МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Гродненский государственный университет имени Янки Купалы»

Факультет математики и информатики

**Кафедра современных технологий программирования**

АЙРАПЕТОВ ЮЛИАН ЮРЬЕВИЧ

**Разработка веб-приложения для организации работы малого транспортного предприятия**

Курсовая работа

по дисциплине «Базы данных»

студента 4 курса специальности

1-40 01 01 «Программное обеспечение информационных технологий» дневной формы получения образования

Научный руководитель

Родченко Вадим Григорьевич, Преподаватель кафедры современных технологий программирования

Гродно, 2022

**РЕЗЮМЕ**

Курсовая работа – «Разработка веб-приложения для организации работы малого транспортного предприятия», 29 страниц, 4 иллюстраций, 9 использованных источников.

Ключевые слова – веб-приложение, транспортное предприятие, Ruby, RubyMine, Ruby on Rails, база данных, PgSQL, API.

Цель исследования – проектирование базы данных к веб-приложению для организации работы малого транспортного предприятия.

Объект исследования – база данных для веб-приложения, а также API, Ruby on Rails, PgSQL.

Предмет исследования – технологии и программные средства, используемые для создания баз данных.

Подобно большинству приложений, реализация которых проходит, с использованием предметной области, в которой нет опыта разработки, одной из важных задач проекта является получение знаний о предметной области, выявление особенностей, и сложностей во время разработки приложения и работе с этой предметной областью.

Целью данной курсовой работы – разработка и реализация backend части веб-приложения, которая будет основана на технологии Rails API, и с помощью которой будет происходить взаимодействие с реляционной базой данных PgSQL.

Для достижения поставленной цели требуется решить следующие основные задачи:

1. Изучение создания баз данных.
2. Изучение существующих решений для взаимодействия базы данных и приложения.
3. Поиск актуального решения.
4. Реализация работы с выбранным решением.

**SUMMARY**

Course work - "Development of a web application for organizing the work of a small transport enterprise", 29 pages, 4 illustrations, 9 references.

Keywords – web application, transport enterprise, Ruby, RubyMine, Ruby on Rails, database, PgSQL, API.

The purpose of the study is to design a database for a web application for organizing the work of a small transport enterprise.

The object of study is a database for a web application, as well as API, Ruby on Rails, PgSQL.

The subject of the study is the technologies and software used to create databases.

Research methods - study and analysis of literature on the creation of applications that work with client applications for the provision of services, comparative analysis, analogy.

Like most applications that are implemented using a subject area in which there is no development experience, one of the important tasks of the project is to gain knowledge about the subject area, identify features, and difficulties during application development and work with this subject area. As part of the development of this project, the language Ruby was chosen, and the development environment RubyMine.

The purpose of this course work is the development and implementation of the backend part of a web application, which will be based on the Rails API technology, and with the help of which interaction with the PgSQL relational database will take place.

To achieve this goal, it is necessary to solve the following main tasks:

1. Learning how to create web applications.

2. Study of existing solutions for creating web applications.

3. Search for an up-to-date solution.

4. Implementation of work with the selected solution.

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 4](#_Toc121790730)

[ВВЕДЕНИЕ 6](#_Toc121790731)

[ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc121790732)

[1.1 Описание предметной области 8](#_Toc121790733)

[1.2 Основные требования к серверной части 9](#_Toc121790734)

[1.3 Вывод по главе 1 10](#_Toc121790735)

[ГЛАВА 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОРГАНИЗАЦИИ РАБОТЫ МАЛОГО ТРАНСПОРТНОГО ПРЕДПРИЯТИЯ 11](#_Toc121790736)

[2.1 Проектирование инфологической модели данных 11](#_Toc121790737)

[2.1 .1 Построение Entity-Relationship модели 11](#_Toc121790738)

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

|  |  |  |
| --- | --- | --- |
| СУБД | Система управления  базами данных | - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных. |
| API | Application Programming Interface | - описание способов взаимодействия одной компьютерной программы с другими. |
| CRUD | Create, Read, Update, Delete | - акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание, чтение, модификация, удаление. |
| RoR | Ruby on Rails | - фреймворк, написанный на языке программирования Ruby, реализует архитектурный шаблон Model-View-Controller. |

# ВВЕДЕНИЕ

В современном мире сложно переоценить значение автомобильных грузоперевозок. Данная услуга востребована не только в бизнесе разного масштаба, но и у частных лиц, различие лишь в объемах перевозимых грузов и расстояний.

На сегодняшний день автомобильные перевозки являются наиболее популярным и перспективным видом транспорта. Все это обусловлено тем, что при доставке груза автомобилем можно легко скорректировать маршрут, есть возможность дополнительной выгрузки и погрузки по пути. Также, в некоторые регионы нашей страны доставить грузы можно только по автомобильным дорогам, что является дополнительным преимуществом данного вида грузоперевозок.

Рынок грузоперевозок, несмотря на все свои проблемы, активно развивается. Темпы развития транспортных компаний увеличиваются с каждым годом, появляются новые лица и организации, и в связи с этим, в условиях упорной борьбы за клиентов, повышается конкуренция. Чтобы оставаться конкурентоспособным в таких условиях очень сложно и поэтому создание сайта в Интернете является правильным путем развития и создание имиджа организации.

Наиболее популярным для этой отрасли направлением в области информационных технологий является сайт. Любая успешная компания давно оценила важность и актуальность размещения в сети информации о себе. Сайт позволит привлечь новых клиентов, обеспечить дополнительные объемы прибыли, вывести бизнес на новый уровень.

Для реализации всех этих возможностей сайт также должен отвечать многим требованиям, иметь при себе привлекательный и удобный интерфейс, грамотную рекламу, доступный и полезный функционал для потенциальных заказчиков, информацию, в доступном для всех виде. Сайт должен своевременно обновляться и отвечать всем необходимым современным требованиям, только тогда сайт будет приносить пользу, повышая имидж организации и привлекая все новых клиентов.

Таким образом, целью данной курсовой работы является подготовить проект реализующий базу данных веб-приложения для организации работы малого транспортного предприятия.

В ходе работы необходимо изучить разные виды баз данных. Определить наиболее функциональные и удобные для реализации, далее предстоит разработать алгоритмы, позволяющие применять это на практике.

В соответствии с объектом, предметом и целью исследования были поставлены следующие задачи:

1. Изучить архитектурные особенности базы данных PgSQL для реализации исходных требований;
2. Спроектировать базу данных;
3. Реализовать базу данных;
4. Создать доступ к базу с помощью API.

# ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

# Описание предметной области

Целью данной курсовой работы является подготовить базу данных веб-приложения для организации работы малого транспортного предприятия, предполагающий возможность оформления заявки на оказание услуг. База данных проекта должны содержать информацию о услугах, предоставляемых предприятием, о машинах, имеющихся в автопарке, о заказах, совершенных клиентами и о других сопутствующих данных. Поскольку некоторая информация должна вноситься в базу администратором, следует хранить данные о пользователях и их ролях. Так как перевозка осуществляется от начального пункта к конечному, следует хранить данных этих точек.

Каждая машина определяется следующими параметрами:

* + брэнд;
  + модель;
  + фото;
  + вместимость, которая содержит:
    - длину;
    - высоту;
    - ширину;
    - количество поддонов;

Одна и та же вместимость может быть характерна для нескольких машин. Также каждая машина может относиться к нескольким заказам.

Для каждого заказа должна быть зафиксирована информация, связанная с:

* + телефоном заказчика;
  + этапом выполнения;
  + машиной;
  + маршрутом, содержащим начальную и конечную точки;
  + пользователем;
  + типом груза;
  + типом доставки;
  + дополнительными услугами.

Для расчета стоимости услуги следует учесть длину маршрута, для этого каждый маршрут хранит начальную и конечную точки. Точки в свою очередь содержат следующую информацию:

* + широта;
  + долгота;
  + адрес;

О каждом пользователе, который зарегистрировался на сайте, имеются

следующие данные:

* + - * + email;
        + пароль;
        + роль;

Кроме того, каждый пользователь имеет связь с аккаунтом, который содержит имя пользователя и телефон.

# Основные требования к серверной части

Основной задачей для построения приложения транспортной компании является реализация полноценной базы данных и функционала, позволяющего работать с ней. Приложение должно выполнять ряд следующих задач

1. **Важные задачи** (без которых программа не имеет смысла):

- Добавление информации о машинах и их характеристиках в базу данных: бренд, модель, длина, ширина, высота и кол-во поддонов.

- Добавление типов груза и доставок, а также дополнительных услуг в базу данных: название и цена.

- Добавление координат и составление маршрутов.

- Добавление заказов в базу данных: телефон, стадия.

2) **Существенные задачи** (если программа не будет их удовлетворять, то она все равно будет выполнять главные функции):

- Выбор точек с помощью интерактивной карты

- Скидочно-накопительная система

# Вывод по главе 1

В первой главе была описана предметная область, были определены параметры для разных объектов и ограничения, которые будут на них накладываться. Также были упомянуты основные задачи, которые должны быть реализованы в системе.

Условно, данные можно разбить на 3 части:

* пользователи;
* заявки;
* услуги;

Основными функциями программы являются следующие:

* создание заявок;
* просмотр заявок;
* просмотр услуг;
* редактирование личной информации;
* модерирование системы;

# ГЛАВА 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОРГАНИЗАЦИИ РАБОТЫ МАЛОГО ТРАНСПОРТНОГО ПРЕДПРИЯТИЯ

# Проектирование инфологической модели данных

Концептуальное (инфологическое) проектирование – это построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создается без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области», поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности [5].

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Чаще всего концептуальная модель базы данных включает в себя:

1. Описание информационных объектов или понятий предметной области и связей между ними.
2. Описание ограничений целостности, т.е. требований к допустимым значениям данных и к связям между ними.

# 2.1.1 Построение Entity-Relationship модели

Модель «сущность-связь» или ER-модель, предложенная П. Ченом в 1976 г., является наиболее известным представителем класса семантических (концептуальных, инфологических) моделей предметной области. ER-модель обычно представляется в графической форме, с использованием оригинальной нотации П. Чена, называемой ER-диаграмма, либо с использованием других графических нотаций.

Основные преимущества ER-моделей:

* Наглядность;
* модели позволяют проектировать базы данных с большим количеством объектов и атрибутов;
* ER-модели реализованы во многих системах автоматизированного проектирования баз данных (например, ERWin).

Основные элементы ER-моделей:

* Объекты (сущности);
* атрибуты объектов;
* связи между объектами.

Сущность — объект предметной области, имеющий атрибуты.

Класс может быть обязательным и необязательным. Если каждый экземпляр сущности участвует в связи, то класс принадлежности — обязательный, иначе — необязательный.

# 2.1.2 Построение концептуальной модели в нотации Чена

Рассматривая последовательно ключевые характеристики каждого из множеств, можно определить соответствующие атрибуты для каждого из рассмотренных объектных множеств.

Для объектного множества USERS из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* email – уникальный однозначный атрибут
* password - однозначный атрибут

Для объектного множества ROLES из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* name – уникальный однозначный атрибут

Для объектного множества PROFILES из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* name – однозначный атрибут
* phone – однозначный атрибут

Для объектного множества ORDERS из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* phone – однозначный атрибут
* stage – однозначный атрибут

Для объектного множества ROUTES из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)

Для объектного множества POINTS из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* latitude – однозначный атрибут
* longitude – однозначный атрибут
* address – однозначный атрибут

Для объектного множества CARGO TYPES из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* name – однозначный атрибут

Для объектного множества DELIVERY TYPES из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* name – однозначный атрибут
* price – однозначный атрибут

Для объектного множества ADDITIONAL SERVICES из описания предметной области выделяются следующие атрибуты:

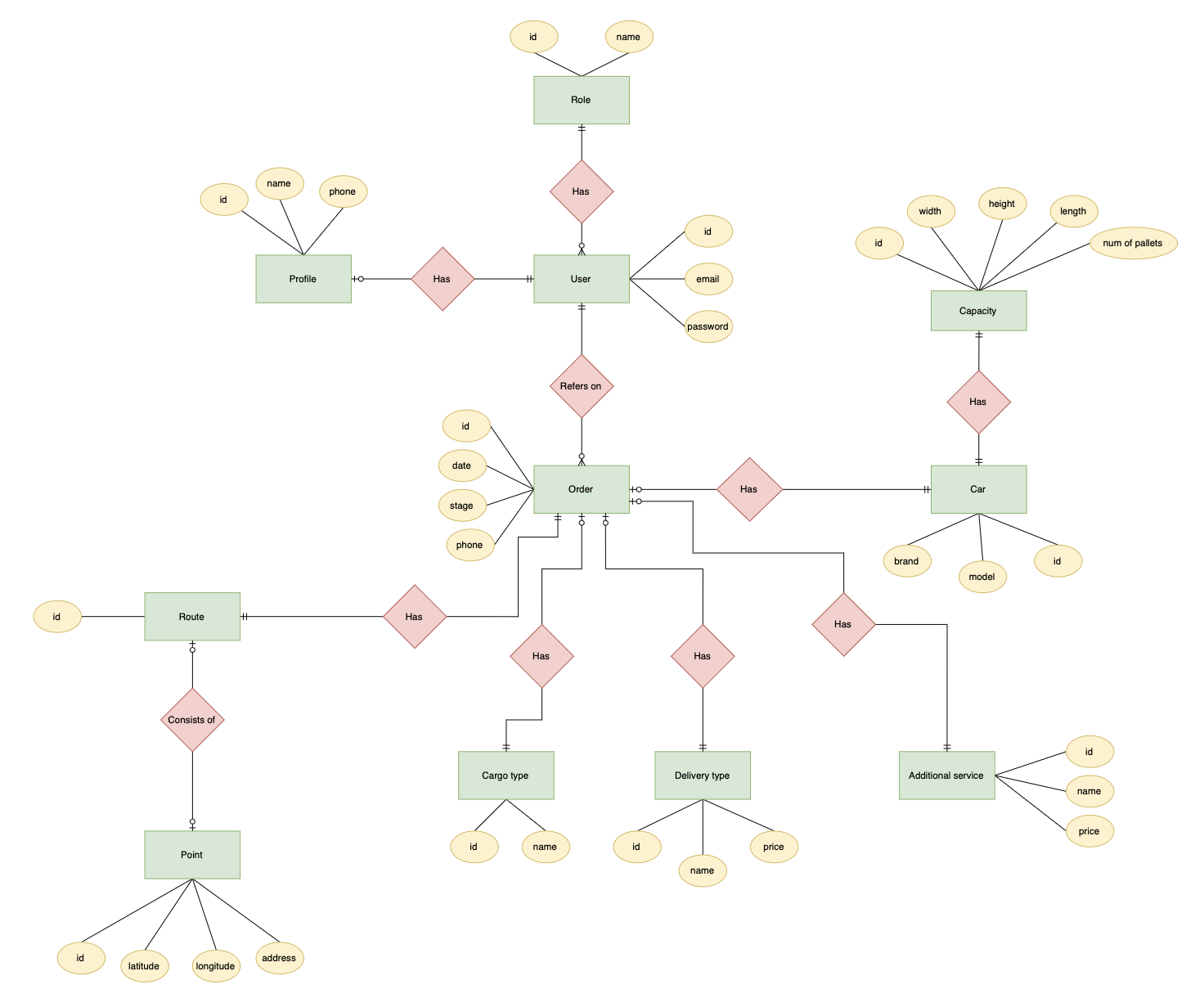
* id – уникальный идентификатор (первичный ключ)
* name– однозначный атрибут
* price – однозначный атрибут

Для объектного множества CARS из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* brand – однозначный атрибут
* model – однозначный атрибут

Для объектного множества Capacity из описания предметной области выделяются следующие атрибуты:

* id – уникальный идентификатор (первичный ключ)
* width – однозначный атрибут
* height – однозначный атрибут
* length – однозначный атрибут
* num of pallets – однозначный атрибут

В результате, с учетом анализа предметной области и соответствующих выводов, можно построить первый вариант концептуальной модели для базы данных. (Рис 2.1.2.1)

**Рисунок 2.1.2.1 –**  Концептуальная модель в нотации Чена системы «Малое транспортное предприятие»

* 1. **Модель функций системы**
     1. **Диаграмма вариантов использования**

Use Case — это сценарная техника описания взаимодействия. С помощью Use Case может быть описано и пользовательское требование, и требование к взаимодействию систем, и описание взаимодействия людей и компаний в реальной жизни. В разработке ПО эту технику часто применяют для проектирования и описания взаимодействия пользователя и системы, поэтому название Use Case часто воспринимает как синоним требования человека-пользователя к решению определенной задачи в системе.

Для общего представления работы системы была построена use-case диаграмма. Данная диаграмма создавалась постепенно. Сначала была выбрана главные роли в системе: гость, пользователь, модератор и администратор. Исходя из этого в use-case диаграмму были добавлены функции, которые использует пользователь в зависимости от его роли. При помощи use-case диаграммы в простом виде можно показать действия участника и функционала системы в целом.[4] (Рис. 2.2.1.1)

**Рисунок 2.2.1.1** – Use-case диаграмма системы «Малое транспортное предприятие»

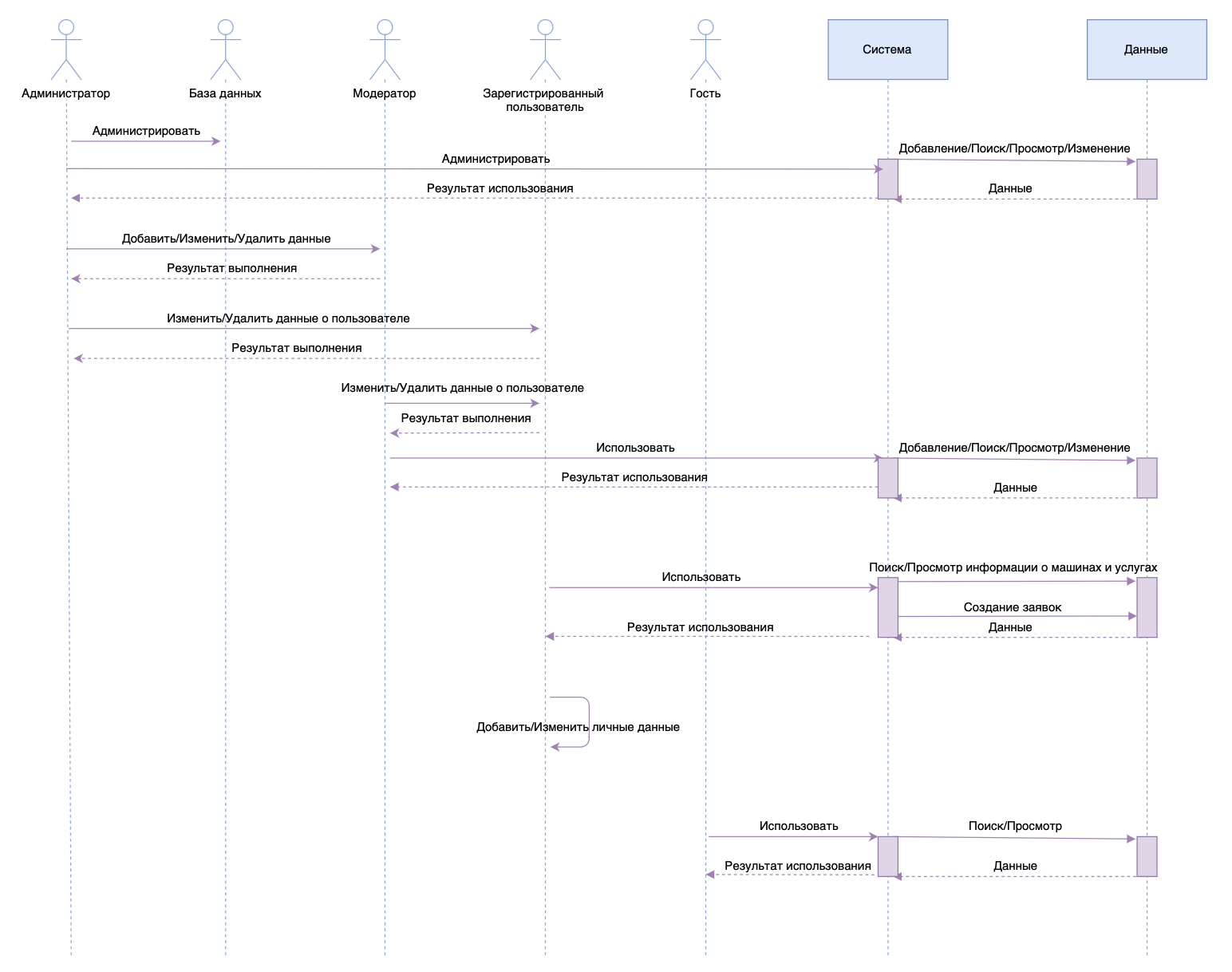
Примечание – Источник: собственная разработка

На диаграмме использования отображаются:

* Гость, пользователь, модератор, администратор – лица, которое взаимодействуют с нашей системой;
* Варианты использования — функции, которые наша система предоставляет:
  + «Просмотр информации о машинах»
  + «Просмотр о услугах и ценах» (позволяет просмотреть услуги, которые предоставляет компания)
  + «Просмотр контактной информации»
  + «Создание заявок на перевозку»
  + «Заполнение и редактирвоание профиля»
  + «Добавление машин и их характеристик»
  + «Добавление дополнительных услуг, новых типов перевозок и грузов»
  + «Подтверждение/отклонение заявок»
  + «Администрирование базы данных»
    1. **Диаграмма последовательности**

Диаграмма последовательности — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента.

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни», отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

Была создана диаграммы последовательности приложения для организации работы малого транспортного предприятия (Рис.2.2.2.1) Эта диаграммы наглядно демонстрируют жизненный цикл определенных актеров и объектов системы, взаимодействие их между собой. В нашем примере актерами является гость, администратор, основные объекты - приложение и база данных.[3]

**Рисунок 2.2.2.1** – Диаграмма последовательности «Малое транспортное предприятие»

Примечание – Источник: собственная разработка

Описание последовательности:

1. Администратор администрирует базу данных
2. Взаимодействие администратора с системой происходит через добавление, поиск, просмотр и изменения каких-либо данных. Запросы администратора отправляются в систему, которая производит выборку данных и результат возвращается администратору.
3. Администратор может взаимодействовать с модератором путем назначения изменения и удаления роли.
4. Администратор имеет возможность изменить и удалить данные пользователя.
5. Как и админ Модератор удаляет и изменяет личные данные пользователей.
6. Взаимодействие модератора с системой происходит через добавление, поиск, просмотр и изменения каких-либо данных. Запросы администратора отправляются в систему, которая производит выборку данных и результат возвращается администратору.
7. Зарегистрированный пользователь имеет доступ к своим личным данным с возможностью добавления и изменения.
8. Взаимодействие зарегистрированного пользователя с системой происходит через поиск и просмотр информации о машинах и услугах, а также через создания заявок на перевозку.
9. Гость взаимодействует с системой через поиск и просмотр информации о услугах и их ценах, а также о машинах.

**2.3 Выводы по главе 2**

Во второй главе были определены функциональные требования к приложению для организации работы малого транспортного предприятия. При проектировании решения были выделены основные функции, которые она должно выполнять. Было рассмотрено как пользователь сможет взаимодействовать с программой, что программа может выполнять в ответ на действие пользователя. А также были спроектированы следующие диаграммы для дальнейшего создания базы данных:

- Диаграмма вариантов использования;

- Диаграмма активностей;

- Диаграмма последовательности;

- Концептуальная модель в нотации Чена.

После проведения данных действий можно приступить к созданию базы данных.

**ГЛАВА 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОРГАНИЗАЦИИ РАБОТЫ МАЛОГО ТРАНСПОРТНОГО ПРЕДПРИЯТИЯ**

**3.1 Физическая модель данных**

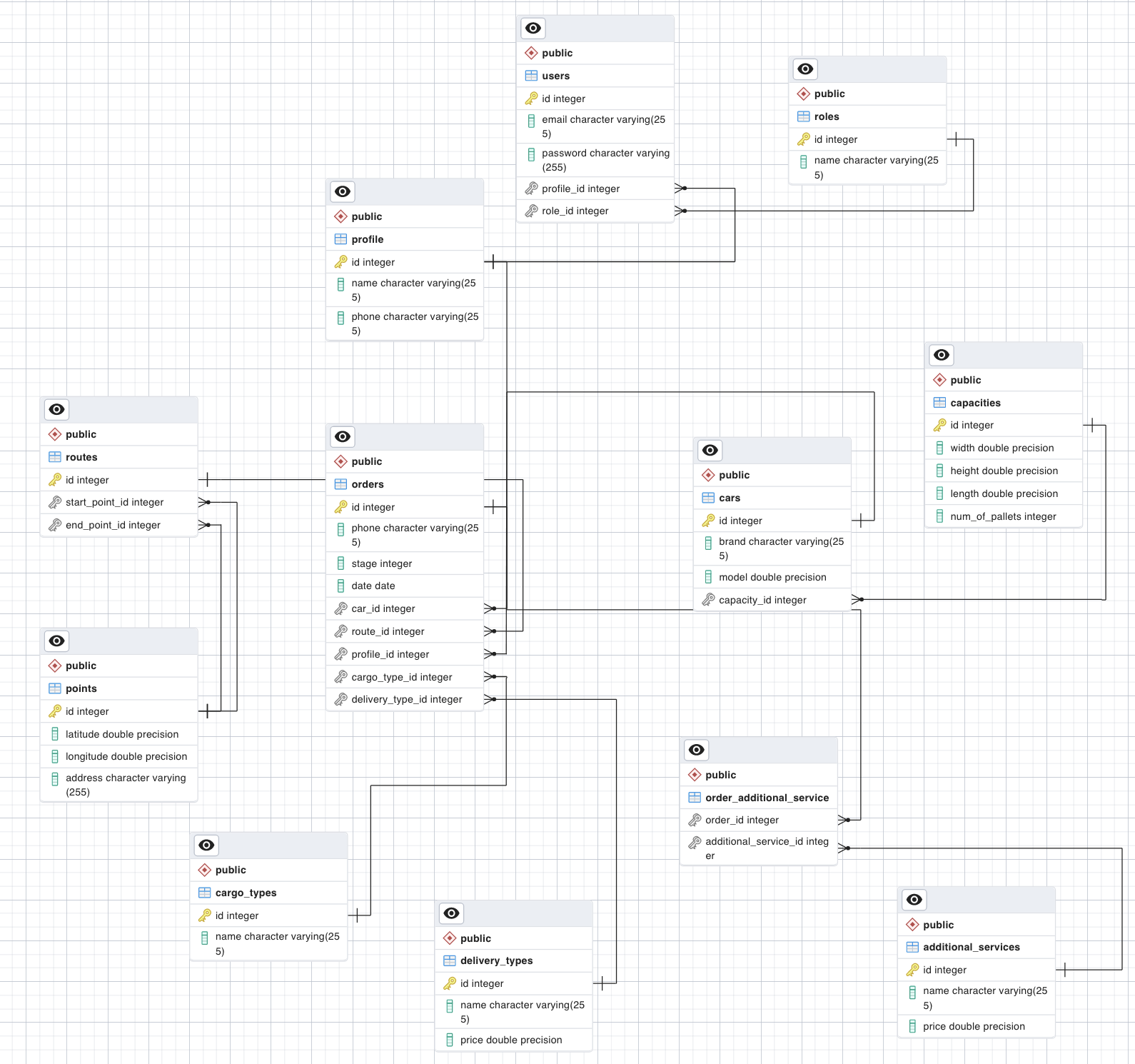
Физическая модель данных сгенерированная, с помощью PgAdmin - клиента для базы данных PgSQL, представлена на рисунке ниже (рисунок 3.1) [9]:

Рисунок 3.1. Физическая модель БД.

Для создания данной базы данных были написаны соответствующие SQL-скрипты, представленные далее.

Первым делом нам необходимо создать таблицу данных c машинам, а также таблицу с характеристиками, чтобы гость мог просматривать информацию на сайте. Таблица данных для характеристик машины имеет ширину, высоту, длину и кол-во поддонов:

*CREATE TABLE Capacities(*

*ID int NOT NULL,*

*width float NOT NULL,*

*height float NOT NULL,*

*length float NOT NULL,*

*num\_of\_pallets int NOT NULL,*

*PRIMARY KEY (ID)*

*);*

Далее создадим таблицу для самих машин, которая будет содержать бренд и модель, а также внешний ключ к предыдущей таблице:

*CREATE TABLE Cars(*

*ID int NOT NULL,*

*brand varchar(255) NOT NULL,*

*model float NOT NULL,*

*capacity\_id int NOT NULL,*

*PRIMARY KEY (ID),*

*FOREIGN KEY (capacity\_id) REFERENCES Capacities(ID)*

*);*

Создадим таблицы для хранения типов груза, типов доставки, а также дополнительных услуг:

*CREATE TABLE Cargo\_types(*

*ID int NOT NULL,*

*name varchar(255) NOT NULL,*

*PRIMARY KEY (ID)*

*);*

*CREATE TABLE Delivery\_types(*

*ID int NOT NULL,*

*name varchar(255) NOT NULL,*

*price float NOT NULL,*

*PRIMARY KEY (ID)*

*);*

*CREATE TABLE Additional\_services(*

*ID int NOT NULL,*

*name varchar(255) NOT NULL,*

*price float NOT NULL,*

*PRIMARY KEY (ID)*

*);*

Для реализации дальнейшего функционала нужно предварительно создать таблицы для регистрации пользователей. Начнём с таблицы для ролей, которая содержит имя:

*CREATE TABLE Roles(*

*ID int NOT NULL,*

*name varchar(255) NOT NULL,*

*PRIMARY KEY (ID)*

*);*

Далее создаем таблицу профиля, которая содержит имя и телефон будущего пользователя:

*CREATE TABLE Profile(*

*ID int NOT NULL,*

*name varchar(255) NOT NULL,*

*phone varchar(255) NOT NULL,*

*PRIMARY KEY (ID)*

*);*

Одной из основных таблиц в предложенной базе данных является пользователь, которы содержит email и пароль для аутентификации в системе, а также внешний ключ на свой профиль:

*CREATE TABLE Users(*

*ID int NOT NULL,*

*email varchar(255) NOT NULL,*

*password varchar(255) NOT NULL,*

*profile\_id int NOT NULL,*

*role\_id int NOT NULL,*

*PRIMARY KEY (ID),*

*FOREIGN KEY (profile\_id) REFERENCES Profile(ID),*

*FOREIGN KEY (role\_id) REFERENCES Roles(ID)*

*);*

Весь функционал базы данных сосредотачивается на создании заказов, поэтому следующим шагом создадим таблицы, ссылки на которые будут храниться в таблице с заказами.

Создадим таблицу с координатными точками, которые содержат широту, долготу и адрес:

*CREATE TABLE Points(*

*ID int NOT NULL,*

*latitude float NOT NULL,*

*longitude float NOT NULL,*

*address varchar(255) NOT NULL,*

*PRIMARY KEY (ID)*

*);*

Заказ выстраивается по определенному маршруту, состоящему из начальной и конечной точки:

*CREATE TABLE Routes(*

*ID int NOT NULL,*

*start\_point\_id int NOT NULL,*

*end\_point\_id int NOT NULL,*

*PRIMARY KEY (ID),*

*FOREIGN KEY (start\_point\_id) REFERENCES Points(ID),*

*FOREIGN KEY (end\_point\_id) REFERENCES Points(ID)*

*);*

Таблица с заказами одна из основных в описываемой базе данных. Каждый заказ состоит из стадии выполнения, телефона заказчика и даты выполнения. Также в этой таблице присутствуют внешние ключи к таблицам профиля, типа доставки, типа груза, машины и маршрута.

*CREATE TABLE Orders(*

*ID int NOT NULL,*

*phone varchar(255) NOT NULL,*

*stage int NOT NULL,*

*date date NOT NULL,*

*car\_id int NOT NULL,*

*route\_id int NOT NULL,*

*profile\_id int NOT NULL,*

*cargo\_type\_id int NOT NULL,*

*delivery\_type\_id int NOT NULL,*

*PRIMARY KEY (ID),*

*FOREIGN KEY (car\_id) REFERENCES Cars(ID),*

*FOREIGN KEY (route\_id) REFERENCES Routes(ID),*

*FOREIGN KEY (profile\_id) REFERENCES Profile(ID),*

*FOREIGN KEY (cargo\_type\_id) REFERENCES Cargo\_types(ID),*

*FOREIGN KEY (delivery\_type\_id) REFERENCES Delivery\_types(ID)*

*);*

Каждый заказ может содержать несколько дополнительных услуг, создадим промежуточную таблицу для реализации связи многое-ко-многим:

*CREATE TABLE Order\_Additional\_service(*

*order\_id int NOT NULL,*

*additional\_service\_id int NOT NULL,*

*FOREIGN KEY (order\_id) REFERENCES Orders(ID),*

*FOREIGN KEY (additional\_service\_id) REFERENCES Additional\_services(ID)*

*);*

**3.2 Общая архитектура серверного приложения**

Backend часть приложения будет реализована на языке программирования Ruby с помощью фреймворка Ruby on Rails.

Ruby on Rails – это универсальный фреймворк с открытым исходным кодом для Ruby.

Rails on Rails — это прежде всего среда разработки, которая великолепно подходит для создания любого типа веб-приложений: систем для управления веб-сайтами и платформ для ведения электронной торговли, программ для организации совместной работы и для веб-сервисов для осуществления коммуникации, для учетных и ERP-систем, статистических и аналитических систем.[2]

Приложение будут разрабатываться по технологии API.

Будут реализованы следующие слои приложения:

1. **Controllers**

Данный слой будет обрабатывать запросы клиента и отправлять ответы, при запросах на определенные методы, а также принимать данные, которые пользователь будет вводить. Все это будет приниматься в виде JSON. Контроллер в свою очередь парсит входную строку, разбивая на параметры для последующего использования. При отправки ответов будут использоваться representor’ы, которые помогут сформировать полный ответ в формате json.

1. **Services**

Данные с контроллеров передаются в сервисы. В сервисах будет происходить вся основная бизнес логика. Из контроллеров будут передаваться объекты, конвертироваться в Entity, которые будут направленны на базу данных с помощью ORM. Все основные действия будут проводиться над Entity объектами.

1. **Models**

На данном слое будет происходить обращение к базе данных. В данном приложении будут использоваться ORM, а именно ActiveRecord.

Каждый слой приложения, с технической точки зрения – отдельный модуль программы. Кроме того, в приложении будут различные вспомогательные классы, которые будут использоваться во всех слоях. Они будут находиться в отдельном модуле – helpers.[8]

В качестве инструмента для хранения данных приложения была выбрана БД, для управления которой была выбрана наиболее быстрая и лёгкая в использовании СУБД PgSQL.[6]

**3.3 Выводы по главе 3**

В данной главе была разобрана физическая модель базы данных, процесс её создания с подробным описанием и общая архитектура серверной части приложения. Были разобраны основные репозитории, а также краткое описание бизнес логики в сервисах и контроллеры. Кроме того был обоснован выбор средств разработки.

Основная цель была достигнута. Помимо этого, проект готов к дополнительным возможным расширениям и дополнениям, что, несомненно, является плюсом.

**ЗАКЛЮЧЕНИЕ**

 Была описана предметная область, были определены параметры для разных объектов и ограничения, которые будут на них накладываться.

Условно, данные можно разбить на 4 части:

* пользователи;
* заказы;
* машины и характеристики;
* услуги.

Основными функциями программы являются следующие:

* создание заказов;
* просмотр услуг и цен;
* просмотр машин и характеристик;
* редактирование личной информации;
* модерирование системы;

В ходе выполнения работы по проектированию приложения было рассмотрено само понятие реализуемого приложения вместе с его основным функционалом. На основе этого были выведены основные функции веб-приложения «Малое транспортное предприятие»:

Для админа:

* Администрирование БД;
* Добавление/редактирование/удаление данных (о машинах и их характеристиках, о типах доставок и грузов, о дом услугах);

Для пользователя:

* Просмотр информации;
* Создание заказов;

Для гостя:

* Просмотр информации;

А также рассмотрена структура проекта, были построены соответствующие диаграммы: диаграмма вариантов использования, диаграмма последовательности, а также концептуальная модель данных, которые в полной мере описывают всю суть разрабатываемого приложения.

Также была разобрана физическая модель базы данных, процесс её создания с подробным описанием и общая архитектура серверной части приложения, а также краткое описание бизнес логики в сервисах и контроллеры.

**CПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. http://rusrails.ru/ – Дата доступа: 20.11.2022
2. https://ru.wikipedia.org/wiki/Ruby\_on\_Rails – Дата доступа: 23.11.2022.
3. https://ru.wikipedia.org/wiki/Диаграмма\_последовательности – Дата доступа: 25.11.2022
4. https://systems.education/use-case – Дата доступа: 30.11.2022
5. <https://app.creately.com/> создание диаграмм – Дата доступа: 06.12.2022
6. https://www.postgresql.org – Дата доступа: 04.12.2022
7. https://ru.wikipedia.org/wiki/База\_данных – Дата доступа: 06.12.2022
8. https://api.rubyonrails.org – Дата доступа: 08.12.2022
9. https://www.pgadmin.org/docs/pgadmin4/development/erd\_tool.html – Дата доступа: 11.12.2022