

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

СМАРТ СИСТЕМА КОНТРОЛЮ КЛІМАТУ

Виконала: студентка 4-го курсу
Ірина РИМЧУК

(підпис)

Науковий керівник:
професор кафедри теоретичної кібернетики
доктор фіз.-мат. наук, професор
Анатолій ПАШКО

(підпис)

Засвідчую, що в цій роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теоретичної кібернетики
« ____ » _____ 2023 р.,

протокол № ____
Завідувач кафедри
доктор фіз.-мат. наук, професор
Юрій КРАК

(підпис)

КИЇВ-2023

РЕФЕРАТ

Обсяг роботи: 55 сторінок, 23 ілюстрації, 11 джерел посилань.

Ключові слова: **МОБІЛЬНИЙ ЗАСТОСУНОК, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, БАГАТОКОРИСТУВАЦЬКА СИСТЕМА, АРІ, КЛІМАТ-КОНТРОЛЬ, РОБОТА З БАЗАМИ ДАНИХ.**

Об'єкт роботи: процес створення системи клімат-контролю.

Предмет роботи: система клімат-контролю, яка включає в себе мобільний додаток, серверну частину та прилад для контролю клімату.

Мета роботи: реалізація програмного продукту з використанням передових комп'ютерних технологій, який забезпечує контроль клімату в приміщенні.

Методи розроблення: комп'ютерне моделювання, моделювання баз даних, розробка програмного продукту.

Інструменти розроблення: інтегроване середовище розробки Visual Studio, платформа .NET Core, фреймворк .NET MAUI Blazor, СУБД MS SQL SERVER.

Результат роботи: розроблений програмний продукт для контролю клімату, який забезпечує автоматичний контроль параметрів клімату в приміщенні, включаючи температуру, вологість та якість повітря. Програмний продукт має інтуїтивний інтерфейс, доступний для користувачів через мобільний додаток, і дозволяє налаштовувати та керувати системою клімат-контролю.

Розроблена система може бути застосована в різних сферах, таких як домашнє використання, офісні приміщення та промислові об'єкти, для створення комфортних умов.

ЗМІСТ

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	5
ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ	8
1.1 Аналіз актуальності обраної теми.	8
1.2 Огляд наявних на ринку систем.....	10
1.2.1 Nest.....	11
1.2.2 Ecobee	13
1.2.3 Tado.....	15
1.3 Постановка задачі, призначення і цілі створення застосунку	17
РОЗДІЛ 2 ВИКОРИСТОВУВАНІ ПРОГРАМНІ ЗАСОБИ ТА ТЕХНОЛОГІЇ	18
2.1 Середовище розробки Visual Studio 2022	18
2.2 Платформа .NET Core.....	19
2.3 Фреймворк .NET MAUI Blazor	20
2.4 База даних MS SQL.....	21
РОЗДІЛ 3 СХЕМА СИСТЕМИ ТА КОМУНІКАЦІЯ МІЖ ЇЇ КОМПОНЕНТАМИ.....	23
3.1 Структура системи.....	23
3.2 Комунікація між компонентами системи.....	24
3.2.1 Комунікація між сервером та приладом	28
3.2.2 Комунікація між приладом та датчиками	29
3.2.3 Комунікація між сервером та мобільним додатком.....	30
РОЗДІЛ 4 ПРОЕКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ	32
4.1 Серверна частина	32
4.2 Прилад для контролю клімату	37

4.3 Мобільний додаток	39
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТОК А	51
ДОДАТОК Б	53

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API – Application Programming Interface

BL – Business Logic

DAL – Data Access Layer

MAUI – Multi-platform App UI

UART – Universal Asynchronous Receiver/Transmitter

БД – База Даних

ВСТУП

Оцінка сучасного стану розробки. Стрімкий ріст свідомості щодо енергоефективності, прагнення людей знизити витрати на енергоспоживання та зменшити негативний вплив на навколишнє середовище призвели до збільшення попиту на застосунки для контролю клімату. Оцінюючи сучасний стан об'єкта розробки - мобільного додатку для контролю клімату - можна сказати, що це перспективний продукт з високим потенціалом. Однак для створення оптимального продукту потрібно враховувати різні вимоги та потреби різних категорій користувачів.

Актуальність роботи та підстави для її виконання. У наш час спостерігається тенденція щодо розширення ринку різноманітних автоматизованих систем, зокрема програмних рішень, що забезпечують збереження енергії та допомагають людям ефективніше використовувати ресурси. Мобільні додатки для контролю клімату стають все більш популярними серед користувачів, оскільки вони дозволяють легко керувати енергоспоживанням та зменшувати витрати на опалення та кондиціонування повітря.

Мета й завдання роботи. Основною метою даної роботи є розробка на основі сучасних комп'ютерних технологій програмного продукту для контролю клімату.

Для досягнення бажаного результату потрібно виконати послідовність наступних завдань:

- проаналізувати потреби та вимоги користувачів щодо контролю клімату;
- створити інтуїтивно зрозумілий та зручний інтерфейс додатку;
- реалізувати функції моніторингу температури та вологості, вмикання та вимикання систем, регулювання температури;
- забезпечити можливість взаємодії з іншими системами домашньої автоматизації, такими як освітлення;

- передбачити можливість віддаленого керування системами опалення та кондиціонування повітря через додаток;
- розробити додаток для різних мобільних платформ, таких як iOS та Android, забезпечити доступність та зручність використання для більшої кількості користувачів.

Об'єкт, методи й засоби розроблення. Об'єктом розробки є процес створення системи для контролю клімату.

При розробці проведено детальний аналіз потреб та вимог користувачів щодо контролю клімату, включаючи функціональність, інтерфейс, безпеку тощо. Використовувались такі інструментальні засоби, бібліотеки та технології: Visual Studio 2022, .NET Core, .NET MAUI Blazor.

Можливі сфери застосування. Розроблена система може бути застосована в різних сферах, таких як домашнє використання, офісні приміщення та промислові об'єкти, для створення комфортних умов.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз актуальності обраної теми.

У сучасному світі проблема енергозбереження стала ще більш гострою, досягнення енергоефективності та збереження енергії стало глобальним пріоритетом. Постійне зростання попиту на енергію в поєднанні з виснаженням обмежених природних ресурсів і згубними наслідками деградації навколишнього середовища викликало нагальну потребу вирішити цю проблему. Вкрай важливо досліджувати ефективні стратегії та інноваційні рішення, які можуть значно зменшити споживання енергії та оптимізувати її використання. Роблячи це, ми можемо зменшити навантаження на обмежені ресурси, мінімізувати вплив на навколишнє середовище та забезпечити стає енергетичне майбутнє для прийдешніх поколінь.

Зростання популярності «розумних» пристроїв і систем домашньої автоматизації в останні роки революціонізувало спосіб нашої взаємодії з нашим житловим і робочим простором. Клімат-контроль, як одна з цих передових систем, став вирішальним компонентом у створенні комфортного та здорового середовища в приміщенні.

Системи клімат-контролю об'єднують різні технології та датчики для регулювання температури, вологості, циркуляції повітря та інших факторів, які безпосередньо впливають на якість повітря, яким ми дихаємо, і наш загальний комфорт. Активно відстежуючи та регулюючи ці змінні, системи клімат-контролю прагнуть підтримувати оптимальний клімат, який задовольняє конкретні потреби та вподобання людей.

Однією з ключових переваг систем клімат-контролю є можливість персоналізувати атмосферу в приміщенні відповідно до індивідуальних уподобань. Будь то регулювання температури до бажаного рівня, встановлення певних рівнів вологості або забезпечення належної циркуляції повітря, користувачі мають можливість адаптувати оточення для покращеного комфорту.

Крім того, системи клімат-контролю сприяють підвищенню продуктивності як у житлових, так і в комерційних умовах. Підтримуючи постійний і комфортний мікроклімат у приміщенні, ці системи мінімізують відволікання, спричинені коливаннями температури або поганою якістю повітря, дозволяючи людям зосередитися та працювати якнайкраще.

Крім того, системи клімат-контролю узгоджуються з більш широкими цілями енергоефективності та сталого розвитку. Оптимізуючи налаштування температури та керуючи споживанням енергії, ці системи допомагають зменшити загальне споживання енергії та пов'язані з цим витрати. Це узгоджується з глобальними зусиллями щодо збереження енергії, скорочення викидів вуглецю та створення більш екологічно чистих житлових і робочих просторів.

Таким чином, системи клімат-контролю стали невід'ємною частиною сучасного життя, пропонуючи людям можливість створювати індивідуальні, комфортні та здорові умови в приміщенні. Оскільки технології продовжують розвиватися, ми можемо очікувати подальших інновацій у кліматичному контролі, які підвищать енергоефективність, інтегруються з екосистемами розумного будинку та сприятимуть більш стійкому майбутньому.

У сучасному світі ефективне управління кліматичним контролем стає все більш важливим, що спонукає прагнення до енергозбереження та екологічної стійкості. Мобільні програми, розроблені спеціально для клімат-контролю, пропонують ряд переваг, які задовольняють потреби користувачів у житлових і комерційних приміщеннях.

Однією з головних переваг цих мобільних додатків є їх здатність оптимізувати використання опалення та кондиціонування повітря. Надаючи користувачам контроль у режимі реального часу та доступ до їхніх систем клімат-контролю, ці програми дають змогу людям приймати обґрунтовані рішення щодо налаштувань температури та споживання енергії. Це призводить до зменшення споживання енергії, що призводить до економії коштів і зменшення вуглецевого сліду.

Зручність і комфорт, які забезпечують системи для клімат-контролю, важко переоцінити. Користувачі мають можливість дистанційно керувати своїми системами опалення та охолодження, дозволяючи їм регулювати параметри температури, активувати режими енергозбереження та персоналізувати клімат у приміщенні відповідно до своїх уподобань і графіків. Цей рівень контролю підвищує комфорт, гарантуючи, що в житлових і робочих приміщеннях завжди підтримується бажана температура, що сприяє продуктивності та добробуту.

Крім того, ці програми використовують передові технології, такі як аналіз даних і машинне навчання для оптимізації продуктивності системи. Аналізуючи схеми споживання енергії, прогнози погоди та переваги користувачів, системи клімат-контролю можуть автоматично регулювати температурні конфігурації та графіки роботи. Ця інтелектуальна автоматизація сприяє підвищенню енергоефективності, оскільки системи адаптуються до мінливих умов і поведінки користувачів, що в кінцевому підсумку призводить до зниження витрат і переваг для навколишнього середовища.

Враховуючи зростаючий інтерес до розумних технологій, акцент на енергоефективності, прагнення до зручності та комфорту, зусилля щодо зниження витрат та прогрес Інтернету речей, очевидно, що розробка мобільного додатку для клімат-контролю є дуже актуальною і має значні перспективи.

1.2 Огляд наявних на ринку систем.

Сьогодні на ринку доступний широкий вибір систем клімат-контролю та мобільних додатків, які пропонують користувачам зручні та ефективні інструменти для керування та моніторингу параметрів клімату в приміщенні. Ці системи спрямовані на забезпечення комфорту, енергоефективності та здорового середовища в житлових і комерційних приміщеннях.

Системи клімат-контролю складаються з різних компонентів, таких як термостати, датчики температури та вологості, а також системи вентиляції та кондиціонування повітря. Ці системи можна інтегрувати з мобільними додатками,

дозволяючи користувачам дистанційно контролювати клімат у приміщенні за допомогою своїх смартфонів або планшетів.

За допомогою цих мобільних додатків користувачі можуть легко встановити бажані параметри температури, регулювати рівень вологості, контролювати циркуляцію повітря та планувати роботу системи. Вони також надають цінну інформацію про споживання енергії та показники якості повітря, дозволяючи користувачам приймати рішення щодо енергоефективності та покращення якості повітря в приміщеннях.

Крім того, ці системи клімат-контролю можна легко інтегрувати з іншими розумними пристроями в домашній або офісній мережі Інтернету речей (IoT), включаючи системи освітлення, безпеки та звуку. Це відкриває можливості для синхронізації та автоматизації різних аспектів комфорту в приміщенні та енергозбереження.

Загалом, ринок систем клімат-контролю та мобільних додатків пропонує широкий спектр інструментів, які дозволяють користувачам ефективно керувати та контролювати кліматичні умови в приміщенні у зручний спосіб. Ці технології сприяють комфорту, енергоефективності та більш здоровому середовищу життя та роботи.

1.2.1 Nest

Система клімат-контролю Nest [1], розроблена компанією Nest Labs (придбана компанією Google), широко визнана одним з найбільш популярних та інноваційних рішень на ринку. Вона пропонує широкий спектр функцій і можливостей для ефективного управління мікрокліматом у приміщенні.

Система Nest має зручний мобільний додаток (рис. 1.1), який дозволяє користувачам дистанційно керувати кліматичними параметрами в своїх будинках. За допомогою додатку користувачі можуть регулювати температуру, встановлювати режими роботи, регулювати рівень вологості та отримувати сповіщення про стан системи.



Рисунок 1.1 – Вигляд мобільного додатку Nest

Особливістю системи Nest є її здатність до самооптимізації. Вона навчається та адаптується до шаблонів активності користувача, налаштовуючи режими роботи системи на основі індивідуальних звичок та вподобань. Наприклад, система може автоматично встановлювати оптимальні температурні налаштування, коли мешканців немає вдома, що призводить до економії електроенергії.

Nest пропонує додаткові датчики, які можна стратегічно розмістити по всьому будинку для досягнення більш точного клімат-контролю. Це дозволяє користувачам налаштовувати температуру в конкретних кімнатах відповідно до своїх уподобань та вимог. Крім того, система Nest використовує інтелектуальні

алгоритми для аналізу та прогнозування кліматичних умов, забезпечуючи стабільно комфортне середовище в приміщенні.

Також дана система може легко інтегруватися з іншими системами розумного будинку, включаючи освітлення, безпеку та автоматизацію. Ця інтеграція дозволяє користувачам створити єдину систему для управління всіма аспектами домашнього середовища, сприяючи збалансованому та ефективному клімат-контролю.

Система клімат-контролю Nest вирізняється сучасним дизайном, зручним інтерфейсом та інтелектуальними функціями, які сприяють створенню комфортного клімату в домі та зниженню енергоспоживання.

Проте система Nest може бути обмежено сумісною з деякими старими системами опалення та охолодження. У випадку вибору даної системи можна зіткнутися з проблемою несумісності та необхідністю проведення додаткових модифікацій або заміни обладнання для належної роботи системи. Також деякі користувачі можуть відчувати сумніви стосовно захисту своїх персональних даних через збір та обробку особистої інформації про них та їхні звички системою Nest.

1.2.2 Ecobee

Ecobee [2] – це інноваційна та багатофункціональна система клімат-контролю, яка забезпечує комплексний контроль температури та інших параметрів у вашому домі.

Мобільний додаток Ecobee (рис. 1.2) дозволяє зручно керувати системою клімат-контролю дистанційно. Зі смартфона або планшета можна регулювати температуру, відстежувати стан системи, встановлювати режими роботи та керувати іншими параметрами.

Ecobee легко інтегрується з популярними голосовими помічниками, такими як Amazon Alexa або Google Assistant. Ця інтеграція дозволяє без зусиль керувати системою за допомогою голосових команд, забезпечуючи режим "вільні руки".

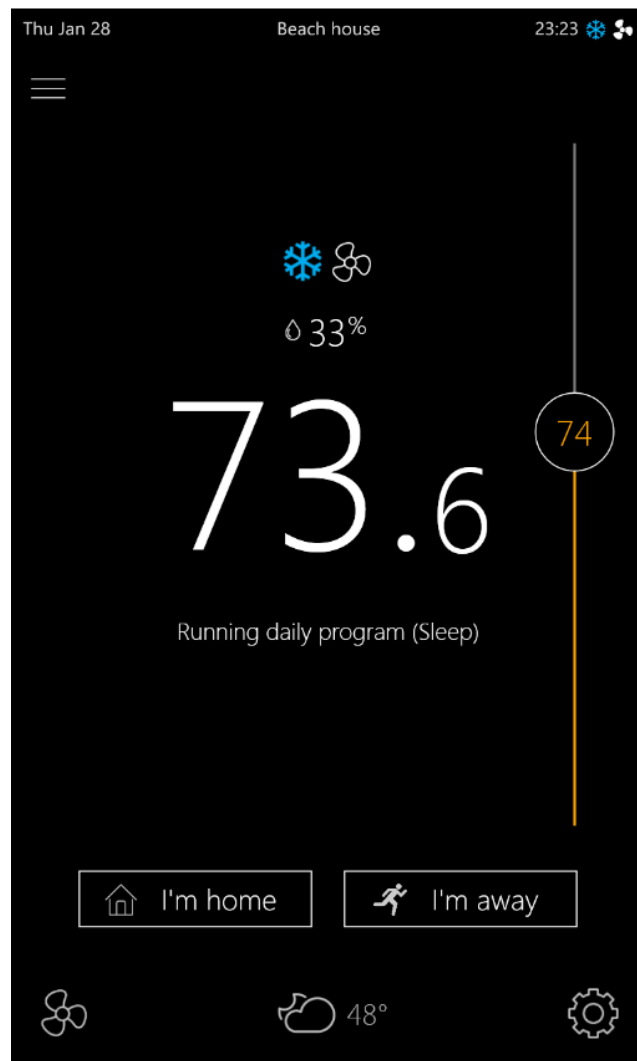


Рисунок 1.2 – Вигляд мобільного додатку Ecobee

Додаткові датчики температури та руху дозволяють системі індивідуально регулювати температуру в кожній кімнаті, забезпечуючи оптимальний комфорт та ефективне використання енергії.

Система Ecobee навчається на основі щоденних рутинних дій і звичок користувача, що дозволяє їй автоматично налаштовувати режими роботи системи. Вона розпізнає закономірності, наприклад, коли користувач зазвичай йде або приходить додому, і відповідно регулює температуру, забезпечуючи комфорт у потрібний час.

Ecobee пропонує розширені функції для ефективного управління та контролю клімату у вашому домі, забезпечуючи підвищений комфорт, енергоефективність та зручність для користувача.

Хоча система Ecobee має багато переваг, вона також має деякі недоліки, такі як вартість, складність встановлення та несумісність зі старими системами опалення або охолодження. Також для повного функціонування системи Ecobee потрібне постійне підключення до Інтернету. Це означає, що якщо з'єднання з Інтернетом вимикається або відсутнє, деякі функції системи можуть бути недоступними.

1.2.3 Tado

Tado [3] – це сучасна система керування домашнім кліматом, яка забезпечує інтелектуальні рішення для опалення, охолодження та енергозбереження.

Tado має функцію інтелектуального опалення, яка автоматично регулює опалення в домі для оптимального комфорту та енергоефективності. Вона враховує зовнішні погодні умови, щоденний графік користувача та інші фактори, щоб забезпечити точний контроль температури.

Tado використовує технологію геолокації для визначення місцезнаходження користувача та автоматично вимикає опалення, коли він виходить з дому, або вмикає його, коли він наближається. Це дозволяє ефективно використовувати енергію та забезпечує комфортні умови після його повернення.

Для керування системами опалення та охолодження Tado пропонує зручний мобільний додаток (рис. 1.3), який дозволяє налаштовувати систему з будь-якого місця. Користувач може змінювати температуру, переглядати статистику енергоспоживання, встановлювати режими роботи та багато іншого безпосередньо зі свого смартфона або планшета.

Tado легко інтегрується з різними розумними системами, такими як Amazon Alexa, Google Assistant та Apple HomeKit. Це забезпечує голосове керування та інтеграцію з іншими розумними пристроями та домашнім обладнанням.



Рисунок 1.3 – Вигляд мобільного додатку Tadoo

Також система Tado використовує можливості адаптивного керування, вивчаючи звички та спосіб життя користувача. Це дозволяє системі автоматично підлаштовуватися під його уподобання, враховуючи такі фактори, як пріоритети, прогноз погоди та інші змінні, щоб оптимізувати клімат-контроль у домі.

Tado є популярним вибором серед користувачів завдяки зручному інтерфейсу, енергоефективності та інтелектуальним функціям, спрямованим на досягнення оптимального клімат-контролю у домі.

Якщо говорити про недоліки системи, то слід зазначити, що вона може бути відносно дорогою, особливо якщо потрібно контролювати кілька зон опалення або охолодження. Більше того, доступність системи Tado може бути обмеженою у деяких регіонах або країнах. Це означає, що користувачі, які проживають за межами обслуговування системи Tado, можуть зіткнутися з труднощами у доступі до продукту.

1.3 Постановка задачі, призначення і цілі створення застосунку

Створення програми клімат-контролю має на меті вирішити існуючі проблеми та досягти конкретних цілей. Вивчивши поточні проблеми, були визначені наступні ключові завдання:

- Комфорт та зручність. Першочерговим завданням додатку є забезпечення комфортних умов у середовищі користувача. Додаток повинен сприяти легкому та зручному управлінню системою клімат-контролю, дозволяючи користувачам регулювати температуру, вентиляцію та інші параметри у зручний для них спосіб.
- Сумісність. Важливим аспектом є сумісність системи з різними пристроями та мережами зв'язку, що забезпечує широку доступність для користувачів. Це означає, що система повинна бути здатна функціонувати з різними типами пристроїв, включаючи смартфони, планшети та комп'ютери. Такий широкий спектр сумісності забезпечує зручність для користувачів, дозволяючи їм керувати системою контролю клімату з будь-якого пристрою та в будь-який час. Крім того, сумісність з іншими пристроями та мережами дозволяє інтегрувати систему контролю клімату з іншими розумними пристроями та автоматизованими системами, тим самим розширюючи її можливості та функціональність.
- Надійність. Система повинна працювати без збоїв та забезпечувати стійке встановлення налаштувань для системи опалення та охолодження.
- Зручний інтерфейс. Додаток повинен мати простий та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко орієнтуватися та налаштовувати параметри клімат-контролю. Інтуїтивно зрозумілі елементи керування, візуальне відображення стану системи та легкий доступ до налаштувань можуть покращити користувацький досвід.

Досягнувши цих цілей, додаток забезпечить ефективний і зручний клімат-контроль в будинках, гарантуючи комфорт користувачам та сприяючи енергоефективності.

РОЗДІЛ 2 ВИКОРИСТОВУВАНІ ПРОГРАМНІ ЗАСОБИ ТА ТЕХНОЛОГІЇ

2.1 Середовище розробки Visual Studio 2022

Visual Studio 2022 [4] - це найновіша версія інтегрованого середовища розробки (IDE) від Microsoft. Його мета – полегшити створення різноманітних програмних продуктів, включаючи веб-додатки, десктопні програми, мобільні додатки, ігри, хмарні сервіси та багато іншого. Ця IDE пропонує широку підтримку багатьох мов програмування, таких як C#, Visual Basic, C++, F#, JavaScript, Python, TypeScript та інших.

Visual Studio 2022 надає потужні інструменти для розробки веб-додатків, включаючи комплексну підтримку ASP.NET, HTML, CSS та JavaScript. Це дозволяє створювати та налагоджувати веб-додатки безпосередньо в IDE.

IDE дозволяє створювати мобільні додатки для платформ Android та iOS. Забезпечується підтримка таких платформ, як Xamarin, MAUI та React Native, що дає змогу розробникам створювати кросплатформні додатки.

Visual Studio 2022 включає в себе набір інструментів, які підвищують швидкість та ефективність кодування. Ці інструменти включають підказки IntelliSense, автоматичне завершення коду, можливості налагодження, системи контролю версій та багато іншого.

Завдяки інтеграції з хмарними сервісами Microsoft Azure Visual Studio 2022 забезпечує швидке розгортання, керування та масштабування додатків у хмарному середовищі.

IDE також впроваджує різноманітні покращення продуктивності, такі як прискорене завантаження проекту, прискорена компіляція та процеси налагодження.

2.2 Платформа .NET Core

.NET Core [5] – це платформа розробки програмного забезпечення, розроблена корпорацією Майкрософт, яка є безкоштовною та з відкритим кодом. Вона слугує універсальною основою для створення широкого спектру додатків, включаючи веб-додатки, сервіси хмарних обчислень, мобільні додатки, мікросервіси тощо.

За результатами щорічного дослідження Stack Overflow [6] (рис. 2.1) .Net залишається найпопулярнішою платформою в 2022 році.

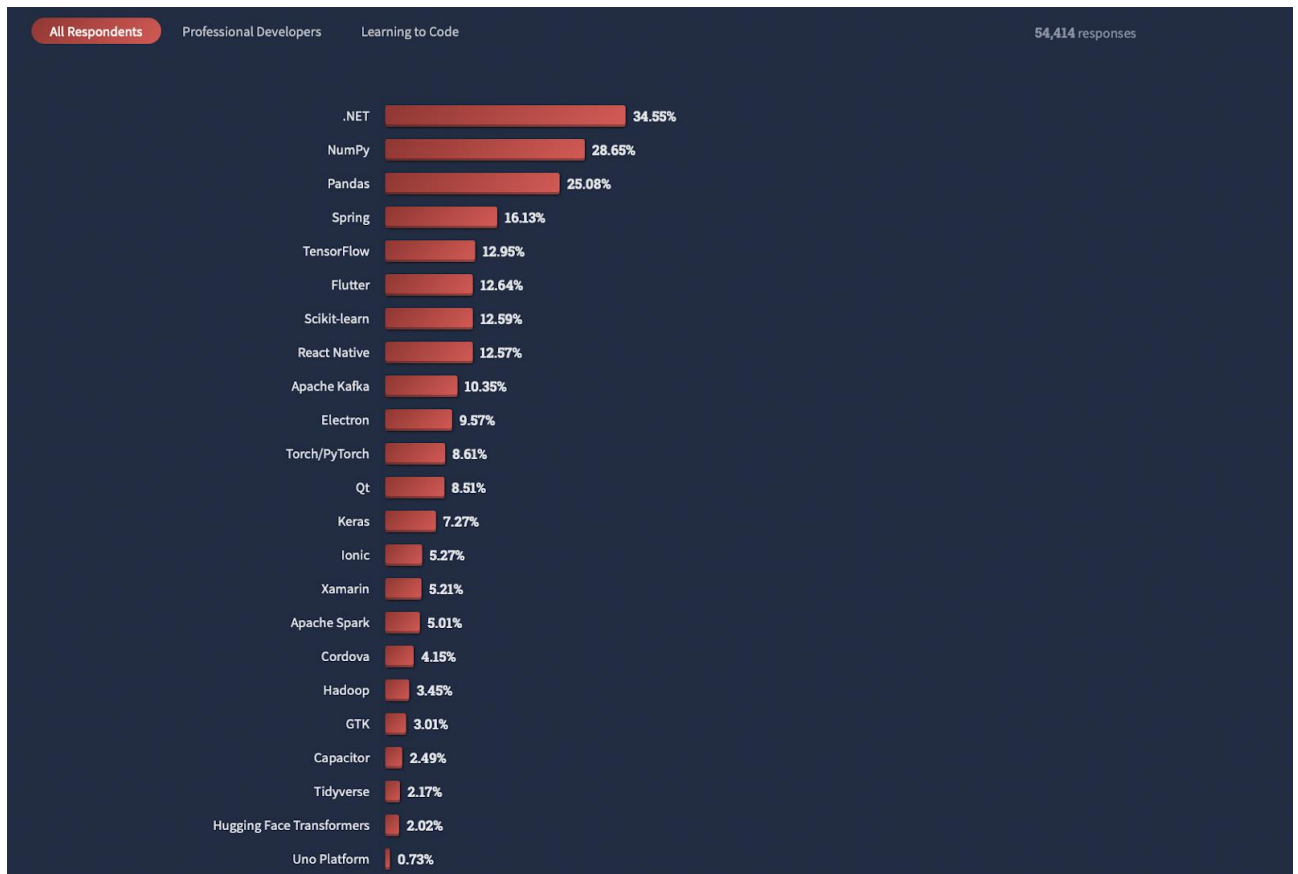


Рисунок 2.1 – Результати дослідження Stack Overflow

Платформа надає набір інструментів командного рядка, таких як dotnet CLI, для розробки, збирання, відлагодження та розгортання додатків. Ці інструменти дозволяють ефективно працювати з проектами .NET Core безпосередньо з командного рядка.

Платформа надає повний набір інструментів командного рядка, таких як dotnet CLI, які полегшують розробку, створення, налагодження та розгортання додатків. Ці інструменти дозволяють ефективно керувати робочим процесом проєктів на .NET Core безпосередньо з командного рядка.

Однією з ключових особливостей .NET Core є його модульна архітектура, що дозволяє розробникам вибірково обирати необхідні компоненти та функціональність для своїх проєктів. Такий підхід зменшує розмір програми та підвищує її ефективність, усуваючи непотрібні накладні витрати. Завдяки оптимізованому виконуваному коду та ефективній обробці, .NET Core дозволяє створювати високопродуктивні додатки, які ефективно використовують системні ресурси, що призводить до швидшої роботи та швидкої реакції програмного забезпечення.

Модульність фреймворку .NET Core дозволяє завантажувати необхідні компоненти веб-додатків як окремі модулі через менеджер пакетів NuGet. Це дозволяє розробникам легко встановлювати, оновлювати та керувати програмними залежностями, що використовуються в їхніх проєктах. NuGet забезпечує безперешкодну інтеграцію сторонніх бібліотек і компонентів у проєкт .NET Core.

Крім того, .NET Core пропонує надійні можливості інтеграції з хмарними сервісами, зокрема з Microsoft Azure. Ця безшовна інтеграція дозволяє використовувати хмарну функціональність у своїх додатках, наприклад, сховища, сервіси штучного інтелекту тощо. .NET Core легко інтегрується з різними інструментами розробки, включаючи Visual Studio, Visual Studio Code, JetBrains Rider та інші.

2.3 Фреймворк .NET MAUI Blazor

.NET MAUI Blazor [7] – це розширення фреймворку .NET MAUI, що об'єднує можливості як крос-платформного фреймворку .NET MAUI, так і веб-фреймворку Blazor. За допомогою .NET MAUI Blazor можна створювати додатки, які можуть працювати на різних платформах, таких як Android, iOS, macOS та веб-браузери.

Процес розробки в .NET MAUI Blazor використовує мову програмування C#, використовуючи знайомі та потужні інструменти C# для створення клієнтських веб-додатків.

Побудований на основі фреймворку Blazor, .NET MAUI Blazor використовує компонентний підхід для створення веб-інтерфейсів. Це дозволяє створювати багаторазові компоненти та сприяє динамічній інтерактивності на стороні клієнта. Фреймворк надає повний набір вбудованих елементів керування та розмітки інтерфейсу, включаючи стандартні елементи керування, такі як кнопки, списки та тексти, а також гнучкість у створенні користувацьких елементів керування.

Однією з визначних особливостей .NET MAUI Blazor є підтримка автономної роботи. Це означає, що додатки можуть продовжувати функціонувати і зберігати дані навіть без підключення до Інтернету. Ця функція особливо цінна для мобільних додатків, які можуть мати обмежений або переривчастий зв'язок.

Крім того, .NET MAUI Blazor оптимізовано для швидкості та продуктивності, що забезпечує ефективне виконання та безперебійну роботу додатків, розроблених за допомогою фреймворку.

2.4 База даних MS SQL

Microsoft SQL Server (MS SQL Server) [8] – це провідна система управління базами даних, розроблена компанією Microsoft. Вона пропонує надійну платформу для зберігання, управління та обробки даних. MS SQL Server складається з декількох компонентів, включаючи сервер баз даних, менеджер конфігурації, мову запитів T-SQL (Transact-SQL), а також інструменти для розробки та адміністрування баз даних.

Ця система дозволяє ефективно обробляти великі обсяги даних і підтримує складні структури баз даних. Вона використовує реляційну модель даних і підтримує різні типи даних, такі як числа, рядки, дати та бінарні дані. У реляційній моделі дані зберігаються у вигляді таблиць, що складаються з рядків і стовпців. Кожен рядок представляє окремий об'єкт, а атрибути цього об'єкта розміщуються

у відповідних стовпцях. Кожен рядок у таблиці ідентифікується за допомогою первинного ключа. Первинний ключ може бути одним або кількома стовпцями. За допомогою первинного ключа ми можемо посилатися на конкретний рядок у таблиці. Дві таблиці можуть бути пов'язані між собою за допомогою ключів, створюючи зв'язки. Таблиця може бути представлена у вигляді відношення (relation).

Запити до бази даних MS SQL Server виконуються за допомогою мови T-SQL, яка надає повний набір команд для маніпулювання даними (створення, читання, оновлення, видалення) та управління базами даних.

MS SQL Server забезпечує цілісність, узгодженість і безпеку даних завдяки підтримці транзакцій. Транзакції гарантують атомарність, цілісність, унікальність та ізолюваність операцій з даними, забезпечуючи коректність та узгодженість даних навіть тоді, коли з базою даних одночасно працюють декілька користувачів. Безпека даних також є пріоритетом, і MS SQL Server пропонує функції контролю доступу до бази даних, автентифікації, авторизації та шифрування для забезпечення конфіденційності та цілісності даних.

Оптимізація запитів - ще один важливий аспект MS SQL Server. Він включає в себе різні механізми, такі як кешування, індексування та паралельне виконання, для підвищення продуктивності запитів до бази даних.

MS SQL Server також надає можливості реплікації, що дозволяє створювати копії баз даних для розподіленого використання та резервні копії для захисту даних і відновлення після збоїв. Крім того, він легко інтегрується з іншими продуктами Microsoft, що дозволяє зручно використовувати базу даних MS SQL Server у різних сценаріях розробки та експлуатації, співпрацюючи з такими інструментами, як Visual Studio, Azure, SharePoint та іншими.

РОЗДІЛ 3 СХЕМА СИСТЕМИ ТА КОМУНІКАЦІЯ МІЖ ЇЇ КОМПОНЕНТАМИ

3.1 Структура системи

Смарт система контролю клімату зазвичай складається з трьох основних компонентів: приладу для клімат-контролю, мобільного додатка та сервера. Кожен з цих компонентів виконує свої функції для забезпечення ефективного та зручного управління кліматичними параметрами приміщення.

Для кращого розуміння роботи системи було побудовано структурну схему комунікації (рис. 3.1).

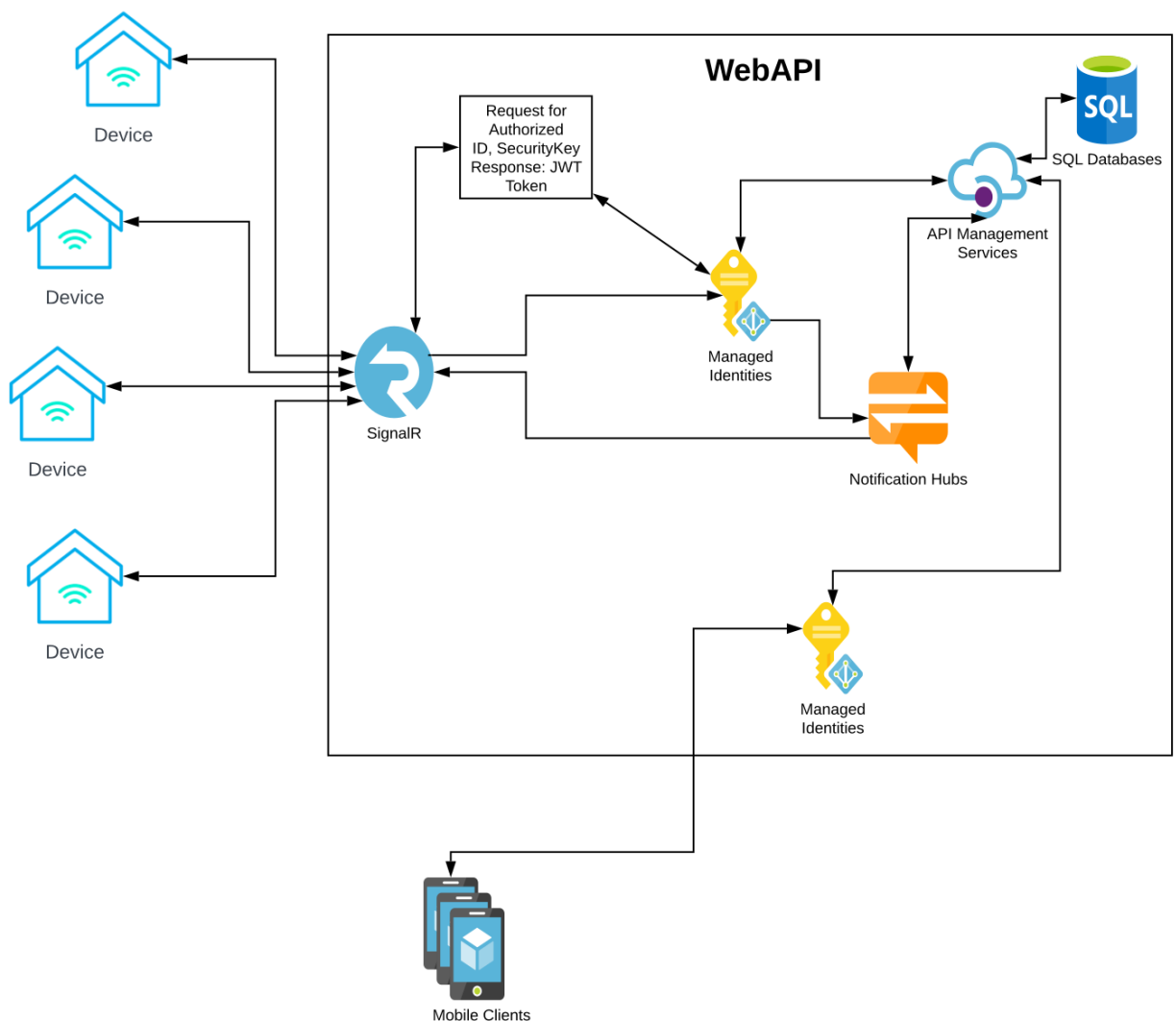


Рисунок 3.1 – Структура системи клімат-контролю

Прилад має датчики температури, вологості та якості повітря. Він здатний отримувати дані з датчиків і виконувати дії, такі як вмикання/вимикання систем опалення, охолодження або вентиляції. Він також здатний взаємодіяти з іншими пристроями у системі "розумного будинку", в моєму випадку з освітленням.

Мобільний додаток забезпечує зручний інтерфейс для користувача для взаємодії зі смарт системою контролю клімату. Користувач може налаштовувати та керувати параметрами клімату, такими як температура, вологість, якість повітря, а також отримувати інформацію про стан приміщення. Мобільний додаток може також надавати функції дистанційного управління, що дозволяє користувачу керувати системою з будь-якого місця через Інтернет.

Сервер – центральна складова смарт системи контролю клімату. Сервер зберігає та обробляє дані, отримані від пристрою і мобільного додатка. Він координує роботу системи та забезпечує комунікацію між пристроєм і мобільним додатком.

Ці три компоненти працюють разом для створення смарт системи контролю клімату, яка дозволяє користувачу зручно та ефективно керувати кліматом у своєму приміщенні. Пристрій зчитує дані з датчиків, мобільний додаток надає користувачу інтерфейс для управління, а сервер забезпечує зберігання, обробку та координацію даних.

3.2 Комунікація між компонентами системи

У наш час розвиток інтелектуальних систем клімат-контролю вимагає ефективного зв'язку між їхніми різними компонентами. Протоколи зв'язку відіграють ключову роль у забезпеченні зв'язку та взаємодії між цими компонентами, що дозволяє передавати дані, керувати режимами роботи та забезпечувати оптимальну функціональність системи контролю клімату.

При виборі протоколів для системи клімат-контролю вирішальне значення має ретельний аналіз конкретних вимог застосування та різних факторів. Одним з основних факторів є масштаб системи. У випадках, коли система складається з

численних компонентів і передбачає передачу значних обсягів даних, протокол, що забезпечує достатню швидкість передачі і масштабованість, набуває важливого значення.

Надійність зв'язку - ще один важливий фактор, на який слід звернути увагу. Системи клімат-контролю не терплять помилок при передачі даних, оскільки вони можуть призвести до неправильних дій і, в кінцевому підсумку, погіршити якість клімат-контролю. Тому обрані протоколи повинні забезпечувати надійну передачу даних і включати механізми перевірки цілісності.

Енергоспоживання є додатковим аспектом, який слід враховувати, особливо при використанні бездротових пристроїв або мобільних додатків в системі клімат-контролю. Енергоефективні протоколи можуть значно подовжити час автономної роботи пристроїв і зменшити загальні витрати на енергоспоживання.

Основними способами комунікації в моїй реалізації системи контролю клімату є HTTP, SignalR та Serial COM Port.

Протокол передачі гіпертексту (HTTP) [9] – це широко використовуваний протокол зв'язку в Інтернеті, який полегшує передачу гіпертекстових документів, між клієнтом і сервером. HTTP є безстандартним протоколом, розглядаючи кожен клієнтський запит як незалежну сутність без попереднього знання про попередні запити. Заснований на моделі клієнт-сервер, HTTP передбачає, що клієнти (наприклад, веб-браузери або мобільні додатки) надсилають запити до серверів, які, в свою чергу, відповідають на ці запити. Запити і відповіді містять різну інформацію, таку як заголовки, дані, статуси тощо. На рисунку 3.2 представлено схему моделі клієнт-сервер.

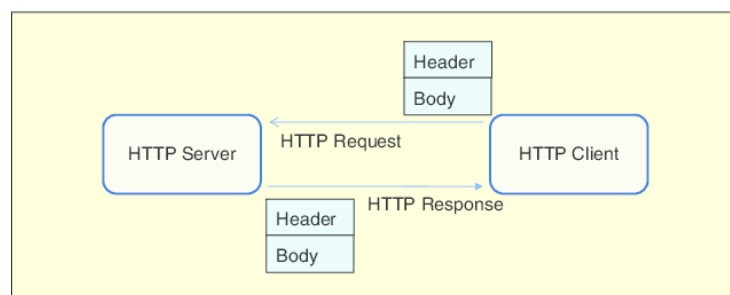


Рисунок 3.2 – Комунікація по HTTP

HTTP використовує різні методи запитів для виконання таких дій, як отримання ресурсу (GET), надсилання даних для обробки на сервер (POST), оновлення ресурсу (PUT або PATCH) або видалення ресурсу (DELETE). Ці методи забезпечують взаємодію і виконання необхідних дій між клієнтом і сервером.

HTTP може передавати дані в різних форматах, включаючи текст, зображення, відео, аудіо та інші, що робить його універсальним і придатним для передачі різних типів інформації.

Хоча HTTP пропонує простоту використання, широкую підтримку і полегшує комунікацію між компонентами системи, він має обмеження щодо швидкості передачі даних і певні проблеми з безпекою.

SignalR [10] – це бібліотека з відкритим вихідним кодом, розроблена компанією Microsoft, яка забезпечує двосторонній зв'язок між клієнтськими програмами та сервером у режимі реального часу. Вона використовує протокол WebSocket та інтелектуально обирає найбільш оптимальний метод передачі даних між клієнтом і сервером на основі підтримки різних транспортних протоколів.

Ключова перевага SignalR полягає в його здатності встановлювати постійне з'єднання між клієнтом і сервером (рис. 3.3). Це забезпечує миттєву передачу даних в режимі реального часу без необхідності постійних запитів, ініційованих клієнтом. Таким чином, SignalR особливо добре підходить для реалізації таких функцій, як сповіщення, чати, оновлення статусів та інших додатків, що працюють в режимі реального часу.

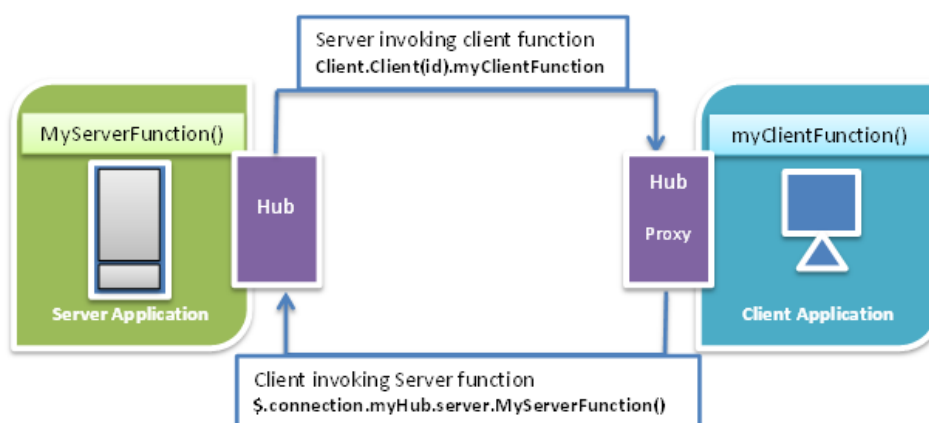


Рисунок 3.3 – Приклад комунікації за допомогою SignalR

SignalR підтримує крос-платформну розробку, що робить його сумісним з серверами Windows і такими платформами, як .NET і .NET Core. Крім того, він дозволяє впроваджувати клієнтську частину на широкому спектрі платформ, включаючи веб, мобільні пристрої та настільні комп'ютери.

SignalR пропонує комплексний набір функцій, що включає групову комунікацію (полегшує обмін повідомленнями між усіма підключеними клієнтами в певній групі), передачу файлів, автоматичне відновлення з'єднання після переривання мережі, механізми аутентифікації та авторизації, можливості масштабування та багато іншого.

Serial COM Port (COM-порт) [11] – це комунікаційний протокол, який використовується для послідовної передачі даних між комп'ютером або іншими пристроями та периферійними пристроями через фізичний інтерфейс RS-232 або USB. Він знаходить широке застосування для встановлення зв'язку з різними пристроями, включаючи датчики, контролери, модеми, адаптери зберігання даних та інші пристрої, які покладаються на послідовний зв'язок.

Serial COM Port забезпечує послідовну передачу даних, де кожен біт представляє сигнал "0" або "1". Для налаштування процесу передачі даних використовуються різні параметри, такі як швидкість передачі, біти даних, парність, стопові біти та керування потоком. Для забезпечення успішного зв'язку ці параметри повинні бути налаштовані однаково на обох пристроях.

Протокол Serial COM Port має низку переваг, серед яких зручність для користувача, широке розповсюдження, сумісність з багатьма пристроями та надійна передача даних. Він особливо цінний у випадках, коли необхідний надійний послідовний зв'язок на невеликих відстанях. Однак він також має свої обмеження, такі як обмежена швидкість передачі даних, обмежена довжина кабелю, а також певні обмеження щодо багатокористувацької підтримки та асинхронного зв'язку.

У системах клімат-контролю послідовний COM-порт можна використовувати для встановлення зв'язку з датчиками та іншими пристроями, що відповідають за передачу даних, пов'язаних з температурою, вологістю, рухом і

різними іншими параметрами. Він дозволяє отримувати ці дані з датчиків і полегшує їх передачу на сервер для аналізу, обробки і подальшого управління системою клімат-контролю.

3.2.1 Комунікація між сервером та приладом

Зв'язок між сервером і пристроєм в системі клімат-контролю відбувається за допомогою низки протоколів. Він починається з авторизації пристрою за допомогою HTTP-запиту, і після успішної авторизації пристрою присвоюється унікальний токен для подальшого забезпечення безпеки комунікації (рис. 3.4).

Згодом сервер і пристрій встановлюють нове з'єднання за допомогою SignalR, що забезпечує двосторонній зв'язок у режимі реального часу. SignalR використовує технологію WebSockets для передачі даних між сервером і приладом, що дозволяє отримувати оновлення та повідомлення миттєво без необхідності постійно виконувати запити на сервер. Для забезпечення безпечного зв'язку реалізовано аутентифікацію за допомогою JWT (JSON Web Token). Після успішної авторизації пристрою сервер генерує унікальний токен, який пристрій використовує для ідентифікації при кожному наступному запиті. Це забезпечує безпечну передачу даних і надає контрольований доступ до функцій системи клімат-контролю.

Використовуючи з'єднання SignalR, сервер може надсилати повідомлення та команди на пристрій, що дозволяє виконувати такі дії, як зміна режимів роботи, регулювання налаштувань тощо. У відповідь пристрій передає дані про стан системи, вимірювання датчиків та іншу важливу інформацію назад на сервер. Такий обмін інформацією в режимі реального часу дозволяє серверу отримувати актуальні дані про кліматичні умови.

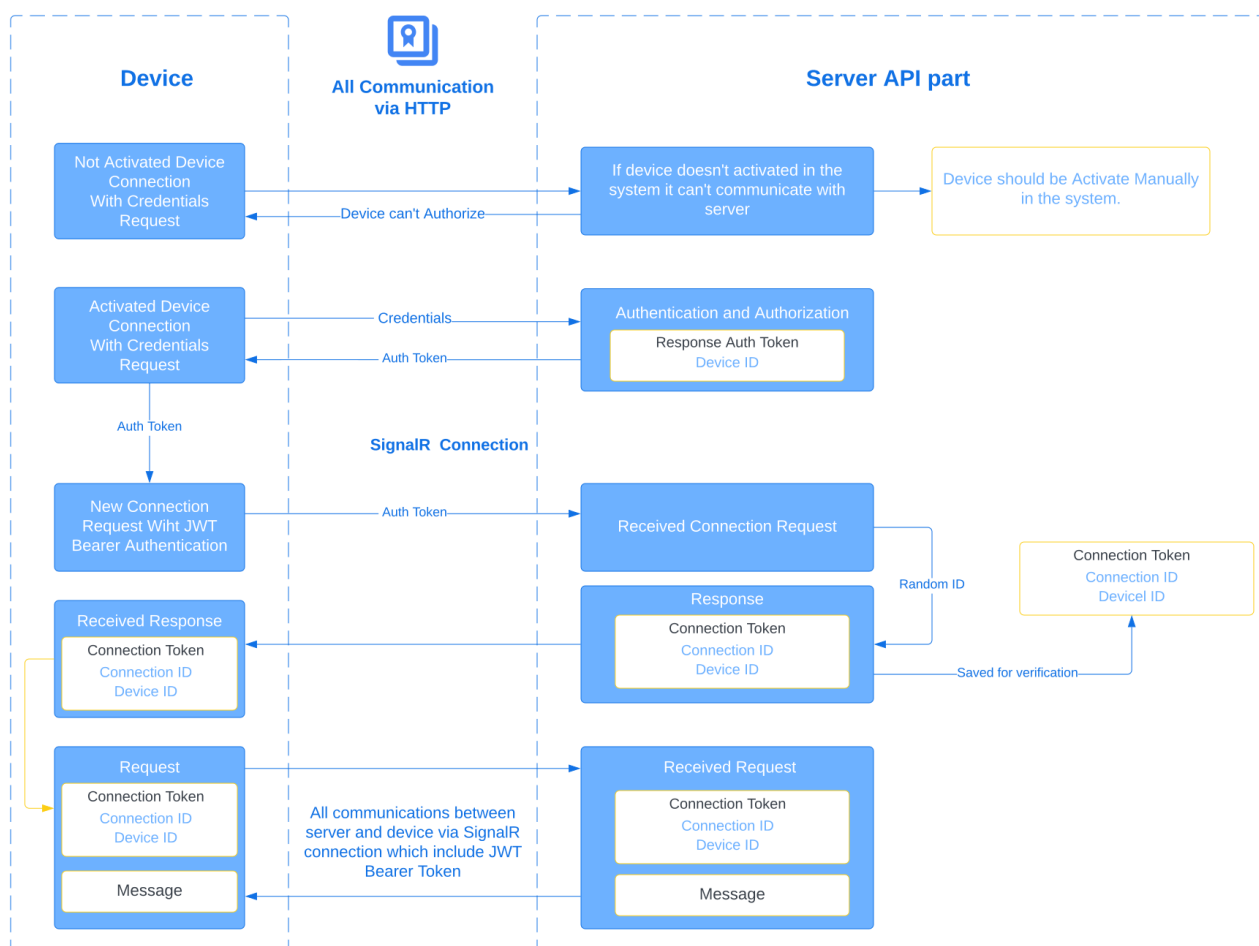


Рисунок 3.4 – Схема комунікації між приладом та сервером

3.2.2 Комунікація між приладом та датчиками

Протокол Serial COM Port слугує поширеним методом для полегшення зв'язку між пристроями та підключеними датчиками в системах клімат-контролю. Використовуючи послідовну передачу даних, він забезпечує ефективний обмін інформацією між близько розташованими пристроями.

Пристрої, оснащені Serial COM Port, можуть встановлювати з'єднання з підключеними датчиками, включаючи датчики температури, вологості, руху та інші типи датчиків. Використовуючи протокол Serial COM Port, ці пристрої можуть ініціювати запити до датчиків і отримувати вимірювання в режимі реального часу. Наприклад, пристрій може запитати поточне значення температури у датчика температури або відправити команду ввімкнутись охолоджуючому елементу.

Дані, отримані від датчиків, можуть бути оброблені та проаналізовані пристроєм. Завдяки використанню Serial COM Port, пристрій може передавати ці дані на сервер для подальшої обробки. Наприклад, він може відправляти дані про коливання температури і вологості на сервер для зберігання і аналізу. Це дозволяє системі контролю клімату відстежувати зміни умов і приймати відповідні рішення щодо регулювання клімату.

На відміну від таких протоколів, як HTTP і SignalR, Serial COM Port покладається на фізичні з'єднання через послідовні порти, такі як RS-232 або USB. Цей протокол включає в себе специфічні вимоги, що стосуються швидкості передачі даних, конфігурації бітової структури та контролю парності. Враховуючи ці унікальні характеристики, система клімат-контролю може ефективно обмінюватися даними з датчиками, забезпечуючи отримання актуальної інформації.

3.2.3 Комунікація між сервером та мобільним додатком

У системі клімат-контролю сервер і мобільний додаток використовують комбінацію HTTP-запитів і SignalR. Це забезпечує багато переваг та функціональних можливостей.

HTTP (Hypertext Transfer Protocol - протокол передачі гіпертексту) служить широко прийнятим протоколом для передачі даних через Інтернет. У контексті зв'язку між сервером і мобільним додатком HTTP-запити полегшують обмін даними між цими двома компонентами. HTTP-запити використовуються для ініціювання взаємодії між сервером і мобільним додатком. Мобільний додаток відправляє HTTP-запити до сервера для реєстрації та входу в систему, отримання даних, виконання певних дій або здійснення запитів на зміну налаштувань системи чи даних користувача. За допомогою HTTP методів, таких як GET, POST, PUT або DELETE, мобільний додаток може комунікувати з сервером і передавати різні типи даних, такі як текстові повідомлення, фотографії або файлові завдання. Сервер

обробляє ці запити і відповідає необхідними даними, забезпечуючи надійну і стандартизовану передачу даних між сервером і мобільним додатком.

SignalR використовується для забезпечення двосторонньої комунікації в режимі реального часу між сервером і мобільним додатком. Цей тип комунікації дозволяє серверу доставляти миттєві сповіщення та оновлення даних до мобільного додатку, усуваючи необхідність постійного опитування. Ця функціональність виявляється особливо цінною при оновленні даних з датчиків клімат-контролю, таких як температура, вологість, якість повітря та інших параметрів, які змінюються в режимі реального часу. SignalR використовує технологію WebSocket для підтримки активного з'єднання між сервером і мобільним додатком, забезпечуючи швидку і ефективну передачу даних в режимі реального часу.

Комбіноване використання HTTP-запитів і SignalR для зв'язку між сервером і мобільним додатком відкриває широкий спектр функціональних можливостей системи клімат-контролю. HTTP-запити полегшують отримання статичних даних, виконання дій на стороні сервера та передачу параметрів. У той же час, SignalR полегшує оновлення даних в режимі реального часу для забезпечення доступності актуальної інформації. Таке поєднання протоколів забезпечує зручний та ефективний зв'язок між сервером і мобільним додатком в системі клімат-контролю, швидку передачу даних, ефективну реалізацію режиму реального часу і зручний обмін даними між сервером і мобільним додатком.

РОЗДІЛ 4 ПРОЕКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ

4.1 Серверна частина

Серверна частина виконує важливі функції, такі як реєстрація та аутентифікація користувачів, керування пристроями та збереження даних з датчиків. Окрім того, серверна частина виступає посередником у комунікації між мобільним додатком і пристроєм. Це дозволяє забезпечити ефективну і безпечну взаємодію між всіма компонентами системи клімат-контролю.

У цій частині системи контролю клімату можна виділити три основних рівні: шар API (Application Programming Interface), бізнес-логіку (BL - Business Logic) та шар доступу до даних (DAL - Data Access Layer). Схему рівнів серверної частини зображено на рисунку 4.1.

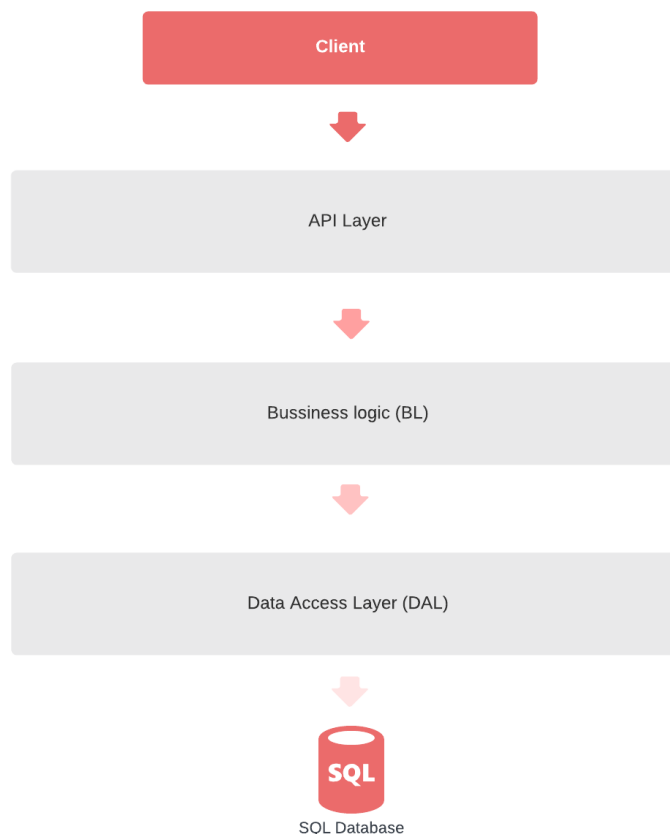


Рисунок 4.1 – Рівні серверної частини системи

Рівень API слугує основним інтерфейсом для підключення зовнішніх систем і клієнтів, включаючи мобільні додатки, до внутрішньої частини системи. Він пропонує набір кінцевих точок, які полегшують взаємодію між клієнтами та системою. Клієнти можуть взаємодіяти з системою за допомогою HTTP-запитів, передаючи дані, як правило, в JSON або інших форматах. Рівень API обробляє ці запити, викликає відповідні методи бізнес-логіки (BL) і повертає відповіді клієнтам.

До обов'язків рівня API входить маршрутизація вхідних запитів, перевірка даних, перетворення об'єктів у формати, зрозумілі клієнтам, і виконання інших функцій, пов'язаних з комунікацією між клієнтом і сервером. Його роль полягає в подоланні розриву між клієнтами та сервером, забезпечуючи безперебійну та ефективну комунікацію між двома сторонами.

Основними контролерами API в моїй системі контролю клімату є:

- AuthController – цей контролер відповідає за авторизацію приладу в системі, реєстрацію нових користувачів та авторизацію існуючих користувачів. Він містить декілька ендпоінтів, таких як DeviceIdentity (рис. 4.2), Login, Register і т.д., які дозволяють користувачам взаємодіяти з системою аутентифікації;
- DeviceController – цей контролер відповідає за отримання даних про стан клімату, керування приладами та налаштування опцій контролю клімату. Він містить ендпоінти, такі як TurnOnLamp, GetDeviceData, SaveClimateControlOptions (рис. 4.3) і т.д., які дозволяють користувачам отримувати актуальні дані про стан клімату, налаштовувати параметри та керувати пристроями;
- UserController – цей контролер відповідає за отримання та налаштування інформації про користувача. Він містить ендпоінти, такі як GetUserProfile, UpdateUserProfile, SavePhoto і т.д., які дозволяють користувачам отримувати свій профіль, оновлювати інформацію та змінювати фото.

```

[HttpPost("DeviceIdentity")]
[ProducesResponseType(200, Type = typeof(DeviceTokenModel))]
[AllowAnonymous]
0 references
public async Task<IActionResult> DeviceIdentity([FromBody] DeviceInfoModel model)
{
    try
    {
        if (model == null || model.Id == Guid.Empty)
            return Unauthorized();
        var device = await _deviceService.GetByIdAndKey(model.Id, model.Key);

        if (device == null && model.Id != Guid.Empty && !string.IsNullOrEmpty(model.Key) && !string.IsNullOrEmpty(model.Name))
        {
            var newDevice = new DeviceBase
            {
                Id = model.Id,
                Name = model.Name,
                IsActive = true
            };

            if(!await _deviceService.Create(newDevice, model.Key))
                return Unauthorized();

            device = await _deviceService.GetByIdAndKey(model.Id, model.Key);
        }

        if (device != null && !device.IsActive)
            return Unauthorized();

        if (device != null && device.IsActive)
        {
            var token = _jwtTokenManager.GetDeviceToken(model);
            return Ok(token);
        }

        return Unauthorized();
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, ex.Message);
        return StatusCode(500);
    }
}

```

Рисунок 4.2 – Метод API для авторизації приладу в системі

```

[HttpPost("{deviceId}/SaveClimateControlOptions")]
0 references
public async Task<IActionResult> SaveClimateControlOptions(Guid deviceId, [FromBody] ClimateControlOption options)
{
    try
    {
        if (deviceId == Guid.Empty)
            return BadRequest();

        await _devicesService.SaveClimateControlOptions(deviceId, options);

        return NoContent();
    }
    catch (Exception ex)
    {
        await CreateLog(ex);
        return StatusCode(500);
    }
}

```

Рисунок 4.3 – Метод API для збереження опцій клімат контролю

Рівень бізнес-логіки відповідає за реалізацію правил і процесів, які регулюють обробку даних у системі. Він охоплює такі функції, як перевірка даних, виконання бізнес-правил, координація операцій і взаємодія з рівнем доступу до

даних (DAL). Цей рівень в першу чергу займається обробкою складної логіки, пов'язаної з різними аспектами системи. Також він виконує найважливіші завдання, включаючи, але не обмежуючись ними, авторизацію, аутентифікацію користувачів, асоціювання пристроїв з користувачами (рис. 4.4), перевірку прав доступу та іншу обробку, що передбачає складну логіку.

```
public async Task<bool> SaveUserToDevice(Guid deviceId, Guid toUserId)
{
    try
    {
        if(deviceId == Guid.Empty)
            throw new ValidationException(message: "DeviceId is required");

        var currentOwner = await _devicesUsersDataController.GetOwnerId(deviceId);

        var isOwner = false;

        if(currentOwner == Guid.Empty)
            isOwner = true;

        if(currentOwner.Equals(toUserId))
            throw new ValidationException("This user is already the owner of this device..");

        var res = await _devicesUsersDataController.AddUserToDevice(deviceId, toUserId, isOwner);

        return res;
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, ex.Message);
        throw;
    }
}
```

Рисунок 4.4 – Метод для збереження зв'язки користувача та приладу

Рівень доступу до даних (DAL) забезпечує зв'язок між системою та базою даних. DAL складається з різних класів і методів, які забезпечують взаємодію з базою даних. Ці операції включають виконання SQL-запитів, встановлення з'єднань з базою даних, зіставлення об'єктів з відповідними таблицями бази даних та інші відповідні функціональні можливості. Завдяки інкапсуляції операцій, пов'язаних з базою даних, у DAL, система може взаємодіяти з базою даних без необхідності знати складні механізми роботи баз даних, що лежать в основі технології.

В якості СУБД використано MS SQL SERVER. Структура бази даних в серверній частині системи клімат-контролю складається з шести таблиць, які забезпечують організацію та збереження даних. Усі таблиці є взаємопов'язаними (рис. 4.5). Основні таблиці включають:

- Users (Користувачі). Ця таблиця містить інформацію про користувачів системи. Вона містить поля, такі як ідентифікатор користувача, ім'я, прізвище, електронну пошту, пароль та інші відомості необхідні для коректної аутентифікації. Ця таблиця використовується для реєстрації та аутентифікації користувачів;
- Devices (Прилади). Таблиця "Devices" містить інформацію про керовані пристрої системи контролю клімату. Кожен запис в цій таблиці представляє окремий пристрій і містить поля, такі як ідентифікатор пристрою, назва, ключ, статус пристрою, статус підключення та інші характеристики;
- SensorsData (Дані з датчиків). Ця таблиця використовується для збереження даних, отриманих з датчиків клімат-контролю. Вона містить поля, такі як ідентифікатор даних, ідентифікатор пристрою, значення датчиків, дата та час отримання даних. Ця таблиця дозволяє зберігати історичні дані з датчиків для подальшого аналізу і візуалізації.

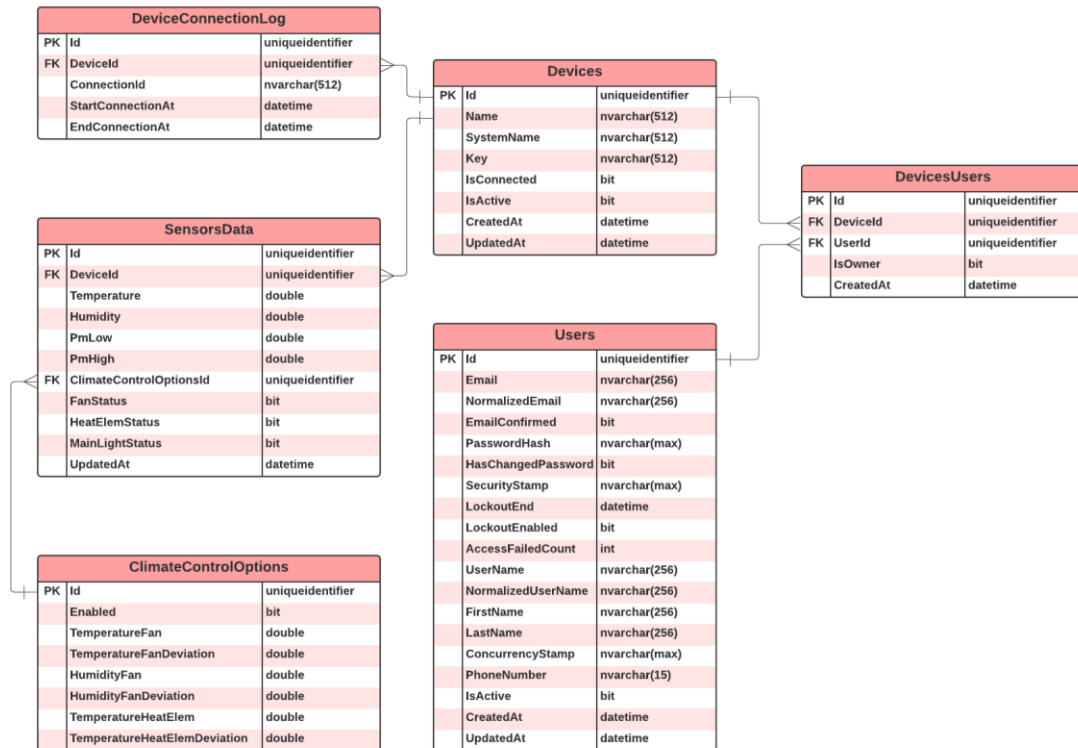


Рисунок 4.5 – Діаграма бази даних

Всі три шари (DAL, BL і API) взаємодіють між собою для забезпечення повної функціональності системи контролю клімату. DAL виконує операції з базою даних, BL здійснює обробку даних та бізнес-логіку, а API забезпечує комунікацію з клієнтами і керує потоком даних в системі. Комбінація цих шарів дозволяє забезпечити ефективну та надійну роботу серверної частини системи клімат-контролю.

4.2 Прилад для контролю клімату

Прилад підключений до одноплатного чи звичайного комп'ютера на операційній системі Linux або Windows по інтерфейсу спілкування UART (Serial). Контролер клімат контролю побудований на мікросхемі STM32F та включає в себе 4 твердотілих реле та 4 механічних реле для контролю нагрівального елемента (в моєму випадку тен з вентилятором) та охолоджувального елемента (потужний вентилятор) а також датчик температури та вологості DHT-11.

Спілкування з контролером клімату відбувається таким чином:

- Контролер раз в секунду запитує дані від датчиків вологості, температури і якості повітря і надсилає цю інформацію через Serial COM Port.
- Засобами мови C# на комп'ютері відкривається Serial COM Port і встановлюється зв'язок з вихідними параметрами, такими як швидкість передачі 115200 (baud rate), 8 бітів даних, паритет None, StopBits.One. Після цього програма очікує відповіді від контролера, який надсилає відповідну структуру даних, що містить інформацію про стан реле (увімкнено чи вимкнено), а також дані про температуру, вологість і зміни параметрів якості повітря.
- Програма обробляє отримані дані і передає їх до додаткових сервісів, які відповідають за керування нагрівальним і охолоджувальним пристроями, а також обробку значень температури і вологості.

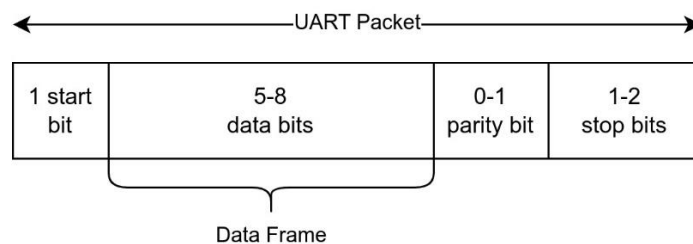


Рисунок 4.6 – Структура UART пакета

Сервіс обробки температури працює таким чином:

- Вентилятор вмикається, коли температура перевищує встановлену температуру плюс відхилення, і вимикається, коли температура стає нижчою за задану.
- Вентилятор вмикається, коли вологість у коробці перевищує встановлену вологість плюс відхилення, і вимикається, коли вологість стає нижчою за встановлену.
- Нагрівальний елемент вмикається, коли температура стає нижчою за задану, і вимикається, коли температура перевищує встановлену температуру плюс відхилення.

4.3 Мобільний додаток

Мобільний додаток системи контролю клімату, розроблений з використанням MAUI Blazor, надає багатофункціональний спосіб керування та моніторингу кліматичних параметрів у приміщенні. Цей додаток дозволяє користувачам здійснювати контроль температури, вологості та якості повітря у реальному часі через їх мобільні пристрої.

Для кращого розуміння принципу роботи мобільного застосунку, було побудовано блок-схему (рис. 4.7).

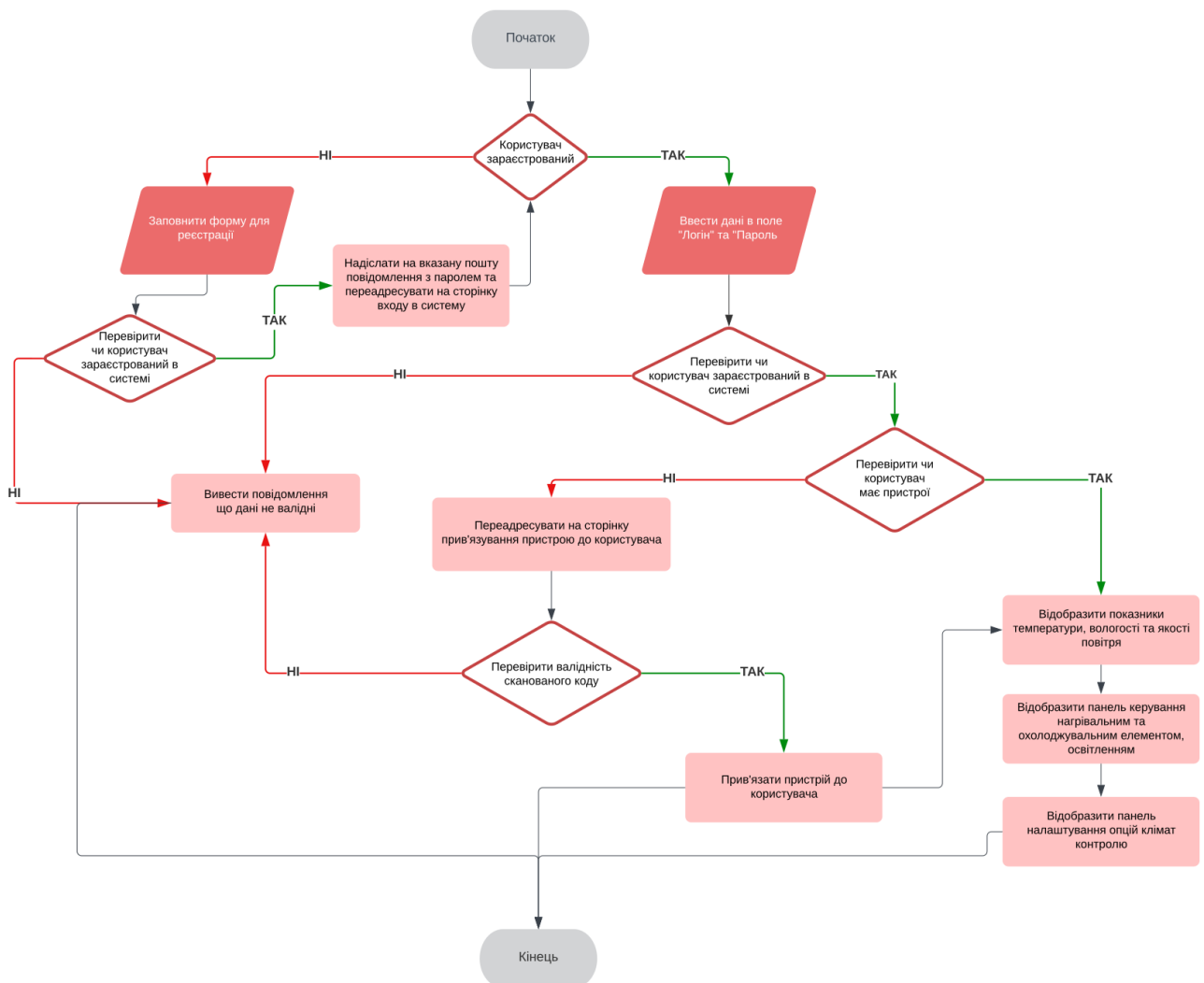


Рисунок 4.7 – Діаграма бази даних

Після встановлення та запуску мобільного застосунку, користувач потрапляє на головну сторінку, де йому надається можливість зареєструватися або авторизуватися в системі. На цій сторінці представлені дві основні опції: «Log in» та «Create account» (рис. 4.8).



Рисунок 4.8 – Вигляд початкової сторінки застосунку

Якщо користувач вперше використовує додаток, він може обрати опцію "Create account". При цьому йому буде запропоновано заповнити форму зі своїми особистими даними, такими як ім'я, прізвище, електронна пошта та номер телефону даними (рис. 4.9). Користувач повинен заповнити всі обов'язкові поля, а потім натиснути кнопку "Create account" для створення нового облікового запису. Після цього, якщо всі дані валідні, на вказану ним електронну пошту, надсилається вітальне повідомлення з паролем до новоствореного облікового запису.

16:37 20%

← Create account ×

Name

LastName

E-mail

Phone

67 555 1234

CREATE ACCOUNT

Рисунок 4.9 – Вигляд сторінки авторизації

Якщо користувач вже має обліковий запис, він може обрати опцію "Log in". При цьому йому буде запропоновано ввести свої облікові дані, такі як електронна пошта та пароль (рис. 4.10). Після введення логіну та паролю, відсилається запит на сервер для перевірки наявності користувача у списку зареєстрованих. Якщо ж користувача не знайдено в БД, виводиться відповідне повідомлення про помилку.

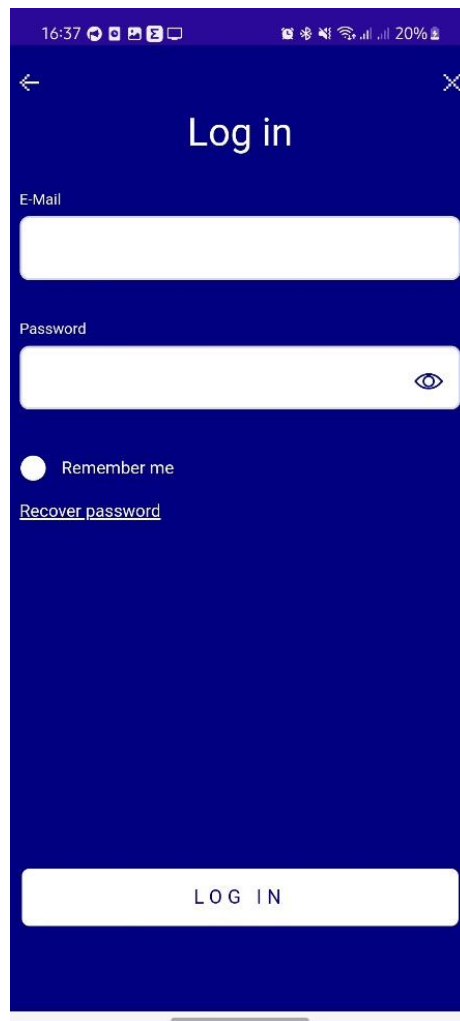


Рисунок 4.10 – Вигляд сторінки реєстрації

Після успішної авторизації користувач отримує доступ до головного інтерфейсу додатка, де він може підключити прилад для контролю клімату до свого облікового запису. Цей процес може бути здійснений за допомогою відсканування QR-коду, який знаходиться на самому пристрої, або ручного введення ідентифікатора приладу.

У випадку, якщо користувач не має можливості сканувати QR-код, він має можливість ввести ідентифікатор приладу вручну (рис. 4.11). Цей ідентифікатор може бути наданий разом з приладом або знайдений у його документації. Користувач вводить цей ідентифікатор у відповідне поле в додатку для підключення приладу до свого облікового запису.



Рисунок 4.11 – Вигляд сторінки прив’язування приладу вручну

Для відсканування QR-коду, користувач використовує камеру свого мобільного пристрою, спрямовуючи її на код на приладі (рис. 4.12). Після успішного сканування, інформація про прилад автоматично використовується для підключення до облікового запису користувача.



Рисунок 4.12 – Вигляд сторінки прив’язування приладу вручну

Після успішного підключення приладу до облікового запису, користувачі мають повний доступ до управління та контролю параметрів клімату, які забезпечуються зазначеним приладом.

Мобільний застосунок пропонує інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам переглядати поточні показники температури, вологості та якості повітря (рис. 4.13). Крім того, вони можуть встановлювати бажані значення для цих параметрів і налаштовувати режими роботи пристроїв контролю, таких як нагрівальний елемент, охолоджувальний елемент та освітлення.

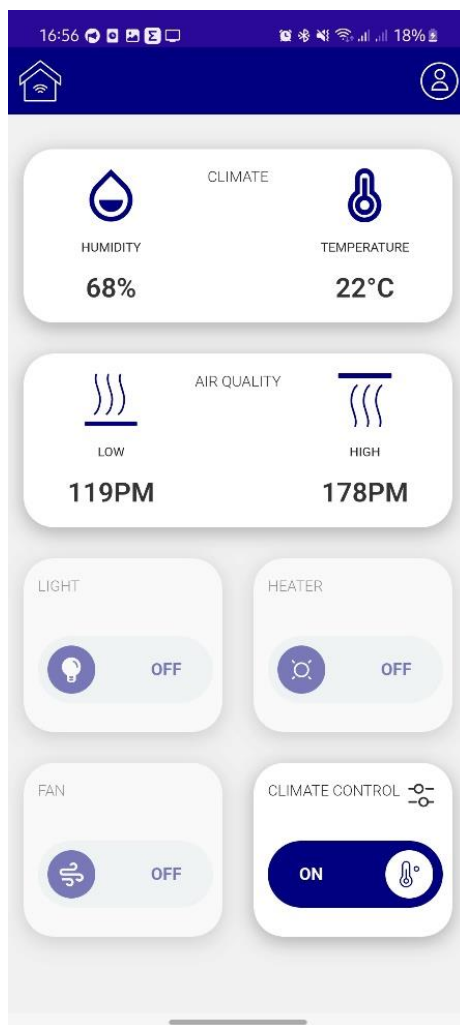


Рисунок 4.13 – Вигляд сторінки з показниками сенсорів

Також користувач може увімкнути та задати параметри для автоматичного контролю клімату (рис. 4.14). Користувач може встановити бажані значення температури, вологості, режимів роботи та інших параметрів в головному інтерфейсі додатка. Наприклад, він може встановити бажану температуру, при перевищенні якої система автоматично увімкне охолоджувальний або нагрівальний елемент. Так само, користувач може встановити бажану вологість, при перевищенні якої система автоматично увімкне вентилятор.

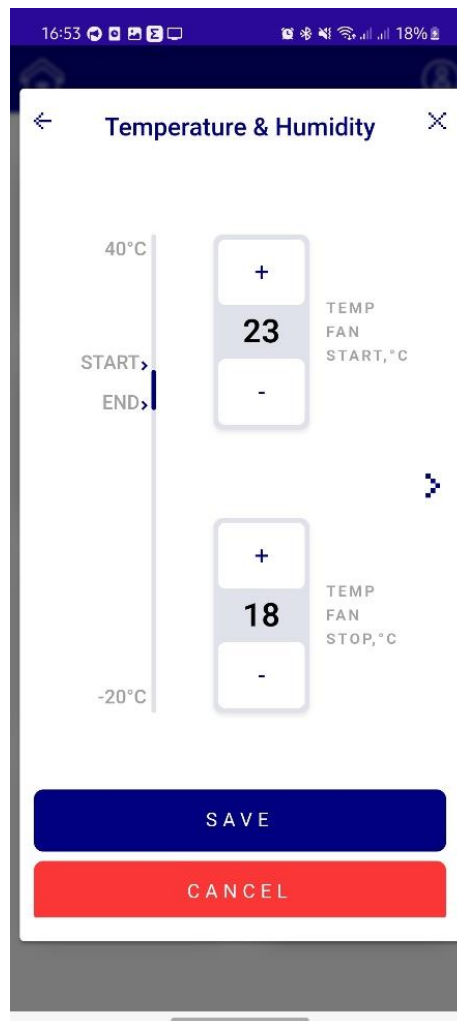


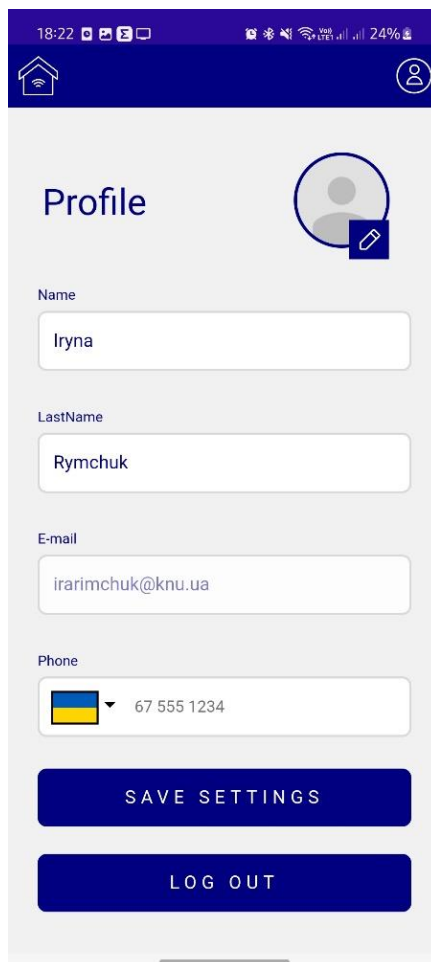
Рисунок 4.14 – Вигляд сторінки налаштувань клімат-контролю

Цей автоматичний контроль дозволяє користувачеві налаштувати систему контролю клімату таким чином, щоб вона самостійно реагувала на зміни умов і забезпечувала оптимальний рівень комфорту та енергоефективності.

В даному застосунку користувачу надається можливість редагувати свій профіль, щоб змінити та оновити свої контактні дані. Це дозволяє користувачам забезпечити актуальну інформацію та персоналізувати свій профіль відповідно до своїх потреб. Це він може зробити перейшовши до розділу «Профіль».

У формі редагування профілю (рис. 4.15) користувач зможе ввести нові дані або оновити існуючі. Наприклад, він може змінити своє прізвище, ім'я та номер телефону, введенням нових значень у відповідні поля.

Користувач також може встановити або оновити фото профілю, надавши зображення зі свого пристрою або використовуючи функцію фотокамери пристрою для зйомки нового зображення.



The screenshot shows a mobile application interface for profile settings. At the top, there is a dark blue header with a home icon on the left and a user profile icon on the right. The main content area is light gray and contains the following elements:

- Profile** title and a circular profile picture placeholder with a blue edit icon.
- Name** field with the text "Iryna".
- LastName** field with the text "Rymchuk".
- E-mail** field with the text "irarimchuk@knu.ua".
- Phone** field with a dropdown menu showing the Ukrainian flag and the text "67 555 1234".
- SAVE SETTINGS** button.
- LOG OUT** button.

Рисунок 4.15 – Вигляд сторінки налаштувань профілю

ВИСНОВКИ

У результаті проведеного дослідження та розробки була створена високоефективна та функціональна система клімат-контролю, що забезпечує точний та надійний контроль параметрів клімату в приміщенні. Структура системи включає в себе мобільний додаток MAUI Blazor, серверну частину та сам прилад для контролю клімату.

Система забезпечує автоматичний контроль параметрів клімату в приміщенні. Завдяки сполученню з датчиками та виконавчими пристроями через послідовний порт, система здатна отримувати регулярні оновлення щодо температури, вологості та якості повітря, а також керувати реле для нагрівання та охолодження приміщення. Мобільний додаток надає зручний та інтуїтивно зрозумілий інтерфейс для користувачів. Вони можуть з легкістю керувати та налаштовувати систему клімат-контролю зі своїх мобільних пристроїв, таких як смартфони або планшети.

Застосування такої системи клімат-контролю може бути корисним у різних сферах, включаючи домашнє використання, офісні приміщення, промислові приміщення та інші. Вона дозволяє забезпечити комфортні умови в приміщенні, економію енергії.

Отримані результати даної роботи можуть бути використані як основа для подальшого розвитку системи та впровадження її на реальних об'єктах з метою створення комфортних умов проживання та роботи. Дана система має потенціал для подальшого вдосконалення та розширення функціональності з урахуванням специфічних потреб користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Is Tony Fadell the next Steve Jobs or the next Larry Page? [Електронний ресурс] // Fortune. – Режим доступу: <https://web.archive.org/web/20201112022959/http://fortune.com/2014/06/12/tony-fadell-nest/> (дата звернення: 10.04.2023). – Назва з екрана.
2. Smart Thermostats & Smart Home Devices | ecobee [Електронний ресурс] // Smart Thermostats & Smart Home Devices | ecobee. – Режим доступу: <http://www.ecobee.com/en-us/> (дата звернення: 01.04.2023). – Назва з екрана.
3. Tado App [Електронний ресурс] // Tado Smart Thermostat | The end of high heating costs. – Режим доступу: <https://www.tado.com/all-en/tado-app> (дата звернення: 12.04.2023). – Назва з екрана.
4. What is Visual Studio? [Електронний ресурс] // Microsoft Learn: Build skills that open doors in your career. – Режим доступу: <https://learn.microsoft.com/uk-ua/visualstudio/get-started/visual-studio-ide?view=vs-2022> (дата звернення: 30.04.2023). – Назва з екрана.
5. Introducing .NET Core - .NET Blog [Електронний ресурс] // .NET Blog. – Режим доступу: <https://devblogs.microsoft.com/dotnet/introducing-net-core/> (дата звернення: 28.04.2023). – Назва з екрана.
6. Stack Overflow Developer Survey 2022 [Електронний ресурс] // Stack Overflow. – Режим доступу: <https://survey.stackoverflow.co/2022> (дата звернення: 29.04.2023). – Назва з екрана.
7. Blazor Hybrid Web Apps with .NET MAUI [Електронний ресурс] // CODE Online. – Режим доступу: <https://www.codemag.com/Article/2111092/Blazor-Hybrid-Web-Apps-with-.NET-MAUI> (дата звернення: 05.05.2023). – Назва з екрана.
8. Henderson K. The Guru's Guide to SQL Server Architecture and Internals / Ken Henderson. – [Б. м.] : Addison-Wesley Professional, 2003. – 1072 с..

9. HTTP - HTTP | MDN [Електронний ресурс] // MDN Web Docs. – Режим доступу: <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview> (дата звернення: 09.05.2023). – Назва з екрана.
10. Introduction to SignalR [Електронний ресурс] // Microsoft Learn: Build skills that open doors in your career. – Режим доступу: <https://learn.microsoft.com/uk-ua/aspnet/signalr/overview/getting-started/introduction-to-signalr> (дата звернення: 01.05.2023). – Назва з екрана.
11. Послідовний порт – Вікіпедія [Електронний ресурс] // Вікіпедія. – Режим доступу:
https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%BE%D1%80%D1%82 (дата звернення: 09.04.2023). – Назва з екрана.

ДОДАТОК А

Фрагмент коду сервісу приладу підписки на отримання даних з датчиків:

```

public async Task Subscribe()
{
    try
    {
        _externalDevicesCommunicationService.OnRelayDataReceive -=
RelayDataReceive;

        var config = await
_advancedSettingsService.GetByType(TypeAdvancedSetting.Fan);
        if (config == null || !config.Enabled || string.IsNullOrEmpty(config.Options))
        {
            Logger.Here().Information("Not subscribe because config == null ||
!config.Enabled");
            return;
        }

        var options = ParseOptions(config.Options);
        if (options?.Relay == null)
        {
            Logger.Here().Information("Not subscribe because options == null ||
string.GpioPin == null");
            return;
        }
        _externalDevicesCommunicationService.OnRelayDataReceive +=
RelayDataReceive;
        _relay = options.Relay;
    }
}

```

```
_urls = options?.CallbackUrl;  
  
}  
catch (Exception e)  
{  
    Logger.Here().Error(e, e.Message);  
}  
  
}
```

ДОДАТОК Б

Фрагмент коду контролю температури:

```
private void ControlTemperature(TempHumDataModel data)
{
    try
    {
        Logger.Here().Debug("Start method");
        //for heating elem
        if (_climateControlSettings.TemperatureHeatElem != null &&
            _climateControlSettings.TemperatureHeatElemDeviation != null)
        {
            if (_climateControlSettings.TemperatureHeatElem -
                _climateControlSettings.TemperatureHeatElemDeviation >
data.Temperature &&
                _heatingElementState == false)
            {
                _heatingElementState = true;
                _ = SwitchOnHeatingElement();
                Logger.Here().Information($"Switch on heating element, state:
{_heatingElementState}");
            }

            if (_climateControlSettings.TemperatureHeatElem < data.Temperature &&
                _heatingElementState == true)
            {
                _heatingElementState = false;
                _ = SwitchOffHeatingElement();
                Logger.Here().Information($"Switch on heating element, state:
{_heatingElementState}");
            }
        }
    }
}
```

```

    }
}

// for fan
if (_climateControlSettings.TemperatureFan != null &&
    _climateControlSettings.TemperatureFanDeviation != null &&
    _climateControlSettings.HumidityFan != null &&
    _climateControlSettings.HumidityFanDeviation != null)
{
    if (_climateControlSettings.TemperatureFan +
        _climateControlSettings.TemperatureFanDeviation <
        data.Temperature && _fanState == false)
    {
        _fanState = true;
        _ = SwitchOnFan();
        Logger.Here().Information($"Switch on fan by temperature, funState:
{_fanState}");
    }

    if (_climateControlSettings.HumidityFan +
        _climateControlSettings.HumidityFanDeviation <
        data.Humidity && _fanState == false)
    {
        _fanState = true;
        _ = SwitchOnFan();
        Logger.Here().Information($"Switch On Fan by humidity, funState:
{_fanState}");
    }
}

```

```
        if (_climateControlSettings.HumidityFan > data.Humidity &&
            _climateControlSettings.TemperatureFan > data.Temperature &&
            _fanState == true)
        {
            _fanState = false;
            _ = SwitchOffFan();
            Logger.Here().Information($"Switch Off Fan by humidity and temperature,
funState: {_fanState}");
        }
    }
}
catch (Exception e)
{
    Logger.Here().Error(e, e.Message);
}
}
```