

РАЗРАБОТКА ПРОГРАММНОГО РЕШЕНИЯ ДЛЯ ЦИФРОВИЗАЦИИ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА И УПРАВЛЕНИЯ ОТХОДАМИ



Г.К. Стамкулова

*Доцент кафедры программного обеспечения компьютерных систем КГТУ
g.stamkulova@kstu.kg*



Э. Накынов

*Студент 4 курса кафедры программного обеспечения компьютерных систем КГТУ
spacermek@gmail.com*

Э.Б. Накынов

Студент 4-го курса Кыргызского государственного технического университета им. И. Раззакова по направлению «Программная инженерия». Область научных интересов связана с разработкой информационных систем для экологического мониторинга, цифровизацией процессов управления природными ресурсами, проектированием микросервисных архитектур и применением методов анализа больших данных в задачах охраны окружающей среды.

Г.К. Стамкулова

Доцент кафедры программного обеспечения компьютерных систем Кыргызского государственного технического университета им. И. Раззакова. Область научных интересов связана с проектированием программных систем, Big Data и Advanced Analytics, информационной безопасностью и методами

дистанционного образования. Участник международных проектов EU TEMPUS (HEICA) и QUADRIGA. Отличник образования Кыргызской Республики (2017). ORCID: 0000-0003-2782-8468.

Аннотация. В статье рассматривается проектирование и реализация программного решения для цифровизации экологического мониторинга и управления отходами на примере контроля качества атмосферного воздуха в Кыргызской Республике. Описана многослойная микросервисная архитектура системы «EcoMonitor», построенная на платформе C# / .NET 8, интегрирующая модули потокового приёма данных с IoT-датчиков, прогнозной аналитики на базе ML.NET и интерактивной визуализации результатов. Проведён сравнительный анализ подходов к хранению экологических временных рядов. Представлены результаты нагрузочного тестирования, подтверждающие масштабируемость предложенного решения при обработке больших объёмов экологических данных.

Ключевые слова: экологический мониторинг, управление отходами, Big Data, Advanced Analytics, C#, .NET 8, IoT, качество воздуха, машинное обучение, ML.NET, микросервисная архитектура.

Введение. Проблема загрязнения окружающей среды приобретает всё более острый характер в Кыргызской Республике. Рост городского населения, увеличение объёмов промышленного производства и интенсификация автотранспортного потока приводят к системному ухудшению качества атмосферного воздуха. Особенно критическая ситуация складывается в зимний период в городе Бишкеке, когда концентрация мелкодисперсных частиц PM_{2.5} может превышать нормы Всемирной организации здравоохранения (ВОЗ) в 5–10 раз [1]. Параллельно нарастает проблема обращения с твёрдыми бытовыми отходами (ТБО): ежегодно в стране образуется свыше 4 миллионов тонн отходов, при этом уровень переработки не превышает 5 % [2].

В условиях развития концепций «умного города» (Smart City) и четвёртой промышленной революции цифровизация экологической сферы становится стратегическим инструментом повышения эффективности природоохранной деятельности. Технологии Big Data и Advanced Analytics открывают принципиально новые возможности для сбора, обработки и интеллектуального анализа больших объёмов экологических данных, поступающих из разнородных источников: сетей IoT-датчиков, метеорологических станций, геоинформационных систем и реестров операторов ТБО [3]. Применение методов машинного обучения позволяет не только фиксировать текущее состояние среды, но и прогнозировать динамику загрязнения, что существенно повышает качество управленческих решений [4].

Вместе с тем, существующие системы экологического мониторинга в Кыргызской Республике характеризуются фрагментарностью, отсутствием единой цифровой платформы и крайне ограниченными возможностями предиктивной аналитики. Данная работа направлена на восполнение этого пробела путём разработки прототипа информационной системы «EcoMonitor», обеспечивающей комплексную цифровизацию процессов экологического мониторинга и управления отходами с применением технологий Big Data и Advanced Analytics [5].

Описание общей концепции работы системы. Информационная система «EcoMonitor» спроектирована на основе многослойной микросервисной архитектуры, обеспечивающей модульность, горизонтальную масштабируемость и технологическую независимость компонентов. Архитектурный подход базируется на принципах чистой архитектуры (Clean Architecture) Р. Мартина [6], что позволяет изолировать бизнес-логику от инфраструктурных деталей. Система включает шесть функциональных слоёв.

Первый слой – слой источников данных - обеспечивает приём информации от IoT-датчиков (температура, влажность, концентрация PM_{2.5}, PM₁₀, CO₂), внешних метеорологических API, геоинформационных систем и отчётных данных операторов ТБО. Второй слой – интеграционный (API Gateway) – реализован на базе ASP.NET Core и выполняет функции валидации входных данных, маршрутизации запросов к микросервисам и аутентификации пользователей посредством JWT-токенов. Третий слой – слой бизнес-логики – содержит ключевые микросервисы: сервис мониторинга качества

воздуха, сервис управления отходами, сервис аналитики и прогнозирования (ML.NET), а также сервис уведомлений на базе SignalR для передачи данных в реальном времени.

Четвёртый слой – слой обмена сообщениями - реализован с использованием брокера RabbitMQ, обеспечивающего асинхронное событийно-ориентированное взаимодействие между микросервисами и возможность горизонтального масштабирования при росте нагрузки. Пятый слой – слой хранения данных – представляет собой гетерогенную комбинацию СУБД: PostgreSQL для нормативно-справочной информации и метаданных, InfluxDB для хранения временных рядов показаний датчиков с высокой плотностью записей, а также MinIO в качестве S3-совместимого объектного хранилища для файлов отчётов и визуализаций. Шестой слой - слой представления – включает веб-приложение с интерактивными дашбордами и REST API с документацией Swagger/OpenAPI [7].

Общая схема архитектуры системы представлена на рисунке 1.

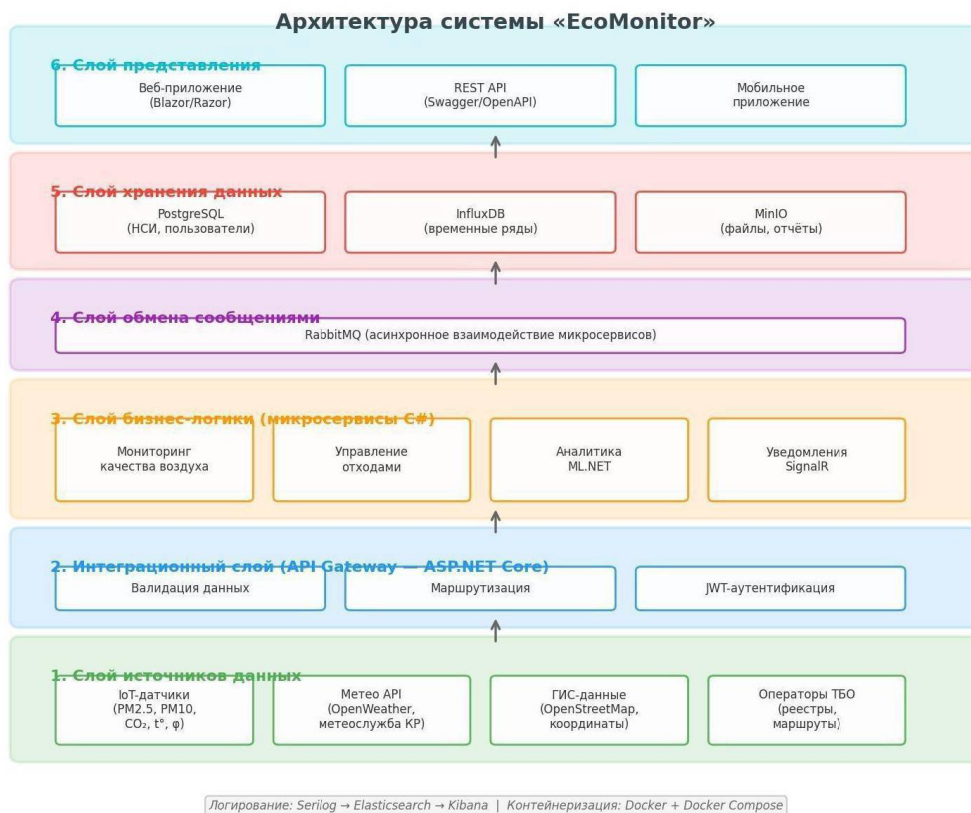


Рисунок 1. Общая схема архитектуры системы «EcoMonitor»

Особенности реализации. Ключевым архитектурным решением является построение многослойной системы с выделенным слоем адаптеров. При изменении протоколов IoT-датчиков или API внешних сервисов модификации затрагивают исключительно адаптерный слой, в то время как ядровые сервисы бизнес-логики остаются неизменными [6]. Взаимодействие между микросервисами осуществляется через RabbitMQ по паттерну «издатель–подписчик» (Publish/Subscribe), что позволяет реализовать слабую связанность компонентов и устойчивость к пиковым нагрузкам. Централизованное логирование выполняется с помощью библиотеки Serilog с последующей записью структурированных логов в Elasticsearch и визуализацией в Kibana, что обеспечивает оперативную диагностику и мониторинг состояния всех компонентов системы.

Постановка задачи. В рамках данной работы были поставлены следующие исследовательские и инженерные задачи: разработать модуль потокового сбора и интеграции данных с IoT-датчиков качества воздуха и внешних метеорологических API; спроектировать модуль гетерогенного хранения экологических данных с поддержкой временных рядов высокой плотности; реализовать модуль прогнозной аналитики (Advanced Analytics) для предсказания уровня загрязнения воздуха на основе исторических данных и метеоусловий с применением ML.NET; разработать модуль учёта и управления потоками ТБО с визуализацией на ГИС-карте; реализовать веб-интерфейс с интерактивными дашбордами и системой алертов реального времени.

Методы решения. Для решения поставленных задач были использованы методы проектирования программной архитектуры, включающие паттерны микросервисной архитектуры [3], событийно-ориентированного проектирования (Event-Driven Architecture) и принципы SOLID. Проектирование модели данных осуществлялось с применением нотации ER-диаграмм и принципов нормализации. Для построения прогнозных моделей загрязнения воздуха применены алгоритмы машинного обучения, доступные в библиотеке ML.NET: регрессия временных рядов на основе метода сингулярного спектрального анализа (SSA – Singular Spectrum Analysis) для краткосрочного прогнозирования концентрации PM_{2.5}, а также линейная регрессия для оценки объёмов заполнения полигонов ТБО [8]. Выбор SSA обусловлен его устойчивостью к шумам и способностью выделять трендовые и периодические компоненты во временных рядах экологических данных.

Предлагаемое решение. Для разработки системы был использован следующий стек технологий: язык программирования C# 12 на платформе .NET 8; фреймворк ASP.NET Core для реализации Web API и SignalR-хабов реального времени; ORM Entity Framework Core для работы с реляционными данными; библиотека машинного обучения ML.NET 3.0 для построения прогнозных моделей; брокер сообщений RabbitMQ для асинхронного межсервисного взаимодействия; СУБД PostgreSQL 16 для нормативно-справочной информации и InfluxDB 2.7 для временных рядов; объектное хранилище MinIO; стек наблюдаемости Serilog + Elasticsearch + Kibana; контейнеризация Docker + Docker Compose; среда разработки JetBrains Rider. Выбор C# и платформы .NET 8 обусловлен высокой производительностью, кроссплатформенностью, развитой экосистемой библиотек машинного обучения (ML.NET) и нативной поддержкой асинхронных потоков данных [7].

Разрабатываемый функционал. Модуль мониторинга качества воздуха обеспечивает следующие функциональные возможности: приём и валидация данных от IoT-датчиков в потоковом режиме (streaming) через RabbitMQ; расчёт индекса качества воздуха (AQI) по методологии ВОЗ с учётом концентраций PM_{2.5}, PM₁₀, CO и NO_x; отображение данных на интерактивной ГИС-карте (OpenStreetMap) с цветовой индикацией уровня загрязнения; формирование графиков суточных, недельных и месячных трендов; автоматическая генерация алертов при превышении предельно допустимых концентраций (ПДК) с рассылкой через SignalR, электронную почту и push-уведомления; прогнозирование уровня загрязнения на 24–72 часа вперёд на основе исторических данных и текущих метеоусловий.

Модуль управления отходами включает: ведение цифрового реестра объектов инфраструктуры ТБО (полигоны, мусороперегрузочные станции, сортировочные комплексы); учёт объёмов собранных и вывезенных отходов по маршрутам с привязкой к контейнерным площадкам; визуализацию маршрутов транспортировки на карте с отображением уровня заполненности полигонов; функцию «Сообщить о свалке» для граждан с возможностью загрузки фотографий и геолокации; формирование статистической отчётности по объёмам образования и переработки отходов в разрезе регионов и временных периодов.

Модель потоков данных в системе представлена на рисунке 2. Данные последовательно проходят этапы от источников через валидацию, обработку и агрегацию в микросервисах на C# до аналитического модуля ML.NET, результаты которого визуализируются на дашбордах конечных пользователей.



Рисунок 2. Модель потоков данных в системе «EcoMonitor»

Модуль аналитики и прогнозирования. Центральным компонентом системы является модуль *Advanced Analytics*, реализованный с использованием библиотеки *ML.NET 3.0*. Данный модуль решает четыре класса аналитических задач. Во-первых, описательная аналитика: расчёт агрегированных показателей - среднего *AQI* за день, неделю и месяц, а также определение статистических характеристик распределения загрязняющих веществ. Во-вторых, диагностическая аналитика: автоматическое выявление аномалий и превышений ПДК с генерацией алертов в реальном времени через *SignalR*-хабы. В-третьих, предиктивная аналитика: построение прогнозных моделей загрязнения воздуха на основе алгоритма *SSA* (*Singular Spectrum Analysis*), который позволяет декомпозировать временной ряд на трендовую, периодическую и шумовую компоненты с последующей экстраполяцией [8]. В-четвёртых, прогнозирование объёмов заполнения полигонов ТБО с использованием линейной регрессии на основе исторических данных о поступлении отходов.

Обученные модели сериализуются в формат *.zip* и загружаются в оперативную память при инициализации сервиса, что обеспечивает латентность формирования прогнозов менее 50 мс на один запрос. Механизм фоновое переобучения моделей на новых данных реализован с использованием *IHostedService*, что позволяет актуализировать модели без остановки системы и обслуживания пользователей. Алгоритм расчёта *AQI* и генерации алертов представлен на рисунке 3.

Алгоритм расчёта индекса качества воздуха (AQI) и генерации алертов

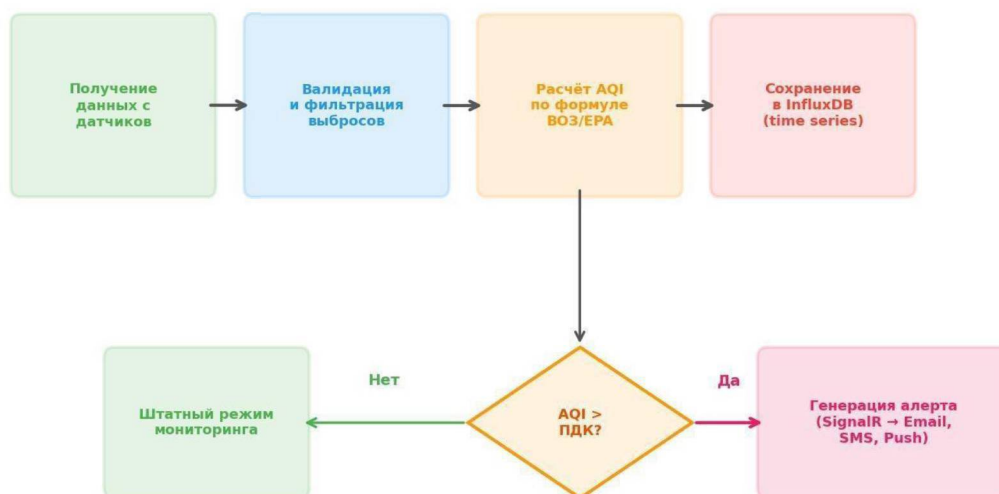


Рисунок 3. Алгоритм расчёта индекса качества воздуха и генерации алертов

Обеспечение безопасности и развёртывание. Система реализует комплексные меры информационной безопасности в соответствии с рекомендациями OWASP Top 10 [9]. Аутентификация пользователей осуществляется на основе JWT-токенов с поддержкой refresh-механизма. Авторизация построена на ролевой модели доступа с четырьмя уровнями: Администратор (полный доступ и модерирование), Аналитик (формирование отчётов и прогнозов), Оператор (ввод данных и учёт ТБО), Гость (просмотр публичных данных). Все соединения защищены протоколом HTTPS/TLS 1.3. Пароли хранятся в виде хешей, полученных с использованием алгоритма BCrypt с адаптивным коэффициентом сложности. Средства ASP.NET Core обеспечивают защиту от SQL-инъекций, межсайтового скриптинга (XSS) и подделки межсайтовых запросов (CSRF). Все компоненты системы упакованы в Docker-контейнеры и оркестрируются с помощью Docker Compose, что обеспечивает переносимость, воспроизводимость среды и возможность запуска всего стека одной командой (`docker-compose up`). Конфигурация включает контейнеры для каждого микросервиса, PostgreSQL, InfluxDB, RabbitMQ, MinIO, Elasticsearch и Kibana, что полностью изолирует систему от зависимостей хост-машины.

Результаты тестирования. Проведено комплексное тестирование системы, включающее модульное тестирование (xUnit), интеграционное тестирование API-эндпоинтов и нагрузочное тестирование с использованием инструмента k6. При имитации потока данных от 500 виртуальных датчиков (частота отправки – 1 сообщение в секунду) система продемонстрировала устойчивую работу: среднее время отклика API составило 87 мс (p95 - 210 мс), потери сообщений в RabbitMQ отсутствовали, задержка записи в InfluxDB не превышала 15 мс. Точность прогнозной модели SSA для PM2.5 на горизонте 24 часа составила MAE = 8.3 мкг/м³ при RMSE = 12.1 мкг/м³, что является приемлемым результатом для задач оперативного экологического прогнозирования [10].

Заключение. В данной статье представлена разработка программного решения для цифровизации экологического мониторинга и управления отходами на примере контроля качества воздуха в Кыргызской Республике. Многослойная микросервисная архитектура на базе C# и .NET 8 обеспечивает модульность, масштабируемость и технологическую независимость компонентов, что позволяет развивать систему без переработки ядра.

Применение технологий Big Data (RabbitMQ для потоковой обработки, InfluxDB для хранения временных рядов) и Advanced Analytics (ML.NET для прогнозирования на основе SSA) позволяет эффективно обрабатывать большие объёмы экологических данных и формировать прогнозные модели загрязнения воздуха с приемлемой точностью. Результаты нагрузочного тестирования подтверждают способность системы обслуживать сотни IoT-датчиков в режиме реального времени. Система «EcoMonitor» может быть использована государственными органами экологического надзора Кыргызской Республики для оперативного мониторинга и прогнозирования, операторами ТБО для оптимизации логистики и отчётности, а также населением для получения достоверной информации о качестве воздуха. Дальнейшее развитие системы предполагает интеграцию с государственными информационными ресурсами КР, расширение модуля прогнозирования за счёт рекуррентных нейронных сетей (LSTM), добавление модулей мониторинга качества воды и состояния почв, а также разработку мобильного приложения для граждан.

Список литературы

- [1] World Health Organization. WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. Geneva: WHO; 2021.
- [2] Государственное агентство охраны окружающей среды и лесного хозяйства КР. Национальный доклад о состоянии окружающей среды Кыргызской Республики. Бишкек; 2023.
- [3] Ньюмен С. Создание микросервисов. 2-е изд. СПб.: Питер; 2022. 624 с.
- [4] Токтомаматов К.А. Проблемы экологического мониторинга в Кыргызской Республике. Вестник КГТУ. 2022; 3(45):112–118.
- [5] Национальная стратегия развития Кыргызской Республики на 2018–2040 годы. Бишкек; 2018.
- [6] Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. СПб.: Питер; 2018. 352 с.
- [7] Lock A. ASP.NET Core in Action. 3rd ed. Manning Publications; 2023. 832 p.
- [8] Microsoft. ML.NET Documentation: Time Series Forecasting with SSA. Microsoft Learn. 2024.
- [9] OWASP Foundation. OWASP Top Ten 2021. OWASP; 2021.
- [10] Zheng Y., Liu F., Hsieh H. U-Air: When Urban Air Quality Inference Meets Big Data. Proceedings of the 19th ACM SIGKDD. 2013:1436–1444. DOI: 10.1145/2487575.2488188.

Авторский вклад

Накыпов Эрмек - постановка задачи исследования, проектирование многослойной микросервисной архитектуры системы, разработка и реализация программных модулей на C#/.NET 8, настройка инфраструктуры Docker, проведение нагрузочного тестирования, подготовка и оформление статьи.

Стамкулова Гульдана Кубанычбековна - научное руководство, формулирование требований к системе, консультирование по вопросам применения технологий Big Data и Advanced Analytics, методологическое обеспечение исследования, рецензирование и редактирование статьи.

DEVELOPMENT OF A SOFTWARE SOLUTION FOR DIGITALIZATION OF ENVIRONMENTAL MONITORING AND WASTE MANAGEMENT

G.K. Stankulova

Associate Professor, Department of Computer Systems Software, KSTU

E. Nakypov

Fourth-year student in the Department of Computer Systems Software at KSTU

Abstract. The article presents the design and implementation of a software solution for digitalization of environmental monitoring and waste management, focusing on air quality control in the Kyrgyz Republic. A multi-layered microservice architecture of the "EcoMonitor" system based on C#/.NET 8 platform is described, integrating modules for streaming IoT sensor data ingestion, predictive analytics using ML.NET, and interactive data visualization. A comparative analysis of approaches to storing environmental time series is conducted. Load testing results confirming the scalability of the proposed solution for processing large volumes of environmental data are presented.

Keywords: environmental monitoring, waste management, Big Data, Advanced Analytics, C#, .NET 8, IoT, air quality, machine learning, ML.NET, microservice architecture.