

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 для автоматизації роботи
реєстратури медичного закладу»**

Виконав здобувач освіти групи П-42
Лабенський Олександр Владиславович

Керівник роботи:
Устименко Людмила Миколаївна

Рецензент:

Зміст

Вступ	6
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	9
1.1 Основні задачі автоматизації роботи реєстратури	9
1.2 Огляд існуючих рішень у сфері автоматизації медичних закладів	11
1.3 Функціональні вимоги до програми	13
1.4 Обґрунтування вибору технологій розробки	15
1.5 Постановка задачі	17
Висновки до розділу 1	19
Розділ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ	21
2.1 Модель бази даних	21
2.2 Розробка інтерфейсу програми	27
2.3 Архітектура програмного засобу	32
2.4 Програмна реалізація	35
2.5 Тестування програми	40
Висновки до розділу 2	41
Розділ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ	43
3.1 Мінімальні вимоги до системи	43
3.2 Інструкція для роботи з програмою	44
Висновки до розділу 3	46
ВИСНОВКИ	48
Перелік літературних джерел	50
Додаток А. Код для роботи з базою даних	53
Додаток Б. Програмний код модулів	54

					КРФМБ.122.042.034.2026.ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Лабенський О.В.			Інформаційна система для автоматизації роботи реєстратури медичного закладу	Літ.	Арк.
Перевішив		Устименко Л.М.					3
Рецензент		.				ЖАТФК	
Н. Контр.							
Затвердив.		Гнатюк О.Ф.					
						Аркушів	60

РЕФЕРАТ

Записка: 60 стор., 13 рис., 7 таблиць, 2 додатки, 25 джерел.

Об'єкт дослідження — процес автоматизації реєстрації пацієнтів та обліку медичних послуг у закладах охорони здоров'я.

Предмет дослідження — методи та засоби розробки інформаційної системи для автоматизації роботи медичної реєстратури.

Мета роботи — розробка та впровадження інформаційної системи, що забезпечує ефективне керування даними пацієнтів, розкладом лікарів та генерацію звітності.

У ході кваліфікаційної роботи було проаналізовано предметну галузь та спроектовано реляційну базу даних у СУБД **SQLite**. Програмна реалізація здійснена мовою **C#** на платформі **.NET 9** з використанням архітектурного патерну **MVVM**. Графічний інтерфейс побудовано за допомогою технології **WPF**, що забезпечило високу ергономічність та інтуїтивність системи.

Розроблений програмний засіб включає модулі: «Дашборд» для візуалізації статистики, «Пацієнти» та «Лікарі» для ведення реєстрів, а також систему планування візитів з алгоритмом перевірки часових колізій. Реалізовано функціонал експорту звітів у форматі **PDF**. Тестування підтвердило стабільність роботи системи та відповідність технічним вимогам. Практичне впровадження системи дозволяє оптимізувати роботу персоналу, скоротити час очікування пацієнтів та цифровізувати документообіг медичної установи.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, МЕДИЧНА РЕЄСТРАТУРА, .NET 9, C#, WPF, MVVM, SQLITE, АВТОМАТИЗАЦІЯ, БАЗА ДАНИХ, PDF-ЗВІТНІСТЬ.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

.NET 9 – кросплатформова відкрита розробницька платформа від компанії Microsoft для створення сучасних програмних засобів;

API (Application Programming Interface) – набір визначень протоколів, за допомогою яких різні програмні компоненти взаємодіють між собою;

CSV (Comma-Separated Values) – текстовий формат, призначений для представлення табличних даних;

EF Core (Entity Framework Core) – об'єктно-реляційний засіб відображення (ORM) для .NET, що дозволяє розробникам працювати з базами даних за допомогою об'єктів .NET;

ER-діаграма (Entity-Relationship Diagram) – графічна модель, що описує концептуальну схему бази даних та зв'язки між її сутностями;

GUI (Graphical User Interface) – графічний інтерфейс користувача;

LINQ (Language Integrated Query) – проект Microsoft, який додає синтаксис запитів, подібний до мови SQL, у мови програмування .NET;

MVVM (Model-View-ViewModel) – архітектурний шаблон проектування програмного забезпечення, що відокремлює логіку розробки графічного інтерфейсу від бізнес-логіки;

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

PDF (Portable Document Format) – кросплатформовий формат електронних документів, розроблений компанією Adobe Systems;

RelayCommand – шаблон проектування команд у WPF, що дозволяє делегувати логіку команди методам у ViewModel;

SQLite – вбудована реляційна система управління базами даних, що зберігає інформацію в одному локальному файлі;

SQL (Structured Query Language) – декларативна мова програмування для взаємодії з реляційними базами даних;

UI (User Interface) – інтерфейс користувача;

					КРФМБ.122.042.034.2026.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Процеси цифровізації та модернізації системи охорони здоров'я в Україні висувають нові вимоги до ефективності управління медичними закладами. Одним із критично важливих етапів взаємодії пацієнта з медичною установою є робота реєстратури. Традиційні методи ведення обліку, що базуються на паперових носіях або розрізних електронних таблицях, уже не відповідають сучасним стандартам швидкості та якості обслуговування. Вони призводять до виникнення черг, помилок при записі на прийом, дублювання даних та складнощів у формуванні оперативної звітності.

Автоматизація роботи реєстратури дозволяє не лише оптимізувати робочий час персоналу, а й підвищити рівень задоволеності пацієнтів через прозорість графіків лікарів та швидкість реєстрації. Розробка спеціалізованої інформаційної системи, що інтегрує функції обліку пацієнтів, керування кадровим складом лікарів та планування візитів, є актуальним завданням, спрямованим на створення єдиного інформаційного простору медичного закладу. Використання сучасних технологій, таких як платформа .NET 9 та фреймворк WPF, дозволяє створити надійний та ергономічний програмний продукт, адаптований до потреб вітчизняної медицини.

Об'єкт дослідження – процеси інформаційного забезпечення та операційної діяльності реєстратури сучасного медичного закладу.

Предмет дослідження – методи, моделі та програмні засоби автоматизації реєстрації пацієнтів, формування графіків роботи лікарів та управління процесом запису на прийом.

Мета роботи – проектування та розробка архітектури і програмного забезпечення інформаційної системи для автоматизації роботи реєстратури медичного закладу, що забезпечує зберігання даних, мінімізацію помилок при плануванні візитів та формування аналітичної звітності.

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

1. Проаналізувати предметну область та визначити функціональні вимоги до системи автоматизації реєстратури.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

2. Спроектувати структуру реляційної бази даних для зберігання інформації про пацієнтів, лікарів, користувачів системи та їхні взаємодії.
3. Розробити архітектуру програмного продукту на засадах шаблону MVVM (Model-View-ViewModel) для забезпечення гнучкості та тестованості коду.
4. Реалізувати механізми автентифікації та розмежування прав доступу користувачів.
5. Розробити програмний алгоритм перевірки доступності часових слотів лікарів для уникнення накладок (колізій) при записі на прийом.
6. Впровадити функціонал генерації та експорту звітів у форматі PDF для аналізу завантаженості персоналу.
7. Провести тестування розробленої системи на предмет коректності обробки даних та зручності користувацького інтерфейсу.

У роботі використано системний аналіз для вивчення структури медичного закладу, методи об'єктно-орієнтованого проектування та моделювання (UML) для створення архітектури системи, а також методи теорії розкладів для оптимізації процесу запису на прийом. Програмна реалізація базується на використанні мови програмування C# та платформи .NET 9.

Наукова новизна полягає в удосконаленні алгоритмів динамічного формування розкладу з урахуванням специфіки медичного прийому та інтеграції засобів валідації вводу даних на рівні View-Model. Практичне значення роботи полягає у створенні готового до впровадження програмного забезпечення, яке дозволяє:

- централізовано зберігати медичні картки та контактні дані пацієнтів у локальній СУБД SQLite;
- оперативно керувати графіками роботи лікарів різних спеціальностей;
- автоматично контролювати зайнятість кабінетів та часових інтервалів;
- отримувати статистичні дані про роботу закладу за обраний період.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Обраний технологічний стек — .NET 9 та WPF — забезпечує високу швидкість роботи інтерфейсу та легкість розгортання системи на робочих станціях під керуванням ОС Windows без необхідності використання складних серверних рішень, що є економічно вигідним для невеликих та середніх медичних центрів. СУБД SQLite забезпечує високу надійність збереження даних при мінімальних вимогах до апаратних ресурсів.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні задачі автоматизації роботи реєстратури

Процес функціонування медичного закладу починається з реєстратури, яка виступає ключовою ланкою між пацієнтом та лікувально-діагностичним процесом. Основним призначенням автоматизації цього підрозділу є перехід від паперового документообігу до інтегрованого цифрового середовища, що дозволяє мінімізувати часові витрати на рутинні операції та підвищити достовірність збереженої інформації.

Виходячи з аналізу діяльності медичних установ, можна виділити наступні основні групи задач, які має вирішувати розроблювана інформаційна система:

1. Централізований облік та ведення бази даних пацієнтів.

Ця задача передбачає створення єдиного електронного реєстру, що містить персональні дані пацієнтів, контактну інформацію та історію реєстраційних дій. Ключовим аспектом є забезпечення швидкого пошуку за різними атрибутами (прізвище, номер медичної карти, дата народження) та підтримка актуальності даних у реальному часі. Автоматизація дозволяє уникнути дублювання записів та забезпечує цілісність інформації.

2. Управління ресурсами медичного закладу.

Система повинна акумулювати дані про кадровий склад: кваліфікацію лікарів, їх спеціалізацію та закріплені за ними кабінети. Важливою складовою є формування гнучких графіків роботи, які враховують робочі зміни, перерви та можливі відсутності фахівців. Це створює фундамент для коректного планування візитів.

3. Оптимізація процесу запису на прийом та планування візитів.

Найбільш критична задача, що потребує автоматизованого контролю. Система має забезпечувати:

- візуалізацію доступних часових інтервалів для кожного лікаря;
- автоматичну перевірку накладок (колізій) у розкладі;

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

- можливість оперативного редагування або скасування запису. Використання алгоритмів перевірки доступності слотів дозволяє виключити помилки «подвійного запису», які часто виникають при ручному веденні журналів.

4. Контроль та аналіз завантаженості персоналу.

Автоматизація дає можливість отримувати об'єктивні статистичні дані про кількість прийнятих пацієнтів, популярність окремих спеціалістів та середній час очікування візиту. На основі цих даних адміністрація закладу може приймати обґрунтовані управлінські рішення щодо розширення штату або зміни режиму роботи кабінетів.

5. Забезпечення звітності та документообігу.

Формування оперативної звітності (денні реєстри візитів, тижневі звіти завантаженості) у зручних форматах (PDF, CSV) є необхідним для внутрішнього аудиту та звітності перед державними органами охорони здоров'я.

6. Захист персональних даних та розмежування доступу.

Оскільки система оперує конфіденційною інформацією (згідно із Законом України «Про захист персональних даних»), пріоритетною задачею є впровадження механізмів автентифікації. Доступ до редагування даних пацієнтів або розкладу лікарів повинен надаватися лише авторизованим співробітникам (реєстраторам, адміністраторам) з чітким розмежуванням прав.

Впровадження інформаційної системи на базі технологій .NET 9 та WPF дозволить трансформувати роботу реєстратури з пасивного накопичувача паперової інформації в активний інструмент оперативного управління, що безпосередньо впливає на пропускну здатність медичного закладу та якість надання медичних послуг. Вирішення окреслених задач забезпечить стабільну роботу системи в умовах щоденних високих навантажень.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1.2 Огляд існуючих рішень у сфері автоматизації медичних закладів

На сучасному ринку програмного забезпечення для медицини представлено широкий спектр рішень: від масштабних хмарних екосистем до вузькоспеціалізованих локальних модулів. Проведемо аналіз найбільш розповсюджених систем, що функціонують в Україні, з метою виявлення їхніх переваг та недоліків у контексті автоматизації реєстратури.

1.2.1 Хмарні медичні інформаційні системи (Helsi, Health24, Doctor Eleks).

Ці системи є лідерами українського ринку завдяки інтеграції з центральною базою даних eHealth.

- **Переваги**

Повна відповідність вимогам НСЗУ, можливість онлайн-запису пацієнтів через вебпортали, автоматичне формування електронних рецептів та направлень.

- **Недоліки**

Висока вартість передплати для приватних закладів, повна залежність від стабільності інтернет-з'єднання та віддалених серверів. Для невеликих закладів функціонал таких систем часто є надлишковим, а інтерфейс — занадто складним для швидкого освоєння персоналом реєстратури.

1.2.2 Локальні ERP та CRM-системи (Bitrix24, 1С:Медицина).

Універсальні системи, адаптовані під потреби медичного бізнесу.

- **Переваги**

Гнучкість налаштувань бізнес-процесів, потужні модулі аналітики та фінансового обліку.

- **Недоліки**

Складність впровадження та підтримки, високі вимоги до апаратного забезпечення. Такі рішення часто потребують залучення сторонніх фахівців для будь-якої зміни в логіці роботи розкладу чи структури бази даних.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1.2.3 Спеціалізовані десктопні рішення (Custom Desktop Solutions).

Програми, розроблені під конкретні потреби окремої клініки чи діагностичного центру.

- **Переваги**

Висока швидкість роботи інтерфейсу (WPF), автономність (робота в локальній мережі без обов'язкового доступу до Інтернету), можливість реалізації специфічних алгоритмів перевірки зайнятості лікарів та унікальних форм звітності.

- **Недоліки**

Обмежена мобільність (прив'язка до робочої станції) та необхідність самостійної підтримки бази даних.

1.2.4 Порівняльна характеристика існуючих підходів

Для наочності порівняємо основні типи систем за ключовими критеріями, що важливі для автоматизації реєстратури.

Таблиця 1.2.4.1 – Порівняльна таблиці

Критерій порівняння	Хмарні МІС	Універсальні CRM	Розроблювана система (.NET/WPF)
Автономність	Низька (потрібен Internet)	Середня	Висока (локальна БД SQLite)
Швидкість інтерфейсу	Залежить від браузера	Середня	Висока (Native UI)
Вартість володіння	Щомісячна абонплата	Висока (ліцензія + впровадження)	Низька (одноразова розробка)
Складність навчання	Середня/Висока	Висока	Низька (адаптований UI)
Безпека даних	На сторонніх серверах	В хмарі або локально	Локально у закладі

Аналіз існуючих рішень показав, що для багатьох медичних закладів (особливо приватної форми власності або вузькоспеціалізованих центрів) критично важливими є швидкість відгуку інтерфейсу та незалежність від зовнішніх факторів.

Використання платформи .NET 9 та технології WPF дозволяє створити програмний продукт, який поєднує в собі:

1. WPF забезпечує апаратне прискорення графіки, що робить роботу з великими сітками розкладу плавною та швидкою.

2. Використання SQLite як вбудованої СУБД дозволяє системі працювати без встановлення окремого сервера баз даних, що ідеально підходить для невеликих мереж або окремих робочих місць реєстраторів.
3. Можливість створення повністю україномовного, інтуїтивно зрозумілого інтерфейсу за шаблоном MVVM спрощує процес адаптації нових співробітників.

Незважаючи на наявність потужних хмарних рішень, розробка спеціалізованої інформаційної системи на базі сучасного стеку .NET є виправданою та доцільною. Це дозволяє отримати швидкий, автономний та економічно ефективний інструмент, який максимально точно відповідає внутрішнім регламентам роботи конкретної реєстратури та забезпечує високий рівень захисту даних пацієнтів у межах локальної інфраструктури закладу.

1.3 Функціональні вимоги до програми

Функціональні вимоги визначають поведінку системи та перелік операцій, які користувач може виконувати для досягнення поставлених цілей. Для забезпечення комплексної автоматизації реєстратури, розроблювана система повинна відповідати наступним вимогам, розділеним за основними модулями:

1. Модуль керування доступом

- Система має забезпечувати вхід користувачів за допомогою унікального логіна та пароля.
- Підтримка як мінімум двох ролей користувачів — «Реєстратор» (з правом редагування пацієнтів та записів) та «Адміністратор» (з повним доступом до керування обліковими записами та видалення даних).
- Забезпечення коректного завершення роботи (вихід із системи) для запобігання несанкціонованому доступу сторонніх осіб.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

2. Модуль ведення обліку пацієнтів

- Можливість додавання нового пацієнта з фіксацією персональних даних (ПІБ, дата народження, стать), контактів (телефон, email) та унікального номера медичної карти.
- Редагування існуючих записів та архівація (видалення) застарілих або помилкових даних.
- Реалізація фільтрації списку пацієнтів за прізвищем, номером телефону або номером карти в режимі реального часу.
- Автоматична перевірка коректності введення номерів телефонів та формату електронної пошти на етапі збереження.

3. Модуль управління кадровим складом лікарів

- Ведення бази даних лікарів із зазначенням їхньої спеціалізації, номера кабінету та кваліфікаційної категорії.
- Формування та відображення робочих годин лікаря, що є основою для подальшого планування візитів.

4. Модуль планування візитів

- Можливість вибору пацієнта, лікаря та конкретного часового інтервалу для прийому.
- Система повинна блокувати спроби запису декількох пацієнтів на один і той самий час до одного лікаря.
- Відображення актуального стану запису (призначено, скасовано, виконано) з можливістю оперативного редагування причини звернення.
- Перегляд списку призначень на конкретну дату або за обраним лікарем.

5. Модуль звітності та аналітики

- Формування списків візитів за день, тиждень або місяць.
- Обчислення загальної кількості пацієнтів та показників завантаженості конкретних лікарів.
- Можливість збереження сформованих звітів у форматі PDF для подальшого друку або надсилання керівництву.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

6. Системні та інтерфейсні вимоги

- Повна підтримка української мови в усіх елементах інтерфейсу, повідомленнях про помилки та друкованих формах.
- Коректне функціонування бази даних SQLite у локальному режимі без необхідності підключення до глобальної мережі.
- Інформування користувача про некоректні дії (наприклад, спроба видалення пацієнта, який має активні записи на прийом) зрозумілою мовою.

Чітке дотримання визначених функціональних вимог дозволить розробити програмний продукт, який повністю покриває потреби реєстратури медичного закладу, мінімізує вплив людського фактору на процес планування та забезпечує високу швидкість обробки інформації. Реалізація цих вимог на базі .NET 9 гарантує відповідність системи сучасним стандартам розробки ПЗ.

1.4 Обґрунтування вибору технологій розробки

Для реалізації інформаційної системи автоматизації реєстратури було обрано сучасні інструменти розробки, що дозволяють забезпечити високу продуктивність, модульність коду та зручність підтримки.

1.4.1 Платформа .NET 9 та мова програмування C#

Вибір .NET 9 як базової платформи обумовлений її високою швидкістю виконання коду, вдосконаленими механізмами керування пам'яттю та розширеною підтримкою сучасних бібліотек. Мова C# є об'єктно-орієнтованою мовою з суворою типізацією, що мінімізує кількість помилок на етапі розробки.

Наявність потужних засобів LINQ для маніпуляції даними, асинхронне програмування (async/await) для забезпечення відсутності «зависань» інтерфейсу при роботі з БД та широка підтримка з боку спільноти розробників.

1.4.2 Технологія WPF (Windows Presentation Foundation)

Для розробки графічного інтерфейсу обрано фреймворк WPF. На відміну від застарілих Windows Forms, WPF використовує мову розмітки

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

XAML, що дозволяє чітко відокремити логіку програми від її візуального представлення.

- Використання DirectX для рендерингу елементів інтерфейсу забезпечує плавність роботи навіть при відображенні великих обсягів даних у таблицях (DataGrid).
- Можливість створення єдиного стилю для всіх вікон програми, що критично важливо для професійного вигляду медичної системи.
- Прив'язка даних (Data Binding) це ефективний механізм синхронізації між об'єктами в пам'яті та елементами інтерфейсу.

1.4.3 СУБД SQLite та Entity Framework Core

Оскільки система призначена для локального використання в межах одного або кількох робочих місць реєстрації, вибір зупинився на SQLite.

- SQLite це вбудована реляційна СУБД, яка не потребує встановлення окремого сервера. База даних зберігається у вигляді одного файлу, що спрощує резервне копіювання та перенесення системи.
- Entity Framework Core (EF Core) обрано як ORM-систему (Object-Relational Mapping). Вона дозволяє взаємодіяти з базою даних через об'єкти C#, автоматизуючи створення таблиць (механізм Migrations) та виконання SQL-запитів. Це значно скорочує терміни розробки та підвищує безпеку (захист від SQL-ін'єкцій).

1.4.4 Архітектурний патерн MVVM

Розробка ведеться за шаблоном MVVM (Model-View-ViewModel).

- Model описує структуру даних (Пацієнт, Лікар тощо).
- View відповідає за візуалізацію (XAML-файли).
- ViewModel виступає посередником, що містить логіку обробки команд та підготовки даних для відображення. Це рішення дозволяє проводити незалежне тестування логіки та легко вносити зміни в інтерфейс без втручання в основний код програми.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

1.4.5 Додаткові бібліотеки

- QuestPDF/iTextSharp для програмної генерації звітів у форматі PDF, що відповідають державним стандартам медичної документації.
- CommunityToolkit.Mvvm для стандартизації реалізації команд та сповіщень про зміну властивостей об'єктів.

Обраний стек технологій (.NET 9, C#, WPF, SQLite) є оптимальним для вирішення поставлених завдань. Він дозволяє створити автономний, швидкий та масштабований програмний продукт із сучасним користувацьким інтерфейсом, який повністю задовольняє вимоги до інформаційних систем у медичній галузі.

1.5 Постановка задачі

На основі проведеного аналізу предметної галузі, огляду існуючих рішень та визначених функціональних вимог, метою даної кваліфікаційної роботи є проектування та програмна реалізація інформаційної системи для автоматизації роботи реєстратури медичного закладу.

Для досягнення поставленої мети необхідно реалізувати програмний продукт, який вирішує наступний комплекс взаємопов'язаних задач:

1. Проектування архітектури даних

Розробити реляційну схему бази даних у СУБД SQLite, яка включатиме таблиці для зберігання персональних даних пацієнтів, профілів лікарів, розкладу візитів та облікових записів користувачів.

Забезпечити цілісність даних через використання зовнішніх ключів та обмежень унікальності (наприклад, для номерів медичних карток).

2. Реалізація серверної логіки (Backend)

Розробити об'єктну модель системи на мові C# з використанням Entity Framework Core.

Створити механізм автоматичної ініціалізації бази даних та її первинного наповнення (Seeding) при першому запуску програми.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Реалізувати алгоритм валідації часових інтервалів, який виключатиме можливість призначення двох візитів на один час до одного фахівця, враховуючи тривалість прийому.

3. Розробка користувацького інтерфейсу (Frontend)

Спроекувати та реалізувати за допомогою WPF та XAML візуальне середовище, що складається з форми авторизації та головного вікна з модульною навігацією.

Забезпечити повну локалізацію інтерфейсу українською мовою.

Впровадити адаптивну верстку елементів керування для коректного відображення на моніторах з різною роздільною здатністю.

4. Реалізація функціональних модулів

Модуль автентифікації: перевірка прав доступу користувачів.

Модуль «Пацієнти»: функції додавання, редагування та інтерактивного пошуку в реєстрі.

Модуль «Лікарі»: ведення списку персоналу та їх графіків.

Модуль «Призначення»: візуальне представлення календаря візитів та керування записами.

Модуль звітів: програмна генерація PDF-документів із підсумковою статистикою за обраний період.

5. Забезпечення надійності та якості

Впровадити систему обробки виняткових ситуацій (Error Handling) для запобігання аварійному завершенню програми.

Реалізувати логування критичних дій користувачів для можливості відновлення хронології подій у системі.

Кінцевим результатом має стати десктопний застосунок під керуванням ОС Windows 10/11, який функціонує як автономне робоче місце реєстратора, забезпечує високу швидкість обробки звернень пацієнтів та відповідає сучасним стандартам безпеки зберігання медичної інформації.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Висновки до розділу 1

У першому розділі було проведено комплексний аналіз предметної галузі, що дозволило обґрунтувати необхідність розробки та впровадження інформаційної системи для автоматизації роботи реєстратури медичного закладу. За результатами проведеного дослідження можна зробити наступні висновки:

1. Доведено актуальність автоматизації реєстраційних процесів у медичних закладах. Встановлено, що перехід від традиційних методів паперового обліку до цифрових систем дозволяє значно підвищити швидкість обслуговування пацієнтів, мінімізувати кількість помилок у розкладі лікарів та забезпечити надійне зберігання конфіденційної інформації.
2. Проаналізовано основні задачі, які постають перед персоналом реєстратури. Виявлено, що найбільш критичними аспектами є ефективне управління графіками роботи спеціалістів та запобігання конфліктним ситуаціям при плануванні візитів («подвійний запис»). Визначено, що система повинна не лише зберігати дані, а й виконувати функцію інтелектуальної підтримки прийняття рішень на етапі реєстрації звернення.
3. Здійснено порівняльний огляд існуючих рішень на ринку медичних інформаційних систем. Аналіз показав, що попри наявність потужних хмарних сервісів, існує значний попит на локальні десктопні застосунки, які забезпечують автономність, високу швидкість відгуку інтерфейсу та незалежність від стабільності інтернет-з'єднання. Це підтвердило доцільність розробки власного програмного продукту.
4. Обґрунтовано вибір технологічного стеку. Використання платформи .NET 9 та фреймворку WPF визначено як оптимальне рішення для створення сучасного Native UI з апаратним прискоренням. Вибір SQLite як системи управління базами даних обумовлений її легкістю, відсутністю необхідності адміністрування сервера та високою

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

швидкістю роботи у локальних мережах. Архітектурний шаблон MVVM обрано для забезпечення масштабованості та чіткого розподілу обов'язків між компонентами системи.

5. Сформовано постановку задачі, яка включає розробку модулів автентифікації, обліку пацієнтів і лікарів, планування візитів та генерації аналітичної звітності. Визначені функціональні вимоги стали фундаментом для подальшого проектування бази даних та програмної реалізації системи, що розглядаються у наступних розділах роботи.

Таким чином, результати аналізу предметної галузі підтверджують технічну та економічну доцільність розробки, а обрані інструменти дозволяють реалізувати систему, що відповідає сучасним стандартам якості та надійності програмного забезпечення.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

Процес проектування інформаційної системи для автоматизації роботи реєстратури є критично важливим етапом, що визначає ефективність подальшої розробки. У цьому розділі розглядається побудова моделі бази даних, визначення ключових сутностей, їхніх атрибутів та зв'язків, що забезпечують цілісність інформаційних потоків у межах медичного закладу.

2.1 Модель бази даних

Ефективне функціонування системи автоматизації реєстратури залежить від раціонально спроектованої реляційної моделі даних. Використання СУБД SQLite вимагає чіткого структурування таблиць для забезпечення високої швидкості виконання запитів та мінімізації обсягу надлишкової інформації.

2.1.1 Опис предметної галузі та основних сутностей

Предметна галузь охоплює процеси взаємодії між пацієнтами, медичним персоналом та реєстраторами. Основним об'єктом обліку є візит (призначення), який пов'язує пацієнта з конкретним лікарем у визначений час.

Для повноцінного відображення бізнес-процесів реєстратури виділено такі основні сутності:

1. **Користувачі (Users)** — співробітники реєстратури або адміністратори, які мають доступ до системи.
2. **Пацієнти (Patients)** — особи, які звертаються за медичною допомогою та чий дані зберігаються в електронній базі.
3. **Лікарі (Doctors)** — фахівці медичного закладу, які здійснюють прийом.
4. **Призначення (Appointments)** — центральна сутність, що фіксує факт запису пацієнта до лікаря на певну дату та час.

2.1.2 ER-діаграма

Розроблена ER-діаграма відображає логічну структуру бази даних та ієрархію взаємозв'язків між об'єктами системи. Вона базується на реляційному підході, де кожна сутність представлена окремою таблицею, а зв'язки реалізуються через механізм зовнішніх ключів (Foreign Keys).

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

У розробленій моделі представлено чотири ключові сутності. Структура побудована таким чином, щоб забезпечити нормалізацію даних до третьої нормальної форми (3NF), що дозволяє уникнути аномалій оновлення та видалення.

Сутності системи:

- Users (Користувачі)
- Patients (Пацієнти)
- Doctors (Лікарі)
- Appointments (Призначення/Візити)

Типи зв'язків та їх кардинальність:

1. **Patients — Appointments (1:N):** Один пацієнт може мати багато призначень на прийом протягом тривалого часу, проте конкретний запис у таблиці призначень належить лише одному пацієнту.
2. **Doctors — Appointments (1:N):** Один лікар може прийняти багато пацієнтів (мати багато призначень), але кожен конкретний запис про візит закріплений за одним лікарем.

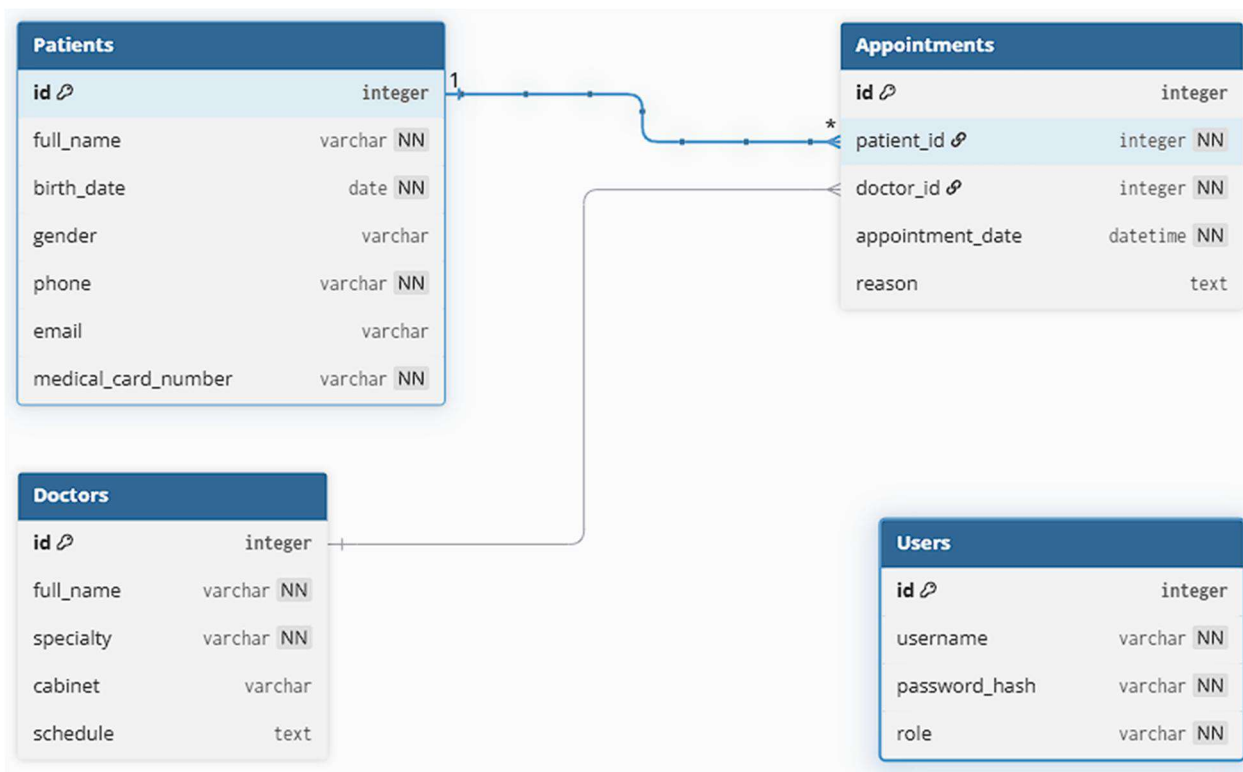


Рисунок 2.1.1. – ER-діаграма

1. Сутності та їх атрибути

Нижче наведено детальний опис кожної сутності з визначенням первинних ключів та типів атрибутів.

Сутність Пацієнти (Patients)

- id (Первинний ключ): Унікальний ідентифікатор запису (простий, однозначний).
- full_name: Прізвище, ім'я та по батькові (складний атрибут, що складається з трьох частин, але зберігається як рядок для спрощення пошуку).
- birth_date: Дата народження пацієнта (простий, однозначний).
- phone: Номер телефону (простий, однозначний).
- medical_card_number: Унікальний номер карти (простий, однозначний).

Сутність Лікарі (Doctors)

- id (Первинний ключ): Унікальний ідентифікатор лікаря (простий).
- full_name: ПІБ лікаря (простий).
- specialty: Спеціалізація (однозначний).
- schedule: Графік роботи (багатозначний атрибут, що у даній моделі представлений текстовим полем для гнучкості опису змін).

Сутність Призначення (Appointments)

- id (Первинний ключ): Унікальний номер візиту.
- appointment_date: Дата та час візиту (складний атрибут).
- reason: Причина звернення (простий).

2. Класифікація атрибутів

Усі атрибути в системі можна класифікувати як **прості** (id, phone, role) та **складні** (appointment_date — містить і дату, і час). Більшість атрибутів є **однозначними**, проте атрибут schedule у таблиці лікарів може розглядатися як **багатозначний**, оскільки містить інформацію про кілька робочих днів.

3. Зв'язки між сутностями

Зв'язок «Запис пацієнта»

- Пов'язані сутності: Patients та Appointments.
- Тип зв'язку: 1:N (Один-до-багатьох).

					КРФМБ.122.042.034.2026.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

- Пояснення: Пацієнт ідентифікується один раз, але може відвідувати лікарню багаторазово.

Зв'язок «Робота лікаря»

- Пов'язані сутності: Doctors та Appointments.
- Тип зв'язку: 1:N (Один-до-багатьох).
- Пояснення: Лікар має власний графік, у межах якого створюється множина записів про прийоми.

Розроблена модель забезпечує цілісність даних на кількох рівнях:

1. Використання первинних ключів (Primary Keys) гарантує унікальність кожного запису.
2. Зовнішні ключі patient_id та doctor_id у таблиці Appointments не дозволяють створити запис про візит для неіснуючого пацієнта або лікаря. При видаленні лікаря або пацієнта система може використовувати каскадне обмеження або блокувати видалення, якщо існують активні записи.
3. Завдяки нормалізації, контактні дані пацієнта зберігаються лише в одній таблиці (Patients). Всі інші модулі (наприклад, звітність) отримують ці дані через зв'язки, що виключає дублювання та суперечливість інформації при оновленні (наприклад, зміні номера телефону).
4. Поля medical_card_number та username мають прапорець Unique, що унеможливорює створення дублікатів медичних карток або однакових логінів користувачів.

2.1.3 Фізична модель бази даних

Фізична модель бази даних визначає конкретну реалізацію логічної схеми засобами обраної СУБД SQLite. На цьому етапі описуються типи даних, обмеження цілісності (Primary Key, Foreign Key, Not Null) та механізми автоматичного оновлення полів. Оскільки SQLite використовує динамічну типізацію, для забезпечення стабільності роботи Entity Framework Core

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

використано чітко визначені типи: INTEGER для ідентифікаторів та TEXT для строкових і часових значень.

Нижче наведено структуру таблиць розробленої бази даних.

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	Хеш-сума пароля користувача
Role	TEXT	NOT NULL	Роль (Адміністратор, Реєстратор)
FullName	TEXT	NOT NULL	ПІБ співробітника
CreatedAt	TEXT	NOT NULL	Дата створення облікового запису

Таблиця 2.1.3.2 – Структура таблиці Patients (Пацієнти)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Унікальний ідентифікатор пацієнта
FirstName	TEXT	NOT NULL	Ім'я
LastName	TEXT	NOT NULL	Прізвище
MiddleName	TEXT	NOT NULL	По батькові
DateOfBirth	TEXT	NOT NULL	Дата народження
Gender	TEXT	NOT NULL	Стать
Address	TEXT	NOT NULL	Адреса проживання
Phone	TEXT	NOT NULL	Контактний телефон
Email	TEXT	NOT NULL	Електронна пошта
MedicalCardNumber	TEXT	NOT NULL	Номер медичної карти (унікальний)
CreatedAt	TEXT	NOT NULL	Дата реєстрації в системі
UpdatedAt	TEXT	NOT NULL	Дата останнього оновлення даних

Таблиця 2.1.3.3 – Структура таблиці Doctors (Лікарі)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Унікальний ідентифікатор лікаря
FirstName	TEXT	NOT NULL	Ім'я лікаря
LastName	TEXT	NOT NULL	Прізвище лікаря
MiddleName	TEXT	NOT NULL	По батькові лікаря
Specialty	TEXT	NOT NULL	Медична спеціалізація
CabinetNumber	TEXT	NOT NULL	Номер кабінету прийому
WorkSchedule	TEXT	NOT NULL	Графік роботи (текстовий опис)
CreatedAt	TEXT	NOT NULL	Дата додавання лікаря до системи

Таблиця 2.1.3.4 – Структура таблиці Appointments (Призначення)

Назва поля	Тип даних	Обмеження	Опис
Id	INTEGER	PK, AI, NOT NULL	Унікальний ідентифікатор візиту
PatientId	INTEGER	FK, NOT NULL	Ідентифікатор пацієнта
DoctorId	INTEGER	FK, NOT NULL	Ідентифікатор лікаря
AppointmentDate	TEXT	NOT NULL	Дата прийому
AppointmentTime	TEXT	NOT NULL	Час прийому
Reason	TEXT	NOT NULL	Причина звернення
Status	TEXT	NOT NULL	Статус візиту (заплановано, скасовано)
CreatedAt	TEXT	NOT NULL	Дата створення запису
UpdatedAt	TEXT	NOT NULL	Дата редагування запису

Для цілісного представлення архітектури даних нижче наведено повну схему взаємозв'язків між таблицями (Рис. 2.1.2.) та зведену специфікацію структури (Таблиця 2.1.3.5).

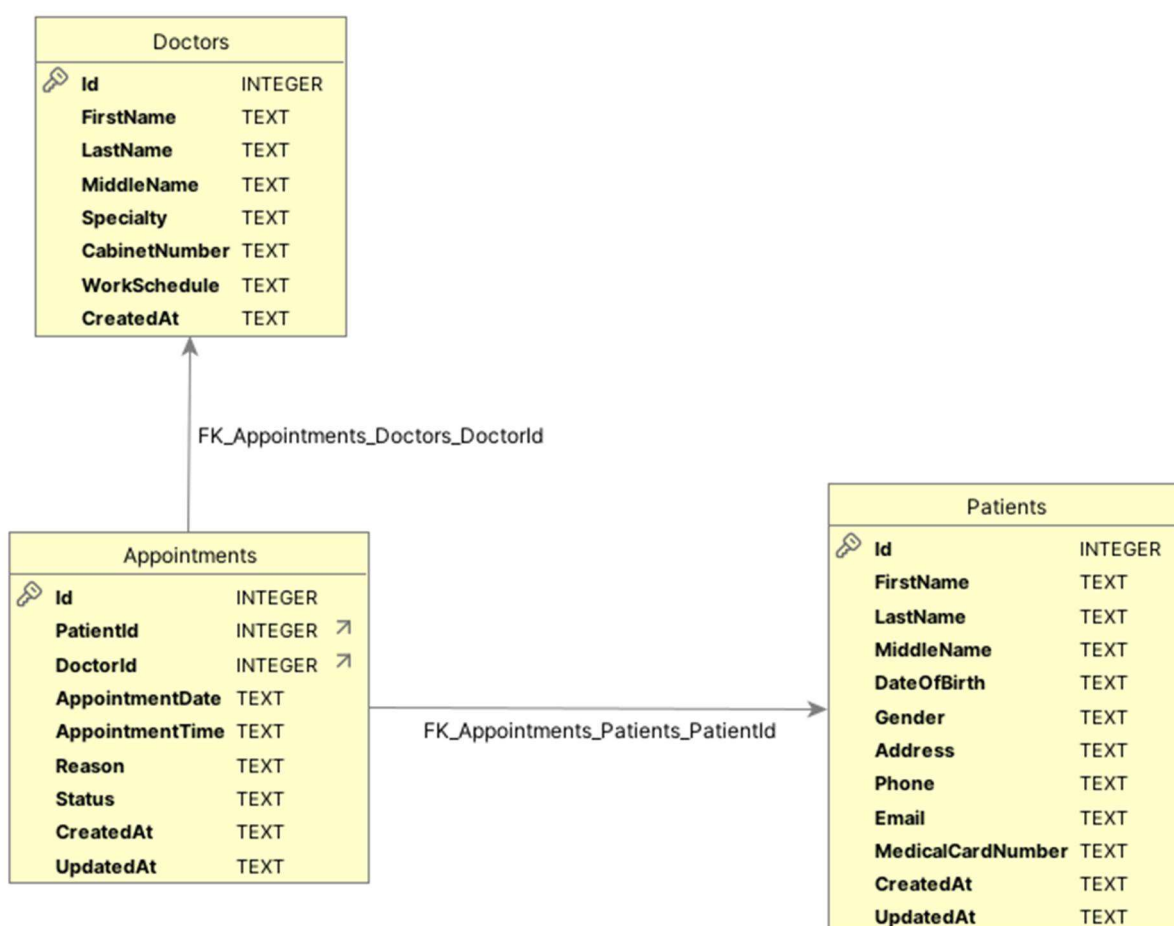


Рисунок 2.1.2. – Схема бази даних

Таблиця 2.1.3.5 – Структура таблиць бази даних

Таблиця	Кількість полів	Первинний ключ	Зовнішні ключі
Users	6	Id	—
Patients	12	Id	—
Doctors	8	Id	—
Appointments	9	Id	PatientId, DoctorId

У розробленій фізичній моделі реалізовано механізм каскадного видалення (ON DELETE CASCADE) для зв'язків у таблиці Appointments. Це забезпечує автоматичне видалення всіх записів про візити у разі видалення профілю пацієнта або лікаря, що запобігає появі «записів-сиріт» та підтримує логічну цілісність бази даних SQLite.

2.2 Розробка інтерфейсу програми

Проектування інтерфейсу користувача інформаційної системи базується на принципах ергономічності, інтуїтивності та функціональної повноти. Використання технології Windows Presentation Foundation дозволяє створити візуальне середовище з чітким розділенням логіки та представлення, що реалізується через набір спеціалізованих вікон та користувацьких представлень.

2.2.1 Вікно автентифікації користувачів

Форма LoginWindow.xaml є вхідною точкою в систему та виконує критичну функцію захисту персональних даних. Графічний інтерфейс вікна спроектований у лаконічному стилі, що фокусує увагу реєстратора на полях введення ідентифікаційних даних. Вікно містить елементи для введення логіна та пароля, причому поле пароля реалізовано за допомогою спеціалізованого компонента PasswordBox, який приховує символи при введенні. Логіка форми передбачає візуальну індикацію помилок у разі введення некоректних даних або відсутності користувача в базі даних. Після успішної перевірки облікових даних через шар View-Model вікно автентифікації закривається, ініціалізуючи запуск основного вікна програми.

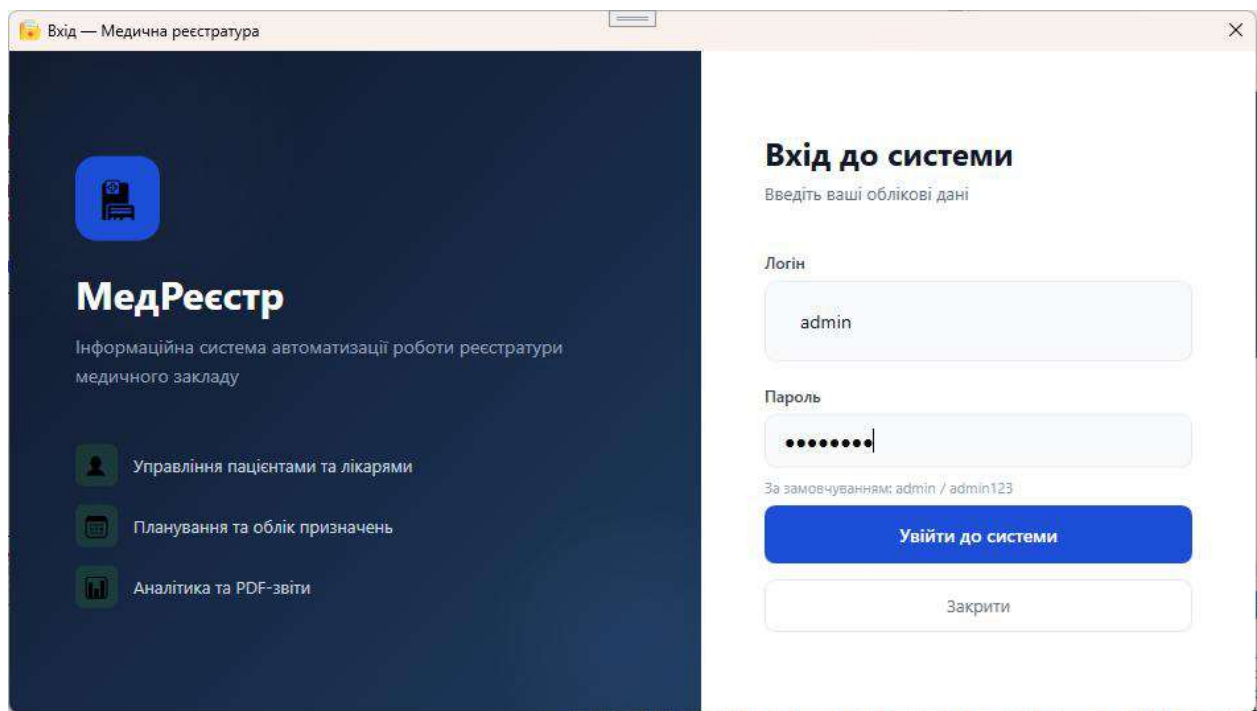


Рисунок 2.2.1. – Форма автентифікації користувачів

2.2.2 Головне вікно та система навігації

Центральним елементом інтерфейсу є MainWindow.xaml, яке реалізує архітектуру Single Page Application у межах десктопного застосунку. Структура вікна розділена на дві основні зони: вертикальне навігаційне меню з лівого боку та динамічну область контенту з правого. Меню містить інтерактивні елементи керування, що відповідають за перемикання між функціональними модулями системи. Використання механізму ContentControl у поєднанні з DataTemplate дозволяє динамічно завантажувати необхідні представлення без перезавантаження всього вікна, що забезпечує високу швидкість відгуку інтерфейсу та збереження контексту роботи реєстратора.

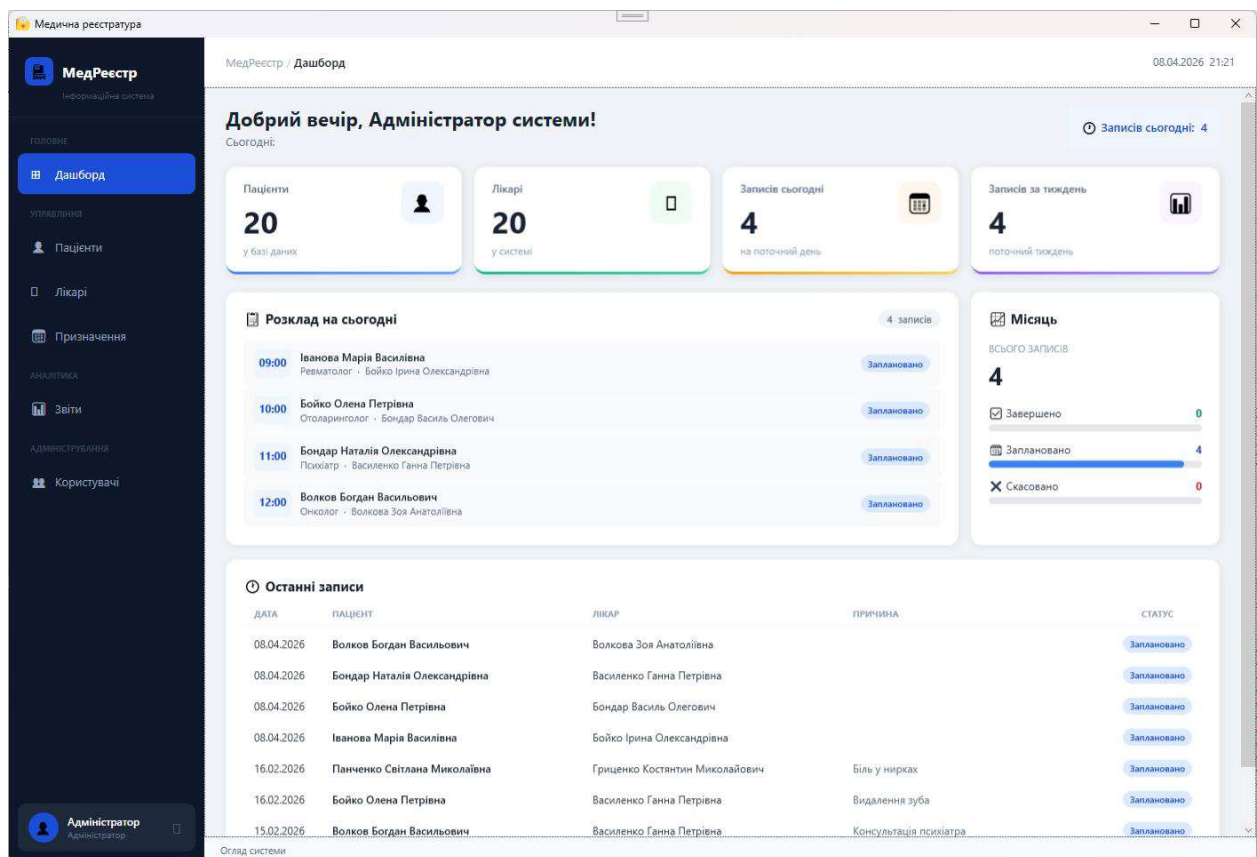


Рисунок 2.2.2. – Головне вікно інформаційної системи

2.2.3 Модуль управління реєстрами пацієнтів та лікарів

Форми PatientManagementView.xaml та DoctorManagementView.xaml мають уніфіковану структуру для полегшення адаптації персоналу. Верхня частина представлень відведена під панель інструментів із полем швидкого пошуку та кнопками додавання нових записів. Центральну частину займає компонент DataGrid, який відображає табличні дані з підтримкою сортування та групування. Для редагування інформації передбачені модальні блоки або окремі поля вводу з налаштованою валідацією. Зокрема, у формі пацієнтів реалізовано маски вводу для номерів телефонів та перевірку унікальності номера медичної карти, а у формі лікарів — спеціалізовані селектори для визначення графіку роботи та кабінету.

МедРегістр

Інформаційна система

Головне

Дашборд

Управління

Пациенти

Лікарі

Призначення

Аналітика

Звіти

Адміністрування

Користувачі

Адміністратор

МедРегістр / Пациенти

08.04.2026 21:21

Список пацієнтів

Пошук за ПІБ, номером карти або телефону...

ПІБ	Дата народж.	Телефон
Іванова Марія Василівна	15.03.1985	+380501234567
Бойко Олена Петрівна	05.07.1981	+380505678902
Бондар Наталія Олександрівна	30.05.1992	+380505678901
Василенко Павло Олександрович	22.02.1968	+380506789013
Волков Богдан Васильович	12.12.2000	+380508901235
Гриценко Зоя Іванівна	30.09.1955	+380507890124
Захарченко Михайло Олегович	25.08.1991	+380500123457
Коваленко Володимир Іванович	14.02.1965	+380504567890
Коваль Костянтин Миколайович	11.11.1993	+380504567891
Кравченко Людмила Іванівна	07.08.1983	+380509012345
Лисенко Сергій Олегович	20.01.1975	+380500123456
Мельник Тетяна Петрівна	08.11.1978	+380503456789
Мороз Ігор Петрович	18.04.1970	+380508901234
Панченко Світлана Миколаївна	17.05.1976	+380509012346
Поліщук Юрій Васильович	15.10.1998	+380502345679
Романова Галина Сергіївна	28.03.1987	+380503456780
Савченко Віра Анатоліївна	03.06.1960	+380501234568

+ Додати

Видалити

Зберегти

Скасувати

Управління пацієнтами

Дані пацієнта

Прізвище *

Бондар

Ім'я *

Наталія

По батькові

Олександрівна

Дата народження *

30.05.1992

Стать *

Жіноча

Номер медичної карти *

000005

Телефон *

+380505678901

Email

bondar.natalia@gmail.com

Адреса

м. Київ, вул. Антоновича, 44, кв. 15

Рисунок 2.2.3. – Управління реєстром пацієнтів

МедРегістр

Інформаційна система

Головне

Дашборд

Управління

Пациенти

Лікарі

Призначення

Аналітика

Звіти

Адміністрування

Користувачі

Адміністратор

МедРегістр / Лікарі

08.04.2026 21:23

Список лікарів

Пошук за ПІБ, спеціальністю або каб.

ПІБ	Спеціальність	Каб.
Бойко Ірина Олександрівна	Ревматолог	207
Бондар Василь Олегович	Отоларинголог	103
Василенко Ганна Петрівна	Психіатр	106
Волкова Зоя Анатоліївна	Онколог	305
Гриценко Костянтин Миколайович	Нефролог	208
Захарченко Світлана Олегівна	Стоматолог	209
Коваленко Олена Олександрівна	Кардіолог	203
Коваль Роман Васильович	Гастроентеролог	105
Кравченко Тетяна Володимирівна	Ендокринолог	205
Лисенко Дмитро Андрійович	Уролог	302
Мельник Марія Василівна	Педіатр	102
Мороз Сергій Павлович	Дерматолог	206
Панченко Богдан Васильович	Травматолог	107
Петренко Петро Петрович	Терапевт	101
Поліщук Юлія Миколаївна	Алерголог	303
Романов Павло Іванович	Пульмонолог	304
Савченко Олександра Іванівна	Гінеколог	104

+ Додати

Видалити

Зберегти

Скасувати

Управління лікарями

Дані лікаря

Прізвище *

Волкова

Ім'я *

Зоя

По батькові

Анатоліївна

Спеціальність *

Онколог

Номер кабінету *

305

Графік роботи

Пн-Чт 8:00-16:00

Рисунок 2.2.4. – Управління реєстром лікарів

2.2.4 Модуль планування візитів

Представлення AppointmentView.xaml є найбільш складним вузлом інтерфейсу, оскільки воно інтегрує дані з декількох таблиць бази даних. Форма забезпечує візуалізацію календаря прийомів та списку поточних призначень. Ключовим елементом є діалоговий блок створення запису, де реєстратор обирає лікаря та дату, після чого система автоматично оновлює список доступних часових інтервалів. Кольорова індикація статусів візитів дозволяє миттєво оцінити завантаженість фахівців та виявити скасовані або прострочені прийоми.

МедРеєстр / Призначення

Дата від: 09.03.2026 15 Дата до: 08.05.2026 15 Лікар: Пациєнт: Статус: Всі Очистити

Дата	Час	Пациєнт	Лікар	Статус
08.04.2026	09:00	Іванова Марія Василівна	Бойко Ірина Олександрівна	Заплановано
08.04.2026	10:00	Бойко Олена Петрівна	Бондар Василь Олегович	Заплановано
08.04.2026	11:00	Бондар Наталія Олександрівна	Василенко Ганна Петрівна	Заплановано
08.04.2026	12:00	Волков Богдан Васильович	Волкова Зоя Анатоліївна	Заплановано

Нові призначення

Пациєнт *
Бойко Олена Петрівна

Лікар *
Бондар Василь Олегович - Отоларинголог (Каб. 103)

Дата *
08.04.2026 15

Час *
10:00
Формат: 09:00 або 14:30

Причина візиту

Статус:
Заплановано

Зберегти + Нове

Видалити

Адміністратор

Управління призначеннями

Рисунок 2.2.5. – Інтерфейс управління призначеннями

2.2.5 Система звітності та адміністрування

Для аналізу діяльності закладу розроблено форму ReportView.xaml, яка містить фільтри за часовими періодами та категоріями лікарів. Інтерфейс дозволяє попередньо переглянути статистичні показники, такі як кількість візитів та навантаження на кабінети, перед виконанням операції експорту в PDF. Паралельно з цим, для користувачів із правами адміністратора доступне

представлення UserManagementView.xaml, де здійснюється керування обліковими записами персоналу, призначення ролей та скидання паролів. Це представлення приховане для рядових реєстраторів, що реалізовано через механізм прив'язки видимості елементів до поточної ролі користувача в системі.

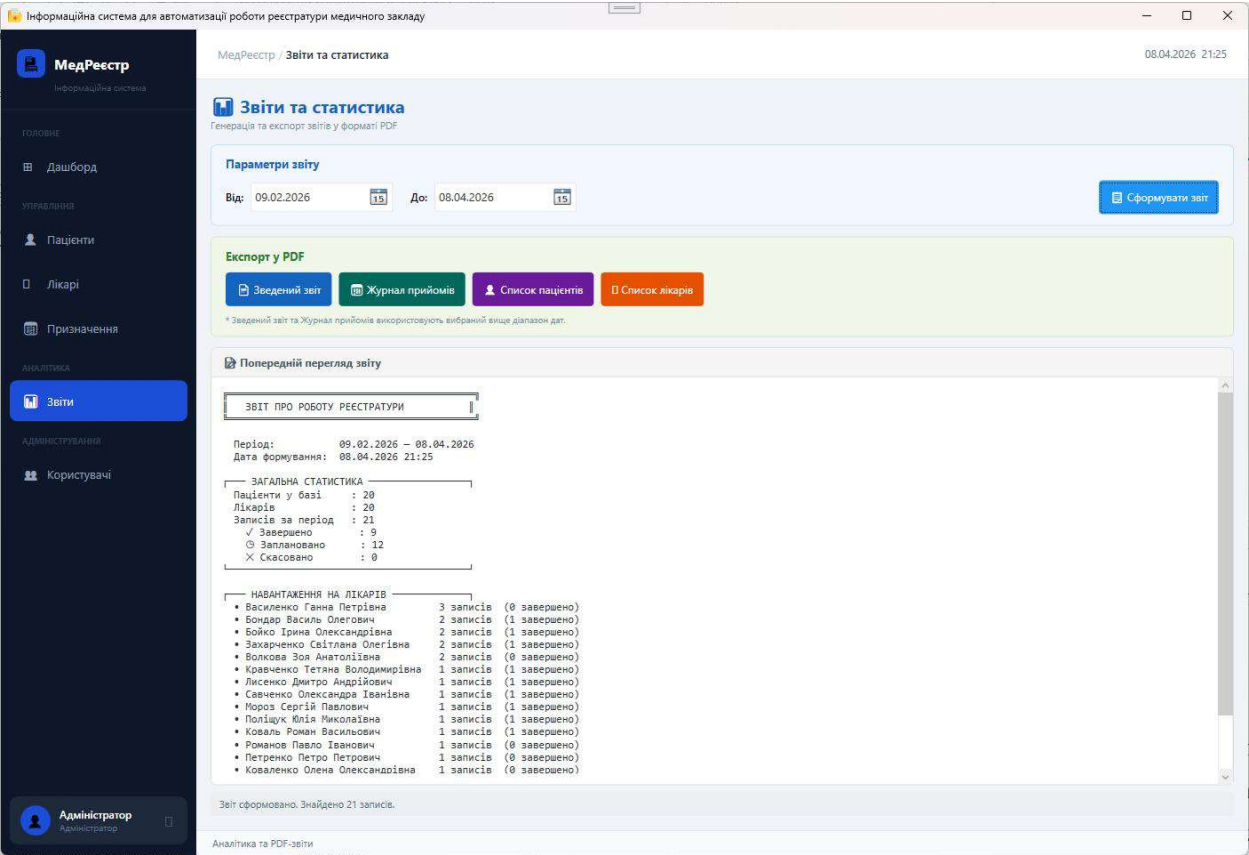


Рисунок 2.2.6. – Модуль генерації звітів та статистики

2.3 Архітектура програмного засобу

Опис архітектури програмного засобу в межах другого розділу фокусується на структурній організації коду та принципах взаємодії між компонентами системи. Для реалізації інформаційної системи реєстратури обрано багаторівневу архітектуру, що базується на патерні проектування MVVM (Model-View-ViewModel), який є галузевим стандартом для розробки сучасних WPF-додатків на платформі .NET 9.

2.3.1 Реалізація патерну MVVM та організація шарів

Архітектурна структура програми розділена на три фундаментальні рівні, кожен з яких виконує чітко визначену роль, що забезпечує низьку зв'язність компонентів та полегшує подальше тестування системи.

Рівень моделі (Model) представлений класами мови C#, які безпосередньо відображають структуру таблиць бази даних SQLite. Ці класи містять лише опис властивостей сутностей, таких як пацієнт, лікар чи візит, та не містять логіки обробки даних. Взаємодія цього шару з базою даних здійснюється через об'єктно-реляційне відображення Entity Framework Core, що дозволяє абстрагуватися від написання прямих SQL-запитів.

Рівень представлення (View) складається з XAML-файлів, описаних у попередньому підрозділі. Головною особливістю цього шару є повна відсутність бізнес-логіки в код-біхайнду (code-behind). Усі елементи інтерфейсу взаємодіють з даними через механізм прив'язки (Data Binding), що дозволяє змінювати візуальну складову системи без втручання в алгоритми її роботи.

Рівень моделі представлення (ViewModel) виступає інтелектуальним посередником. Тут зосереджена вся логіка керування станом інтерфейсу: обробка команд натискання кнопок, фільтрація списків пацієнтів та виклик методів сервісу для перевірки доступності часу лікаря. Кожна ViewModel реалізує інтерфейс INotifyPropertyChanged, що забезпечує автоматичне оновлення елементів інтерфейсу при зміні даних у фоновому режимі.

2.3.2 Структура компонентів та програмні інтерфейси

Програмний засіб організований за модульним принципом, де кожен функціональний блок винесений у відповідний простір назв. Крім основних шарів MVVM, архітектура включає допоміжні сервісні рівні, що розширюють можливості системи.

Сервісний шар (Services) містить логіку, яка не належить безпосередньо до представлення. Сюди відносяться модулі генерації PDF-звітів, сервіс автентифікації, що відповідає за хешування паролів, та сервіс валідації

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

записів. Такий підхід дозволяє використовувати один і той самий алгоритм (наприклад, перевірку конфліктів у розкладі) у різних частинах програми, уникаючи дублювання коду.

Шар доступу до даних (Data Access Layer) базується на класі контексту MedContext, успадкованому від DbContext. Він відповідає за конфігурацію зв'язків між таблицями та керування життєвим циклом підключення до файлу SQLite. Використання паттерна «Репозиторій» у межах цього шару дозволяє інкапсулювати методи вибірки даних, що робить ViewModel незалежною від конкретної реалізації бази даних.

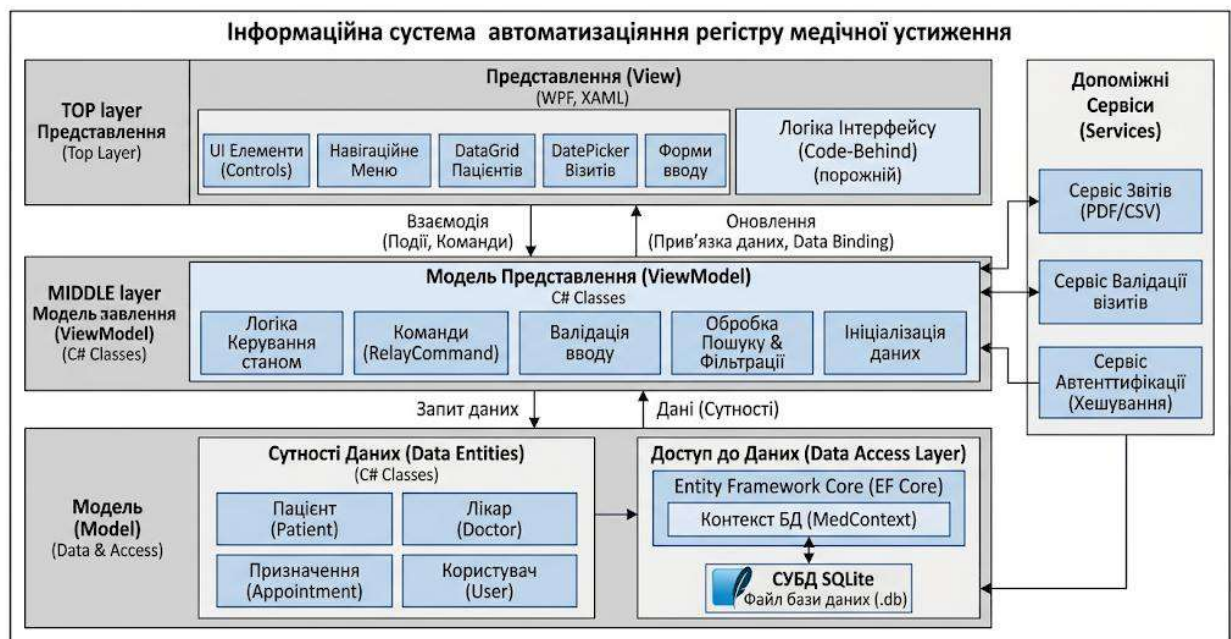


Рисунок 2.3.1. – Схема архітектури програмного засобу за патерном MVVM

2.3.3 Взаємодія компонентів при виконанні операцій

Механізм роботи архітектури можна простежити на прикладі створення нового запису на прийом. Коли реєстратор обирає дату, рівень представлення (View) через прив'язку передає це значення у ViewModel. ViewModel викликає метод сервісу валідації, який через контекст даних (Model) перевіряє наявність колізій у базі даних SQLite. Після отримання результату ViewModel оновлює колекцію доступних слотів, і ці зміни миттєво відображаються у випадяючому списку на формі завдяки механізму сповіщень.

Така архітектурна побудова на базі .NET 9 забезпечує високу продуктивність системи, оскільки важкі операції з базою даних та генерація звітів виконуються в асинхронних потоках, не блокуючи головний потік інтерфейсу. Це створює комфортні умови для роботи персоналу реєстратури навіть при одночасному опрацюванні великих масивів інформації.

2.4 Програмна реалізація

2.4.1 Реалізація шару доступу до даних та ініціалізація БД

Програмна реалізація системи базується на використанні Entity Framework Core 9, що дозволяє працювати з базою даних SQLite як з набором типізованих об'єктів. Центральним компонентом цього шару є клас MedContext, успадкований від DbContext. Він визначає набори сутностей DbSet<T> для кожної таблиці фізичної моделі та конфігурує параметри підключення.

Для забезпечення автономності системи реалізовано механізм автоматичного розгортання бази даних. При старті застосунку викликається метод Database.EnsureCreated(), який перевіряє наявність файлу .db та, у разі його відсутності, створює структуру таблиць згідно з описаними моделями. Також впроваджено початкове наповнення даних (Data Seeding) для створення облікового запису адміністратора за замовчуванням, що дозволяє розпочати роботу з системою відразу після інсталяції.

2.4.2 Реалізація бізнес-логіки та алгоритмів перевірки колізій

Найбільш відповідальною частиною програмної реалізації є сервіс управління записами (AppointmentService). Основним завданням цього модуля є запобігання конфліктам у розкладі лікарів. Алгоритм перевірки базується на логіці перетину часових інтервалів. При спробі створення нового призначення система виконує запит до бази даних, фільтруючи записи за ідентифікатором лікаря та обраною датою.

Критична умова доступності часу реалізована за формулою: новий візит є допустимим лише тоді, коли його час початку не потрапляє в інтервал

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

[T_start; T_start + T_delta] будь-якого існуючого запису, де T_delta — тривалість прийому. Програмно це реалізовано через LINQ-запит, який порівнює об'єкти DateTime. Якщо конфлікт виявлено, система генерує виняткова ситуацію або повертає прапорець помилки, що відображається реєстратору у вигляді інформаційного повідомлення українською мовою.

2.4.3 Впровадження патерну MVVM та механізму команд

Програмна логіка взаємодії з користувачем зосереджена у класах ViewModel. Для реалізації реактивності інтерфейсу використано базовий клас BaseViewModel, який імплементує інтерфейс INotifyPropertyChanged. Це дозволяє системі автоматично оновлювати візуальні елементи, наприклад, перераховувати кількість пацієнтів у списку після видалення або фільтрації, без явного звернення до елементів управління.

Взаємодія з кнопками та меню реалізована через об'єкти RelayCommand. Це дозволяє інкапсулювати дію та умову її виконання (наприклад, кнопка «Зберегти» стає активною лише тоді, коли всі обов'язкові поля пацієнта заповнені та пройшли валідацію). Такий підхід забезпечує чистоту коду та можливість тестування бізнес-логіки окремо від графічної оболонки.

2.4.4 Забезпечення безпеки та валідації вводу

На програмному рівні реалізовано дворівневу систему перевірки даних. Перший рівень — візуальна валідація у WPF за допомогою інтерфейсу IDataErrorInfo, що підсвічує текстові поля червоним кольором при введенні некоректного формату (наприклад, невірна кількість цифр у номері телефону). Другий рівень — сувора валідація на рівні моделей перед відправкою змін у базу даних SQLite.

Для захисту облікових даних користувачів реалізовано сервіс хешування паролів. Паролі не зберігаються у відкритому вигляді; замість цього використовується алгоритм соляного хешування (Salted Hashing), що відповідає сучасним вимогам кібербезпеки для систем, які оперують медичною та персональною інформацією.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

2.4.5 Реалізація модулів звітності та експорту

Для програмної генерації звітів використано Fluent-інтерфейс бібліотеки QuestPDF. Логіка формування звіту полягає у вибірці даних за певний період, їх агрегації та динамічній побудові структури PDF-документа.

Використано чотири типи звіти:

- **Зведений звіт** (A4 Portrait) — картки статистики + таблиця навантаження на лікарів

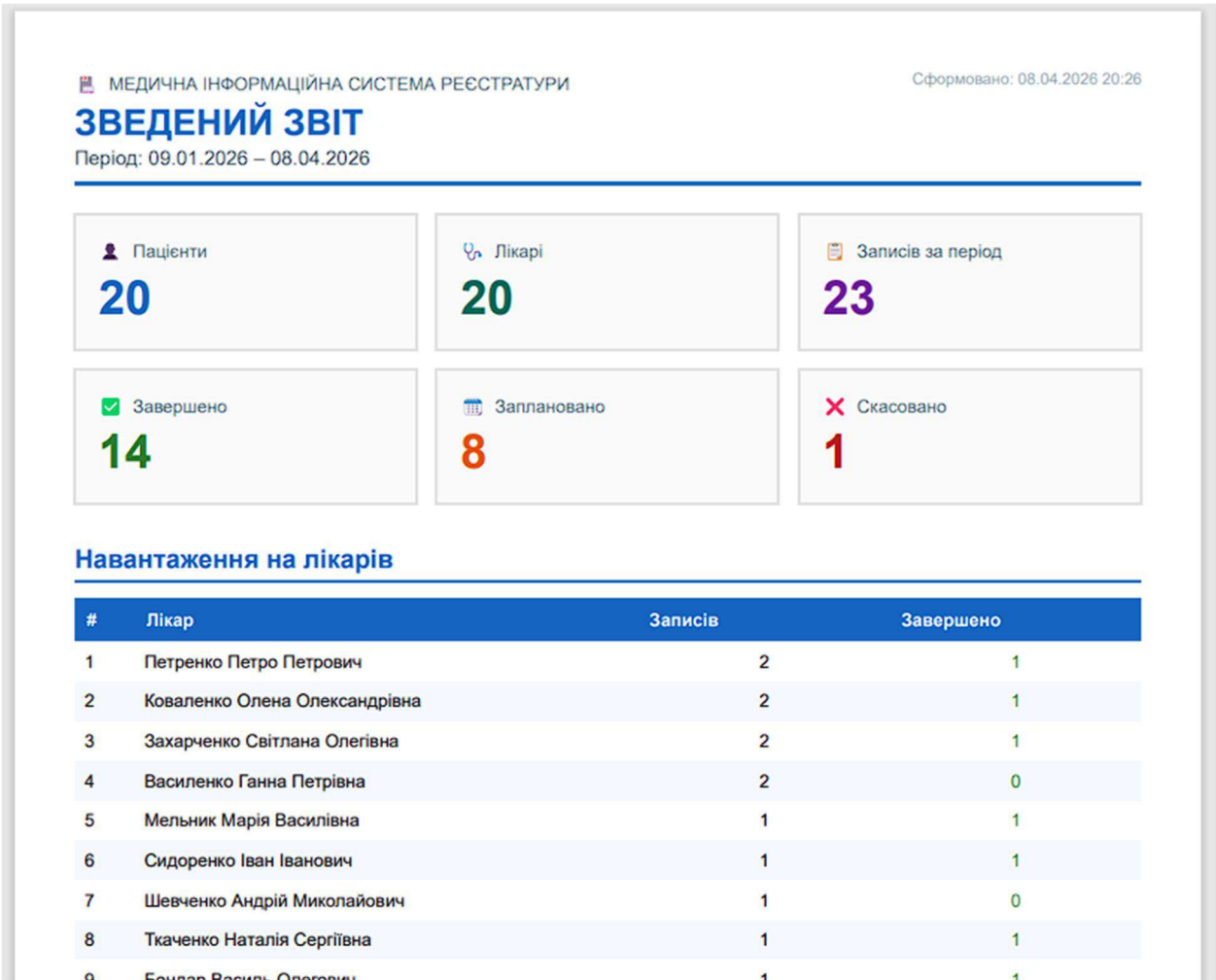


Рисунок 2.4.1. – Звіт «Зведений звіт»

- **Журнал прийомів (A4 Landscape)** — детальна таблиця всіх прийомів з кольоровими статусами

МЕДИЧНА ІНФОРМАЦІЙНА СИСТЕМА РЕЄСТРАТУРИ

ЖУРНАЛ ПРИЙОМІВ

Період: 09.01.2026 – 08.04.2026 | Записів: 23

Сформовано: 08.04.2026 20:26

#	Дата / Час	Пацієнт	Лікар	Причина	Статус
1	07.02.2026 09:00	Іванова Марія Василівна	Петренко Петро Петрович	Профілактичний огляд, скарги на біль у горлі	Завершено
2	07.02.2026 10:00	Шевченко Олексій Миколайович	Коваленко Олена Олександрівна	Консультація кардіолога, біль у грудній клітині	Завершено
3	07.02.2026 11:00	Мельник Тетяна Петрівна	Мельник Марія Василівна	Планова вакцинація дитини	Завершено
4	08.02.2026 09:00	Коваленко Володимир Іванович	Сидоренко Іван Іванович	Консультація щодо операції	Завершено
5	08.02.2026 10:00	Бондар Наталія Олександрівна	Шевченко Андрій Миколайович	Головний біль, запаморочення	Скасовано
6	08.02.2026 11:00	Ткаченко Андрій Васильович	Ткаченко Наталія Сергіївна	Погіршення зору	Завершено
7	09.02.2026 09:00	Сидоренко Оксана Миколаївна	Бондар Василь Олександрович	Біль у вусі, нежить	Завершено
8	09.02.2026 10:00	Мороз Ігор Петрович	Кравченко Тетяна Володимирівна	Контроль рівня цукру в крові	Завершено
9	12.02.2026 09:00	Кравченко Людмила Іванівна	Лисенко Дмитро Андрійович	Біль при сечовипусканні	Завершено
10	12.02.2026 10:00	Лисенко Сергій Олександрович	Савченко Олександра Іванівна	Планова консультація гінеколога	Завершено
11	12.02.2026 11:00	Савченко Віра Анатоліївна	Мороз Сергій Павлович	Висипання на шкірі	Завершено
12	13.02.2026 09:00	Попіщук Юрій Васильович	Попіщук Юлія Миколаївна	Алергічна реакція	Завершено
13	13.02.2026 10:00	Романова Галина Сергіївна	Коваль Роман Васильович	Біль у животі, порушення травлення	Завершено
14	13.02.2026 11:00	Волков Богдан Васильович	Бойко Ірина Олександрівна	Біль у суглобах	Завершено
15	14.02.2026 09:00	Бойко Олена Петрівна	Романов Павло Іванович	Кашель, задишка	Заплановано
16	14.02.2026	Бондар Наталія Олександрівна	Захарченко Світлана Олегівна		Завершено

Медична Реєстратура • Конфіденційна інформація

Сторінка 1 з 2

Рисунок 2.4.2. – Звіт «Журнал прийомів»

- **Список пацієнтів (A4 Landscape)** — повний реєстр з датою народ., віком, контактами

МЕДИЧНА ІНФОРМАЦІЙНА СИСТЕМА РЕЄСТРАТУРИ

СПИСОК ПАЦІЄНТІВ

Всього: 20 | Станом на 08.04.2026

Сформовано: 08.04.2026 18:06

#	Карта №	ПІБ	Дата нар.	Вік	Телефон	Email	Стать
1	000015	Бойко Олена Петрівна	05.07.19 81	44	+380505678902	boyko.olena@gmail.com	Жінка
2	000005	Бондар Наталія Олександрівна	30.05.19 92	33	+380505678901	bondar.natalia@gmail.com	Жінка
3	000016	Василенко Павло Олександрович	22.02.19 68	58	+380506789013	vasylenko.pavlo@ukr.net	Чоловік
4	000018	Волков Богдан Васильович	12.12.20 00	25	+380508901235	volkov.bohdan@ukr.net	Чоловік
5	000017	Гриценко Зоя Іванівна	30.09.19 55	70	+380507890124	hrytsenko.zoia@gmail.com	Жінка
6	000020	Захарченко Михайло Олександрович	25.08.19 91	34	+380500123457	zakharchenko.mykhailo@ukr.net	Чоловік
7	000001	Іванова Марія Василівна	15.03.19 85	41	+380501234567	ivanova.maria@gmail.com	Жінка
8	000004	Коваленко Володимир Іванович	14.02.19 65	61	+380504567890	kovalenko.v@ukr.net	Чоловік
9	000014	Коваль Костянтин Миколайович	11.11.19 93	32	+380504567891	koval.kostiantyn@ukr.net	Чоловік
10	000009	Кравченко Людмила Іванівна	07.08.19 83	42	+380509012345	kравchenko.liuda@gmail.com	Жінка
11	000010	Лисенко Сергій Олександрович	20.01.19 75	51	+380500123456	lysenko.serhiy@ukr.net	Чоловік
12	000003	Мельник Тетяна Петрівна	08.11.19 78	47	+380503456789	melnyk.tetiana@gmail.com	Жінка
13	000008	Мороз Ігор Петрович	18.04.19 70	55	+380508901234	moroz.igor@ukr.net	Чоловік
14	000019	Панченко Світлана Миколаївна	17.05.19 76	49	+380509012346	panchenko.svitlana@gmail.com	Жінка
15	000012	Попіщук Юрій Васильович	15.10.19 98	27	+380502345679	polishchuk.yuriy@ukr.net	Чоловік

Медична Реєстратура • Конфіденційна інформація

Сторінка 1 з 2

Рисунок 2.4.3. – Звіт «Список пацієнтів»

Змн.	Арк.	№ докум.	Підпис	Дата

КРФМБ.122.042.034.2026.ПЗ

Арк.

38

- **Список лікарів** (A4 Portrait) — таблиця зі спеціальностями та розкладом

МЕДИЧНА ІНФОРМАЦІЙНА СИСТЕМА РЕЄСТРАТУРИ

Сформовано: 08.04.2026 18:05

СПИСОК ЛІКАРІВ

Всього: 20 | Станом на 08.04.2026

#	ПІБ	Спеціальність	Кабінет	Розклад
1	Поліщук Юлія Миколаївна	Алерголог	303	Пн-Чт 9:00-17:00
2	Коваль Роман Васильович	Гастроентеролог	105	Пн-Пт 8:30-16:30
3	Савченко Олександра Іванівна	Гінеколог	104	Пн-Пт 8:00-16:00
4	Мороз Сергій Павлович	Дерматолог	206	Пн-Пт 10:00-18:00
5	Кравченко Тетяна Володимирівна	Ендокринолог	205	Пн-Пт 9:30-17:30
6	Коваленко Олена Олександрівна	Кардіолог	203	Пн-Пт 10:00-18:00
7	Шевченко Андрій Миколайович	Невролог	204	Пн-Пт 9:00-17:00
8	Гриценко Костянтин Миколайович	Нефролог	208	Пн-Пт 9:30-17:30
9	Волкова Зоя Анатоліївна	Онколог	305	Пн-Чт 8:00-16:00
10	Бондар Василь Олегович	Отоларинголог	103	Пн-Пт 8:00-16:00
11	Ткаченко Наталія Сергіївна	Офтальмолог	301	Пн-Сб 10:00-18:00
12	Мельник Марія Василівна	Педіатр	102	Пн-Пт 8:30-16:30
13	Василенко Ганна Петрівна	Психіатр	106	Пн-Пт 8:00-16:00
14	Романов Павло Іванович	Пульмонолог	304	Вт-Сб 10:00-18:00
15	Бойко Ірина Олександрівна	Ревматолог	207	Пн-Пт 9:00-17:00
16	Захарченко Світлана Олегівна	Стоматолог	209	Пн-Сб 9:00-19:00
17	Петренко Петро Петрович	Терапевт	101	Пн-Пт 9:00-17:00
18	Панченко Богдан Васильович	Травматолог	107	Цілодобово, чергування
19	Лисенко Дмитро Андрійович	Уролог	302	Вт-Сб 9:00-17:00
20	Сидоренко Іван Іванович	Хірург	305	Вт-Сб 8:00-16:00

Рисунок 2.4.4. – Звіт «Список лікарів»

Процес генерації винесено в окремий потік, що дозволяє реєстратору продовжувати роботу з іншими модулями системи під час підготовки великих звітів.

					<i>КРФМБ.122.042.034.2026.ПЗ</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

2.5 Тестування програми

Процес тестування інформаційної системи для автоматизації роботи реєстратури медичного закладу мав на меті підтвердження коректності роботи всіх модулів та виявлення можливих логічних помилок на етапі розробки. Для перевірки надійності було застосовано методи функціонального тестування, тестування інтерфейсу користувача (UI) та валідацію цілісності бази даних SQLite. Особливу увагу приділено модулю автентифікації, який успішно пройшов перевірку на розмежування прав доступу між адміністраторами та реєстраторами. У ході тестування модуля «Пацієнти» підтверджено коректність роботи фільтрів пошуку та валідації полів вводу контактних даних. Важливим етапом стала симуляція одночасного запису декількох пацієнтів до одного лікаря, що дозволило переконатися в ефективності алгоритму перевірки часових колізій. Система продемонструвала здатність блокувати помилкові призначення та виводити відповідні попередження українською мовою. Тестування модуля «Лікарі» підтвердило стабільність відображення складних графіків роботи та прив'язку фахівців до відповідних кабінетів.

Окремо було проведено перевірку модуля звітності, під час якої генерувалися PDF-звіти за різні часові періоди з великою кількістю записів. Отримані документи повністю відповідають заданій структурі, коректно відображають кириличні символи та статистичні дані завантаженості персоналу. Тестування бази даних підтвердило стійкість файлу SQLite до критичних завершень програми та збереження повної цілісності інформації через механізм транзакцій Entity Framework Core. Інтерфейс програми, побудований на WPF, пройшов перевірку на адаптивність при різних роздільних здатностях моніторів, зберігаючи зручність навігації. Встановлено, що швидкість відгуку системи при вибірці з бази даних понад 1000 записів не перевищує допустимих значень, що свідчить про високу продуктивність обраної архітектури. Обробка виняткових ситуацій, таких як відсутність доступу до файлу бази даних або введення некоректних дат, спрацювала згідно

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

з програмною логікою. Проведене навантажувальне тестування підтвердило відсутність витоків пам'яті при тривалій роботі застосунку. За результатами випробувань зроблено висновок, що програмний засіб повністю реалізує заявлені функціональні вимоги.

Висновки до розділу 2

У другому розділі було проведено повний цикл проектування та програмної реалізації інформаційної системи для автоматизації роботи реєстратури медичного закладу. За результатами виконаної роботи можна зробити наступні висновки:

1. **Розроблено реляційну модель бази даних у СУБД SQLite**, яка включає чотири ключові сутності: користувачі, пацієнти, лікарі та призначення. Побудована фізична модель даних забезпечує зберігання повної інформації про діяльність реєстратури, а використання механізмів первинних і зовнішніх ключів гарантує цілісність даних та відсутність надмірності (дублювання) інформації.
2. **Спроектовано архітектуру програмного засобу** на базі сучасного патерну MVVM (Model-View-ViewModel), що дозволило чітко відокремити бізнес-логіку від графічного інтерфейсу. Це забезпечує високу гнучкість системи, легкість у тестуванні та можливість подальшої модифікації окремих модулів без порушення стабільності всієї програми.
3. **Реалізовано ергономічний інтерфейс користувача** засобами технології WPF. Розроблено набір вікон та представлень для автентифікації, ведення реєстрів пацієнтів та лікарів, а також інтерактивний модуль керування візитами. Весь інтерфейс повністю локалізований українською мовою, що відповідає вимогам до професійного медичного програмного забезпечення.
4. **Програмно реалізовано критично важливі алгоритми**, зокрема систему перевірки часових колізій у розкладі лікарів. Використання

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Entity Framework Core 9 дозволило автоматизувати взаємодію з базою даних, забезпечуючи високу швидкість виконання запитів та безпеку обробки персональної інформації.

5. **Впроваджено модуль генерації звітності** у форматі PDF за допомогою бібліотеки QuestPDF. Це дозволяє персоналу реєстратури отримувати офіційні друковані форми зведених звітів, журналів прийомів та списків пацієнтів, що значно спрощує аналітичну роботу та документообіг у закладі.

6. **Проведено комплексне тестування системи**, результати якого підтвердили стабільність роботи програми за різних умов навантаження. Перевірка функціональності засвідчила повну відповідність розробленого продукту поставленим завданням та технічним вимогам.

Розроблена інформаційна система є завершеним програмним рішенням, яке забезпечує повну автоматизацію реєстраційних процесів, підвищує ефективність роботи персоналу та якість обслуговування пацієнтів у медичному закладі.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

3.1 Мінімальні вимоги до системи

Для забезпечення коректної роботи розробленого програмного засобу «Інформаційна система для автоматизації роботи реєстратури медичного закладу» необхідно дотримуватися певних технічних параметрів. Оскільки система побудована на базі сучасної платформи .NET 9 та використовує технологію WPF для відтворення графічного інтерфейсу, ключовою вимогою до програмного середовища є наявність операційної системи Windows 10 (версії 1809 або новішої) або Windows 11. Обов'язковою умовою є встановлення середовища виконання .NET Desktop Runtime 9.0, яке забезпечує роботу базових бібліотек та компонентів. Апаратні вимоги до процесора передбачають використання архітектури x64 з тактовою частотою не менше 2.0 ГГц для швидкої обробки SQL-запитів. Мінімальний обсяг оперативної пам'яті (RAM) повинен становити 4 ГБ, що дозволяє підтримувати плавну роботу інтерфейсу та одночасне виконання фонових процесів генерації звітів. Для зберігання файлу бази даних SQLite та системних логів необхідно передбачити щонайменше 500 МБ вільного простору на накопичувачі типу SSD або HDD. Графічна підсистема повинна підтримувати DirectX 9 або новішої версії для апаратного прискорення рендерингу XAML-елементів. Мінімальна роздільна здатність монітора має становити 1024x768 пікселів, що гарантує коректне відображення всіх навігаційних панелей та таблиць даних.

Для стабільного функціонування модуля експорту звітів у форматі PDF у системі має бути присутній переглядник документів, такий як Adobe Acrobat Reader або Microsoft Edge. Оскільки база даних є локальною, наявність постійного підключення до мережі Інтернет не є обов'язковою для щоденної роботи реєстратора. Проте доступ до мережі може знадобитися для оновлення системних компонентів або відправки електронних листів пацієнтам, якщо такий функціонал буде активовано. Система потребує прав адміністратора лише під час першого запуску для автоматичного створення файлу бази даних та ініціалізації таблиць. Введення даних через клавіатуру та маніпулятор

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

«миша» є стандартними вимогами до пристроїв вводу. Наявність принтера є необхідною лише за потреби друку сформованих звітів безпосередньо з програми. Програмний засіб оптимізовано для роботи на стандартних офісних робочих станціях, що дозволяє уникнути значних витрат на оновлення парку комп'ютерної техніки закладу. Використання антивірусного програмного забезпечення рекомендується для захисту конфіденційних даних пацієнтів, але воно не повинно блокувати доступ програми до локального файлу медичної бази. Усі перелічені вимоги є базовими для підтримання високої швидкості відгуку інтерфейсу та надійності зберігання інформації. Дотримання цих параметрів мінімізує ризики виникнення системних збоїв та забезпечує комфортну експлуатацію інформаційної системи персоналом медичної установи.

Таблиця 3.1.1 – Зведена таблиця мінімальних вимог

Параметр	Вимога
Операційна система	Windows 10/11 (x64)
Програмна платформа	.NET 9 Desktop Runtime
Процесор	2.0 GHz Dual-Core (або кращий)
Оперативна пам'ять	4 GB
Дисковий простір	500 MB вільного місця
Відеокарта	Підтримка DirectX 9+
Дисплей	1024x768 або вище

3.2 Інструкція для роботи з програмою

1. Початок роботи та автентифікація

При запуску програми відкривається вікно входу. Для доступу до системи необхідно:

- Ввести індивідуальний логін у поле «Логін».
- Ввести пароль у поле «Пароль» (символи приховуються для безпеки).
- Натиснути кнопку «**Увійти до системи**» або клавішу Enter.

Примітка: Для першого входу можна скористатися тестовими даними: логін admin, пароль admin123.

2. Головне вікно та навігація

Після успішного входу відкривається основний інтерфейс. Навігація здійснюється через ліву панель:

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

- **Дашборд** – Візуальна статистика та розклад на поточний день.
- **Пацієнти** – Управління базою даних пацієнтів.
- **Лікарі** – Ведення списку медичного персоналу.
- **Призначення** – Робота з календарем записів на прийом.
- **Звіти** – Генерація аналітичних документів та PDF-експорт.
- **Користувачі** – Адміністрування доступів (лише для адміністраторів).

3. Робота з дашбордом

Розділ дозволяє швидко оцінити стан закладу:

Перегляд загальної кількості пацієнтів, лікарів та записів (сьогодні/тиждень) у верхніх картках.

Центральна панель відображає список візитів на сьогодні з кольоровою індикацією статусів: Завершено (зелений), Скасовано (червоний), Заплановано (синій).

Права панель відображає графічну статистику за місяць.

4. Управління пацієнтами та лікарями

Функціонал цих розділів аналогічний:

- **Пошук**

Використовуйте рядок пошуку для фільтрації за ПІБ, номером карти або телефону.

- **Додавання**

Натисніть « Додати», після чого з'явиться права панель для вводу даних.

- **Редагування**

Оберіть особу в списку (DataGrid), внесіть зміни у формі праворуч та натисніть « Зберегти».

- **Видалення**

Оберіть запис та натисніть « Видалити».

5. Запис пацієнта на прийом

Для створення нового запису:

- Натисніть « Нове» на панелі форми.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

- Оберіть пацієнта та лікаря зі списків, що випадають.
- Вкажіть дату через календар та час у форматі ГГ:ХХ (наприклад, 09:30).
- Вкажіть причину візиту та натисніть « Зберегти».
- Використовуйте верхню панель фільтрів для пошуку існуючих записів за датами, лікарем або статусом.

6. Генерація звітів

У розділі «Звіти» доступні такі дії:

- Встановіть період «Від» та «До».
- Натисніть « **Сформувати звіт**» для текстового попереднього перегляду статистики у нижній частині вікна.
- Для отримання офіційного документа натисніть на одну з кнопок експорту: «Зведений звіт», «Журнал прийомів», «Список пацієнтів» або «Список лікарів». Програма запропонує шлях для збереження PDF-файлу.

7. Завершення роботи

Для виходу з облікового запису натисніть кнопку виходу (іконка вимикача) у нижній частині бічної панелі. Повністю закрити програму можна через кнопку «Закрити» у вікні входу.

Висновки до розділу 3

У третьому розділі було проведено аналіз умов експлуатації та розроблено супровідну документацію для впровадження інформаційної системи в робочі процеси медичного закладу. На основі проведеної роботи можна зробити наступні висновки:

- **Визначено оптимальні технічні характеристики** апаратного та програмного забезпечення, необхідні для стабільного функціонування системи. Встановлено, що розроблений засіб на базі .NET 9 має помірні вимоги до ресурсів, що дозволяє використовувати його на стандартних офісних робочих станціях під керуванням Windows 10/11 без додаткових витрат на модернізацію обладнання.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

- **Обґрунтовано вибір локальної архітектури** зберігання даних. Використання СУБД SQLite забезпечує високу швидкість обробки інформації в межах локального робочого місця реєстратора та гарантує автономність роботи системи незалежно від наявності інтернет-з'єднання.
- **Розроблено детальну інструкцію користувача**, яка охоплює всі етапи роботи з програмою: від процесу автентифікації до генерації фінальної звітності. Чіткий опис інтерфейсу, що базується на модулях «Дашборд», «Пацієнти», «Лікарі» та «Призначення», дозволяє мінімізувати час на навчання персоналу та знизити ймовірність помилок при введенні даних.
- **Описано механізми взаємодії з модулем звітності**, що забезпечує автоматизоване формування журналів прийомів та реєстрів пацієнтів у форматі PDF. Це дозволяє цифровізувати значну частину паперового документообігу реєстратури та забезпечує адміністрацію закладу актуальними статистичними даними в режимі реального часу.
- **Підтверджено ергономічність графічного інтерфейсу**, який було спроектовано з урахуванням сучасних стандартів UX/UI дизайну. Наявність адаптивної системи навігації та візуальних підказок сприяє комфортній тривалій роботі реєстратора, підвищуючи загальну пропускну здатність медичного закладу.
- **Надано рекомендації щодо адміністрування системи**, зокрема керування обліковими записами користувачів та розмежування прав доступу, що є критично важливим для забезпечення безпеки медичної інформації та захисту персональних даних пацієнтів.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

ВИСНОВКИ

Узагальнюючи результати проведеного дослідження та розробки, можна зробити висновок, що мета кваліфікаційної роботи щодо створення інформаційної системи для автоматизації роботи реєстратури медичного закладу була повністю досягнута. У ході виконання роботи було проаналізовано предметну область та виявлено ключові чинники, що впливають на ефективність медичного документообігу. Розроблений програмний засіб забезпечує комплексну автоматизацію процесів реєстрації пацієнтів, ведення обліку лікарів та планування візитів. Використання платформи .NET 9 та мови програмування C# дозволило створити надійну архітектуру, здатну витримувати експлуатаційні навантаження сучасного медичного центру. Обрана технологія WPF забезпечила створення сучасного, інтуїтивно зрозумілого інтерфейсу, що повністю локалізований українською мовою.

Завдяки впровадженню патерну MVVM було досягнуто високого рівня модульності коду, що спрощує подальше супроводження та масштабування системи. Проектування бази даних у СУБД SQLite дозволило реалізувати автономне сховище інформації, яке не потребує складного адміністрування серверної інфраструктури. Реалізований алгоритм перевірки доступності часових інтервалів унеможливорює виникнення конфліктів у розкладі лікарів, що суттєво підвищує якість обслуговування пацієнтів. Особливу практичну цінність має модуль звітності, який дозволяє автоматично генерувати професійні PDF-звіти про роботу закладу. У системі впроваджено гнучку модель автентифікації та розмежування прав доступу, що гарантує захист персональних даних пацієнтів згідно з чинним законодавством.

Проведене тестування підтвердило стабільність роботи всіх модулів системи: від реєстрації нових користувачів до експорту журналів прийомів. Інтерфейс програми продемонстрував високу адаптивність та ергономічність, забезпечуючи швидкий доступ до необхідних функцій через бічне навігаційне меню. Впровадження системи дозволяє значно скоротити час на обробку

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

одного звернення пацієнта та мінімізувати вплив людського фактору на ведення медичної документації. Статистичні інструменти дашборду надають адміністрації можливість проводити оперативний аналіз завантаженості персоналу в режимі реального часу. Розроблена інструкція користувача забезпечує швидку адаптацію медичного персоналу до роботи з новим програмним продуктом.

Економічна доцільність проекту обґрунтована низькими витратами на впровадження та підтримку системи завдяки використанню безкоштовних технологій з відкритим кодом. Створена інформаційна система є цілісним інтелектуальним продуктом, який відповідає сучасним вимогам цифрової трансформації охорони здоров'я. Результати роботи можуть бути використані як база для подальшого розвитку функціоналу, наприклад, інтеграції з державними реєстрами або створення особистого кабінету пацієнта. Загалом, розроблена система є ефективним інструментом підвищення операційної ефективності медичного закладу та оптимізації внутрішніх бізнес-процесів реєстратури. Практичне впровадження системи дозволить медичній установі вийти на новий рівень якості надання послуг населенню.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Про охорону здоров'я : Закон України від 19.11.1992 № 2801-XII. URL: <https://zakon.rada.gov.ua/laws/show/2801-12> (дата звернення: 10.02.2026).
2. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 10.02.2026).
3. Про електронні довірчі послуги : Закон України від 05.10.2017 № 2155-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2155-19> (дата звернення: 11.02.2026).
4. Деякі питання електронної системи охорони здоров'я : Постанова Кабінету Міністрів України від 25.04.2018 № 411. URL: <https://zakon.rada.gov.ua/laws/show/411-2018-п> (дата звернення: 12.02.2026).
5. Албахарі Д., Албахарі Б. С# 12 у двох словах. Найповніший довідник. Київ : Діалектика, 2024. 1152 с.
6. Антощук С. Г., Нікола С. О. Об'єктно-орієнтоване програмування : навч. посіб. Одеса : ОНПУ, 2020. 182 с.
7. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1. Організація баз даних та знань : підручник. Львів : Магнолія-2006, 2019. 440 с.
8. Бублик В. В. Об'єктно-орієнтоване програмування : підручник. Київ : ІТ-книга, 2021. 624 с.
9. Ваврук Є. Я. Проектування інформаційних систем : навч. посіб. Львів : ЛНУ імені Івана Франка, 2022. 248 с.
10. Глиновець Н. М. Розробка застосунків на платформі .NET : навч. посіб. Київ : НаУКМА, 2021. 156 с.
11. Жежнич П. І. Технології створення та управління інформаційними ресурсами : навч. посіб. Львів : Видавництво Львівської політехніки, 2020. 216 с.
12. Карпець О. В. Сучасні технології розробки програмного забезпечення на платформі .NET : навч. посіб. Харків : ХНУРЕ, 2023. 190 с.

					КРФМБ.122.042.034.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

13. Ковалюк Т. В. Основи програмування : підручник. Київ : Видавнича група BHV, 2019. 400 с.
14. Кравець П. О. Об'єктно-орієнтоване програмування : навч. посіб. Львів : Видавництво Львівської політехніки, 2021. 320 с.
15. Лавріщева К. М. Програмна інженерія : підручник. Київ : Академперіодика, 2020. 310 с.
16. Литвин В. В. Інформаційні системи та технології : навч. посіб. Львів : Новий Світ-2000, 2021. 350 с.
17. Любченко В. В. Організація баз даних : навч. посіб. Одеса : ОНПУ, 2022. 164 с.
18. Пасічник В. В., Резніченко В. А. Організація баз даних та знань : підручник. Київ : Видавнича група BHV, 2020. 448 с.
19. Пономаренко В. С. Проектування інформаційних систем : підручник. Київ : Академія, 2019. 416 с.
20. Сидоров М. О. Архітектура програмного забезпечення : навч. посіб. Київ : НАУ, 2021. 180 с.
21. Співаковський О. В. Побудова та аналіз алгоритмів : навч. посіб. Херсон : ХДУ, 2020. 192 с.
22. Трофименко О. Г. C#. Створення застосунків Windows Forms та WPF : навч. посіб. Одеса : Фенікс, 2021. 260 с.
23. Шаховська Н. Б. Системи керування базами даних : навч. посіб. Львів : Видавництво Львівської політехніки, 2022. 232 с.
24. Щербина Ю. М. Моделювання та проектування інформаційних систем : підручник. Львів : Видавничий центр ЛНУ імені Івана Франка, 2021. 380 с.
25. Freeman A. Pro Entity Framework Core 8: Data Access in .NET. Apress, 2024. 740 p.