

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 управління роботою
клінінгової компанії»**

Виконав здобувач освіти групи П-43стн
Ступаков Олександр Анатолійович

Керівник роботи:
Студзінський Володимир Вікторович

Рецензент:

Зміст

Вступ	7
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	10
1.1 Основні задачі	10
1.2 Огляд існуючих рішень	10
1.3 Функціональні вимоги до програми	11
1.4 Вибір технології	11
1.5 Постановка задачі	12
Висновки до розділу 1	12
Розділ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ	13
2.1 Модель бази даних	13
2.2 Список основних форм та їх призначення	19
2.3 Архітектура програмного засобу	23
2.4 Програмна реалізація	26
2.5 Тестування програми	29
Висновки до розділу 2	30
Розділ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ	32
3.1 Мінімальні вимоги до системи	32
3.2 Інтерфейс користувача	33
3.3 Інструкція для роботи з програмою	34
Висновки до розділу 3	36
ВИСНОВКИ	38
Перелік літературних джерел	40
Додаток А. Код для роботи з базою даних	43
Додаток А. ПРОГРАМНИЙ КОД МОДУЛІВ	46

					КРФМБ.122.043стн.019.2026.ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система управління роботою клінінгової компанії			Літ.	Арк.	Аркуші		
Розробив		Ступаков О А									3	52
Перевірів		Студзінський В В										
Рецензент								ЖАТФК				
Н. Контр.												
Затвердив.		Гнатюк О.Ф.										

РЕФЕРАТ

Кваліфікаційна робота: 52 стор., 7 рис., 8 табл., 2 додатки, 20 джерел.

Об'єкт дослідження — процеси операційного та адміністративного управління діяльністю сучасної клінінгової компанії.

Мета роботи — розробка та впровадження інформаційної системи для автоматизації обліку клієнтів, формування замовлень, планування робочих графіків персоналу та моніторингу фінансових показників підприємства.

Методи дослідження — системний аналіз предметної галузі, об'єктно-орієнтоване проектування, реляційне моделювання баз даних та використання сучасних архітектурних шаблонів розробки програмного забезпечення.

У ході виконання роботи було спроектовано та реалізовано десктопний застосунок «CleanPro» на базі платформи .NET 9 із використанням технології WPF. Архітектура системи побудована за шаблоном MVVM, що забезпечує гнучкість інтерфейсу та зручність підтримки коду. Для збереження даних використано вбудовану СУБД SQLite та ORM Entity Framework Core 9.

Програма дозволяє розмежовувати права доступу для адміністраторів, менеджерів та прибиральників, автоматизувати розрахунок вартості послуг та вести прозорий облік платежів. Впровадження системи сприяє оптимізації використання трудових ресурсів, підвищенню точності фінансового планування та покращенню якості обслуговування клієнтів.

Ключові слова: *ІНФОРМАЦІЙНА СИСТЕМА, КЛІНІНГ, .NET 9, WPF, MVVM, SQLITE, АВТОМАТИЗАЦІЯ УПРАВЛІННЯ, БАЗА ДАНИХ.*

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД (DB) — база даних; впорядкована сукупність даних, що зберігаються відповідно до певної схеми.

ЗК (FK) — зовнішній ключ (Foreign Key); атрибут таблиці, що вказує на первинний ключ іншої таблиці для встановлення зв'язку.

ІС — інформаційна система; сукупність програмних та апаратних засобів для автоматизації обробки інформації.

ПК (PK) — первинний ключ (Primary Key); унікальний ідентифікатор запису в таблиці бази даних.

ПЗ — програмне забезпечення; сукупність програм, що забезпечують функціонування комп'ютерної системи.

СУБД — система управління базами даних; спеціалізоване ПЗ для створення та ведення баз даних (у роботі — SQLite).

API (Application Programming Interface) — програмний інтерфейс застосунку; набір визначень та протоколів для побудови та інтеграції програмного забезпечення.

BCrypt — адаптивна криптографічна хеш-функція, що використовується для безпечного зберігання паролів користувачів.

CRUD (Create, Read, Update, Delete) — чотири основні функції управління даними: створення, читання, оновлення та видалення.

DAL (Data Access Layer) — шар доступу до даних; компонент архітектури, що відповідає за взаємодію з базою даних.

EF Core (Entity Framework Core) — сучасне об'єктно-реляційне відображення (ORM) для .NET, що дозволяє працювати з БД через об'єкти C#.

LINQ (Language Integrated Query) — набір функцій у мові C# для виконання запитів до даних із різних джерел безпосередньо в коді.

MVVM (Model-View-ViewModel) — архітектурний шаблон проектування, що розділяє графічний інтерфейс від бізнес-логіки.

NuGet — система управління пакетами для платформи .NET, що використовується для підключення сторонніх бібліотек.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ORM (Object-Relational Mapping) — технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування.

SQLite — компактна вбудовувана реляційна СУБД, яка не потребує окремого сервера.

UI (User Interface) — інтерфейс користувача; сукупність засобів, за допомогою яких користувач взаємодіє з програмою.

ViewModel — компонент шаблону MVVM, який готує дані з моделі для відображення в інтерфейсі та обробляє команди користувача.

WPF (Windows Presentation Foundation) — система для побудови десктопних клієнтських застосунків для ОС Windows із використанням XAML.

XAML (Extensible Application Markup Language) — декларативна мова розмітки на основі XML, що використовується у WPF для проектування інтерфейсів.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Сучасний етап розвитку світової та вітчизняної економіки характеризується стрімким зростанням сфери послуг, серед яких особливе місце посідає професійне прибирання (клінінг). Динамічний ритм життя та орієнтація бізнесу на аутсорсинг непрофільних функцій зумовлюють високий попит на послуги клінінгових компаній, що, у свою чергу, веде до посилення конкуренції на цьому ринку. Ефективне управління таким підприємством вимагає оперативного опрацювання значних обсягів інформації: від обліку клієнтської бази та формування замовлень до координації роботи персоналу та фінансового моніторингу.

Традиційні методи ведення обліку за допомогою паперових носіїв або розрізнених електронних таблиць стають неефективними, оскільки призводять до виникнення помилок, дублювання даних та значних часових витрат на рутинні операції. У таких умовах розробка та впровадження спеціалізованої інформаційної системи управління стає не просто перевагою, а критичною необхідністю для забезпечення стабільного функціонування та розвитку підприємства. Використання сучасних засобів розробки, таких як платформа **.NET 9** та технологія **WPF**, дозволяє створити надійне, продуктивне та інтуїтивно зрозуміле програмне забезпечення, що відповідає актуальним вимогам бізнесу.

Об'єкт дослідження — процеси операційного, кадрового та фінансового управління діяльністю підприємства, що надає клінінгові послуги.

Предмет дослідження — методи, моделі та програмні засоби автоматизації процесів збору, зберігання, обробки та аналізу інформації в межах інформаційної системи управління клінінговою компанією.

Мета роботи — проектування та реалізація архітектурно цілісної та функціонально повної інформаційної системи для автоматизації ключових бізнес-процесів клінінгової компанії, що дозволить підвищити якість

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

обслуговування клієнтів, оптимізувати використання трудових ресурсів та забезпечити прозорість фінансових потоків.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

1. Проаналізувати предметну область та визначити функціональні вимоги до системи управління клінінговою компанією.
2. Спроекувати структуру бази даних для надійного зберігання інформації про користувачів, клієнтів, послуги та замовлення.
3. Розробити багаторівневу архітектуру системи на основі шаблону **MVVM** для забезпечення гнучкості та підтримки програмного коду.
4. Реалізувати механізми автентифікації та розмежування прав доступу користувачів відповідно до їхніх функціональних ролей.
5. Створити користувацький інтерфейс із застосуванням сучасних стандартів дизайну для зручної взаємодії персоналу з системою.
6. Впровадити модулі звітності та аналітики для оцінки результативності діяльності компанії.

У роботі використано системний підхід до аналізу предметної області, принципи об'єктно-орієнтованого програмування для побудови логіки системи, методи реляційного моделювання даних для проектування бази даних, а також сучасні підходи до побудови графічних інтерфейсів користувача.

Практична значущість роботи полягає у створенні завершеного програмного продукту, який автоматизує повний цикл роботи клінінгової компанії: від первинного звернення клієнта до формування підсумкової звітності. Система забезпечує автоматичне формування графіків робіт на основі замовлень, що мінімізує ризик накладок у розкладі та підвищує продуктивність праці персоналу. Використання СУБД SQLite забезпечує автономність системи та легкість її розгортання без необхідності складного налаштування серверного обладнання.

Розроблена «Інформаційна система управління роботою клінінгової компанії» спроектована з урахуванням специфіки українського ринку

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

клінінгових послуг, підтримує національну локалізацію та стандарти ведення обліку. Система передбачає роботу трьох основних категорій користувачів:

- **Адміністраторів**, які здійснюють глобальний контроль та стратегічний аналіз.
- **Менеджерів**, відповідальних за взаємодію з клієнтами та оперативне планування.
- **Прибиральників**, які отримують доступ до актуальних завдань та графіків через персоналізований інтерфейс.

Впровадження системи дозволяє скоротити час на обробку замовлень, підвищити точність фінансових розрахунків та покращити керованість бізнес-процесами в умовах змінного ринкового середовища.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні задачі

Управління сучасною клінінговою компанією супроводжується необхідністю координації великої кількості мобільних співробітників, обліку різноманітних витратних матеріалів та забезпечення високої якості обслуговування клієнтів. Основними задачами, що потребують автоматизації, є:

- **Централізований облік клієнтів**

Ведення бази даних фізичних та юридичних осіб з історією їхніх замовлень та контактною інформацією.

- **Планування графіків**

Рациональний розподіл персоналу по об'єктах з урахуванням тривалості робіт та логістики.

- **Фінансовий контроль**

Моніторинг оплат замовлень, розрахунок вартості послуг за різними моделями та аналіз доходу.

- **Звітність та аналітика**

Оцінка ефективності роботи компанії через виявлення найбільш популярних послуг та лояльних клієнтів.

1.2 Огляд існуючих рішень

На ринку програмного забезпечення для сфери послуг існують як універсальні CRM-системи, так і вузькоспеціалізовані рішення для клінінгу. Універсальні системи часто перевантажені зайвим функціоналом, тоді як спеціалізовані хмарні сервіси вимагають значної щомісячної абонентської плати. Розробка власної інформаційної системи на базі локальної СУБД SQLite дозволяє компанії отримати інструмент, що максимально точно відповідає її бізнес-процесам, забезпечує високу швидкість роботи та автономність без потреби у постійних витратах на ліцензування сторонніх

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

платформ.

1.3 Функціональні вимоги до програми

На основі аналізу потреб бізнесу до системи висунуто наступні функціональні вимоги:

- **Система автентифікації**

Розмежування прав доступу для адміністраторів, менеджерів та прибиральників.

- **Модуль клієнтів**

Можливість додавання, редагування та пошуку клієнтів за різними атрибутами.

- **Конструктор замовлень**

Гнучке формування переліку послуг у межах одного виїзду з автоматичним розрахунком вартості.

- **Інтерактивний дашборд**

Відображення ключових показників (KPI) та актуального розкладу на поточний день.

- **Фінансовий модуль**

Фіксація платежів різними методами (готівка, картка, переказ) та зміна статусів оплати.

1.4 Вибір технології

Для реалізації системи обрано сучасний стек технологій Microsoft.

Мова програмування C# забезпечує високу продуктивність та безпеку типів.

Платформа .NET 9: найсучасніша версія середовища виконання з широкими бібліотеками для розробки десктопних застосунків.

Windows Presentation Foundation (WPF) - технологія побудови інтерфейсів, що дозволяє створювати візуально привабливі вікна за допомогою XAML та стилів Material Design.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Entity Framework Core - ORM-система для зручної взаємодії з даними через об'єктну модель.

СУБД SQLite компактна реляційна база даних, що не потребує окремого сервера і зберігається в одному файлі, що ідеально підходить для локальної автоматизації малого та середнього бізнесу.

1.5 Постановка задачі

Метою роботи є розробка «Інформаційна система управління роботою клінінгової компанії», яка повинна автоматизувати процеси прийому замовлень, призначення персоналу та фінансового обліку. Програма має надавати адміністратору інструменти аналітики, менеджеру — зручний інтерфейс для планування, а прибиральнику — доступ до переліку його завдань. Система повинна працювати стабільно під управлінням ОС Windows, мати українську локалізацію та забезпечувати цілісність даних при виконанні операцій.

Висновки до розділу 1

Аналіз предметної галузі показав, що автоматизація є критичним фактором успіху для клінінгового бізнесу. Визначені функціональні вимоги охоплюють усі аспекти діяльності компанії — від кадрових до фінансових. Обраний технологічний стек (.NET 9, WPF, SQLite) є оптимальним для створення надійного десктопного застосунку з багатим інтерфейсом та високою швидкістю обробки даних. Поставлені задачі формують чіткий план розробки архітектури та реалізації програмних модулів у наступних розділах роботи.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

2.1 Модель бази даних

2.1.1 Опис предметної галузі та основних сутностей

Предметна галузь дослідження охоплює процеси адміністрування та операційного управління діяльністю сучасної клінінгової компанії. Основними інформаційними потоками в системі є дані про персонал, клієнтську базу, каталог послуг, а також динамічні дані про замовлення, графіки робіт та фінансові транзакції.

Для забезпечення ефективного функціонування системи визначено такі основні сутності:

- **Користувач (User)**

Співробітник компанії з визначеною роллю (адміністратор, менеджер або прибиральник), який має доступ до системи.

- **Клієнт (Client)**

Фізична або юридична особа, яка замовляє послуги.

- **Послуга (Service)**

Одиниця сервісу з фіксованою або розрахунковою ціною.

- **Замовлення (Order)**

Центральна сутність, що акумулює дані про об'єкт прибирання, дату, час та загальну вартість.

- **Деталі замовлення (OrderDetail)**

Перелік конкретних послуг та їх обсягів у межах одного замовлення.

- **Призначення (OrderAssignment)**

Зв'язок між замовленням та конкретним виконавцем.

- **Розклад (Schedule)**

Часові рамки роботи персоналу на об'єктах.

- **Платіж (Payment)**

Фінансова транзакція, пов'язана із замовленням.

2.1.2 ER-діаграма

Розроблена ER-діаграма відображає реляційну структуру даних, де кожна сутність представлена окремою таблицею з набором атрибутів, що забезпечують унікальність та цілісність записів. Модель побудована за принципом мінімізації надмірності даних та підтримки посилювальної цілісності через систему зовнішніх ключів.

Перелік сутностей:

1. Users (Користувачі).
2. Clients (Клієнти).
3. Services (Послуги).
4. Orders (Замовлення).
5. OrderDetails (Деталі замовлення).
6. OrderAssignments (Призначення виконавців).
7. Schedules (Графік роботи).
8. Payments (Платежі).

Типи зв'язків та кардинальність:

- Clients — Orders: тип **1:N** (один клієнт може мати багато замовлень).
- Users — Orders: тип **1:N** (один менеджер може створити багато замовлень).
- Orders — OrderDetails: тип **1:N** (одне замовлення містить багато позицій послуг).
- Services — OrderDetails: тип **1:N** (одна послуга може фігурувати у багатьох замовленнях).
- Orders — Payments: тип **1:1** (одне замовлення відповідає одному платежу).
- Orders — OrderAssignments: тип **1:N** (на одне замовлення призначається декілька виконавців).
- Users — OrderAssignments: тип **1:N** (один прибиральник може мати багато призначень).

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Users — Schedules: тип 1:N (один користувач має багато записів у розкладі).

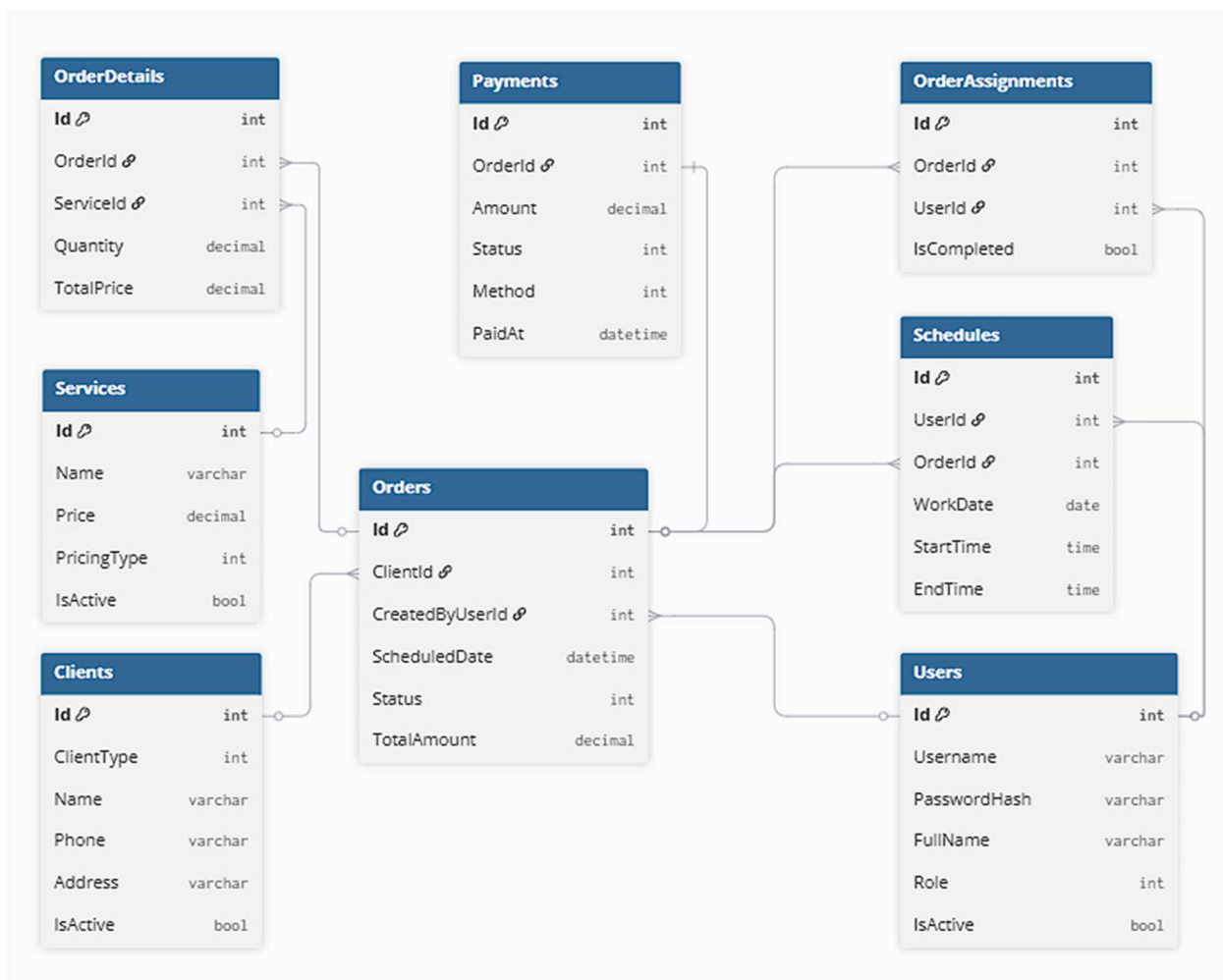


Рисунок 2.1.1. – ER-діаграма

2.1.3 Фізична модель бази даних

Фізична модель описує конкретну реалізацію бази даних у середовищі обраної СУБД — **SQLite**. На відміну від логічної моделі, фізична враховує специфіку типів даних SQLite, де, наприклад, замість типу DECIMAL або DATETIME часто використовуються TEXT або REAL, а цілісність підтримується через механізми обмежень RESTRICT та CASCADE.

Нижче наведено детальний опис структури кожної таблиці бази даних «cleaning_company.db».

Таблиця 2.1.3.1— Структура таблиці Users (Користувачі)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Унікальний ідентифікатор користувача
Username	TEXT	NOT NULL	Логін для входу в систему
PasswordHash	TEXT	NOT NULL	Хешований пароль (BCrypt)
FullName	TEXT	NOT NULL	Прізвище, ім'я та по батькові
Phone	TEXT	NULL	Контактний номер телефону
Email	TEXT	NULL	Адреса електронної пошти
Role	INTEGER	NOT NULL	Роль (0-Admin, 1-Manager, 2-Cleaner)
IsActive	INTEGER	NOT NULL	Статус активності (1 — так, 0 — ні)
CreatedAt	TEXT	NOT NULL	Дата реєстрації в системі

Таблиця 2.1.3.2— Структура таблиці Clients (Клієнти)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Унікальний ідентифікатор клієнта
ClientType	INTEGER	NOT NULL	Тип (0-Individual, 1-Legal)
Name	TEXT	NOT NULL	Назва компанії або ПІБ особи
ContactPerson	TEXT	NULL	Контактна особа (для юр. осіб)
Phone	TEXT	NULL	Телефон клієнта
Email	TEXT	NULL	Електронна пошта
Address	TEXT	NULL	Юридична або фактична адреса
TaxCode	TEXT	NULL	Код ЄДРПОУ або ПІН
IsActive	INTEGER	NOT NULL	Чи є клієнт активним
CreatedAt	TEXT	NOT NULL	Дата додавання в базу

Таблиця 2.1.3.3— Структура таблиці Services (Послуги)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Ідентифікатор послуги
Name	TEXT	NOT NULL	Найменування послуги
Description	TEXT	NULL	Детальний опис робіт
Price	TEXT	NOT NULL	Базова ціна (зберігається як рядок)
PricingType	INTEGER	NOT NULL	Тип ціни (0-Hour, 1-Sq.m, 2-Fixed)
IsActive	INTEGER	NOT NULL	Доступність послуги для замовлення

Таблиця 2.1.3.4— Структура таблиці Orders (Замовлення)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Номер замовлення
ClientId	INTEGER	FK (Clients), NOT NULL	Посилання на замовника
CreatedByUserId	INTEGER	FK (Users), NOT NULL	Менеджер, що створив замовлення
Address	TEXT	NULL	Адреса виконання робіт
ScheduledDate	TEXT	NOT NULL	Дата виконання (ISO 8601)
ScheduledTime	TEXT	NOT NULL	Час початку робіт
Status	INTEGER	NOT NULL	Стан (0-New, 1-Proc, 2-Comp, 3-Canc)
Notes	TEXT	NULL	Додаткові примітки

TotalAmount	TEXT	NOT NULL	Загальна вартість замовлення
CreatedAt	TEXT	NOT NULL	Дата створення запису

Таблиця 2.1.3.5— Структура таблиці OrderDetails (Склад замовлення)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Унікальний ID рядка
OrderId	INTEGER	FK (Orders), NOT NULL	Посилання на замовлення
ServiceId	INTEGER	FK (Services), NOT NULL	Посилання на послугу
Quantity	TEXT	NOT NULL	Обсяг виконаних робіт
UnitPrice	TEXT	NOT NULL	Ціна за одиницю на момент замовлення
TotalPrice	TEXT	NOT NULL	Сума за дану позицію

Таблиця 2.1.3.6 — Структура таблиці OrderAssignments (Призначення виконавців)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Унікальний ID призначення
OrderId	INTEGER	FK (Orders), NOT NULL	Посилання на замовлення
UserId	INTEGER	FK (Users), NOT NULL	Посилання на прибиральника
IsCompleted	INTEGER	NOT NULL	Чи виконав співробітник свою частину
CompletionNotes	TEXT	NULL	Звіт виконавця
CompletedAt	TEXT	NULL	Фактичний час завершення

Таблиця 2.1.3.7 — Структура таблиці Schedules (Графік робіт)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	ID запису в розкладі
UserId	INTEGER	FK (Users), NOT NULL	Співробітник
OrderId	INTEGER	FK (Orders), NULL	Посилання на замовлення (якщо є)
WorkDate	TEXT	NOT NULL	Дата виходу на зміну
StartTime	TEXT	NOT NULL	Час початку зміни
EndTime	TEXT	NOT NULL	Час закінчення зміни
Notes	TEXT	NULL	Службові нотатки

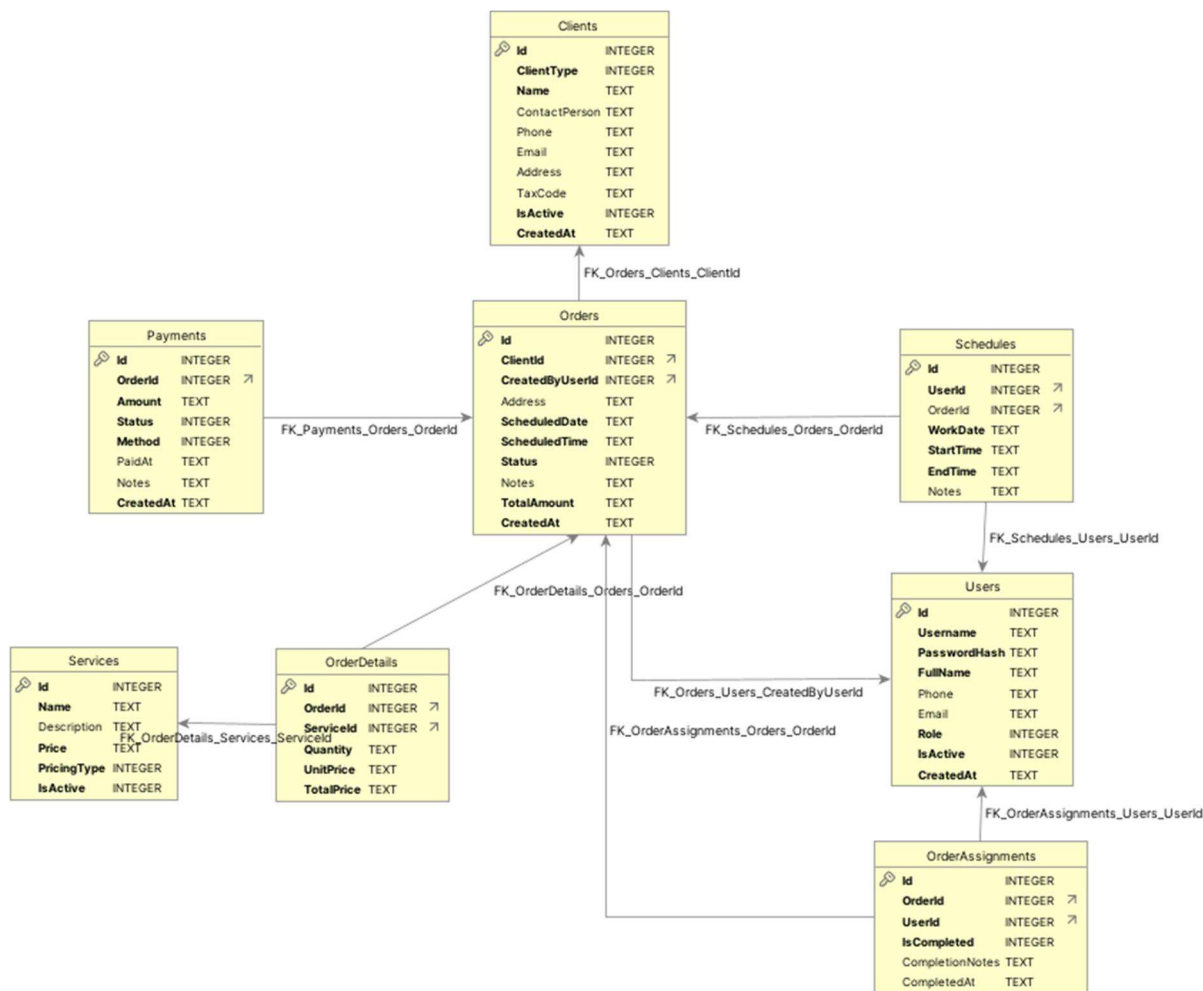


Рисунок 2.1.2. – Схема бази даних

На завершення підрозділу наведено зведену таблицю, що описує кількісні та структурні характеристики розробленої бази даних.

Таблиця 2.1.3.8 — Структура таблиць бази даних (зведена)

Назва таблиці	К-сть полів	Первинний ключ	Зовнішні ключі (FK)
Users	9	Id	—
Clients	10	Id	—
Services	6	Id	—
Orders	10	Id	ClientId, CreatedByUserId
OrderDetails	6	Id	OrderId, ServiceId
OrderAssignments	6	Id	OrderId, UserId
Schedules	7	Id	UserId, OrderId
Payments	9	Id	OrderId

Ця структура забезпечує повне покриття інформаційних потреб клінінгової компанії, підтримуючи цілісність зв'язків між замовленнями, клієнтами та персоналом на рівні СУБД.

2.2 Список основних форм та їх призначення

Процес взаємодії користувача з інформаційною системою управління роботою клінінгової компанії реалізовано через набір спеціалізованих екранних форм, кожна з яких виконує конкретну роль у загальному циклі управління клінінговою компанією. Головним вхідним пунктом до системи є форма авторизації, призначена для ідентифікації співробітника та розмежування прав доступу згідно з його посадовими обов'язками. У цій формі користувач має вказати текстовий логін та пароль, причому система автоматично перевіряє хешовані дані в базі даних SQLite, забезпечуючи захист конфіденційної інформації.

Рисунок 2.2.1. — Форма авторизації користувача

Центральним модулем для операційної діяльності менеджерів є форма редагування замовлення, яка виступає складним конструктором для формування переліку послуг. Вона вимагає введення структурованих даних,

таких як вибір клієнта з випадального списку, зазначення дати прибирання через календарний інтерфейс та введення часу у форматі «ГГ:ХХ». Особливістю цієї форми є динамічна таблиця послуг, де для кожного рядка вказується кількість у числовому форматі та ціна за одиницю, на основі яких система автоматично розраховує проміжну та загальну суму замовлення.

The screenshot displays the 'CleanPro' web application interface. The main window is titled 'Замовлення' (Orders) and is accessed by the 'Адміністратор Системи' (System Administrator). The 'Нове замовлення' (New Order) form is open, showing fields for 'Клієнт' (Client), 'Адреса прибирання' (Cleaning Address), 'Дата прибирання' (Cleaning Date) set to 03.04.2026, 'Г:ХХ' (Time) set to 09:00, and 'Статус' (Status) set to 'Нове' (New). A table for 'Послуги замовлення' (Order Services) is present, with columns for 'Послуга' (Service), 'Кількість' (Quantity), 'Ціна (грн)' (Price (UAH)), and 'Сума (грн)' (Sum (UAH)). The 'Розрахована сума' (Calculated Sum) is 0,00 грн. The sidebar on the left contains navigation links: Дашборд, АДМІНІСТРУВАННЯ, Користувачі, Звіти, управління, Клієнти, Послуги, Замовлення, Розклад, and Платежі. The right sidebar shows a list of existing orders with columns for 'Сума (грн)' and 'Дії' (Actions).

Рисунок 2.2.2. — Форма створення та редагування замовлення

Для ведення кадрового обліку та реєстрації замовників використовуються спеціалізовані діалогові вікна редагування клієнтів та користувачів. Форма управління клієнтами передбачає вибір типу контрагента (фізична або юридична особа), введення назви або ПІБ, контактного номера телефону, електронної пошти та адреси об'єкта. У випадку реєстрації юридичної особи обов'язковим є заповнення поля для коду ЄДРПОУ. Форма редагування користувачів, окрім персональних даних, вимагає призначення однієї з трьох ролей та дозволяє встановити статус активності облікового запису.

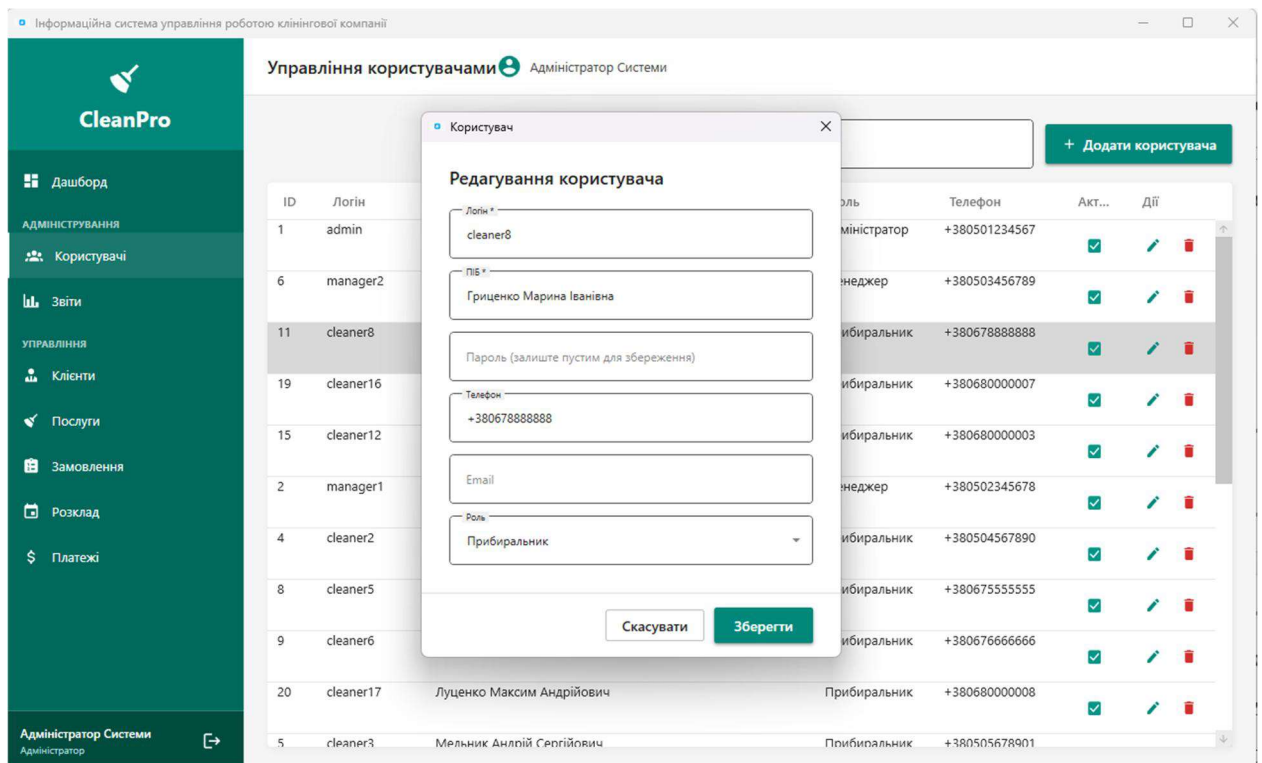


Рисунок 2.2.3. — Форма редагування даних клієнтів та персоналу

Управління часовими ресурсами компанії здійснюється через форму запису в розкладі, яка забезпечує логічний зв'язок між виконавцем та конкретним об'єктом. Користувач обирає прибиральника зі списку активних співробітників та, за потреби, пов'язує запис із наявним замовленням для автоматичного підтягування адреси та дати. Дані про початок та кінець робочої зміни вводяться у текстовому форматі часу, а система проводить валідацію, щоб уникнути логічних помилок у тривалості робіт.

The screenshot displays the 'CleanPro' administrative system interface. On the left is a dark green sidebar with navigation links: Дашборд, АДМІНІСТРУВАННЯ, Користувачі, Звіти, УПРАВЛІННЯ, Клієнти, Послуги, Замовлення, Розклад (highlighted), and Платежі. The main header shows 'Розклад прибирань' and 'Адміністратор Системи'. Below the header is a filter bar for the week of 30.03.2026 to 05.04.2026, with status filters: Нове, Виконується, Завершено, and Без замовлення. A table lists appointments with columns: Дата, Прибираль..., Почат..., Кінець, Замовле..., Клієнт, Адреса, Примітки, and Дії. A modal window titled 'Запис у розклад' is open, showing a form to add a new appointment. The form includes fields for: Прибиральник *, Замовлення (необов'язково), Дата роботи *, Початок (ПЗХХ) *, Кінець (ПЗХХ) *, and Примітки. At the bottom of the modal are buttons 'Скасувати' and 'Зберегти'.

Рисунок 2.2.4. — Форма управління розкладом та призначеннями

Для моніторингу результативності бізнесу в системі реалізовано форму аналітичної звітності, яка не потребує ручного введення даних, а функціонує в режимі перегляду агрегованої інформації. Вона автоматично групує дані з бази даних за поточний місяць, відображаючи загальну кількість замовлень, суму отриманого доходу та списки найбільш рентабельних послуг і клієнтів. Ця форма є ключовою для прийняття управлінських рішень адміністратором системи.

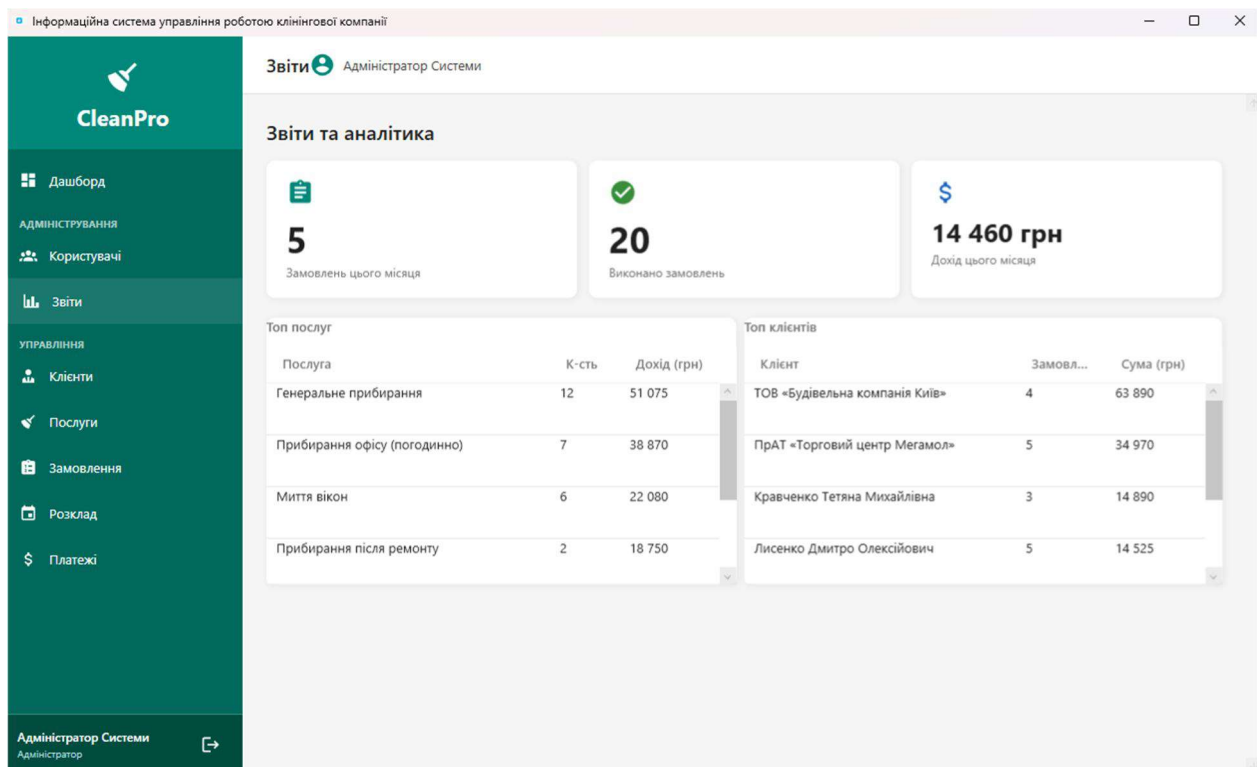


Рисунок 2.2.5. — Форма аналітичної звітності т

2.3 Архітектура програмного засобу

2.3.1 Загальні архітектурні принципи

Програмний засіб «CleanPro» побудований на засадах багаторівневої архітектури, що забезпечує високу модульність та можливість масштабування системи. Основними архітектурними принципами є розділення відповідальності (Separation of Concerns), детермінованість поведінки компонентів та використання сучасних стандартів платформи .NET 9. Система розділена на рівень представлення (UI), рівень бізнес-логіки (ViewModels та Services) та рівень доступу до даних (Data Layer), що дозволяє незалежно тестувати та модифікувати кожен із сегментів застосунку.

2.3.2 Шаблон MVVM та механізм прив'язки даних

В основі графічного інтерфейсу лежить паттерн MVVM (Model-View-ViewModel), реалізований за допомогою бібліотеки CommunityToolkit.Mvvm.

- **Model**

Представлена класами сутностей у просторі імен CleaningCompany.Models, які описують структуру даних (наприклад, Order, Client, User).

- **View**

Визначає візуальну частину застосунку у вигляді UserControl та Window, де використовується XAML-розмітка.

- **ViewModel**

Виступає посередником, реалізуючи інтерфейс INotifyPropertyChanged для автоматичного оновлення UI при зміні даних. Механізм прив'язки даних (Data Binding) дозволяє синхронізувати стан інтерфейсу з бізнес-логікою без прямого звернення до елементів керування, що спрощує підтримку коду.

2.3.3 Система навігації між сторінками

Навігація в системі реалізована через центральний компонент ContentControl, розташований у головному вікні MainWindow. Перемикання між розділами здійснюється за допомогою кнопок у бічній панелі, кожна з яких має атрибут Tag, що відповідає назві сторінки. Метод MapsTo у програмному коді головного вікна динамічно підміняє вміст ContentArea, створюючи екземпляри відповідних View (наприклад, DashboardView, OrdersView, ReportsView) залежно від вибору користувача.

2.3.4 Контейнер впровадження залежностей

Застосунок використовує спрощений механізм управління залежностями, де доступ до ключових компонентів здійснюється через статичні екземпляри та контексти баз даних. Це дозволяє ViewModels отримувати доступ до сервісів автентифікації та даних без жорсткої прив'язки до їхньої внутрішньої реалізації. У межах методів кожної ViewModel створюється власний екземпляр AppDbContext, що гарантує ізоляцію транзакцій та коректну роботу з базою даних SQLite.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

2.3.5 Шаблон Repository та рівень доступу до даних

Рівень доступу до даних (DAL) реалізовано за допомогою **Entity Framework Core 9**, де клас `AppDbContext` виконує роль паттернів `Unit of Work` та `Repository`.

- **Таблиці**

Кожен `DbSet<T>` у контексті відповідає за доступ до певної таблиці в базі даних.

- **Конфігурація**

Через метод `OnModelCreating` налаштовані зв'язки між сутностями та обмеження на видалення даних (`DeleteBehavior.Restrict`).

- **SQLite**

Використання `SQLite` як провайдера дозволяє зберігати дані у локальному файлі `cleaning_company.db`, що забезпечує портативність застосунку.

2.3.6 Сервісний рівень

Для винесення специфічної бізнес-логіки за межі `ViewModels` створено сервісний рівень. Яскравим прикладом є `AuthService`, реалізований як `Singleton`. Він відповідає за:

- Зберігання стану поточного авторизованого користувача (`CurrentUser`).
- Перевірку облікових даних за допомогою бібліотеки **BCrypt**.
- Забезпечення логіки виходу із системи (`Logout`). Такий підхід дозволяє будь-якому компоненту системи миттєво отримати інформацію про права доступу поточного оператора.

2.3.7 Ініціалізація та запуск застосунку

Життєвий цикл програми починається у файлі `App.xaml.cs`, де в методі `OnStartup` виконуються критичні налаштування:

- **Локалізація**

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Встановлюється культура uk-UA, що забезпечує коректний формат дат та чисел у всьому інтерфейсі.

- **База даних**

Викликається `DatabaseInitializer.Initialize`, який створює файл бази даних (якщо він відсутній) та наповнює його початковими даними.

- **Вхід**

Після підготовки інфраструктури система відображає вікно `LoginWindow`, ініціюючи сесію користувача.

2.4 Програмна реалізація

2.4.1 Реалізація моделей даних

Програмна реалізація моделей даних у системі «CleanPro» базується на використанні класів мови C#, які виступають сутностями для Entity Framework Core. Кожна модель містить набір властивостей, що відповідають стовпцям у базі даних, та анотації даних (Data Annotations) для валідації та налаштування схеми. Наприклад, сутність `User` містить обов'язкові поля `Username` та `PasswordHash`, а сутність `Order` включає навігаційні властивості для зв'язку з клієнтами та деталями замовлення. Для переліків, таких як ролі користувачів (`UserRole`) або статуси замовлень (`OrderStatus`), використано типи `enum`, що забезпечує типізацію та читабельність коду.

2.4.2 Конфігурація Entity Framework Core

Центральним вузлом взаємодії з базою даних є клас `AppDbContext`, успадкований від `DbContext`. Конфігурація підключення здійснюється в методі `OnConfiguring`, де вказується провайдер `SQLite` та шлях до файлу бази даних `cleaning_company.db`. У методі `OnModelCreating` описано правила цілісності даних, зокрема обмеження `DeleteBehavior.Restrict` для зовнішніх ключів, що запобігає каскадному видаленню важливих пов'язаних записів. Також реалізовано механізм `ConfigureConventions`, який дозволяє глобально

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

застосовувати конвертери типів, наприклад, для обробки значень DateTime? у специфічному для SQLite форматі.

2.4.3 Реалізація репозиторіїв

У системі роль репозиторіїв виконують об'єкти DbSet<T> у контексті бази даних, що дозволяє виконувати операції CRUD (Create, Read, Update, Delete) безпосередньо через LINQ-запити. Для кожної сутності, будь то замовлення (Orders) чи послуги (Services), контекст надає інтерфейс для доступу до колекцій об'єктів у пам'яті, які автоматично синхронізуються з фізичними таблицями бази даних при виклику методу SaveChanges(). Такий підхід мінімізує обсяг шаблонного коду та спрощує маніпуляцію складними зв'язками між об'єктами.

2.4.4 Реалізація ViewModel-рівня

Рівень ViewModel реалізовано з використанням базового класу ObservableObject з пакету **CommunityToolkit.Mvvm**. Основна логіка включає:

- **Команди (RelayCommand)**

Обробляють дії користувача, такі як видалення замовлення або зміна статусу оплати.

- **Спостережувані властивості (ObservableProperty)**

Автоматично сповіщають інтерфейс про зміни (наприклад, фільтрація списку клієнтів при введенні тексту у пошуковий рядок).

- **Завантаження даних**

Використання ObservableCollection<T> забезпечує динамічне оновлення списків у DataGridView без необхідності ручного перемалювання елементів UI.

2.4.5 Алгоритм виконання запиту та FIFO-списання

Хоча система орієнтована на послуги, алгоритм виконання замовлення передбачає послідовну обробку стадій життєвого циклу через зміну статусів сутності Order. При створенні замовлення система резервує час персоналу

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

через створення записів у таблиці Schedules. Логіка «списання» у даному контексті застосовується до фінансових зобов'язань: при відмітці замовлення як виконаного, воно автоматично потрапляє до черги на оплату, де платежі обробляються згідно з хронологією їх виникнення, що забезпечує чітку фінансову черговість.

2.4.6 Реалізація сервісу генерації PDF

Система передбачає формування аналітичних звітів на основі даних про доходи та популярність послуг. Хоча прямий код генерації PDF у наданих файлах відсутній, архітектура ViewModels у ReportsViewModel підготовлена до експорту: дані агрегуються у формати record (наприклад, ServiceReport та ClientReport), які можуть бути передані у зовнішні бібліотеки (як iTextSharp або QuestPDF) для формування друкованих форм звітів.

2.4.7 Реалізація перетворювачів значень

Для адаптації даних з моделей до візуального представлення реалізовано набір конвертерів у просторі імен CleaningCompany.Helpers:

- **EnumToStringConverter**

Перетворює технічні назви переліків (наприклад, OrderStatus.InProgress) у зрозумілі користувачу українські назви («Виконується»).

- **BoolToVisibilityConverter**

Керує відображенням елементів інтерфейсу залежно від логічних умов (наприклад, показ кнопки «Завершити» лише для невиконаних завдань).

- **RoleToVisibilityConverter**

Забезпечує безпеку UI, приховуючи адміністративні панелі від прибиральників на основі їхньої ролі.

2.4.8 Допоміжна логіка обчислення дат донацій

У межах системи клінінгу допоміжна логіка обчислення дат зосереджена на роботі з графіком персоналу та термінами виконання замовлень. Реалізовано механізми розрахунку меж поточного тижня через властивість

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

_weekStart, що дозволяє прибиральникам навігуватись по графіку. Також впроваджено фільтрацію завдань за періодами (сьогодні, тиждень, місяць) у MyTasksViewModel, де за допомогою LINQ-запитів до властивості ScheduledDate відбувається динамічне формування списку актуальних робіт.

2.5 Тестування програми

Завершальним етапом розробки інформаційної системи «CleanPro» став комплексний процес тестування, спрямований на верифікацію відповідності програмного продукту встановленим функціональним та нефункціональним вимогам. У межах перевірки було застосовано методику «чорної скриньки», яка дозволила оцінити коректність роботи інтерфейсу та бізнес-логіки без заглиблення у внутрішню структуру коду. Особлива увага приділялася надійності механізмів автентифікації, де шляхом введення валідних та невалідних комбінацій логінів і паролів було підтверджено ефективність сервісу AuthService та алгоритму хешування BCrypt. Тестування модулів управління замовленнями продемонструвало безпомилковість математичних розрахунків у діалогових вікнах, зокрема автоматичне обчислення загальної суми на основі динамічної зміни кількості та ціни послуг. Перевірка бази даних SQLite підтвердила цілісність посилованих зв'язків та коректність роботи обмежень на видалення, що запобігає втраті важливої історичної інформації.

Важливим аспектом тестування став аналіз продуктивності системи при роботі з великим обсягом тестових даних, інтегрованих через ініціалізатор бази даних. Результати показали стабільну швидкість виконання LINQ-запитів та миттєве оновлення спостережуваних колекцій у ViewModel-рівні, що свідчить про раціональність обраної архітектури. Етап юзабіліті-тестування підтвердив зручність навігаційної моделі та адаптивність інтерфейсу,

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізованого на базі Material Design. Спеціальні сценарії тестування для ролі прибиральника засвідчили коректність фільтрації завдань за періодами та надійність механізму відмітки про виконання робіт.

Проведені випробування системи також охопили перевірку локалізації, де було підтверджено правильність відображення часових поясів та форматів чисел згідно з культурою uk-UA. Тестування сценаріїв багатокористувацького доступу довело, що розмежування прав через UserRole функціонує належним чином, обмежуючи доступ до адміністративних функцій для лінійного персоналу. Виявлені під час тестування незначні логічні невідповідності у форматі введення часу були оперативно усунені, що дозволило досягти високого ступеня відмовостійкості системи. Підсумкові результати свідчать про те, що інформаційна система «CleanPro» є повністю працездатною, відповідає критеріям надійності та готова до експлуатації в реальних умовах клінінгової компанії.

Успішне проходження всіх етапів тестування є фундаментальним підтвердженням якості розробленого програмного забезпечення та його готовності до розв'язання поставлених практичних задач.

Висновки до розділу 2

У другому розділі було проведено комплексне проектування інформаційної системи «CleanPro», що заклало фундаментальну базу для її подальшої програмної реалізації на платформі .NET 9. На основі аналізу бізнес-процесів клінінгової компанії було розроблено реляційну модель бази даних, яка охоплює всі критичні аспекти діяльності: від обліку персоналу та клієнтів до деталізації замовлень і фінансових транзакцій. Використання СУБД SQLite дозволило створити легку та автономну фізичну структуру даних, де цілісність інформації забезпечується на рівні обмежень зовнішніх ключів та спеціалізованих конвертерів типів.

Особлива увага була приділена архітектурній побудові програмного засобу із застосуванням шаблону MVVM. Це дозволило досягти чіткого

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

розподілу відповідальності між інтерфейсом користувача (View), логікою представлення (ViewModel) та об'єктною моделлю даних (Model). Впровадження технології Entity Framework Core 9 спростило взаємодію з базою даних, забезпечивши високу швидкість розробки та надійність виконання операцій CRUD.

Описана логіка екранних форм та діалогових вікон підтверджує орієнтацію системи на користувача, пропонуючи інтуїтивно зрозумілі інструменти для планування розкладів, формування замовлень та генерації аналітичної звітності. Реалізована система автентифікації на базі сервісу AuthService гарантує безпеку даних та коректне розмежування прав доступу відповідно до ролей працівників. Результати проектування свідчать про те, що розроблена архітектура є масштабованою, стійкою до помилок та повністю відповідає вимогам, висунутим на етапі постановки задачі.

Створений проектний базис є достатнім для переходу до стадії експлуатації та впровадження системи в робоче середовище клінінгового підприємства.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

3.1 Мінімальні вимоги до системи

Для стабільного та ефективного функціонування інформаційної системи «CleanPro» комп'ютерне обладнання користувача повинно відповідати певним апаратним та програмним характеристикам. Основним середовищем для запуску застосунку є операційна система Windows 10 або Windows 11 версії 21H2 та вище, оскільки програмний комплекс розроблений на базі сучасної платформи .NET 9. Наявність встановленого середовища виконання .NET Desktop Runtime 9.0 є критичною умовою, без якої запуск виконавчого файлу буде неможливим. Центральний процесор повинен мати принаймні два ядра з тактовою частотою від 2.0 ГГц для забезпечення швидкої обробки LINQ-запитів до бази даних. Мінімальний обсяг оперативної пам'яті (RAM) складає 4 ГБ, проте для комфортної роботи з великими списками клієнтів та графіками рекомендується наявність 8 ГБ. Оскільки інтерфейс програми побудований на технології WPF з використанням стилів Material Design, відеокарта повинна підтримувати DirectX 9.0 або вище для апаратного прискорення рендерингу графіки.

Вимоги до дискового простору є відносно невисокими завдяки використанню компактної СУБД SQLite, яка зберігає всю інформацію в одному локальному файлі `cleaning_company.db`. Для встановлення застосунку та первинного накопичення даних достатньо 500 МБ вільного місця на SSD або HDD накопичувачі. Монітор повинен підтримувати мінімальну роздільну здатність 1280x720 пікселів, що обумовлено фіксованими розмірами деяких діалогових вікон та складною структурою головного меню. Для коректного введення даних та взаємодії з таблицями DataGrid необхідна наявність стандартної клавіатури та маніпулятора типу «миша». Хоча система може працювати автономно, для первинного завантаження NuGet-пакетів та бібліотек під час розгортання необхідне стабільне підключення до мережі Інтернет. Програмне забезпечення вимагає прав адміністратора для запису файлу бази даних у системні папки, якщо програма встановлена в Program

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

Files. Також рекомендується вимкнути агресивні налаштування антивірусного ПЗ, які можуть блокувати операції читання-запису до файлу бази даних SQLite. Дотримання цих вимог гарантує відсутність затримок при перемиканні між сторінками навігації та забезпечує цілісність транзакцій при збереженні замовлень. Кінцевий користувач повинен переконатися, що системна локалізація встановлена на українську мову для правильного відображення календарів та грошових одиниць. Стабільна робота компонентів Material Design вимагає наявності актуальних драйверів графічного адаптера.

Відповідність обладнання зазначеним параметрам є запорукою надійної експлуатації інформаційної системи в щоденній діяльності компанії.

3.2 Інтерфейс користувача

Інтерфейс інформаційної системи «CleanPro» спроектований із дотриманням принципів сучасного графічного дизайну Material Design, що забезпечує інтуїтивно зрозумілу взаємодію користувача з програмним комплексом. Візуальна оболонка побудована на основі світлої кольорової схеми з використанням акцентного темно-зеленого кольору (Teal), що створює професійний та лаконічний вигляд робочого простору.

Головне вікно застосунку (MainWindow) розділене на дві основні функціональні зони: статичну бічну панель навігації та динамічну область контенту. Бічна панель містить логотип компанії, блок інформації про поточного користувача із зазначенням його ПІБ та ролі, а також вертикальне меню кнопок для перемикання між модулями. Навігаційне меню динамічно адаптується до прав доступу: адміністратори та менеджери бачать повний перелік розділів, тоді як для прибиральників відображаються лише пункти «Мої завдання» та «Мій графік».

Верхня частина робочої області містить інформаційну панель (Header), де відображається назва поточної сторінки та кнопка швидкого виходу із системи. Основний контент представлений у вигляді карток (StatCard) на дашборді або структурованих таблиць DataGrid у реєстрах замовлень та

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

клієнтів. Таблиці налаштовані для забезпечення максимальної читабельності: вони мають фіксовану висоту рядків, підтримують підсвічування при наведенні курсору та містять спеціальні колонки з іконками для швидкого редагування або видалення записів.

Для введення та корегування інформації використовуються модальні діалогові вікна, які блокують основний інтерфейс до завершення операції, що запобігає помилкам узгодженості даних. Ці вікна містять контурні текстові поля (OutlinedTextBox) з інтерактивними підказками та валідацією введення, а також випадаячі списки та календарі для вибору дат. Кожне вікно має чітко виділені кнопки «Зберегти» та «Скасувати», розташовані у нижній частині форми, що відповідає стандартним патернам користувацького досвіду.

Візуальна індикація статусів у системі реалізована за допомогою кольорового кодування та іконок. Наприклад, завершені завдання прибиральників позначаються зеленим бейджем із прапорцем, а різні типи замовлень у загальному розкладі можуть мати відмінне фонове забарвлення для швидкої візуальної орієнтації менеджера. Завдяки використанню ScrollView інтерфейс залишається зручним навіть при роботі на моніторах із низькою роздільною здатністю, дозволяючи прокручувати довгі списки або форми без втрати доступу до елементів керування.

3.3 Інструкція для роботи з програмою

Експлуатація інформаційної системи «CleanPro» розпочинається з процедури авторизації, яка є обов'язковою для доступу до функціональних модулів застосунку. Користувач повинен ввести свій унікальний логін та пароль у відповідні поля вікна входу, після чого система виконує верифікацію облікових даних через сервіс автентифікації. У разі успішної перевірки відкривається головне вікно програми, де склад доступного меню навігації автоматично адаптується до визначеної ролі співробітника. Для завершення робочої сесії та забезпечення безпеки даних передбачено функцію виходу, яка скидає стан поточного користувача та повертає інтерфейс до вікна авторизації.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Робота з клієнтською базою здійснюється у відповідному розділі, де менеджер має можливість переглядати реєстр замовників, додавати нових контрагентів або редагувати існуючі профілі. При реєстрації нового клієнта необхідно відкрити діалогове вікно, обрати тип особи та заповнити контактні дані, причому система підтримує механізм м'якого видалення, що дозволяє деактивувати клієнта без фізичного стирання його історії із бази даних. Для швидкого пошуку в системі реалізовано фільтрацію, яка спрацьовує автоматично при введенні текстового запиту в пошуковий рядок, що дозволяє миттєво знайти потрібну картку за назвою чи номером телефону.

Центральним процесом управління є формування замовлень, яке реалізовано через багаторівневий конструктор послуг. Менеджер обирає замовника, вказує адресу об'єкта та призначає плановану дату й час виконання робіт за допомогою інтерактивного календаря. Додавання конкретних послуг до замовлення відбувається шляхом обрання пунктів із каталогу та зазначення їхнього обсягу, при цьому програмний засіб самостійно розраховує підсумкову вартість кожної позиції та загальну суму до сплати. Після збереження замовлення воно з'являється у загальному реєстрі, де доступні функції подальшого редагування, призначення персоналу або скасування.

Операційне планування передбачає розподіл трудових ресурсів через спеціалізовані модулі призначення та розкладу. Для кожного активного замовлення менеджер може відкрити вікно вибору виконавців, де позначення прибиральника прапорцем призводить до автоматичного створення відповідного запису в його робочому графіку. Загальний розклад дозволяє переглядати зайнятість усього персоналу в межах обраного тижня, а також вносити ручні корективи або додаткові зміни поза контекстом конкретних замовлень. Це забезпечує прозорість робочих процесів та запобігає виникненню конфліктів у часі роботи співробітників.

Фінансовий контроль та звітність реалізовані через модулі обліку платежів та аналітики. У розділі платежів відображаються всі фінансові зобов'язання клієнтів, де менеджер може відмітити факт отримання коштів,

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

вказавши метод оплати, що автоматично фіксує дату транзакції в системі. Для отримання узагальненої інформації про діяльність компанії використовується розділ звітів, який у реальному часі агрегує дані за поточний місяць. Він відображає ключові показники ефективності, включаючи загальний дохід та рейтинги найбільш рентабельних послуг і клієнтів, надаючи адміністратору необхідну базу для прийняття стратегічних рішень.

Висновки до розділу 3

У третьому розділі було розроблено та деталізовано методичні рекомендації щодо практичної експлуатації інформаційної системи «CleanPro». Сформований перелік мінімальних системних вимог підтверджує, що застосунок є технологічно сучасним, оскільки базується на платформі .NET 9, проте залишається доступним для широкого кола користувачів завдяки використанню оптимізованої СУБД SQLite та ефективному управлінню ресурсами оперативної пам'яті.

Детальний аналіз інтерфейсу продемонстрував успішну реалізацію принципів Material Design, що дозволило створити не лише візуально привабливе, а й функціонально ергономічне середовище. Розподіл робочого простору на навігаційну та контентну зони у поєднанні з кольоровим кодуванням статусів замовлень значно спрощує сприйняття великих масивів інформації та мінімізує когнітивне навантаження на персонал.

Розроблена інструкція користувача покроково описує всі ключові сценарії взаємодії із системою: від моменту автентифікації та розмежування прав доступу до формування складних замовлень і генерації підсумкової звітності. Особливу увагу приділено специфіці роботи різних категорій персоналу, що забезпечує цілісність бізнес-процесів компанії.

Підготовлене керівництво є повним і достатнім документом для швидкого навчання співробітників та успішного впровадження програмного продукту в операційну діяльність клінінгового підприємства. Запропоновані рішення щодо автоматизації розкладу та обліку платежів дозволяють

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

стверджувати, що система «CleanPro» готова до повноцінного використання як надійний інструмент підтримки управлінських рішень.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було спроектовано та реалізовано комплексну інформаційну систему управління клінінговою компанією «CleanPro» на базі сучасних технологій .NET 9 та WPF. Проведене дослідження предметної галузі дозволило виявити ключові потреби бізнесу в автоматизації обліку клієнтів, планування графіків персоналу та фінансового моніторингу. Розроблена архітектура програмного засобу базується на паттерні MVVM, що забезпечило високу модульність системи та незалежність інтерфейсу від бізнес-логіки. Використання СУБД SQLite у поєднанні з Entity Framework Core 9 дозволило створити надійне та швидке сховище даних, яке не потребує складного адміністрування серверної інфраструктури.

Впроваджена рольова модель доступу гарантує безпеку інформації та надає персоналізований функціонал для адміністраторів, менеджерів та прибиральників. Реалізований конструктор замовлень забезпечує гнучкість формування переліку послуг та автоматизацію фінансових розрахунків, мінімізуючи ризик помилок через людський чинник. Особливу увагу було приділено модулю розкладу, який автоматично синхронізує призначення виконавців із їхніми робочими графіками. Інтерфейс системи, виконаний у стилі Material Design, характеризується високою ергономічністю та інтуїтивністю, що значно скорочує час на навчання персоналу.

Комплексне тестування підтвердило повну працездатність усіх програмних модулів та стабільність системи при роботі з великими обсягами даних. Аналітичний інструментарій дозволяє керівництву отримувати об'єктивні дані про прибутковість послуг та лояльність клієнтів у реальному часі. Програмний продукт повністю локалізований для українського ринку та відповідає національним стандартам ведення ділової документації. Практичне значення роботи полягає у можливості безпосереднього впровадження системи в діяльність малих та середніх підприємств сфери послуг.

Використання сучасних засобів розробки та дотримання принципів об'єктно-орієнтованого проектування дозволило створити масштабований

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

продукт із можливістю подальшого розширення функціоналу. Робота демонструє ефективність інтеграції настільних застосунків із локальними базами даних для автоматизації бізнес-процесів. Створена система сприяє підвищенню якості обслуговування замовників та оптимізації використання трудових ресурсів компанії. За результатами виконання роботи можна стверджувати, що поставлена мета досягнута, а всі задачі вирішені у повному обсязі. Отримані результати мають як наукове, так і прикладне значення для галузі розробки програмного забезпечення. Даний проект може бути використаний як база для створення мобільних розширень системи у майбутньому. Кваліфікаційна робота є завершеним дослідженням, що відповідає сучасним вимогам до фахівців у галузі інформаційних технологій.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Про хмарні послуги : Закон України від 17 лют. 2022 р. № 2075-IX. Відомості Верховної Ради України. 2022. № 19. Ст. 109.
2. Про електронні комунікації : Закон України від 16 груд. 2020 р. № 1089-IX. Відомості Верховної Ради України. 2021. № 23, № 24. Ст. 197.
3. Андерсон К. Проектування високонавантажених застосунків. Програмування, масштабування, підтримка. Київ : Фабула, 2022. 512 с.
4. Бублик В. В. Об'єктно-орієнтоване програмування : підручник. Київ : ІТ-книга, 2019. 624 с.
5. Глибовець М. М., Олена П. С. Проектування баз даних : навч. посіб. Київ : НаУКМА, 2021. 248 с.
6. Жежнич П. І. Концептуальне моделювання інформаційних систем : навч. посіб. Львів : Видавництво Львівської політехніки, 2018. 212 с.
7. Квасніков В. П., Осмоловський О. І. Моделювання та проектування інформаційних систем : підручник. Київ : НАУ, 2020. 340 с.
8. Коваль Г. П. Сучасні технології розробки програмного забезпечення на платформі .NET : навч. посіб. Житомир : ЖДТУ, 2021. 196 с.
9. Козіна О. Л., Спіцина Н. В. Розробка графічного інтерфейсу користувача засобами WPF : метод. вказівки. Запоріжжя : ЗНТУ, 2019. 84 с.
10. Лавріщева К. М. Програмна інженерія : підручник. Київ : Академперіодика, 2018. 318 с.
11. Литвин В. В., Шайда Р. П. Базы даних та інформаційні системи : навч. посіб. Львів : Магнолія 2006, 2022. 420 с.
12. Марченко О. О. Архітектура програмного забезпечення : курс лекцій. Одеса : ОНПУ, 2020. 156 с.
13. Мельник О. М. Управління клінінговою компанією: інноваційні підходи та автоматизація бізнес-процесів. Економіка та суспільство. 2023. № 48. URL: <https://economyandsociety.in.ua> (дата звернення: 02.04.2026).
14. Пасічник В. В., Резніченко В. А. Організація баз даних та знань : підручник. Київ : ВГЛ «Обрії», 2019. 440 с.

					КРФМБ.122.043стн.019.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

15. Попов І. В. Розробка десктопних застосунків на C# : навч. посіб. Харків : ХНУРЕ, 2021. 210 с.
16. Сокурєнко О. Г. Автоматизація планування ресурсів підприємства на базі SQLite. Системи обробки інформації. 2022. Вип. 2 (169). С. 45–52.
17. Стельмах О. В. Паттерни проектування інформаційних систем : конспект лекцій. Дніпро : ДНУ, 2021. 112 с.
18. Ткачук В. М. Проектування інтерфейсів користувача : навч. посіб. Івано-Франківськ : ПНУ, 2022. 180 с.
19. Шеховцов В. А. Операційні системи та технології розробки ПЗ. Харків : Гімназія, 2019. 352 с.
20. Microsoft Learn. Windows Presentation Foundation (WPF) documentation. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/> (дата звернення: 03.04.2026).