ДОДАТОК В

Специфікація програмного забезпечення

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук

Кафедра програмної інженерії

СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Програмна система для контролю споживання води у приміщенні

Студентка гр. ПЗПІ-22-7 Кравченко В.А.

Харків 2025

**ЗМІСТ**

[Вступ](#_Toc105146293) 63

[1.1 Огляд продукту](#_Toc105146295) 63

[1.2 Мета](#_Toc105146296) 64

1.3 Межі 64

1.4 Посилання 65

1.5 Означення та абревіатури 66

[2 Загальний опис](#_Toc105146301) 68

[2.1 Перспективи продукту](#_Toc105146302) 68

[2.2 Функції продукту](#_Toc105146303) 69

[2.3 Характеристики користувачів](#_Toc105146304) 70

[2.4 Загальні обмеження](#_Toc105146303) 71

[2.5 Припущення й залежності](#_Toc105146304) 72

[3 Конкретні вимоги](#_Toc105146307) 73

[3.1 Вимоги до зовнішніх інтерфейсів](#_Toc105146308) 73

[3.2 Властивості програмного продукту](#_Toc105146309) 74

[3.3 Атрибути програмного продукту](#_Toc105146310) 75

[3.4 Вимоги бази даних](#_Toc105146309) 76

[3.5 Інші вимоги](#_Toc105146310) 77

**ВСТУП**

1.1 Огляд продукту

Програмна система для контролю споживання води у приміщенні «AquaTrack» є інформаційною системою, призначеною для моніторингу, аналізу та управління витратами водних ресурсів. Система розроблена з метою забезпечення прозорого обліку споживання води, встановлення лімітів використання, оперативного виявлення перевищень і формування звітів для подальшого аналізу споживання.

Програмний продукт охоплює як клієнтську частину, реалізовану у вигляді десктопного застосунку на базі Windows Forms, так і вебінтерфейс, побудований з використанням React. Завдяки цьому система забезпечує доступ до функціоналу як з локального комп’ютера, так і через браузер.

Основним призначенням системи є обслуговування офісів, будівель громадського користування, навчальних закладів та інших організацій, де необхідно вести облік водоспоживання без прив’язки до конкретного користувача. «AquaTrack» надає адміністраторам та користувачам змогу переглядати історію використання води за вибраний період, аналізувати активність пристроїв, встановлювати обмеження на витрати води, а також оперативно реагувати на перевищення норм.

Система підтримує багаторівневий доступ, що дозволяє розмежовувати функціональні можливості між адміністраторами та звичайними користувачами. Адміністративний інтерфейс включає управління акаунтами, перегляд усіх лімітів, масову перевірку споживання по всіх пристроях та резервне копіювання даних. Користувацький інтерфейс зосереджений на наданні зручного доступу до поточних та історичних даних водоспоживання, з урахуванням можливості візуального відображення інформації та локалізації інтерфейсу.

«AquaTrack» спроєктовано з урахуванням сучасних вимог до продуктивності, масштабованості та безпеки. Для обміну даними між клієнтськими та серверними компонентами використовується REST API. Збереження даних здійснюється у реляційній базі даних SQL Server з використанням ORM-технології Entity Framework. Уся бізнес-логіка реалізована мовою програмування C# з підтримкою .NET Framework. Завдяки використанню Microsoft Visual Studio як інтегрованого середовища розробки, забезпечено ефективне тестування та налагодження всіх компонентів системи.

1.2 Мета

Метою створення програмної системи для контролю споживання води у приміщенні є розробка ефективного, надійного та масштабованого програмного забезпечення для автоматизованого контролю споживання води у приміщеннях, яке забезпечує збирання, обробку та зберігання інформації про обсяг використаної води. У результаті реалізації проєкту очікується покращення якості управління ресурсами, скорочення витрат води, виявлення аномалій у споживанні, а також підвищення відповідальності за раціональне використання водних ресурсів.

Розробка системи передбачає створення зручного інтерфейсу користувача, який забезпечуватиме легкий доступ до даних про витрати води за різні часові періоди та по різних пристроях, як у вигляді табличних, так і графічних уявлень. Програмний продукт також має включати адміністративний модуль з розширеними функціями управління системою, користувачами, пристроями та обмеженнями.

Програмне забезпечення повинно підтримувати роботу у багатокористувацькому середовищі, передбачати обмеження доступу до функціоналу залежно від ролі користувача, забезпечувати локалізацію інтерфейсу, а також підтримку резервного копіювання та відновлення даних.

1.3 Межі

Програмна система призначена для моніторингу, збору та аналізу даних про споживання води у межах одного об’єкта, наприклад, офісного приміщення, житлового будинку або установи. Основною задачею є надання користувачам актуальної інформації про обсяг використаної води та виявлення перевищення встановлених лімітів.

Система не призначена для прямого управління роботою водопостачального обладнання або регулювання подачі води. Вона не виконує роль промислової автоматики, а виключно надає інформацію про вже здійснене споживання. Таким чином, втручання у фізичні процеси водопостачання залишається поза межами реалізації даного програмного забезпечення.

Інформаційна модель системи не передбачає інтеграцію з платіжними сервісами чи автоматичне нарахування вартості спожитої води. Уся обробка зосереджена виключно на обсягах, часі та джерелах споживання, без фінансових розрахунків.

Система не підтримує мобільні платформи у вигляді окремого застосунку, однак вебінтерфейс дозволяє частково компенсувати це обмеження через доступ з браузера. Також система не передбачає голосового управління чи елементів штучного інтелекту, які аналізують поведінку користувача або автоматично формують рекомендації.

1.4 Посилання

У процесі розробки, проектування, реалізації та документування програмної системи для контролю споживання води у приміщенні «AquaTrack» було використано низку нормативних документів, технічних специфікацій, інструкцій, фреймворків, бібліотек та офіційної документації від виробників технологій.

У частині використання бібліотек і технологій для роботи з базами даних застосовувалась документація Microsoft до Entity Framework Core, яка визначає підходи до ORM-реалізації, способи конфігурації моделей, зв’язків між таблицями та виконання запитів LINQ. Також були використані офіційні матеріали Microsoft щодо створення додатків з використанням Windows Forms на базі .NET Framework.

При реалізації вебчастини та REST API орієнтиром слугувала документація до React.js, а також офіційні гіди Microsoft із побудови серверної частини з використанням ASP.NET Core. Для організації обміну даними між клієнтською та серверною частиною були враховані специфікації RESTful-архітектури.

З метою забезпечення інтеграції з базою даних SQL Server використовувались рекомендації з Microsoft SQL Server Management Studio, що дозволили правильно реалізувати структуру бази даних, зв’язки між сутностями та логіку обробки транзакцій.

При розробці також були враховані принципи об’єктно-орієнтованого програмування, шаблони проектування, такі як MVC, а також загальні підходи до проєктування програмних систем з урахуванням майбутнього розширення, модульності та супроводжуваності коду.

1.5 Означення та абревіатури

REST API — інтерфейс програмування, що дозволяє клієнтам обмінюватися даними з сервером за допомогою стандартних HTTP-запитів (GET, POST тощо).

SQL Server — система керування базами даних від Microsoft, яка використовується для зберігання та обробки структурованої інформації.

ORM (Object-Relational Mapping) — технологія, що дозволяє працювати з базою даних як з об’єктами у коді, без написання SQL-запитів.

.NET Framework — платформа від Microsoft для розробки програм під Windows, яка містить бібліотеки, середовище виконання та інструменти.

LINQ (Language Integrated Query) — мова запитів, вбудована в C#, що дозволяє виконувати запити до колекцій, баз даних або XML у зручному синтаксисі.

React.js — бібліотека JavaScript для створення динамічних користувацьких інтерфейсів у вебзастосунках.

RESTful-архітектура — стиль побудови вебсервісів, що дотримується принципів REST і забезпечує просту та масштабовану взаємодію клієнт-сервер.

MVC (Model-View-Controller) — шаблон проєктування, який розділяє програму на три частини: модель (дані), уявлення (інтерфейс), контролер (логіка).

IoT (Internet of Things) — концепція, за якою пристрої (сенсори, лічильники тощо) підключені до інтернету й можуть взаємодіяти один з одним.

HTML (HyperText Markup Language) — мова розмітки для створення структури вебсторінок.

CSS (Cascading Style Sheets) — мова стилів, яка відповідає за зовнішній вигляд елементів вебсторінки (кольори, шрифти, розміщення).

JavaScript — мова програмування для додавання інтерактивності до вебсторінок (анімація, обробка подій, динамічні зміни).

HTTP (HyperText Transfer Protocol) — протокол, за допомогою якого браузер та сервер обмінюються даними у вебмережі.

MQTT (Message Queuing Telemetry Transport) — легкий мережевий протокол, що використовується в IoT для передачі даних між пристроями з низьким енергоспоживанням.

JSON (JavaScript Object Notation) — формат обміну даними у вигляді тексту, зручний для читання людиною та обробки машиною.

C# (C-Sharp) — мова програмування від Microsoft, яка використовується для розробки різноманітних додатків, особливо в середовищі .NET.

**2 ЗАГАЛЬНИЙ ОПИС**

2.1 Перспективи продукту

Програмна система контролю споживання води у приміщенні «AquaTrack» є важливим компонентом цифрової екосистеми, що спрямована на підвищення екологічної свідомості користувачів, оптимізацію витрат водних ресурсів та впровадження енергоефективних технологій у повсякденне життя. На етапі проектування система передбачала можливість масштабування та розширення функціональності, що дозволяє вважати її перспективною платформою для розвитку інтелектуальних систем управління ресурсами.

Програмне забезпечення вже на початковому етапі реалізує ключові функції для збору, обробки та візуалізації даних про споживання води, підтримує введення лімітів, контроль у реальному часі та інструменти адміністрування. Надалі можливе розширення функціоналу за рахунок інтеграції з іншими системами інтернету речей (IoT), наприклад, автоматичними клапанами перекриття водопостачання, сенсорами витоку, системами розумного будинку або платформами аналітики.

Завдяки побудові програмної архітектури на основі модульного підходу, система має високий потенціал для адаптації під потреби різних користувачів: від приватних осіб до комерційних об’єктів, офісів або муніципальних служб. Передбачена можливість інтеграції з мобільними застосунками, що надасть доступ до інформації про споживання води в будь-який час з будь-якого пристрою, підвищуючи мобільність та зручність у використанні.

Також перспективним напрямом розвитку є розширення можливостей аналітики, впровадження машинного навчання для прогнозування споживання, виявлення аномалій у поведінці споживачів або аварійних ситуацій. З урахуванням екологічних викликів та зростаючої вартості ресурсів, система «AquaTrack» може стати основою для створення комплексної платформи моніторингу не лише води, але й електроенергії, тепла, що значно підвищить її значення в інтелектуальних рішеннях управління приміщеннями.

2.2 Функції продукту

Програмна система реалізує широкий спектр функцій, які забезпечують ефективний моніторинг, аналіз та управління водними ресурсами в межах одного або декількох пристроїв, що встановлені у конкретному об’єкті. Функціональність програмного забезпечення орієнтована на задоволення потреб як користувача, так і адміністратора системи, забезпечуючи облік, контроль та взаємодію з даними про водоспоживання.

До основних функціональних можливостей системи належить збір та збереження даних про об’єм спожитої води за певні проміжки часу. Користувачі можуть здійснювати вибір діапазону дат для перегляду історичних даних, що дозволяє аналізувати динаміку витрат та виявляти відхилення від звичних обсягів. Також реалізовано функцію виведення споживання в режимі реального часу з графічним відображенням за допомогою візуальних компонентів інтерфейсу.

Окремим елементом функціональності є система встановлення індивідуальних лімітів водоспоживання. Користувач має змогу задати граничні значення на добу, тиждень або місяць, а система відстежує перевищення та сповіщає про них у зручному форматі. Це дозволяє запобігати надмірному споживанню, сприяє підвищенню екологічної відповідальності та дає змогу раціонально розподіляти ресурси.

Функціонал авторизації та реєстрації забезпечує захист персональних даних та індивідуальний доступ до особистого кабінету.

Інтерфейс адміністратора дозволяє здійснювати управління користувачами, переглядати загальні статистичні дані, контролювати пристрої, що підключені до системи, а також виконувати технічне обслуговування, включаючи резервне копіювання та відновлення бази даних. Таким чином, адміністратор має змогу забезпечувати стабільну роботу програмного комплексу та керувати його складовими частинами.

Програмне забезпечення підтримує локалізацію, що дає змогу використовувати систему різними мовами. Реалізація цієї функції передбачає підвищення доступності продукту для ширшого кола користувачів.

Уся функціональність системи тісно інтегрована із базою даних, де зберігається інформація про користувачів, пристрої, водоспоживання та встановлені ліміти. Завдяки використанню ORM-технологій (зокрема, Entity Framework), забезпечується зручна, безпечна та надійна робота з даними на серверному рівні.

2.3 Характеристики користувачів

Система орієнтована на декілька категорій користувачів, кожна з яких має власний набір потреб та рівень взаємодії з функціоналом системи. Для всіх типів користувачів спільними є вимоги до базової комп’ютерної грамотності та наявності доступу до пристроїв із встановленим додатком або можливістю використання вебінтерфейсу через браузер. Програмна система має інтуїтивно зрозумілий інтерфейс, що дозволяє взаємодіяти з нею навіть користувачам, які не мають спеціалізованої технічної підготовки.

Основну частину користувачів складають кінцеві користувачі — мешканці або персонал приміщення, де встановлені пристрої моніторингу споживання води. Вони можуть переглядати дані про споживання води в реальному часі, аналізувати інформацію за обраний період, встановлювати особисті ліміти споживання та отримувати сповіщення про їхнє перевищення. Ця категорія користувачів не потребує глибокого розуміння роботи з базами даних або технічної структури системи. Їм достатньо базових навичок користування графічним інтерфейсом Windows-додатків або веб браузера.

Окремою категорією є адміністратори системи, які відповідають за технічне обслуговування програмного забезпечення, ведення обліку користувачів, управління пристроями збору даних, аналіз усієї сукупності інформації про водоспоживання та забезпечення цілісності даних у базі. Адміністратори повинні мати достатній рівень підготовки у сфері інформаційних технологій, розуміти принципи функціонування баз даних, клієнт-серверної архітектури, систем резервного копіювання та відновлення.

2.4 Загальні обмеження

Обмеження визначають як технічні, так і організаційні межі, в яких функціонує програмне забезпечення, і впливають на його можливості, адаптивність та потенціал подальшого розвитку.

Одним із ключових обмежень є залежність системи від постійної стабільності інтернет-з’єднання у випадку використання вебінтерфейсу або передачі даних IoT-пристроями. У разі відсутності підключення система не може своєчасно оновлювати інформацію, що особливо критично для реального моніторингу споживання води. Це знижує оперативність прийняття рішень на основі даних та обмежує функціональність окремих модулів.

Іншим суттєвим обмеженням є апаратна залежність від встановлених сенсорів або пристроїв збору даних. У разі некоректної роботи датчиків або збою під час збору інформації програма може відображати неточні або застарілі дані, що впливає на достовірність звітів та порівняльного аналізу. Таким чином, загальна ефективність системи частково залежить від фізичної інфраструктури моніторингу.

Ще одним обмеженням є масштабованість. Поточна архітектура програмної системи розроблена для використання в рамках одного приміщення або обмеженої кількості об’єктів. При суттєвому збільшенні кількості користувачів або пристроїв можуть виникати навантаження на серверну частину, що потребуватиме оптимізації запитів до бази даних, розширення обчислювальних ресурсів або модернізації архітектури, наприклад шляхом переходу до мікросервісної структури.

До функціональних обмежень також належить відсутність повної автономності системи у випадку виникнення помилок або нестандартних ситуацій. У деяких випадках для усунення збоїв необхідне втручання адміністратора, оскільки не всі процедури відновлення реалізовані автоматично. Також програмне забезпечення не підтримує розширені засоби машинного навчання чи прогнозування майбутнього водоспоживання, що могло б значно підвищити його аналітичні можливості.

2.5 Припущення й залежності

Розробка, функціонування та ефективне використання програмної системи базується на низці припущень та залежностей, які мають бути дотримані для забезпечення коректної та стабільної роботи системи в реальному середовищі.

Припускається, що у приміщеннях, де буде впроваджено програмну систему, встановлено відповідне обладнання для вимірювання споживання води, зокрема сенсори потоку або аналогічні IoT-пристрої, які здатні надавати точні цифрові показники обсягів води у визначених інтервалах часу. Це обладнання має бути сумісним із прийнятим протоколом передачі даних, зокрема підтримувати передачу інформації у форматі, що інтерпретується серверною частиною системи.

Припускається також, що комп’ютери та сервери, на яких розміщено програмну систему, мають мінімально необхідні технічні характеристики для запуску програмного забезпечення, зокрема підтримку Microsoft Windows, встановлене середовище .NET Framework та інші супутні бібліотеки. Усі компоненти повинні бути оновлені до актуальних версій для забезпечення сумісності та захисту від потенційних вразливостей.

Важливою умовою для роботи програмної системи є стабільне інтернет-з’єднання, оскільки передача даних від сенсорів, робота з вебінтерфейсом та взаємодія між клієнтською і серверною частинами реалізується через REST API. У разі порушення мережевого зв’язку обмін даними може бути призупинений, що призведе до втрати актуальності інформації або переривання моніторингу.

У процесі експлуатації система передбачає, що користувачі володіють базовими навичками роботи з комп’ютером, зокрема можуть заповнювати форми, працювати з елементами інтерфейсу, переглядати графіки й звіти. Вважається, що адміністративні дії виконуються особами, які мають відповідні повноваження та технічну компетентність.

**3 КОНКРЕТНІ ВИМОГИ**

3.1 Вимоги до зовнішніх інтерфейсів

Програмна система має низку вимог до зовнішніх інтерфейсів, що забезпечують взаємодію між користувачем, клієнтською частиною, сервером, базою даних та зовнішніми пристроями збору даних.

Графічний інтерфейс користувача реалізується у двох основних формах: у вигляді десктопного застосунку, створеного на базі Windows Forms, а також у вигляді вебінтерфейсу, реалізованого за допомогою технологій HTML, CSS, JavaScript та фреймворку React.js. Десктопний інтерфейс призначено для локального використання системи адміністраторами та зареєстрованими користувачами, тоді як вебінтерфейс надає можливість віддаленого доступу до функціоналу з будь-якого пристрою через браузер. Інтерфейс користувача має бути інтуїтивно зрозумілим, багатомовним (мінімум українська та англійська мови), з підтримкою доступності й можливістю масштабування для різних роздільних здатностей екрана.

Програмна система передбачає взаємодію з сенсорними IoT-пристроями, які вимірюють споживання води. Дані з пристроїв передаються на сервер за допомогою протоколу HTTP або MQTT, у форматі JSON. Система повинна приймати ці повідомлення, перевіряти їх на коректність, зберігати у базу даних і використовувати для обчислення та візуалізації показників у реальному часі.

Серверна частина системи реалізована з використанням REST API, що забезпечує взаємодію між клієнтами (десктопними і веб) та базою даних. Інтерфейс API має підтримувати стандартні HTTP-методи (GET, POST, PUT, DELETE), а також аутентифікацію користувачів. Вимоги до API включають відповідність сучасним стандартам RESTful-архітектури, захист від несанкціонованого доступу, можливість обробки помилок і відправлення відповідних повідомлень клієнтам.

Інтерфейс взаємодії з базою даних реалізується за допомогою Entity Framework, що надає ORM-засоби для абстракції запитів і взаємодії з SQL Server. Вимоги до інтерфейсу бази даних включають забезпечення цілісності, уніфікованої структури даних, збереження історії змін (логування), а також швидке виконання запитів за допомогою індексів і оптимізації запитів.

Усі зовнішні інтерфейси повинні підтримувати механізми захисту даних, включаючи шифрування переданої інформації, захист паролів (зберігання в хешованому вигляді), контроль доступу до ресурсів та обмеження прав користувачів відповідно до їхніх ролей. Для цього передбачено розмежування прав доступу на рівні інтерфейсів та окремі API-ендпоінти для адміністраторів, користувачів і пристроїв.

3.2 Властивості програмного продукту

Програмний продукт має сукупність властивостей, які забезпечують його ефективну роботу в умовах постійного моніторингу водоспоживання у приміщеннях. Однією з основних властивостей є здатності системи охоплювати всі ключові процеси, що стосуються збору, обробки, зберігання та візуалізації інформації про обсяги використаної води. Система реалізує модулі авторизації та автентифікації користувачів, ведення обліку споживання води, управління користувачами та пристроями, встановлення і контроль лімітів, перегляд статистики за обраний період, а також отримання повідомлень про перевищення порогових значень.

Продукт має високу зручність використання завдяки інтуїтивно зрозумілому інтерфейсу, що реалізований у десктопному застосунку на основі WinForms і вебінтерфейсі з використанням React.js. Обидва інтерфейси дозволяють працювати з системою користувачам з базовими навичками користування комп’ютером, завдяки логічно побудованому меню, зрозумілим позначенням та підтримці локалізації.

Програмна система відзначається надійністю, оскільки використовує сучасні технології перевірки достовірності даних, контролю доступу, обробки винятків та ведення журналу дій. Усі критичні процеси — зокрема запис до бази даних, надсилання запитів до серверу та обробка показників пристроїв — виконуються з використанням засобів перевірки цілісності та обробки помилок.

Суттєвою властивістю є розширюваність, яка полягає у можливості додавання нових функціональних модулів без потреби кардинальної перебудови архітектури. Завдяки модульному підходу до реалізації, система може бути доповнена новими видами пристроїв, алгоритмами обробки даних або додатковими звітами.

Продукт має властивість масштабованості, що дозволяє використовувати його як у невеликих офісних приміщеннях, так і у багатоповерхових будівлях або промислових об’єктах. Архітектура системи орієнтована на роботу з декількома пристроями одночасно, кожен з яких може бути асоційований з окремою зоною або користувачем. Таким чином, система може обробляти великі обсяги даних без втрати продуктивності.

Значною властивістю є підтримка роботи в реальному часі. Дані, що надходять від сенсорів споживання води, передаються до серверної частини із мінімальною затримкою і візуалізуються, що дозволяє оперативно реагувати на зміни ситуації.

Програмна система забезпечує сумісність з сучасними інструментами розробки та експлуатації. Вона створена з використанням C#, .NET Framework, SQL Server та Entity Framework, що забезпечує її інтеграцію з іншими системами та гнучкість у налаштуванні.

Безпека системи гарантується застосуванням методів автентифікації, хешування паролів, авторизації з розмежуванням прав доступу, захистом API-ендпоінтів, а також можливістю створення резервних копій бази даних. Завдяки цьому система може бути використана в середовищах із підвищеними вимогами до збереження інформації.

3.3 Атрибути програмного продукту

Програмний продукт має високий рівень надійності, що забезпечується через реалізацію механізмів обробки винятків, перевірку вхідних даних, логування дій користувачів. У разі виникнення помилки система не припиняє роботу, а коректно повідомляє про проблему і зберігає стабільність усіх процесів.

Продуктивність системи досягається завдяки використанню ефективного програмного коду на мові C#, оптимізованих запитів до бази даних SQL Server із застосуванням Entity Framework, а також кешування оброблених результатів. Це дозволяє швидко обробляти великі обсяги даних навіть при одночасному використанні кількома користувачами.

Система характеризується високою зручністю у використанні для різних категорій користувачів. Інтерфейс WinForms є інтуїтивно зрозумілим, а для віддаленого доступу створено вебінтерфейс на основі React.js з адаптивним дизайном, що дозволяє працювати з системою через браузер на ПК або мобільному пристрої.

Розроблений програмний продукт підтримує портативність та сумісність з іншими системами. Завдяки REST API, система може бути інтегрована з іншими інформаційними рішеннями, зокрема з мобільними застосунками або системами «розумний будинок», що розширює сфери її застосування.

Безпека реалізується через автентифікацію користувачів, розмежування ролей (адміністратор, звичайний користувач), хешування паролів та захищену передачу даних між клієнтом і сервером. Також передбачено можливість збереження резервних копій бази даних, що гарантує збереження критично важливої інформації.

Система має гнучкість у налаштуванні, що дозволяє змінювати параметри обліку, вводити нові пристрої, конфігурувати періоди контролю та межі допустимого водоспоживання. Це робить її адаптивною до різних сценаріїв використання — від квартир до великих офісних будівель.

3.4 Вимоги базі даних

Для реалізації бази даних у системі використовується Microsoft SQL Server, який забезпечує високу продуктивність, масштабованість та стабільність при роботі з великими обсягами даних. Створення та супровід структури бази даних здійснюється за допомогою засобів Microsoft SQL Server Management Studio (SSMS), а доступ до даних з боку програми реалізується через технологію Entity Framework, що дозволяє працювати з базою на рівні об’єктів, зменшуючи кількість прямого SQL-коду.

У базі даних повинні бути реалізовані таблиці для зберігання таких сутностей, як користувачі, IoT девайси, записи про споживання води, межі (ліміти) споживання, ролі користувачів, налаштування сповіщень, резервні копії. Кожна таблиця повинна мати первинний ключ, чітко визначені типи даних для кожного стовпця, зв’язки між таблицями реалізуються через зовнішні ключі з підтримкою референтної цілісності. Зокрема, для записів водоспоживання обов’язковими є поля дати й часу реєстрації, обсягу використаної води, а також ідентифікатора пристрою, до якого відноситься цей запис.

Важливою вимогою є підтримка механізмів фільтрації та сортування даних для формування звітів та аналітики. Також передбачається реалізація представлень (views) і збережених процедур (stored procedures) для спрощення доступу до стандартних операцій, таких як підрахунок споживання за певний період, перевірка відповідності лімітам, генерація повідомлень при перевищенні норм.

База даних повинна забезпечувати високий рівень безпеки. Усі облікові дані користувачів, зокрема паролі, мають зберігатися у зашифрованому вигляді за допомогою криптографічних хеш-функцій. Доступ до бази даних повинен бути обмежений авторизованими запитами, які здійснюються через програмний інтерфейс, а не напряму.

База даних має бути масштабованою, з можливістю обробки даних для великої кількості пристроїв і користувачів одночасно, що забезпечить стабільну роботу системи в умовах зростання обсягу інформації.

3.5 Інші вимоги

Система повинна забезпечувати зручну локалізацію інтерфейсу користувача. На першому етапі реалізації передбачено двомовну підтримку — українською та англійською мовами, з можливістю швидкого перемикання мовного режиму через елемент інтерфейсу. Локалізація має охоплювати всі текстові елементи, включаючи повідомлення про помилки, назви кнопок, підказки, написи у вкладках та формах, а також одиниці вимірювання.

До інших важливих вимог належить функціональність резервного копіювання даних. Програмна система повинна мати можливість створювати резервні копії бази даних вручну або автоматично за розкладом. Функція резервного копіювання необхідна для збереження цілісності та доступності інформації у разі аварійного завершення роботи, втрати даних або збоїв апаратного забезпечення. Повинна бути реалізована також можливість відновлення даних із резервної копії у повному обсязі.

Система має працювати у середовищі Windows та підтримувати як настільний інтерфейс на основі WinForms, так і вебінтерфейс, створений за допомогою React.js. Це дає змогу користувачам мати доступ до інформації як з локального комп’ютера, так і через браузер у мережі, що підвищує мобільність та гнучкість використання системи. Дані, що відображаються у реальному часі, мають оновлюватися без потреби в ручному перезавантаженні сторінки.

Серед нефункціональних вимог слід зазначити потребу в модульності програмного коду. Архітектура системи має бути спроєктована таким чином, щоб дозволити швидке оновлення, додавання нових функціональних блоків або зміну існуючих без необхідності повної переробки системи.

Система повинна мати низькі апаратні вимоги для клієнтської частини, дозволяючи її встановлення на середньостатистичні офісні ПК без втрати продуктивності. Серверна частина має бути оптимізована для роботи в середовищах з обмеженими ресурсами, із можливістю масштабування у випадку збільшення навантаження, наприклад, при додаванні нових приміщень або підключенні більшої кількості пристроїв обліку води.

До інших вимог також належить інтеграція з IoT-пристроями. У перспективі система повинна підтримувати взаємодію з розумними водомірами, які передаватимуть показники у систему в автоматичному режимі. Такий підхід дозволяє уникати ручного введення даних, підвищити точність обліку та оперативність аналізу.

Крім того, система повинна підтримувати обробку даних про перевищення лімітів споживання та надсилання сповіщень користувачам.

Таким чином, інші вимоги до програмного забезпечення AquaTrack забезпечують додаткову функціональність, гнучкість, зручність використання, інтеграцію з розумними пристроями, безпеку та надійність системи, що є важливими аспектами для її подальшого масштабування і застосування у різних типах приміщень.