УДК 004

ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ДЛЯ УПРАВЛІННЯ ВЕНТИЛЯЦІЄЮ

Хижняк Д.С.
e-mail: dmytro.khyzhniak@nure.ua
Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

In modern conditions, automating indoor environment management is becoming increasingly relevant, particularly in ensuring comfort, health, and energy efficiency. This project aims to develop a ventilation management system that automatically monitors and regulates air parameters such as temperature, humidity, and CO2 levels. The system integrates IoT devices based on Raspberry Pi, ASP.NET for backend services, and a PostgreSQL database. Users can control the system via a web interface, or a mobile application built with Kotlin. Firebase Auth and Firebase Cloud Messaging are used for authentication and notifications. The system enhances air quality management while optimizing energy consumption.

У сучасних умовах автоматизація управління середовищем у приміщеннях стає все більш актуальною, особливо для забезпечення комфорту, здоров’я та енергоефективності. Ключовим аспектом є якість повітря, яка впливає на продуктивність та самопочуття людей. Погана вентиляція може призводити до накопичення CO2, підвищеної вологості та розвитку шкідливих мікроорганізмів [1]. Тому виникає потреба у створенні програмної системи управління вентиляцією, яка автоматизує моніторинг та регулювання параметрів повітря.

Метою проекту є розробка системи управління вентиляцією, яка автоматично контролює та регулює параметри повітря (температура, вологість, рівень CO2) у приміщеннях. Система повинна підтримувати оптимальні умови для комфорту та здоров’я, знижувати потребу у ручному втручанні та забезпечувати енергоефективність. Вона також повинна бути гнучкою для різних типів приміщень (житлові будинки, офіси, виробничі приміщення).

Основні функції системи включають:

1. Автоматичне регулювання: аналіз даних від IoT-сенсорів та автоматичне коригування роботи вентиляційних пристроїв.
2. Дистанційне керування: веб-інтерфейс або мобільний додаток для налаштування параметрів та моніторингу стану повітря в реальному часі.
3. Сповіщення: повідомлення користувачам у разі виходу параметрів за встановлені межі.
4. Індивідуальні налаштування: можливість задавати порогові значення для температури, вологості та рівня CO2 відповідно до потреб користувача.

Для реалізації системи використовується сучасний технологічний стек, що забезпечує надійність, масштабованість та ефективність обробки даних. Серверна частина розроблена з використанням ASP.NET для веб-сервісів, що дозволяє швидко обробляти запити та забезпечувати взаємодію між компонентами. Для управління даними використовується PostgreSQL, що забезпечує збереження, обробку та аналіз інформації про параметри повітря.

Для кешування даних і черг повідомлень використовується Redis, що допомагає зменшити навантаження на базу даних та покращити швидкодію системи. IoT-компоненти базуються на Raspberry Pi з зовнішніми модулями для підключення сенсорів, які збирають дані про температуру, вологість та рівень CO_2 у приміщеннях. Для обміну даними між пристроями та сервером використовується MQTT Broker, що дозволяє швидко передавати показники сенсорів та керувати вентиляційними пристроями.

Клієнтські частини системи включають веб-інтерфейс та мобільний додаток. Веб-клієнт розроблений на Vue.js з TypeScript і стилізований за допомогою Tailwind CSS, що забезпечує швидке завантаження сторінок та зручний користувацький інтерфейс. Мобільний додаток створений на основі Android SDK з використанням Kotlin, що дає змогу користувачам керувати системою, отримувати сповіщення та контролювати стан вентиляції зі смартфона.

Для автентифікації користувачів та розсилки сповіщень використовуються хмарні сервіси Firebase Authentication та Firebase Cloud Messaging [2]. Firebase Authentication гарантує безпечний доступ до системи, а Firebase Cloud Messaging дозволяє надсилати push-сповіщення про критичні зміни параметрів повітря.

Розгортання системи здійснюється за допомогою Docker та Docker Compose [3]. Це дозволяє ізолювати компоненти, спрощує керування залежностями та забезпечує швидке масштабування. Основні сервіси системи (API сервер, база даних, Redis, MQTT Broker) працюють у вигляді окремих контейнерів, які взаємодіють між собою через внутрішню мережу.

Для управління трафіком використовується Nginx, який виконує роль зворотного проксі-сервера, що спрямовує запити клієнтів до відповідних сервісів. Окрім цього, Nginx забезпечує балансування навантаження у разі високого трафіку та обслуговує статичні файли веб-додатка, що покращує продуктивність системи.

MQTT Broker також розгортається як окремий сервіс, що забезпечує ефективну передачу даних між IoT-пристроями та сервером у реальному часі.

Процес деплойменту автоматизований за допомогою GitHub Workflow [4]. Це забезпечує автоматичне тестування, збірку контейнерів та розгортання оновлень у production-середовищі, що підвищує надійність та зменшує ризик виникнення помилок.

Архітектура розгортання системи зображена на діаграмі розгортання (рисунок 1), яка відображає взаємодію між сервісами та клієнтськими застосунками.

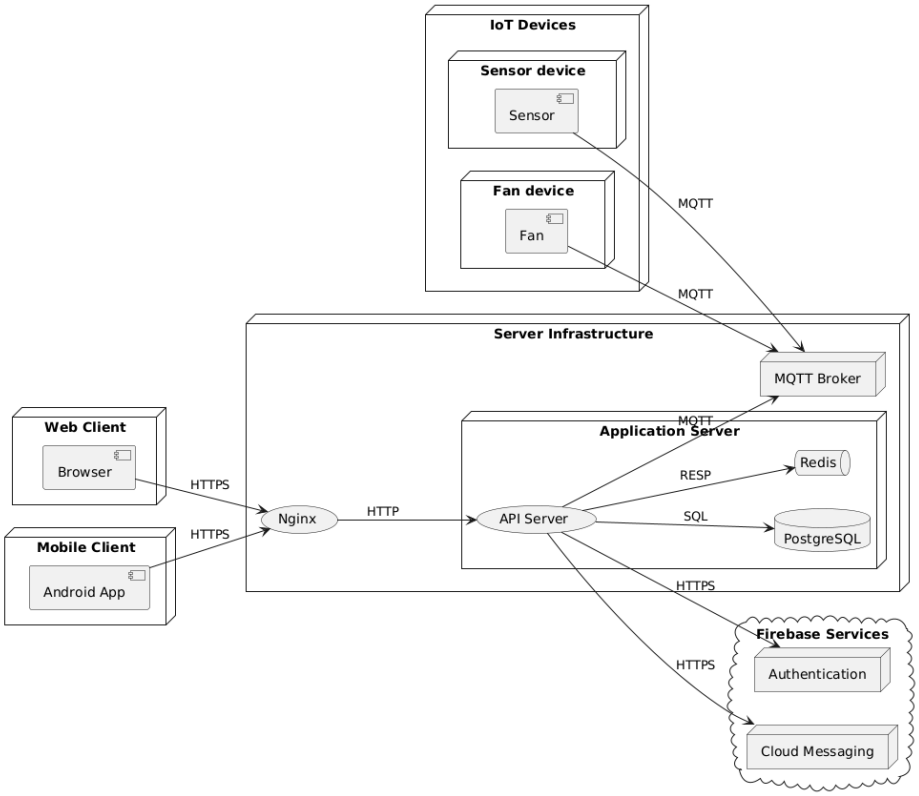

Рисунок 1 – Діаграма розгортання

Розробка програмної системи управління вентиляцією є актуальним завданням, яке дозволяє забезпечити комфортні та енергоефективні умови у приміщеннях різного типу. Використання сучасних IoT-технологій, хмарних сервісів та автоматизації дозволяє створити гнучке та зручне рішення для управління мікрокліматом. Проект має значний потенціал для розширення на ринку малих та середніх підприємств, а також для індивідуальних користувачів, які прагнуть покращити якість повітря у своїх приміщеннях.

Список використаних джерел:

1. Якість повітря та здоров’я. Всесвітня організація охорони здоров’я. URL: https://www.who.int/health-topics/air-pollution (дата звернення: 02.03.2025).

2. Firebase Documentation. Firebase. URL: https://firebase.google.com/docs (дата звернення: 02.03.2025).

3. Docker Documentation. Docker. URL: https://docs.docker.com/ (дата звернення: 02.03.2025).

4. GitHub Actions. GitHub. URL: https://docs.github.com/en/actions (дата звернення: 02.03.2025).