

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(наименование профиля, специализации)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему: «Разработка программного обеспечения для мониторинга и управления грузоперевозками»

Обучающийся

В.К. Глазырин

(И.О. Фамилия)

(личная подпись)

Руководитель

к.п.н., доцент, Е.А. Ерофеева

(ученая степень, звание, И.О. Фамилия)

Тольятти 2024

Аннотация

Тема: Разработка программного обеспечения для мониторинга и управления грузоперевозками.

Ключевые слова: информационная система, грузоперевозки, программное обеспечение, управление поставками.

Целью выпускной квалификационной работы является разработка программного обеспечения для мониторинга и управления грузоперевозками для повышения эффективности деятельности строительной фирмы.

Предметом исследования является автоматизация деятельности по мониторингу и управлению грузоперевозками.

Объектом исследования является автоматизация процессов мониторинга и управления грузоперевозками.

В работе проведен анализ существующего процесса мониторинга и управления грузоперевозками, выполнено концептуальное моделирование, в ходе которого разработаны процессные и объектные модели, разработана логическая модель базы данных. Выполнено проектирование, разработка и тестирование программного обеспечения для мониторинга и управления грузоперевозками.

Выпускная квалификационная работа состоит из 45 страниц, 17 рисунков, 5 таблиц, 20 источников.

Содержание

Введение.....	4
1 Анализ объекта исследования.....	6
1.1 Управление грузоперевозками в рамках организационной структуры предприятия	6
1.2 Функциональная модель управления грузоперевозками.....	8
1.3 Анализ лучших практик автоматизации процессов управления и мониторинга грузоперевозок	12
2 Проектирование программного обеспечения управления и мониторинга грузоперевозок.....	17
2.1 Проектирование программного обеспечения.....	17
2.2 Архитектура и описание информационной модели проекта.....	20
3 Разработка программного обеспечения управления грузоперевозками	25
3.1 Выбор технологий реализации программного обеспечения	25
3.2 Реализация информационной модели программного обеспечения.....	28
3.3 Разработка программного обеспечения.....	30
3.4 Тестирование программного обеспечения	36
Заключение	42
Список используемой литературы и используемых источников.....	44

Введение

Эффективное управление грузоперевозками является критическим фактором успеха для современных строительных организаций. Своевременная доставка строительных материалов на объекты напрямую влияет на сроки выполнения проектов, общую стоимость и, в конечном итоге, на прибыльность компании. Однако, традиционные методы управления, зачастую основанные на ручном планировании и контроле, оказываются недостаточно эффективными, приводя к задержкам, избыточным затратам на топливо и транспорт, а также к сложностям в отслеживании местонахождения грузов.

Данная выпускная квалификационная работа посвящена разработке программного обеспечения для автоматизации управления и мониторинга грузоперевозок в строительной организации.

Актуальность темы обусловлена растущей потребностью строительных компаний в повышении эффективности логистических операций, снижении издержек и улучшении контроля над процессом доставки материалов. Использование информационных технологий позволяет автоматизировать рутинные задачи, такие как планирование маршрутов, отслеживание местонахождения грузов в режиме реального времени, управление транспортными средствами и водителями, а также анализ эффективности логистических процессов.

Предметом исследования является автоматизация управления грузоперевозками, объектом – сама деятельность по их осуществлению.

Цель работы – создание программного обеспечения для повышения эффективности работы предприятия за счет оптимизации логистики.

Для достижения этой цели были поставлены следующие задачи:

- анализ существующих процессов грузоперевозок;
- предложение мер по их улучшению;
- разработка концептуальной модели программного обеспечения;

- проектирование базы данных;
- разработка и тестирование самого приложения.

Работа состоит из введения, трех глав, заключения и списка литературы.

Первая глава описывает организационную структуру строительной компании и её бизнес-процессы, связанные с грузоперевозками, с использованием методологии IDEF0. Во второй главе представлено концептуальное моделирование информационной системы, включая диаграммы прецедентов и диаграммы активности. Третья глава посвящена архитектуре, разработке и тестированию программного обеспечения, а также созданию реляционной базы данных на основе разработанной концептуальной модели. Заключение содержит выводы и перспективы внедрения разработанного программного обеспечения.

1 Анализ объекта исследования

1.1 Управление грузоперевозками в рамках организационной структуры предприятия

Строительное предприятие специализируется на выполнении широкого спектра строительно-монтажных работ, охватывающих как гражданское, так и промышленное строительство. Деятельность компании включает в себя полный цикл работ, начиная от проектирования и подготовки проектной документации, закупки необходимых материалов и оборудования, до непосредственного выполнения строительных и монтажных работ и последующей сдачи объекта в эксплуатацию.

Ключевым направлением деятельности является возведение зданий и сооружений различного назначения: жилые комплексы, офисные здания, производственные цеха, складские помещения, объекты инфраструктуры. Компания использует современные технологии и материалы, обеспечивая высокое качество строительства и соблюдение всех необходимых стандартов и норм. В штате предприятия работают высококвалифицированные специалисты: инженеры-проектировщики, строители, монтажники, специалисты по контролю качества, а также управленческий персонал, обеспечивающий эффективное планирование и организацию всех процессов.

Неотъемлемой частью деятельности предприятия является управление закупками и логистикой. Обеспечение строительных площадок необходимыми материалами, оборудованием и инструментами осуществляется с помощью собственного автопарка и привлечения сторонних транспортных компаний. Компания осуществляет строгий контроль качества закупаемых материалов, обеспечивая соответствие их заявленным характеристикам. Логистика включает в себя планирование маршрутов доставки, оптимизацию транспортных расходов, отслеживание местонахождения грузов и контроль за своевременностью поставок. Это

является критическим фактором для поддержания ритмичности строительных работ и соблюдения сроков сдачи объектов.

Предприятие также уделяет значительное внимание вопросам безопасности труда. Все работы проводятся в строгом соответствии с действующими нормами и правилами техники безопасности. Компания регулярно проводит обучение персонала, обеспечивает рабочие места необходимыми средствами индивидуальной защиты и контролирует соблюдение всех требований безопасности на строительных площадках. Благодаря комплексному подходу к организации деятельности, компания добивается высокой эффективности работы и выполняет взятые на себя обязательства перед заказчиками. Ориентация на качество, соблюдение сроков и высокий профессионализм сотрудников являются основными принципами работы предприятия. В связи с масштабами проводимых работ, эффективная система управления грузоперевозками является критическим фактором для обеспечения бесперебойного функционирования всех процессов.

Организационная структура предприятия показана на рисунке 1.

Директор возглавляет предприятие и осуществляет общее руководство деятельностью.

В прямом подчинении директора находятся такие подразделения как служба снабжения, служба по строительству, возглавляемая заместителем директора по строительству, служба по развитию, возглавляемая заместителем директора по развитию и финансовая служба, возглавляемая финансовым директором.

Грузоперевозки выполняет служба снабжения. Заказчиком выступает служба по строительству.

Непосредственно грузоперевозки выполняются водителями. Формирование грузоперевозок выполняют диспетчеры. Отпуск грузов со склада выполняют кладовщики. Непосредственными заказчиками являются бригадиры, работающие на строительных объектах.

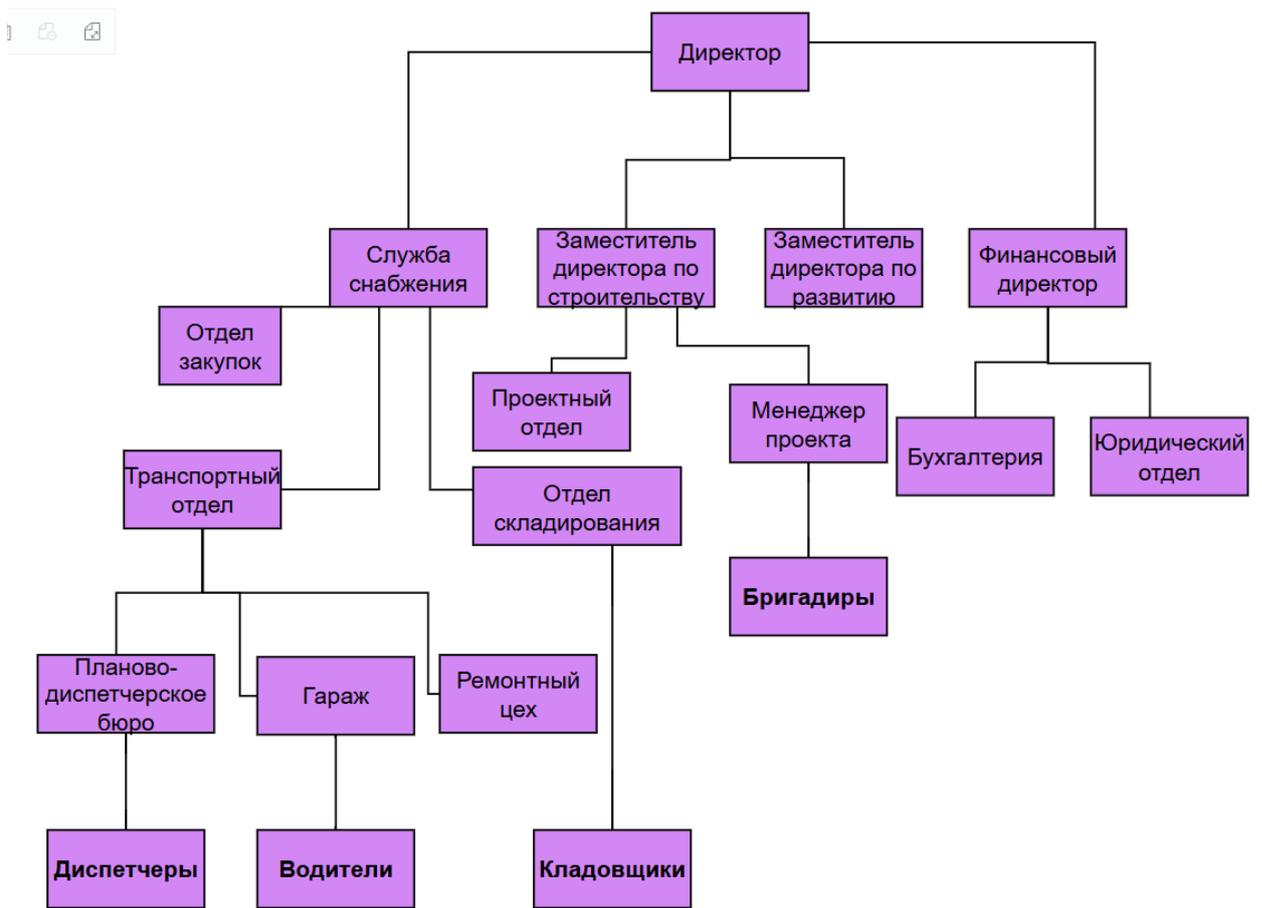


Рисунок 1 – Организационная структура предприятия

Непосредственные участники процессов грузоперевозок на организационной схеме выделены жирным шрифтом.

1.2 Функциональная модель управления грузоперевозками

Для функционального моделирования системы управления и мониторинга грузоперевозок была выбрана методология IDEF0, отличающаяся простотой и эффективностью на начальном этапе анализа [4, 8, 10]. Этот метод позволяет наглядно представить взаимодействие системы с внешней средой и внутренние процессы.

На первом уровне моделирования, представленном контекстной диаграммой (рисунок 2), вся деятельность обобщенно описана как единый

процесс «Управление и мониторинг грузоперевозок». Данное представление обеспечивает максимальную абстракцию, фокусируясь на ключевом аспекте – взаимодействии системы с внешним окружением.

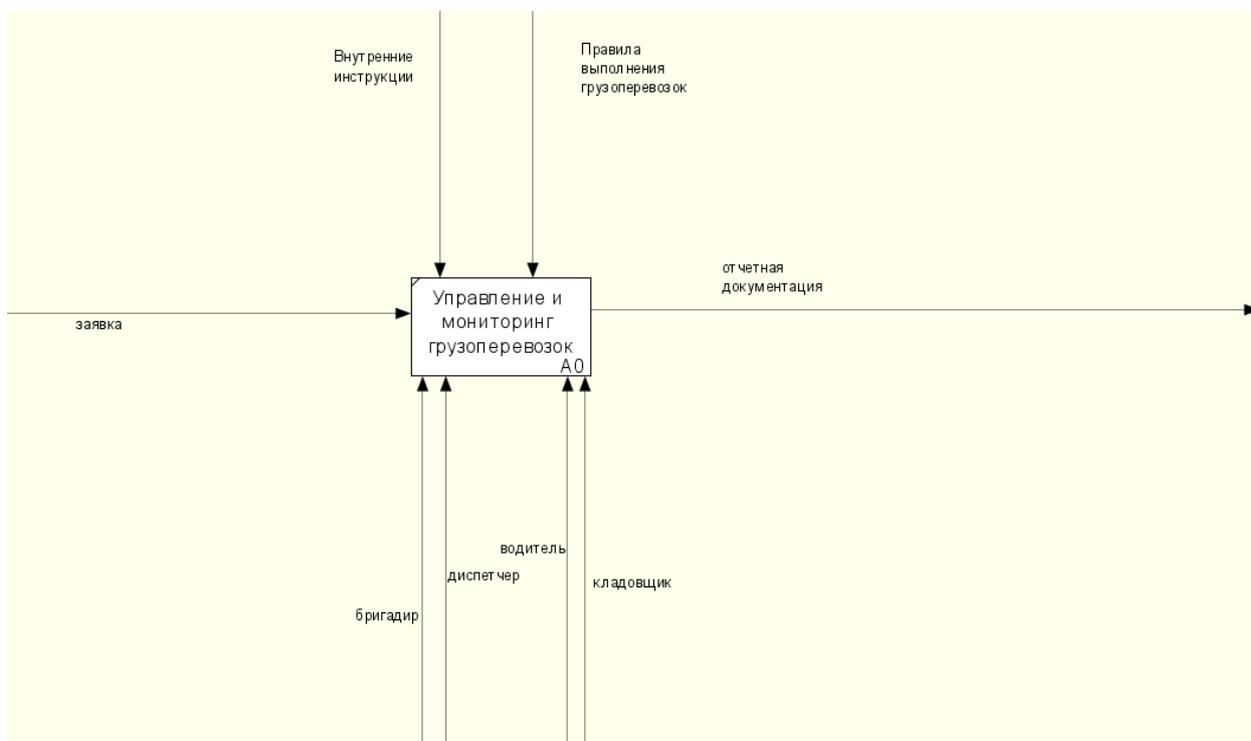


Рисунок 2 – Контекстная диаграмма

Входной информацией для этого процесса является заявка на доставку груза, содержащая все необходимые данные о заказе: тип груза, место отправления, место назначения, объем, сроки доставки и прочие релевантные параметры. Выходная информация представлена отчетной документацией, подводящей итоги выполненных грузоперевозок. Эта документация может включать сведения о выполненных рейсах, затраченном времени, расходах топлива, а также другие важные метрики, необходимые для анализа эффективности логистических операций и планирования будущих перевозок.

Сам процесс «Управление и мониторинг грузоперевозок» регулируется внутренними инструкциями и правилами компании, определяющими порядок выполнения работ, стандарты обслуживания и требования к отчетности. В его

реализации участвуют различные сотрудники: бригадиры, ответственные за организацию и контроль на местах, диспетчеры, координирующие транспортные средства и маршруты, водители, непосредственно осуществляющие доставку грузов, и кладовщики, ответственные за прием и выдачу грузов на складах. Взаимодействие этих сотрудников и упорядоченность их действий определяются внутренними регламентами и протоколами компании.

Контекстная диаграмма IDEF0, таким образом, предоставляет упрощенное, но достаточно информативное представление о системе управления и мониторинга грузоперевозок. Она демонстрирует ключевые входные и выходные данные, а также факторы, влияющие на функционирование системы, без углубления в детали внутренних процессов. Более подробное описание внутренней структуры и взаимодействия компонентов системы осуществляется на последующих уровнях декомпозиции в рамках методологии IDEF0. Это позволяет поэтапно и структурировано рассмотреть все аспекты работы системы.

Диаграмма детализации верхнего уровня (рисунок 3) содержит три процесса:

- формирование перевозки;
- выполнение перевозки;
- контроль и мониторинг грузоперевозки.

Процесс контроля и мониторинга грузоперевозок получает на входе информацию о грузах и выполненных грузоперевозках.

По результатам анализа выявлено, пополнение и управление грузоперевозками выполняется не системно, что затрудняет эффективное и быстрое выполнение заявок, оптимальное формирование перевозок, быстрое составление отчетности.

Решение описанной проблемы возможно путем разработки программного обеспечения для управления и мониторинга грузоперевозками.

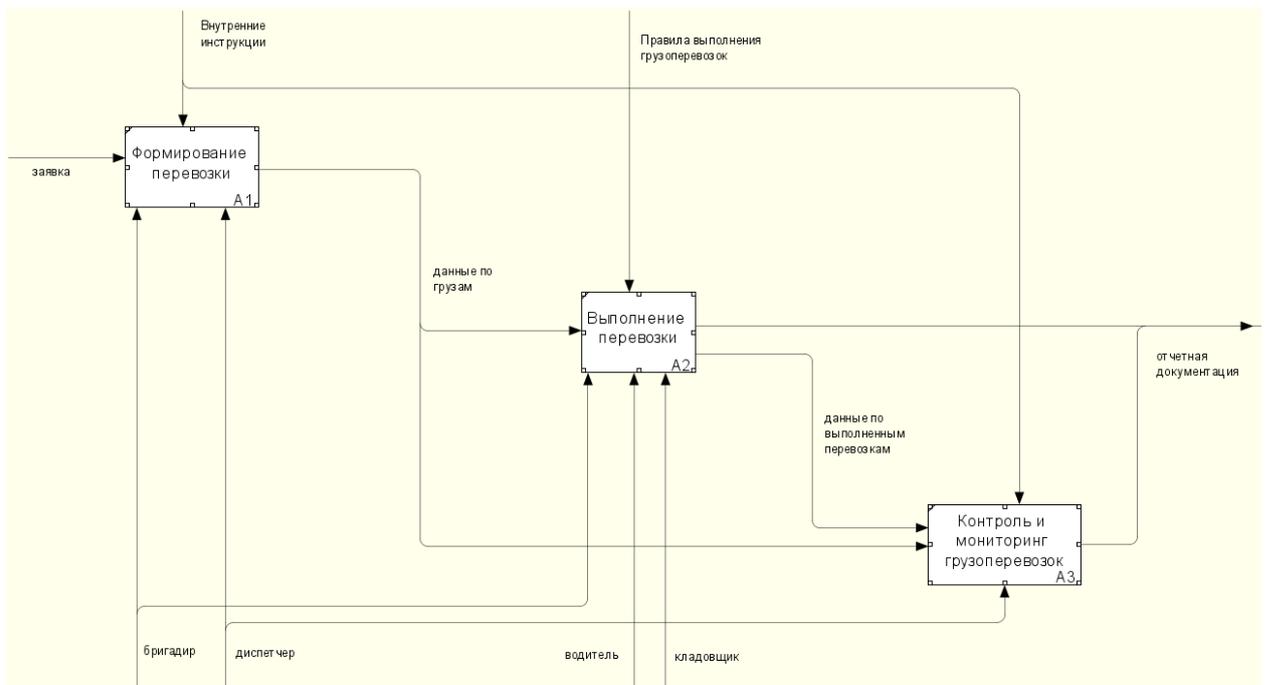


Рисунок 3 – Диаграмма верхнего уровня

Выполнена постановка задачи на разработку программного обеспечения.

Программное обеспечение для мониторинга и управления грузоперевозками должно обеспечивать:

- ввод заявки на грузоперевозку и мониторинг ее состояния;
- формирование грузоперевозок из заявок;
- учет отпуска и приема грузов в ходе грузоперевозки;
- мониторинг текущих грузоперевозок и вывод статистики по выполненным грузоперевозкам.

Разработаны функциональные и нефункциональные требования к программному обеспечению по управлению и мониторингу грузоперевозок.

Функциональные требования:

- ввод заявки на грузоперевозку и мониторинг ее состояния;
- формирование грузоперевозок из заявок;
- учет отпуска и приема грузов в ходе грузоперевозки;
- мониторинг текущих грузоперевозок;

– вывод статистики по выполненным грузоперевозкам.

Требования к удобству использования (UX): требуется графический интерфейс для пользователей.

Требования к надежности: требуется простая база данных.

Требования к производительности: не предъявляется.

Требования к поддержке: требуется программное обеспечение, не требующее специальных навыков от пользователей.

1.3 Анализ лучших практик автоматизации процессов управления и мониторинга грузоперевозок

Управление и мониторинг грузоперевозок является распространенной задачей автоматизации и имеется большое количество готовых программных продуктов в данной области. Данный класс программного обеспечения – системы управления логистикой.

«Logistics Management Systems, (LMS) позволяют управлять транспортировкой и перевозками, флотом (или парком) средств перевозок, цепочками поставок, а также планировать маршруты поставок и решать специализированные задачи логистики. Программное обеспечение для логистики позволяет предприятиям управлять ресурсами и повышать эффективность затрат на материально-техническое обеспечение и транспортное обеспечение в целом.

Программный продукт LMS должен соответствовать критериям:

- Возможность точного планирования и управления поставками товаров и услуг в определенное время и место.
- Высокая степень автоматизации и интеграции с внешними и внутренними информационными системами и базами данных.
- Оптимизация загрузки транспортных средств и маршрутизации для эффективного использования ресурсов и снижения затрат.

- Обеспечение контроля за всеми этапами процесса и мониторингом выполнения задач.
- Гибкость и адаптивность системы для адекватной реакции на изменения внешней и внутренней среды.
- Аналитическая возможность для прогнозирования рыночных тенденций, определения проблем и выявления возможностей для улучшения эффективности логистических операций.
- Обеспечение высокого уровня обслуживания клиентов через точность и своевременность поставок.»[19]

«Возможность администрирования позволяет осуществлять настройку и управление функциональностью системы, а также управление учётными записями и правами доступа к системе.

Функции Бюджетирования позволяют планировать и отслеживать финансовые бюджеты и планы для управления финансовыми средствами в различных департаментах или для контроля со стороны учредителей, акционеров и инвесторов

Возможность импорта и/или экспорта данных в продукте позволяет загрузить данные из наиболее популярных файловых форматов или выгрузить рабочие данные в файл для дальнейшего использования в другом ПО.

Возможность многопользовательской доступа в программную систему обеспечивает одновременную работу нескольких пользователей на одной базе данных под собственными учётными записями. Пользователи в этом случае могут иметь отличающиеся права доступа к данным и функциям программного обеспечения.»[19]

«Часто при использовании современного делового программного обеспечения возникает потребность автоматической передачи данных из одного ПО в другое. Например, может быть полезно автоматически передавать данные из Системы управления взаимоотношениями с клиентами (CRM) в Систему бухгалтерского учёта (БУ). Для обеспечения такого и подобных сопряжений программные системы оснащаются специальными

Прикладными программными интерфейсами (англ. API, Application Programming Interface). С помощью таких API любые компетентные программисты смогут связать два программных продукта между собой для автоматического обмена информацией.

Функции Оптимизации маршрутов реализуются часто с применением сложных алгоритмов, используются для определения наиболее оптимального и экономически эффективного транспортно-логистического маршрута. Часто используется алгоритм построения гипотез и альтернатив, чтобы найти оптимальный маршрут

Наличие у продукта функций подготовки отчётности и/или аналитики позволяют получать систематизированные и визуализированные данные из системы для последующего анализа и принятия решений на основе данных.

Функции Планирования и прогнозирования спроса позволяют планировать спрос путём либо ручного заполнения значений спроса на услуги поставки, либо путём прогнозирования, часто по методологии временных рядов с выявлением сезонности и тенденций спроса»[19]

«Функции Управления заказами позволяют осуществлять работу и контроль выполнения продаж и заказов на товары и услуги

Функции Управления запасами позволяют выполнять высокоуровневое планирование и контроль запасов, а также оперативное управление хранением, размещением, инвентаризацией или распределением запасов и единиц хранения

Функции Управления перевозками позволяют автоматизировать контроль длительных перевозок для производителей и поставщиков товаров

Функции Управления поставщиками позволяют упростить работу с закупками, необходимыми для повседневной работы организации, у управляемого пула поставщиков

Функции Управления проектами позволяют контролировать операции от идеи и планирования до этапов поставки и постпроектной отчётности,

организовывать ресурсы, планировать бюджеты и повышать эффективность в достижении операционных целей»[19]

«Программа АвтоПлан [1] обеспечивает следующий функционал:

- журнал механика по выпуску ТС (фиксация выезда и заезда транспорта, замечания к его состоянию);
- разрядка (создание планов выпуска и печать путевых листов);
- обработка путевых листов (контроль использования ТС и его расходов, формирование отчетов, печать актов);
- табель (учет и контроль рабочего времени, расчет зарплаты, формирование отчетов);
- склад (формирование прихода-расхода ТМЦ, их распределение, контроль расхода);
- журнал ремонта (контроль ремонта, учет шин и АКБ, приобретение запчастей);
- управление (сбор информации, формирование отчетов).

Достоинствами программы являются бесплатное техническое внедрение и хороший функционал для контроля за транспортом. К недостаткам программы можно отнести слабый функционал для логистики и устаревший интерфейс.»[13]

«Программа TransTrade [20] обеспечивает следующий функционал:

- работа с заказами;
- база контрагентов и ТС;
- формирование и печать транспортных документов (договоры, путевые листы, накладные, доверенности);
- мини CRM (учет обращений);
- планирование маршрутов;
- отчеты по задолженности, ДДС, расходам и рентабельности, обороту.

Достоинствами программы являются наличие демо-версии и возможность интеграции. К недостаткам программы можно отнести стоимость, слабый базовый функционал, устаревший интерфейс.»[13]

«Программа LogistPro [7] обеспечивает следующий функционал:

- работа с заявками (сводная информация, база данных с заказчиками);
- заказы (отслеживание статусов, расчет тарифов и стоимости);
- водители (создание заказов, назначение автомобилей, расчет зарплаты, анализ эффективности);
- автомобили (отслеживание по GPS/ГЛОНАСС, анализ прибыли).

Достоинствами программы являются наличие демо-версии и пробного периода. К недостаткам программы можно отнести стоимость, слабый функционал для логистики.»[13]

Анализ существующих решений показал, что применительно к решаемой задаче целесообразно выполнить разработку собственного программного обеспечения.

Выводы по разделу. В первом разделе описана деятельность организации, показана ее организационная структура. Разработана процессная модель, описывающая основные бизнес-процессы, связанные с грузоперевозками. Деятельность по управлению и мониторингу грузоперевозок описана с использованием методологии IDEF0. Выполнен анализ и постановка задачи на разработку программного обеспечения, разработка требований и обзор аналогов.

2 Проектирование программного обеспечения управления и мониторинга грузоперевозок

2.1 Проектирование программного обеспечения

Для четкого определения пользователей и их взаимодействия с разрабатываемой системой управления и мониторинга грузоперевозок была построена диаграмма вариантов использования (рисунок 4). Эта диаграмма, также известная как диаграмма прецедентов, является фундаментальным инструментом объектно-ориентированного моделирования на языке UML, позволяющим наглядно представить взаимодействие системы с внешним миром [3, 5, 6, 9, 12].

Диаграмма прецедентов служит первой ступенью в процессе моделирования, задачей которой является определение внешних субъектов (акторов), взаимодействующих с системой, и описание того, как эти акторы используют систему для достижения своих целей. В данном контексте, акторы представляют собой пользователей системы, каждый из которых имеет определенные роли и права доступа.

Каждый актер взаимодействует с системой, инициируя определенные сценарии использования (прецеденты). Прецедент описывает последовательность действий, выполняемых системой в ответ на запрос актора.

Диаграмма вариантов использования визуализирует эти акторы и прецеденты, показывая взаимосвязи между ними. Это позволяет четко определить функциональные требования к системе и обеспечить ее соответствие потребностям пользователей. Диаграмма является важным этапом проектирования, закладывая основу для последующих этапов разработки программного обеспечения. Она обеспечивает понимание того, как будет взаимодействовать система с пользователями, что позволяет создать интуитивно понятный и эффективный инструмент.



Рисунок 4 – Диаграмма вариантов использования

Акторами или внешними сущностями являются:

- диспетчер;
- водитель;
- кладовщик;
- бригадир.

Управление грузоперевозками выполняется в следующем порядке. Бригадир создает заявку на доставку груза – заявку на грузоперевозку. Диспетчер, ориентируясь на имеющиеся заявки, формирует грузоперевозку – назначая один или несколько заказов определенному автомобилю и водителю. Водитель на основании своей грузоперевозки получает грузы на складе или на нескольких складах и выполняет доставку грузов до получателя (бригадира). Отпуская грузы со склада, кладовщик фиксирует факт отпуска. Получив груз, бригадир фиксирует выполнение доставки груза. На основании выполненных доставок грузов диспетчер фиксирует выполнение всей грузоперевозки.

В части мониторинга диспетчер может просмотреть текущие грузоперевозки с информацией какой груз получен, какой доставлен, а также

сформировать отчет по выполненным грузоперевозкам. Бригадир может видеть состояние своих заказов – назначена ли перевозка, отпущен ли груз со склада.

Каждый пользователь имеет доступ к действиям в системе согласно своей роли. Так бригадир может создавать заявки, мониторить состояние своих заявок и закрывать заявки. Диспетчер имеет возможность просмотра поступивших заявок и формирования из них грузоперевозки с назначением машины и водителя. Водитель имеет возможность просмотра закрепленных за ним текущих (не выполненных) грузоперевозок с перечнем грузов и адресов доставки. Кладовщик имеет возможность указать количество отпущенного по определённой заявке груза. При этом фиксируется дата и время отпуска и отпустивший груз кладовщик.

Прецедент «Формирование грузоперевозки» детально описан с использованием диаграммы деятельности (рисунок 5).

Открыв не назначенные к перевозке заявки на доставку, диспетчер определяет наличие свободной машины и водителя. Если свободных нет – формирование грузоперевозки откладывается до появления свободной машины. В случае если свободная машина есть диспетчер добавляет заказы в перевозку на основании суммарной массы и объёма грузов в заказах и грузоподъемности и объёма машины.

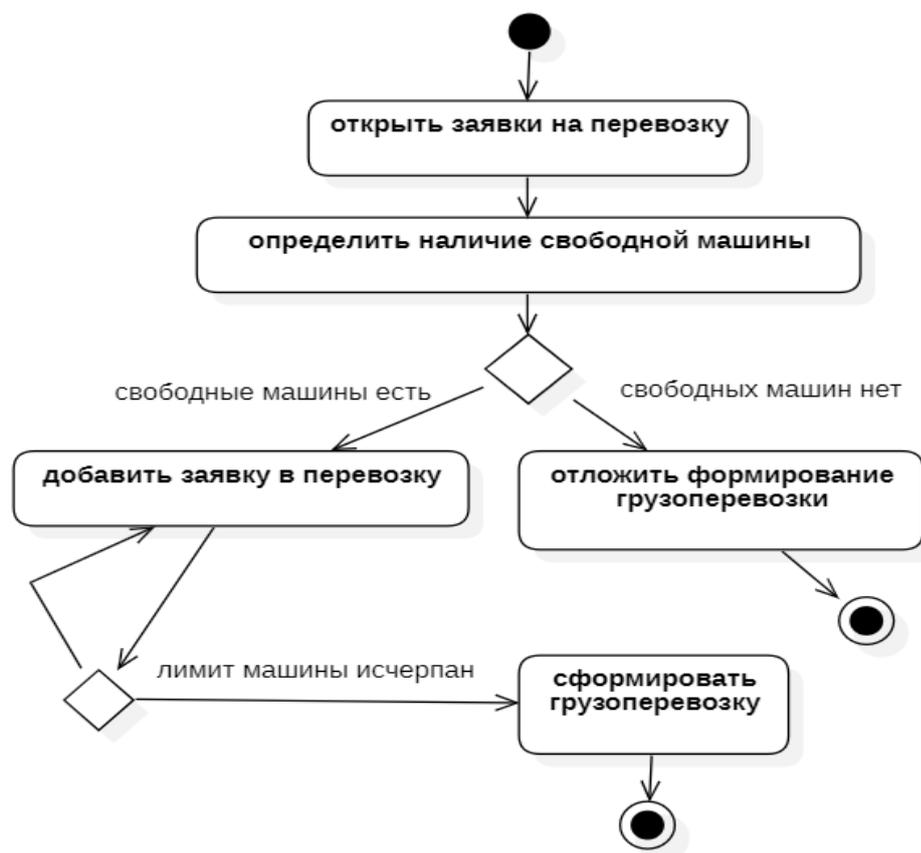


Рисунок 5 – Диаграмма деятельности прецедента «Формирование грузоперевозки»

Когда лимит грузоподъемности или объёма машины исчерпан или закончились заявки на доставку выполняется формирование грузоперевозки. Данные по грузам становятся доступны водителю, который назначен на грузоперевозку.

2.2 Архитектура и описание информационной модели проекта

«Клиент-серверная архитектура – это модель организации вычислительных систем, в которой задачи распределены между клиентами и серверами. В такой архитектуре клиент, обычно являющийся пользователем или программой, запрашивает услуги или ресурсы у сервера, который отвечает на запросы, предоставляя необходимые данные или функциональность.»[11]

«Клиент-серверная архитектура является распространенной моделью для построения сетевых приложений. Она состоит из двух основных компонентов: клиента и сервера, которые взаимодействуют друг с другом посредством сетевого соединения.

Основные особенности клиент-серверной архитектуры:

Разделение функций: в клиент-серверной модели клиент выполняет запросы к серверу, а сервер осуществляет обработку этих запросов и предоставляет клиенту необходимые ресурсы или услуги. Разделение функций позволяет распределить нагрузку между клиентом и сервером, улучшить масштабируемость и обеспечить более эффективную обработку запросов.

Сервер как «черный ящик»: клиент не знает, каким образом сервер выполняет его запросы и какие конкретно ресурсы используются. Для клиента сервер выглядит как единая сущность, с которой он взаимодействует, без необходимости знания о его внутренней работе.

Надежность: клиент-серверная архитектура позволяет повысить надежность системы за счет распределения нагрузки между серверами. Например, в случае отказа одного сервера, клиенты могут переключиться на другой без прерывания обслуживания.»[11]

«Масштабируемость: клиент-серверная архитектура позволяет добавлять новых клиентов и сервера, что обеспечивает горизонтальную и вертикальную масштабируемость. Это позволяет системе эффективно обрабатывать растущую нагрузку и адаптироваться к изменениям в требованиях пользователей.

Централизованное управление: сервер выполняет управление и контроль за ресурсами, данные и услуги которых предоставляются клиентам. Это упрощает управление системой и обеспечивает централизованные политики безопасности и доступа к данным.

Клиент-серверная архитектура позволяет взаимодействовать с различными платформами и технологиями, используя открытые стандарты для обмена данными и коммуникаций» [11]

На основании изложенного, для проекта выбрана клиент-серверная архитектура

Из описания и моделирования предметной области для информационной модели можно выделить три основных сущности – доставка, заказ и груз.

Между сущностями определены следующие связи – заказ состоит из грузов, перевозка состоит из заказов.

Сущность «перевозка» имеет атрибуты:

- идентификатор перевозки;
- машина;
- водитель;
- дата и время формирования перевозки;
- дата и время завершения перевозки;
- примечания к перевозке.

Сущность «заказ» имеет атрибуты:

- идентификатор заказа;
- идентификатор перевозки;
- заказчик;
- адрес доставки;
- дата и время создания заказа;
- дата и время отгрузки заказа со склада;
- кладовщик;
- дата и время получения заказа;
- примечания к заказу.

Сущность «груз» имеет атрибуты:

- идентификатор груза;
- идентификатор заказа;

- наименование груза;
- обозначение или маркировка;
- объём;
- масса;
- количество;
- единица измерения;
- примечания к грузу;

Атрибуты сущностей показаны на концептуальной модели данных (рисунок 6).

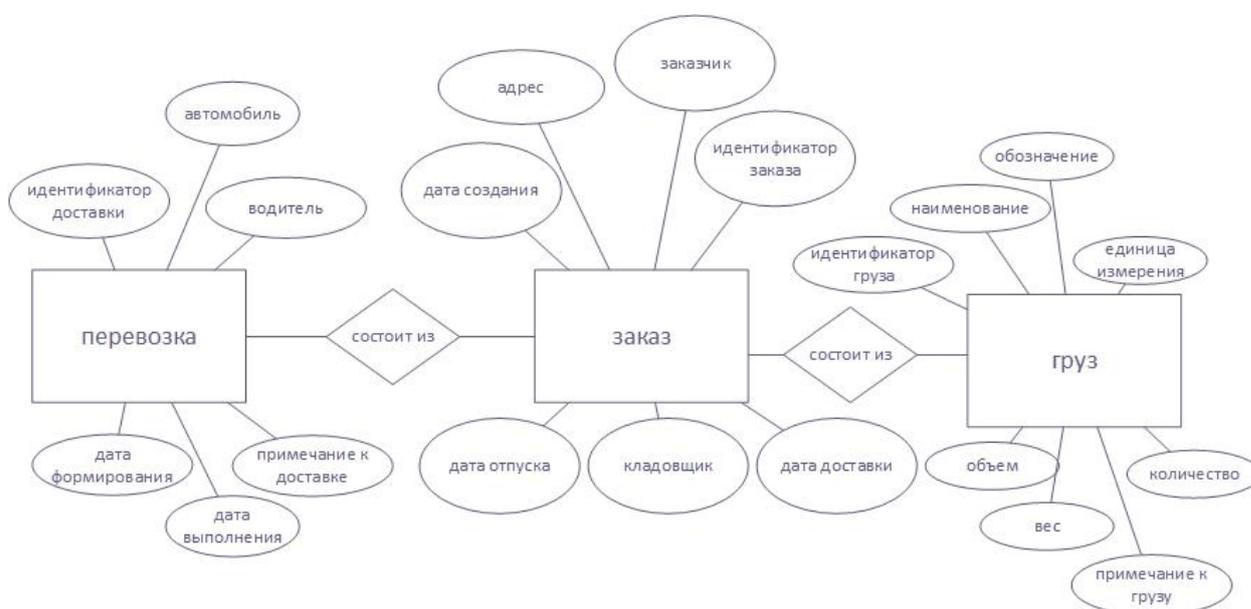


Рисунок 6 – Концептуальная модель данных

На основании концептуальной модели разработана логическая модель базы данных [14, 15, 16] (рисунок 7), состоящая из трех таблиц. Модель выполнена в нотации IDEF1X, выделены ключевые поля и типы данных для каждого поля. Установлена кардинальность отношений и определены внешние ключи в дочерних таблицах.

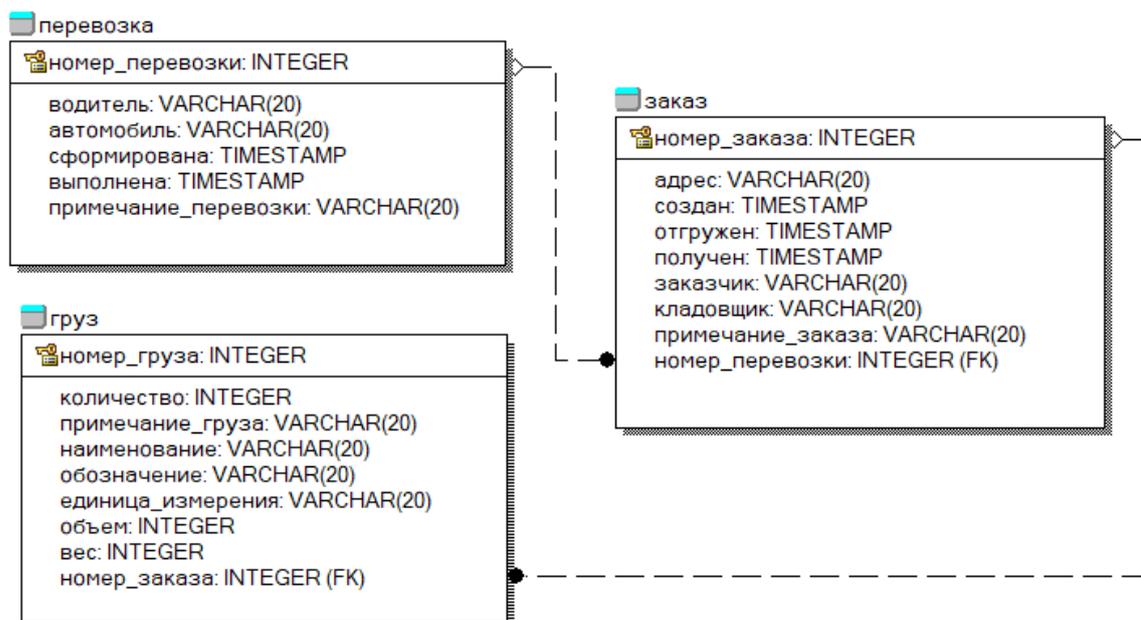


Рисунок 7 – Логическая модель данных

Выводы по разделу.

Во втором разделе разработана диаграмма вариантов использования, на которой выделены основные классы пользователей и представлены варианты использования программного обеспечения управления грузоперевозками.

Детализация вариантов использования use case диаграммы выполнена с помощью диаграммы деятельности, на которой показан алгоритм формирования грузоперевозки диспетчером.

Выполнен выбор архитектуры и моделирование данных, в ходе которого разработаны концептуальная и логическая модели данных программного обеспечения управления и мониторинга грузоперевозок.

3 Разработка программного обеспечения управления грузоперевозками

3.1 Выбор технологий реализации программного обеспечения

В качестве языка программирования выбран Python.

Выбор Python в качестве языка программирования для реализации программного обеспечения для управления и мониторинга грузоперевозок обоснован следующими соображениями.

Python отличается своей простотой и читаемостью, что делает его идеальным языком для разработки сложных систем. Простой синтаксис позволяет быстро освоить язык и сконцентрироваться на решении бизнес-задач, а не на тонкостях синтаксиса.

Python обладает обширной экосистемой библиотек и фреймворков, которые значительно упрощают разработку. Библиотеки для работы с базами данных (SQLAlchemy) и фреймворки для веб приложений (Django) значительно сокращают время разработки и повышают качество кода.

Python – это интерпретируемый язык, что позволяет быстро тестировать и отлаживать код. Отсутствие необходимости компиляции ускоряет цикл разработки и упрощает внесение изменений в проект.

Python активно используется в сфере машинного обучения и искусственного интеллекта, что позволяет в дальнейшем развивать систему в плане прогнозирования маршрутов, оптимизации логистических процессов, анализа больших данных и другие функции, которые повысят эффективность грузоперевозок.

Python является кроссплатформенным языком, что позволяет создавать системы, работающие на разных операционных системах. Это обеспечивает гибкость и масштабируемость проекта, позволяя интегрировать систему с различными устройствами и платформами.

В качестве фреймворка выбран – Django

Выбор Django в качестве фреймворка для разработки программного обеспечения для управления и мониторинга грузоперевозок обусловлен рядом преимуществ.

Django – это высокоуровневый фреймворк, который предоставляет готовые решения для многих задач, таких как аутентификация пользователей, обработка форм, работа с базами данных и многое другое. Это значительно сокращает время разработки и позволяет сосредоточиться на решении бизнес-задач.

Django обладает мощной системой администрирования, которая позволяет управлять данными, создавать пользователей и выполнять другие задачи без написания дополнительного кода. Это упрощает процесс управления системой и делает ее более доступной для нетехнических пользователей.

Django – это фреймворк, основанный на архитектуре MVC (Model-View-Controller), которая способствует структуризации кода и повышает его читаемость. Это делает систему более стабильной, упрощает ее поддержку и модификацию.

Django активно развивается и поддерживается большим сообществом разработчиков. Это гарантирует доступность документации, помощи и решений для возникающих проблем.

Django – это фреймворк, который применим как для создания как больших, так и для небольших проектов. Он обладает высокой гибкостью и масштабируемостью, позволяя адаптировать систему к изменяющимся потребностям и расти вместе с бизнесом.

В качестве IDE выбрано Visual Studio Code

Выбор Visual Studio Code (VSCode) в качестве IDE для разработки программного обеспечения для управления и мониторинга грузоперевозок обусловлен рядом преимуществ:

VSCode – это бесплатная и кроссплатформенная IDE, что делает ее доступной для всех разработчиков и позволяет работать на разных операционных системах.

VSCode обладает богатым набором функций, необходимых для эффективной разработки, таких как автоматическая доработка кода, интеллектуальный анализ кода, отладка, тестирование и многое другое.

VSCode предлагает широкий спектр расширений, которые расширяют функциональность IDE и делают ее более удобной для разработки на Python и Django. Существуют расширения для автоматического форматирования кода, интеграции с системами версионного контроля, отладки, автоматического импорта библиотек и многих других задач.

VSCode обладает высокой производительностью, что позволяет работать с большими проектами и обеспечивает плавную работу. Также отмечается низкое потребление ресурсов, что делает ее доступной для использования на различных компьютерах.

VSCode имеет интуитивно понятный интерфейс, что делает ее доступной как для опытных, так и для новичков в программировании.

В качестве СУБД выбрана SQLite.

Выбор SQLite в качестве СУБД для программного обеспечения для управления и мониторинга грузоперевозок является оправданным решением по следующим причинам:

SQLite – это встраиваемая база данных, что означает, что она не требует отдельного сервера и может быть интегрирована непосредственно в приложение. Это упрощает развертывание и настройку системы, особенно если проект предполагает работу на нескольких компьютерах или мобильных устройствах.

SQLite – это легкая и быстрая база данных, которая отлично подходит для небольших и средних проектов. Она обеспечивает высокую производительность даже при обработке больших объемов данных, что важно для эффективного отслеживания грузоперевозок.

SQLite – это бесплатная и открытая база данных, что делает ее доступной для всех разработчиков и позволяет свободно использовать ее в коммерческих проектах без дополнительных лицензионных платежей.

SQLite – это простая в использовании база данных, которая не требует специальных знаний в области администрирования баз данных. Она поддерживает стандартный язык SQL, что делает ее доступной для разработчиков с различным уровнем опыта.

SQLite – это надежная и стабильная система с высокой степенью защиты данных. Она используется во многих популярных приложениях, что подтверждает ее качество и надежность.

Согласно обзору [2] SQLite занимает третье место по популярности среди СУБД после MySQL и PostgreSQL. Таким образом, SQLite является оптимальным выбором СУБД для разработки программного обеспечения для управления и мониторинга грузоперевозок, особенно для проектов с ограниченными ресурсами и требованиями к масштабированию.

3.2 Реализация информационной модели программного обеспечения

Переход от логической к физической модели связан с выбором конкретной системы управления базой данных. В данном случае в качестве СУБД выбран SQLite [17, 18].

Поля таблиц физической модели представлены в таблицах 1-3.

Таблица 1 - Атрибуты сущности cargo

Имя	Тип	Размер	Обязательность	Описание
cargo_id	AutoField	–	обязательное	идентификатор груза, первичный ключ
order_id	ForeignKey	–	обязательное	идентификатор заказа, внешний ключ, каскадное удаление
designation	CharField	255	обязательное	наименование груза
marking	FloatField	–	не обязательное	обозначение или маркировка

Продолжение таблицы 1

Имя	Тип	Размер	Обязательность	Описание
volume	FloatField	–	не обязательное	объём м3
weight	FloatField	–	не обязательное	масса кг
quantity	IntegerField	–	не обязательное	количество
measurement_unit	FloatField	25	не обязательное	единица измерения
cargo notes	CharField	255	не обязательное	примечания к грузу

Таблица 2 - Атрибуты сущности order

Имя	Тип	Размер	Обязательность	Описание
order_id	AutoField	–	обязательное	идентификатор заказа, первичный ключ
trucking_id	ForeignKey	–	обязательное	идентификатор перевозки, внешний ключ, каскадное удаление
customer	CharField	255	обязательное	заказчик
address	FloatField	–	не обязательное	адрес доставки
created	DateTime	–	не обязательное	дата и время создания заказа
shipped	DateTime	–	не обязательное	дата и время отгрузки заказа со склада
storekeeper	CharField	255	не обязательное	кладовщик
delivered	DateTime	–	не обязательное	дата и время получения заказа
order notes	CharField	255	не обязательное	примечания к заказу

Таблица 3 - Атрибуты сущности trucking

Имя	Тип	Размер	Обязательность	Описание
trucking_id	AutoField	–	обязательное	идентификатор перевозки, первичный ключ
car	CharField	255	не обязательное	машина
driver	FloatField	–	не обязательное	водитель
formed	DateTime	–	не обязательное	дата и время формирования перевозки

Продолжение таблицы 3

Имя	Тип	Размер	Обязательность	Описание
Completed	DateTime	–	не обязательное	дата и время завершения перевозки
trucking_notes	CharField	255	не обязательное	примечания к перевозке

Программная реализация физической модели в рамках фреймворка Django выполнена в файле `models.py`.

Разработанные модели данных позволяют эффективно организовать хранение и обновление информации для управления и мониторинга грузоперевозок.

3.3 Разработка программного обеспечения

В рамках выбранного фреймворка Django веб-приложение на состоит из следующих компонентов:

- `django framework`;
- база данных;
- шаблоны;
- веб-браузер.

В состав фреймворка входят файлы `views.py`, `URLs.py`, `settings.py`, `models.py`, `WSGI.py`.

Файл `views.py` определяют логику приложения, которая обрабатывает запросы от браузера, получает данные из базы данных, формирует HTML-код и отправляет его обратно в браузер.

Файл `URLs.py` сопоставляет URL-адреса с соответствующими представлениями (`views`).

Файл `settings.py` хранит конфигурацию приложения, включая настройки базы данных, таймзону, языки, пути к файлам и другие важные параметры.

Файл `models.py` представляет структуру данных приложения, определяя таблицы в базе данных и взаимосвязи между ними.

Файл `WSGI.py` (Web Server Gateway Interface) – это интерфейс, позволяющий веб-серверу взаимодействовать с приложением Django.

База данных SQLite3 это встроенная в Python база данных, которая хорошо подходит для небольших проектов или прототипов.

Шаблоны (Templates) представляют собой HTML-файлы. Они содержат статический контент (текст, изображения, CSS), который отображается в браузере, а также используют специальные теги Django для динамического отображения данных из моделей и представлений.

Веб-браузер – это клиентская часть, которая отправляет запросы на сервер, отображает HTML-код и взаимодействует с пользователем.

Веб-приложение на Django работает по следующей схеме:

Пользователь вводит URL-адрес в браузер.

Браузер отправляет запрос на сервер.

Сервер Django обрабатывает запрос, находит соответствующий URL-адрес в конфигурации приложения и запускает соответствующее представление (view).

Представление (view) взаимодействует с моделями (models), чтобы получить необходимые данные из базы данных.

Представление (view) рендерит HTML-шаблон с использованием полученных данных.

Браузер получает от сервера готовый HTML-код и отображает его пользователю.

Диаграмма компонентов представлена на рисунке 8.

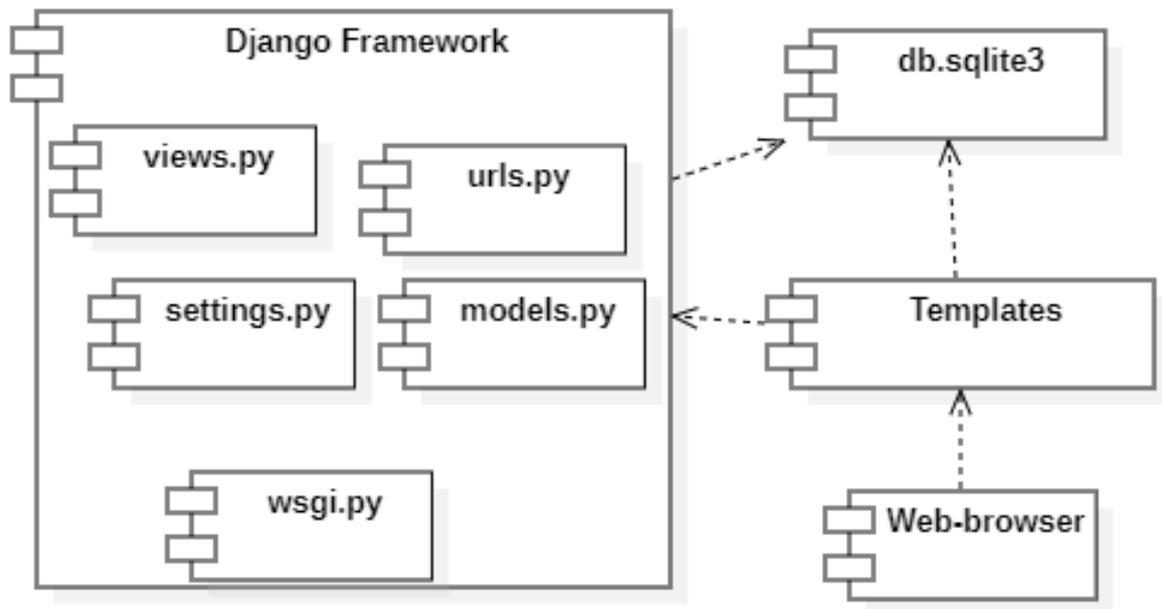


Рисунок 8 - Диаграмма компонентов приложения

Диаграмма развертывания показана на рисунке 9. Она содержит узлы (Nodes) – веб-сервер, сервер приложений и базу данных.

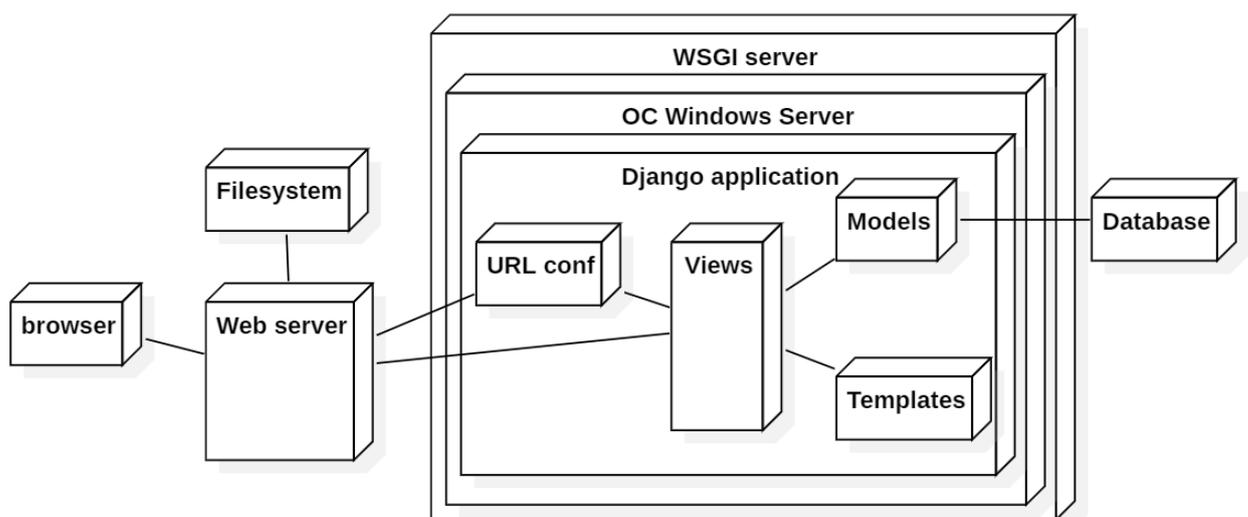


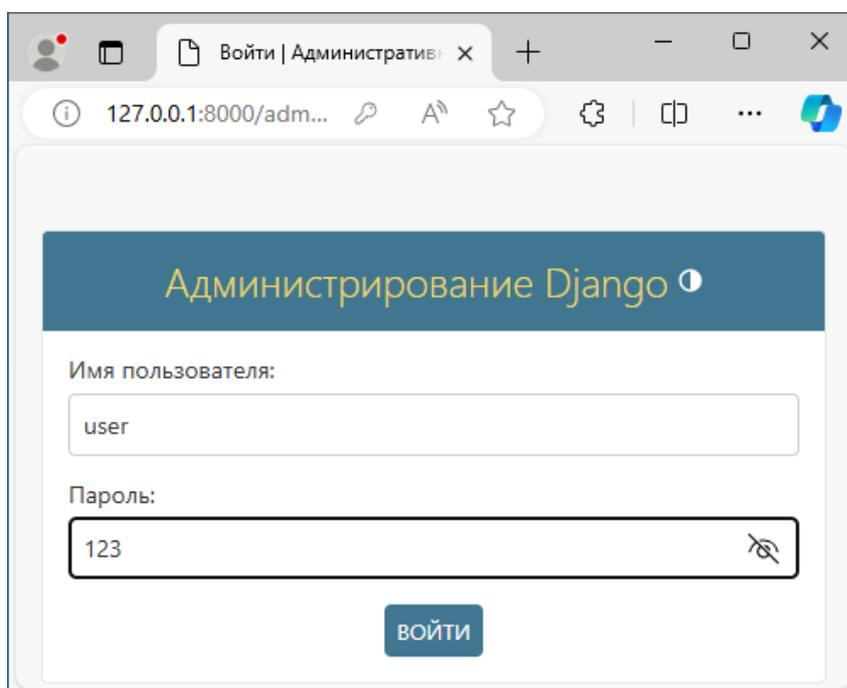
Рисунок 9 - Диаграмма развертывания

Для управления базой данных в режиме администратора в файле `admin.py` записан код, который позволяет администратору:

- вывести список всех грузов с возможностью поиска по названию груза;
- вывести список всех заказов с возможностью фильтрации по грузу, поиска по заказчику и сортировки по дате создания;
- вывести список всех перевозок с возможностью фильтрации по заказу, поиска по машине и водителю, а также сортировки по дате формирования.

```
from django.contrib import admin
from .models import Cargo, Order, Trucking
@admin.register(Cargo)
class CargoAdmin(admin.ModelAdmin):
    list_display = ('cargo_id', 'designation', 'marking', 'volume', 'weight',
'quantity', 'measurement_unit')
    search_fields = ('designation',)
@admin.register(Order)
class OrderAdmin(admin.ModelAdmin):
    list_display = ('order_id', 'cargo', 'customer', 'address', 'created', 'shipped',
'storekeeper', 'delivered')
    list_filter = ('cargo',)
    search_fields = ('customer',)
    date_hierarchy = 'created'
@admin.register(Trucking)
class TruckingAdmin(admin.ModelAdmin):
    list_display = ('trucking_id', 'order', 'car', 'driver', 'formed', 'completed')
    list_filter = ('order',)
    search_fields = ('car', 'driver',)
    date_hierarchy = 'formed'
```

С использованием административного доступа (рисунок 10) в программном обеспечении выполняется создание ролей и пользователей



. Рисунок 10 – Окно авторизации административного интерфейса

После авторизации административная панель (рисунок 11) содержит все введенные таблицы базы данных.

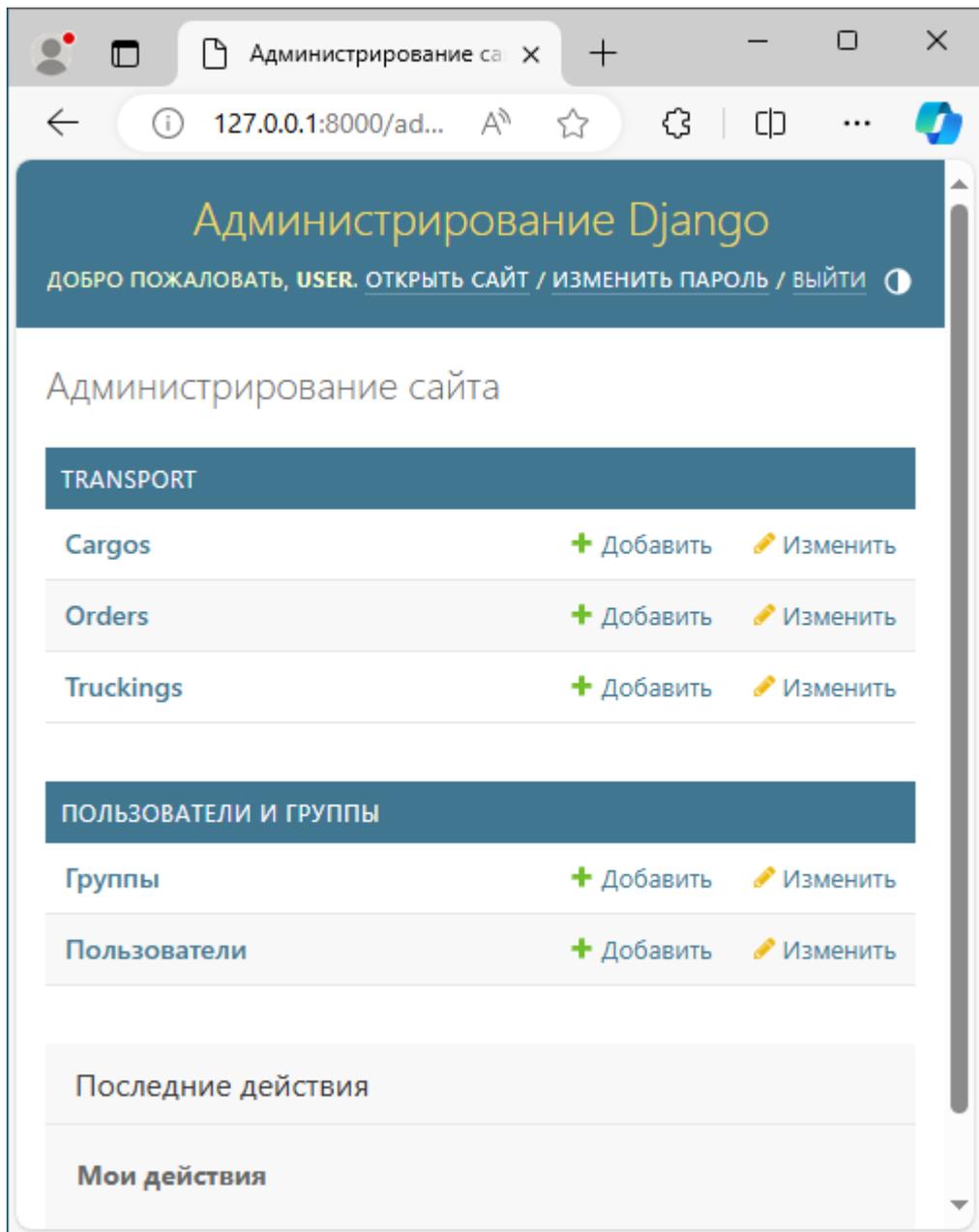


Рисунок 11 Административная панель приложения

Первым действием в тестировании разработанного программного обеспечения будет создание пользователей.

3.4 Тестирование программного обеспечения

В ходе тестирования программного обеспечения выполнено:

- создание пользователей (диспетчер, водители, заказчики, кладовщик);
- ввод заказчиком заявки;
- формирование диспетчером перевозки из нескольких заявок;
- просмотр водителем перевозки;
- отпуск кладовщиком груза;
- закрытие заявки заказчиком;
- мониторинг грузоперевозки диспетчером.

В ходе работы приложения в пользовательском режиме заказчик формирует заказ, диспетчер формирует перевозку из заказов, водитель просматривает состав своей перевозки, кладовщик отпускает грузы, заказчик принимает грузы. Таким образом в приложении необходимо 4 роли:

- заказчик;
- диспетчер;
- водитель;
- кладовщик.

Группа Заказчики может осуществлять действия с заказами и грузами из которых формируются заказы (рисунок 12)

Группа Диспетчеры обладает правами на все таблицы базы данных. Группа Водители может просматривать все таблицы базы данных. Группа Кладовщики имеет право на таблицы с заказами и грузами для того чтобы отметить дату и время отпуска груза по заказу и возможность отредактировать количество отпущенных грузов в случае когда запрашиваемое количество отличается от фактически отпущенного. По сравнению с группой Заказчики для Кладовщиков отсутствуют права на удаление и создание (рисунок 13).

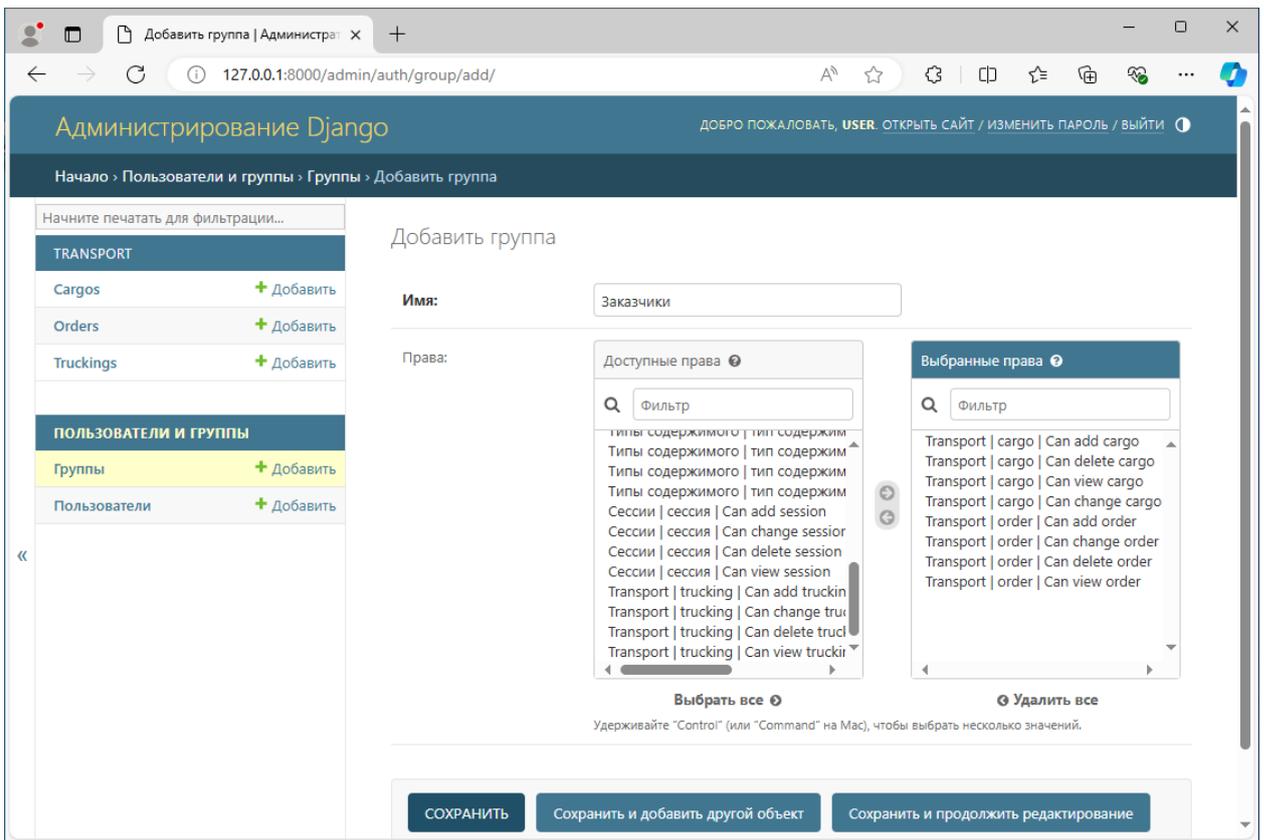


Рисунок 12 – Права группы Заказчики

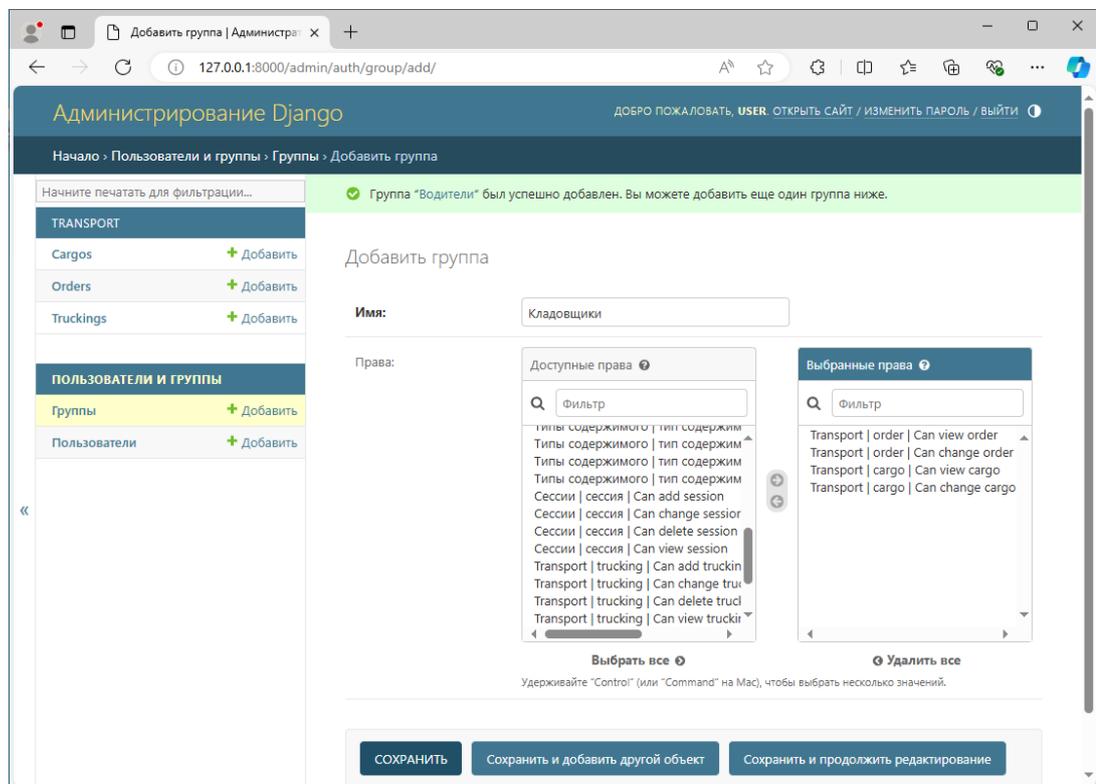


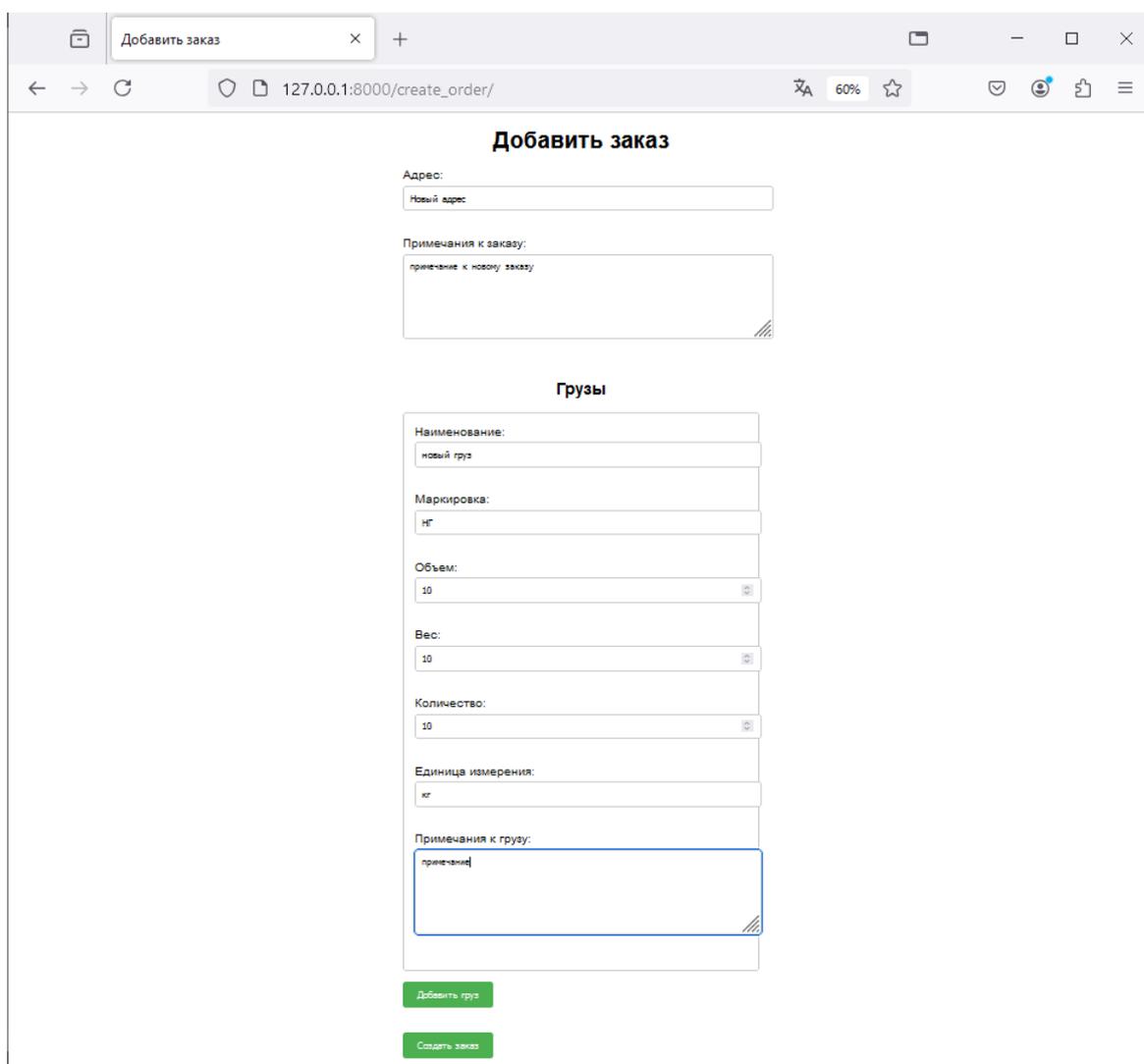
Рисунок 13 – Права группы Кладовщики

Далее создадим пользователей согласно таблице 4

Таблица 4 – Пользователи приложения

Логин	Пароль	Имя	Фамилия	Входит в группы
zak1	zakaz4ik1	Захар	Зиновьев	Заказчики
zak2	zakaz4ik2	Зинаида	Завьялова	Заказчики
dis1	dispat4er1	Дмитрий	Дуров	Диспетчеры
dis2	dispat4er2	Додон	Додонов	Диспетчеры
vod1	voditel1	Владимир	Водилов	Водители
vod2	voditel2	Варфоломей	Заводилов	Водители
klad1	kladov4ik1	Константин	Складской	Кладовщики
klad2	kladov4ik2	Клементий	Кладовой	Кладовщики

Экран формирования заявки показан на рисунке 14.



Добавить заказ

Адрес:
Новый адрес

Примечания к заказу:
примечание к новому заказу

Грузы

Наименование:
новый груз

Маркировка:
нг

Объем:
10

Вес:
10

Количество:
10

Единица измерения:
кг

Примечания к грузу:
примечание

Добавить груз

Создать заказ

Рисунок 14 – Формирование заявки

Экран формирования перевозки показан на рисунке 15.

Текущие перевозки

ID	Сформирована	Автомобиль	Водитель	Действия
1	23 августа 2024 г. 14:04	машина 1	Владимир Водилев	Просмотреть Закреть

Заказы

ID	Заказчик	Адрес	Суммарный вес	Суммарный объем	Действия
5	Захар Зиновьев	адрес первого заказчика	15,0	10,0	Просмотреть заказ Добавить в перевозку
6	Захар Зиновьев	адрес нового заказа	121,0	121,0	Просмотреть заказ Добавить в перевозку

Новая перевозка

Автомобиль:

Водитель:

Примечания:

ID	Заказчик	Адрес	Суммарный вес	Суммарный объем	Действия
6	Захар Зиновьев	адрес нового заказа	121,0	121,0	Просмотреть заказ Убрать из перевозки

[Сформировать перевозку](#)

Рисунок 15 – Формирование перевозки

Экраны мониторинга перевозки бригадиром показаны на рисунках 16 и 17.

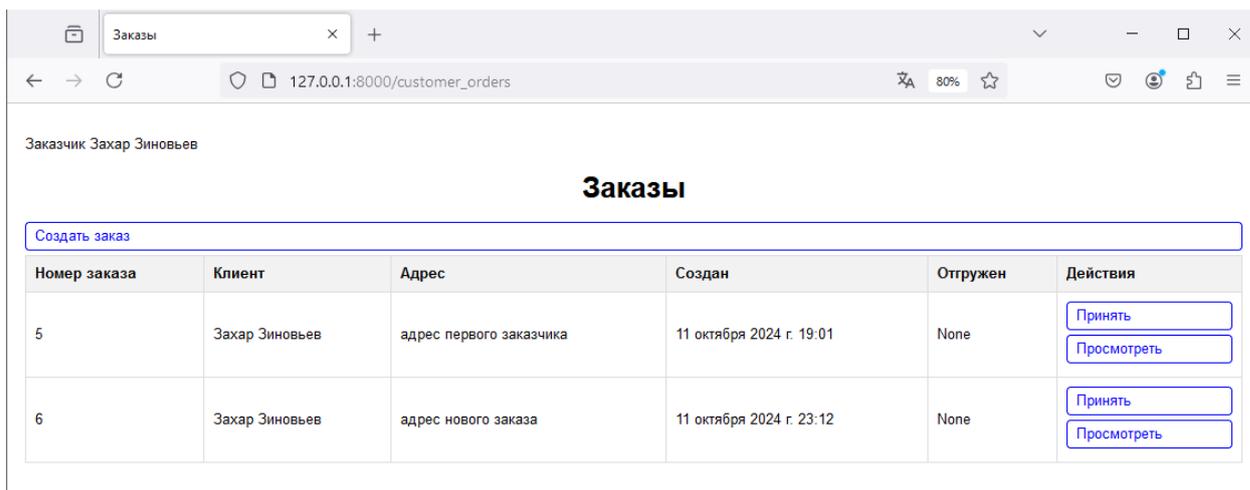


Рисунок 16 – Мониторинг заказов бригадиром

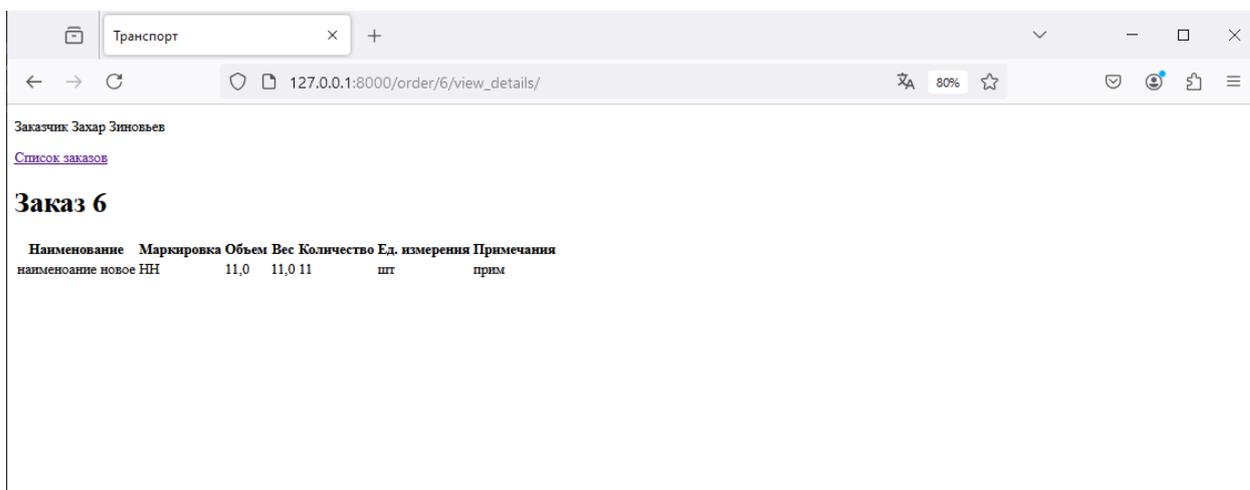


Рисунок 17 – Просмотр заказа бригадиром

Результаты тестирования приведены в таблице 5

Таблица 5 – Результаты тестирования

Действие	Результат
Создание пользователей (Диспетчер, Водители, Заказчики, Кладовщик)	Успешно. Сбоев и ошибок не выявлено
Ввод Заказчиком заявки	Успешно. Сбоев и ошибок не выявлено
Формирование Диспетчером перевозки из нескольких заявок	Успешно. Сбоев и ошибок не выявлено

Продолжение таблицы 5

Действие	Результат
Просмотр Водителем перевозки	Успешно. Сбоев и ошибок не выявлено
Отпуск Кладовщиком груза	Успешно. Сбоев и ошибок не выявлено
Закрытие заявки Заказчиком	Успешно. Сбоев и ошибок не выявлено
Мониторинг грузоперевозки Диспетчером	Успешно. Сбоев и ошибок не выявлено

Выводы по разделу.

Проведен анализ языков программирования, фреймворков и СУБД, в результате чего для разработки программного обеспечения для управления и мониторинга грузоперевозок выбран язык Python, фреймворк Django и СУБД SQLite.

В результате реализации информационной модели программного обеспечения разработана физическая модель данных для выбранной СУБД и написан соответствующий код.

В ходе моделирования стадии разработки проведен анализ структуры разрабатываемого приложения и физического размещения его элементов. Результаты представлены на диаграмме компонентов и диаграмме размещения.

В результате тестирования программного обеспечения проверены основные функции - создание пользователей (диспетчер, водители, заказчики, кладовщик), ввод заказчиком заявки, формирование диспетчером перевозки из нескольких заявок, просмотр водителем перевозки, отпуск кладовщиком груза, закрытие заявки заказчиком, мониторинг грузоперевозки диспетчером. Тестирование прошло успешно.

Заключение

В работе выполнена разработка программного обеспечения для управления и мониторинга грузоперевозок

Описана деятельность организации, показана ее организационная структура. Разработана процессная модель, описывающая основные бизнес-процессы, связанные с грузоперевозками. Деятельность по управлению и мониторингу грузоперевозок описана с использованием методологии IDEF0. Выполнен анализ и постановка задачи на разработку программного обеспечения, разработка требований и обзор аналогов.

Разработана диаграмма вариантов использования, на которой выделены основные классы пользователей и представлены варианты использования программного обеспечения управления грузоперевозками.

Детализация вариантов использования use case диаграммы выполнена с помощью диаграммы деятельности, на которой показан алгоритм формирования грузоперевозки диспетчером.

Выполнен выбор архитектуры и моделирование данных, в ходе которого разработаны концептуальная и логическая модели данных программного обеспечения управления и мониторинга грузоперевозок

Проведен анализ языков программирования, фреймворков и СУБД, в результате чего для разработки программного обеспечения для управления и мониторинга грузоперевозок выбран язык Python, фреймворк Django и СУБД SQLite.

В результате реализации информационной модели программного обеспечения разработана физическая модель данных для выбранной СУБД и написан соответствующий код.

В ходе моделирования стадии разработки проведен анализ структуры разрабатываемого приложения и физического размещения его элементов. Результаты представлены на диаграмме компонентов и диаграмме размещения.

В результате тестирования программного обеспечения проверены основные функции - создание пользователей (диспетчер, водители, заказчики, кладовщик), ввод заказчиком заявки, формирование диспетчером перевозки из нескольких заявок, просмотр водителем перевозки, отпуск кладовщиком груза, закрытие заявки заказчиком, мониторинг грузоперевозки диспетчером. Тестирование прошло успешно.

В целом разработанное программное обеспечение для управления и мониторинга грузоперевозок соответствует требованиям к ПО, обладает необходимой функциональностью и проверено в использовании посредством тестирования.

Таким образом все задачи выпускной квалификационной работы выполнены, цель выпускной квалификационной работы достигнута.

Список используемой литературы и используемых источников

1. Автоплан [Электронный ресурс]. URL: <https://avtoplan.ru/> (дата обращения: 19.10.2024).
2. Анализ популярных реляционных систем управления базами данных [Электронный ресурс]. URL: <https://drach.pro/blog/hi-tech/item/196-popular-relational-dbms-2022> (дата обращения: 12.10.2024).
3. Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс]. URL: <https://habr.com/ru/post/566218/> (дата обращения: 24.09.2024).
4. Казиев, В. М. Введение в анализ, синтез и моделирование систем: учебное пособие. Москва, Саратов: Интернет Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 270 с.
5. Корпоративный сайт Creatly [Электронный ресурс] : UML Diagram Types Guide: Learn About All Types of UML Diagrams with Examples URL: <https://creately.com/blog/diagrams/uml-diagram-types-examples/> (дата обращения 19.09.2024).
6. Ларман К.Ф. Применение UML 2.0 и шаблонов проектирования М: Диалектика; Москва, 2016. – 736с.
7. ЛогистПро [Электронный ресурс]. URL: <https://logistpro.su/> (дата обращения: 19.10.2024).
8. Моделирование данных: обзор [Электронный ресурс]. URL: <https://habr.com/ru/post/556790/> (дата обращения: 11.09.2024).
9. Носова, Л. С. Case-технологии и язык UML: учебно-методическое пособие. Челябинск, Саратов: Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. 67 с.
10. Стешин, А. И. Информационные системы в организации: учебное пособие. Саратов: Вузовское образование, 2019. 194 с.

11. Тараканов О.В. Базы данных. Национальный исследовательский ядерный университет «МИФИ». ООО «Научно-издательский центр ИНФРАМ»; Москва, 2019. 304 с.
12. Теория и практика UML. Диаграмма деятельности [Электронный ресурс]. URL: http://www.it-gost.ru/articles/view_articles/96 (дата обращения: 12.09.2024).
13. ТОП-20 CRM-систем для логистики и грузоперевозок [Электронный ресурс]. URL: <https://otzyvmarketing.ru/articles/crm-sistemy-dlya-logistiki-i-gruzoperevozok/> (дата обращения: 19.10.2024).
14. A Guide to the Entity Relationship Diagram (ERD) [Электронный ресурс]. URL: <https://www.databasestar.com/entity-relationship-diagram/> (дата обращения: 24.09.2024).
15. Date C. J. Database Design and Relational Theory: Normal Forms and All That Jazz Paperback, O'Reilly Media, 2019. 472 p.
16. ISO/IEC/IEEE 31320-2:2012 Information technology. Modeling Languages. Part 2: Syntax and Semantics for IDEF1X [Электронный ресурс]. <https://www.iso.org/standard/60614.html> (дата обращения: 14.09.2024).
17. SQLite [Электронный ресурс]. URL: <https://www.sqlite.org/> (дата обращения: 29.09.2024).
18. SQLite замечательная встраиваемая БД [Электронный ресурс]. <https://habr.com/ru/articles/149356/> (дата обращения: 19.09.2024).
19. Soware [Электронный ресурс]. URL: <https://soware.ru/categories/logistics-management-systems> (дата обращения: 19.10.2024).
20. TransTrade [Электронный ресурс]. URL: <https://transoft.ru/> (дата обращения: 19.10.2024).