



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

КУРСОВАЯ РАБОТА

НА ТЕМУ:

«Разработка базы данных для хранения и обработки
данных...»

Студент ИУ7-62Б
(Группа)

(Подпись, дата)

М.В. Мамаев
(И. О. Фамилия)

Руководитель курсовой работы

(Подпись, дата)

Д.Ю. Пудов
(И. О. Фамилия)

2025 г.

РЕФЕРАТ

Данная курсовая работа представляет собой разработку информационной системы SensorTrack Pro, предназначенной для мониторинга движущихся объектов (транспорт, грузы, техника) в реальном времени с использованием IoT-датчиков. Система анализирует телеметрические данные, отслеживает пересечение геозон и автоматически генерирует оповещения при критических событиях, таких как выход объекта за установленные границы или отклонение от маршрута. Решение ориентировано на оптимизацию логистических процессов, повышение безопасности и снижение рисков потери грузов.

Для реализации проекта выбран следующий технологический стек:

Backend: Python 3 с использованием асинхронного фреймворка FastAPI для создания API. База данных: PostgreSQL Frontend: HTML и CSS для построения интерфейса.

Ключевые слова: мониторинг, геозоны, FastAPI, PostgreSQL, телеметрия.

СОДЕРЖАНИЕ

Введение	5
1 Аналитическая часть	6
1.1 Существующие решения	6
1.2 Формализация задачи	7
1.3 Формализация данных	9
1.4 Формализация категорий пользователя	11
1.5 Модели баз данных	12
1.5.1 Основные типы моделей	12
1.5.2 Дореляционные модели	12
1.5.3 Реляционные модели	12
1.5.4 Постреляционные модели	13
1.6 Вывод	13

ВВЕДЕНИЕ

В современных условиях глобализации и цифровизации ключевую роль в оптимизации бизнес-процессов играют технологии IoT (Интернета вещей) и системы мониторинга в реальном времени. Согласно исследованию MarketsandMarkets, мировой рынок IoT-решений к 2026 году достигнет 650 млрд, а объем данных, генерируемых подключенными устройствами, ежегодно растёт на 30-50 млрд.

Проект SensorTrack Pro направлен на решение этих проблем. Система обеспечивает мониторинг движущихся объектов (транспорт, спецтехника, грузы) с помощью IoT-датчиков, анализирует телеметрию, отслеживает пересечение геозон и автоматически генерирует оповещения при критических событиях. Актуальность проекта подкреплена ограничениями существующих аналогов: например, 70

Целью курсовой работы является разработка информационной системы для мониторинга объектов в реальном времени с интеграцией IoT-устройств и аналитическим функционалом. Для достижения цели поставлены следующие задачи:

- провести анализ существующих систем мониторинга и выявить их недостатки;
- спроектировать архитектуру базы данных с поддержкой геопространственных данных;
- рассмотреть модели баз данных и выбрать подходящую;
- проанализировать существующие СУБД и выбрать нужную;
- спроектировать и разработать БД;
- спроектировать и разработать WEB-интерфейс.

1 Аналитическая часть

1.1 Существующие решения

Учитывая стремительный рост рынка IoT-решений и высокий спрос на автоматизацию логистических процессов, на рынке уже существуют решения, предоставляющие различный функционал.

Рассмотрим только самые популярные из них, такие как:

- FleetMind;
- GeoGuard;
- TrackFlow;

Выделим следующие критерии для сравнения выбранных решений:

1. гибкость зон;
2. типы метрик;
3. ролевая модель;

Результаты сравнения выбранных решений по заданным критериям представлены в таблице

Таблица 1.1 – Сравнение существующих решений

Критерий	FleetMind	GeoGuard	TrackFlow
Типы поддерживаемых зон	Только круги	Только прямоугольники	Статичные зоны
Типы метрик	Фиксированный набор метрик	Ограниченные метрики	Базовые метрики
Ролевая модель	2 роли (админ/оператор)	1 роль (админ)	2 роли без настройки прав

Таким образом, ни одно из трех рассмотренных решений не удовлетворяет всем критериям сравнения. Также стоит отметить, что все они являются зарубежными, отечественные аналоги либо отсутствуют, либо слишком непопулярны.

1.2 Формализация задачи

В рамках курсовой работы требуется разработать информационную систему для мониторинга движущихся объектов (транспорт, грузы, спецтехника) с использованием IoT-датчиков. Система должна включать:

1. Базу данных для хранения

- Данных о движущихся объектах (ID, тип, статус, привязанные датчики).
- Параметров геозон (координаты, тип зоны: круг/прямоугольник, радиус/границы).
- Телеметрии (время, местоположение, скорость, кастомизируемые метрики).
- Пользователей с ролевой моделью (администратор, оператор, аналитик).
- Событий (пересечение зон, критические показатели датчиков).

2. Веб-приложение с функционалом

- Реального времени: отображение позиций объектов на карте, оповещения о событиях.
- Управления геозонами: создание, редактирование, удаление зон.
- Аналитики: генерация отчетов по историческим данным (графики потребления топлива, статистика нарушений маршрутов).

3. Требования к системе

- Поддержка кастомизируемых метрик (например, температура, уровень топлива) через ключ-значение.

4. Разграничение прав доступа

- Администратор: управление пользователями, объектами, зонами.
- Оператор: мониторинг данных и реагирование на события.
- Аналитик: работа с отчетами и историческими данными.

5. Дополнительные задачи

- Реализация API для интеграции с внешними IoT-устройствами.
- Оптимизация запросов к БД для обработки данных в реальном времени.
- Тестирование системы на устойчивость к пиковым нагрузкам.

1.3 Формализация данных

Разрабатываемая база данных предназначена для хранения информации о движущихся объектах, датчиках, событиях, зонах, пользователях и их взаимодействии.

Таблица 1.2 – Категории данных в БД и их атрибуты

Категория	Атрибуты
Объект (Objects)	ID объекта, название, тип объекта, дата создания
Датчик (Sensors)	ID датчика, ID объекта, тип датчика, местоположение, статус, дата установки
Событие (Events)	ID события, ID датчика, временная метка, широта, долгота, скорость, тип события
Зона (Zones)	ID зоны, название зоны, координаты границ, тип зоны (круг/прямоугольник)
Оповещение (Alerts)	ID оповещения, ID события, тип оповещения, уровень критичности, сообщение, дата создания
Маршрут (Routes)	ID маршрута, ID объекта, время начала, время окончания, статус маршрута
Телеметрия (Telemetry)	ID записи, ID объекта, временная метка, уровень заряда батареи, температура, уровень сигнала
Пользователь (Users)	ID пользователя, логин, хеш пароля, роль (администратор/оператор/аналитик)
Связь объект-зона (ObjectZones)	ID объекта, ID зоны, время входа в зону, время выхода из зоны
Связь пользователь-объект (UserObjects)	ID пользователя, ID объекта, уровень доступа

Также на рисунке ?? изображена ER-диаграмма системы в нотации Чена.

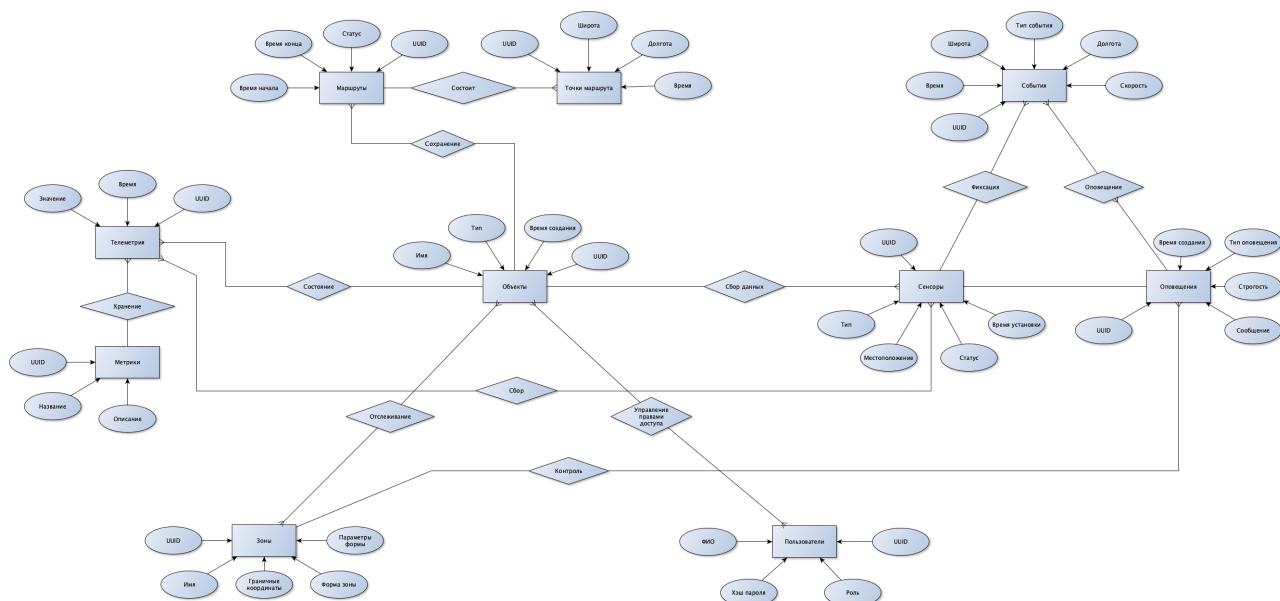


Рисунок 1.1 – ER-диаграмма в нотации Чена

1.4 Формализация категорий пользователя

Для взаимодействия с системой SensorTracker Pro выделено три категории пользователей: оператор, аналитик и администратор. Каждая роль предоставляет уникальный набор прав, соответствующий задачам управления мониторингом объектов.

Оператор имеет доступ к базовому функционалу мониторинга:

- Просмотр позиций объектов на карте в режиме реального времени.
- Отслеживание событий (пересечение зон, критические показатели датчиков).
- Управление оповещениями: подтверждение, закрытие, отправка уведомлений.
- Фильтрация объектов по типу, статусу или привязанным зонам.

Аналитик работает с историческими данными и аналитикой:

- Генерация отчетов по метрикам (расход топлива, средняя скорость, нарушения маршрутов).
- Экспорт отчетов в форматах CSV.

Администратор обладает полным контролем над системой:

- Управление пользователями: создание, удаление, назначение ролей.
- Настройка геозон (добавление, изменение границ, привязка к объектам).
- Конфигурация метрик: создание кастомизируемых параметров (например, температура, давление).
- Доступ к журналам событий и аудиту действий пользователей.



Рисунок 1.2 – Use-case Диаграмма

1.5 Модели баз данных

База данных — это структурированная совокупность данных, организованная для эффективного хранения, обработки и доступа к информации. В контексте IoT-систем, таких как SensorTracker Pro, базы данных играют ключевую роль в управлении потоками телеметрии, событиями и геопространственными данными.

Модель базы данных определяет принципы организации данных, их взаимосвязи и методы взаимодействия. Выбор модели напрямую влияет на производительность системы, масштабируемость и поддержку сложных запросов.

1.5.1 Основные типы моделей

- Дореляционные модели — иерархическая и сетевая;
- Реляционные модели — табличные структуры с отношениями;
- Постреляционные модели — NoSQL, графовые, документно-ориентированные.

1.5.2 Дореляционные модели

Дореляционные подходы, такие как иерархическая и сетевая модели, устарели для современных IoT-решений.

Иерархическая модель организует данные в древовидные структуры («родитель-потомок»), но не поддерживает связи «многие-ко-многим», что критично для отслеживания объектов в нескольких зонах. Сетевая модель позволяет множественные связи, но её сложно адаптировать под динамически изменяемые метрики (например, добавление новых типов датчиков). Обе модели обладают низкой структурной гибкостью, что делает их непригодными для систем, требующих частых изменений схемы данных.

1.5.3 Реляционные модели

Реляционные базы данных (РБД) организуют данные в таблицы с четкими связями. Для SensorTracker Pro это обеспечивает:

Целостность данных: транзакции ACID гарантируют корректность операций (например, обновление статуса объекта при пересечении зоны). Гибкость запросов: SQL позволяет агрегировать данные (средняя скорость объекта, частота событий) и работать с геопространственными расширениями (PostGIS). Структурированность: таблицы Objects, Sensors, Events связаны через внешние ключи, что упрощает анализ маршрутов и оповещений. Пример использования: хранение координат объектов в формате WKT (Well-Known Text) для визуализации на карте.

1.5.4 Постреляционные модели

NoSQL-решения (документные, ключ-значение) подходят для обработки больших потоков данных в реальном времени, но имеют ограничения:

Документные БД (MongoDB) удобны для хранения JSON-данных телеметрии, но сложны для выполнения JOIN-запросов между объектами и зонами. Графовые БД (Neo4j) эффективны для анализа связей, но избыточны для задач мониторинга. Временные ряды (InfluxDB) оптимизированы под метрики, но не поддерживают геоданные.

1.6 Вывод

Для SensorTracker Pro выбрана реляционная модель (PostgreSQL + PostGIS), так как она:

Обеспечивает целостность данных при высокой нагрузке (до 1000+ объектов). Поддерживает геопространственные запросы (поиск объектов в зоне, расчет расстояний). Позволяет гибко настраивать связи между сущностями (например, ObjectZones для отслеживания пересечений). Интегрируется с аналитическими инструментами (генерация отчетов, визуализация на карте). Альтернативные модели не покрывают все требования системы: NoSQL не гарантирует ACID, а дореляционные подходы устарели для IoT-решений.