**федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ» (КНИТУ-КАИ)**

**ОТДЕЛЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**Д И П Л О М Н Ы Й П Р О Е К Т**

**Тема:** Разработка приложения для автоматизации транспортной логистики\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель**  преподаватель |  |  |  | Л. М. Сунгатуллина |
| *(должность)* |  | *(подпись)* |  | *(И.О. Фамилия)* |
| **Обучающийся**  4435 |  |  |  | Н. А. Кравченко |
| *(группа)* |  | *(подпись)* |  | *(И.О. Фамилия)* |

**Специальность** 09.02.07 «Информационные системы и программирование»

**Казань 2023** **г.**

Лист задания Лист задания

**АННОТАЦИЯ**

Тема дипломной работы: «Разработка приложения для автоматизации транспортной логистики».

Цель разработки дипломного проекта – создание функционирующих модулей клиентов, менеджеров логистов и администрации программной системы, предназначенной для автоматизации решения транспортной задачи.

Для достижения данной цели были использована платформа Windows Presentation Foundation для создания десктоп-приложений для Windows.

Разработанный программный продукт, в качестве десктоп-приложения в дальнейшем может использоваться организациями, которые сталкиваются с задачами транспортной логистики.

Объем дипломной работы составляет 137 страниц, в состав которых входят: 107 рисунков, 3 таблиц. В процессе разработки дипломного проекта было использовано 20 литературных источников.

Ключевые слова: функциональные модули, десктоп-приложение, логистика, транспортная задача, опорный план.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc130151655)

[ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc130151656)

[1.1 Описание целевой организации 8](#_Toc130151657)

[1.2 Автоматизируемые бизнес-процессы целевой организации 8](#_Toc130151658)

[1.3 Существующие аналоги разрабатываемой системы 8](#_Toc130151659)

[ГЛАВА 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ 9](#_Toc130151660)

[2.1 Постановка задачи 9](#_Toc130151661)

[2.2 Диаграмма вариантов использования 9](#_Toc130151662)

[2.3 Разработка схемы пользовательского интерфейса 9](#_Toc130151663)

[2.4 Диаграмма деятельности 9](#_Toc130151664)

[2.5 Диаграмма состояний 9](#_Toc130151665)

[2.6 Проектирование базы данных 9](#_Toc130151666)

[2.7 Проектирование архитектуры информационной системы 9](#_Toc130151667)

[ГЛАВА 3 ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ 10](#_Toc130151668)

[3.1 Руководство пользователя 10](#_Toc130151669)

[3.2 Руководство программиста 10](#_Toc130151670)

[3.3 Руководство по установке программного обеспечения 10](#_Toc130151671)

[ЗАКЛЮЧЕНИЕ 11](#_Toc130151672)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc130151673)

[ПРИЛОЖЕНИЕ А 15](#_Toc130151674)

[ПРИЛОЖЕНИЕ Б 16](#_Toc130151675)

[ПРИЛОЖЕНИЕ В 17](#_Toc130151676)

# ВВЕДЕНИЕ

В бизнесе происходят десятки процессов: поиск поставщиков, перевозчиков, производство, работа склада, доставка заказов покупателям. Логистика помогает оптимально организовать все эти процессы.

Она затрагивает все процессы от планирования производства до доставки покупателю.

Логистика управляет разными производственными потоками — то есть перемещением груза на пути от производства к потребителю.

Цель логистики — минимизировать затраты, оптимизировать процессы, а значит, снизить расходы и увеличить прибыль.

Актуальность транспортной логистики заключается в том, что транспорт является связующим звеном между предприятием производителем и потребителем. Кроме того, фактор логистики усиливает внешнеэкономически связи между государствами, способствует международному разделению труда. Сам процесс производства заканчивается именно в тот момент, когда товар доставлен потребителю. Фактор логистики является одним из важнейших при размещении производства.

Одним из ключевых элементов успешной деятельности в этой области являются эффективные транспортные перевозки. Информационная система автоматизации транспортной логистики значительно повысит уровень автоматизации и управления в этой сфере, что делает данную тему актуальной и необходимой для разработки и реализации. В рамках данного дипломного проекта будет создана система, которая поможет оптимизировать процессы решения транспортной задачи, облегчит выбор поставщика и расчет стоимости поставок.

Таким образом, актуальной задачей, решаемой в дипломной работе, является автоматизация работы логистической компании.

Целью данной работы является оптимизация процесса сбора данных о пунктах, снижение количества ошибок при распределении поставок, повышение качества обслуживания клиентов, подбор наиболее подходящих перевозчиков для клиентов логистической компании. Эффективность системы будет определяться точностью и скоростью решения транспортных задач, быстротой нахождение путей перевозок для клиентов логистической компании и наличием актуальной информации о пунктах и путях перевозки.

Достижение поставленной цели потребовало решения следующих задач:

1. Анализ предметной области транспортной логистики;
2. Проектирование системы оформления и решения заказов;
3. Разработка базы данных для программного модуля оформления и решения заказов;
4. Разработка программного модуля;
5. Разработка технической документации для дальнейшего сопровождения и эксплуатации разработанного программного решения целевыми пользователями.

Проект разработки информационной системы для решения задач транспортной логистики будет иметь следующие положительные результаты:

1. Оптимизация работы транспортной компании и контроль поставок. Использование информационной системы позволит увеличить точность и скорость планирования поставок, что снизит количество ошибок и затрат при выполнении поставок.
2. Повышение качества обслуживания клиентов. Благодаря более эффективной работе транспортной компании, клиенты получат свои поставки быстрее и с меньшими затратами.
3. Выбор подходящего перевозчика. Использование ИС позволит выбрать для исполнения поставок наиболее подходящего для клиента поставщика, благодаря чему увеличится качество исполнения поставок и снизятся расходы на выполнение поставок.
4. Повышение общей эффективности организации. Благодаря более эффективной работе транспортной компании, а также улучшению качества обслуживания клиентов, организация сможет повысить свою конкурентоспособность на рынке и увеличить свою прибыль.

# ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

# Цель проекта

Целью дипломной работы является разработка приложения для логистической компаний, которое позволит автоматизировать работу менеджеров-логистов в целевой организации.

# Автоматизируемые бизнес-процессы целевой организации

Разработка программного решения позволит автоматизировать следующие бизнес-процессы:

* Составления оптимальных маршрутов движения и планирование транспортного процесса;
* Улучшение качества обслуживания клиентов;
* Расчет стоимости перевозки;
* Оптимальное распределение заявок;
* Выбор перевозчика.

# 1.3 Существующие аналоги разрабатываемой системы

Одним из аналогов разрабатываемого программного решения является «Программа для решения транспортной задачи» (Рисунок 1). Это приложение позволяет решать закрытые транспортные задачи в отсутствии интернет-трафика различными методами и оптимизировать составленный ею опорный план.

Минусами данного приложения является: Неудобный ввод и вывод данных в систему путем чтения и записи файла формата .txt, наличие множества критичных ошибок, ведущих к завершению работы программы, неудобный пользовательский интерфейс.

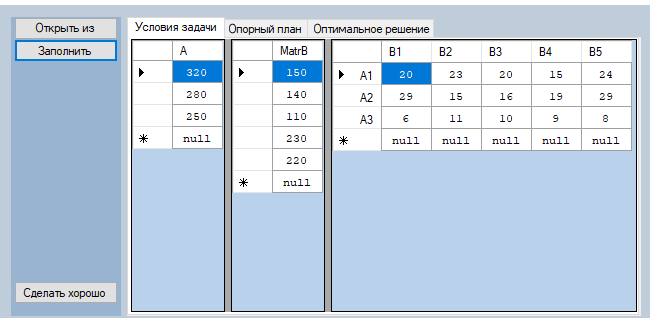


Рисунок –«Программа для решения транспортной задачи»

Другим аналогом является программа «Axelot WMS X5» (Рисунок 2). Данное комплексное программное обеспечение, позволяет компаниям управлять логистическими процессами и адаптироваться к конъюнктуре рынка. Программа имеет следующий функционал:

* автоматизирует работу склада;
* повышает эффективность работы персонала;
* помогает управлять заявками и планировать рейсы;
* ведет учет собственного автопарка и ГСМ;
* помогает контролировать водителей в течение рабочего дня;

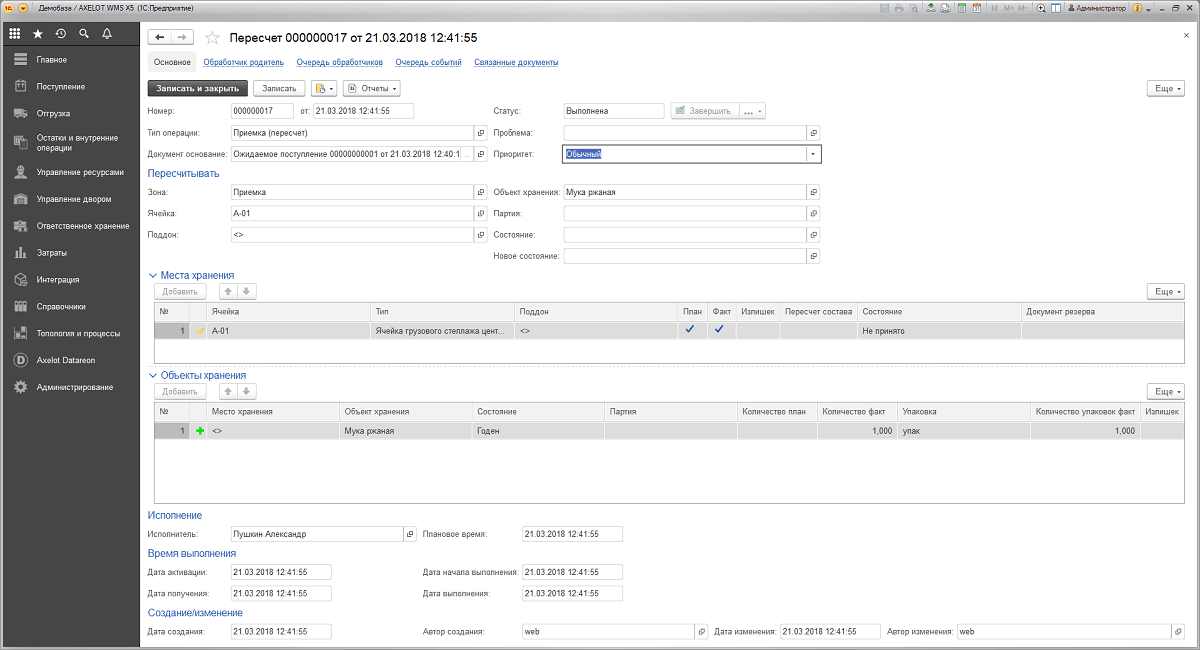


Рисунок 2 – Программное решение «Axelot WMS X5»

Минусами приложения является высокая стоимость, относительная сложность обучения, относительно высокое потребление вычислительных мощностей компьютера.

Еще одним аналогом является программное решение «SAP. ERP система» (Рисунок 3). Она помогает компаниям работать быстрее, проще и эффективнее. Функционал программы:

* учет командировочных расходов, бухгалтерский учет и данные платежных карт в единой системе;
* управление взаимодействием с поставщиками, начиная с их поиска и заканчивая заключением контрактов;
* формирует базу данных, которая используется для расширенной аналитики;
* улучшает управление расходами компании за счет автоматической обработки отчетов.

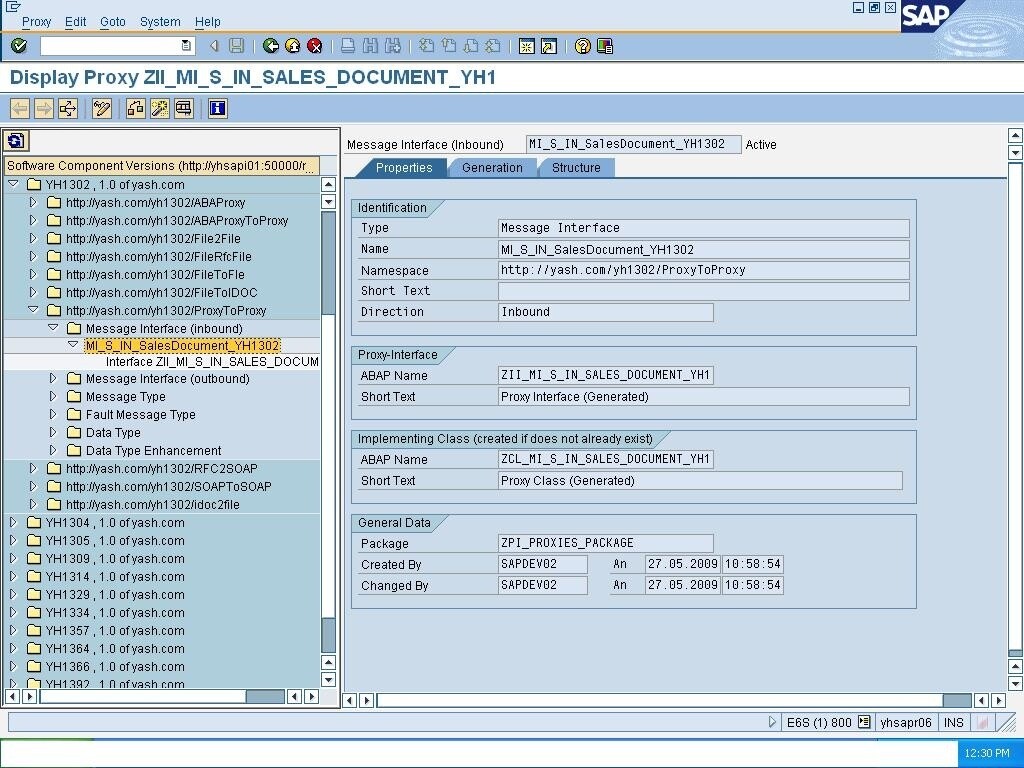


Рисунок 3 – Программное решение «SAP. ERP система»

Минусами данной системы является очень высокий срок окупаемости инвестиций в установку системы, замыкание компании на контракте с поставщиком – контракт привязывает компанию к поставщику на срок действия контакта, что делает экономически невыгодным смену поставщика, а так-же относительно долгий период обучения сотрудников работе в данной системе.

# ГЛАВА 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

# 2.1 Постановка задачи

Целью данной разработки является создание функционирующего модуля менеджера-логиста, предназначенного для создания заявок на осуществление поставок, создание объектов перевозчиков для исполнения поставок, составления оптимальных путей распределения поставок, создание объектов клиентов логистической компании.

Модуль менеджера-логиста клиентами должен содержать сведения и возможности для создания, просмотра и редактирования данных о клиентах, заявках, перевозчиков, а также инструменты на решения задач транспортной логистики.

Для достижения поставленной цели необходимо прийти к решению и реализовать следующие задачи:

1. Провести анализ предметной области для выявления требования к информационной системе;

2. Выбрать средства разработки и реализации программных модулей;

3. Спроектировать и создать дизайн программных модулей;

4. Спроектировать архитектуру информационной системы;

5. Спроектировать базу данных под нужды создаваемой системы.

6. Разработать пользовательский интерфейс для всех пользователей системы;

8. Разработать функциональные модули программной системы в соответствии с выбранными средствами разработки;

9. Выполнить разработку руководств пользователя, программиста и установке программного обеспечения для дальнейшего сопровождения и помощи в эксплуатации пользователями разработанного программного продукта.

# Выбор средств реализации

При реализации настольного(desktop) приложения информационной системы будет использована Windows Presentation Foundation(WPF) на платформе .NET Framework;

.NET Framework — это среда для разработки программного обеспечения, которая позволяет создавать и запускать приложения. .NET — это большая платформа для разработчиков. Она состоит из разных инструментов, языков программирования и библиотек. И помогает программистам разрабатывать веб-сайты, приложения, игры и сервисы. Главная задача этого инструмента — упростить работу разработчика. .NET Framework распространяется вместе Windows и используют его исключительно для создания настольных приложений Windows — чаще всего это масштабные корпоративные приложения.

Windows Presentation Foundation (WPF) - это платформа пользовательского интерфейса для создания настольных приложений Windows. WPF поддерживает широкий набор функций разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязку данных и документы. WPF использует расширяемый язык разметки приложений (XAML), главным назначением которого является конструирование пользовательских интерфейсов WPF.

Приложения WPF основаны на архитектуре векторной графики. Это позволяет приложениям отлично смотреться на мониторах с высоким разрешением, поскольку их можно масштабировать бесконечно. WPF также включает гибкую модель хостинга, которая упрощает размещение видео, например, с помощью кнопки.

Следует отметить, что неотъемлемой частью любой [информационной системы](https://pandia.ru/text/category/informatcionnie_sistemi/) является база данных. База данных (БД) — это организованная структура, которая предназначается для хранения, обработки и изменения большого количества информации.

База данных необходима для работы с большим объемом упорядоченной информации и предназначено для работы нескольких пользователей, которые вносят изменения одновременно, независимо друг от друга.

Для управления базой данных используется [система управления базами данных (СУБД)](https://ktonanovenkogo.ru/voprosy-i-otvety/subd-chto-ehto-takoe.html). СУБД представляет собой интерфейс между самой БД и пользователями и дает возможность получать и обновлять информацию, управлять ее упорядочиванием и оптимизацией. Она также позволяет пользователю следить за производительностью, настраивать, выполнять резервное копирование и восстановление БД.

При реализации дипломного проекта в качестве СУБД будет использоваться SQL Server Management Studio (SSMS). SSMS - это мощный инструмент от Microsoft, который, среди прочего, позволяет управлять базами данных SQL Server. Преимущества решения:

* Тесная интеграция с операционной системой Windows.
* Высокая производительность, отказоустойчивость.
* Поддержка многопользовательской среды.
* Расширенные функции резервирования данных.
* Работа с удаленным подключением.

В качестве архитектуры разрабатываемой информационной системы была выбрана двухуровневая архитектура, так как она обеспечивает изоляцию операций управления данными, обработки данных и операций представления.

* Клиент содержит слои презентации, бизнес-логики и передачи данных.
* Сервер включает хранилища и базы данных.

Сервер будет располагаться на одном устройстве с базой данных и будет взаимодействовать посредством технологии Entity Framework Core современного средства для отображения объектов базы данных для .NET. Оно поддерживает запросы LINQ, отслеживание изменений, обновления и миграции схемы. Entity Framework Core успешно работает с SQL Server. Клиент и сервер будут взаимодействовать друг с другом через Интернет.

Схема двухуровневой архитектуры на рисунке 4.

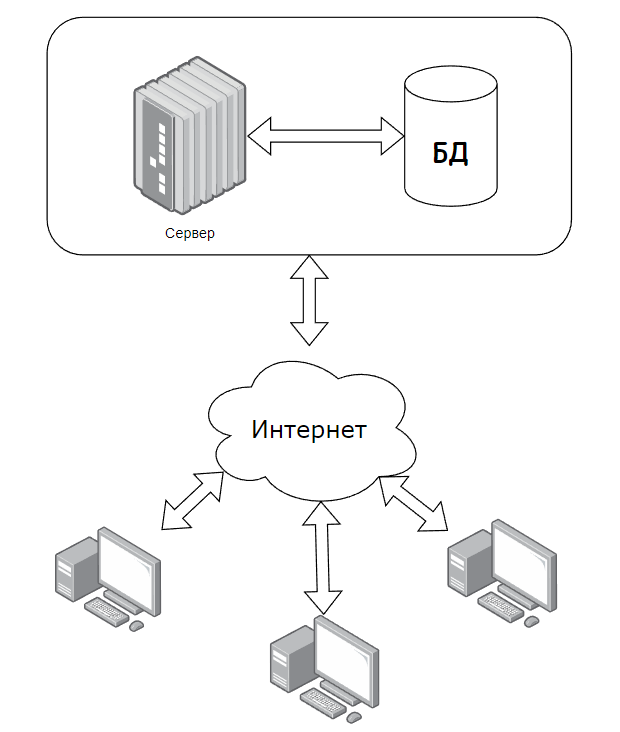


Рисунок 4 – Архитектура приложения

# Концептуальная модель информационной системы

На основе бизнес-процессов, указанных в пункте 1.2 были сформированы пользовательские требования к разрабатываемой информационной системе (Таблица 1).

Таблица 1 – Выявленные пользовательские требования

|  |  |
| --- | --- |
| № | Пользовательское требование |
| 1 | Подать заявку на постройку оптимального маршрута снабжения. |
| 2 | Возможность внести данные о стоимости перевозок между пунктами. |
| 3 | Возможность оптимально распределить поставки от производителя к потребителю. |
| 4 | Возможность отслеживать и просматривать ход выполнения заявки. |
| 5 | Возможность хранить информацию о перевозчиках. |
| 6 | Импорт результатов в документ формата .docx. |
| 7 | Добавить ограничения к распределению поставок. |
| 8 | Возможность работы с пунктами клиентов. |

Для разрабатываемой системы были выявлены следующие акторы, взаимодействующие с ней:

* Неавторизованный пользователь;
* Менеджер-логист;

Описание акторов представлено вТаблице 2.

Таблица 2 – Выявление акторов

|  |  |
| --- | --- |
| Акторы | Краткое описание |
| Неавторизованный пользователь | Пользователь, который не авторизован в системе. |
| Менеджер-логист | Пользователь, производящий работу с клиентами, перевозчиками и заявками, поступающими в систему. |

Выявленные варианты использования для ранее указанных акторов представлены в таблице 3.

Таблица 3 – Выявленные варианты использования

|  |  |  |
| --- | --- | --- |
| Основной актор | Наименование | Формулировка |
| Неавторизованный пользователь | Авторизоваться | Неавторизованный пользователь может ввести логин и пароль что бы авторизоваться в системе |
| Менеджер логист | Создать клиента | Менеджер логист может создать объект клиента для хранения соответствующей информации |
| Менеджер логист | Посмотреть список клиентов | Менеджер логист может просмотреть информацию о клиентах логистической компании |
| Менеджер логист | Изменить список клиентов | Менеджер логист может изменять информацию о клиентах логистической компании для актуализации информации о нем |
| Менеджер логист | Внести информацию о новой заявке | Менеджер логист может создать объект заявки в системе для последующего хранения |
| Менеджер логист | Внести список ограничений к заявке | Менеджер логист может добавить ограничения к уже созданной заявке |

Таблица 3 – Выявленные варианты использования (Продолжение)

|  |  |  |
| --- | --- | --- |
| Менеджер логист | Обработать заявку | Менеджер логист может обработать созданную заявку для получения оптимального решения |
| Менеджер логист | Посмотреть список заявок | Менеджер логист может просматривать информацию о созданных заявках |
| Менеджер логист | Изменить список заявок | Менеджер логист может изменять информацию о созданных заявках |
| Менеджер логист | Создать перевозчика | Менеджер логист может создать объект перевозчика для хранения информации о нем |
| Менеджер логист | Просмотреть список перевозчиков | Менеджер логист может просмотреть информацию о перевозчиках |
| Менеджер логист | Изменить список перевозчиков | Менеджер логист может изменить информацию о перевозчиках |
| Менеджер логист | Посмотреть список услуг перевозчика | Менеджер логист может просмотреть список услуг перевозчика |
| Менеджер логист | Изменить список услуг перевозчика | Менеджер логист может изменить список услуг выбранного перевозчика |

# Диаграмма вариантов использования

Диаграмма вариантов использования — это визуальное обобщение взаимодействий и отношений внутри системы. Эти диаграммы показывают очень широкое представление о системе. Они могут показывать системы в компьютерном программном обеспечении, бизнесе или клиентском опыте. На диаграмме вариантов использования показан модельный сценарий, в котором люди взаимодействуют с системой, используя набор специальных символов и соединителей.

Исходя из выявленных ранее вариантов использования (Таблица 3) для акторов (Таблица 2), на рисунке 5 была представлена диаграмма вариантов использования.

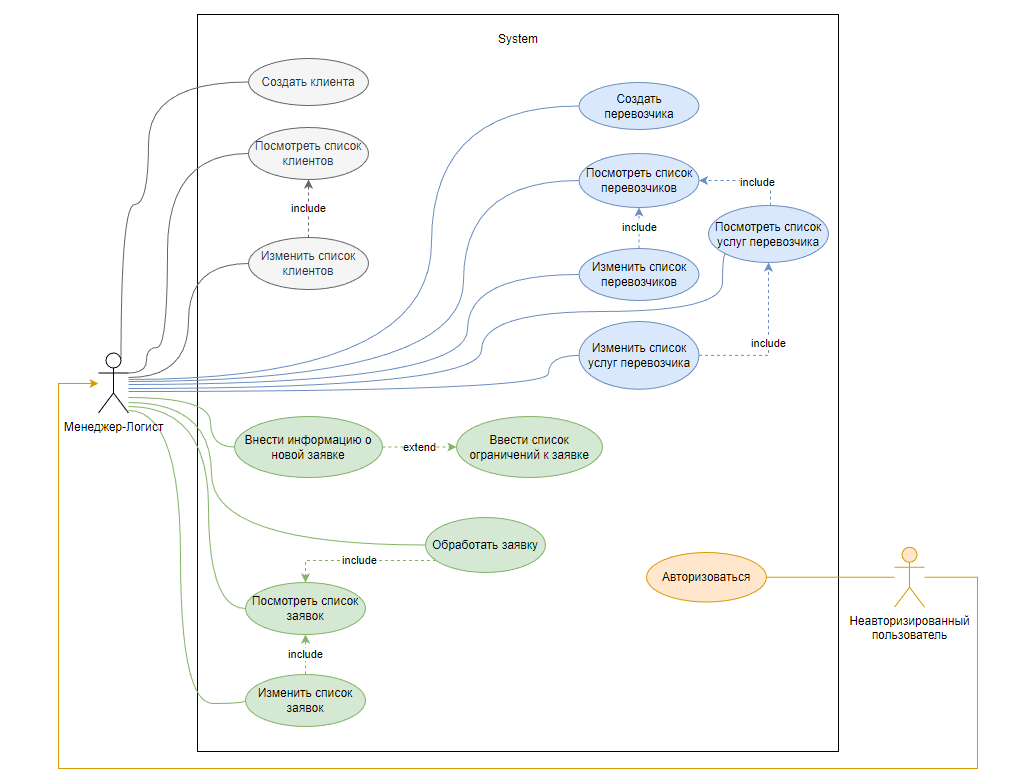


Рисунок 5 – Диаграмма вариантов использования

# Диаграмма последовательности

**Диаграмма последовательности** (sequence diagram) – диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления.

С помощью диаграммы последовательности можно представить взаимодействие элементов модели как своеобразный временной график "жизни" всей совокупности объектов, связанных между собой для реализации варианта использования программной системы, достижения бизнес-цели или выполнения какой-либо задачи.

Как правило, диаграмма взаимодействия описывает поведение только одного варианта использования. На такой диаграмме отображаются только экземпляры объектов и сообщения, которыми они обмениваются между собой в рамках данного варианта использования.

На рисунке 6 представлена спроектированная диаграмма последовательности для варианта использования «Авторизация».

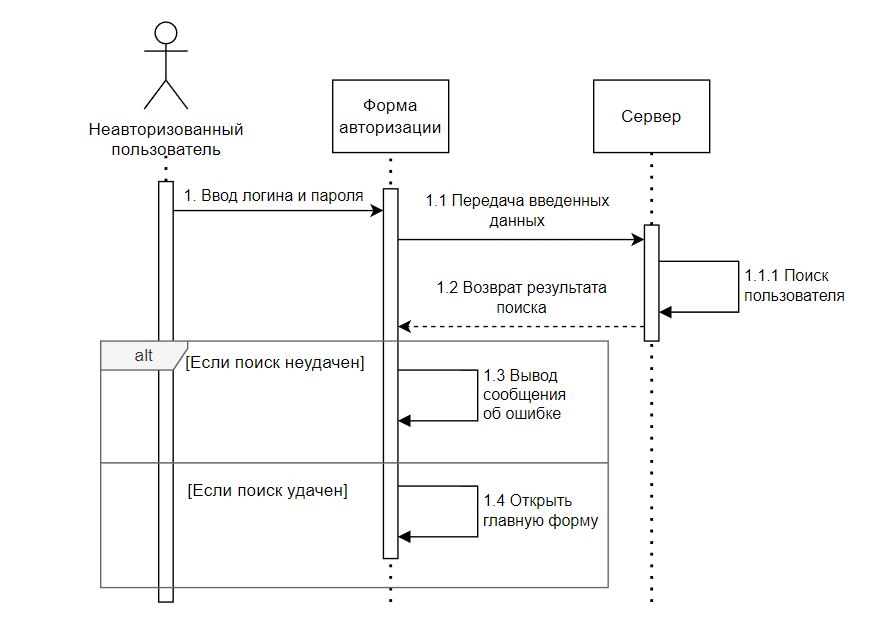


Рисунок 6 – Диаграмма последовательности прецедента «Авторизация»

На рисунке 7 представлена спроектированная диаграмма последовательности для варианта использования «Создать клиента».

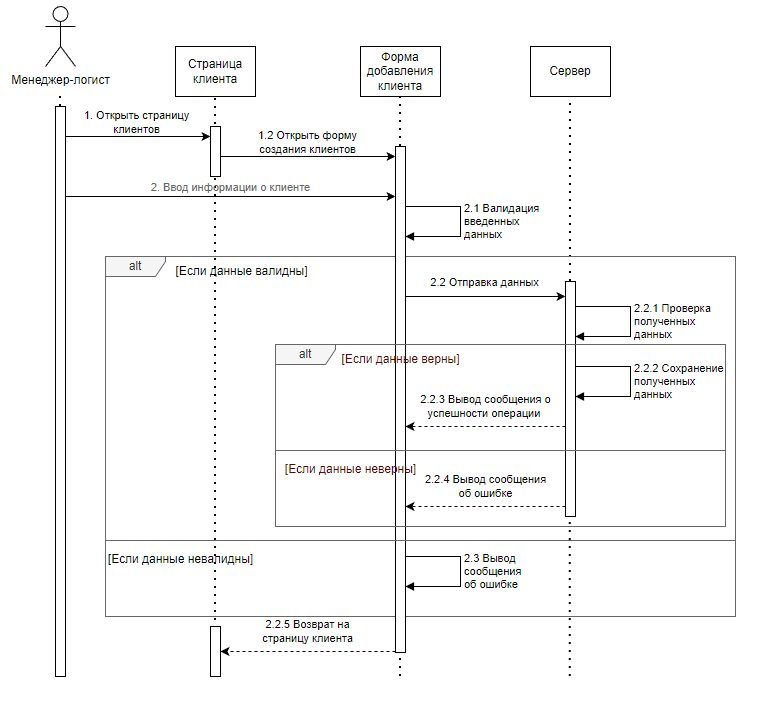


Рисунок 7 – Диаграмма последовательности прецедента «Создать клиента»

# Диаграмма деятельности

Диаграммы деятельности (Activity Diagram) создаются для детализации прецедентов, выделенных в диаграмме прецедентов. Диаграммы деятельности служат для иллюстрации последовательности действий, которая выполняется для реализации каждого отдельно взятого прецедента. Виды деятельности и отдельные действия соединяются между собой потоками, движение которых происходит от выходов одного узла к входам другого.

На рисунке 8 представлена спроектированная диаграмма последовательности для прецедента обработать заявку.

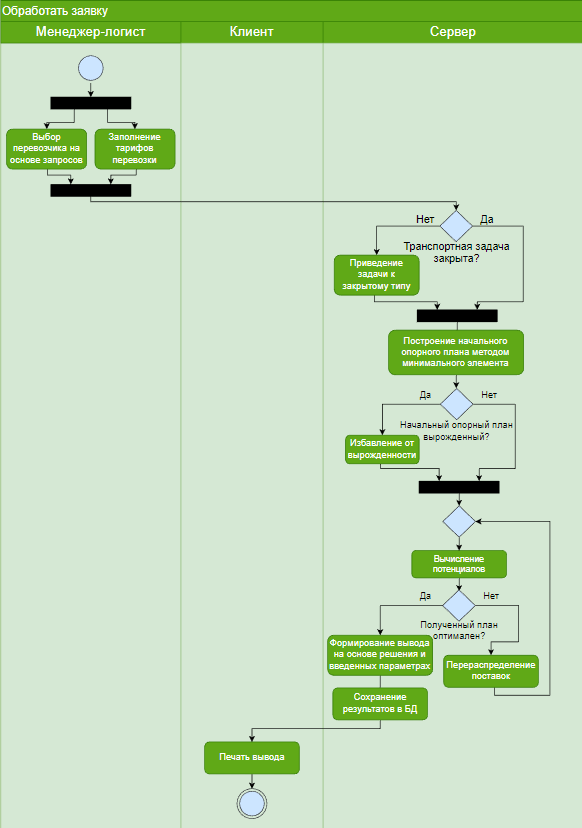


Рисунок 8 – Диаграмма деятельности для прецедента «Обработать заявку»

На рисунке 9 представлена спроектированная диаграмма последовательности для прецедента «Внести информацию о новой заявке»

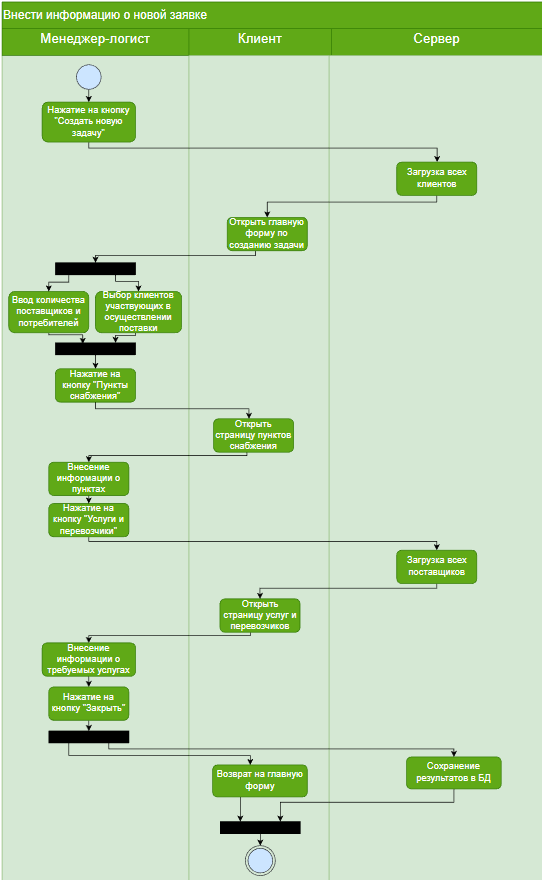


Рисунок 9 – Диаграмма деятельности для варианта использования

«Внести информацию о новой заявке»

На рисунке 10 представлена спроектированная диаграмма последовательности для прецедента ввести список ограничений к заявке

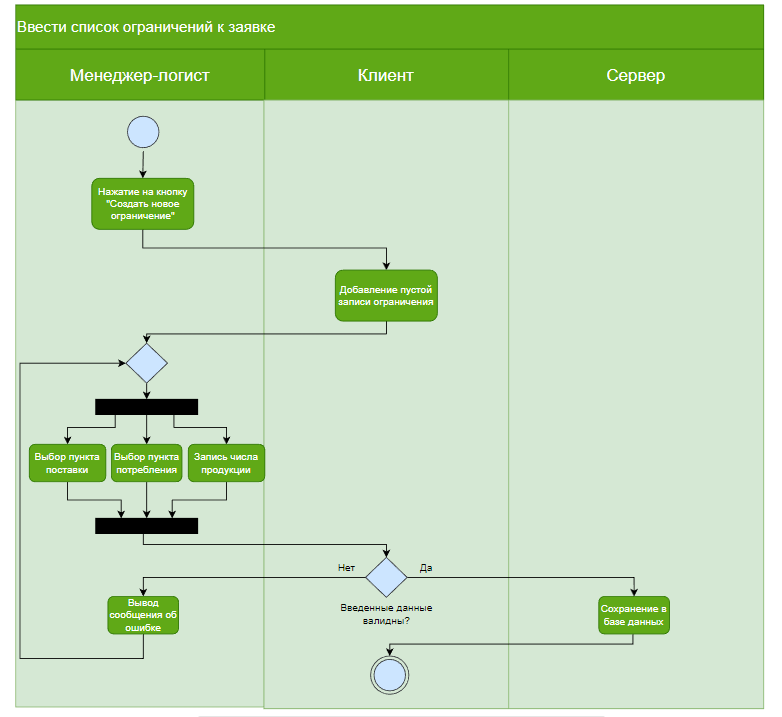


Рисунок 10 – Диаграмма деятельности для варианта использования

«Ввести список ограничений к заявке»

# 2.7 Диаграмма состояний

Диаграмма состояний показывает, как объект переходит из одного состояния в другое. Она служит для моделирования динамических аспектов системы. Данная диаграмма полезна при моделировании жизненного цикла объекта. От других диаграмм диаграмма состояний отличается тем, что описывает процесс изменения состояний только одного экземпляра определенного класса - одного объекта, причем объекта реактивного, то есть объекта, поведение которого характеризуется его реакцией на внешние события.

Составим диаграмму состояний для объекта «Пользователь» (Рис. 11).

Описание состояний:

1. Не авторизирован. Пользователь не авторизован в системе, требуется ввести логин и пароль для доступа к системе;

2. Авторизован. Пользователь авторизован в системе и имеет полный доступ ко всему функционалу приложения.

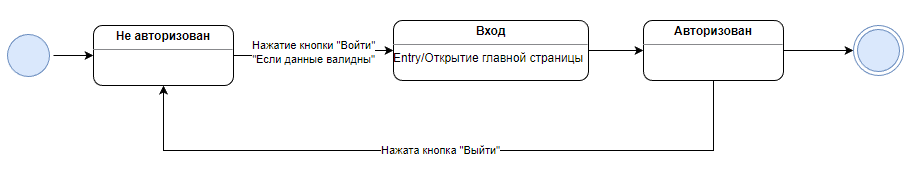


Рисунок 11 – Диаграмма состояния для объекта «Пользователь»

Составим диаграмму состояний для объекта «Задача» (Рисунок 12).

Описание состояний:

1. Создана. Задача создана и готова к дальнейшей обработке менеджером-логистом;

2. Обработана. Задача обработана менеджером-логистом и готова к отправке перевозчику;

3. Отправлена. Задача прошла обработку и была отправлена перевозчику на выполнение;

4. Редактирована. Задача была отредактирована менеджером и требует повторной обработки перед следующей отправке перевозчику;

5. Удалена. Задача удалена из системы и более никак не будет использоваться.

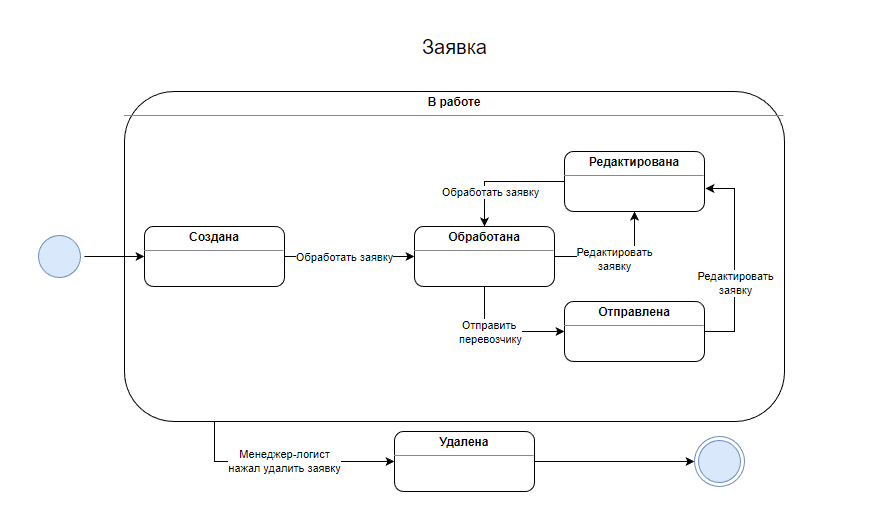


Рисунок 12 – Диаграмма состояния для объекта «Пользователь»

# Диаграмма классов

Диаграмма классов (class diagram) служит для представления статической структуры модели системы в объектно-ориентированном программировании. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

Разработанная диаграмма классов для проектируемого приложения изображена на рисунке 13.

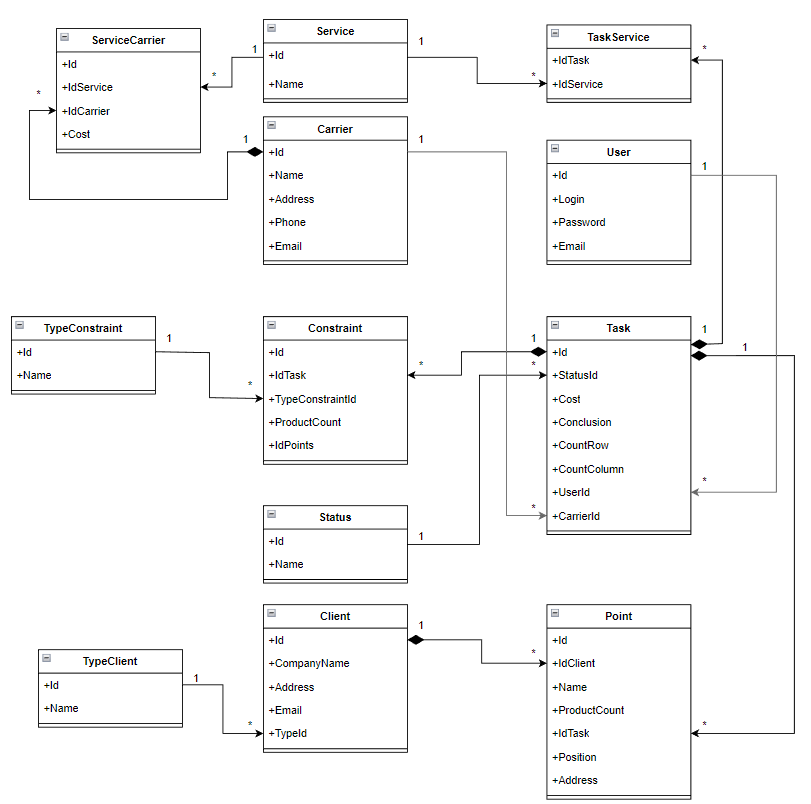


Рисунок 13 – Диаграмма классов

# Проектирование интерфейса приложения

Исходя из диаграммы вариантов использования, разработанной в пункте 2.4 можно сказать, что для успешной работы менеджера-логиста в системе необходимы следующие окна:

* Авторизации;
* Клиентов;
* Перевозчиков;
* Заявок;
* Добавления клиента;
* Добавления перевозчика;
* Работы с клиентами;
* Точек снабжения;
* Работы над услугами заявок;
* Работы с ограничениями;
* Работы с тарифами;
* Заключения.

Все макеты, которые будут рассмотрены далее, были разработаны в графическом редакторе WPF (онлайн редактора draw.io).

При запуске приложения неавторизованного пользователя встречает окно авторизации (Рисунок 14).

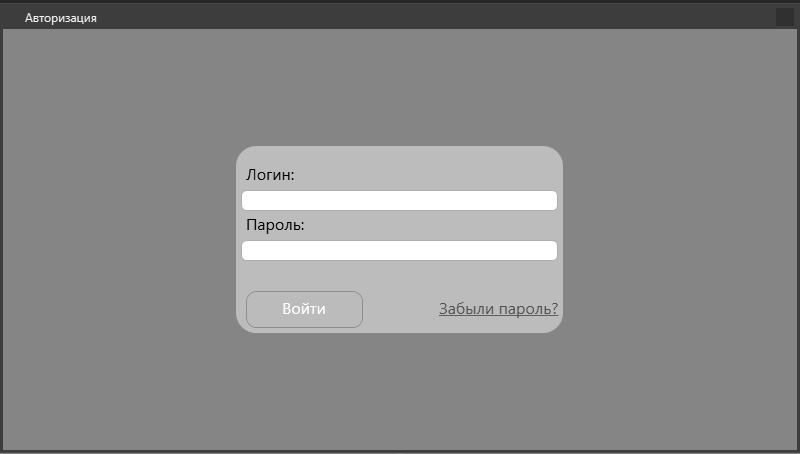


Рисунок 14 – Окно авторизации

После успешной авторизации менеджер логист попадает на главную форму, где отображается окно клиентов, на котором можно работать со списком клиентов (Рисунок 15).

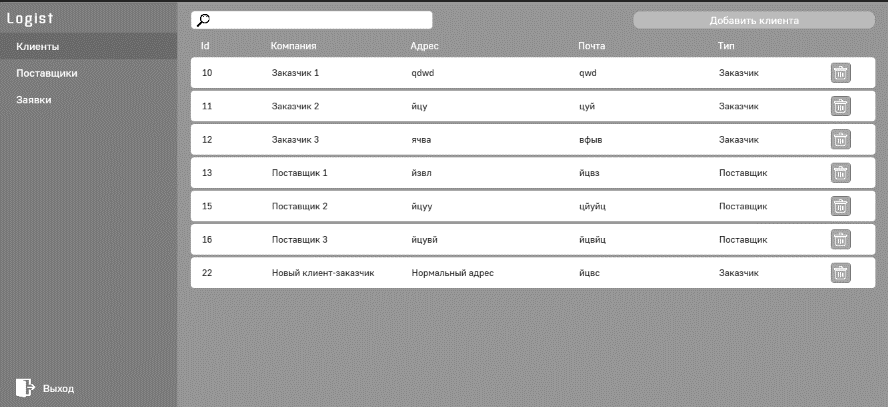


Рисунок 15 – Окно клиенты

Нажав на кнопку «Добавить клиента», откроется форма по созданию нового клиента (Рисунок 16).

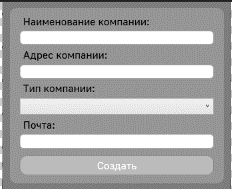


Рисунок 16 – Окно создания клиента

При нажатии на кнопку «Поставщики» расположенной в левой части приложения, менеджер-логист попадет на форму работы с данными перевозчиков (Рисунок 17).

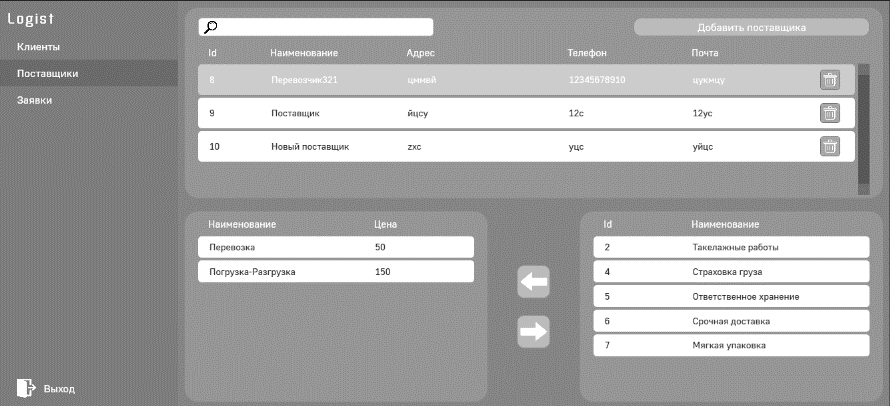


Рисунок 17 –Окно перевозчики

Нажав на кнопку «Добавить перевозчика», откроется форма по добавлении нового перевозчика в систему (Рисунок 18).

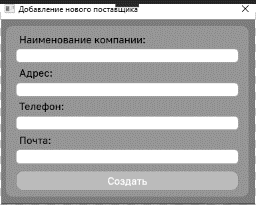


Рисунок 18 – Окно создания перевозчика

При нажатии на кнопку «Заявки», расположенной в левой части приложения, менеджер-логист попадет на форму с заявками, где он сможет работать с ними (Рисунок 19). Для редактирования заявки нужно дважды нажать на требуемую запись, для удаления нажать на изображение корзины, для создания нажать на кнопку «Создать новую задачу» в верхней части экрана.

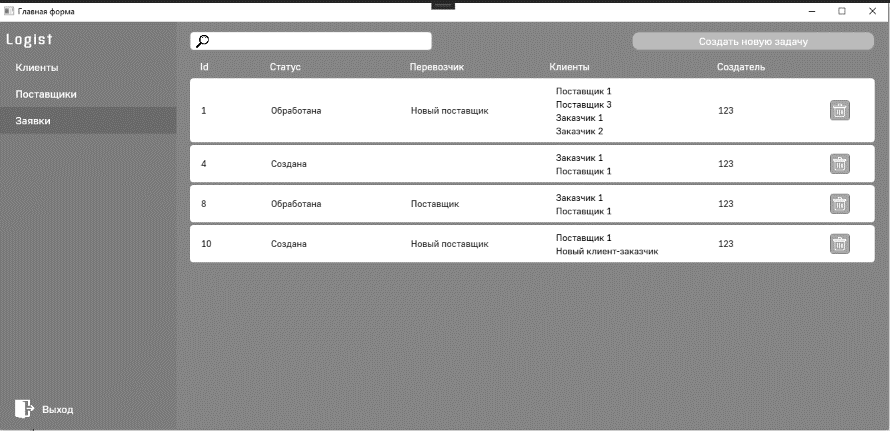


Рисунок 19 – Окно заявки

При переходе на форму создания новой задачи путем нажатия на кнопку «Создать новую задачу», на окне заявки, откроется форма работы с заявкой окно работы с клиентами (Рисунок 20), где пользователь сможет указать число и требуемых клиентов в заявке.

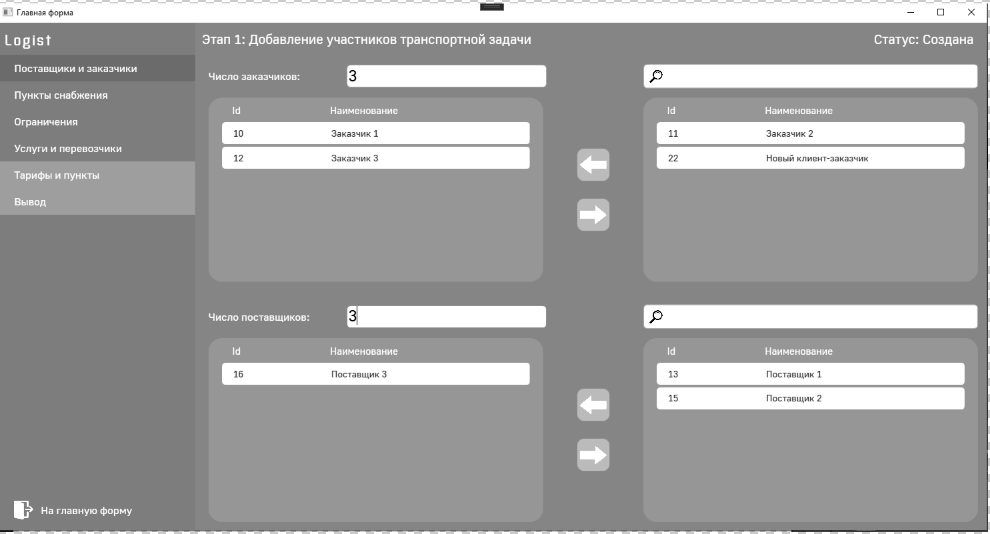


Рисунок 0 – Окно работы с клиентами

При нажатии на кнопку «Пункты снабжения», расположенной в левой части приложения, менеджер-логист попадет на окно точек снабжения, где он сможет работать с ними (Рисунок 21).

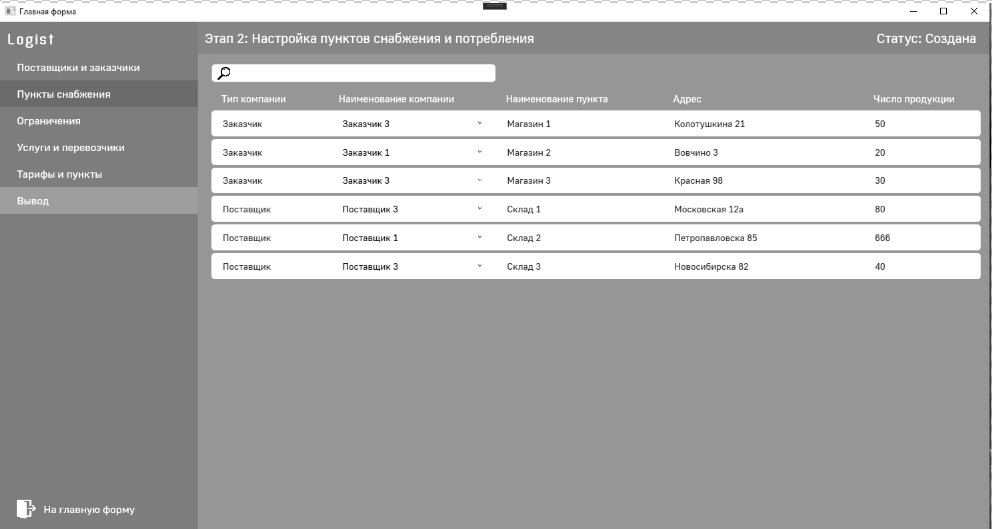


Рисунок 21 – Окно точек снабжения

При нажатии на кнопку «Ограничения», менеджер-логист попадет на окно работы с ограничениями, где он получит возможность накладывать ограничения на решаемую транспортную задачу (Рисунок 22).

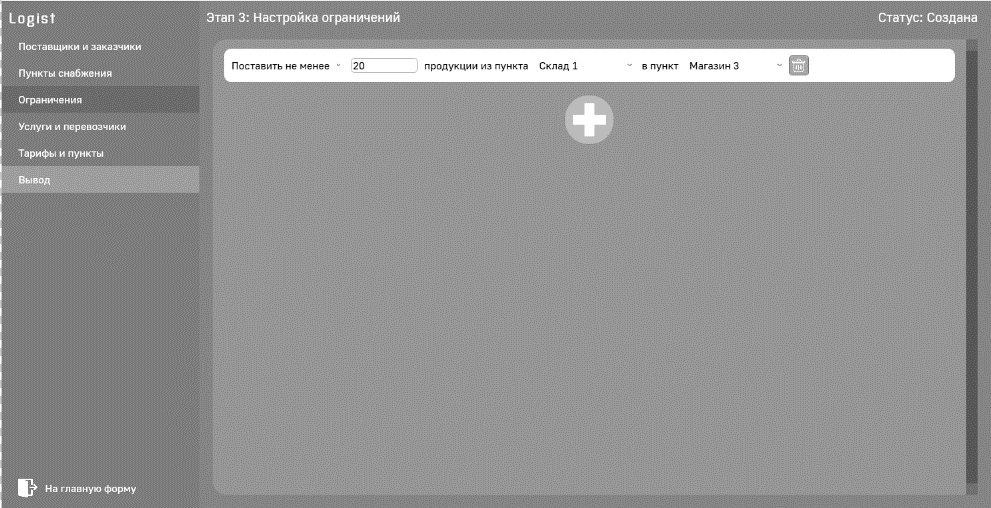


Рисунок 22 – Окно работы с ограничениями

После перехода на окно работы над услугами заявок пользователь сможет выбрать перевозчика и предоставляемые услуги (Рисунок 23).

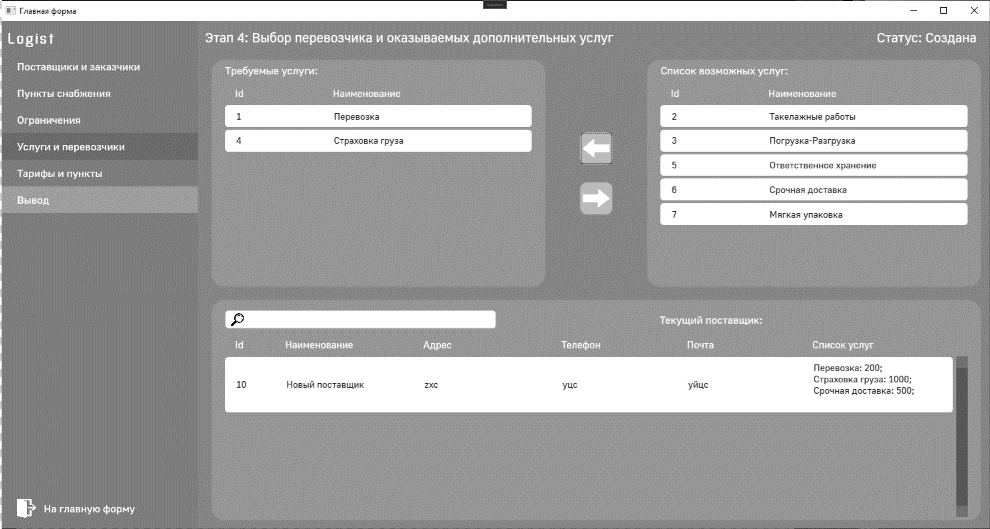


Рисунок 23 – Окно работы над услугами

Далее на окне работы с тарифами, переход на которое производится при помощи меню слева, пользователь сможет ввести стоимость перевозки между пунктами (Рисунок 24).

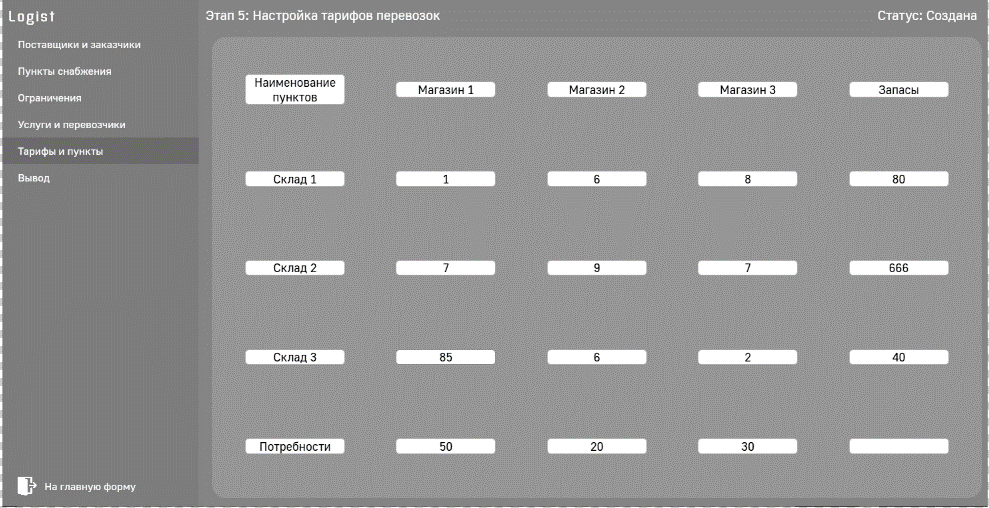


Рисунок 24 – Окно тарифов

На окне заключения пользователь сможет сгенерировать решение транспортной задачи (Рисунок 25).

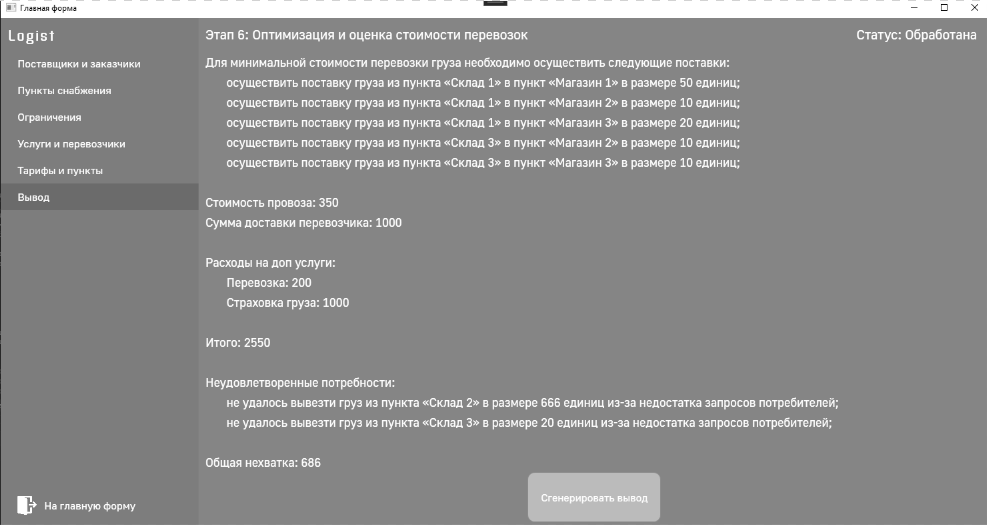


Рисунок 25 – Окно заключения

После того, как разработаны макеты всех необходимых окон, можно сформировать диаграмму потоков экранов (Рисунок 26).

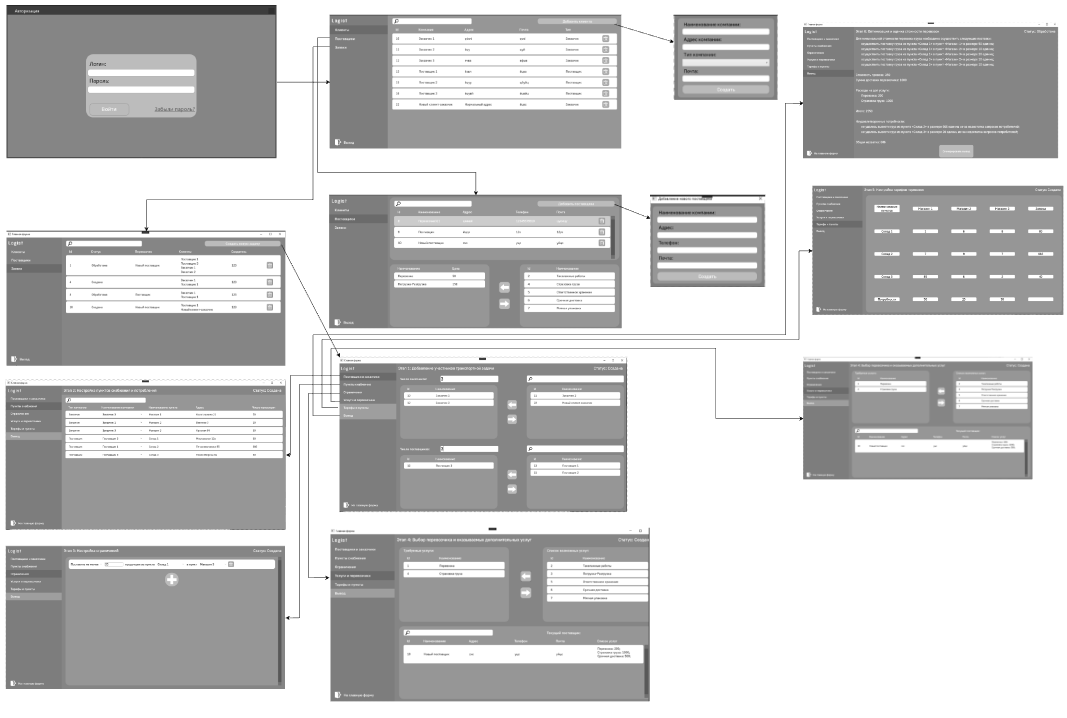


Рисунок 26 – Диаграмма потоков экранов

# Математическое обеспечение

# 2.10.1 Определение транспортной задачи

Наиболее распространенной задачей линейного программирования является транспортная задача - задача о поиске экономичного плана перевозок однородного продукта из пунктов производства в пункты потребления.

Одной из видов транспортной задачи является классическая. Это задача оптимального плана перевозок однородного продукта в однородные пункты потребления на однородных транспортных средствах.

Транспортная задача является особой формой поиска оптимального плана перевозок груза с минимальными затратами. С помощью транспортной задачи можно найти оптимальный план перевозки грузов, с минимальными затратами.

Цель транспортной задачи - обеспечить доставку продукции потребителю в нужное время и место при минимальной стоимости трудовых, материальных и финансовых ресурсов.

# 2.10.2 Постановка транспортной задачи

В общей постановке транспортная задача выглядит следующим образом.

Имеется m пунктов отправления (поставщиков)  с запасами  единиц груза. Имеется n пунктов назначения (потребителей)  с потребностями  . Груз из пунктов отправления должен быть доставлен в пункты назначения. Известны транспортные издержки , связанные с перевозкой единицы груза из пункта  в пункт .

Требуется составить такой план перевозок, при котором весь груз из пунктов отправления был бы доставлен потребителям и при этом спрос потребителей был бы удовлетворён, а транспортные издержки были минимальными.

Для наглядности транспортную задачу удобно представлять в виде таблицы, которую называют распределительной.

В таблице количество груза, перевозимого от поставщика  к потребителю  обозначено через .

Матрица  называется матрицей тарифов, а числа  – тарифами.

Матрица   называется планом транспортной задачи (Таблица 4). Здесь каждое число  обозначает количество единиц груза, который надо доставить от i-го поставщика к j-му потребителю.

Таблица 4 – План транспортной задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поставщики | Потребители | | | | Запасы  груза |
|  |  | … |  |  |
|  |  |  | … |  |  |
|  |  |  | … |  |  |
| … | … | … | … | … | … |
|  |  |  | … |  |  |
| Потребность  в грузе |  |  | … |  |  |

Транспортная задача решается несколькими методами:

1. Метод «северо-западного угла».
2. Метод минимального элемента.
3. Метод Фогеля.

# 2.10.3 Метод минимального элемента

При этом методе первой заполняется та клетка таблицы, имеющая наименьшее число. Если такая клетка не одна, то можно заполнить любую из них. Процесс распределения запасов длится до тех пор, пока все запасы не будут удовлетворены.

Этот метод решения транспортной задачи был выбран в следствии простоты автоматизации, большей эффективности среди других методов нахождения опорного плана, потребляет меньше вычислительных и временных ресурсов.

# 2.10.4 Метод потенциалов

После нахождения опорного плана необходимо проверить его на **оптимальность.** Чтобы установить является ли опорный план оптимальным, надо проверить, как повлияет на величину целевой функции любое возможное перераспределение поставок.

План распределения поставок будет оптимальным лишь в том случае, когда целевая функция имеет минимальное значение, т.е. когда дальнейшее уменьшение затрат на поставку будет невозможно.

Для улучшения опорного плана будет использоваться метод потенциалов. Он позволяет упростить наиболее трудоемкую часть вычислений — нахождение оценок свободных клеток.

Его достоинствами являются: не надо искать циклов с отрицательной ценой. Достаточно быстро определяет оптимальный план, относительная простота автоматизации.

# Проектирование базы данных

Традиционно процедуру проектирования базы данных разбивают на три этапа, каждый из которых завершается созданием соответствующей информационной модели.

[**Этап 1-й. Концептуальное проектирование**](https://github.com/kolei/PiRIS/blob/master/articles/5_1_1_1_erd2.md#%D0%9A%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – создание схемы БД, включающего определение важнейших сущностей (таблиц) и связей между ними, но не зависящего от модели БД и физической реализации (целевой СУБД).

[**Этап 2-й. Логическое проектирование**](https://github.com/kolei/PiRIS/blob/master/articles/5_1_1_1_erd2.md#%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – развитие концептуальной схемы БД с учетом принимаемой модели.

[**Этап 3-й. Физическое проектирование**](https://github.com/kolei/PiRIS/blob/master/articles/5_1_1_1_erd2.md#%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – развитие логической схемы БД с учетом выбранной целевой СУБД.

# 2.11.1 Концептуальное проектирование базы данных

Цель **концептуального проектирования** – создание концептуальной схемы данных на основе представлений о предметной области каждого отдельного типа пользователей. Концептуальная схема представляет собой описание основных сущностей (таблиц) и связей между ними без учета принятой модели БД и синтаксиса целевой СУБД. Часто на такой схеме отображаются только имена сущностей (таблиц) без указания их атрибутов. Представление пользователя включает в себя данные, необходимые конкретному пользователю для принятия решений или выполнения некоторого задания.

В соответствии с требуемыми данными необходимыми пользователю, для информационной системы можно выделить следующие основные сущности:

Пользователь;

Клиент;

Пункт;

Перевозчик;

Услуга;

Задача;

Тарифы перевозки;

Ограничения.

На основе выделенных сущностей можно построить концептуальную схему базы данных (Рисунок 27).

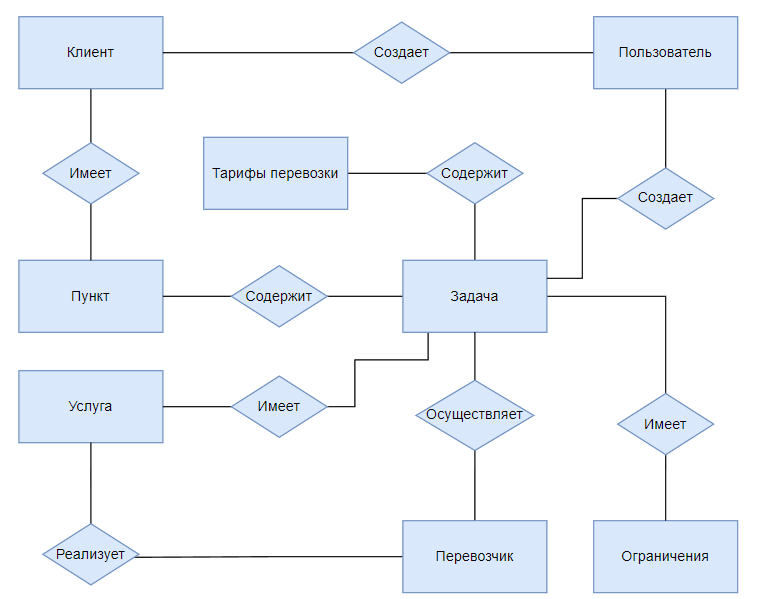


Рисунок 27 – Концептуальная модель БД

# 2.11.2 Логическое проектирование базы данных

Логическое (даталогическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Логическая структура БД должна соответствовать логической модели предметной области и учитывать связь модели данных с поддерживаемой СУБД. Поэтому этап начинается с выбора модели данных, где важно учесть её простоту и наглядность.

Цель **логического проектирования** – развить концептуальную схему БД с учетом принимаемой модели БД (иерархической, сетевой, реляционной и т. д.).

Принципы нормализации базы данных описывают, как надо правильно проектировать таблицы для хранения данных. Если им не следовать, то потом с БД будет неудобно работать.

Нормальные формы, пронумерованные от 1NF до 6NF. Каждая форма содержит определенный список требований, которым должна соответствовать база данных. При этом требования для N-й нормальной формы включают в себя и требования к формам с меньшими номерами. Обычно достаточно, чтобы БД находилась в третьей нормальной форме (3NF).

Требования 3NF:

* Все значения в ячейках должны быть атомарными, то есть содержать ровно одно неделимое значение, а не список из нескольких значений.
* Все поля в таблице должны полностью зависеть от первичного ключа целиком, а не от его части.
* Значения полей должны зависеть только от первичного ключа, а не от других полей.

Примем в качестве модели реляционную БД в третьей нормальной форме. В качестве мер нормализации выделим часто повторяющиеся данные в отдельные таблицы: TypeClient, Status, TypeConstraint. Добавим вспомогательные таблицы для таблиц со связью многие ко многим, а именно: TaskService, ServiceCarrier. После всех преобразований построим и построим логическую модель базы данных (Рисунок 28).

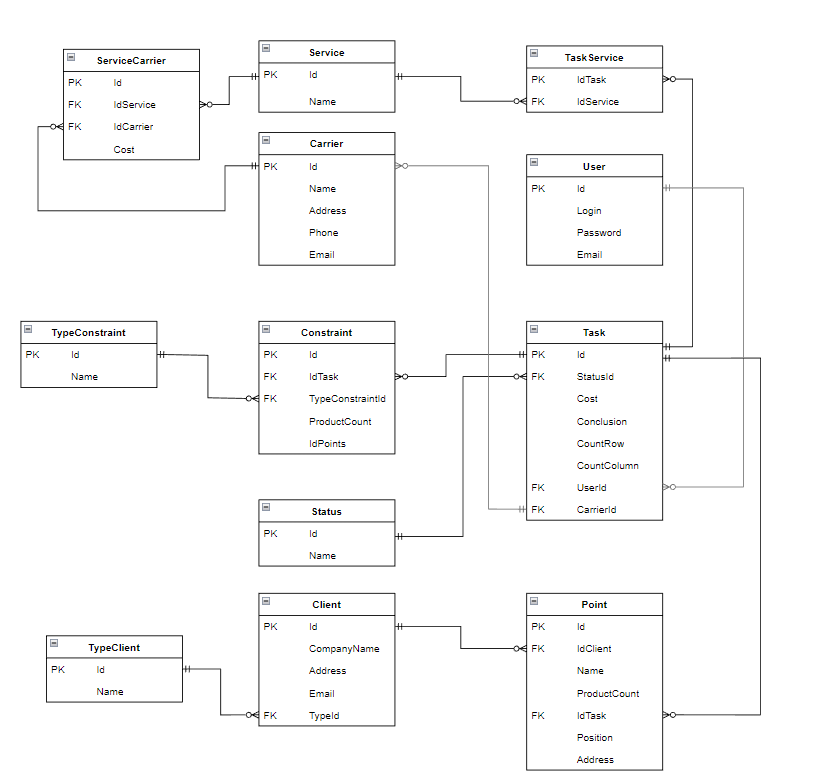


Рисунок 28 – Логическая модель БД

# 2.11.3 Физическое проектирование базы данных

Цель **физического проектирования** – преобразование логической схемы с учетом синтаксиса, семантики и возможностей выбранной целевой СУБД.

На основе СУБД, указанной в пункте 2.2 было произведено физическое проектирование базы данных (Рисунок 29). Подробное описание типов данных указано в словаре данных разработанном на основе диаграммы сущность-связь из пункта 2.11.2 в приложении А.

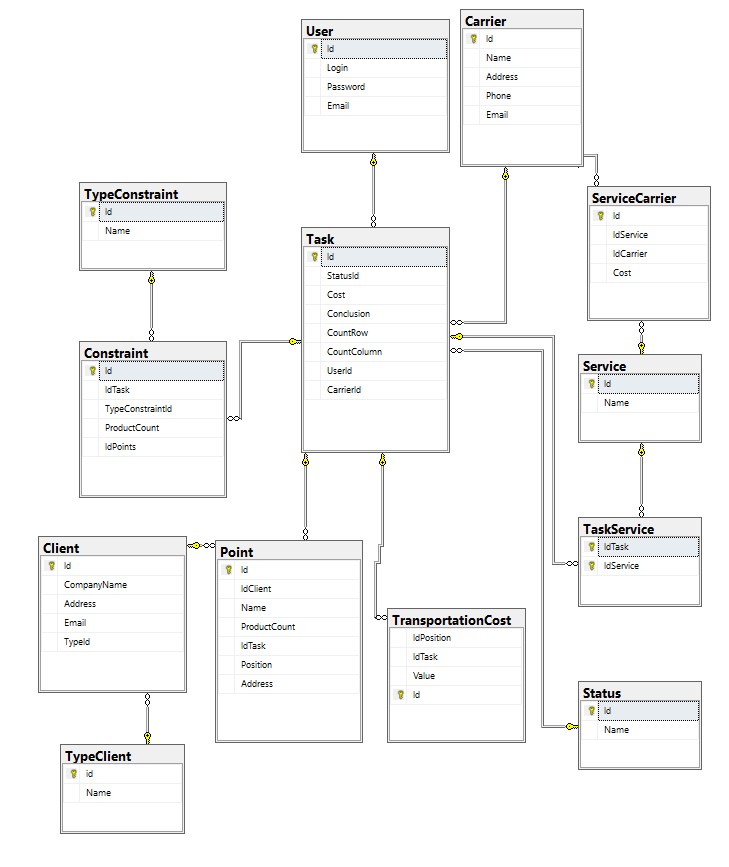


Рисунок 29 – Диаграмма базы данных в SSMS

# ГЛАВА 3 ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

# 3.1 Руководство пользователя

…

# 3.2 Руководство программиста

…

# 3.3 Руководство по установке программного обеспечения

…

# ЗАКЛЮЧЕНИЕ

…

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Робсон Э., Фримен Э. Изучаем HTML, XHTML и CSS. Изд. Питер, 2019 – 720 с.
2. Макфарланд Д. Новая большая книга CSS. Изд. Питер, 2019 – 719 с.
3. Макфарланд Д. JavaScript и jQuery. Исчерпывающее руководство. Изд. Эксмо, 2016 – 880 с.
4. HtmlBook – блог-справочник по HTML5 [Электронный ресурс]: самоучитель HTML, справочник HTML, HTML5. URL: http://htmlbook.ru/html (дата обращения: 11.05.2022). Режим доступа: свободный.
5. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. Изд. Эксмо, 2020 – 480 с.
6. Крокфорд Д. Как устроен JavaScript. Изд. Питер, 2019 – 304 с.
7. Браун Э. Изучаем JavaScript. Руководство по созданию современных веб-сайтов. Изд. Вильямс, 2017 – 363 с.
8. Learn JavaScript – блог, содержащий информацию обо всех основах языка программирования JavaScript [Электронный ресурс]: современный учебник по изучению языка программирования JavaScript. URL: https://learn.javascript.ru/ (дата обращения: 11.05.2022). Режим доступа: свободный.
9. React документация [Электронный ресурс] URL: https://reactjs.org/docs/getting-started.html (дата обращения: 11.05.2022). Режим доступа: свободный.
10. React-query документация [Электронный ресурс] URL: <https://react-query.tanstack.com/overview> (дата обращения: 11.05.2022). Режим доступа: свободный.
11. Мартишин С., Симонов В., Храпченко М. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench. Изд. Форум, 2019 – 160 с.
12. Заяц А., Васильев Н. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript. Изд. Лань, 2021 – 120 с.
13. Гниденко, И. Г*.* Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2017. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/408655 (дата обращения: 11.05.2022). Режим доступа: для авториз. пользователей.
14. Черткова, Е. А.  Программная инженерия. Визуальное моделирование программных систем: учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2017. — 168 с. — (Профессиональное образование). — ISBN 978-5-534-04928-2. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/408222 (дата обращения: 11.05.2022). Режим доступа: для авториз. пользователей.
15. Трофимов, В. В.  Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва: Издательство Юрайт, 2018. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/422888 (дата обращения: 11.05.2022). Режим доступа: для авториз. пользователей.
16. Соколова, В. В.  Разработка мобильных приложений : учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2019. — 175 с. — (Профессиональное образование). — ISBN 978-5-534-10680-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/431172> (дата обращения: 11.05.2022). - Режим доступа: для авториз. пользователей.
17. Нестеров, С. А.  Базы данных: учебник и практикум для среднего профессионального образования / С. А. Нестеров. — Москва: Издательство Юрайт, 2019. — 230 с. — (Профессиональное образование). — ISBN 978-5-534-11629-8. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/445770 (дата обращения: 11.05.2022). Режим доступа: для авториз. пользователей.
18. Стружкин, Н. П.  Базы данных: проектирование: учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва: Издательство Юрайт, 2019. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/445776 (дата обращения: 11.05.2022). Режим доступа: для авториз. пользователей.
19. NextJS документация [Электронный ресурс] URL: https://nextjs.org/docs/getting-started (дата обращения: 11.05.2022). Режим доступа: свободный.
20. Next-Auth документация [Электронный ресурс] URL: https://next-auth.js.org/ (дата обращения: 11.05.2022). Режим доступа: свободный.

# ПРИЛОЖЕНИЕ А

**Листинг программы**

# ПРИЛОЖЕНИЕ Б

**Словарь данных**

# ПРИЛОЖЕНИЕ В

**Презентация**