Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Модели данных и системы управления базами данных»

«К ЗАЩИТЕ ДОПУСТИТЬ»
Руководитель курсового проекта
ассистент
_____А.В. Давыдчик
___.____.2023

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему:
«ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ЛОГИСТИКИ ГРУЗОПЕРЕВОЗОК»

БГУИР КП 1-40 04 01 007 ПЗ

Выполнил студент группы 053504
Долгих Дмитрий Владимирович
_____
(подпись студента)
Курсовой проект представлен на проверку ___.____.2023
_____
(подпись студента)

Минск 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

С развитием современных информационных технологий и цифровизации бизнес-процессов стало очевидным, что индустрия логистики грузоперевозок нуждается в эффективных решениях, способных оптимизировать и автоматизировать множество операций. В этом контексте веб-приложения стали неотъемлемой частью транспортно-логистической отрасли, предоставляя компаниям их участникам инструменты для более эффективного управления грузами, транспортом, складами и другими компонентами логистической цепи.

База данных является неотъемлемой частью любого веб-приложения, предназначенного для управления и оптимизации логистики грузоперевозок. Важно подчеркнуть роль базы данных и преимущества ее использования:

1 Централизация и хранение информации

База данных позволяет хранить всю необходимую информацию, связанную с грузоперевозками, в одном месте. Это включает в себя данные о клиентах, грузах, транспорте, складах, маршрутах и других ключевых параметрах логистических операций. Централизованное хранение данных упрощает доступ к ним и обеспечивает их целостность.

2 Улучшенная прозрачность и мониторинг

Благодаря базе данных логистические компании могут отслеживать статус грузов, перемещение транспорта, историю заказов и другие данные в режиме реального времени. Это позволяет оперативно реагировать на изменения в процессе и уведомлять клиентов о статусе их грузов, повышая прозрачность и доверие.

3 Автоматизация и оптимизация процессов

База данных позволяет внедрить автоматизацию в ряд логистических операций, таких как определение оптимальных маршрутов, планирование доставок и управление запасами на складах. Автоматические оповещения и сигналы также могут быть настроены для уведомления о нештатных ситуациях.

4 Улучшение аналитики и принятия решений

База данных предоставляет ценные данные для анализа производительности и принятия стратегических решений. Собранные в базе данных данные могут быть использованы для выявления тенденций, определения эффективности процессов и разработки стратегий для сокращения издержек и оптимизации ресурсов.

5 Обеспечение безопасности данных

Базы данных позволяют обеспечить безопасное хранение и доступ к информации. Механизмы аутентификации и авторизации позволяют установить права доступа к данным, обеспечивая конфиденциальность и целостность информации.

Цель данной курсовой работы заключается в проектировании и оптимизации базы данных для веб-приложения по логистике грузоперевозок. В ходе исследования будут рассмотрены ключевые аспекты, связанные с созданием и внедрением веб-приложений в сфере логистики, а также особенности проектирования базы данных, которая играет критическую роль в обеспечении эффективной работы системы.

В итоге, база данных является фундаментальным элементом веб-приложения для логистики грузоперевозок, обеспечивая его эффективную работу и предоставляя множество преимуществ, включая улучшенную прозрачность, автоматизацию, аналитику и безопасность данных. В рамках данной курсовой работы будет уделено внимание исследованию методологий проектирования и оптимизации такой базы данных, а также ее роли в обеспечении эффективности веб-приложения для логистики грузоперевозок.

1 МОДЕЛИРОВАНИЕ ДАННЫХ

Моделирование данных – это процесс создания визуального представления или чертежа, определяющего системы сбора и управления информацией в любой организации. Этот план или модель данных помогает различным заинтересованным сторонам, таким как аналитики данных, ученые и инженеры, создать единое представление о данных организации. Модель описывает, какие данные собирает компания, взаимосвязь между различными наборами данных и методы, которые будут использоваться для хранения и анализа данных.

Модели данных строятся на основе бизнес-потребностей. Правила и требования к модели данных определяются заранее на основе обратной связи с бизнесом, поэтому их можно включить в разработку новой системы или адаптировать к существующей.

1.1 Типы моделей данных

Разработка баз данных и информационных систем начинается с высокого уровня абстракции и с каждым шагом становится все точнее и конкретнее. В зависимости от степени абстракции модели данных можно разделить на три категории. Процесс начинается с концептуальной модели, переходит к логической модели и завершается физической моделью.

Концептуальные модели данных, которые также называются моделями предметной области, описывают общую картину: что будет содержать система, как она будет организована и какие бизнес-правила будут задействованы. Концептуальные модели обычно создаются в процессе сбора исходных требований к проекту. Как правило, они включают классы сущностей (вещи, которые бизнесу важно представить в модели данных), их характеристики и ограничения, отношения между сущностями, требования к безопасности и целостности данных. Любые обозначения обычно просты.

Заинтересованные стороны и аналитики обычно создают концептуальную модель. Это простое диаграммное представление, которое не следует формальным правилам моделирования данных. Важно то, что оно помогает как техническим, так и нетехническим заинтересованным сторонам разделить общее видение и согласовать цель, объем и дизайн проекта по работе с данными. Концептуальные модели выступают в качестве моста между бизнес-правилами и лежащей в их основе физической системой управления базами данных (СУБД).

Логические модели данных отображают концептуальные классы данных на технических структурах данных. Они дают более подробную информацию о концепциях данных и их сложных отношениях, которые были определены в концептуальной модели данных, например:

- типы данных различных атрибутов (например, строка или число)
- взаимосвязи между объектами данных
- первичные атрибуты или ключевые поля в данных

Архитекторы данных и аналитики совместно работают над созданием логической модели. Они следуют одной из нескольких формальных систем моделирования данных для создания представления. Иногда гибкие команды могут пропустить этот шаг и перейти от концептуальных к физическим моделям напрямую. Однако эти модели полезны для проектирования больших баз данных, называемых хранилищами данных, и для проектирования систем автоматической отчетности.

Логические модели служат связующим звеном между концептуальной моделью данных и базовой технологией и языком баз данных, которые разработчики используют для создания базы данных. Однако они не зависят от технологии, и вы можете реализовать их на любом языке баз данных. Инженеры по данным и заинтересованные стороны обычно принимают технологические решения после создания логической модели данных. Логическая модель данных изображена на рисунке 2.

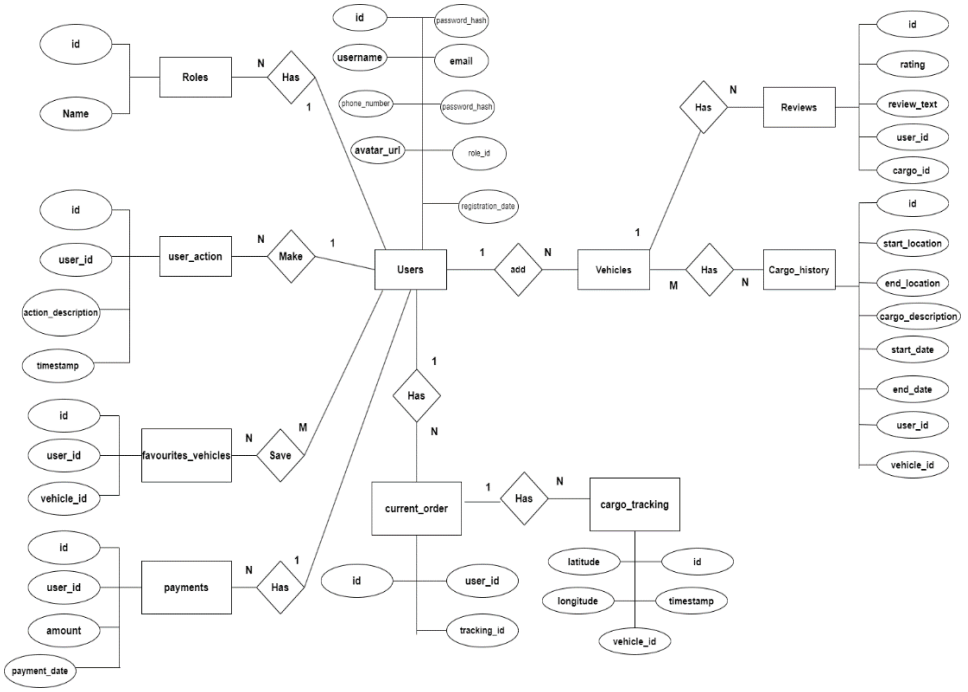

Рисунок 2 – Логическая модель

Физические модели данных представляют схему того, как данные будут храниться в базе. По сути, это наименее абстрактные из всех моделей. Они предлагают окончательный дизайн, который может быть реализован как реляционная база данных, включающая ассоциативные таблицы, которые иллюстрируют отношения между сущностями, а также первичные и внешние ключи для связи данных. Производительность базы данных, стратегия индексации, физическое хранилище и денормализация — важные параметры физической модели. Физическая модель данных выступает в качестве моста между логической моделью данных и конечной технологической реализацией. Физическая модель данных показана на рисунке 3.

Физическая модель данных зависит от конкретной системы управления базами данных, фактически являясь отображением системного каталога. В физической модели содержится информация о всех объектах базы данных. Поскольку стандартов на объекты базах данных не существует, например, нет стандарта на типы данных, физическая модель зависит от конкретной реализации систему управления базами данных. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т. д.

1.2 Методы моделирования данных

Методы моделирования данных – это различные методы, которые можно использовать для создания различных моделей данных. Эти подходы развивались с течением времени в результате инноваций в концепциях баз данных и управления данными. Далее представлены основные типы моделирования данных.

Иерархические модели данных представляют отношения «один ко многим» в древовидном формате. В модели этого типа каждая запись имеет единственный корень или родительский элемент, который сопоставляется с одной или несколькими дочерними таблицами. Эта модель была реализована в IBM Information Management System (IMS) в 1966 году и быстро нашла широкое применение, особенно в банковской сфере. Хотя этот подход менее эффективен, чем недавно разработанные модели баз данных, он все еще

используется в системах расширяемого языка разметки (XML) и географических информационных системах (ГИС).

При иерархическом моделировании данных можно представить отношения между различными элементами данных в древовидном формате. Иерархические модели данных представляют собой отношения «один ко многим», когда родительские или корневые классы данных отображаются на несколько дочерних.

Иерархическое моделирование данных со временем превратилось в графовое. Графовые модели данных представляют отношения данных, в которых сущности рассматриваются одинаково. Сущности могут связываться друг с другом отношениями «один ко многим» или «многие ко многим» без понятия «родительский» или «дочерний».

Реляционные модели данных были предложены исследователем IBM Э. Ф. Коддом в 1970 году. Они до сих пор встречаются во многих реляционных базах данных, обычно используемых в корпоративных вычислениях. Реляционное моделирование не требует детального понимания физических свойств используемого хранилища данных. В нем сегменты данных объединяются с помощью таблиц, что упрощает базу данных.

Моделирование данных «сущность-связь» (ER) использует формальные диаграммы для представления отношений между сущностями в базе данных. Архитекторы данных используют несколько инструментов ER-моделирования для представления данных.

В объектно-ориентированном программировании для хранения данных используются структуры данных, называемые объектами. Эти объекты данных являются программными абстракциями объектов реального мира. Например, в объектно-ориентированной модели данных автосалон будет иметь такие объекты данных, как «Клиенты», с такими атрибутами, как имя, адрес и номер телефона. Вы будете хранить данные о клиентах таким образом, чтобы каждый реальный клиент был представлен в виде объекта данных о клиенте.

Современные корпоративные вычисления используют технологию хранилищ данных для хранения больших объемов данных для аналитики. Вы можете использовать проекты многомерного моделирования данных для их высокоскоростного хранения и извлечения из хранилища. Многомерные модели используют дублирование или избыточные данные и отдают приоритет производительности, а не использованию меньшего пространства для хранения данных.

1.3 Системы управления базами данных

Система управления базами данных (СУБД) — это программное обеспечение, предназначенное для хранения, извлечения и управления данными. Наиболее распространенной СУБД в системе баз данных предприятия является СУРБД. Полная форма СУБД — это система управления реляционными базами данных.

Согласно реляционной модели Э. Ф. Кодда, СУБД позволяет пользователям создавать, обновлять, управлять реляционной базой данных и взаимодействовать с ней, позволяя хранить данные в табличной форме.

Система управления реляционными базами данных организует данные в таблицах, которые могут быть связаны внутри в зависимости от общих данных. Это позволяет пользователю легко получить одну или несколько таблиц с помощью всего одного запроса. С другой стороны, в плоском файле данные хранятся в единой табличной структуре, что менее эффективно и требует больше места и памяти.

Наиболее коммерчески доступной и общекорпоративной системой управления базами данных или системой управления реляционными базами данных, используемой сегодня, является язык структурированных запросов (база данных *SQL*) для доступа к базе данных. Другие широко используемые системы управления реляционными базами данных в компаниях включают базу данных *Oracle*, *MySQL*, *PostgreSQL* (реляционная база данных с открытым исходным кодом) и *Microsoft SQL Server*. Структуры РСУБД обычно используются для выполнения четырех основных операций: *CRUD* (создание, чтение, обновление и удаление), которые имеют решающее значение для поддержки согласованного управления данными.

2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Система управления базами данных: PostgreSQL

Система управления базами данных (СУБД) является неотъемлемой частью веб-приложения для логистики грузоперевозок. В данной курсовой работе была выбрана PostgreSQL как СУБД по ряду существенных преимуществ:

2.1.1 Открытый исходный код.

PostgreSQL является с открытым исходным кодом, что означает, что она доступна бесплатно и обладает активным сообществом разработчиков. Это обеспечивает постоянное обновление и улучшение системы, а также гарантирует ее стабильность и надежность.

2.1.2 Расширяемость и гибкость.

PostgreSQL предоставляет широкие возможности для расширения функциональности путем разработки собственных пользовательских функций, операторов и типов данных. Это позволяет легко адаптировать СУБД к конкретным потребностям проекта.

2.1.3. Поддержка геопространственных данных.

Важным аспектом в логистике грузоперевозок является работа с геопространственными данными. PostgreSQL предоставляет мощные инструменты для хранения, анализа и запросов геоданных, что делает ее идеальным выбором для логистического приложения.

2.1.4. Масштабируемость и производительность.

PostgreSQL спроектирована с учетом производительности и масштабируемости. Она может обрабатывать как небольшие проекты, так и высоконагруженные системы, что делает ее универсальной для различных масштабов приложений.

2.2 Среда разработки: Visual Studio Code (VSCode)

Среда разработки играет важную роль в создании веб-приложения. В данной курсовой работе была выбрана Visual Studio Code, так как она предоставляет множество преимуществ для разработчиков:

2.2.1 Открытый исходный код.

Visual Studio Code также является продуктом с открытым исходным кодом и доступен бесплатно. Его активное сообщество разработчиков

постоянно создает расширения и плагины, что позволяет настраивать среду под конкретные потребности проекта.

2.2.2 Множество расширений.

VSCode предлагает огромное количество расширений, которые упрощают разработку веб-приложений. Например, для работы с PostgreSQL, разработчики могут установить расширение для SQL, которое обеспечит подсветку синтаксиса, автодополнение и интеграцию с базой данных.

2.2.3 Интеграция с Git.

Git является популярной системой контроля версий, и VSCode встроенно интегрирована с Git, обеспечивая возможность управления версиями кода и совместной работы в команде.

2.2.4 Легковесность и производительность.

VSCode обладает низким потреблением ресурсов и быстрым запуском, что позволяет разработчикам эффективно работать даже на менее мощных компьютерах.

Выбор PostgreSQL в качестве СУБД и Visual Studio Code в качестве среды разработки обусловлен их открытым исходным кодом, гибкостью, мощностью и активными сообществами разработчиков. Эти инструменты обеспечивают надежную и удобную платформу для создания веб-приложения для логистики грузоперевозок с оптимизированной базой данных и удобной средой разработки.

3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ

Разработка веб-приложения для логистики грузоперевозок имеет важное теоретическое обоснование, основанное на современных концепциях и практиках в области логистики, информационных технологий и управления данными. В данном разделе мы рассмотрим ключевые аспекты, которые обосновывают необходимость и актуальность создания такого приложения.

3.1 Роль информационных технологий в логистике

Логистика, как важное звено в современном бизнесе, подверглась значительным изменениям и усовершенствованиям благодаря информационным технологиям. Автоматизация процессов, управление данными и оперативная связь между всеми участниками логистической цепи стали неотъемлемой частью успешной логистики. Создание веб-приложения, специализированного на логистике грузоперевозок, обосновывается не только оптимизацией процессов, но и увеличением эффективности, снижением издержек и соблюдением высоких стандартов обслуживания.

3.2 Преимущества веб-приложений в логистике

Веб-приложения предоставляют ряд существенных преимуществ в сфере логистики грузоперевозок:

3.2.1 Улучшение прозрачности.

Веб-приложение позволяет заказчикам и поставщикам отслеживать статус грузов в режиме реального времени, что повышает прозрачность и уровень доверия.

3.2.2. Оптимизация маршрутов и ресурсов.

Автоматизированный анализ и оптимизация маршрутов и использования ресурсов, включая транспорт и склады, позволяют снижать издержки и повышать эффективность операций.

3.2.3. Быстрое реагирование на изменения.

Система способствует оперативному реагированию на нештатные ситуации и изменения в заказах, позволяя минимизировать риски и улучшать обслуживание клиентов.

3.3 Роль базы данных в логистическом приложении

База данных является структурой, которая хранит, управляет и предоставляет доступ к данным, необходимым для функционирования веб-приложения. Важными аспектами, обосновывающими использование базы данных в данном контексте, являются:

3.3.1 Централизованное хранение данных.

Централизация данных в базе позволяет упростить их управление, обеспечивая доступность информации для всех участников логистической цепи.

3.3.2 Многопользовательский доступ.

База данных обеспечивает многопользовательский доступ, позволяя одновременно работать с данными нескольким пользователям, что важно в условиях совместной работы множества участников.

3.3.3 Автоматизация и аналитика.

Системы управления базами данных позволяют автоматизировать процессы, включая определение оптимальных маршрутов, а также предоставляют данные для анализа производительности и принятия решений.

3.3.4 Безопасность данных.

Базы данных предоставляют механизмы для обеспечения конфиденциальности и целостности данных, что является критическим аспектом в логистике грузоперевозок.

Таким образом, разработка веб-приложения для логистики грузоперевозок имеет теоретическое обоснование, опирающееся на преимущества информационных технологий, веб-приложений и баз данных в сфере логистики. Создание такого приложения обеспечивает улучшение эффективности, оптимизацию процессов и повышение уровня обслуживания клиентов.

4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Проектирование начнем с формирования DDL-скриптов:

```
CREATE TABLE users (
    user_uid UUID NOT NULL PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password_hash VARCHAR(100) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    phone_number VARCHAR(50) UNIQUE NOT NULL,
    avatar_url VARCHAR(255) NOT NULL,
    role_uid UUID REFERENCES roles(role_uid),
    registration_date TIMESTAMP DEFAULT current_timestamp
);
CREATE TABLE roles (
  role_uid UUID NOT NULL PRIMARY KEY,
  role_name VARCHAR(50) NOT NULL
)

CREATE TABLE user_actions (
    action_id UUID NOT NULL PRIMARY KEY,
    user_uid UUID REFERENCES users(user_uid),
    action_description TEXT NOT NULL,
    log_date TIMESTAMP DEFAULT NOT NULL,
);
CREATE TABLE vehicles (
    vehicle_uid UUID NOT NULL PRIMARY KEY,
    user_uid UUID REFERENCES users(user_uid),
    vehicle_name VARCHAR(100) NOT NULL,
    vehicle_capacity DECIMAL(10, 2),
    vehicle_price DECIMAL(10, 2),
    vehicle_description VARCHAR(255) UNIQUE NOT NULL,
    image_url VARCHAR(255) NOT NULL,
```

```
    mark VARCHAR(50) NOT NULL,
    model VARCHAR(50) NOT NULL,
);
CREATE TABLE favorites (
    favorite_id UUID NOT NULL PRIMARY KEY,
    user_uid UUID REFERENCES users(user_uid),
    vehicle_uid UUID REFERENCES vehicles(vehicle_uid),
);
CREATE TABLE current_order (
    current_order_id UUID NOT NULL PRIMARY KEY,
    user_uid UUID REFERENCES users(user_uid),
    tracking_uid UUID REFERENCES cargo_tracking(tracking_uid),
);

CREATE TABLE cargo_tracking (
    tracking_uid UUID NOT NULL PRIMARY KEY,
    latitude DECIMAL(9, 6),
    longitude DECIMAL(9, 6),
    timestamp TIMESTAMP DEFAULT current_timestamp,
    vehicle_uid UUID REFERENCES vehicles(vehicle_uid),
);
CREATE TABLE payments (
    payment_id UUID NOT NULL PRIMARY KEY,
    amount DECIMAL(10, 2) NOT NULL,
    payment_date TIMESTAMP DEFAULT current_timestamp,
    user_uid UUID REFERENCES users(user_uid),
);
CREATE TABLE cargo_history (
    cargo_uid UUID NOT NULL PRIMARY KEY,
    start_location VARCHAR(100) NOT NULL,
    end_location VARCHAR(100) NOT NULL,
    cargo_description TEXT,
```

```
    start_date DATE,
    end_date DATE,
    user_uid UUID REFERENCES users(user_uid),
    vehicle_uid UUID REFERENCES vehicles(vehicle_uid),
);
CREATE TABLE reviews (
    review_id UUID NOT NULL PRIMARY KEY,
    rating INT NOT NULL CHECK (rating >= 1 AND rating <= 5),
    review_text TEXT,
    review_date TIMESTAMP DEFAULT current_timestamp,
    user_uid UUID REFERENCES users(user_uid),
    cargo_uid UUID REFERENCES cargo_history(cargo_uid),
);
```

И на этом пока что закончим.

5 ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ВЕБ-ПРИЛОЖЕНИЯ

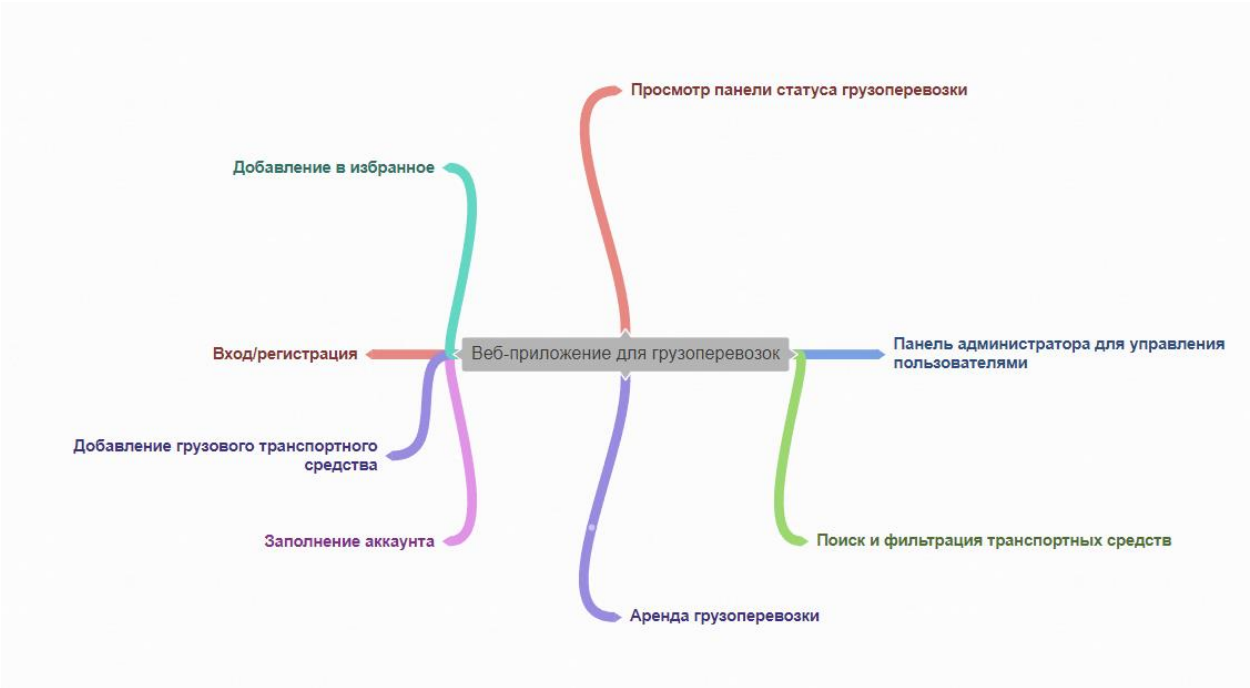

Рисунок 5.1 – Функциональная карта программного продукта

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ