

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
ФАКУЛЬТЕТ №2**

Кафедра інформаційних технологій

**ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ УПРАВЛІННЯ ВИКЛИКАМИ,
ОСОБОВИМ СКЛАДОМ І ТЕХНІКОЮ ВІДДІЛУ ПОЛІЦІЇ**

кваліфікаційна робота
здобувача вищої освіти
4 курсу денної форми навчання
Олег ВОЙТАНОВСЬКИЙ

Науковий керівник:
доцент, кандидат технічних наук
Тарас РУДИЙ

Рецензент:
доцент, кандидат технічних наук
Ірина Борецька

Кваліфікаційна робота допущена до захисту
«___» _____ 2026 р., протокол № _____

Завідувач кафедри інформаційних технологій
_____ **Олег ЗАЧЕК**
(підпис)

Львів
2026

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

НПУ - Національна поліція України

ІС - інформаційна система

БД - база даних

СУБД - система управління базами даних

MVC - Model-View-Controller (модель-представлення-контролер)

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

HTTP - HyperText Transfer Protocol (протокол передачі гіпертексту)

API - Application Programming Interface (інтерфейс прикладного програмування)

EF - Entity Framework

UI - User Interface (інтерфейс користувача)

CRUD - Create, Read, Update, Delete (створення, читання, оновлення, видалення)

SQL - Structured Query Language (мова структурованих запитів)

DBMS - Database Management System (система управління базами даних)

ER - Entity-Relationship (сутність-зв'язок)

АНОТАЦІЯ

ВОЙТАНОВСЬКИЙ О. Інформаційна система для управління викликами, особовим складом і технікою відділу поліції. - Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 126 «Інформаційні системи та технології». - Львівський державний університет внутрішніх справ, МВС України, Львів, 2025.

У кваліфікаційній роботі розглянуто процес розроблення інформаційної системи для автоматизації обліку викликів, управління особовим складом і технічними засобами підрозділів Національної поліції України. У ході виконання роботи проведено аналіз предметної області, досліджено особливості діяльності підрозділів поліції та існуючі інформаційні рішення у сфері правоохоронної діяльності.

У роботі визначено основні вимоги до інформаційної системи, спроектовано її архітектуру та структуру бази даних. Реалізовано Web-застосунок із використанням сучасних технологій, який забезпечує централізоване зберігання даних, облік викликів, управління персоналом і технікою, а також підтримує базові механізми авторизації та контролю доступу.

У заключній частині роботи проведено оцінку ефективності розробленої системи та визначено напрями її подальшого розвитку.

Ключові слова: інформаційна система, Web-застосунок, управління викликами, Національна поліція, база даних, автоматизація, MVC.

ABSTRACT

VOITANOVSKYI O. Information System for Managing Calls, Personnel, and Equipment of a Police Department - Manuscript.

Research for obtaining a bachelor's degree in specialty 126 «Information systems and technologies». Lviv State University of Internal Affairs, MIA of Ukraine, Lviv, 2025.

This work is devoted to the development of an information system for automating call management, personnel, and technical resources in units of the National Police of Ukraine. During the research, the subject area was analyzed, the specifics of police unit activities were studied, and existing information systems in the field of law enforcement were reviewed.

The main requirements for the system were defined, its architecture and database structure were designed. A web application was developed using modern technologies, which provides centralized data storage, call registration, personnel and equipment management, as well as basic authorization and access control mechanisms.

In the final part of the work, the efficiency of the developed system was evaluated and directions for its further development were determined.

Keywords: information system, web application, call management, National Police, database, automation, MVC.

ЗМІСТ

В

РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ІНФОРМАЦІОННОГО

З

Ж 1

Н 1

Е 1

П 2

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ

Е 3

У 3

Ч 0

П 2

Е 2

Р 2

Н 2

А 2

Н 2

В 2

Я 3

Л 3

І 3

У 3

Н 3

П 3

Н 3

Я 3

А 3

В 3

В 3

Н 3

О 3

К 3

О 3

Л 3

И 3

Я 3

К 3

І 3

А 3

В 3

Н 3

И 3

Д 3

Н 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

С 3

О 3

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням автоматизованих інформаційних систем у діяльність державних органів, зокрема у сферу забезпечення правопорядку та громадської безпеки. Застосування програмного забезпечення сприяє підвищенню результативності управлінських рішень, прискоренню обробки інформації та зниженню ризику виникнення помилок, зумовлених впливом людського фактора. Особливої актуальності це набуває для підрозділів НПУ, діяльність яких пов'язана з оперативним реагуванням на події, обробкою значних обсягів звернень та координацією дій особового складу і технічних засобів.

У процесі щоденної роботи підрозділи поліції стикаються з необхідністю ведення обліку викликів, розподілу ресурсів, контролю за виконанням завдань, а також збереження й аналізу інформації про події, працівників і задіяну техніку. За відсутності сучасних автоматизованих систем значна частина цих процесів виконується вручну або із застосуванням застарілих програмних засобів, що ускладнює оперативність прийняття рішень і знижує загальну ефективність діяльності підрозділу.

На сьогодні у правоохоронних органах використовуються різні інформаційно-аналітичні системи, призначені для обліку подій, осіб та інших об'єктів оперативного інтересу. Водночас значна частина таких систем відзначається складністю впровадження, потребує суттєвих обчислювальних потужностей, спеціалізованого програмного забезпечення, а також належної підготовки персоналу. Це обмежує можливість їх використання у навчальних цілях або для моделювання діяльності підрозділів у локальному середовищі.

У зв'язку з цим актуальним є створення спрощеної, але функціонально достатньої інформаційної системи, яка дозволяє автоматизувати основні

процеси управління викликами, особовим складом і технікою. Така система має забезпечувати централізоване зберігання інформації, оперативний пошук і доступ до даних, зручний користувацький інтерфейс, а також базові засоби контролю доступу. Реалізування системи у вигляді Web-застосунку дозволяє забезпечити доступність, масштабованість і наочність її використання.

Метою кваліфікаційної роботи є створення інформаційної системи для управління викликами, особовим складом і технічними ресурсами підрозділу поліції у формі Web-застосунку, який забезпечує збереження, обробку та відображення даних, а також підтримує базові механізми авторизації та адміністрування користувачів.

Для досягнення поставленої мети в межах роботи передбачено виконання сукупності взаємопов'язаних завдань. Зокрема, необхідно провести аналіз предметної області та специфіки функціонування підрозділів Національної поліції, дослідити існуючі програмні рішення у сфері інформаційного забезпечення правоохоронних органів і сформулювати вимоги до майбутньої системи. Крім того, передбачено проектування структури бази даних і архітектури програмного забезпечення, реалізацію Web-застосунку з використанням сучасних технологій, а також перевірку його працездатності на тестових даних.

Об'єктом дослідження є процеси організації інформаційного забезпечення та управління діяльністю підрозділів Національної поліції України.

Предметом дослідження виступають методи та програмні засоби, спрямовані на автоматизацію обліку викликів, особового складу та технічних ресурсів із використанням Web-технологій.

У процесі виконання роботи застосовано методи аналізу та синтезу, принципи об'єктно-орієнтованого програмування, підходи до моделювання інформаційних систем, а також методи програмної реалізації з використанням платформи .NET, мови програмування C# та сучасних Web-технологій.

Практична значущість роботи полягає у створенні функціонального прототипу інформаційної системи, який може бути використаний у навчальному процесі для демонстрації принципів побудови інформаційних систем, а також слугувати основою для подальшого розвитку й адаптації до реальних умов діяльності підрозділів Національної поліції України.

РОЗДІЛ 1.

АНАЛІЗ СУЧАСНОГО СТАНУ ІНФОРМАЦІОННОГО ЗАБЕПЕЧЕННЯ ДІЯЛЬНОСТІ ПІДРОЗДІЛІВ ПОЛІЦІЇ

1.1. Особливості організації роботи підрозділів Національної поліції.

Діяльність підрозділів Національної поліції України спрямована на забезпечення публічної безпеки та правопорядку, захист прав і свобод громадян, протидію злочинності, а також оперативне реагування на події різного характеру. Для реалізації цих функцій працівники поліції щоденно працюють із великими обсягами інформації, що включає звернення громадян, дані про виїзди на місце подій, облік особового складу, використання службового транспорту та інших ресурсів.

Функціонування підрозділів Національної поліції регулюється чинним законодавством України. Зокрема, відповідно до Закону України «Про Національну поліцію» [1], до основних завдань належать забезпечення публічної безпеки, захист прав і свобод громадян і боротьба зі злочинністю. Виконання цих завдань вимагає ефективної організації інформаційного забезпечення та застосування сучасних інформаційних технологій.

Робота підрозділів поліції має чітко визначену структуру та ґрунтується на постійному інформаційному обміні між черговими частинами, оперативними підрозділами та керівництвом. У процесі реагування на події здійснюється реєстрація викликів, визначення їх пріоритетності, призначення відповідальних виконавців, залучення необхідних ресурсів і контроль за виконанням поставлених завдань. У зв'язку з цим інформація, яка обробляється, повинна бути впорядкованою, актуальною та доступною в будь-який момент.

Важливим напрямом діяльності є також ведення обліку особового складу та технічних ресурсів. Система повинна забезпечувати швидкий доступ до відомостей про працівників, їхню спеціалізацію, а також про стан і доступність службового транспорту. Особливої ваги це набуває під час реагування на надзвичайні ситуації або в умовах підвищеного навантаження на підрозділи.

Діяльність підрозділів Національної поліції України характеризується високим рівнем відповідальності, необхідністю оперативного реагування та постійною взаємодією між різними структурними елементами. До основних учасників процесу реагування на події належать чергова частина, оперативні групи, патрульні підрозділи та керівний склад.

Чергова частина здійснює прийом повідомлень від громадян, їх реєстрацію та первинну обробку. На цьому етапі визначається характер події, рівень її пріоритетності та необхідність залучення відповідних сил і засобів. Оперативні групи та патрульні екіпажі забезпечують виїзд на місце події, виконують поставлені завдання та фіксують результати реагування.

У процесі обробки викликів задіяні різні категорії працівників, кожна з яких виконує визначені функції та працює з відповідними інформаційними ресурсами. Для більш наочного відображення ролей і обов'язків учасників доцільно узагальнити їх у табличному вигляді. Основні функції учасників процесу реагування на виклик наведено в табл. 1.1.

Таблиця 1.1

Основні функції учасників процесу реагування на виклик

Учасник	Основні обов'язки	Інформація, з якою працює
Черговий диспетчер	Прийом звернень громадян, первинна реєстрація та класифікація події	Адреса події, суть правопорушення, контактні дані заявника
Патрульний екіпаж	Виїзд на місце події, оперативне реагування, фіксація обставин	Деталі виклику, орієнтування, статус виконання
Керівник підрозділу	Моніторинг загальної оперативної обстановки, контроль якості роботи	Аналітичні звіти, статистика викликів, завантаженість персоналу
Системний адміністратор	Забезпечення безперебійної роботи системи, управління доступом	Облікові записи користувачів, системні логи, резервні копії

Джерело: складено автором на основі використаних матеріалів

Стандартний сценарій обробки виклику включає такі етапи: надходження повідомлення, реєстрація виклику, визначення відповідальних осіб, призначення ресурсів (особового складу та техніки), виконання завдання та формування звіту. Кожен із цих етапів супроводжується обробкою значного обсягу інформації, що підвищує вимоги до організації інформаційного забезпечення.

Таким чином, ефективність діяльності підрозділів поліції значною мірою залежить від швидкості обробки інформації, доступу до актуальних даних та можливості координації дій у режимі реального часу.

Значну роль відіграє робота в умовах підвищеного навантаження. Під час масових заходів, аварійних ситуацій або інших надзвичайних подій кількість звернень може суттєво зростати. У таких умовах особливо важливо, щоб система обліку викликів не лише фіксувала звернення, але й забезпечувала визначення його пріоритетності, доступних ресурсів та поточного стану виконання.

Ще одним важливим аспектом є накопичення та аналіз історичних даних. Інформація про попередні виклики, використання ресурсів та результати виконання завдань має значну аналітичну цінність. Вона дає змогу оцінювати результативність діяльності підрозділу, виявляти проблемні аспекти та приймати обґрунтовані управлінські рішення.

Водночас діяльність підрозділів Національної поліції України характеризується складністю, багаторівневістю та динамічністю, що зумовлює потребу у використанні сучасних спеціалізованих інформаційних систем. Такі системи повинні забезпечувати централізоване зберігання даних, оперативний доступ до них, підтримку процесу прийняття рішень і контроль за виконанням поставлених завдань.

Аналіз існуючих інформаційних систем у правоохоронних органах.

У діяльності правоохоронних органів України активно використовуються інформаційні системи, призначені для збору, зберігання та обробки оперативної інформації. Такі системи охоплюють облік кримінальних правопорушень, дані про осіб, транспортні засоби, події, а також результати оперативно-розшукової діяльності. Вони забезпечують централізований доступ до інформації та сприяють підвищенню ефективності роботи працівників поліції.

З метою визначення переваг і недоліків різних підходів до обліку інформації було проведено їх порівняльний аналіз. Це дозволяє обґрунтувати доцільність переходу до автоматизованих систем управління. Результати такого порівняння наведені в таблиці 1.2.

Таблиця 1.2

Порівняльна характеристика існуючих підходів до обліку інформації

Підхід	Переваги	Недоліки	Доцільність використання
Паперовий журнал	Незалежність від електроживлення, простота ведення	Повільний пошук, ризик втрати або пошкодження, відсутність аналітики	Лише як резервний метод на випадок аварій
Електронні таблиці (Excel)	Швидке впровадження, базові можливості фільтрації та пошуку	Відсутність розмежування прав доступу, ризик пошкодження файлу	Доцільно для малих обсягів неструктурованих даних
Локальні СУБД (Access)	Структурованість даних, можливість створення форм	Складність підтримки, обмеження на багатокористувачку роботу	Тимчасове рішення для окремих робочих місць
Спеціалізована Webсистема	Централізований доступ, розмежування прав, висока швидкість, інтеграція	Необхідність фінансових та часових витрат на розробку	Хороший варіант для повноцінної автоматизації підрозділу

Джерело: складено автором за матеріалами

Окрім цього, значна частина існуючих рішень орієнтована на використання в межах централізованої інфраструктури та не пристосована для навчальних цілей або моделювання роботи підрозділів у спрощеному середовищі. Їх впровадження супроводжується складною процедурою налаштування, обмеженим доступом до функціоналу та підвищеними вимогами до інформаційної безпеки.

На практиці це призводить до того, що в окремих підрозділах використовуються допоміжні засоби обліку, такі як електронні таблиці, локальні бази даних або паперова документація. Подібний підхід не забезпечує належного рівня інтеграції даних, ускладнює контроль за виконанням завдань і не дозволяє ефективно аналізувати накопичену інформацію.

У правоохоронних органах можна виділити кілька основних типів інформаційних систем: централізовані державні системи, локальні програмні рішення та допоміжні інструменти обліку.

До централізованих систем належать інформаційно-аналітичні комплекси, що забезпечують облік правопорушень, осіб і подій. Вони характеризуються широким функціоналом і високим рівнем інтеграції, однак часто є складними у використанні та потребують спеціального навчання персоналу.

Локальні програмні рішення, навпаки, є простішими у впровадженні та не потребують складної інфраструктури. Їх перевагами є швидкий початок роботи та невисокі технічні вимоги. Водночас такі рішення мають суттєві обмеження: вони не забезпечують повноцінної багатокористувацької роботи, централізованого зберігання даних та ефективного формування аналітичної звітності.

Аналіз існуючих інформаційних систем доцільно проводити не лише з точки зору їх функціональності, але й з урахуванням практичної придатності для повсякденної роботи користувача. У реальних умовах система повинна бути зручною, зрозумілою, стабільною та достатньо швидкою. Тому важливими характеристиками є зручність інтерфейсу, швидкість пошуку даних, рівень інтеграції між підсистемами, доступність для різних категорій користувачів і простота адміністрування.

Більшість длі централізованих систем орієнтовані на обробку великих обсягів даних і значну кількість користувачів. Це забезпечує високий аналітичний потенціал, однак ускладнює їх використання в навчальних або

демонстраційних цілях. Вони потребують спеціального доступу, відповідного технічного забезпечення та детального ознайомлення з внутрішніми регламентами.

Отже, проведений аналіз показує, що існуючі підходи до організації обліку інформації мають як переваги, так і суттєві недоліки. У зв'язку з цим доцільним є створення інформаційної системи, яка поєднує простоту впровадження, достатній функціонал для управління викликами та ресурсами, а також зручний Web-інтерфейс для користувача.

Постановка задачі та обґрунтування необхідності розробки системи.

Проведений аналіз організації діяльності підрозділів Національної поліції України та існуючих інформаційних рішень свідчить про наявність проблем, пов'язаних із обробкою та зберіганням інформації. Зокрема, відсутність універсального інструменту для обліку викликів, особового складу та технічних засобів ускладнює процес управління, знижує оперативність реагування та ускладнює контроль за виконанням службових завдань.

Метою розробки даної інформаційної системи є створення програмного засобу, що забезпечує автоматизацію процесів обліку викликів, управління особовим складом і технікою підрозділу поліції. Система реалізується у вигляді Web-застосунку, що забезпечує зручний доступ через браузер, простоту використання та можливість подальшого розширення функціональних можливостей.

Для досягнення визначеної мети в межах цієї роботи передбачено виконання низки ключових завдань, зокрема аналіз предметної області та формування вимог до системи, проектування структури бази даних, розроблення логіки функціонування програмного застосунку, створення користувацького інтерфейсу, а також перевірку працездатності системи на тестових даних. Особлива увага при цьому приділяється зручності використання та наочності відображення інформації.

Постановка задачі в межах даної роботи передбачає формування чіткої логіки розроблення програмного засобу, що відповідає вимогам предметної області. У цьому контексті необхідно не лише визначити перелік функцій майбутньої системи, але й обґрунтувати їх практичну доцільність, взаємозв'язок та очікувані результати використання.

Окрім функцій безпосереднього обліку, система повинна реалізовувати базові механізми контролю доступу. Доступ до функціоналу мають отримувати лише авторизовані користувачі, а операції створення, редагування та перегляду даних повинні здійснюватися відповідно до визначених ролей і прав доступу.

Формування постановки задачі передбачає також визначення основних функцій, які має виконувати інформаційна система. Узагальнення цих функцій дозволяє краще зрозуміти її призначення та структуру. Основні задачі, що вирішуються системою, наведені в таблиці 1.3.

Таблиця 1.3

Основні задачі інформаційної системи

Задача	Призначення	Очікуваний результат
Реєстрація викликів	Фіксація інформації про правопорушення у єдиній базі даних	Створення електронної картки виклику із зазначенням усіх необхідних параметрів
Розподіл ресурсів	Призначення відповідальних поліцейських та транспортних засобів на виклик	Мінімізація часу очікування вільного екіпажу та раціональне використання техніки
Моніторинг виконання	Відстеження поточного статусу опрацювання кожного виклику	Забезпечення контролю за своєчасністю та якістю реагування на події
Управління складом	Ведення обліку кадрів та доступного технічного забезпечення	Актуальна картина наявності сил і засобів у режимі реального часу
Формування звітності	Автоматична генерація аналітичних довідок за заданий період	Зменшення паперового навантаження на персонал під час підготовки звітів

Джерело: укладено автором за результатами аналізу матеріалів

Узагальнюючи викладене, можна зробити висновок, що поставлена задача є комплексною, проте цілком придатною для реалізації в межах кваліфікаційної роботи. Її вирішення дозволяє створити функціональний

прототип інформаційної системи, який поєднує теоретичні підходи до автоматизації з практичною реалізацією у вигляді Web-застосунку.

РОЗДІЛ 2.

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ УПРАВЛІННЯ ВИКЛИКАМИ, ОСОБОВИМ СКЛАДОМ І ТЕХНІКОЮ ВІДДІЛУ ПОЛІЦІЇ

Загальна концепція та архітектура системи.

Проєктування інформаційної системи для управління викликами, особовим складом і технічними ресурсами здійснюється з урахуванням принципів модульності, зручності використання та можливості подальшого розширення функціоналу. З огляду на поставлені завдання обрано Web-орієнтований підхід до реалізації, що забезпечує доступ до системи через стандартний Web-браузер без необхідності встановлення додаткового програмного забезпечення.

В основі розроблюваного програмного рішення лежить архітектурний шаблон MVC (Model-View-Controller), який широко застосовується під час створення Web-застосунків і забезпечує чітке розмежування функцій між компонентами системи. Модель відповідає за збереження та обробку даних, представлення — за їх відображення користувачу, тоді як контролер здійснює обробку запитів і координує взаємодію між іншими складовими.

Архітектурний підхід MVC є одним із найбільш поширених у сучасній розробці програмних систем. Як зазначено у працях з програмної інженерії [9 - 10], його використання дозволяє відокремити бізнес-логіку, дані та інтерфейс, що спрощує супровід, тестування та масштабування програмного продукту.

Для наочного представлення загальної архітектури системи доцільно використати відповідну схему. Архітектурну модель Web-застосунку, побудованого за шаблоном MVC, наведено на рис. 2.1.

Ієрархія архітектури MVC

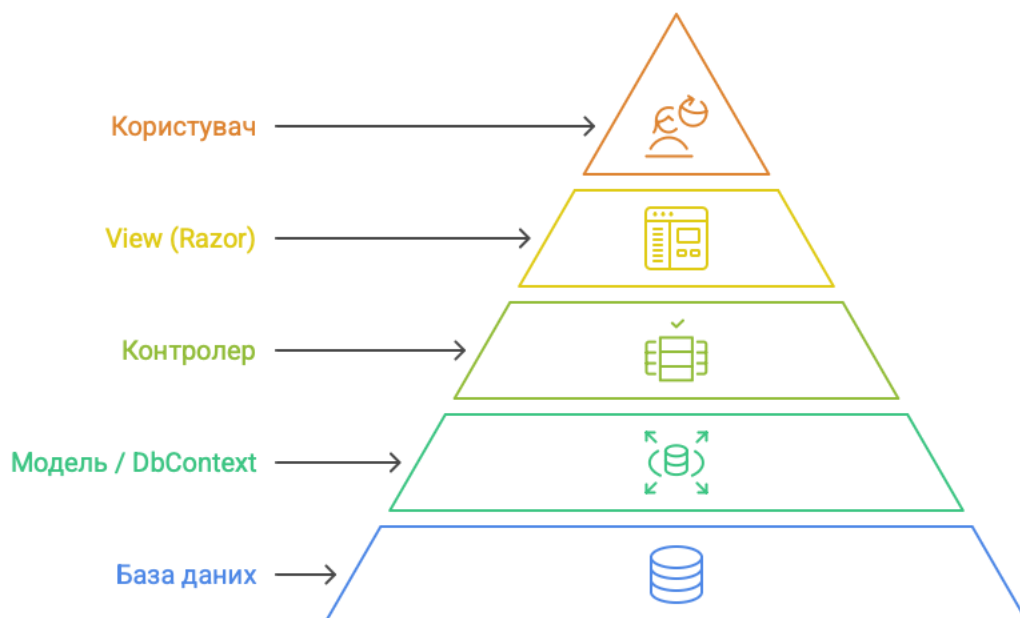


Рисунок 2.1 — Архітектура Web-застосунку на основі MVC

Така архітектура дозволяє підвищити зрозумілість коду, спростити супровід системи та забезпечити її масштабованість. Крім того, використання MVC сприяє відокремленню логіки роботи з даними від інтерфейсу користувача, що є важливим при подальшій модернізації або розширенні функціоналу.

Система розробляється у вигляді клієнт-серверного Web-застосунку, в якому серверна частина відповідає за реалізацію бізнес-логіки, взаємодію з базою даних і контроль доступу, тоді як клієнтська частина забезпечує зручну взаємодію користувача з системою через Web-інтерфейс.

Вибір архітектури MVC є обґрунтованим, оскільки вона належить до найбільш поширених підходів у розробці Web-застосунків. Її використання забезпечує чітке розмежування рівнів даних, логіки та представлення, що сприяє підвищенню структурованості коду, його зрозумілості та зручності супроводу. У межах дипломної роботи це також має важливе значення, адже

дозволяє наочно продемонструвати архітектурну побудову програмного продукту.

Клієнтська частина системи відповідає за взаємодію з користувачем. Саме на цьому рівні здійснюється відображення списків викликів, працівників і техніки, введення нових даних через форми та ініціювання операцій редагування або видалення записів. Серверна частина обробляє запити, перевіряє коректність отриманих даних, виконує бізнес-логіку та взаємодіє з БД.

Для кращого розуміння ролі кожного елемента архітектури MVC доцільно розглянути їх призначення та функціональні можливості. Узагальнення цієї інформації дозволяє більш чітко уявити структуру системи. Відповідні характеристики компонентів наведені в таблиці 2.1.

Таблиця 2.1

Призначення компонентів архітектури MVC

Компонент	Призначення	Функції в системі
Model (Модель)	Опис структури даних та бізнес-логіки додатка	Взаємодія з базою даних через EF Core, обробка інформації, представлення об'єктів
View (Представлення)	Формування користувацького інтерфейсу	Відображення даних з моделі у вигляді HTML-сторінок, взаємодія з користувачем
Controller (Контролер)	Обробка вхідних запитів та координація роботи системи	Отримання даних від View, звернення до Model для обробки, повернення результату
Routing (Маршрутизація)	Співставлення URL-адрес із	Забезпечення логічної та зрозумілої

	відповідними методами контролерів	структури посилань у системі
--	--------------------------------------	---------------------------------

Джерело: розроблено автором на основі опрацьованих джерел

Важливою перевагою обраної архітектури є можливість подальшого розширення системи. Якщо в майбутньому виникне потреба додати нові модулі, наприклад систему звітності, ролі користувачів або об'єднання з іншими банками даних, це можна буде зробити без повного перепроєктування вже реалізованих компонентів.

Таким чином, загальна концепція та архітектура системи відповідають як вимогам предметної області, так і сучасним принципам побудови Web-застосунків.

Розроблення структури даних та логічних зв'язків між об'єктами

Функціональні вимоги до інформаційної системи формувалися на основі аналізу діяльності підрозділів поліції та типових задач, що виникають у процесі їх роботи. Система повинна забезпечувати можливість ведення обліку викликів, зберігання інформації про особовий склад і технічні засоби, а також підтримувати взаємозв'язок між цими об'єктами.

Основною функцією системи є робота з викликами. Користувач повинен мати змогу створювати нові виклики, вказувати їхню назву, адресу, час надходження, рівень пріоритету та поточний статус. Також передбачається можливість редагування та перегляду інформації про виклики, що дозволяє контролювати хід їх виконання.

Для кращого розуміння складу основних модулів і взаємозв'язків між ними доцільно подати структуру системи у графічному вигляді. Структуру інформаційної системи наведено на рис. 2.2.

Блоки системи



Рисунок 2.2 — Структура інформаційної системи

Система повинна забезпечувати облік викликів, збереження інформації про особовий склад і технічні ресурси, а також підтримку взаємозв'язків між цими елементами.

Розроблювана інформаційна система реалізована на основі клієнт-серверної архітектури, яка передбачає розподіл функціональних обов'язків між серверною та клієнтською частинами. Серверна складова відповідає за обробку запитів, взаємодію з базою даних і реалізацію бізнес-логіки, тоді як клієнтська забезпечує відображення інформації та взаємодію користувача із системою.

Обмін даними між клієнтом і сервером здійснюється за допомогою HTTP-запитів, що дає змогу працювати із системою через Web-браузер без необхідності встановлення додаткового програмного забезпечення.

Використання архітектури MVC забезпечує чітке розмежування функцій між компонентами системи, що спрощує супровід програмного коду, підвищує його масштабованість і полегшує внесення змін у подальшому.

У ході проектування системи було визначено ключові сутності, які відображають об'єкти предметної області. Кожна з них має власне призначення та набір характеристик. Узагальнені відомості про основні сутності системи наведено в табл. 2.2.

Таблиця 2.2

Основні сутності інформаційної системи

Сутність	Призначення	Основні атрибути
Виклик (Call)	Зберігання повної інформації про зареєстровану подію	Ідентифікатор, назва, адреса, час, пріоритет, статус
Поліцейський (Officer)	Облік особового складу відділу поліції	Ідентифікатор, ПІБ, звання, посада, контактний телефон
Техніка (Vehicle)	Фіксація наявних транспортних засобів та спецтехніки	Ідентифікатор, марка, державний номер, тип, стан доступності
Користувач (User)	Забезпечення авторизованого доступу до системи	Логін, хеш пароля, роль, прив'язка до поліцейського

Джерело: розроблено автором на основі опрацьованих джерел

Таким чином, структура даних і логічні зв'язки між об'єктами системи є важливою складовою якісного проектування, оскільки вони безпосередньо впливають на зручність реалізації функціоналу, стабільність роботи та можливість подальшого розвитку системи.

Проектування структури бази даних.

Реляційні бази даних є одним із найбільш поширених рішень в інформаційних системах завдяки своїй надійності та впорядкованій структурі. Як зазначено у фундаментальних дослідженнях у цій сфері [11 - 12],

застосування принципів нормалізації та встановлення зв'язків між таблицями сприяє забезпеченню цілісності даних і підвищенню ефективності їх обробки.

Ключовим елементом функціонування інформаційної системи є база даних, яка забезпечує збереження, цілісність і узгодженість інформації. Під час проєктування її структури було враховано потребу у зберіганні даних про виклики, персонал і технічні ресурси, а також у встановленні зв'язків між цими об'єктами.

У системі реалізовано використання реляційної бази даних, що дає змогу логічно впорядкувати інформацію та забезпечити ефективний доступ до неї. Основними сутностями виступають «Виклик», «Поліцейський» і «Техніка». Кожен виклик пов'язується з конкретним працівником та, за необхідності, з одиницею техніки. Такий підхід дає змогу відстежувати, хто саме виконував завдання та які ресурси були задіяні.

Для більш детального аналізу структури зберігання даних у системі доцільно розглянути ER-діаграму бази даних, яка відображає ключові сутності та взаємозв'язки між ними. ER-діаграму бази даних інформаційної системи наведено на рис. 2.3.

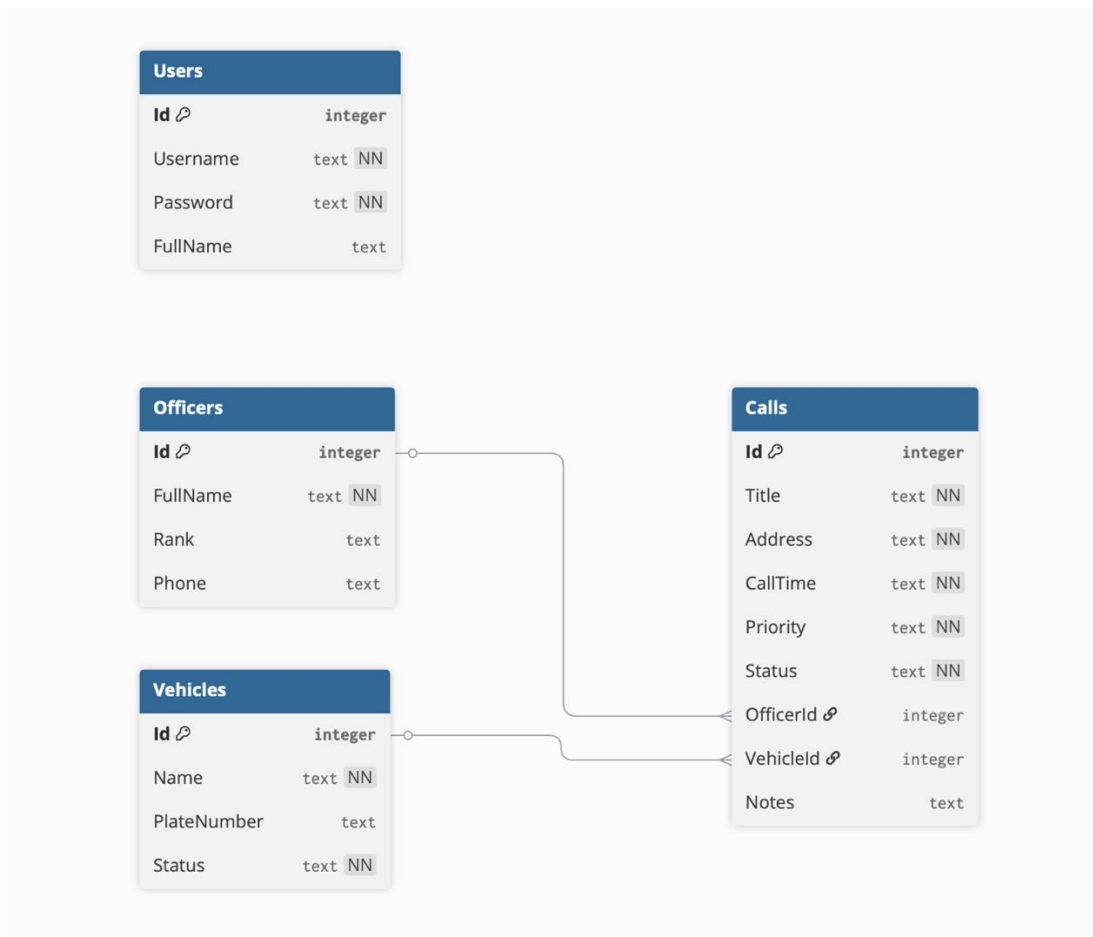


Рисунок 2.3 — ER-діаграма бази даних інформаційної системи

Для кожної сутності визначено сукупність атрибутів, що описують відповідний об'єкт. Взаємозв'язки між таблицями реалізовано за допомогою зовнішніх ключів, що забезпечує цілісність даних і запобігає появі некоректних записів.

Використання об'єктно-реляційного відображення за допомогою EF Core дозволяє спростити роботу з базою даних, зменшити кількість шаблонного коду та забезпечити зручну взаємодію між моделями предметної області та таблицями БД.

База даних виступає ключовим елементом інформаційної системи, оскільки забезпечує збереження, цілісність і узгодженість даних. У межах розробленої системи застосовується реляційна модель, що дає змогу організувати інформацію у вигляді взаємопов'язаних таблиць.

Більш детальне уявлення про структуру БД можна отримати шляхом аналізу окремих таблиць та їхніх полів. Це дозволяє оцінити повноту та

логічність збереження інформації. Структура основних таблиць БД наведена в таблицях 2.3-2.5.

Таблиця 2.3

Структура таблиці «Виклик»

Поле	Тип даних	Призначення
Id	Integer	Первинний ключ і унікальний ідентифікатор виклику
Title	String	Короткий опис або кваліфікація події (наприклад, "Крадіжка")
Address	String	Місце скоєння правопорушення або адреса заявника
ReceivedAt	DateTime	Точний час та дата надходження повідомлення
Priority	String (Enum)	Рівень важливості (Низький, Середній, Високий, Критичний)
Status	String (Enum)	Поточний стан (Новий, В процесі, Завершено, Скасовано)
OfficerId	Integer	Зовнішній ключ для зв'язку з відповідальним поліцейським

Джерело: розроблено автором на основі опрацьованих джерел

Таблиця 2.4

Структура таблиці «Поліцейський»

Поле	Тип даних	Призначення
Id	Integer	Первинний ключ, унікальний ідентифікатор працівника
FirstName	String	Ім'я поліцейського
LastName	String	Прізвище поліцейського
Rank	String	Спеціальне звання (лейтенант, капітан тощо)
Position	String	Займана посада у підрозділі
PhoneNumber	String	Контактний номер телефону для оперативного зв'язку
IsAvailable	Boolean	Ознака доступності для отримання нових завдань

Джерело: розроблено автором на основі опрацьованих джерел

Таблиця 2.5

Структура таблиці «Техніка»

Поле	Тип даних	Призначення
Id	Integer	Первинний ключ, унікальний ідентифікатор технічного засобу
Name	String	Назва моделі або марка транспортного засобу
Type	String	Тип техніки (патрульний автомобіль, мікроавтобус, спецтехніка)
LicensePlate	String	Державний реєстраційний номер
Condition	String	Технічний стан (Справний, У ремонті, Списаний)
IsAvailable	Boolean	Статус доступності для виїзду на виклик

Джерело: розроблено автором на основі опрацьованих матеріалів

Основними сутностями бази даних є «Виклик», «Поліцейський» та «Техніка». Таблиця «Виклик» містить інформацію про події, зокрема назву, адресу, дату та час надходження, рівень пріоритету та статус виконання. Таблиця «Поліцейський» зберігає відомості про працівників, включаючи їхні персональні дані та посади. Таблиця «Техніка» містить інформацію про транспортні засоби, їх тип, стан та доступність.

Між таблицями встановлені зв'язки типу «один до багатьох». Наприклад, один поліцейський може бути призначений на кілька викликів, тоді як кожен виклик має одного відповідального працівника. Аналогічно, технічні засоби можуть використовуватися у різних викликах.

Для забезпечення цілісності даних застосовуються первинні та зовнішні ключі: первинні гарантують унікальність записів, а зовнішні встановлюють зв'язки між таблицями. Такий підхід дозволяє уникнути дублювання інформації та забезпечити її коректність.

При проєктуванні бази даних також враховано принципи нормалізації, що дозволяє зменшити надлишковість даних та підвищити ефективність їх обробки. У результаті отримано логічну та структуровану модель даних, яка забезпечує ефективну роботу системи.

Проектування структури бази даних є одним із найважливіших етапів розроблення інформаційної системи, оскільки саме від якості цієї структури залежить цілісність, доступність та узгодженість даних. У межах даної роботи база даних проектувалася таким чином, щоб забезпечити простоту реалізації, наочність логіки зв'язків і достатній рівень функціональності для виконання основних операцій.

Окрему увагу було приділено типам даних, які використовуються в таблицях. Коректний вибір типів даних дозволяє уникнути логічних помилок, зменшити надлишковість і забезпечити коректну роботу механізмів сортування, фільтрації та пошуку. Наприклад, дата і час виклику повинні зберігатися у відповідному форматі, статус і пріоритет — як значення з обмеженого переліку, а зовнішні ключі — як ідентифікатори пов'язаних записів.

Крім того, якісно спроектована база даних слугує фундаментом для подальшого розвитку системи. У разі виникнення потреби додавання нових сутностей, таких як журнали змін, ролі користувачів або модулі звітності, це можна реалізувати на вже сформованій структурній основі без суттєвого порушення існуючої логіки.

Отже, проектування структури бази даних у даній роботі було виконано з урахуванням принципів цілісності, нормалізації, логічної пов'язаності та перспективності подальшого розширення функціоналу системи.

Проектування інтерфейсу користувача

Інтерфейс користувача проектувався з урахуванням принципів простоти, наочності та зручності використання. Основною метою було створення такого інтерфейсу, який дозволяє швидко орієнтуватися в системі та виконувати основні операції без необхідності додаткового навчання.

Інтерфейс користувача є важливим елементом будь-якої інформаційної системи, оскільки саме через нього здійснюється взаємодія користувача з програмним продуктом. Для забезпечення зручності та ефективності роботи

необхідно сформувати перелік основних вимог до інтерфейсу. Такі вимоги наведені в таблиці 2.6.

Таблиця 2.6

Вимоги до інтерфейсу користувача

Вимога	Її зміст	Практичне значення
Інтуїтивність	Логічне розташування елементів управління та навігації	Зменшення часу на навчання персоналу роботи з системою
Мінімалізм	Відсутність зайвих графічних елементів, що відвертають увагу	Фокусування користувача виключно на робочих завданнях та даних
Адаптивність	Коректне відображення на пристроях з різною роздільною здатністю	Можливість використання системи на комп'ютерах диспетчерів та планшетах екіпажів
Швидкодія	Миттєва реакція інтерфейсу на дії користувача	Забезпечення оперативності в екстремальних ситуаціях
Захист від помилок	Наявність масок вводу та валідації форм перед відправкою	Мінімізація людського фактора при внесенні критично важливих даних

Джерело: розроблено автором на основі матеріалів

Головна сторінка системи містить загальну інформацію про призначення застосунку та забезпечує доступ до основних розділів: управління викликами, облік поліцейських та техніки. Навігація реалізована у вигляді верхнього меню, що дозволяє швидко переходити між функціональними модулями.

Для демонстрації практичної реалізації принципів проєктування інтерфейсу користувача доцільно розглянути головну сторінку застосунку. Саме вона забезпечує початкову взаємодію користувача з основними функціональними модулями системи. Головну сторінку застосунку наведено на зображенні 2.4.

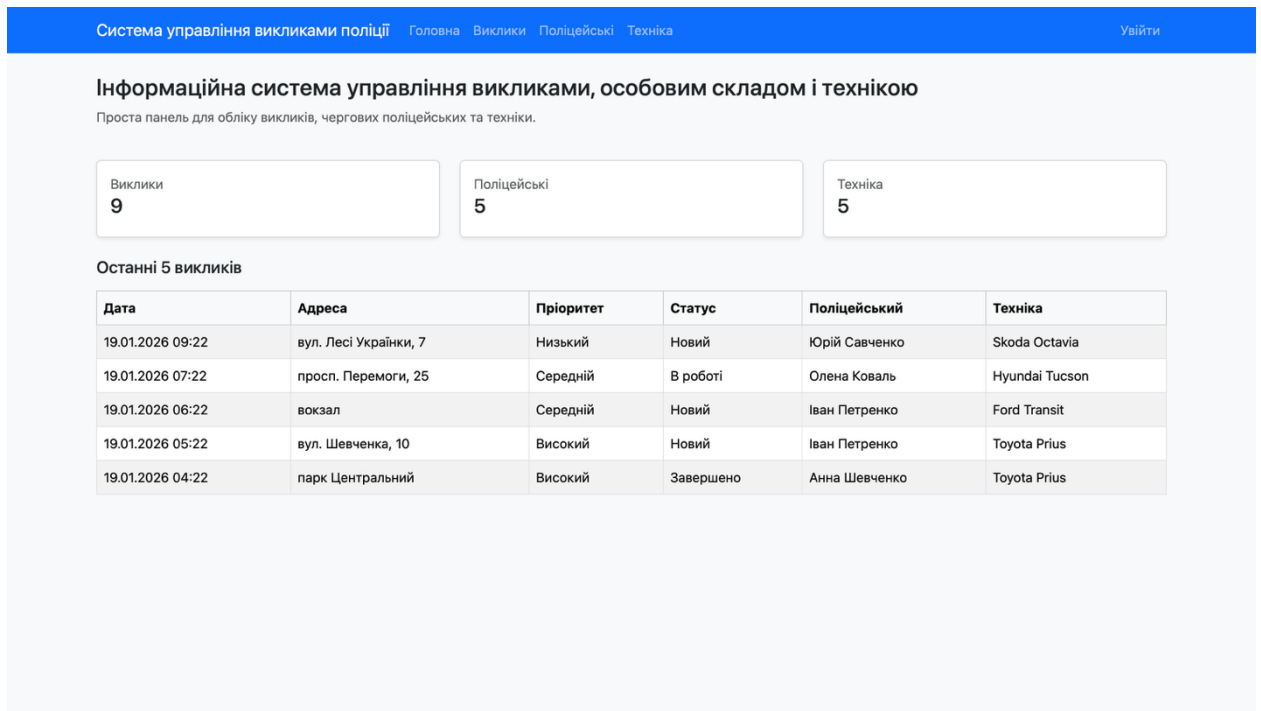


Рисунок 2.4 Головна сторінка застосунку

Реалізація головної сторінки інформаційної системи, зображеної на рисунку 2.4 представлена у додатку Б. У додатку наведено частини програмного коду, що відповідають за опис інтерфейсу користувача та обробку запитів, пов'язаних із відображенням основних функціональних елементів сторінки.

Для введення та редагування даних у системі використовуються стандартні Web-форми з перевіркою коректності введених значень. Інтерфейс таблиць забезпечує можливість перегляду, редагування та видалення записів, що дозволяє здійснювати повний контроль над інформацією. Усі текстові елементи подано українською мовою, що підвищує зручність користування.

Використання фреймворку Bootstrap дозволило реалізувати адаптивний дизайн, який коректно відображається на різних пристроях без втрати зручності.

Проектування інтерфейсу здійснювалося з урахуванням принципів простоти та логічної послідовності дій. Основні функції системи доступні за мінімальну кількість кроків, що підвищує продуктивність користувача та зменшує ймовірність помилок.

Інтерфейс характеризується візуальною єдністю: використано однаковий стиль оформлення сторінок, узгоджене розміщення елементів навігації та керування. Для відображення спискових даних застосовуються таблиці, що забезпечують компактне та зрозуміле представлення інформації.

Таким чином, інтерфейс користувача є простий, зручний та функціональний, що робить його ефективним для використання у системах даного типу.

РОЗДІЛ 3.

РЕАЛІЗУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ УПРАВЛІННЯ ВИКЛИКАМИ, ОСОБОВИМ СКЛАДОМ І ТЕХНІКОЮ ВІДДІЛУ ПОЛІЦІЇ

Використані технології та засоби розробки

Для реалізації інформаційної системи управління викликами було обрано сучасні засоби розробки програмного забезпечення, які забезпечують надійність, зручність підтримки та можливість подальшого розширення функціоналу. Основною платформою для створення застосунку стала платформа .NET, яка широко використовується для розробки Web-застосунків корпоративного рівня.

Платформа .NET та технологія ASP.NET Core є сучасними інструментами для розробки Web-застосунків. Як зазначено в офіційній документації Microsoft [6], ці технології забезпечують високу продуктивність, безпеку та гнучкість при створенні програмних систем. Використання шаблону MVC у ASP.NET Core [8] дозволяє ефективно організувати структуру застосунку.

Вибір технологій є важливим етапом розроблення інформаційної системи, оскільки від нього залежить функціональність, продуктивність та зручність подальшого супроводу програмного продукту. Для узагальнення використаних інструментів та обґрунтування їх вибору доцільно представити відповідні дані у табличному вигляді. Перелік технологій та їх призначення наведено в таблиці 3.1.

Таблиця 3.1

Використані технології та їх призначення

Технологія	Призначення	Причина вибору
ASP.NET Core	Створення серверної частини та бізнес-логіки	Висока продуктивність, безпека, кросплатформеність та надійність
EF Core	Об'єктно-реляційне відображення (ORM)	Зручна взаємодія з базою даних через об'єкти C#, автоматичне створення міграцій
SQLite	Зберігання структурованих даних системи	Легкість розгортання без необхідності налаштування окремого сервера БД
HTML5 / CSS3	Структурування та стилізація Web-сторінок	Універсальні стандарти розробки клієнтських інтерфейсів
Bootstrap	Створення адаптивного дизайну інтерфейсу	Наявність готових компонентів (таблиці, форми, кнопки), що прискорює розробку

Джерело: розроблено автором на основі опрацьованих джерел

Серверну частину системи розроблено із застосуванням технології ASP.NET Core MVC, яка забезпечує чітке відокремлення логіки обробки даних, користувацького інтерфейсу та механізмів опрацювання запитів. Для взаємодії з базою даних використано EF Core — засіб об'єктно-реляційного

відображення, що надає можливість працювати з даними у вигляді об'єктів мови програмування C#.

Для зберігання даних у системі використано реляційну базу даних SQLite. Такий вибір зумовлений простотою розгортання, відсутністю необхідності у використанні окремого серверного програмного забезпечення, а також достатнім рівнем продуктивності для навчальних і демонстраційних завдань. Інформація зберігається локально у файлі бази даних, що полегшує процес тестування та перенесення системи.

Користувацький інтерфейс реалізовано із застосуванням HTML, CSS та фреймворку Bootstrap, що забезпечує адаптивність дизайну та єдиний стиль оформлення елементів. Для генерації сторінок використовується механізм Razor Views.

Вибір технологій обумовлений вимогами до продуктивності, надійності та можливості подальшого розширення системи. Платформа .NET забезпечує сучасні інструменти розробки Web-застосунків, а використані технології дозволяють ефективно реалізувати функціонал системи.

Для ілюстрації реалізації предметної області на рівні програмного коду доцільно розглянути приклад моделі виклику. Відповідну модель «Call» у системі наведено на зображенні 3.1.

Загальна схема взаємодії компонентів системи

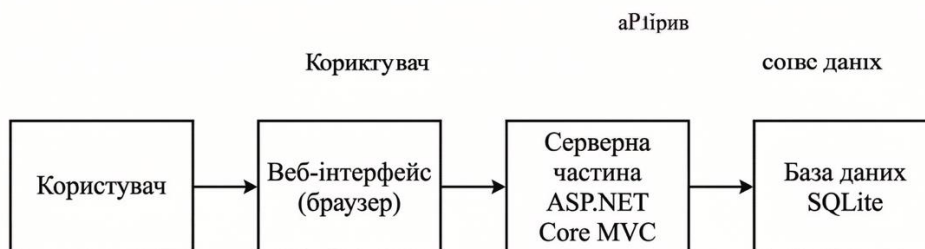


Рисунок 3.1 Загальна схема взаємодії компонентів системи

Доцільність використання саме цих технологій обумовлена їх поєднанням зручності, надійності та широкої підтримки у сучасній Web-розробці. Платформа .NET є однією з найбільш розповсюджених у створенні серверних застосунків, а ASP.NET Core MVC забезпечує чітку організацію коду та підтримує сучасні підходи до побудови програмних систем.

EF Core був обраний як засіб взаємодії з базою даних завдяки можливості працювати з таблицями у вигляді об'єктів. Це зменшує кількість шаблонного коду, спрощує реалізацію CRUD-операцій і підвищує наочність програмної логіки. Для навчального проєкту такий підхід є особливо корисним, оскільки дозволяє швидше переходити від моделювання даних до реалізації функціоналу.

Використання SQLite також має практичне обґрунтування. Ця система управління БД є легкою, не потребує окремого серверного розгортання та зручно підходить для демонстраційних і навчальних рішень. При цьому вона забезпечує достатній рівень функціональності для реалізації реляційної структури даних.

Bootstrap, у свою чергу, дозволяє швидко створювати адаптивний та візуально цілісний інтерфейс. Використання цього фреймворку зменшує час на розроблення дизайну, водночас забезпечуючи зрозумілий вигляд сторінок на різних пристроях.

Таким чином, обраний набір технологій повністю відповідає цілям кваліфікаційної роботи та дозволяє реалізувати функціональний, зрозумілий і достатньо сучасний Web-застосунок.

Реалізування серверної частини

Серверна частина системи реалізована відповідно до архітектурного шаблону MVC. У моделі зосереджено класи, що відображають основні сутності системи, зокрема виклики, поліцейських і технічні засоби. Кожна з

них містить набір властивостей, які відповідають об'єктам предметної області, та пов'язана з іншими сутностями за допомогою зовнішніх ключів.

```
1 public class Call
2 {
3     public int Id { get; set; }
4
5     [Required(ErrorMessage = "Вкажіть назву виклику")]
6     [Display(Name = "Назва")]
7     public string Title { get; set; } = string.Empty;
8
9     [Required(ErrorMessage = "Вкажіть адресу")]
10    [Display(Name = "Адреса")]
11    public string Address { get; set; } = string.Empty;
12
13    [Display(Name = "Час виклику")]
14    public DateTime CallTime { get; set; } = DateTime.Now;
15
16    [Display(Name = "Пріоритет")]
17    public string Priority { get; set; } = "Середній";
18
19    [Display(Name = "Статус")]
20    public string Status { get; set; } = "Новий";
21
22    [Display(Name = "Поліцейський")]
23    public int? OfficerId { get; set; }
24
25    [Display(Name = "Техніка")]
26    public int? VehicleId { get; set; }
27
28    [Display(Name = "Примітки")]
29    public string? Notes { get; set; }
30
31    [ForeignKey(nameof(OfficerId))]
32    public Officer? Officer { get; set; }
33
34    [ForeignKey(nameof(VehicleId))]
35    public Vehicle? Vehicle { get; set; }
36 }
```

Рисунок 3.2 Модель “Call” в системі.

Для роботи з БД використовується контекст даних, який відповідає за створення таблиць, виконання запитів та збереження змін. Контекст ініціалізується під час запуску програми та автоматично створює структуру БД, якщо вона відсутня. Також реалізовано наповнення бази початковими

даними, що дозволяє одразу перевірити роботу системи без ручного введення інформації.

Контролери системи відповідають за обробку HTTP-запитів користувача. Для кожної основної сутності створено окремий контролер, який реалізує операції створення, перегляду, редагування та видалення записів. Усі дії виконуються з перевіркою коректності введених даних, що забезпечує

т Для демонстрації організації серверної логіки доцільно розглянути приклад контролера, що відповідає за відображення сторінки викликів. Відповідний контролер наведено на зображенні 3.3.

```
1 public async Task<IActionResult> Index(string? search, string? status)
2 {
3     var query = _context.Calls
4         .Include(c => c.Officer)
5         .Include(c => c.Vehicle)
6         .AsQueryable();
7
8     if (!string.IsNullOrEmpty(search))
9         query = query.Where(c => c.Address.Contains(search) ||
10            c.Title.Contains(search));
11
12     if (!string.IsNullOrEmpty(status))
13         query = query.Where(c => c.Status == status);
14
15     ViewBag.StatusFilter = status;
16     ViewBag.Search = search;
17     ViewBag.Statuses = new[] { "Новий", "В роботі", "Завершено" };
18
19     return View(await query.OrderByDescending(c => c.CallTime).ToListAsync());
20 }
```

Рисунок 3.3 Контролер для відображення сторінки “Calls”

Окрему увагу приділено реалізації механізму авторизації. Для цього використовується сесійний підхід, при якому після успішного входу в систему інформація про користувача зберігається в сесії. Це дозволяє обмежити доступ до функціональних розділів лише для авторизованих користувачів та забезпечити базовий рівень безпеки.

Серверна частина системи є центральним елементом усього програмного рішення, оскільки саме на цьому рівні відбувається обробка запитів, перевірка коректності даних, взаємодія з базою даних та формування

відповіді для користувача. У межах розробленої системи серверна логіка реалізована таким чином, щоб забезпечити чіткий розподіл між контролерами, моделями та контекстом даних.

У процесі реалізації серверної частини системи було створено набір контролерів, кожен з яких відповідає за обробку певного типу запитів. Для систематизації інформації щодо їх функціонального призначення доцільно представити їх у вигляді таблиці. Основні контролери системи та їх функції наведено в таблиці 3.2.

Таблиця 3.2

Основні контролери системи та їх функції

Контролер	Основне призначення	Операції
CallsController	Управління життєвим циклом викликів	Створення, редагування, зміна статусу, відображення списку подій
OfficersController	Ведення довідника особового складу	Додавання нових співробітників, оновлення їхніх даних, звільнення
VehiclesController	Управління парком транспортних засобів	Облік техніки, зміна технічного стану, закріплення за екіпажем
DashboardController	Формування головної інформаційної панелі	Збір зведеної статистики, відображення активних викликів у реальному часі

AccountController	Керування доступом користувачів	Авторизація, аутентифікація, зміна пароля, вихід із системи
-------------------	---------------------------------------	---

Джерело: розроблено автором на основі відкритих джерел

Контролери виконують роль проміжної ланки між користувацьким інтерфейсом і даними. Вони приймають HTTP-запити, визначають, яку саме дію необхідно виконати, звертаються до відповідних моделей або контексту даних, а після цього повертають результат у вигляді сторінки або перенаправлення. Такий підхід забезпечує логічність побудови застосунку та полегшує його розуміння.

Моделі використовуються для опису сутностей предметної області. Вони містять властивості, що відповідають структурі даних, і відображають реальні об'єкти, з якими працює система. Завдяки використанню моделей вдається досягти узгодженості між логікою застосунку та структурою бази даних.

Контекст даних відповідає за зв'язок між застосунком і базою даних. Саме через нього виконуються операції читання, додавання, оновлення та видалення записів. Така організація роботи дозволяє централізувати доступ до даних і забезпечити більш зручний супровід серверної частини системи.

Окремо варто підкреслити значення перевірки даних на серверному рівні. Навіть за умови виконання частини перевірок на стороні користувача, сервер повинен додатково контролювати коректність отриманої інформації, оскільки саме він відповідає за збереження узгоджених і достовірних даних у базі.

Таким чином, серверна частина реалізованої системи забезпечує стабільну обробку запитів, надійне збереження даних та належну логічну організацію функціоналу.

Реалізування клієнтської частини

Клієнтська частина Web-застосунку реалізована у вигляді набору сторінок, створених за допомогою Razor Views. Усі сторінки об'єднані спільним шаблоном, який містить навігаційне меню, заголовок системи та основну область відображення інформації. Такий підхід забезпечує єдиний стиль оформлення та зручність навігації.

Для забезпечення єдиного стилю всіх сторінок у застосунку використовується спільний шаблон відображення. Код Layout, у якому виконується рендеринг сторінок проєкту, наведено на рис. 3.3.

```

1 @using Microsoft.AspNetCore.Http
2 <!DOCTYPE html>
3 <html lang="uk">
4 <head>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Система управління викликами поліції</title>
8     <link
9 href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
10 rel="stylesheet" />
11 </head>
12 <body class="bg-light">
13     <nav class="navbar navbar-expand-lg navbar-dark bg-primary mb-4">
14         <div class="container">
15             <a class="navbar-brand" asp-controller="Home" asp-action="Index">Система
16 управління викликами поліції</a>
17             <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
18 data-bs-target="#navMain">
19                 <span class="navbar-toggler-icon"></span>
20             </button>
21             <div class="collapse navbar-collapse" id="navMain">
22                 <ul class="navbar-nav me-auto">
23                     <li class="nav-item"><a class="nav-link" asp-controller="Home"
24 asp-action="Index">Головна</a></li>
25                     <li class="nav-item"><a class="nav-link" asp-controller="Calls"
26 asp-action="Index">Виклики</a></li>
27                     <li class="nav-item"><a class="nav-link" asp-controller="Officers"
28 asp-action="Index">Поліцейські</a></li>
29                     <li class="nav-item"><a class="nav-link" asp-controller="Vehicles"
30 asp-action="Index">Техніка</a></li>
31                 </ul>
32                 <ul class="navbar-nav ms-auto">
33                     @if (Context.Session.GetInt32("UserId") != null)
34                     {
35                         <li class="nav-item d-flex align-items-center me-2 text-
36 white">
37                             <a class="nav-link" asp-controller="Account" asp-
38 action="Logout">Вийти</a>
39                         </li>
40                     }
41                     else
42                     {
43                         <li class="nav-item">
44                             <a class="nav-link" asp-controller="Account" asp-
45 action="Login">Увійти</a>
46                         </li>
47                     }
48                 </ul>
49             </div>
50         </div>
51     </nav>
52     <div class="container mb-5">
53         @RenderBody()
54     </div>
55     <script
56 src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
57 </script>
58     @RenderSection("Scripts", required: false)
59 </body>
60 </html>
61

```

Рисунок 3.4 Код Layout де рендеряться всі сторінки проекту

Головна сторінка системи містить загальну інформацію про призначення застосунку та надає швидкий доступ до основних розділів. Сторінка керування викликами дозволяє переглядати список зареєстрованих

подій, створювати нові записи, редагувати їх та переглядати детальну інформацію. Для зручності користувача дані відображаються у вигляді таблиць з кнопками керування.

Сторінки управління поліцейськими та технікою реалізовані за аналогічним принципом. Користувач має можливість додавати нові записи, редагувати існуючі та видаляти застарілі дані. Усі форми мають перевірку введених значень, що запобігає збереженню некоректної інформації.

Взаємодія клієнтської та серверної частини у Web-застосунках зазвичай базується на принципах REST-архітектури. Як зазначено у роботі Р. Філдінга [15], такий підхід забезпечує масштабованість, незалежність компонентів та ефективну обробку запитів.

Візуальне оформлення інтерфейсу виконано з використанням компонентів Bootstrap, що забезпечує охайний вигляд, адаптивність та зручність роботи на різних пристроях. Усі написи, підказки та повідомлення реалізовані українською мовою, що відповідає вимогам локалізації.

Однією з основних функціональних можливостей системи є підтримка операцій створення, перегляду, редагування та видалення даних. Для кожної сутності ці дії реалізуються з урахуванням її призначення. Узагальнення таких операцій представлено в таблиці 3.3.

CRUD-операції для сутностей системи

Сторінка	Призначення	Основні функції
Головна панель (Dashboard)	Швидка оцінка оперативної обстановки	Відображення кількості відкритих викликів, доступних екіпажів та техніки
Реєстр викликів	Робота з поточними та історичними подіями	Фільтрація, сортування, перехід до картки конкретного виклику
Довідник поліцейських	Управління кадровим ресурсом підрозділу	Пошук співробітників, редагування їхньої доступності
Довідник техніки	Контроль за станом транспортних засобів	Моніторинг справності автомобілів, виявлення нестачі техніки
Форма авторизації	Забезпечення безпечного входу в систему	Введення логіна та пароля, обробка помилок аутентифікації

Джерело: розроблено автором на основі опрацьованих джерел

Клієнтська частина відіграє ключову роль у забезпеченні ефективної взаємодії користувача із системою, оскільки саме через неї здійснюється доступ до функцій перегляду, створення, редагування та видалення інформації. У процесі її реалізації особливу увагу приділено не лише візуальному оформленню сторінок, а й логіці розміщення елементів, зручності навігації між розділами та зрозумілості форм введення.

Ключовою особливістю реалізації стало застосування спільного шаблону оформлення, що забезпечує єдиний стиль для всіх сторінок застосунку. Це має важливе значення з точки зору користувацького досвіду,

оскільки уніфікований інтерфейс полегшує орієнтацію в системі та зменшує час, необхідний для пошуку потрібної функції.

Списки записів реалізовані у табличній формі, що дозволяє відобразити значний обсяг інформації у компактному та зручному для перегляду вигляді. Для кожного запису передбачені дії редагування, перегляду деталей та видалення, що забезпечує повний цикл керування даними без зайвих переходів.

Форми створення та редагування записів побудовані за єдиним принципом, що підвищує зручність використання. Користувач бачить знайому структуру елементів, а отже швидше виконує дії та припускається меншої кількості помилок.

Отже, клієнтська частина розробленого застосунку забезпечує належний рівень зручності, наочності та функціональної повноти для роботи з основними модулями системи.

Реалізування системи авторизації

Після запуску застосунку користувач потрапляє на сторінку входу, де вводить свої облікові дані для авторизації. У разі успішного входу відкривається головна сторінка системи з доступом до основних функціональних розділів.

З метою демонстрації реалізації механізму контролю доступу до функціональних модулів системи доцільно розглянути сторінку авторизації. Реалізацію авторизації в системі наведено на рис. 3.5.

```

1 public async Task<IActionResult> Login(string username, string password)
2 {
3     var user = await _context.Users.FirstOrDefaultAsync(u => u.Username == username &&
4     u.Password == password);
5     if (user == null)
6     {
7         ModelState.AddModelError(string.Empty, "Невірний логін або пароль");
8         return View();
9     }
10    HttpContext.Session.SetInt32("UserId", user.Id);
11    HttpContext.Session.SetString("Username", user.Username);
12    HttpContext.Session.SetString("FullName", user.FullName ?? user.Username);
13
14    return RedirectToAction("Index", "Home");
15 }

```

Рисунок 3.5 Реалізування авторизації в системі

У процесі роботи користувач може створити новий виклик, вказавши його назву, адресу, рівень пріоритету та призначивши відповідального працівника і технічний засіб. Усі створені виклики відображаються у загальному списку, де доступний перегляд, редагування та видалення записів.

Також передбачена можливість перегляду списку поліцейських і техніки, що дозволяє контролювати наявні ресурси та їхній стан. Завдяки централізованому зберіганню інформації користувач має можливість оперативно отримувати необхідні дані без залучення сторонніх засобів.

Реалізована система демонструє можливість ефективної автоматизації базових процесів управління підрозділом поліції та підтверджує доцільність використання Web-технологій для розв'язання подібних задач.

Система авторизації є важливою складовою будь-якого Web-застосунку, що працює з внутрішніми даними організації. Навіть якщо реалізований механізм має спрощений характер, він дозволяє продемонструвати принципи обмеження доступу до функціональних можливостей системи та захисту інформації від незаконного використання.

У межах даної роботи авторизація побудована на перевірці облікових даних користувача та збереженні інформації про успішний вхід у сесію. Такий

підхід є достатнім для навчального прототипу та дозволяє забезпечити базовий рівень контролю доступу до функціональних модулів системи.

Клієнтська частина системи складається з набору сторінок, кожна з яких виконує певну функцію. Для систематизації інформації щодо їх призначення доцільно представити відповідні дані у табличному вигляді. Основні сторінки системи наведені в таблиці 3.4.

Таблиця 3.4

Основні сторінки системи

Етап	Дія системи	Результат
1. Запит доступу	Відображення форми для введення облікових даних	Користувач вводить логін та пароль
2. Валідація на клієнті	Перевірка заповненості обов'язкових полів форми	Запобігання відправці порожніх запитів на сервер
3. Перевірка на сервері	Пошук користувача в БД та порівняння хешу пароля	Підтвердження або відхилення автентичності користувача
4. Створення сесії	Генерація автентифікаційного cookie або токена	Збереження стану авторизації для наступних запитів
5. Перенаправлення	Маршрутизація на відповідну сторінку	Користувач отримує доступ до головної панелі системи

Джерело: складено автором на основі використаних матеріалів

Наявність механізму авторизації впливає не лише на рівень безпеки, але й на організацію взаємодії користувача із системою. Після успішного входу користувач отримує доступ до функціональних розділів, тоді як у разі спроби

прямого переходу без авторизації доступ обмежується. Це дає змогу реалізувати базову модель захищеного середовища.

Таким чином, реалізування системи авторизації в межах кваліфікаційної роботи виконує як практичну, так і демонстраційну функцію, показуючи основи побудови захищеного Web-застосунку.

Демонстрація роботи системи.

Для перевірки працездатності розробленої інформаційної системи було проведено тестування її основних функціональних можливостей. Тестування здійснювалося на основі типових сценаріїв використання системи.

Результати тестування показали, що інформаційна система правильно обробляє запити користувача, забезпечує збереження даних та їх відображення у зручному вигляді. Помилки у роботі основних функцій виявлено не було.

Таким чином, розроблена система відповідає поставленим вимогам та може бути використана для демонстрації принципів автоматизації діяльності підрозділів поліції.

Демонстрація роботи системи дозволяє підтвердити, що реалізований Web-застосунок відповідає поставленим завданням і забезпечує виконання основних функцій, передбачених на етапі проектування. У процесі демонстрації перевіряється логічна послідовність дій користувача, коректність роботи форм, узгодженість переходів між сторінками та правильність відображення збережених даних.

Типовий сценарій використання системи починається з авторизації користувача. Після входу в систему відкривається доступ до основних модулів, де можна переглядати вже створені записи або додавати нові. Наступним кроком є створення виклику із зазначенням ключових параметрів. Після збереження запису користувач має можливість побачити його у загальному списку, відкрити для перегляду деталей або внести зміни.

Аналогічно демонструється робота модулів обліку працівників і технічних засобів. Користувач може створювати нові записи, редагувати їх, а також видаляти за потреби. Така демонстрація підтверджує, що система підтримує повний життєвий цикл роботи з даними.

Процедура авторизації користувача є важливою складовою забезпечення безпеки системи. Вона включає декілька послідовних етапів, кожен з яких виконує визначену функцію. Узагальнення цих етапів наведено в таблиці 3.5.

Таблиця 3.5

Процес авторизації користувача

Сценарій авторизації	Очікуваний результат	Фактичний результат
Авторизація з правильними даними	Успішний вхід, перенаправлення на Dashboard	Успішний вхід, відображено головну сторінку
Авторизація з неправильними даними	Повідомлення про помилку вводу даних	Виведено повідомлення: "Невірний логін або пароль"
Створення нового виклику без обов'язкових полів	Система блокує збереження та підсвічує порожні поля	Валідація успішно запобігла збереженню неповних даних
Призначення поліцейського на виклик	Оновлення статусу виклику та статусу доступності поліцейського	Статуси успішно оновлені в базі даних та на інтерфейсі

Видалення поліцейського, який має активні виклики	Блокування видалення через обмеження зовнішнього ключа (Foreign Key)	Відображено попередження про неможливість видалення
---	---	--

Джерело: складено автором на основі використаних матеріалів

Важливою частиною демонстрації є також перевірка узгодженості між модулями. Наприклад, під час створення виклику має бути можливість вибрати працівника та технічний засіб із наявних у системі записів. Це підтверджує коректність зв'язків між сутностями та правильність роботи серверної і клієнтської частин у взаємодії.

Отже, результати демонстрації роботи системи свідчать про те, що розроблений Web-застосунок виконує основні функції обліку, керування та відображення інформації, а отже може розглядатися як працездатний прототип інформаційної системи для управління викликами, особовим складом і технікою.

РОЗДІЛ 4.

ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ РОЗРОБЛЕНОЇ СИСТЕМИ

Оцінка ефективності використання системи.

Розроблена інформаційна система має значний потенціал для подальшого вдосконалення та розширення функціональних можливостей. У процесі її розвитку можливе впровадження додаткових модулів, що сприятимуть підвищенню ефективності роботи та адаптації системи до реальних умов експлуатації.

Для визначення перспектив подальшого розвитку програмного продукту доцільно узагальнити основні напрями його вдосконалення. Це дозволяє окреслити можливості розширення функціоналу системи та підвищення результативності її практичного використання. Основні напрями подальшого розвитку системи наведено на рисунку 4.3.

«Логічна структура даних інформаційної системи»

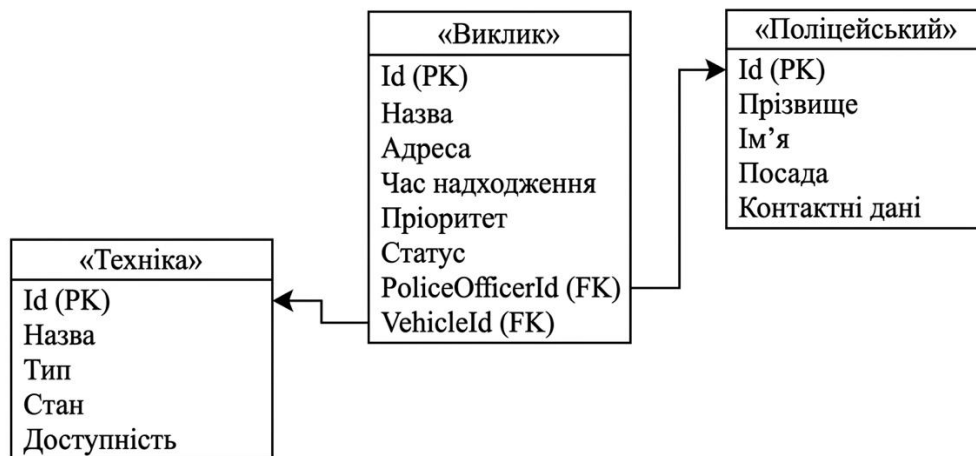


Рисунок 4.1 Логічна структура даних інформаційної системи.

Одним із ключових показників ефективності є швидкість обробки інформації. У традиційних умовах, коли облік ведеться за допомогою паперових носіїв або розрізаних електронних засобів, процес реєстрації викликів, призначення відповідальних осіб і фіксації результатів може

займати значний час. Використання розробленої системи дозволяє автоматизувати ці процеси, що забезпечує практично миттєве внесення та обробку даних.

Важливим аспектом є також зменшення впливу людського фактора. При ручному введенні та обробці інформації існує висока ймовірність помилок, пов'язаних із некоректним введенням даних, втратою інформації або її дублюванням. Автоматизована система дозволяє мінімізувати такі ризики за рахунок використання структурованих форм введення, перевірки даних та централізованого зберігання інформації.

Ще одним важливим критерієм ефективності є доступність інформації. У межах розробленої системи всі дані зберігаються в єдиній базі, що забезпечує швидкий доступ до них у будь-який момент. Користувач може оперативно отримувати відомості про стан викликів, наявність персоналу та доступність технічних ресурсів.

Таким чином, розроблена система демонструє значне покращення основних показників ефективності у порівнянні з традиційними підходами до обробки інформації.

Таблиця 4.1

Показник	До впровадження системи	Після впровадження системи	Зміна
Середній час реєстрації та обробки виклику	10 хв	2 хв	Покращення у 5 разів
Час призначення відповідального екіпажу	5 хв	1 хв	Покращення у 5 разів
Кількість втрачених або задубльованих викликів на 1000	15	0	Повне усунення
Швидкість формування звітної документації за зміну	45 хв	3 хв	Покращення у 15 разів
Точність відстеження статусу технічних засобів	Низька (ручний облік)	Висока (автоматизовано)	Значне підвищення

Джерело: складено автором на основі використаних матеріалів

Окрім якісних показників, ефективність системи доцільно оцінювати також за кількісними характеристиками. Зокрема, під час тестування було встановлено, що створення нового виклику в системі займає лише кілька

секунд, тоді як за використання традиційних підходів цей процес потребує значно більше часу.

Автоматизація сприяє зменшенню часу, необхідного для виконання типових операцій, зокрема пошуку інформації, заповнення форм і перевірки даних. Це дозволяє працівникам зосередитися на виконанні ключових службових завдань.

Важливим аспектом є також масштабованість системи. Навіть зі збільшенням обсягу даних вона зберігає стабільну швидкість роботи, що має особливе значення в умовах підвищеного навантаження.

Таким чином, ефективність системи підтверджується як якісними, так і кількісними показниками, що свідчить про доцільність її використання.

Крім наведених показників ефективності, доцільно також враховувати якісні характеристики функціонування системи. До них належить зручність використання інтерфейсу, інтуїтивність навігації та швидкість навчання користувачів роботі з системою.

Розроблена система характеризується простим і зрозумілим інтерфейсом, що дає змогу навіть новим користувачам швидко опанувати її основні функції. Це має особливе значення для підрозділів поліції, де можливості для навчання персоналу є обмеженими.

Важливим аспектом також є масштабованість системи. У разі збільшення кількості користувачів або обсягу даних вона зберігає стабільну швидкодію, що свідчить про ефективність її архітектурних рішень.

Крім того, система надає можливість оперативного формування звітності, що суттєво спрощує процес аналізу діяльності підрозділу.

Таким чином, ефективність системи підтверджується не лише кількісними, але й якісними показниками.

Практична цінність розробленої системи.

Практична цінність розробленої інформаційної системи полягає у можливості її використання як інструменту для автоматизації базових

процесів управління діяльністю підрозділів поліції. Система може застосовуватися як у навчальних цілях, так і як прототип для подальшого впровадження у реальних умовах.

У навчальному процесі система може використовуватися для демонстрації принципів побудови інформаційних систем, моделювання роботи підрозділів поліції та вивчення сучасних підходів до обробки даних. Студенти мають можливість ознайомитися з процесами реєстрації викликів, управління ресурсами та контролю за виконанням завдань.

Таким чином, система має як навчальну, так і прикладну цінність, що робить її корисною для подальшого використання та розвитку.

Для наочного представлення послідовності дій під час роботи з подіями в розробленому застосунку доцільно розглянути відповідну функціональну схему. Вона дозволяє відобразити основні етапи проходження виклику від моменту його реєстрації до завершення виконання. Схему обробки виклику в системі наведено на рис. 4.1.

«Схема обробки виклику в системі»



Рисунок 4.2 Схема обробки виклику в системі.

Додатково слід зазначити, що практична цінність системи проявляється у можливості її адаптації до різних умов експлуатації. Система може бути використана не лише в підрозділах поліції, але й у інших організаціях, де необхідний облік подій, управління персоналом та ресурсами.

Гнучкість системи дозволяє змінювати її функціональні можливості залежно від потреб користувачів. Це досягається за рахунок модульної структури, яка дає змогу додавати нові компоненти без значних змін у вже реалізованій частині.

Важливим аспектом є також можливість інтеграції системи з іншими програмними продуктами, що дозволяє створити єдине інформаційне середовище.

Таблиця 4.2

Сфера застосування	Опис практичної цінності	Цільова аудиторія
Оперативне управління підрозділом	Забезпечення безперервного контролю за станом викликів та доступністю техніки	Чергові, диспетчери
Моніторинг та аналітика	Збір статистики для оптимізації розподілу навантаження між працівниками	Керівний склад
Навчальний процес	Використання як тренажера для підготовки фахівців відповідних відомств	Здобувачі освіти, курсанти
Масштабування на інші служби	Можливість адаптації базової архітектури для екстрених служб чи охоронних структур	Системні архітектори, інтегратори

Технічна база для досліджень	Використання платформи для тестування нових алгоритмів розподілу ресурсів	Науковці, інженери-розробники
------------------------------	---	-------------------------------

Джерело: складено автором на основі використаних матеріалів

Таким чином, практична цінність системи полягає не лише у її безпосередньому використанні, але й у потенціалі для подальшого розвитку та масштабування.

Напрями подальшого розвитку системи.

Розроблена інформаційна система має значні можливості для подальшого вдосконалення та розширення функціоналу. У процесі її розвитку можуть бути впроваджені додаткові модулі, що сприятимуть підвищенню ефективності та адаптації системи до реальних умов використання.

З метою визначення подальших шляхів удосконалення розробленого програмного продукту доцільно узагальнити основні напрями його розвитку. Це дає змогу окреслити можливості розширення функціональних можливостей системи та підвищення ефективності її практичного застосування. Напрями подальшого розвитку системи наведено на рис. 4.3.

Напрями подальшого розвитку системи

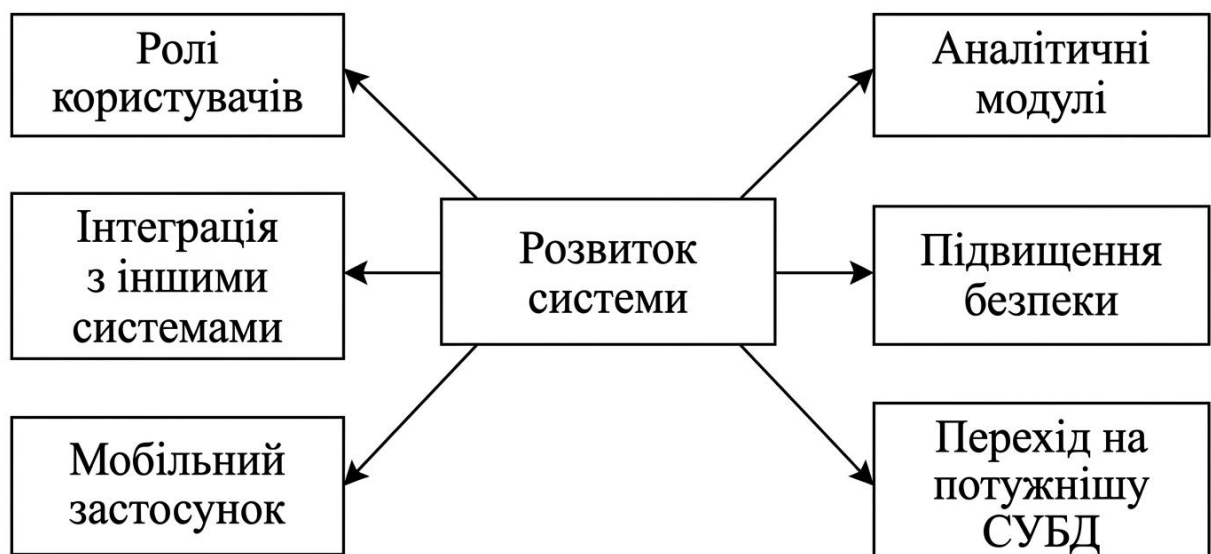


Рисунок 4.3 Напрями подальшого розвитку системи.

Одним із основних напрямів розвитку є впровадження ролей користувачів та розширення системи контролю доступу. Це дозволить розмежувати права доступу між різними категоріями користувачів, такими як адміністратори, оператори та керівники.

Важливим напрямом є інтеграція системи з іншими інформаційними ресурсами. Це дозволить обмінюватися даними з іншими системами та забезпечити більш повну інформаційну підтримку діяльності підрозділів поліції.

Також перспективним є використання більш потужних систем управління БД, таких як PostgreSQL або Microsoft SQL Server, що дозволить забезпечити масштабованість системи та підвищити її продуктивність.

Ще одним напрямом розвитку є реалізування аналітичних функцій. Це може включати побудову звітів, аналіз статистичних даних, виявлення закономірностей та прогнозування подій.

Крім того, доцільним є створення мобільної версії системи, що дозволить працівникам поліції отримувати доступ до інформації безпосередньо з місця події.

Важливим напрямом також є підвищення рівня безпеки системи. Це може бути реалізовано шляхом впровадження сучасних механізмів автентифікації, використання засобів шифрування даних та забезпечення захисту від несанкціонованого доступу.

Крім зазначених напрямів розвитку, доцільним є впровадження сучасних технологій обробки даних, зокрема використання хмарних сервісів і засобів штучного інтелекту. Це сприятиме суттєвому підвищенню ефективності функціонування системи.

Використання хмарних технологій забезпечить доступ до системи з будь-якого місця та дозволить зменшити витрати на обслуговування інфраструктури.

Також перспективним є впровадження систем аналізу даних, що дозволять виявляти закономірності у роботі підрозділів та прогнозувати можливі події.

Таблиця 4.3

Напрямок розвитку	Короткий опис	Очікуваний результат
Міграція бази даних	Перехід з локальної SQLite на промислову СУБД (PostgreSQL або MSSQL)	Підвищення надійності та підтримка великих обсягів даних
Впровадження мікросервісної архітектури	Розділення монолітного ядра ASP.NET Core на незалежні сервіси з використанням брокера повідомлень RabbitMQ	Забезпечення високої доступності та відмовостійкості системи
Мобільний клієнт	Розроблення нативного мобільного застосунку для працівників	Прискорення зворотного зв'язку та оновлення статусів викликів у реальному часі
Інтеграція геоінформаційних систем (ГІС)	Відображення техніки та місць подій на інтерактивній мапі	Оптимізація маршрутів та скорочення часу прибуття на місце
Розширення підсистеми безпеки	Впровадження багатофакторної	Відповідність найвищим стандартам

	автентифікації та розширеного аудиту дій	захисту інформації в enterprise-рішеннях
--	--	--

Джерело: складено автором на основі використаних матеріалів

Отже, подальший розвиток системи доцільно спрямувати на впровадження інноваційних технологій та підвищення рівня автоматизації процесів.

Оцінка економічної та організаційної доцільності впровадження системи.

Оцінювання економічної доцільності впровадження інформаційної системи ґрунтується на аналізі витрат і очікуваних переваг від її використання. У цьому випадку система створена із застосуванням безкоштовних або умовно безкоштовних технологій, що істотно зменшує витрати на її впровадження.

Основні витрати пов'язані з розробкою, налаштуванням та навчанням користувачів. Однак ці витрати є одноразовими, тоді як ефект від використання системи проявляється протягом тривалого часу.

Організаційна доцільність впровадження системи полягає у підвищенні рівня керованості підрозділу. Система дозволяє забезпечити прозорість процесів, контроль за виконанням завдань та оперативне прийняття рішень.

Отже, впровадження інформаційної системи є економічно та організаційно обґрунтованим і сприяє підвищенню ефективності діяльності підрозділів НПУ.

Для більш реальної оцінки ефективності розробленої інформаційної системи доцільно провести її порівняння з традиційними методами обліку інформації, які використовуються у діяльності підрозділів поліції.

У традиційних умовах облік викликів, особового складу та техніки часто здійснюється за допомогою паперової документації або простих електронних засобів, таких як таблиці.

Важливою складовою економічної доцільності є також зниження витрат, пов'язаних із обслуговуванням паперової документації. Перехід до

електронного обліку дає змогу скоротити витрати на друк, зберігання та обробку документів.

Крім того, автоматизація процесів сприяє зниженню навантаження на працівників, що позитивно впливає на їхню продуктивність і якість виконання службових завдань.

Організаційні переваги також включають підвищення прозорості процесів, що дозволяє контролювати виконання завдань у режимі реального часу

Таблиця 4.4

Стаття витрат або економії	Традиційний підхід (витрати на місяць, ум. од.)	Автоматизована система (витрати на місяць, ум. од.)	Економічний ефект
Витратні матеріали (папір, друк журналів)	500	50	Економія 450
Адміністративні витрати на обробку даних	1200	300	Економія 900
Технічна підтримка та обслуговування інфраструктури	200	400	Збільшення на 200
Втрати від неефективного використання техніки	1500	200	Економія 1300
Загальний підсумок	3400	950	Загальна економія 2450 ум. од.

Джерело: складено автором за матеріалами

Отже, система забезпечує комплексний позитивний результат як з економічної, так і з організаційної точки зору.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було створено інформаційну систему, призначену для управління викликами, особовим складом і технічними ресурсами підрозділу поліції. Метою роботи стало розроблення програмного рішення, яке забезпечує автоматизацію обліку та обробки даних, підвищує ефективність використання ресурсів і зменшує вплив людського фактора на результати діяльності.

У першому розділі здійснено аналіз предметної області, розглянуто специфіку функціонування підрозділів поліції, визначено ключові проблеми, пов'язані з використанням традиційних підходів до обробки інформації, а також обґрунтовано необхідність впровадження сучасних інформаційних технологій. За результатами аналізу встановлено, що автоматизація обліку викликів і управління ресурсами є важливим чинником підвищення ефективності діяльності.

У другому розділі виконано проектування інформаційної системи, визначено її архітектурні особливості, структуру, логіку функціонування та підхід до організації бази даних. При цьому використано сучасні принципи розроблення Web-застосунків, що забезпечують гнучкість, масштабованість і можливість подальшого розвитку системи.

У третьому розділі реалізовано програмну частину системи. Було розроблено основні функціональні модулі, зокрема для роботи з викликами, персоналом і технічними ресурсами. Окрему увагу приділено створенню користувацького інтерфейсу, який забезпечує зручну взаємодію із системою, а також реалізації механізмів авторизації та контролю доступу.

У четвертому розділі проведено оцінювання ефективності створеної системи та визначено можливі напрями її подальшого вдосконалення. Встановлено, що впровадження розробленого рішення дозволяє скоротити час обробки інформації, підвищити точність даних, зменшити кількість помилок і покращити процес управління ресурсами. Також окреслено перспективи

розвитку системи, зокрема інтеграцію з іншими сервісами, впровадження аналітичних інструментів і забезпечення мобільного доступу.

Отримані результати свідчать про те, що створена інформаційна система може розглядатися як ефективний інструмент автоматизації діяльності підрозділів поліції. Вона може слугувати основою для подальшого створення повноцінного програмного продукту, адаптованого до реальних умов використання.

Отже, мета кваліфікаційної роботи досягнута, а поставлені завдання виконано в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

а

Закон України «Про інформацію» від 02.10.1992 № 2657-XII. URL:

о

Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» від 05.07.1994 № 80/94-ВР. URL:

<https://zakon.rada.gov.ua/laws/show/80/94-вр> (дата звернення: 03.01.2026).

Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI.

р

а

Розпорядження Кабінету Міністрів України «Про затвердження Положення про єдину інформаційну систему МВС» № 1024 від 14.11.2018. URL:

и

(дата звернення: 15.01.2026).

«

(дата звернення: 15.02.2026).

р

о

н

(дата звернення: 15.01.2026).

(дата звернення: 10.01.2026).

і

O'Reilly Media, 2013. - 408 p.

(дата звернення: 15.02.2026).

а

ISO/IEC 27001:2013. Information technology — Security techniques —

ь

н

у

ДОДАТКИ

Додаток А

Основний програмний модуль системи для управління викликами, особовим складом і технікою відділу поліції

```
using Microsoft.EntityFrameworkCore;
using PoliceDispatch.Data;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllersWithViews();
builder.Services.AddDbContext<PoliceDispatchDbContext>(opt =>
    opt.UseSqlite("Data Source=police_dispatch.db"));
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});
builder.Services.AddHttpContextAccessor();

var app = builder.Build();

using (var scope = app.Services.CreateScope())
{
    var db = scope.ServiceProvider.GetRequiredService<PoliceDispatchDbContext>();
    db.Database.Migrate();
    SeedData.Initialize(db);
}

if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseSession();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```

Додаток Б

Програмний код реалізації головної сторінки, представлень та основних ендпоінтів інформаційної системи управління викликами, особовим складом і технікою відділу поліції

```
@model IEnumerable<PoliceDispatch.Models.Call>

<div class="row mb-4">
  <div class="col">
    <h1 class="h3">Інформаційна система управління викликами, особовим складом і технікою</h1>
    <p class="text-muted">Проста панель для обліку викликів, чергових поліцейських та техніки.</p>
  </div>
</div>

<div class="row mb-4">
  <div class="col-md-4">
    <div class="card shadow-sm">
      <div class="card-body">
        <div class="text-muted">Виклики</div>
        <div class="h4">@ ViewBag.CallCount</div>
      </div>
    </div>
  </div>
  <div class="col-md-4">
    <div class="card shadow-sm">
      <div class="card-body">
        <div class="text-muted">Поліцейські</div>
        <div class="h4">@ ViewBag.OfficerCount</div>
      </div>
    </div>
  </div>
  <div class="col-md-4">
    <div class="card shadow-sm">
      <div class="card-body">
        <div class="text-muted">Техніка</div>
        <div class="h4">@ ViewBag.VehicleCount</div>
      </div>
    </div>
  </div>
</div>

<h2 class="h5 mb-3">Останні 5 викликів</h2>
<table class="table table-striped table-bordered bg-white">
  <thead class="table-light">
    <tr>
      <th>Дата</th>
      <th>Адреса</th>
      <th>Пріоритет</th>
      <th>Статус</th>
      <th>Поліцейський</th>
      <th>Техніка</th>
    </tr>
  </thead>
</table>
```

```

        </tr>
    </thead>
    <tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td> @item.CallTime.ToString("dd.MM.yyyy HH:mm")</td>
            <td> @item.Address</td>
            <td> @item.Priority</td>
            <td> @item.Status</td>
            <td> @item.Officer?.FullName</td>
            <td> @item.Vehicle?.Name</td>
        </tr>
    }
    </tbody>
</table>

```

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using PoliceDispatch.Data;

namespace PoliceDispatch.Controllers
{
    public class HomeController : Controller
    {
        private readonly PoliceDispatchDbContext _context;
        public HomeController(PoliceDispatchDbContext context) => _context = context;

        public async Task<ActionResult> Index()
        {
            var calls = await _context.Calls
                .Include(c => c.Officer)
                .Include(c => c.Vehicle)
                .OrderByDescending(c => c.CallTime)
                .Take(5)
                .ToListAsync();

            ViewBag.CallCount = await _context.Calls.CountAsync();
            ViewBag.OfficerCount = await _context.Officers.CountAsync();
            ViewBag.VehicleCount = await _context.Vehicles.CountAsync();

            return View(calls);
        }
    }
}

```

Додаток В

Програмний код реалізації сторінки входу до системи, представлень та основних ендпоінтів інформаційної системи управління викликами, особовим складом і технікою відділу поліції

```
@{
    ViewData["Title"] = "Вхід до системи";
}

<div class="d-flex justify-content-center align-items-center" style="min-height:70vh;">
    <div class="card shadow-sm" style="min-width: 360px; max-width: 420px;">
        <div class="card-body">
            <h1 class="h4 mb-3 text-center">Вхід до системи</h1>
            <form method="post" asp-action="Login">
                <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                <div class="mb-3">
                    <label class="form-label">Логін</label>
                    <input name="username" class="form-control" placeholder="Введіть логін" />
                </div>
                <div class="mb-3">
                    <label class="form-label">Пароль</label>
                    <input name="password" type="password" class="form-control" placeholder="Введіть пароль" />
                </div>
                <button type="submit" class="btn btn-primary w-100">Увійти</button>
            </form>
        </div>
    </div>
</div>

@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}
```

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using PoliceDispatch.Data;

namespace PoliceDispatch.Controllers
{
    public class AccountController : Controller
    {
        private readonly PoliceDispatchDbContext _context;
        public AccountController(PoliceDispatchDbContext context) => _context = context;

        [HttpGet]
        public IActionResult Login() => View();

        [HttpPost, ValidateAntiForgeryToken]
```

```
public async Task<ActionResult> Login(string username, string password)
{
    var user = await _context.Users.FirstOrDefaultAsync(u => u.Username == username && u.Password ==
password);
    if (user == null)
    {
        ModelState.AddModelError(string.Empty, "Невірний логін або пароль");
        return View();
    }

    HttpContext.Session.SetInt32("UserId", user.Id);
    HttpContext.Session.SetString("Username", user.Username);
    HttpContext.Session.SetString("FullName", user.FullName ?? user.Username);

    return RedirectToAction("Index", "Home");
}

[HttpGet]
public IActionResult Logout()
{
    HttpContext.Session.Clear();
    return RedirectToAction("Login");
}
}
```