Министерство образования и науки Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ФИТР

Кафедра ПИн

БАКАЛАВРСКАЯ РАБОТА

Тема Разработка приложения-терминала для ИС системы доставки грузов

Руководитель

Колпаков А.А.

(оценка

(подпись) (дата)

Члены комиссии Студент ПИНз-120

(группа)

Чернышев А.Е.

(подпись) (дата)

(подпись) (дата) (подпись) (дата)

Муром 2025 г.

Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

# Факультет информационных технологий и радиоэлектроники

# «УТВЕРЖДАЮ»

# Зав. кафедрой ПИН \_\_\_\_\_\_\_\_\_\_\_А.Л. Жизняков

(подпись)

# *«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.*

## ЗАДАНИЕ

**На курсовую работу по курсу  *Разработка приложений для мобильных операционных систем***

Студенту *Чернышеву А Егр. ПИНз-120*

1. Тема работы Разработка приложения-терминала для ИС системы доставки грузов

2. Сроки сдачи студентом законченного проекта  *«21» декабря 2024 г.*

3. Исходные данные к проекту

Разработать приложение для мобильной операционной системы позволяющее работать в системе доставки грузов. Приложение должно представлять собой терминал для курьеров и менеджеров складов. Для курьеров приложение должно предоставлять функционал создания грузоперевозки, добавления грузов в доставку использую считыванием qrcode груза, удаления доставок без грузов. Для менеджера приложение должно предоставлять механизм выбора склада на котором осуществляется приемка грузов и иметь возможность выполнять прием груза используя qrcode грузов.

Дополнительные требования к разрабатываемой системе:

1. Операционная система Android, язые Kotlin;
2. Общение с сервером посредством REST API;
3. Разделение частей приложения для разных ролей;
4. Репозиторий системы контроля версий – GitHub;
5. Провести функциональное тестирование разработанного приложения.

4. Содержание расчетно–пояснительной записки (перечень подлежащих разработке вопросов).

*Аннотация (на двух языках)*

|  |
| --- |
| *Содержание* |
| *Введение* |
| *1. Анализ технического задания* |
| *2. Разработка алгоритмов* |
| *3. Проектирование работы системы* |
| *4. Руководство программиста* |
| *5. Руководство пользователя* |
| *Заключение* |
| *Список используемой литературы* |
| *Приложение 1. Текст программы* |
| *Приложение 2. Снимки окон программы (скриншоты программы)* |

5. Перечень графического материала (с точным указанием обязательных чертежей и графиков)

|  |
| --- |
| *Описание структур данных* |
| *Текст программы с комментариями* |
| *Скриншоты окон программы* |

6. Рекомендуемая литература

|  |
| --- |
| *1. Введение в разработку приложений для смартфонов на ОС Android / А.Семакова – М.: Национальный открытый Университет «ИНТУИТ», 2016* |
| *2. Колисниченко Д.Н. Программирование для Android 5. Самоучитель. — СПб.: БХВ-Петербург, 2015. — 303 с.* |
| *3. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е изд. — СПб.: Питер, 2016.* |
| *4. Гриффитс Дэвид, Гриффитс Дон Head First. Программирование для Android. 2-е изд. — СПб.: Питер, 2018. — 912 с.* |

*/*

7. Дата выдачи задания  *14.09.2024 .*

8. Календарный график работы над проектом (на весь период проектирования, с указанием сроков выполнения и трудоемкости отдельных этапов)

|  |
| --- |
| *Анализ технического задания* *10%, 3 нед* |
| *Разработка моделей данных 20%, 4 нед.* |
| *Проектирование работы системы 35%, 6 нед.* |
| *Разработка и реализация системы 70%, 12 нед.* |
| *Тестирование системы 90%, 14 нед.* |
| *Оформление пояснительной записки* *100%, 15 нед.* |

**Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(подпись)

Задание принял к исполнению (дата)

**Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Примечание. Это задание прилагается к законченному проекту.

Аннотация на русском

English annotation.

Содержание

[Введение 5](#_Toc198302281)

[1 Анализ технического задания 7](#_Toc198302282)

[1.1 Описание предметной области 7](#_Toc198302283)

[1.2 План интервью для определения бизнес-требований 7](#_Toc198302284)

[1.3 Формирование требований 8](#_Toc198302285)

[1.4 Поиск и сравнение аналогов 10](#_Toc198302286)

[1.5 Функциональные возможности 15](#_Toc198302287)

[1.6 Обоснование выбора средств реализации 17](#_Toc198302288)

[2 Проектирование архитектуры системы 19](#_Toc198302289)

[2.1 Диаграмма вариантов использований 19](#_Toc198302290)

[2.2 Диаграмма последовательнотей 20](#_Toc198302291)

[2.3 Логическая модель данных 21](#_Toc198302292)

[2.4 Физическая модель данных 21](#_Toc198302293)

[Список литературы 23](#_Toc198302294)

[Приложение А 24](#_Toc198302295)

[Приложение Б 28](#_Toc198302296)

Введение

Бурное развитие информационных технологий в последние десятилетия привело к необходимости поиска новых решений для автоматизации деятельности различных организаций, предприятий и служб. Рабочие процессы таких организаций часто связаны с обработкой и хранением больших объемов информации. Особенно это актуально для служб, занимающихся учетом данных о клиентах, товарах, доставках и других аспектах бизнеса.

Еще совсем недавно информация подобного рода хранилась в картотеках с использованием бумажных носителей. Этот процесс требовал значительных физических ресурсов, данных заносились вручную на карточки, что не только занимало много времени, но и создавало большие неудобства, повышая вероятность ошибок. Вся информация должна была быть найдена вручную, а сама картотека со временем изнашивалась, что снижало ее надежность.

Сегодня, в эпоху компьютерных технологий и автоматизации, физические картотеки и бумажные носители постепенно уступают место более современным и удобным решениям. Вместо сложных и неудобных процессов теперь используются мощные компьютерные системы, обеспечивающие быструю и надежную обработку данных. Однако, несмотря на значительный прогресс, по-прежнему существуют проблемы, такие как необходимость быстрого поиска нужной информации, обеспечение надежности хранения данных и соблюдение конфиденциальности.

Для решения этих задач используются специализированные программные продукты, которые часто объединены в крупные информационно-справочные системы. Эти системы предназначены для управления различными массивами данных, включая базы данных, и предоставляют удобный интерфейс для пользователей. Современные информационные системы включают в себя функции для добавления, редактирования, удаления данных, а также для их поиска и простого просмотра. Важным аспектом таких систем является обеспечение надежности хранения информации и предотвращение ее утрат. Реализация этих решений зависит от возможностей используемых технологий, поставленных задач и квалификации разработчика.

Разработка комплексного программного решения для автоматизации логистических процессов службы доставки грузов, включающего:

1. Кроссплатформенную мобильную систему:
   * Реализация на Kotlin для Android-устройств
   * Поддержка двух ролевых моделей (доставщик/менеджер)
   * Интеграция с QR-код технологиями
2. Серверную часть:
   * RESTful API для централизованного управления данными
   * Безопасные HTTP-коммуникации
   * Масштабируемая архитектура
3. Ключевые функциональные модули:
   * Система управления доставками (маршрутизация, трекинг)
   * Складской учет с технологией QR-верификации
   * Аналитический dashboard для менеджмента

Особое внимание уделяется:

* Обеспечению надежности хранения данных
* Реализации удобного пользовательского интерфейса
* Соблюдению требований информационной безопасности
* Оптимизации бизнес-процессов доставки

Работа представляет собой законченное решение, сочетающее современные подходы к разработке мобильных приложений с эффективной backend-архитектурой, соответствующее актуальным требованиям цифровизации логистических операций.

1 Анализ технического задания

1.1 Описание предметной области

Предметная область проекта связана с разработкой веб-приложения для автоматизации логистических процессов компании, занимающейся доставкой грузов. Основные задачи системы включают:

* Управление заказами (создание, отслеживание, отмена).
* Приём и выдача грузов на складах (ПВЗ).
* Контроль за доставками и маршрутами.
* Управление пользователями (регистрация, авторизация, роли).
* Формирование цены и обработка оплат.

Система должна обеспечивать функциональность для различных классов пользователей:

* + - * Клиенты: оформление заказов, отслеживание статусов, просмотр истории.
      * Менеджеры склада: приём и выдача грузов, управление оплатами.
      * Доставщики: приём грузов в доставку, указание параметров транспорта.
* Администраторы: управление ролями, настройка системы, аналитика.

1.2 План интервью для определения бизнес-требований

Для уточнения требований к системе необходимо провести интервью с заинтересованными лицами. Основные вопросы:

1. Какие основные задачи должна решать система?
2. Кто будет основными пользователями (клиенты, менеджеры, доставщики, администраторы)?
3. Какие функции являются приоритетными для каждого класса пользователей?
4. Какие данные необходимо учитывать при управлении заказами, складами и доставками?
5. Какие ограничения по времени и ресурсам существуют для выполнения заказов?
6. Какие требования к безопасности и авторизации пользователей?
7. Какие интеграции с другими системами необходимы (CRM, ERP, платежные системы)?
8. Какие отчеты и аналитика должны быть доступны в системе?
9. Какие требования к производительности и доступности системы?
10. Какие бизнес-цели должны быть достигнуты с помощью системы?

1.3 Формирование требований

При проектировании системы автоматизации логистических процессов ключевое значение имеет четкое определение функциональных и нефункциональных характеристик, которые обеспечат эффективную работу всех участников цепочки доставки. Исходя из анализа бизнес-процессов компании и потребностей различных категорий пользователей, система должна реализовывать комплексный подход к управлению заказами, складами и доставками, обеспечивая при этом высокий уровень безопасности, удобства взаимодействия и масштабируемости. Основные требования к системе можно структурировать следующим образом:

1.3.1 Функцпональные требования

* Управление пользователями и доступом:
  + Многоуровневая система регистрации с дифференциацией по типам пользователей (клиенты, курьеры, менеджеры складов, администраторы)
  + Механизм назначения и динамического изменения ролей с детализированными правами доступа
  + Возможность делегирования полномочий между сотрудниками
* Работа с заказами:
  + Полноценный жизненный цикл обработки заказов от создания до финального исполнения
  + Интеллектуальная система отмены заказов с учетом временных ограничений и бизнес-правил
  + Архив выполненных заказов
* Складская логистика:
  + Учет операций приема-передачи грузов
  + Интеграция с системами маркировки (QR-коды)
* Организация доставки:
  + Учет транспортных характеристик
  + Мониторинг выполнения доставок в реальном времени

1.3.2 Нефункциональные требования

* Безопасность и надежность:
  + Ролевая модель доступа с детализированными правами
* Пользовательский опыт:
  + Адаптивный интерфейс для разных устройств
  + Персонализированные рабочие пространства
* Интеграционные возможности:
  + API для подключения внешних сервисов
  + Поддержка стандартных протоколов обмена данными
* Масштабируемость:
  + Возможность наращивания функционала
  + Поддержка роста количества пользователей
  + Горизонтальное масштабирование системы

1.4 Поиск и сравнение аналогов

Для проведения всестороннего анализа рыночных решений и выявления оптимальных подходов к реализации системы рекомендуется выполнить детальное исследование существующих платформ, специализирующихся на логистике и управлении доставками. В рамках сравнительного анализа целесообразно рассмотреть следующие популярные отраслевые решения:

1. 4Logist.com - Комплексная платформа для управления логистикой, предлагающая:
   * Интегрированные инструменты управления складскими операциями
   * Систему маршрутизации и планирования доставок
   * Мобильные приложения для курьеров и клиентов
   * Аналитические инструменты для оптимизации логистических процессов
2. Битрикс24 (логистический модуль) - Корпоративное решение, включающее:
   * CRM-систему с интеграцией логистических функций
   * Инструменты управления транспортными средствами
   * Систему учета грузоперевозок
   * Возможности автоматизации документооборота
3. TMS (Transportation Management System) системы:
   * Oracle Transportation Management
4. Специализированные SaaS-решения:
   * LogistiX

При проведении анализа следует учитывать:

* Соответствие функциональных возможностей требованиям бизнеса
* Гибкость и масштабируемость архитектуры
* Уровень интеграции с существующей ИТ-инфраструктурой
* Стоимость владения и масштабирования
* Опыт внедрения в аналогичных бизнес-моделях
* Качество технической поддержки и экосистемы разработчиков

Такой комплексный подход к анализу позволит не только выбрать оптимальные технологические решения, но и избежать типичных ошибок при проектировании системы.

1.4.1 4Logist.com

Специализированная логистическая платформа для среднего и крупного бизнеса. Предлагает комплексное решение для управления цепочками поставок с акцентом на маршрутизацию и контроль доставок. Отличается глубокой отраслевой специализацией и развитыми аналитическими инструментами. Пример рабочего окна платформы представлен на рисунке 1.

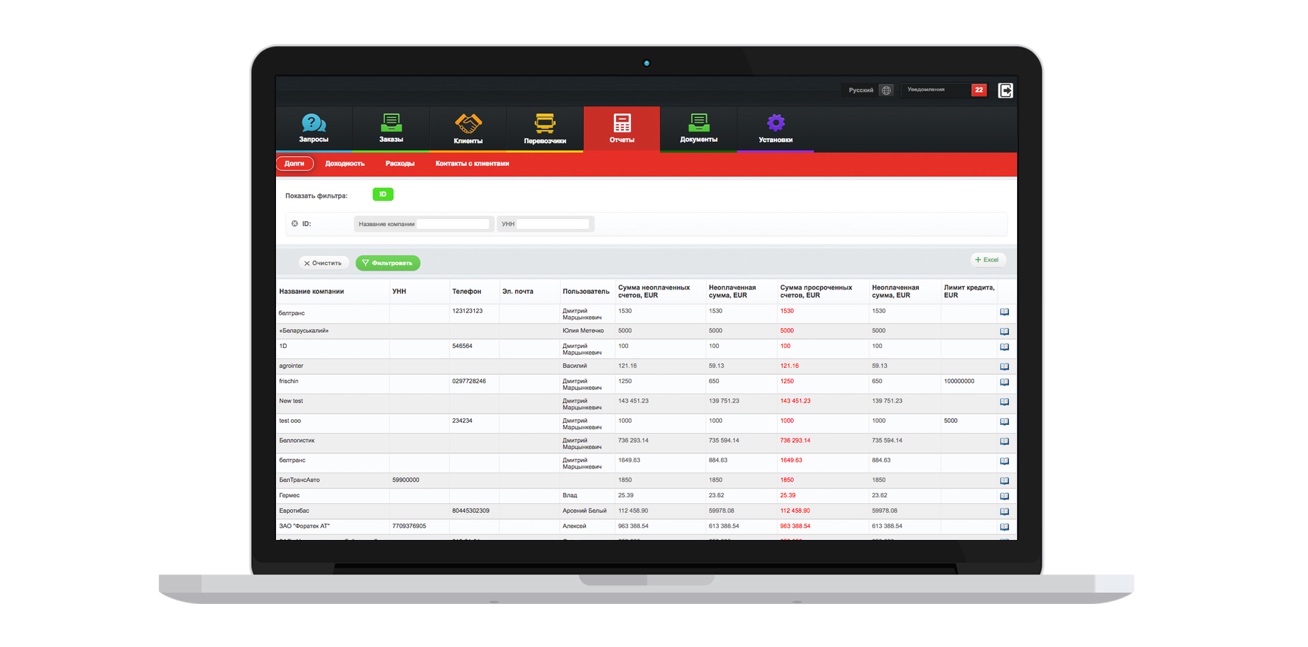


Рисунок 1 – Рабочее окно платформы 4Logist.

1.4.2 Битрикс24 (Логистический модуль)

CRM-система с расширенными возможностями для управления заказами и базовыми логистическими функциями. Оптимальна для компаний, где логистика тесно интегрирована с продажами и клиентским сервисом. Ограничена в специализированных функциях для сложной логистики. Рабочее окно модуля представлено на рисунке 2.

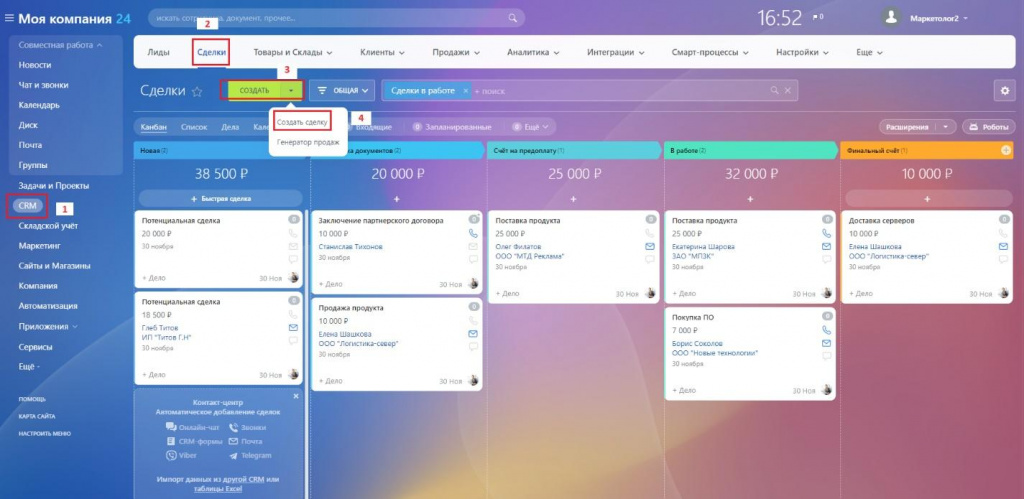


Рисунок 2 – Процесс доставки в Битрикс24.

1.4.3 Oracle Transportation Management (OTM)

Корпоративная система управления транспортом класса Enterprise. Обеспечивает глобальное управление цепочками поставок с мощными алгоритмами оптимизации. Требует значительных ресурсов для внедрения и поддержки. Организация рабочего процесса в OTM представлена на рисунке 3.

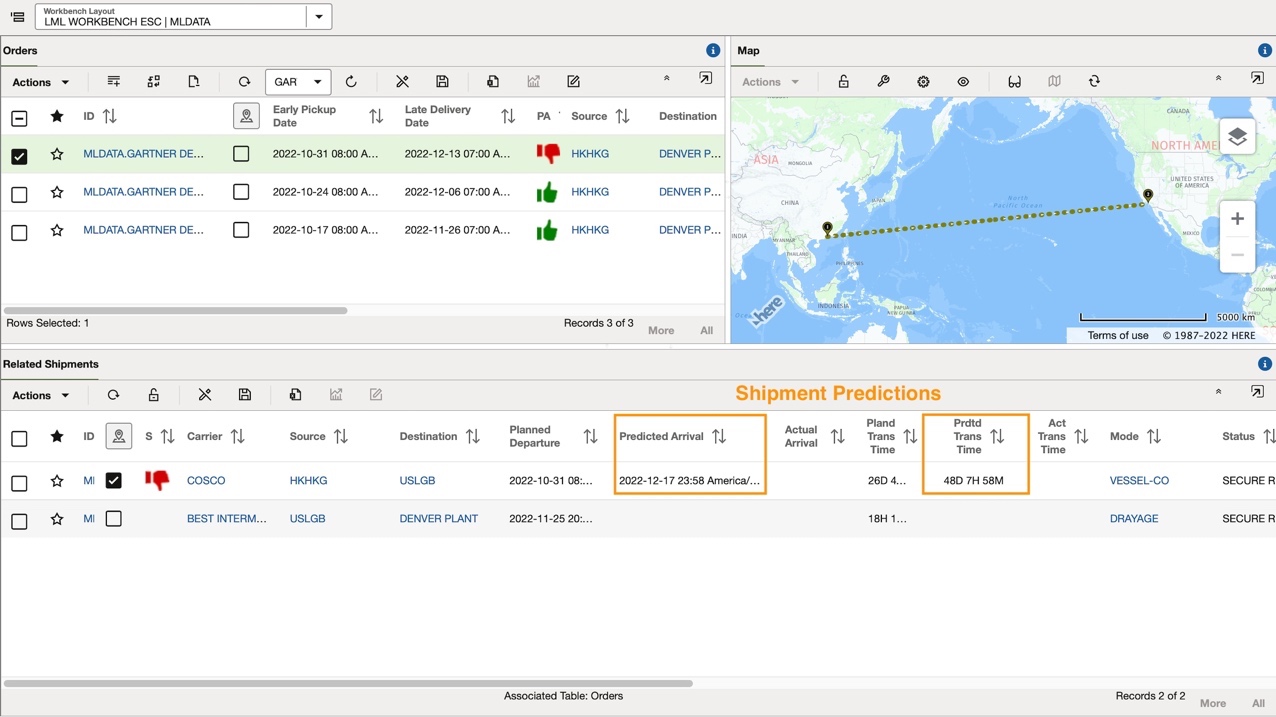


Рисунок 3 – Рабочее окно приложения Oracle Transportation Management.

1.4.4 LogistiX

Облачное SaaS-решение для оперативного управления доставками. Отличается быстрым развертыванием и гибкой подпиской. Подходит для малого и среднего бизнеса с типовыми логистическими процессами. Рабочее окно приложения представлено на рисунке 4.

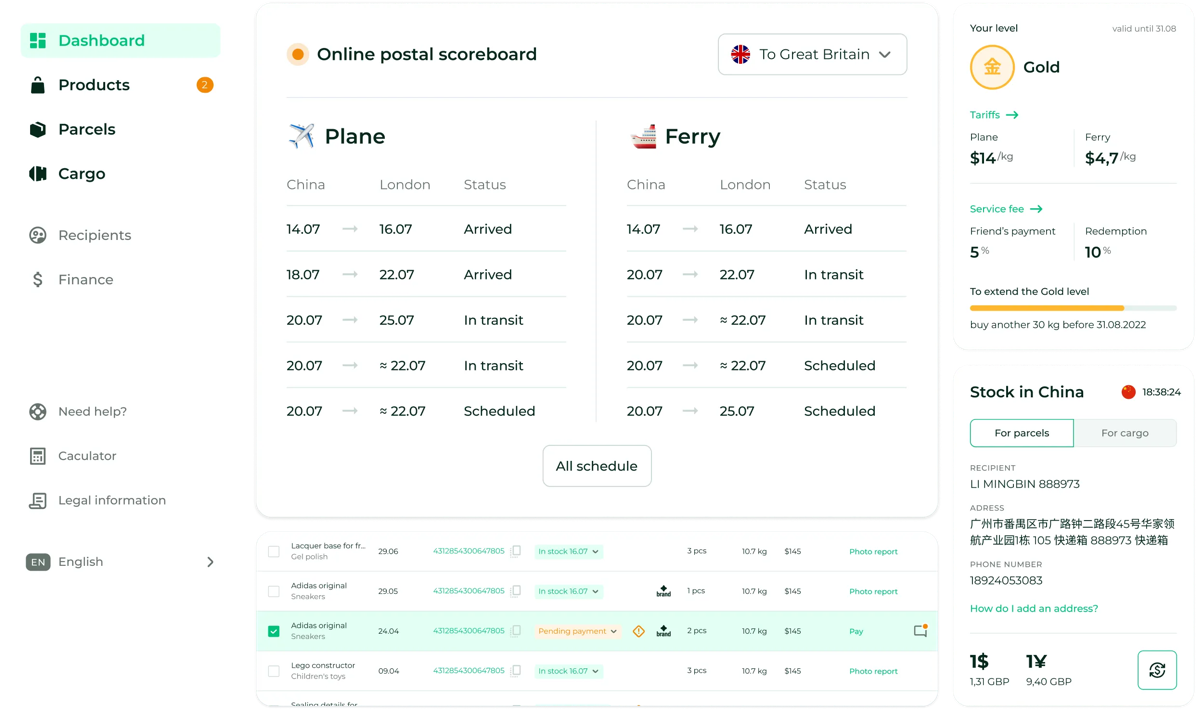


Рисунок 4 – Рабочее окно приложения LogistiX.

1.4.5 Сравнение аналогов

Было проведено сравнение характеристик рассмотренных аналогов, которое отображено в таблице 1.

Таблица 1 – Сравнение аналогов разрабатываемой система

| Параметр сравнения | 4Lаogist.com | Битрикс24 (Логистика) | Oracle TMS | LogistiX |
| --- | --- | --- | --- | --- |
| Тип решения | Специализированная логистическая платформа | CRM с логистическим модулем | Корпоративная TMS-система | Облачное SaaS-решение |
| Основные функции | Управление заказами, маршрутизация, складской учет | Управление заказами, CRM, документооборот | Планирование перевозок, оптимизация маршрутов, фрахт | Управление доставками в реальном времени |
| Интеграции | API для 1С, ERP, маркетплейсов | Встроенная интеграция с продуктами 1С | Интеграция с SAP, Oracle ERP | REST API, Webhooks |
| Мобильные приложения | Да (отдельные для клиентов и курьеров) | Только общее мобильное приложение | Нет (только веб-интерфейс) | Да (кроссплатформенное) |
| Аналитика | Встроенные BI-инструменты | Базовые отчеты | Расширенная аналитика и прогнозирование | Пользовательские дашборды |
| Ценовая модель | Подписка (от $50/пользователь) | Помесячная оплата (от $20/польз.) | Корпоративные лицензии | Pay-as-you-go |
| Плюсы | Глубокая специализация на логистике | Удобство интеграции с CRM | Мощные алгоритмы оптимизации | Гибкость и быстрое развертывание |
| Минусы | Высокий порог входа | Ограниченная логистическая функциональность | Сложность внедрения | Ограниченная кастомизация |
| Поддержка | 24/7 с SLA | Бизнес-часы | Премиум-поддержка | Чат/email поддержка |
| Масштабируемость | До 1000+ пользователей | До 500 пользователей | Неограниченная | До 300 пользователей |

1.5 Функциональные возможности

1.5.1 Управление пользователями и доступом

Регистрация и профили:

* Многоуровневая регистрация с верификацией:
  + Клиенты (email)
  + Сотрудники (корпоративная авторизация)
* Расширенные профили с:
  + Предпочтениями доставки (клиенты)
  + Квалификациями и допусками (сотрудники)
* Авторизация и безопасность:
* Ролевая модель:
* Иерархия ролей (4 уровней доступа):
  + Клиент
  + Курьер
  + Менеджер склада
  + Администратор.

1.5.2 Управление заказами

Создание заказа:

* Мультиплатформенное оформление:
  + Веб-интерфейс
  + API для интеграций

1.5.3 Складские операции

Прием грузов:

* Автоматизированная сверка:
  + По штрих-кодам/QR
* Контроль качества:
  + Весовой контроль

1.5.4 Идентификация грузов

Для идентификации грузов используются QR-коды, которые выпускает система. QR-коды содержат информацию по грузу и заказу.

1.5.6 Жизненный цикл заказа

Каждый заказ в системе проходит четко структурированный путь, обеспечивающий прозрачность и контроль на всех этапах. Разрабоан оптимизированный workflow, который автоматизирует ключевые процессы, минимизирует ручные операции и предоставляет полную видимость статусов для всех участников процесса. Отображение жизненного цикла представлено на рисунке 5.

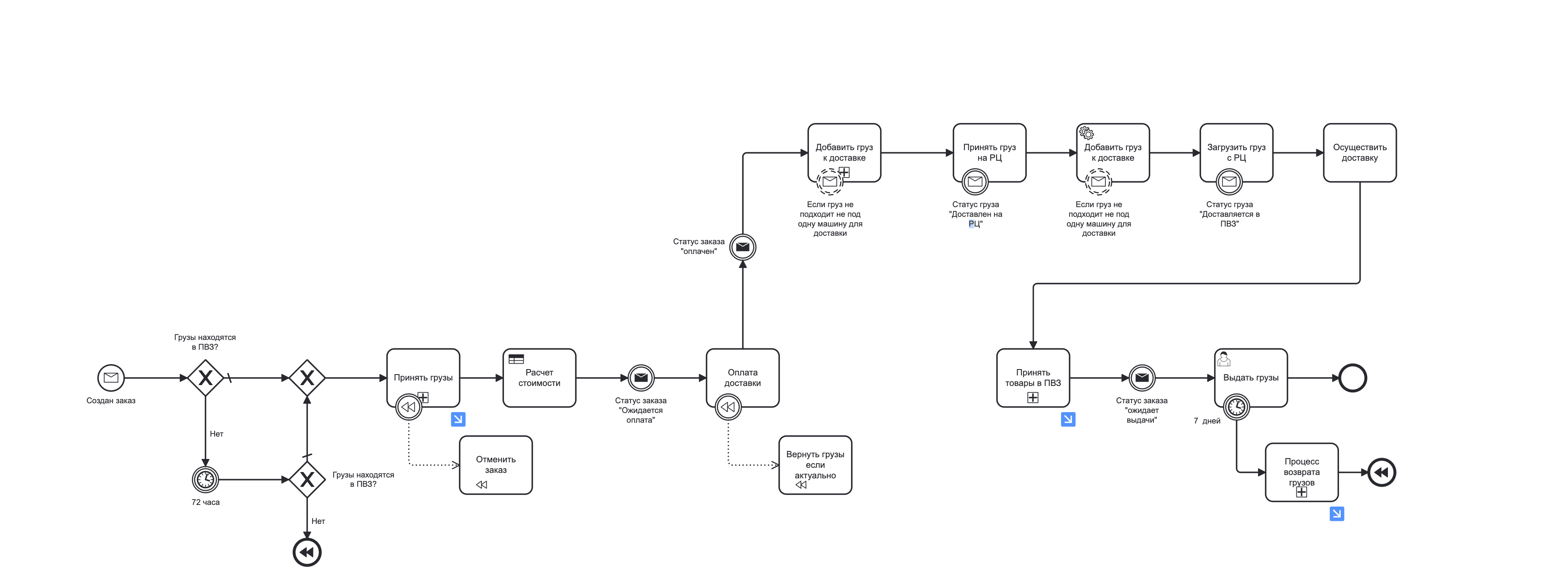


Рисунок 5 - Жизненный цикл заказа доставки.

1.6 Обоснование выбора средств реализации

1.6.1 Ruby on Rails

6. Обоснование выбора средств реализации

Для разработки системы выбраны следующие технологии:

* Backend: Ruby on Rails. Выбор обусловлен высокой производительностью, удобством разработки и поддержкой MVC-архитектуры.
* Frontend: React. Позволяет создавать динамичные и интерактивные пользовательские интерфейсы.
* База данных: PostgreSQL. Надежная СУБД с поддержкой сложных запросов и транзакций.
* Деплой: Docker + AWS/Heroku. Обеспечивает масштабируемость и удобство развертывания.

Использование данного стека технологий позволит создать надежное, производительное и легко поддерживаемое решение.

1.6.2 React + Mui/Joy

Комбинация React с библиотеками MUI и Joy UI обеспечивает:

1. Эффективность разработки:
   * Готовые компоненты для:
     + Интерактивных карт (Mapbox)
     + Таблиц с виртуализацией (TanStack Table)
     + Сложных форм (Formik + Yup)
   * ThemeProvider для быстрой смены тем
2. Особенности реализации:

<ThemeProvider theme={logisticsTheme}>

<JoyCssVarsProvider>

<MapDashboard>

<RouteOptimizer />

<RealTimeTracking />

</MapDashboard>

</JoyCssVarsProvider>

</ThemeProvider>

1. Преимущества Joy UI:
   * Специализированные компоненты для:
     + Визуализации маршрутов
     + Таблиц с большими объемами данных

1.5.3 База данных PostrgeSQL

База данных: PostgreSQL

Выбор PostgreSQL обусловлен требованиями к логистической системе:

Ключевые возможности:

1. Поддержка геопространственных данных (PostGIS)
2. Оконные функции для аналитики
3. Полнотекстовый поиск
4. JSONB для гибких схем данных

2 Проектирование архитектуры системы

Чтобы снизить количество ошибок при проектировании структуры базы данных и бизнес-логики приложения, важно наглядно представить и систематизировать информацию. Для этого были разработаны:

* Диаграмма вариантов использования (отображает взаимодействие пользователей с системой),
* Диаграмма состояний (визуализирует состояния системы и точки перехода),
* Логическая модель данных (описывает сущности и их взаимосвязи без привязки к СУБД),
* Физическая модель данных (определяет конкретную реализацию базы данных с учетом технических ограничений).

Это позволяет четко спланировать архитектуру системы и избежать недочетов на этапе разработки.

2.1 Диаграмма вариантов использований

Эта диаграмма была разработана для наглядного представления функциональных требований к системе. С её помощью удалось:

* определить ключевые сценарии взаимодействия пользователей с системой,
* выделить основные функции, которые система должна предоставлять,
* проанализировать, как эти функции удовлетворяют потребности пользователей.
* позволяет определить роли пользователей.

Таким образом, диаграмма вариантов использования стала важным инструментом для проектирования логики приложения и уточнения его функционала.

Диаграмма вариантов использования представлена на рисунке 6.

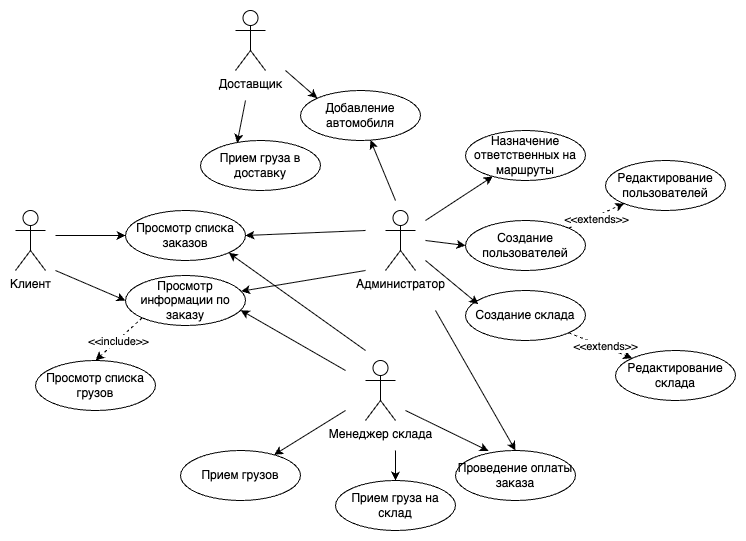


Рисунок 6 – Диаграмма вариантов использования.

2.2 Диаграмма последовательнотей

Диаграмма последовательности действий была использована для наглядного представления взаимодействия между модулями системы и её участниками в хронологическом порядке. Такой подход необходим для получения целостного представления о поведении системы во времени. Диаграмма позволила подробно отразить порядок вызовов, передаваемые параметры и характер взаимодействия компонентов, что способствовало раннему выявлению и устранению потенциальных проблем в архитектуре.

Этот тип диаграммы помогает обнаружить скрытые зависимости и возможные узкие места, которые могут негативно повлиять на производительность или привести к сбоям в работе системы. Проведённый анализ на основе диаграммы последовательности повысил качество проектных решений и позволил значительно сократить время, затрачиваемое на выявление и исправление ошибок в процессе тестирования и внедрения.

Указанная диаграммы приведены в (Приложение А. Рисунок 1–4).

2.3 Логическая модель данных

Логическая модель данных представляет собой абстрактное описание структуры информации в системе, включающее сущности, их атрибуты и взаимосвязи между ними, без учета технических аспектов хранения. Она играет ключевую роль в проектировании информационных систем, позволяя формализовать и организовать данные, необходимые для работы системы. Благодаря ей обеспечивается целостность, согласованность и корректность данных, что существенно упрощает последующую разработку физической модели, уже учитывающей конкретные технологические и аппаратные условия хранения.

Кроме того, логическая модель данных способствует упрощению и ускорению процесса проектирования, повышая качество и надежность итоговой системы. Она позволяет выявить и устранить потенциальные ошибки на ранних этапах разработки, улучшить согласованность данных и создать эффективную архитектуру для обработки и хранения больших объемов информации.

Логическая модель представлена на изображении (Приложение Б. Рисунок 1).

2.4 Физическая модель данных

Физическая модель данных была разработана для конкретизации структуры данных с учетом особенностей реализации в СУБД PostgreSQL. При построении данной модели было определено, как именно данные будут физически храниться, индексироваться и управляться в базе данных. Это включало описание таблиц, столбцов, типов данных, индексов, первичных и внешних ключей, а также взаимосвязей между таблицами.

Разработка физической модели данных позволила повысить производительность системы, обеспечить эффективное хранение информации и быстрый доступ к ней. Модель также послужила основой для создания и настройки объектов базы данных в рамках проектируемой системы, что значительно упростило процесс интеграции с существующей инфраструктурой.

Процесс создания модели включал анализ и выбор оптимальных структур хранения с учетом специфики PostgreSQL, а также использование профильной литературы [3]. Определение ключей и связей между таблицами обеспечило соблюдение целостности и согласованности данных в системе.

Физическая модель данных представлена на изображении (Приложение Б. Рисунок 2).

Список литературы

1. ГОСТ 34.201–89. Информационная технология. Взаимосвязь открытых систем. Основные положения. — М.: Изд-во стандартов, 1989. — 24 с.
2. ISO/IEC 12207:2008. Systems and software engineering — Software life cycle processes. — 3rd ed. — Geneva: ISO, 2008. — 118 p.
3. Молчанов, В. Г. Методология проектирования автоматизированных систем управления / В. Г. Молчанов. — М.: Радио и связь, 1985. — 232 с.
4. Бочкарёв, В. В. Теория информационных процессов и систем / В. В. Бочкарёв. — М.: Горячая линия — Телеком, 2009. — 382 с.
5. Глинский, Я. М. Стандарты в области информационных технологий / Я. М. Глинский. — М.: Финансы и статистика, 2006. — 224 с.
6. Вопросы анализа и моделирования информационных систем: учеб. пособие / под ред. С. И. Бештина. — СПб.: Питер, 2010. — 304 с.
7. Семакин, И. Г. Информатика: Учебник для 10–11 классов / И. Г. Семакин, Е. К. Хеннер. — М.: БИНОМ. Лаборатория знаний, 2020. — 384 с.

Приложение А

Диаграмма последовательности действий

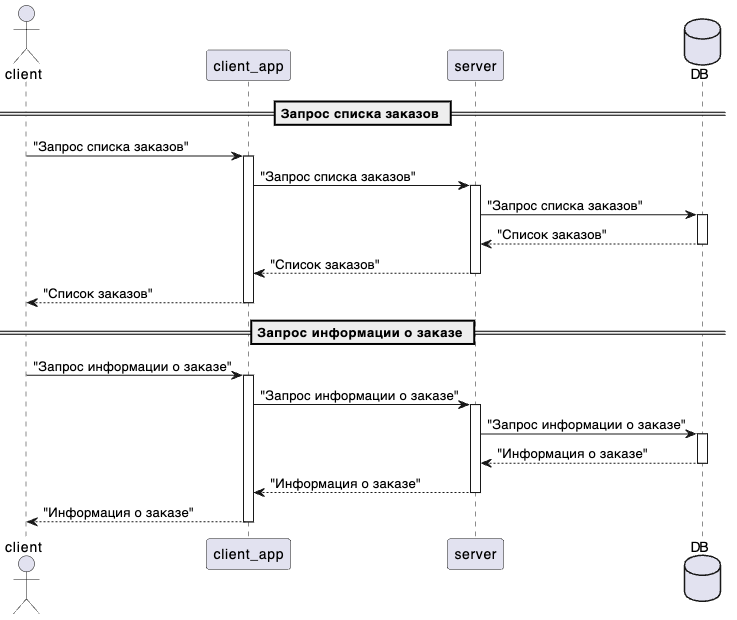


Рисунок 1 – Диаграмма последовательностей пользователя.

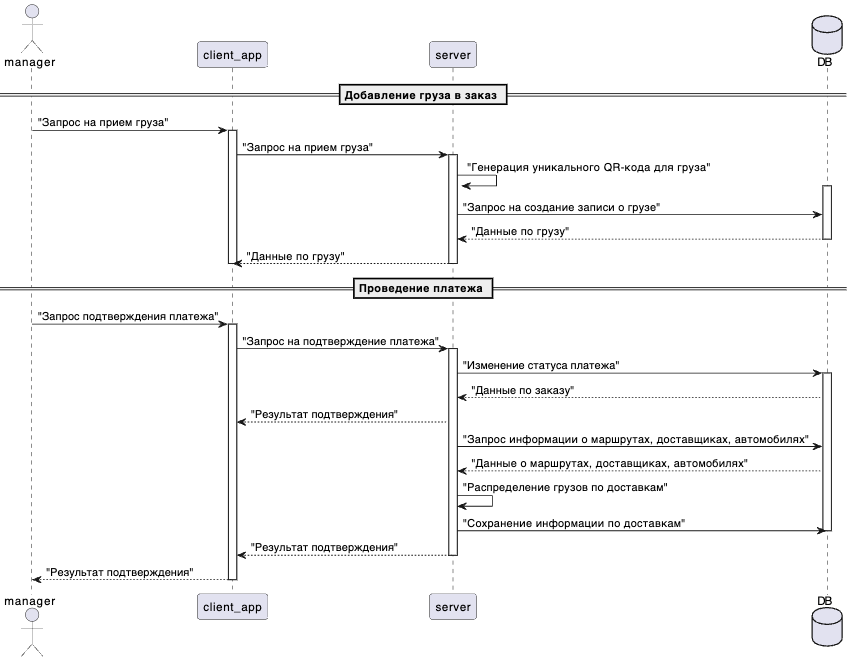


Рисунок 2 – Диаграмма посделовательностей для менеджера склада, часть 1.

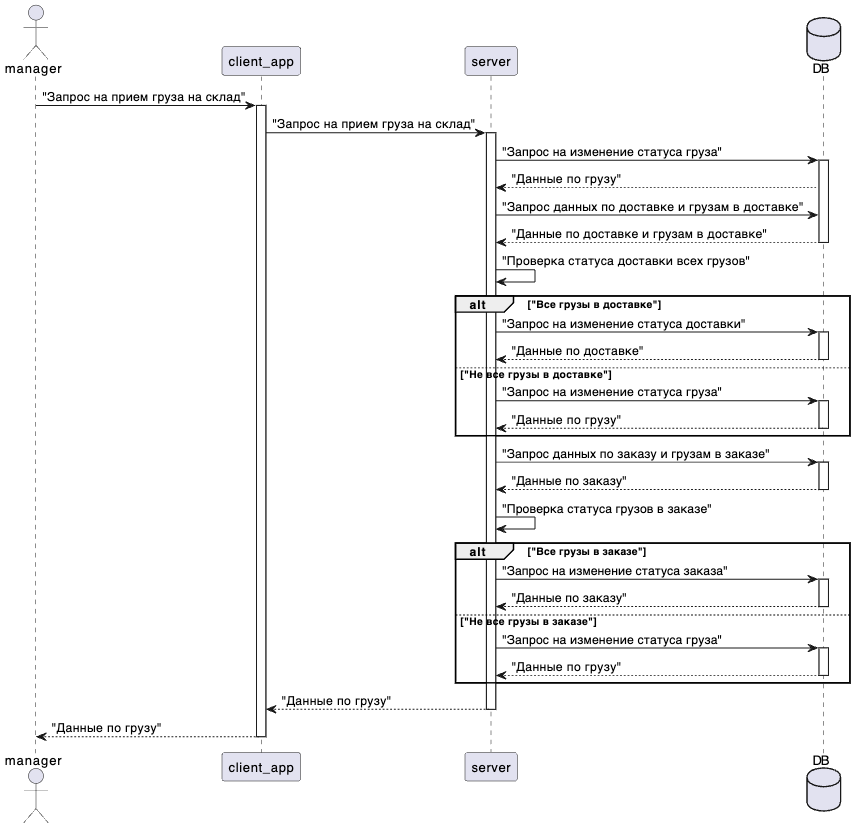


Рисунок 3 – Диаграмма последовательностей для менеджера склада, часть 2.

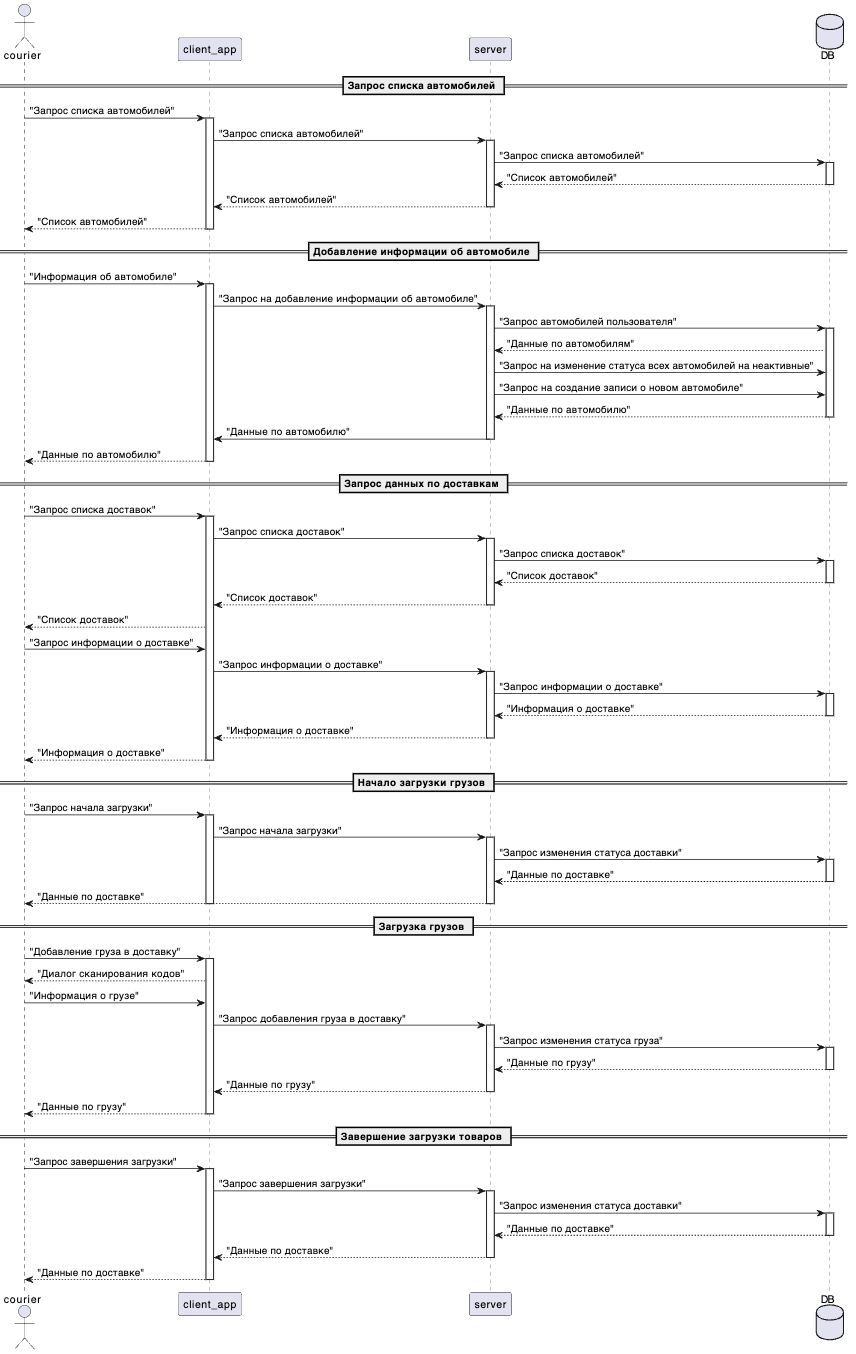


Рисунок 3 – Диаграмма последовательностей для курьера.

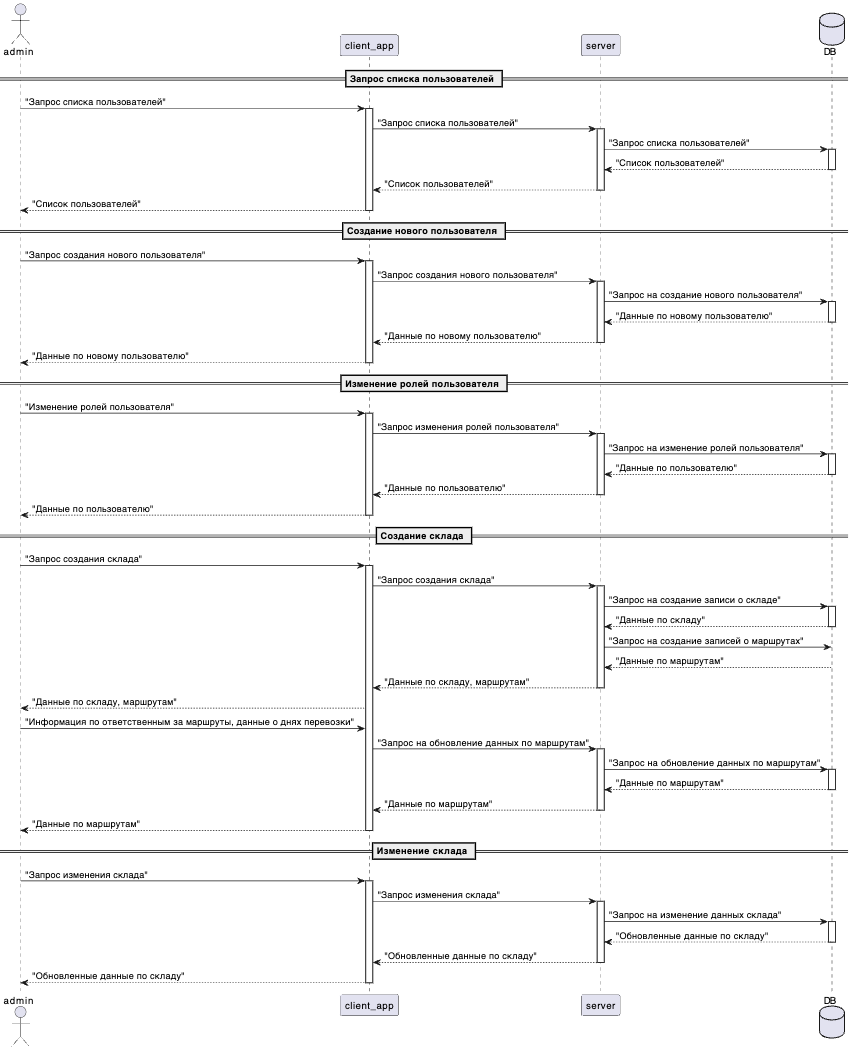


Рисунок 4 – Диаграмма последовательностей для администратора.

Приложение Б

Моделирование данных

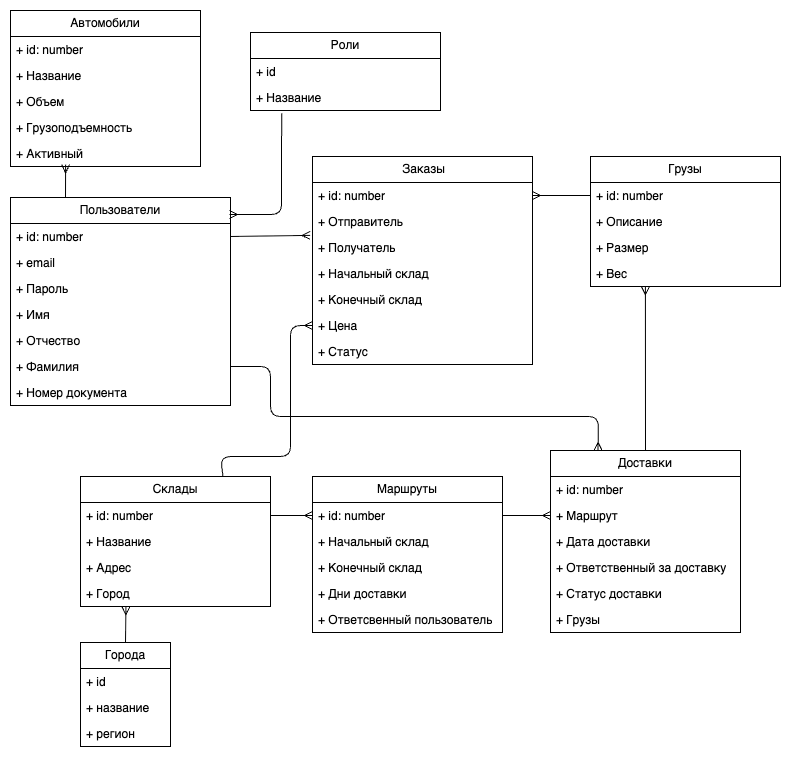


Рисунок 1 – Логическая модель данных

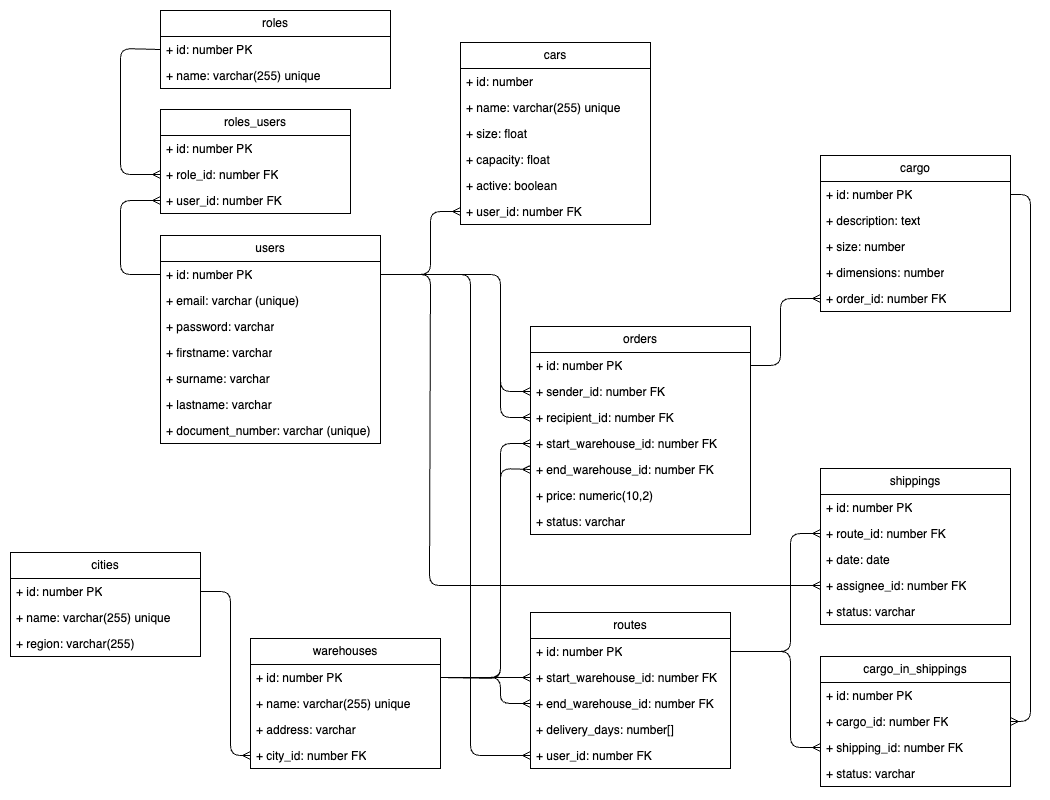


Рисунок 2 – Физическая модель данных.