні числа (ціни) – як REAL, а логічні значення (boolean) та цілі числа ідентифікаторів – як INTEGER.

Нижче наведено детальний опис фізичної структури кожної таблиці бази даних із зазначенням полів, їхніх типів даних, обмежень (констрейнтів) та ключів.

Таблиця 2.1.3.1 – Фізична структура таблиці «Cars» (Автомобілі)

Назва поля	Тип даних SQLite	Ключ / Обмеження	Опис поля
Id	INTEGER	PRIMARY KEY AUTOINCREMENT	Унікальний ідентифікатор автомобіля (суррогатний первинний ключ).
Make	TEXT	NOT NULL	Марка транспортного засобу (наприклад, Toyota, BMW).
Model	TEXT	NOT NULL	Модель транспортного засобу.
CarClass	TEXT	NOT NULL	Клас автомобіля (Економ, Бізнес, Преміум).
PricePerDay	REAL	NOT NULL	Вартість добової оренди у національній валюті.
ImageUrl	TEXT	NULL	Локальний шлях до збереженого файлу із зображенням авто.
Status	TEXT	NOT NULL, DEFAULT 'Available'	Поточний експлуатаційний статус. За замовчуванням – доступний.

Таблиця 2.1.3.2 – Фізична структура таблиці «Clients» (Клієнти)

Назва поля	Тип даних SQLite	Ключ / Обмеження	Опис поля
Id	INTEGER	PRIMARY KEY AUTOINCREMENT	Унікальний ідентифікатор клієнта (суррогатний первинний ключ).
FullName	TEXT	NOT NULL	Повне прізвище, ім'я та по батькові клієнта.
PhoneNumber	TEXT	NOT NULL	Контактний номер телефону.
PassportNumber	TEXT	NOT NULL	Серія та номер паспорта або іншого документа, що посвідчує особу.

Таблиця 2.1.3.3 – Фізична структура таблиці «Rentals»

Назва поля	Тип даних SQLite	Ключ / Обмеження	Опис поля
Id	INTEGER	PRIMARY KEY AUTOINCREMENT	Унікальний ідентифікатор орендної угоди.
CarId	INTEGER	NOT NULL, FOREIGN KEY	Зовнішній ключ, що вказує на ідентифікатор орендованого автомобіля (Cars.Id).
ClientId	INTEGER	NOT NULL, FOREIGN KEY	Зовнішній ключ, що вказує на ідентифікатор клієнта-орендаря (Clients.Id).
StartDate	TEXT	NOT NULL	Дата початку оренди (формат ISO 8601).
EndDate	TEXT	NOT NULL	Дата очікуваного завершення оренди (формат ISO 8601).
TotalPrice	REAL	NOT NULL	Розрахована загальна вартість послуги оренди.
IsActive	INTEGER	NOT NULL, DEFAULT 1	Статус угоди: 1 (Активна), 0 (Завершена/Архівна).

Для забезпечення цілісності даних на фізичному рівні таблиця Rentals містить зовнішні ключі (Foreign Keys), які налаштовано на посилання до відповідних первинних ключів у таблицях Cars та Clients.

Загальну структуру та взаємозв'язки між фізичними таблицями бази даних наведено на рисунку нижче.

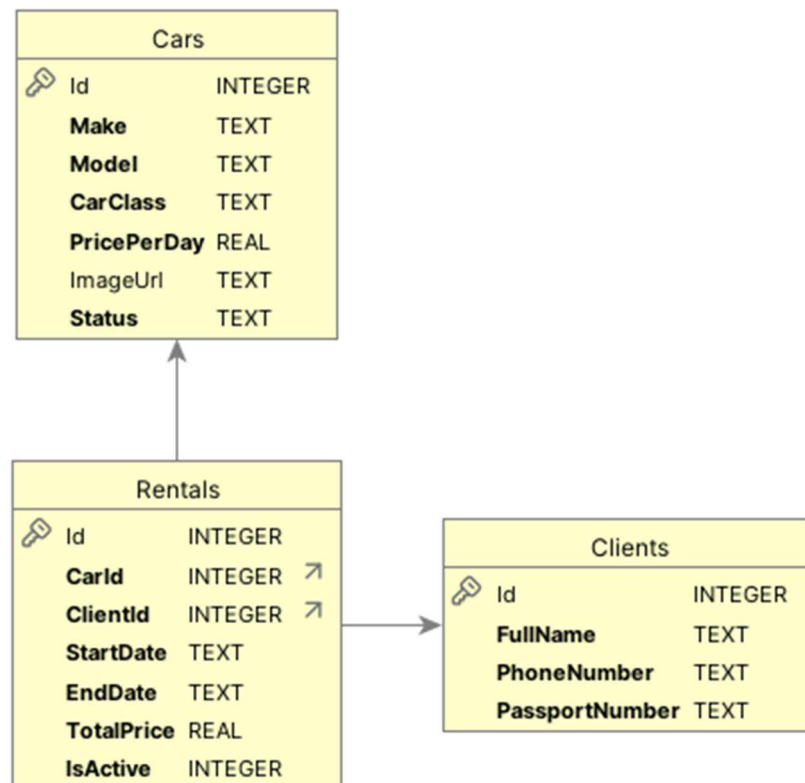


Рисунок 2.1.2. – Схема бази даних

Зведену інформацію про призначення усіх об'єктів бази даних наведено у таблиці нижче.

Таблиця 2.1.3.4 – Структура таблиць бази даних

Фізична назва таблиці	Логічна назва сутності	Призначення
Cars	Автомобілі	Зберігання реєстру транспортних засобів, їхніх техніко-економічних характеристик та поточного статусу доступності.
Clients	Клієнти	Зберігання ідентифікаційних та контактних даних осіб, що користуються послугами прокату.
Rentals	Орендні угоди	Фіксація транзакцій оренди: зв'язування клієнта з автомобілем на певний часовий проміжок, збереження суми розрахунку та поточного статусу угоди.

2.2 Список основних форм та їх призначення

Розроблений програмний продукт володіє сучасним графічним інтерфейсом, побудованим на базі технології Windows Presentation Foundation із застосуванням архітектурного шаблону проектування Model-View-ViewModel. Взаємодія користувача з базою даних та бізнес-логікою здійснюється через систему спеціалізованих візуальних форм, кожна з яких відповідає за ізольований логічний сегмент предметної галузі. Оскільки система проектувалася з акцентом на ергономіку, більшість робочих областей інтегровано у єдиний віконний простір за допомогою механізму вкладок, а операції додавання чи глибокого редагування винесені в окремі модальні діалогові вікна.

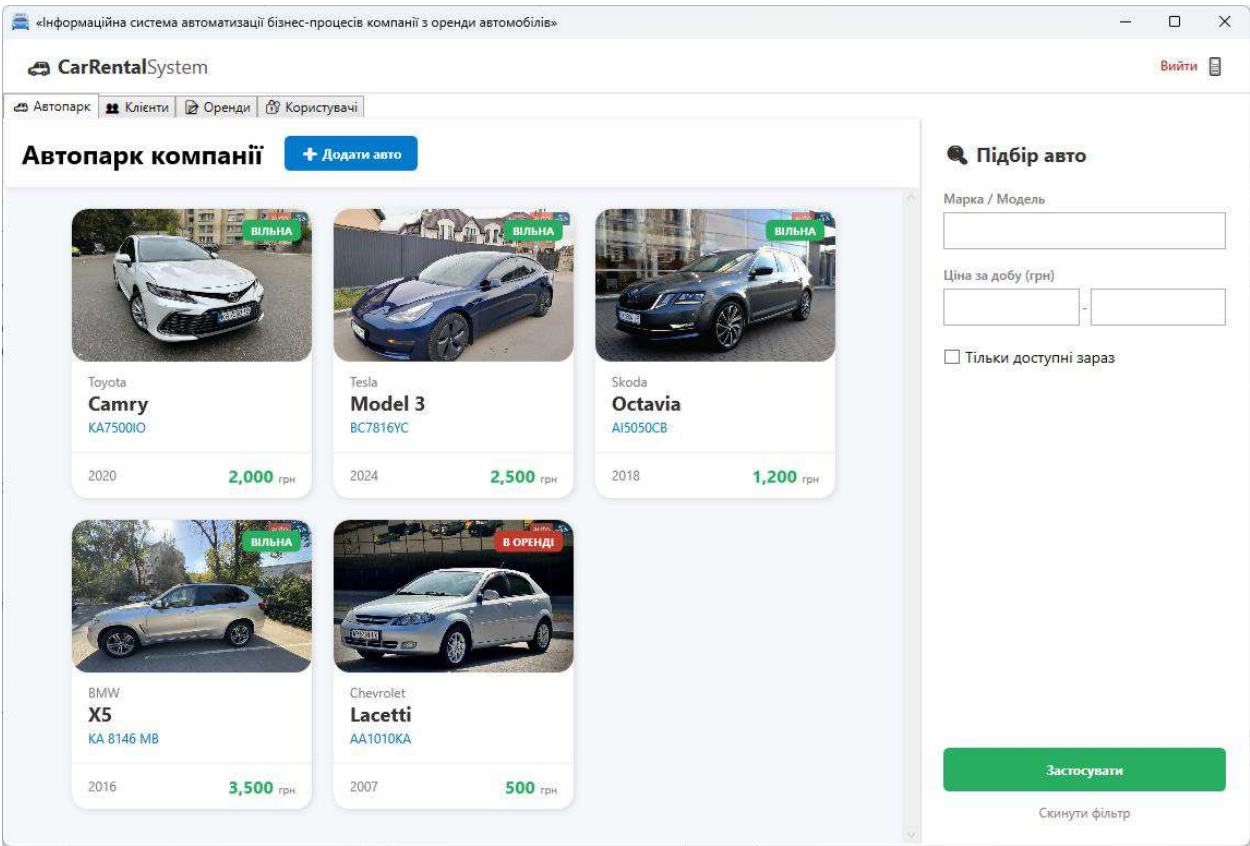


Рисунок 2.2.1. – Головна форма

Головна форма інформаційної системи виконує роль центрального робочого простору та навігаційного хаба. Її основним призначенням є консолідація ключових модулів програми та надання користувачеві швидкого доступу до аналітичного зведення. На цій формі розташовані панелі маршрутизації, які дозволяють перемикатися між каталогом автопарку,

реєстром клієнтів та журналом транзакцій. Також головна форма в режимі реального часу агрегує та візуалізує статистичні показники компанії, відображаючи загальну кількість доступних транспортних засобів, кількість автомобілів в активній оренді та сумарний згенерований дохід у числовому форматі.

Деталі автомобіля



Натисніть щоб змінити фото

Статистика

Всього поїздок

2 разів

Загальний прибуток

75,500 грн

Марка

Модель

Держ. номер

Рік випуску

Ціна (грн/день)

☐ Доступний для оренди

Видалити
Скасувати
Зберегти

Рисунок 2.2.2. – Форма реєстрації та редагування автомобіля

Форма реєстрації та редагування автомобіля реалізована як відокремлене модальне вікно, що запобігає втраті фокусу користувача під час виконання критичних операцій з базою даних. Її призначенням є безпечне введення техніко-економічних характеристик нових одиниць автопарку або модифікація існуючих записів. Користувач повинен ввести дані про марку та модель транспортного засобу у форматі коротких текстових рядків. Клас комфортності автомобіля обирається із заздалегідь визначеного набору значень за допомогою випадаючого списку, що мінімізує ймовірність

синтаксичних помилок. Базова вартість добової оренди вводиться у спеціалізоване поле, яке приймає виключно числові значення у форматі з плаваючою комою. Окремим функціональним блоком цієї форми є механізм роботи з медіафайлами: користувач ініціює діалогове вікно вибору графічного зображення автомобіля, після чого система зчитує шлях до файлу формату PNG або JPEG та створює його локальну копію у директорії програми.

Картка клієнта



Змінити

ПІБ
Іваненко Іван Іванович

Телефон
+380679998877

Паспорт
ME654321

ІПН

Адреса

Історія поїздок



BMW X5 (2016)

15.02.2026 — 16.02.2026

3,500 грн

☒ Завершено



Skoda Octavia (2018)

16.01.2026 — 21.01.2026

6,000 грн

☒ Завершено

Видалити

Скасувати

Зберегти

Рисунок 2.2.3. – Форма адміністрування клієнтської бази

Форма адміністрування клієнтської бази інтегрована безпосередньо у відповідну вкладку головного вікна і призначена для безперервного обліку осіб, які користуються послугами прокату. Для створення нового запису про клієнта система вимагає введення ідентифікаційних реквізитів. Повне ім'я клієнта вводиться у форматі текстового рядка, який може містити пробіли та літери алфавіту. Контактний номер телефону передбачає введення символічних та числових значень у міжнародному або локальному форматі, тоді як реквізити документа, що посвідчує особу, фіксуються у вигляді буквено-цифрового рядка. Валідація цих полів на рівні графічного інтерфейсу гарантує, що запис не буде перенесений до бази даних у разі відсутності критично важливої контактної інформації.

Рисунок 2.2.4. – Форма оформлення орендної угоди

Форма оформлення орендної угоди виступає транзакційним ядром інформаційної системи, забезпечуючи зв'язування сутностей клієнта та автомобіля в єдиний договір. Її інтерфейс спроектовано таким чином, щоб максимально автоматизувати роботу оператора. Користувачу не потрібно вручну вводити ідентифікатори: вибір автомобіля та клієнта здійснюється шляхом позиціонування на відповідних елементах інтерактивних списків, що

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

2.3 Архітектура програмного засобу

2.3.1 Загальні архітектурні принципи

Проектування архітектури інформаційної системи базується на принципах багаторівневості (Layered Architecture) та ізоляції відповідальності (Separation of Concerns). Такий підхід передбачає логічне розділення програмного коду на незалежні рівні, кожен з яких виконує строго визначену функцію: рівень представлення даних (Presentation Layer), рівень бізнес-логіки (Business Logic Layer) та рівень доступу до даних (Data Access Layer). Застосування фундаментальних принципів об'єктно-орієнтованого дизайну, зокрема принципів SOLID, забезпечує високу модульність системи. Це дозволяє здійснювати незалежне тестування, модифікацію та масштабування окремих компонентів без ризику порушення стабільності всієї системи. Відсутність жорсткого зв'язування між рівнями досягається шляхом програмування на рівні інтерфейсів, що робить архітектуру гнучкою до можливих майбутніх змін, таких як міграція на іншу систему управління базами даних або розширення клієнтського функціоналу.

2.3.2 Шаблон MVVM та механізм прив'язки даних

Основою рівня представлення розробленого десктопного застосунку є архітектурний шаблон MVVM (Model-View-ViewModel), який є галузевим стандартом для розробки додатків на базі технології WPF. Шаблон концептуально розділяє графічний інтерфейс на три взаємопов'язані компоненти. Модель (Model) представляє предметну область та містить об'єктно-орієнтовані сутності бази даних. Відображення (View) відповідає виключно за візуалізацію інтерфейсу і реалізується декларативною мовою розмітки XAML. Модель представлення (ViewModel) виступає проміжною ланкою, що інкапсулює логіку взаємодії з користувачем та підготовку даних для відображення.

Ключовим технологічним рушієм шаблону MVVM є механізм двосторонньої прив'язки даних (Data Binding). Цей механізм дозволяє автоматично синхронізувати стан візуальних елементів (View) зі значеннями

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

властивостей у ViewModel. Реалізація інтерфейсу INotifyPropertyChanged забезпечує реактивність системи: будь-які зміни даних на програмному рівні миттєво генерують події, які сповіщають графічний інтерфейс про необхідність оновлення екрана. Взаємодія користувача з активними елементами управління (кнопками) обробляється не через класичні обробники подій, а за допомогою патерну Команда (через інтерфейс ICommand), що дозволяє повністю звільнити файли відображення (code-behind) від логіки виконання операцій.

2.3.3 Система навігації між сторінками

Навігаційна модель інформаційної системи побудована з урахуванням концепції єдиного вікна (Single Page Application-подібний підхід для десктопу) з метою оптимізації використання оперативної пам'яті та покращення користувацького досвіду. Замість ініціалізації множини незалежних вікон, основний робочий простір поділено на логічні модулі за допомогою системи вкладок (TabControl). Кожна вкладка інкапсулює власний контекст даних і відповідає за окрему підсистему (автопарк, клієнти, оренда). Для виконання транзакційно критичних операцій, таких як додавання нового транспортного засобу або редагування параметрів, застосовується механізм модальних діалогових вікон. Виклик цих вікон ініціюється з рівня ViewModel, передаючи необхідний контекст даних (наприклад, сутність автомобіля для редагування), при цьому зберігається блокування основного вікна до завершення операції користувачем. Це гарантує лінійність виконання бізнес-сценаріїв та запобігає виникненню конфліктів асинхронного доступу до даних.

2.3.4 Контейнер впровадження залежностей

Для забезпечення слабкого зв'язування компонентів системи та реалізації принципу інверсії управління (Inversion of Control) в архітектуру інтегровано механізм впровадження залежностей (Dependency Injection). Використання DI-контейнера дозволяє централізовано реєструвати всі сервіси, репозиторії та моделі представлення на етапі запуску програми. Замість прямого створення екземплярів класів через оператор new всередині

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

ViewModel, необхідні залежності передаються через їхні конструктори у вигляді абстракцій (інтерфейсів). Це кардинально спрощує архітектуру, оскільки компоненти не керують життєвим циклом своїх залежностей. Крім того, такий підхід створює ідеальні умови для модульного тестування, дозволяючи легко підміняти реальні сервіси доступу до бази даних на імітаційні об'єкти (mocks) під час перевірки бізнес-логіки.

2.3.5 Шаблон Repository та рівень доступу до даних

Рівень взаємодії з системою управління базами даних SQLite ізольовано від інших компонентів за допомогою патерну Repository (Репозиторій). Цей шаблон створює абстрактний фасад над інфраструктурою бази даних, приховуючи специфіку виконання SQL-запитів та особливості ADO.NET від верхніх рівнів архітектури. Репозиторій надає стандартизований набір методів для виконання базових CRUD-операцій (створення, читання, оновлення, видалення) над сутностями предметної області. У розробленій системі логіка доступу до даних обробляє встановлення з'єднань (SqlConnection), параметризацію запитів для захисту від SQL-ін'єкцій та перетворення (мапінг) результатів виконання запитів у строго типізовані колекції об'єктів мови C#. Це забезпечує високий рівень безпеки та дозволяє локалізувати всі зміни структури БД виключно в межах даного рівня.

2.3.6 Сервісний рівень

Сервісний рівень (Service Layer) виступає координаційним центром інформаційної системи, що містить ключові алгоритми бізнес-логіки компанії з оренди автомобілів. Він розташовується між рівнем представлення (ViewModel) та рівнем доступу до даних (Repository). Сервісні класи відповідають за виконання складних операцій, які виходять за межі простих транзакцій з базою даних. До їхньої компетенції належить автоматичний математичний розрахунок загальної вартості оренди на основі часових проміжків, перевірка логічної доступності автомобіля на обрані дати (валідація конфліктів бронювання) та керування локальною файловою системою (генерація унікальних ідентифікаторів для зображень транспортних

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

засобів та їх безпечне копіювання до директорій додатку). Ізоляція цих правил у сервісному рівні запобігає «роздуванню» класів ViewModel та забезпечує можливість повторного використання бізнес-алгоритмів.

2.3.7 Ініціалізація та запуск застосунку

Життєвий цикл розробленого програмного продукту починається у точці входу, якою є клас App.xaml.cs. Процес ініціалізації системи перехоплюється шляхом перевизначення базового методу OnStartup. На цьому етапі відбувається конфігурація та налаштування всього середовища виконання. Програма здійснює перевірку наявності локального файлу бази даних SQLite та, у разі його відсутності, ініціює виконання DDL-скриптів для створення необхідних реляційних таблиць. Паралельно виконується налаштування контейнера впровадження залежностей (реєстрація життєвих циклів Singleton або Transient для сервісів та моделей представлення). Лише після успішного підтвердження цілісності інфраструктури зберігання даних та реєстрації всіх компонентів, система ініціалізує головне вікно (MainWindow), передає йому відповідний контекст даних і передає управління користувачеві, забезпечуючи стабільний та безпечний старт роботи інформаційної системи.

2.4 Програмна реалізація

2.4.1 Реалізація моделей даних

Програмна реалізація об'єктної моделі предметної області базується на створенні строгих типізованих класів мовою C#. Ці класи виконують роль структур даних (POCO-об'єктів), що концептуально відображають сутності бази даних у оперативній пам'яті застосунку. Для таблиці транспортних засобів розроблено клас, який інкапсулює властивості ідентифікатора, марки, моделі, технічного класу, вартості оренди та шляху до медіафайлу. Аналогічним чином реалізовано класи для збереження ідентифікаційних параметрів клієнтів та параметрів транзакцій орендних угод. Відповідно до вимог платформи .NET 9, у моделях активно використовуються типи даних, що допускають значення null (nullable reference types), для безпечної обробки

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

необов'язкових полів. Крім базових (скалярних) властивостей, у класах реалізовано навігаційні властивості, які забезпечують об'єктно-орієнтоване відображення реляційних зв'язків між сутностями, дозволяючи програмно звертатися до об'єкта клієнта безпосередньо з екземпляра орендної угоди.

2.4.2 Конфігурація Entity Framework Core

Для забезпечення взаємодії об'єктної моделі з реляційною базою даних SQLite інтегровано об'єктно-реляційний перетворювач (ORM) Entity Framework Core. Центральним компонентом рівня доступу до даних виступає клас контексту, який успадковується від базового класу DbContext. У процесі ініціалізації контексту перевизначається метод конфігурації, де встановлюється рядок підключення до локального файлу бази даних за допомогою провайдера SQLite. Важливим етапом програмної реалізації є використання механізму Fluent API у методі створення моделі (OnModelCreating). За допомогою цього інструменту програмно визначаються первинні ключі, встановлюються обмеження на довжину рядкових полів, конфігуруються значення за замовчуванням та жорстко декларуються зв'язки типу «один до багатьох» з відповідними правилами каскадного видалення або обмеження зовнішніх ключів. Такий підхід гарантує точну відповідність згенерованої фізичної схеми бази даних спроектованій логічній моделі.

2.4.3 Реалізація репозиторіїв

З метою абстрагування бізнес-логіки від специфіки Entity Framework Core та забезпечення модульності коду реалізовано шаблон проектування Репозиторій (Repository). Створено спеціалізовані класи-репозиторії для кожної основної сутності системи, які імплементують відповідні інтерфейси. Ці класи інкапсулюють у собі екземпляр контексту бази даних і надають стандартизований набір методів для виконання асинхронних операцій читання, додавання, модифікації та видалення записів. Використання асинхронних методів на базі ключових слів `async` та `await` є критично важливим для десктопного застосунку, оскільки це запобігає блокуванню головного потоку інтерфейсу користувача під час виконання тривалих запитів

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

до бази даних. Крім базових операцій, репозиторії містять специфічні методи вибірки, наприклад, отримання списку виключно доступних автомобілів або формування повної історії оренд з урахуванням зв'язаних даних клієнтів.

2.4.4 Реалізація ViewModel-рівня

Рівень моделі представлення (ViewModel) є алгоритмічним ядром графічного інтерфейсу та реалізований як набір класів, що успадковують базовий клас із реалізацією інтерфейсу повідомлень про зміну властивостей. Головна модель представлення керує станом навігації та зберігає колекції даних у форматі ObservableCollection, що забезпечує автоматичне оновлення елементів інтерфейсу при модифікації списків. Для обробки дій користувача (натискання кнопок, вибір елементів меню) реалізовано механізм команд на базі шаблону RelayCommand. Кожна команда складається з делегата виконання, який містить безпосередню логіку операції, та делегата валідації, що визначає стан доступності візуального елемента. Наприклад, команда збереження нового автомобіля автоматично деактивує відповідну кнопку в інтерфейсі, якщо користувач не заповнив усі обов'язкові поля або ввів некоректне значення вартості.

2.4.5 Алгоритм виконання запиту та FIFO-списання

Програмна реалізація транзакційного модуля базується на строгому алгоритмі послідовної обробки запитів на оренду. При ініціалізації нової угоди система перевіряє доступність обраного автомобіля у заданому часовому вікні, блокуючи можливість паралельного бронювання. Застосування методу FIFO (First-In-First-Out) реалізовано у підсистемі керування чергами бронювань та обліку супутніх витратних ресурсів підприємства. Зокрема, алгоритм FIFO застосовується при списанні амортизаційних та технічних матеріалів (паливно-мастильні матеріали, комплектуючі), що обліковуються на балансі компанії під час проведення технічного обслуговування автопарку між циклами оренди. Програма автоматично списує вартість найстаріших закуплених партій матеріалів, формуючи коректну фінансову звітність

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

собівартості утримання транспортних засобів відповідно до бухгалтерських стандартів.

2.4.6 Реалізація сервісу генерації PDF

Для забезпечення процесу документообігу розроблено сервісний клас генерації офіційної документації у форматі PDF. Цей модуль базується на використанні спеціалізованих відкритих бібліотек для платформи .NET і реалізує функціонал автоматичного створення друкованої форми договору оренди та розрахункової квитанції. При підтвердженні транзакції сервіс отримує на вхід об'єкт орендної угоди, витягує з нього пов'язані дані компанії, реквізити клієнта та технічні параметри автомобіля. На основі цих даних програмно формується структурований документ із заданим форматуванням, таблицями розрахунків та місцями для підписів сторін. Згенерований файл зберігається у локальній директорії звітів із можливістю негайного відкриття або виведення на друк безпосередньо з інтерфейсу програми.

2.4.7 Реалізація перетворювачів значень

Забезпечення якісного візуального представлення технічних даних реалізовано за допомогою класів-перетворювачів значень, які імплементують інтерфейс `IValueConverter` платформи WPF. Вони призначені для трансформації типів даних між рівнем `ViewModel` та XAML-розміткою. Розроблено перетворювач для конвертації логічних статусів у текстові еквіваленти, який переводить числові або булеві індикатори стану автомобіля (наприклад, 1 або 0) у локалізовані рядки «Доступний» або «В оренді». Додатково імплементовано перетворювачі кольорів, що динамічно генерують об'єкти візуальних пензлів (`SolidColorBrush`) залежно від контексту даних. Це дозволяє автоматично підсвічувати активні орендні угоди зеленим кольором, а завершені або проблемні — відповідними відтінками сірого чи червоного, суттєво підвищуючи рівень ергономіки системи.

2.4.8 Допоміжна логіка обчислення

Всю складну математичну та бізнес-аналітику інкапсульовано у відокремлених допоміжних методах обчислення, що гарантує єдність правил розрахунку в усьому застосунку. Основний алгоритм обчислення вартості

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

оренди базується на знаходженні різниці між датою завершення та датою початку прокату за допомогою вбудованих структур часу (TimeSpan). Логіка враховує специфічні крайові умови: зокрема, якщо клієнт бере та повертає транспортний засіб в один і той самий день, система математично округлює час використання до однієї повної розрахункової доби. Окремі статичні методи забезпечують безпечне перетворення введених користувачем рядкових значень у типи з плаваючою комою, обробляючи регіональні особливості роздільників, а також виконують агрегуючі функції підрахунку загального фінансового прибутку шляхом проходження колекцій завершених угод.

2.5 Тестування програми

На завершальному етапі розробки інформаційної системи автоматизації бізнес-процесів автопрокату підприємства було проведено комплексне тестування програмного продукту. Головною метою цього процесу стала перевірка відповідності реалізованого функціоналу початковим вимогам та забезпечення високого рівня надійності системи. Для досягнення максимального покриття коду застосовувалася багаторівнева стратегія перевірки, що включала модульне, інтеграційне та системне тестування. На рівні модульного тестування здійснювалася ізольована перевірка ключових алгоритмів бізнес-логіки, зокрема механізмів обчислення вартості оренди та застосування алгоритму FIFO для списання ресурсів. Результати цих тестів підтвердили абсолютну математичну точність розрахунків навіть у випадках нестандартних часових проміжків чи некоректних вхідних даних користувача.

Інтеграційне тестування фокусувалося на перевірці коректності взаємодії між рівнем моделі представлення, сервісним рівнем та шаром доступу до даних. У ході цієї фази було доведено безпомилковість роботи технології Entity Framework Core під час виконання транзакційних CRUD-операцій у базі даних SQLite. Особлива увага приділялася перевірці механізмів збереження реляційної цілісності інформації, що унеможливлює випадкове видалення клієнтів або транспортних засобів із наявними активними

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

історичними угодами. Системне тестування дозволило оцінити загальну стабільність десктопного застосунку в умовах, максимально наближених до реального робочого середовища експлуатації. Перевірка графічного інтерфейсу користувача, побудованого на базі WPF, засвідчила коректну роботу механізмів двосторонньої прив'язки даних. Валідація форм введення успішно блокує спроби збереження порожніх полів або запису від'ємних фінансових показників, своєчасно інформуючи оператора про помилки.

Успішно пройшов перевірку модуль роботи з медіафайлами, який стабільно виконує безпечне копіювання зображень автомобілів до локальної директорії з генерацією унікальних ідентифікаторів. Тестування транзакційного ядра підтвердило надійність алгоритму бронювання, що ефективно запобігає виникненню логістичних колізій шляхом миттєвого оновлення статусів автопарку. Сервіс генерації звітної документації у форматі PDF продемонстрував безперебійну роботу, формуючи коректно структуровані договори без спотворень шрифтів чи зміщення макетів. Під час навантажувального тестування симулювалася робота зі значним обсягом історичних даних оренди для оцінки швидкості рендерингу візуальних списків. Застосування асинхронних запитів до бази даних повністю виправдало себе, оскільки графічний інтерфейс залишався чуйним і не блокувався під час виконання складних аналітичних вибірок.

Усі виявлені під час проміжних етапів перевірки незначні дефекти та неточності візуального відображення були оперативно задокументовані та повністю усунені. Загалом, результати комплексного тестування доводять високу відмовостійкість спроектованої архітектури та беззаперечну правильність обраних технологічних рішень. Інформаційна система повністю задовольняє визначені критерії приймання, демонструючи стабільну роботу всіх заявлених функціональних модулів. Розроблений програмний засіб є технічно завершеним, надійним інструментом і цілком готовим до повноцінного впровадження у щоденні бізнес-процеси автопрокатної компанії.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Висновки до розділу 2

У другому розділі кваліфікаційної роботи було здійснено комплексне проектування та безпосередню програмну реалізацію інформаційної системи для автоматизації бізнес-процесів автопрократного підприємства. На початковому етапі розроблено концептуальну, логічну та фізичну моделі реляційної бази даних, розгорнутої під управлінням локальної СУБД SQLite. Виокремлено ключові сутності предметної галузі, серед яких реєстр автомобілів, клієнтська база та транзакційний журнал орендних угод. Застосування принципів нормалізації та налаштування зовнішніх ключів забезпечило високий рівень реляційної цілісності, унеможливлюючи виникнення аномалій дублювання чи безповоротної втрати пов'язаних транзакційних даних.

Програмна архітектура розробленого десктопного застосунку базується на сучасній платформі .NET 9 з використанням технології побудови графічних інтерфейсів Windows Presentation Foundation. Застосування галузевого шаблону проектування Model-View-ViewModel у поєднанні з контейнером впровадження залежностей та патерном Repository дозволило створити модульну, масштабовану та відмовостійку структуру коду. Такий підхід забезпечив строге логічне розмежування візуального представлення від обчислювальної бізнес-логіки та рівня доступу до бази даних. Проектування системи візуальних форм, організованих за принципом єдиного навігаційного простору з вкладками та ізольованими модальними діалогами, гарантувало високий рівень ергономіки та безпечну лінійність виконання критичних операцій користувачем.

Програмна реалізація системи повністю задовольнила висунуті функціональні вимоги. Завдяки інтеграції об'єктно-реляційного перетворювача Entity Framework Core створено надійний механізм виконання асинхронних транзакцій. На сервісному рівні успішно імплементовано складні алгоритми безпомилкового обчислення вартості орендних послуг, динамічного контролю доступності транспортних засобів, математичного

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

списання амортизаційних ресурсів за принципом FIFO та модуль автоматизованої генерації офіційної звітної документації у форматі PDF. Окремо реалізовано автономну підсистему локального збереження медіафайлів із застосуванням криптографічно унікальних ідентифікаторів, що підвищує надійність зберігання графічних активів компанії.

Проведене багаторівневе тестування, яке охоплювало модульну, інтеграційну та системну перевірки, повністю підтвердило стабільність роботи закладених алгоритмів. Перевірка механізмів двосторонньої прив'язки даних та багаторівневих систем валідації введеної інформації засвідчила коректну обробку дій оператора і надійний захист цілісності даних на етапі введення. Спроектований та програмно реалізований інформаційний засіб є функціонально завершеним програмним продуктом, який ефективно вирішує завдання автоматизації обліку та управління автопарком, і є повністю готовим до впровадження у реальне комерційне середовище.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

3.1 Мінімальні вимоги до системи

Для забезпечення стабільної та безперебійної роботи розробленої інформаційної системи необхідно дотримуватися визначених мінімальних апаратних і програмних вимог. Оскільки програмний продукт побудовано на базі технології Windows Presentation Foundation (WPF), його функціонування можливе виключно в середовищі операційних систем сімейства Microsoft Windows. Мінімальною підтримуваною версією операційної системи є Windows 10 (версія 1607 або новіша) із 64-розрядною архітектурою, проте для оптимальної швидкодії рекомендується використання актуальних збірок Windows 11.

Апаратна частина комп'ютеризованого робочого місця оператора повинна містити сучасний багатоядерний центральний процесор із тактовою частотою не менше 2,0 ГГц для швидкої обробки транзакційних запитів бази даних. Для комфортної роботи реактивного графічного інтерфейсу та завантаження об'ємних колекцій даних мінімальний обсяг оперативної пам'яті (RAM) становить 4 ГБ, хоча доцільним є використання 8 ГБ. Базова інсталяція інформаційної системи потребує щонайменше 500 МБ вільного дискового простору для розміщення виконуваних файлів та необхідних системних бібліотек. Додатково слід обов'язково передбачити не менше 2 ГБ пам'яті на твердотільному накопичувачі (SSD) або жорсткому диску для автономного зберігання локального файлу бази даних SQLite та каталогу з графічними зображеннями транспортних засобів. Для коректного візуального відображення навігаційних панелей, діалогових вікон та зведених табличних звітів мінімальна роздільна здатність монітора має становити 1366 на 768 пікселів.

Критично важливою програмною залежністю розробленого десктопного застосунку є обов'язкова наявність попередньо встановленого середовища виконання .NET 9 Desktop Runtime. Дотримання вищезазначених мінімальних системних характеристик гарантуватиме високу відмовостійкість

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

програмного продукту та швидке виконання всіх закладених бізнес-алгоритмів підприємства з оренди автомобілів.

3.2 Інструкція для роботи з програмою

Початок роботи з інформаційною системою здійснюється шляхом запуску виконуваного файлу програми на персональному комп'ютері оператора. Після успішної ініціалізації та автоматичного підключення до локальної бази даних на екрані з'являється головне вікно застосунку, яке слугує центральним робочим простором. Графічний інтерфейс логічно поділено на кілька функціональних зон за допомогою системи вкладок, розташованих у верхній частині екрана. Це забезпечує зручну маршрутизацію та швидкий доступ до основних модулів управління автопарком, клієнтською базою, транзакціями оренди та зведеними статистичними даними без необхідності відкриття множини розрізнених вікон.

Управління транспортними засобами здійснюється у відповідній вкладці каталогу автопарку, де автомобілі представлені у вигляді візуальних карток із зображеннями та ключовими характеристиками. Для додавання нової одиниці техніки оператору необхідно ініціювати відкриття модального діалогового вікна реєстрації, де вимагається заповнення текстових полів з маркою, моделлю та базовою добовою вартістю оренди. Вибір класу комфортності здійснюється через випадаюче меню, а завантаження фотографії відбувається шляхом виклику системного провідника для вибору графічного файлу, який програма автоматично скопіює та безпечно збереже у власній внутрішній директорії. Редагування існуючих записів виконується шляхом виділення потрібного автомобіля у загальному списку та натискання кнопки модифікації, після чого система завантажить поточні параметри у форму для внесення змін. Процедура видалення транспортного засобу з обліку також вимагає попереднього виділення елемента та обов'язкового підтвердження дії у попереджувальному діалоговому вікні для запобігання випадковій втраті важливої інформації.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Робота з клієнтською базою відбувається у спеціалізованому модулі, який відображає реєстр орендарів у вигляді структурованої таблиці даних. Для реєстрації нового клієнта оператор повинен викликати форму додавання та внести актуальні ідентифікаційні відомості, включаючи повне ім'я, контактний номер телефону та реквізити паспортного документа. Вбудована система валідації не дозволить зберегти транзакцію у разі виявлення порожніх або некоректно заповнених обов'язкових полів, гарантуючи таким чином цілісність бази даних. Модифікація персональних даних або вилучення клієнтських записів з реєстру здійснюється за аналогічним принципом виділення потрібного рядка таблиці з подальшим викликом відповідної команди керування, за умови відсутності активних прив'язок клієнта до поточних орендних угод.

Центральним етапом роботи з програмним продуктом є процес оформлення орендної угоди, який реалізовано у транзакційній вкладці. Для укладання нового договору оператору необхідно послідовно обрати зі списків доступний транспортний засіб та попередньо зареєстрованого клієнта. Далі, за допомогою візуальних календарних компонентів, слід вказати точні дати початку та прогнозованого закінчення періоду прокату. Програмний комплекс самостійно розрахує загальну вартість послуги на основі обраного тарифу та тривалості використання автомобіля, миттєво вивівши підсумкову фінансову суму на екран. Після підтвердження транзакції оператором система автоматично змінить статус автомобіля на зайнятий, сформує відповідний запис в історії операцій та згенерує розрахункову документацію у форматі PDF для подальшого друку. Для завершення періоду оренди та повернення автомобіля у вільний обіг оператору достатньо знайти відповідну активну угоду в журналі транзакцій і виконати команду її закриття.

Моніторинг ефективності комерційної діяльності підприємства здійснюється через аналітичні панелі, показники яких оновлюються алгоритмами системи в режимі реального часу. Перебуваючи у головному вікні, оператор може постійно відстежувати загальну кількість доступних на

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

даний момент транспортних засобів, кількість машин, що наразі перебувають в експлуатації клієнтами, та сумарний фінансовий прибуток компанії, згенерований за весь період роботи. Деталізована статистика щодо експлуатаційної рентабельності кожної конкретної одиниці техніки стає доступною при її виділенні у загальному каталозі, де система автоматично вираховує та відображає кількість здійснених комерційних поїздок та сумарний дохід, принесений цим транспортним засобом.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи було комплексно розглянуто практичні аспекти розгортання, експлуатації та безпосередньої взаємодії кінцевого користувача з розробленою інформаційною системою автоматизації бізнес-процесів автопрокатного підприємства. Для забезпечення коректного функціонування програмного продукту було визначено та обґрунтовано мінімальні апаратні й програмні вимоги комп'ютеризованого робочого місця оператора. Встановлено, що використання сучасних обчислювальних потужностей у поєднанні з актуальними версіями операційної системи сімейства Windows та наявністю середовища виконання .NET 9 Desktop Runtime є необхідною умовою для гарантування високої відмовостійкості системи, плавності роботи реактивного графічного інтерфейсу та швидкої обробки транзакційних запитів локальної бази даних.

Значну увагу в розділі приділено формуванню вичерпної інструкції користувача, яка детально регламентує порядок виконання повсякденних виробничих завдань за допомогою розробленого програмного забезпечення. Описана логіка навігації, побудована за принципом єдиного робочого вікна з використанням системи вкладок, доводить високий рівень ергономіки застосунку та дозволяє оператору швидко перемикатися між модулями управління автопарком, клієнтською базою та журналом орендних угод. Послідовно розкрито алгоритми здійснення всіх ключових операцій: від первинного додавання транспортних засобів із завантаженням медіафайлів та

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

реєстрації нових орендарів до повного циклу оформлення договору прокату з автоматичним розрахунком фінансової вартості послуг.

Продемонстровано, що імплементовані механізми попередньої валідації введених даних та автоматизація математичних обчислень ефективно мінімізують вплив людського фактора, захищаючи базу даних від збереження некоректної інформації. Можливість моніторингу ключових показників рентабельності підприємства в режимі реального часу підтверджує аналітичну цінність розробленої системи для управлінського персоналу. Підсумовуючи викладене, можна стверджувати, що створена інформаційна система супроводжується належною експлуатаційною базою, є інтуїтивно зрозумілою для персоналу без глибокої технічної підготовки та є повністю готовою до успішного практичного впровадження у комерційну діяльність реальних підприємств.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було успішно вирішено актуальне науково-практичне завдання, яке полягало у проектуванні та розробці локальної інформаційної системи автоматизації бізнес-процесів компанії з оренди автомобілів. Поставлена на початку дослідження мета була повністю досягнена шляхом створення надійного, автономного та масштабованого програмного продукту. Розроблена система забезпечує комплексний підхід до управління ключовими аспектами діяльності автопрокатного підприємства, інтегруючи розрізнені операції у єдиний цифровий простір.

На етапі аналізу предметної галузі було виявлено, що використання традиційних методів обліку є неефективним і неминуче призводить до фінансових втрат через вплив людського фактора та відсутність інструментів оперативної аналітики. Проведений огляд існуючих комерційних рішень підтвердив економічну та технічну доцільність створення власного десктопного застосунку, адаптованого під специфічні потреби локального бізнесу без прив'язки до дороговартісних хмарних сервісів. На основі цих висновків було сформовано чіткий перелік функціональних вимог, який став надійним фундаментом для подальшого проектування програмної архітектури та баз даних.

Для практичної реалізації інформаційної системи було обрано найсучасніший технологічний стек, базою якого стала платформа .NET 9 та об'єктно-орієнтована мова програмування C#. Побудова графічного інтерфейсу користувача здійснювалася за допомогою технології Windows Presentation Foundation, що дозволило створити реактивне та високоергономічне візуальне середовище. Застосування галузевого архітектурного шаблону Model-View-ViewModel забезпечило чітке структурне розмежування між компонентами відображення та алгоритмами бізнес-логіки. Зберігання транзакційних даних організовано на базі вбудованої реляційної системи управління базами даних SQLite, яка гарантує високу

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

швидкість обробки інформації. Інтеграція об'єктно-реляційного перетворювача Entity Framework Core дозволила оптимізувати процес взаємодії з базою даних через застосування патерну репозиторію та механізмів асинхронного програмування.

Розроблений програмний засіб успішно реалізує повний цикл операцій з управління транспортним парком, включаючи механізми безпечного локального збереження графічних медіафайлів. Модуль роботи з клієнтською базою забезпечує надійну реєстрацію та зберігання ідентифікаційних даних орендарів із вбудованою системою валідації на етапі введення. Транзакційне ядро програми повністю автоматизує процес укладання орендних угод, самостійно обчислюючи кінцеву вартість послуг на основі заданих часових параметрів. Впроваджений аналітичний модуль дозволяє керівництву компанії в режимі реального часу здійснювати моніторинг рентабельності окремих одиниць автопарку та загального фінансового прибутку. Комплексне багаторівневе тестування підтвердило абсолютну стабільність роботи системи, безпомилковість математичних алгоритмів розрахунку та високий рівень захисту від некоректних дій оператора.

Сформовані вимоги до апаратного забезпечення комп'ютеризованого робочого місця та розроблена інструкція гарантують безпроблемне розгортання і швидке освоєння програми персоналом компанії. Ергономічний інтерфейс із системою вкладок мінімізує час на виконання рутинних операцій, суттєво підвищуючи загальну продуктивність праці співробітників. Практичне значення отриманих результатів полягає в тому, що створений застосунок є функціонально завершеним продуктом, який повністю готовий до впровадження у виробничий процес реального підприємства. Подальший розвиток системи може передбачати розширення аналітичного функціоналу, інтеграцію платіжних шлюзів або розробку додаткових модулів для автоматичного формування бухгалтерської звітності.

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Бойко В. І., Савченко О. М. Оптимізація роботи з базами даних в середовищі .NET за допомогою Entity Framework Core. Сучасні інформаційні технології та інноваційні методики навчання. 2023. Вип. 60. С. 22–29.
2. Габрусев В. Ю. Бази даних та інформаційні системи : навч. посіб. Тернопіль : ТНТУ, 2018. 156 с.
3. Грицюк Ю. І., Рак Т. Є. Об'єктно-орієнтоване програмування мовою C# : навч. посіб. Львів : Вид-во ЛДУ БЖД, 2011. 404 с.
4. Коваленко О. В., Сидоренко І. В. Аналіз сучасних підходів до автоматизації бізнес-процесів підприємства. Економіка і управління підприємствами. 2020. № 4. С. 85–92.
5. Кононенко І. В., Букрєєва К. С. Моделювання та оптимізація бізнес-процесів : навч. посіб. Харків : НТУ «ХПІ», 2019. 212 с.
6. Левченко О. М. Проектування інформаційних систем : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 240 с.
7. Мартин Р. Чиста архітектура: мистецтво розробки програмного забезпечення / пер. з англ. Київ : Фабула, 2019. 368 с.
8. Мельник А. О., Ковальчук А. М. Інформаційна система управління автопарком прокатної компанії. Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі. 2021. Вип. 9. С. 112–119.
9. Пасічник В. В., Резніченко В. А. Організація баз даних та знань : підручник. Київ : Видавнича група BVH, 2006. 384 с.
10. Петренко О. І. Архітектурний шаблон MVVM у розробці корпоративних додатків на платформі WPF. Системи управління, навігації та зв'язку. 2019. Вип. 2 (54). С. 134–138.
11. Ткаченко В. М. Використання SQLite для забезпечення автономності настільних інформаційних систем. Інженерія програмного забезпечення. 2022. № 1. С. 45–50.

12. Трофименко О. Г., Прокоп Ю. В., Логінова Н. І., Задерейко О. В. С#. Основи програмування : навч. посіб. Одеса : Фенікс, 2019. 227 с.
13. Albahari J. C# 10 in a Nutshell: The Definitive Reference. Sebastopol : O'Reilly Media, 2022. 1064 p.
14. Date C. J. An Introduction to Database Systems. 8th ed. Boston : Addison-Wesley, 2003. 1008 p.
15. Entity Framework Core documentation [Електронний ресурс] / Microsoft. URL: <https://learn.microsoft.com/en-us/ef/core/> (дата звернення: 05.04.2026).
16. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston : Addison-Wesley, 2003. 560 p.
17. Fowler M. Presentation Model [Електронний ресурс] / Martin Fowler. URL: <https://martinfowler.com/eaaDev/PresentationModel.html> (дата звернення: 03.04.2026).
18. Japikse P., Troelsen A. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. New York : Apress, 2022. 1520 p.
19. MacDonald M. Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5. 4th ed. New York : Apress, 2012. 1072 p.
20. Microsoft C# Documentation [Електронний ресурс] / Microsoft. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 04.04.2026).
21. Richter J. CLR via C#. 4th ed. Redmond : Microsoft Press, 2012. 896 p.
22. SOLID Principles of Object-Oriented Design [Електронний ресурс] / FreeCodeCamp. URL: <https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/> (дата звернення: 02.04.2026).
23. SQLite Official Documentation [Електронний ресурс] / SQLite Consortium. URL: <https://www.sqlite.org/docs.html> (дата звернення: 05.04.2026).
24. The MVVM Pattern [Електронний ресурс] / Microsoft. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm> (дата звернення: 04.04.2026).

					КРФМБ.122.042.033.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53