



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 _____ ИУ «Информатика и системы управления»

КАФЕДРА _____ ИУ-1 «Системы автоматического управления»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

*Программная имитация командного пункта
управления беспилотными аппаратами
по доставке грузов*

Студент _____ ИУ1-52Б
(Группа)

(Подпись, дата)

(И.О. Фамилия)

Руководитель

(Подпись, дата)

(И.О. Фамилия)

2023 г.

**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)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ-1
(Индекс)

К.А. Неусыпин
(И.О. Фамилия)

« 11 » сентября 20 23 г.

**З А Д А Н И Е
на выполнение научно-исследовательской работы**

по теме Программная имитация командного пункта управления
беспилотными аппаратами по доставке грузов

Студент группы ИУ1-52Б
Уткин Тимофей Олегович
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
учебная

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения НИР: 25% к ___ н., 50% к ___ н., 75% к ___ н., 100% к ___ н.

Техническое задание Изучить предметную область; Разработать
архитектуру системы; Выбрать и обосновать программные технологии

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на _____ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
презентация на слайдах

Дата выдачи задания « 11 » сентября 20 23 г.

Руководитель НИР

11/09/2023
(Подпись, дата)

А. А. Николаев
(И.О. Фамилия)

Студент

11/09/2023
(Подпись, дата)

Т. О. Уткин
(И.О. Фамилия)

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	1
СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ.....	2
ВВЕДЕНИЕ	3
1. Обзор предметной области	4
1.1. Текущее состояние управления беспилотными аппаратами	4
1.2. Текущие тенденции и инновации	4
1.3. Главные проблемы подобной системы	4
2. Архитектура приложения	5
2.1. Уровень Моделирования (Model Layer):.....	5
2.2. Уровень Управления (Control Layer):.....	7
2.3. Уровень Инфраструктуры (Infrastructure Layer):	8
2.4. Уровень Интерфейсов (Interface Layer):.....	9
2.5. Уровень Безопасности (Security Layer):	10
2.6. Уровень Обработки Исключений и Логирования (Exception Handling and Logging):.....	10
3. Применяемые технологии.....	12
3.1. Java	13
3.2. Spring.....	14
3.3. PostgreSQL	15
3.4. Визуализация на карте.	16
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	18

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

БД - база данных.

БПЛА - Беспилотный летательный аппарат.

ВВЕДЕНИЕ

Данная научно-исследовательская работа посвящена созданию имитационной модели командного пункта управления беспилотными аппаратами для оптимизации логистических процессов доставки грузов. Основное внимание уделяется построению эффективной логистики, определению оптимальных маршрутов, а также обеспечению безопасности и координации действий.

Актуальность данного исследования определяется не только развитием беспилотных технологий, но и необходимостью улучшения эффективности логистических систем в условиях растущего объема грузоперевозок.

В контексте данной задачи, целью исследования является создание модели, способной эффективно управлять беспилотными аппаратами, оптимизировать маршруты и обеспечивать безопасность доставки грузов. Для достижения этой цели на данном этапе решаются следующие задачи:

1. Изучить предметную область.
2. Разработать архитектуру системы.
3. Выбрать и обосновать программные технологии.

1. Обзор предметной области

1.1. Текущее состояние управления беспилотными аппаратами

В настоящее время логистика и управление беспилотными аппаратами представляют собой важные компоненты современных технологических решений для эффективной доставки грузов. Логистические системы активно применяются в различных отраслях, включая торговлю, медицину и производство. Существующие модели включают в себя координацию маршрутов, управление складами и механизмы обеспечения безопасности.

Однако, вопреки значительному прогрессу, актуальной задачей остается разработка усовершенствованных систем управления, способных автоматизировать и оптимизировать логистические процессы с учетом особенностей беспилотных аппаратов.

1.2. Текущие тенденции и инновации

Современные тенденции в области беспилотных технологий и логистики сосредотачиваются на повышении автономности беспилотных аппаратов, улучшении алгоритмов маршрутизации и разработке систем, способных эффективно взаимодействовать с динамическим окружением. Конечно, на данный момент относительно распространены только доставки с применением наземных дронов. Доставка с помощью БПЛА же сейчас является практически неосуществимой задачей вследствие текущего ограничения на полеты над многими частями России. Однако, эта отрасль не является тупиковой, и разработать систему, способную управлять БПЛА уже сейчас было бы крайне предусмотрительным решением.

1.3. Главные проблемы подобной системы

Использование летательных аппаратов для курьерской доставки содержит в себе ряд трудностей.

Например, квадрокоптеры не имеют возможности совершать полет при плохой погоде. Сложные метеорологические условия, такие как снег, дождь,

сильный ветер, являются однозначными препятствиями, способными вывести из строя этот беспилотный летательный аппарат.

Трудности касаются доставки в многоэтажные дома, бизнес-центры, поскольку для организации этого процесса требуется написание специального сложного программного обеспечения.

Летательный аппарат не является стопроцентно защищенным, поэтому подвержен краже вместе с перевозимым им грузом путем перехвата или кибератаки.

Учитывая эти аспекты, необходимо внимательно рассматривать и дорабатывать технологические и организационные аспекты использования беспилотных курьеров для обеспечения надежности и эффективности данной системы доставки.

2. Архитектура приложения

Архитектура данной имитационной модели командного пункта управления беспилотными аппаратами должна быть организована в соответствии с принципами модульности, масштабируемости и удобства поддержки. Для решения представленных задач систему необходимо разделить на различные уровни.

2.1. Уровень Моделирования (Model Layer):

Уровень моделирования выполняет ключевую роль в представлении основных объектов и процессов, которые подлежат симуляции. Этот уровень охватывает все аспекты, связанные с представлением беспилотных аппаратов, складов и командного пункта, а также их взаимодействия в ходе выполнения логистических операций. Данный уровень решает следующие задачи:

1. Представление Объектов:

- **Беспилотные аппараты (Drones):**
 - *Цель:* Моделирование физических и технических характеристик беспилотных аппаратов, их поведения и взаимодействия с окружающей средой. Предоставление системе реалистичной

имитации работы беспилотных аппаратов, учет их особенностей при планировании и выполнении миссий.

- **Склады (Warehouses):**

- *Цель:* Моделирование структуры складов, мест хранения грузов, процессов приема и отгрузки. Обеспечение реалистичного воспроизведения операций на складах, что важно для оптимизации логистических процессов.

- **Командный пункт (Command Center):**

- *Цель:* Моделирование системы управления, координации беспилотных аппаратов и выполнение логистических операций. Создание центрального узла, который управляет всей системой, принимает решения и обеспечивает эффективное взаимодействие всех компонентов.

2. Операционная Логика:

- *Цель:* Определение логики работы каждого объекта и их взаимодействия в ходе симуляции. Гарантия того, что имитационная модель будет отражать реальные процессы и операции, связанные с управлением беспилотными аппаратами и логистикой доставки грузов.

3. Взаимодействие Объектов:

- *Цель:* Определение правил взаимодействия между беспилотными аппаратами, складами и командным пунктом. Обеспечение реалистичности и координации операций, управление обменом информацией и выполнение задач по доставке грузов.

Параметры и Состояния:

- *Цель:* Учет параметров и состояний каждого объекта, влияющих на его поведение и принимаемые решения. Обеспечение уровня детализации, достаточного для моделирования конкретных характеристик и изменений состояний объектов.

4. Обработка Событий:

- *Цель:* Учет и моделирование событий, которые могут влиять на работу системы. Предоставление возможности реагировать на различные условия, события и внешние воздействия.

Уровень моделирования позволяет создать абстракцию реального мира для более эффективного решения задачи.

2.2. Уровень Управления (Control Layer):

Уровень управления решающую роль в координации и управлении различными компонентами системы. Этот уровень включает в себя логику, отвечающую за принятие решений, оптимизацию процессов и обеспечение эффективной работы беспилотных аппаратов и логистических операций. Данный уровень решает выполняет следующие функции:

1. Менеджер Беспилотных Аппаратов (Drone Manager):

- *Цель:* Координация действий беспилотных аппаратов, распределение миссий и мониторинг выполнения задач. Обеспечение согласованного и эффективного взаимодействия между беспилотными аппаратами, а также назначение им оптимальных маршрутов.

2. Система Планирования и Маршрутизации:

- *Цель:* Оптимизация маршрутов для эффективной доставки грузов, учет ограничений и определение оптимальных путей. Гарантия оптимального использования ресурсов, минимизация времени доставки и обеспечение безопасности при движении беспилотных аппаратов.

3. Система Управления Логистикой:

- *Цель:* Распределение ресурсов, учет запасов, управление заказами и обеспечение эффективной логистики. Оптимизация процессов хранения, управление потоками грузов, предотвращение конфликтов и обеспечение равномерной загрузки складов.

4. Интерфейсы API для Внешних Систем:

- *Цель:* Предоставление стандартных интерфейсов для взаимодействия с внешними системами, включая веб-приложения и другие сервисы. Содействие интеграции с другими системами, создание универсальных интерфейсов для обмена данными.

Уровень управления обеспечивает систему интеллектуальными функциями и решениями, которые необходимы для эффективной координации и управления процессами модели.

2.3. Уровень Инфраструктуры (Infrastructure Layer):

На уровне инфраструктуры модель включает в себя средства, обеспечивающие надежное хранение данных, интеграцию с геоданными, а также поддержку визуализации и взаимодействия с внешними системами. Этот уровень представляет собой основную техническую базу для работы всей системы.

1. База данных (Databases):

- *Цель:* Обеспечение устойчивого и эффективного хранения данных о беспилотных аппаратах, грузах, складах и логистических операциях. Гарантия доступности актуальной информации для всех компонентов системы, поддержка операций чтения и записи данных.

2. Интеграция с Геоданными:

- *Цель:* Использование геоданных для определения маршрутов, высот полета и других географических аспектов. Поддержка работы с пространственной информацией, обеспечение точности в процессах навигации и маршрутизации.

3. Интеграция с Системами Визуализации:

- *Цель:* Предоставление информации для визуализации работы системы в реальном времени. Создание наглядного представления процессов, обеспечение оперативного контроля и анализа данных.

4. Системы Хранения Файлов и Конфигураций:

- *Цель:* Сохранение конфигураций, настроек и промежуточных данных. Обеспечение гибкости в настройке и адаптации системы, хранение данных для восстановления после сбоев.

Уровень инфраструктуры предоставляет фундаментальные средства для хранения, обработки и обеспечения безопасности данных, а также инструменты для визуализации и интеграции с внешними сервисами. Это ядро технической архитектуры, которое поддерживает работу всех уровней системы и обеспечивает их согласованное взаимодействие.

2.4. Уровень Интерфейсов (Interface Layer):

Уровень интерфейсов представляет собой точки взаимодействия между системой и внешним миром, обеспечивая пользовательский доступ, интеграцию с другими приложениями и прозрачное управление функциональностью системы.

1. Пользовательский Интерфейс (Web UI):

- *Цель:* Предоставление пользовательского интерфейса для визуализации работы системы, мониторинга и управления ею. Обеспечение доступа к функциональности системы через интуитивно понятный и удобный веб-интерфейс.

2. API для Беспилотных Аппаратов (Drone API):

- *Цель:* Предоставление интерфейса для взаимодействия с беспилотными аппаратами, передачи команд и получения данных. Создание стандартного способа управления и мониторинга беспилотных аппаратов.

3. API для Логистических Операций (Logistics API):

- *Цель:* Обеспечение внешним системам доступа к функциональности управления логистикой, в том числе планирования маршрутов и управления складами. Интеграция с другими логистическими системами и создание унифицированного интерфейса для обмена данными.

4. Системы Уведомлений и Оповещений:

- *Цель:* Предоставление механизмов для уведомлений и оповещений пользователей о важных событиях. Обеспечение оперативного реагирования на изменения состояния системы и событий.

5. Интерфейсы для Тестирования и Отладки:

- *Цель:* Предоставление средств для тестирования, отладки и мониторинга системы. Упрощение процессов разработки и обеспечение высокого качества программного продукта.

Уровень интерфейсов обеспечивает разнообразные способы взаимодействия с системой, поддерживая как пользовательский опыт через веб-интерфейс, так и интеграцию с внешними системами через стандартизированные API. Это делает систему более гибкой и способной взаимодействовать с другими компонентами модели.

2.5. Уровень Безопасности (Security Layer):

На уровне безопасности системы реализуются два ключевых аспекта - аутентификация и шифрование, которые обеспечивают надежность и конфиденциальность данных.

1. Модуль Аутентификации и Авторизации:

- *Цель:* Гарантировать, что только авторизованные пользователи и системы имеют доступ к функциональности системы. Защита от несанкционированного доступа, обеспечение привилегированного доступа только для аутентифицированных пользователей и систем.

Уровень безопасности создает защитный барьер вокруг системы, предотвращая несанкционированный доступ и обеспечивая безопасность данных на разных этапах их передачи и обработки.

2.6. Уровень Обработки Исключений и Логирования (Exception Handling and Logging):

На уровне обработки исключений и логирования система создает механизмы для эффективной обработки ситуаций, когда что-то идет не так, и для записи событий с целью последующего анализа и отладки.

1. Обработка Исключений:

- *Цель:* Предоставление механизмов обработки ошибок и исключительных ситуаций для поддержки стабильной работы системы. Обеспечение гибкости и надежности системы в случае отклонения от нормы, предотвращение критических сбоев и снижение вероятности потери данных.

2. Система Логирования:

- *Цель:* Ведение журнала событий для отслеживания операций, выявления проблем и поддержки отладки. Создание истории действий и событий в системе, что позволяет анализировать производительность, выявлять проблемы и предоставлять информацию для быстрого реагирования на инциденты.

Уровень обработки исключений и логирования предоставляет средства для выявления и анализа проблем, которые могут возникнуть в ходе работы системы. Он позволяет оперативно реагировать на ситуации нештатного характера и обеспечивает прозрачность работы системы для поддержки ее эффективного функционирования и обслуживания. Можно сказать, что это главный модуль разработки любой системы.

Данная архитектура обеспечивает четкую структуру, разделяющую отдельные слои ответственности, что делает систему более модульной и легкой для поддержки и расширения. Каждый уровень выполняет конкретные задачи, что упрощает взаимодействие между компонентами и обеспечивает более высокую гибкость и масштабируемость проекта.

Обобщенная схема работы системы представлена на рисунке 1.

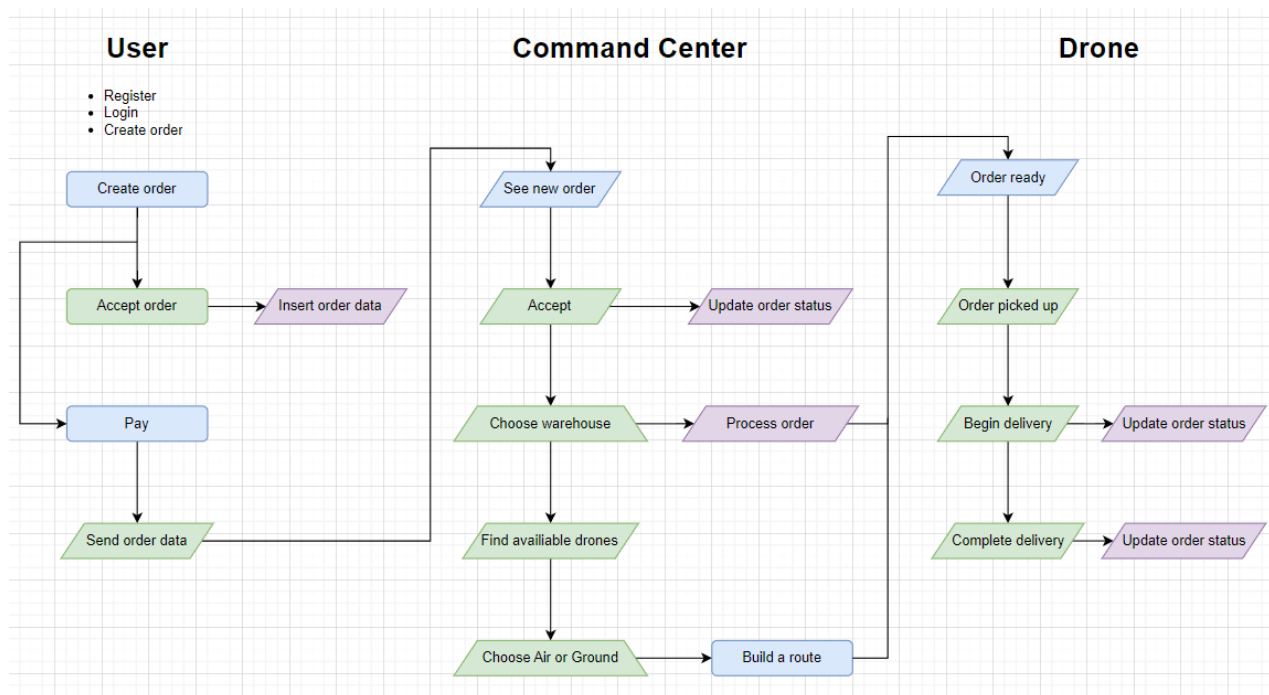


Рисунок 1 — Схема работы системы.

3. Применяемые технологии

В основе данного проекта лежит использование языка программирования Java. Java является универсальным языком программирования, обеспечивающим портативность кода практически на любые платформы. Он подходит для создания модели и реализации логистических алгоритмов именно за счет своей универсальности.

Также уникальным инструментом Java является библиотека Spring, применяемая в решении обширного ряда различных задач. Приведу ряд реализованных в данной библиотеке инструментов:

В качестве базы данных используется реляционная БД PostgreSQL, как самое современное решение для подобного рода задач. Реляционные базы данных предоставляют структурированное хранилище для данных о грузах, беспилотных аппаратах и других объектах системы.

3.1. Java

Использование языка программирования Java в данной задаче может быть обоснованным по нескольким причинам:

1. Кроссплатформенность:

Java является кроссплатформенным языком программирования, что означает, что программы, написанные на Java, могут выполняться на различных операционных системах без изменений в коде. Это обеспечивает гибкость в развертывании приложения на различных устройствах и серверах.

2. Обширная экосистема:

Java обладает обширной библиотекой и экосистемой, что облегчает разработку и поддержку приложений. Существует множество готовых библиотек и фреймворков, которые могут быть полезными при создании системы управления беспилотными аппаратами и логистики.

3. Многопоточность:

Java имеет встроенную поддержку многопоточности, что является важным аспектом для систем реального времени, таких как управление беспилотными аппаратами. Возможность эффективно управлять несколькими задачами одновременно обеспечивает отзывчивость системы.

4. Безопасность:

Java предоставляет множество механизмов безопасности, таких как управление памятью, проверка типов и возможность создания безопасных архитектурных решений. Важно обеспечить безопасность в системах управления беспилотными аппаратами, особенно в случаях, когда речь идет о передаче и обработке информации.

5. Сообщество и поддержка:

Java обладает большим сообществом разработчиков, что может быть полезным для получения советов, решения проблем и обмена опытом. Высокий уровень поддержки также означает наличие большого количества ресурсов, документации и сторонних библиотек.

3.2. Spring

В решении данной задачи есть 2 варианта – сделать монолитное приложение, которое будет работать одним целым, но в таком случае нельзя говорить о простоте масштабируемости и добавления новых возможностей; или же при помощи библиотеки Spring разработать микросервисное приложение, при этом разбив его на различные уровни ответственности, как и было предложено ранее. Почему это может быть полезно:

1. Инверсия управления (IoC):

Spring применяет принцип инверсии управления, что упрощает управление зависимостями в приложении. Это особенно полезно в системах, где требуется гибкость в управлении компонентами и их жизненным циклом.

2. Внедрение зависимостей (DI):

Фреймворк Spring обеспечивает механизм внедрения зависимостей, что улучшает модульность и тестируемость кода. Это может быть важным аспектом при разработке сложных систем управления, где важна отладка и тестирование отдельных компонентов.

3. Spring Boot для упрощенного развертывания:

Spring Boot предоставляет простой способ создания автономных, готовых к использованию приложений. Это упрощает процесс развертывания и управления приложением, особенно в случае больших и сложных систем.

4. Модульность и расширяемость:

Spring позволяет разделять приложение на модули, что упрощает его разработку и поддержку. Возможность использования различных модулей, таких как Spring MVC для веб-приложений или Spring Data для работы с базами данных, обеспечивает гибкость и расширяемость вашего приложения.

5. Безопасность и управление транзакциями:

Spring предоставляет мощные средства для обеспечения безопасности и управления транзакциями, что может быть критичным для систем, работающих с критической информацией, такой как управление беспилотными аппаратами и логистика.

3.3. PostgreSQL

Использование базы данных PostgreSQL в задаче имитации командного пункта управления беспилотными аппаратами для доставки грузов может быть обоснованным по нескольким причинам:

1. Расширенная функциональность и стандарты:

PostgreSQL предоставляет обширный набор функциональных возможностей, включая поддержку сложных запросов, геоданных и многих других расширений. Это особенно полезно в системах логистики, где требуется сложный анализ данных.

2. Геоданные и пространственные запросы:

PostgreSQL обеспечивает поддержку геоданных и пространственных запросов через расширение PostGIS. Это является важным аспектом для систем управления беспилотными аппаратами, где требуется работа с географическими данными для оптимизации маршрутов и логистики.

3. Масштабируемость и производительность:

PostgreSQL хорошо масштабируется и предоставляет эффективные механизмы оптимизации запросов. Это важно для систем, где производительность и масштабируемость играют ключевую роль, так как в вашей задаче, вероятно, будет много данных и запросов.

Обобщенная схема БД представлена на рисунке 2.

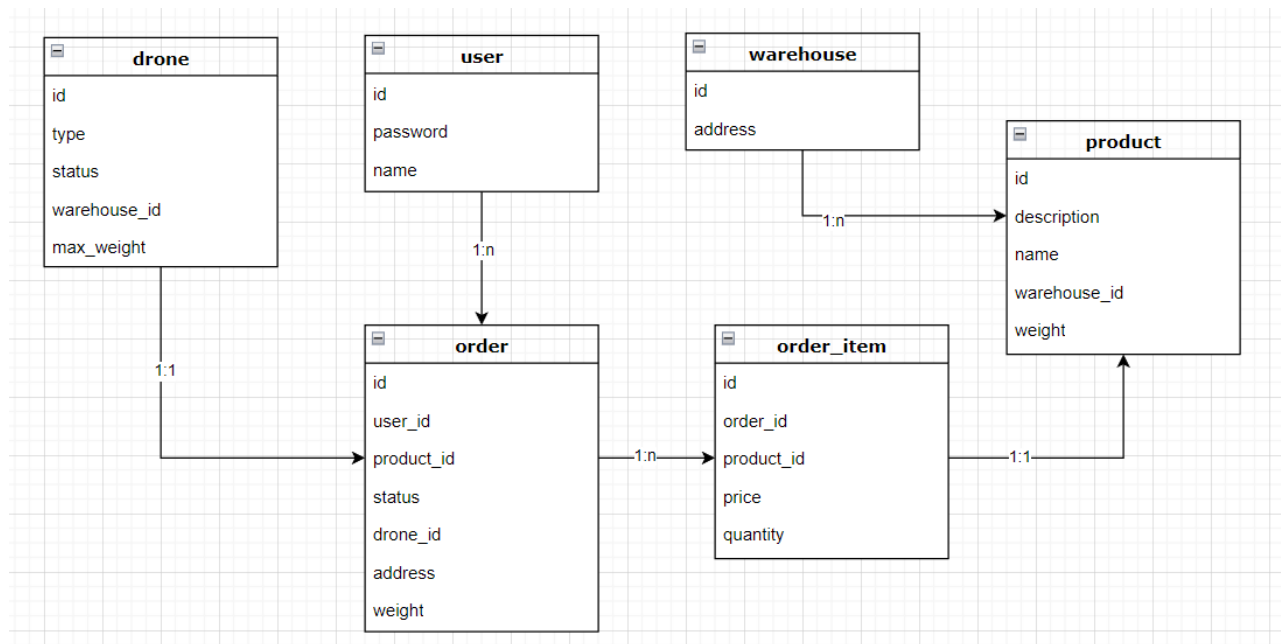


Рисунок 2 — схема БД.

3.4. Визуализация на карте.

Примерная визуализация на карте представлена на рисунке 3.

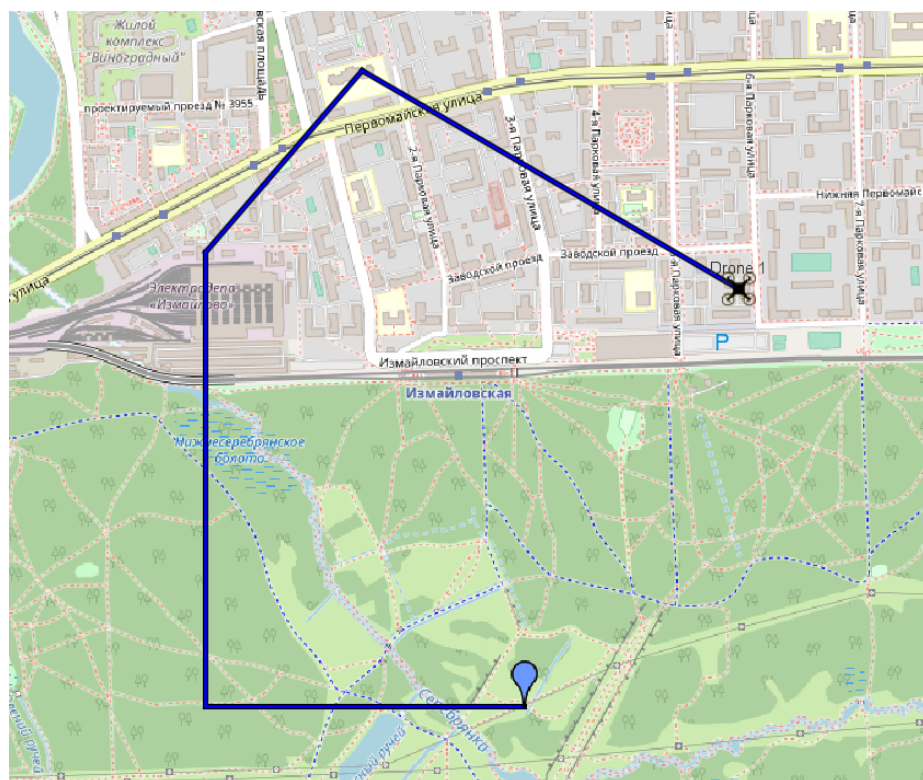


Рисунок 3 — дрон на карте.

Карта является полностью интерактивной и обновляется в реальном времени.

ЗАКЛЮЧЕНИЕ

// Какое-то заключение

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ