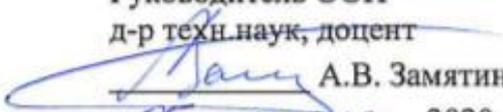


НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук
Кафедра теоретических основ информатики

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
д-р техн. наук, доцент


А.В. Замятин

« 05 » июня 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

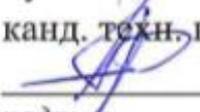
WEB-ПРИЛОЖЕНИЕ ДЛЯ РАСЧЁТА РАЗМЕРА ВРЕДА,
ПРИЧИНЯЕМОГО ТРАНСПОРТНЫМИ СРЕДСТВАМИ,
ОСУЩЕСТВЛЯЮЩИМИ ПЕРЕВОЗКИ ТЯЖЕЛОВЕСНЫХ ГРУЗОВ

по основной образовательной программе подготовки бакалавров
«Фундаментальная информатика и информационные технологии»
направления подготовки 02.03.02 Фундаментальная информатика и
информационные технологии

Иванов Владислав Дмитриевич

Руководитель ВКР

канд. техн. наук


А. В. Приступа

подпись

« 05 » июня 2020 г.

Автор работы

студент группы № 1461


В. Д. Иванов

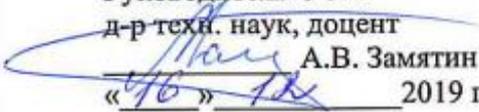
подпись

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра теоретических основ информатики

УТВЕРЖДАЮ

Руководитель ООП

д-р техн. наук, доцент

 А.В. Замятин

« 16 » 12 2019 г.

ЗАДАНИЕ

по подготовке ВКР _____ бакалавра _____
студенту Иванову Владиславу Дмитриевичу группы № 1461

1. Тема ВКР: Web-приложение для расчёта размера вреда, причиняемого транспортными средствами, осуществляющими перевозки тяжеловесных грузов.
2. Срок сдачи студентом выполненной ВКР:
а) на кафедре: « 05 » 06 2020 г.
б) в ГЭК: « 11 » 06 2020 г.

3. Исходные данные к работе:

Цель: создание web-приложения для использования владельцами автомобильных дорог и грузоперевозчиками, предоставляющего функционал расчёта размера вреда (в рублях), причиняемого тяжеловесными транспортными средствами.

Задачи:

1. Определение основного функционала.
2. Изучение предметной области.
3. Выбор инструментов разработки.
4. Проектирование приложения.
5. Разработка приложения.

Объект исследования: перевозки тяжеловесных грузов

Метод исследования: практическая реализация на ЭВМ

4. Краткое содержание работы:

- 10.02.2020 - 14.02.2020: Определить функциональные требования.
- 15.02.2020 - 25.02.2020: Изучить предметную область.
- 26.02.2020 - 04.03.2020: Выбрать и изучить инструменты разработки.
- 05.03.2020 - 15.03.2020: Спроектировать архитектуру приложения.
- 15.03.2020 - 27.04.2020: Разработать приложение.

5. Организация, по заданию которой выполняется работа: ТГУ

6. Дата выдачи задания: 16 декабря 2019 г.

Руководитель ВКР

канд. техн. наук, доцент кафедры ТОИ

НИИ ТГУ 

Приступа А.В.

Задание принял к исполнению

16.12.19 Иванова
дата, подпись студента

РЕФЕРАТ

Выпускная квалификационная работа 45 страниц, 25 рисунков, 12 источников.

WEB-ПРИЛОЖЕНИЕ, ВЕБ-ПРИЛОЖЕНИЕ, ANGULAR, GRAPHQL

Цель работы – разработать полноценное web-приложение, предоставляющее функционал для расчёта размера вреда в рублях, причиняемого транспортными средствами, осуществляющими перевозки тяжеловесных грузов.

Результаты работы:

- Проведен подробный анализ функциональных и нефункциональных требований.
- Проведен выбор инструментов реализации.
- Спроектирована архитектура и завершена разработка веб-приложения.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ	6
1 Определение требований	7
1.1 Анализ предметной области	7
1.1.1 Методика расчета размера платы в счет возмещения вреда	7
1.1.2 Методика расчета вреда на одну ось	8
1.1.3 Методика расчета осевых нагрузок	10
1.2 Анализ существующих решений	13
1.3 Анализ требований	14
1.4 Модель предметной области	17
2 Инструменты	20
3 Проектирование	21
3.1 Архитектура системы	21
3.2 База данных	24
3.3 Пакет Models	26
3.4 Пакет Resolvers	27
3.5 Пакет Calc	28
3.6 Frontend приложение	30
4 Реализация	37
ЗАКЛЮЧЕНИЕ	41
ЛИТЕРАТУРА	42

ВВЕДЕНИЕ

Рынок автомобильных грузоперевозок являются ключевой статьей в структуре рынка грузоперевозок в России и удельный вес транспортной логистики в общем объеме ВВП России составляет примерно 6% [1]. Большие объемы грузоперевозок создают серьезные нагрузки на транспортное полотно. Для возмещения вреда, причиненного перегрузками, с владельцев машин взимается плата, которая зависит от большого количества факторов.

Исходя из этого, процесс расчета осевых нагрузок и размеров вреда вручную может занять продолжительное время, так как нужно учитывать множество исходных данных, правила перевозок грузов и правила возмещения вреда, использовать различные формулы подсчета.

Итак, целью работы является создание web-приложения для владельцев автомобильных дорог и грузоперевозчиков, предоставляющего функционал расчёта размера вреда дорожному полотну в рублях, причиняемого транспортными средствами, осуществляющими перевозки тяжеловесных грузов.

Эта цель определяет следующие задачи:

- Определение основных функциональных требований
- Анализ существующих решений
- Выбор и изучение средств разработки
- Проектирование web-приложения
- Разработка web-приложения

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ

1. CRUD (Create, Read, Update, Delete) – функции, которые используются при взаимодействии с сущностью в базе данных: создание, чтение, модификация, удаление.
2. Тележка – одна ось, группа осей или седельно-сцепное устройство транспортного средства.
3. Транспортное средство – автопоезд (тягач + прицеп/полуприцеп) или одиночный грузовой автомобиль без прицепа.

1 Определение требований

1.1 Анализ предметной области

В первую очередь необходимо выделить основные понятия и сущности, которые раскрывают предметную область.

Под транспортным средством, осуществляющим перевозки тяжеловесных грузов, понимается транспортное средство, или комбинация транспортных средств (автопоезд), масса которого с грузом или без груза превышает допустимые массы транспортных средств и (или) допустимые осевые нагрузки, установленные правилами перевозки грузов[2], решением о временном ограничении движения транспортных средств по автомобильным дорогам и запрещающими дорожными знаками «Ограничение массы» и (или) «Ограничение массы, приходящейся на ось транспортного средства».

Автопоезд состоит из тягача и прицепа (или полуприцепа), характеристики каждого из которых учитываются при расчете вреда.

Для тягача такими характеристиками являются масса тягача без груза, масса полностью груженого тягача, количество осей в тележках. Если предварительное взвешивание транспортного средства не проведено и осевые нагрузки при данной нам массе груза неизвестны, то особенно важны коэффициенты распределения массы на переднюю и заднюю тележки. Для прицепа характеристики аналогичны.

Итоговый результат, который должна предоставить программа – отчет, отображающий информацию о перевозчике, получателе груза, подробностях расчета вреда, размер вреда в рублях.

1.1.1 Методика расчета размера платы в счет возмещения вреда

Согласно постановлению Правительства РФ [2] размер платы в счет возмещения вреда рассчитывается с использованием формулы:

$$P_p = [P_{пм} + (P_{пом1} + P_{пом2} + \dots + P_{помi})] \times S \times T_{тг} \quad (1)$$

Где:

- P_p – размер платы в рублях
- $P_{пм}$ – размер вреда при превышении значения допустимой массы транспортного средства, которая определена в приложении №1 к Правилам перевозок грузов автомобильным транспортом:

Таблица 1 – Допустимые массы транспортных средств

Тип транспортного средства или комбинации транспортных средств, количество и расположение осей	Допустимая масса транспортного средства, тонн
Одиночные автомобили	
двухосные	18
трехосные	25
четырёхосные	32
пятиосные	35
Автопоезда седельные и прицепные	
трехосные	28
четырёхосные	36
пятиосные	40
шестиосные и более	44

- $R_{\text{пом}i}$ – размер вреда при превышении значения допустимой нагрузки на i -ю ось транспортного средства, которая определена в приложении №2 к Правилам перевозок грузов автомобильным транспортом (Приложение А).

- S – протяженность маршрута перевозки.

- $T_{\text{тг}}$ – базовый компенсационный индекс текущего года, который рассчитывается по следующей формуле:

$$T_{\text{тг}} = T_{\text{пт}} \times I_{\text{тг}} \quad (2)$$

Где:

- $T_{\text{пт}}$ – базовый компенсационный индекс предыдущего года.

- $I_{\text{тг}}$ – индекс-дефлятор инвестиций в основной капитал. Его вычисление не входит в функционал данного web-приложения.

1.1.2 Методика расчета вреда на одну ось

При расчете размера вреда на одну ось используются формулы:

а) Для дорог с одеждой капитального (облегченного) типа:

$$R_{\text{пом}i} = K_{\text{дкз}} \times K_{\text{кап.рем}} \times K_{\text{сез}} \times R_{\text{исх.ось}} \times (1 + 0,2 \times \text{П}_{\text{ось}}^{1,92} \times (a / H - b)) \quad (3)$$

б) Для дорог с одеждой переходного типа:

$$R_{\text{пом}i} = K_{\text{кап.рем}} \times K_{\text{сез}} \times R_{\text{исх.ось}} \times (1 + 0,14 \times \text{П}_{\text{ось}}^{1,24} \times (a / H - b)) \quad (4)$$

Так же необходимо рассчитывать размер вреда при превышении допустимой массы с помощью формулы:

$$P_{\text{пм}} = K_{\text{кап.рем}} \times K_{\text{пм}} \times P_{\text{исх.пм}} \times (1 + c \times P_{\text{пм}}) \quad (5)$$

Где:

- $K_{\text{дкз}}$ - коэффициент, учитывающий условия дорожно-климатических зон (таблица 2).
- $K_{\text{кап.рем}}$ - коэффициент, учитывающий стоимость выполнения работ по капитальному ремонту в зависимости от расположения автомобильной дороги (таблица 2).
- $K_{\text{сез}}$ - коэффициент, учитывающий природно-климатические условия, равный 1 при неблагоприятных природно-климатических условиях, в остальное время равный 0,35.
- $P_{\text{исх.ось}}$ - исходное значение размера вреда при превышении допустимых нагрузок на ось транспортного средства для автомобильной дороги (таблица 3).
- $P_{\text{ось}}$ - величина превышения фактической нагрузки на ось транспортного средства над допустимой для автомобильной дороги.
- H - нормативная нагрузка на ось транспортного средства для автомобильной дороги.
- a, b, c - постоянные коэффициенты (таблица 3).

Таблица 2 – Коэффициенты для федеральных округов

Федеральный округ	$K_{\text{дкз}}$	$K_{\text{кап.рем}}$	$K_{\text{пм}}$
Центральный	2,07	1,00	1
Северо-Западный	2,14	1,07	0,838
Южный	1,59	0,96	1,255
Приволжский	1,67	0,94	0,498
Уральский	2,10	1,03	0,426
Сибирский	2,06	1,01	0,403
Дальневосточный	2,14	1,35	0,508
Северо-Кавказский	1,48	0,96	0,595

Таблица 3 – Постоянные коэффициенты

Нормативная (расчетная) осевая нагрузка для автомобильной дороги, тонн/ось	$P_{\text{исх.}}$ руб./100 км	Постоянные коэффициенты			
		a	b	c	d
6	8500	7,3	0,27	7365	123,4
10	1840	37,7	2,4	7365	123,4
11,5	840	39,5	2,7	7365	123,4

Указанные выше формулы используются только тогда, когда превышение осевых нагрузок по отношению к нормативным является большим или равным 60% (Превышение массы рассчитывается по формуле в независимости от процента превышения). В противном случае используются таблицы фиксированных тарифов, которые указаны в постановлении Правительства РФ [3] (Таблица 4).

Таблица 4 – Размер вреда, причиняемого транспортными средствами, осуществляющими перевозки тяжеловесных грузов, при движении таких транспортных средств по автомобильным дорогам федерального значения, рассчитанным на нормативную (расчетную) осевую нагрузку 10 тонн/ось, вследствие превышения допустимых осевых нагрузок на каждую ось транспортного средства.

Превышение допустимых осевых нагрузок (процентов)	Федеральный округ							
	Центральный	Северо-Западный	Южный	Приволжский	Уральский	Сибирский	Дальневосточный	Северо-Кавказский
До 10	2747	2386	1637	1520	2059	1963	2773	1387
От 10 до 20	4392	3815	2618	2430	3291	3139	4434	2218
От 20 до 30	7026	6102	4188	3887	5265	5021	7093	3548
От 30 до 40	10616	9221	6329	5874	7956	7587	10718	5361
От 40 до 50	15141	13152	9026	8377	11347	10821	15286	7646
От 50 до 60	20584	17879	12271	11389	15426	14711	20781	10395
От 60	рассчитывается по формулам, указанным выше							

1.1.3 Методика расчета осевых нагрузок

В случае, если осевые нагрузки транспортного средства неизвестны, то необходимо их рассчитать. Это можно сделать, основываясь на его паспортных данных. Такими паспортными данными являются масса груза, масса порожнего и полностью нагруженного транспортного средства и распределение нагрузок на переднюю и заднюю тележки.

Первым этапом расчета осевых нагрузок необходимо рассчитать коэффициенты распределения массы для передней и задней тележек транспортного средства.

Исходя из данных завода-производителя мы знаем массу, которая приходится на переднюю тележку и на заднюю тележку снаряженного (без груза) транспортного средства (Рисунок 1). Также мы знаем данные характеристики для транспортного средства в условиях полной нагрузки (Рисунок 2). На примере седельного тягача КАМАЗ-5410 можно увидеть, как

меняются осевые нагрузки, если нагрузить тягач полностью. Предположим, что грузоподъемность транспортного средства составляет 8,2 тонны, масса, приходящаяся на заднюю тележку порожнего ТС – 3,3 тонны, полностью груженого – 10,96 тонн; на переднюю тележку порожнего ТС – 3,4, полностью груженого – 3,94 тонн. При максимальной нагрузке транспортного средства веса, приходящиеся на обе тележки, распределяются определенным образом: на заднюю тележку добавляется 7,66 тонн от веса груза, а на переднюю – 0,54 тонны.

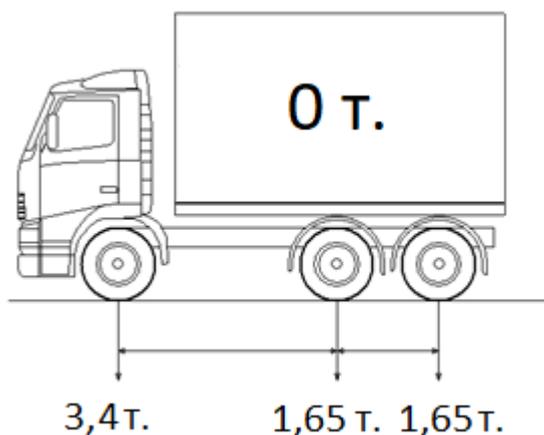


Рисунок 1 – Распределение нагрузок на тележки ТС в порожнем (пустом) состоянии

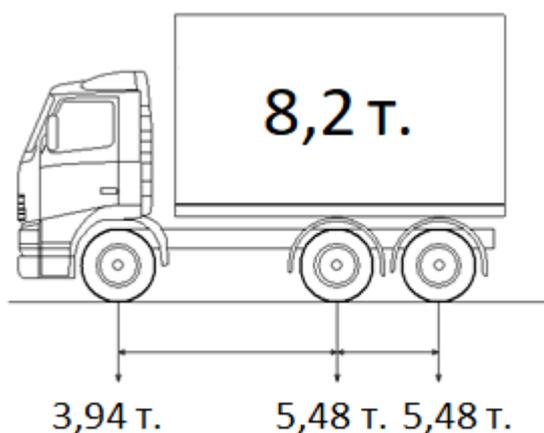


Рисунок 2 – Распределение нагрузок на тележки ТС в условиях полной загрузки

Для справедливости дальнейших рассуждений будем исходить из предположения, что транспортные средства подвергаются **равномерной** загрузке, т.е. в данной работе положение груза не учитывается. В этих условиях каждая тонна перевозимого груза распределяется между тележками в соответствии с некоторым коэффициентом по линейному закону.

Следовательно, для любой массы груза справедлива формула линейной интерполяции:

$$\varphi(x) = f(a) + (x - a) \frac{f(b) - f(a)}{b - a} \quad (6)$$

В приведенном выше примере $a = 0$, $b = 8.2$ (максимальная грузоподъемность), $f(b)$ является массой, приходящейся на заднюю тележку при максимальной нагрузке на средство, $f(a)$ является массой, приходящейся на заднюю тележку при отсутствии груза, а x является текущей массой груза. После необходимых подстановок данная формула примет вид:

$$m_B = w_B^1 + x * k_B, k_B = \frac{(w_B^2 - w_B^1)}{w_M} \quad (7)$$

Где:

- m_B – масса, приходящаяся на заднюю тележку при массе x груза.
- w_B^1, w_B^2 - массы, приходящиеся на заднюю тележку при порожнем и полностью груженом транспортном средстве.
- w_M - максимальная грузоподъемность транспортного средства.

Из этого следует, что коэффициент k представляет собой отношение разности масс, приходящихся на тележку в полностью груженом и порожнем состояниях, к максимальной грузоподъемности транспортного средства. Таким образом, отношение нагрузки на заднюю тележку ко всему весу транспортного средства будет равно 0,934.

Если в расчете кроме основного транспортного средства (тягача или грузовика) имеется прицеп, то расчет следует начинать с прицепа, потому что, в дальнейшем, при расчете осевых нагрузок тягача массой, которая приходится на седельно-сцепное устройство тягача, а следовательно и на его заднюю тележку, будет являться масса, приходящаяся на седельно-сцепное устройство прицепа (переднюю тележку прицепа).

После вычисления коэффициента массы, которая приходится на заднюю тележку, можно получить коэффициент массы, приходящейся на переднюю тележку, вычитая из единицы полученный в ходе вычисления коэффициент:

$$k_F = 1 - k_B \quad (8)$$

Исходя из формулы (8) для нашего примера можно вычислить отношение нагрузки на переднюю тележку ко всему весу – оно станет равно 0,066.

Затем следует умножить сумму массы транспортного средства и груза на полученные коэффициенты и, тем самым получить массы, которые приходятся на переднюю и заднюю тележки соответственно.

$$w_B = w_M * k_B, w_F = w_M * k_F \quad (9)$$

Нагрузки на оси вычисляются простым делением массы, приходящуюся на конкретную тележку, на количество осей в тележке.

$$w_B^i = \frac{w_B}{n_B}, i = 1..n_B; w_F^i = \frac{w_F}{n_B}, i = 1..n_B \quad (10)$$

Где:

- w_B^i, w_F^i – нагрузки, приходящиеся на оси задней и передней тележек.

- n_B, n_F – количества осей на задней и передней тележках.

Таким образом, для нашего примера в условиях частичной загрузки распределение массы будет выглядеть следующим образом:

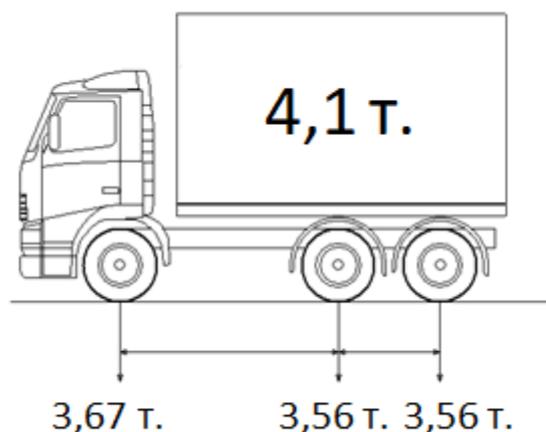


Рисунок 3 – Распределение нагрузок на тележки ТС в условиях частичной загрузки

При массе груза 4,1 тонны вес на передней тележке будет равен $m_F = 3,4 + 0.066 * 4,1 = 3,67$ тонны, на задней тележке $m_B = 3,3 + 0.934 * 4,1 = 7,13$ тонны. Поделив вес, приходящийся на заднюю тележку, на число осей в тележке, получим 3,56 тонны на каждую ось.

Только после получения осевых нагрузок продолжается процесс расчета стоимости вреда, основываясь на таблицах и формулах, приведенных в главах 1.1.1 и 1.1.2.

1.2 Анализ существующих решений

Единственным существующим решением на данный момент является программа КТГ-Калькулятор, которая представляет собой веб-сайт, где есть различные формы для вычисления осевых нагрузок и причиненного вреда. Из недостатков данного решения можно выделить:

- Доступ к функциям предоставляется платно и через регистрацию в несколько этапов (Выбор нужного тарифа, отправка заявки, получение данных для входа через почту), что занимает довольно продолжительное время.
- Нет возможности сохранить введенные данные для повторных вычислений и формирований отчетов. Это накладывает определенные ограничения на удобство использования и требует большего количества времени на создание расчета.

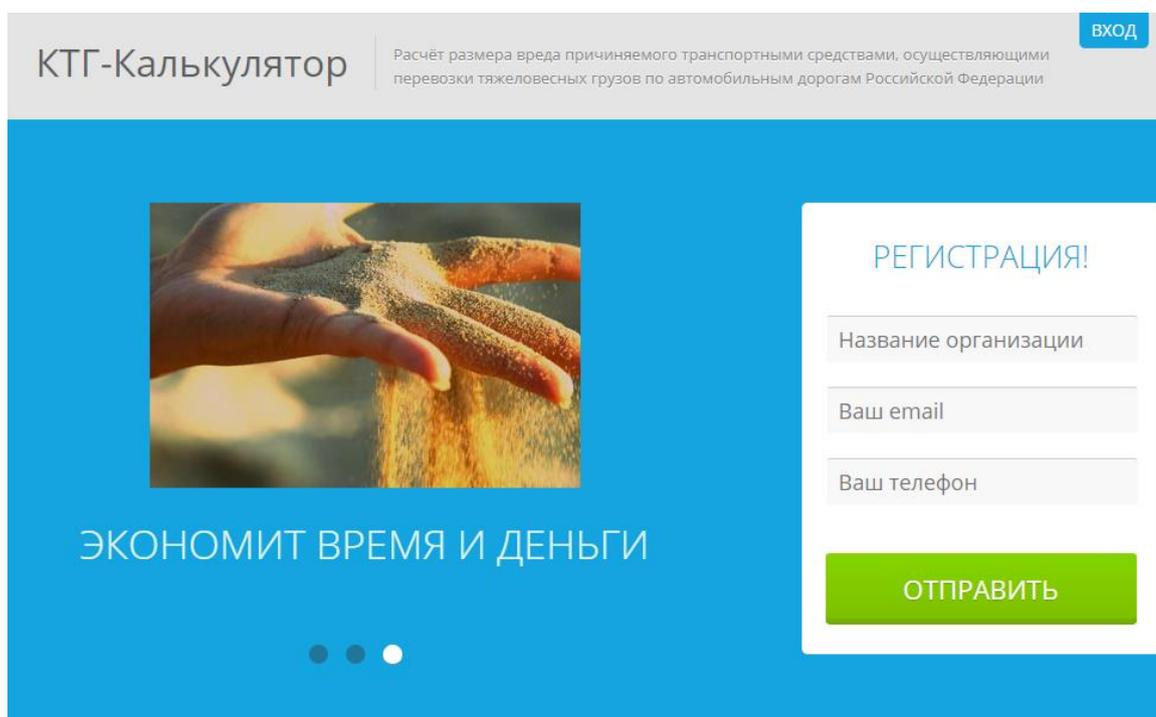


Рисунок 4 – КТГ-Калькулятор

1.3 Анализ требований

Для графического описания моделирования системы используется язык UML [11].

В ходе анализа выявлены следующие функциональные требования для приложения:

- Приложение должно обеспечивать возможность формирования отчета, в котором будет содержаться вся необходимая для последующей оплаты налога информация.
- Должно поддерживать авторизацию, чтобы созданные расчеты сохранялись в конкретном рабочем пространстве. Аккаунты должны быть привязаны к рабочему пространству, которое соответствует определенной организации.
- Иметь возможность расчета осевых нагрузок. Расчет осевых нагрузок необходим, если осевые нагрузки неизвестны и не измерены заранее.
- Иметь возможность сохранить сформированные отчеты и данные, необходимые для его формирования, для дальнейшего повторного использования этих данных и сохранения истории отчетов.
- Должна присутствовать возможность печати отчета.
- Разработчик программы должен иметь возможность поддерживать используемые в расчетах справочники в актуальном состоянии через панель администратора.

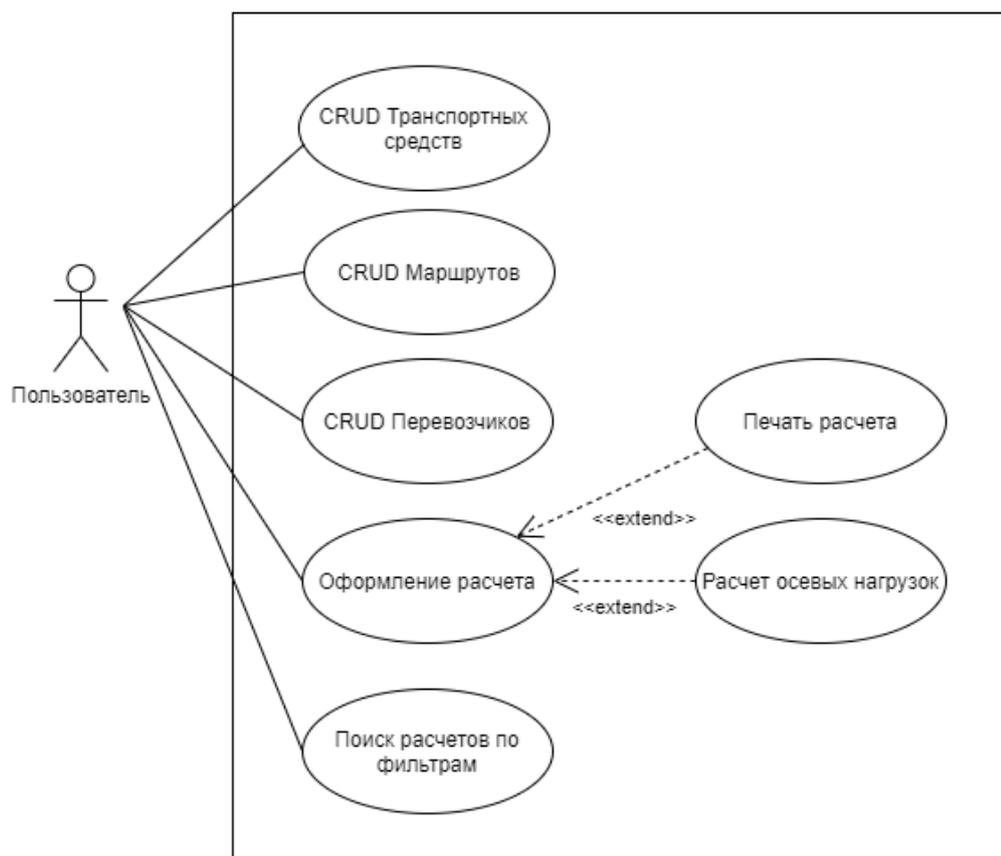


Рисунок 5 – Диаграмма вариантов использования для пользователя

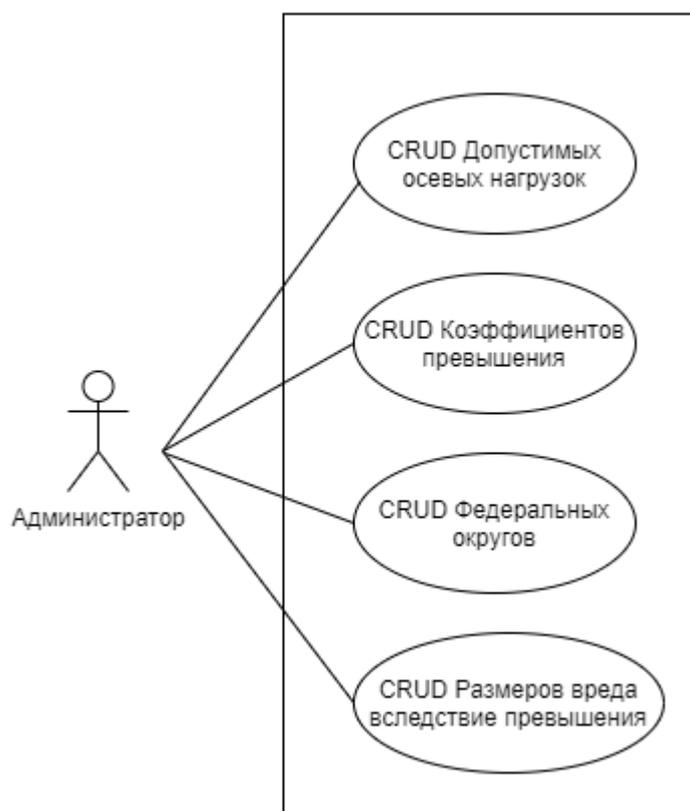


Рисунок 6 – Диаграмма вариантов использования для администратора

В представленном ниже сценарии вариантов использования отображен типичный ход событий при использовании web-приложения.

Таблица 5 – Главный раздел сценария вариантов использования

<i>Главный раздел</i>	
Вариант использования	Оформление расчета
Актеры	Пользователь, Система
Цель	Получение подробной информации о грузоперевозке и причиненном вреде в рублях
Краткое описание	Пользователь указывает данные о грузоперевозке. Система рассчитывает причиненный дороге вред в рублях. Система отображает подробную информацию о грузоперевозке.
Предусловие	Пользователь авторизован в системе
Ссылки на другие варианты использования	Может быть расширен следующими ВИ: - Печать расчета

Таблица 6 – Типичный ход событий сценария вариантов использования

<i>Типичный ход событий</i>	
Действия актера	Отклик системы
1. Пользователь переходит на вкладку «Отчеты»	2. Система отображает страницу с созданными отчетами
3. Пользователь нажимает на кнопку «Создать отчет»	4. Система отображает форму с созданием отчета
5. Пользователь выбирает модель тягача	6. Система отображает данные о тягаче
7. Пользователь выбирает модель прицепа	8. Система отображает данные о прицепе
9. Пользователь выбирает маршрут	10. Система отображает данные о маршруте
11. Пользователь выбирает перевозчика	12. Система отображает данные о перевозчике
13. Пользователь вводит вес груза, фактические осевые нагрузки и нажимает кнопку «Создать»	14. Система сверяет осевые нагрузки с допустимыми осевыми нагрузками 15. Система рассчитывает процент превышения осевых нагрузок допустимых осевых нагрузок

Продолжение таблицы 6

<p>Исключение №1: Пользователь не знает фактические осевые нагрузки</p>	<p>16. Система вычисляет вред в рублях 17. Система отображает отчет со всей информацией в виде HTML-документа</p>
--	---

Таблица 7 – Исключения сценария вариантов использования

<i>Исключения</i>	
Действия актера	Отклик системы
Исключение №1: Пользователь не знает фактические осевые нагрузки	
	<p>14. Система рассчитывает осевые нагрузки на основе данных о тягаче и прицепе 15. Система сверяет осевые нагрузки с допустимыми осевыми нагрузками 16. Система рассчитывает процент превышения осевых нагрузок допустимых осевых нагрузок 17. Система вычисляет вред в рублях 18. Система отображает отчет со всей информацией в виде HTML-документа</p>

1.4 Модель предметной области

На рисунке 7 изображена модель предметной области разрабатываемой системы без учета особенностей выбранных инструментов и среды разработки.

В представленной модели предметной области присутствуют:

- Пользователь системы.
- Организация, к которой будет привязан пользователь. Необходима для того, чтобы несколько пользователей имели доступ к расчетам одной организации.

- Расчет, который будет результатом работы программы. Расчет содержит в себе всю информацию о перевозчике, подробности самого процесса расчета, даты начала и конца перевозки, дату, номер и назначение расчета.

- Перевозчик груза. Нужно отметить, что перевозчик имеет атрибуты получателя груза. Решено объединить эти сущности т.к. на практике получатель груза всегда пользуется услугами одного и того же перевозчика.

- ПозицияРасчета является комбинацией маршрута и расчетных данных транспортных средств. Эта сущность необходима, потому что один расчет может содержать более одной такой комбинации.

- Маршрут содержит в себе данные о протяженности, нормативной осевой нагрузке и наличии промерзшего грунта.

- РасчетныеДанныеСредства включают в себя все расчетные данные транспортного средства, которые используются в процессе расчета.

- ОсеваяНагрузка включает в себя информацию о конкретной оси транспортного средства, которая формируется в процессе расчета в несколько этапов.

- ТранспортноеСредство, которое может представлять собой как тягач (грузовик), так и прицеп (указывается в поле «Тип»). Данные сущности объединены в одну т.к. имеют одинаковые атрибуты.

- ДопустимаяОсеваяНагрузка – справочник, содержащий допустимые осевые нагрузки транспортных средств в зависимости от нормативной осевой нагрузки маршрута, расстояния между осями, типа тележки и числа колес на оси.

- ДопустимаяМасса – справочник, который указывает допустимые массы транспортного средства в зависимости от типа транспортного средства, количества и расположения осей.

- КоэффициентПревышения – таблица, содержащая зависимости постоянных коэффициентов a , b , c , d и исходного значения вреда от нормативной осевой нагрузки автомобильной дороги.

- ФедеральныйОкруг – таблица, содержащая коэффициенты, касающиеся конкретного федерального округа.

- РазмерВредаВследствиеПревышения – таблица, содержащая размер вреда на 100 км в рублях в зависимости от федерального округа, превышения допустимых осевых нагрузок на ось транспортного средства в процентах и нормативную осевую нагрузку дороги.

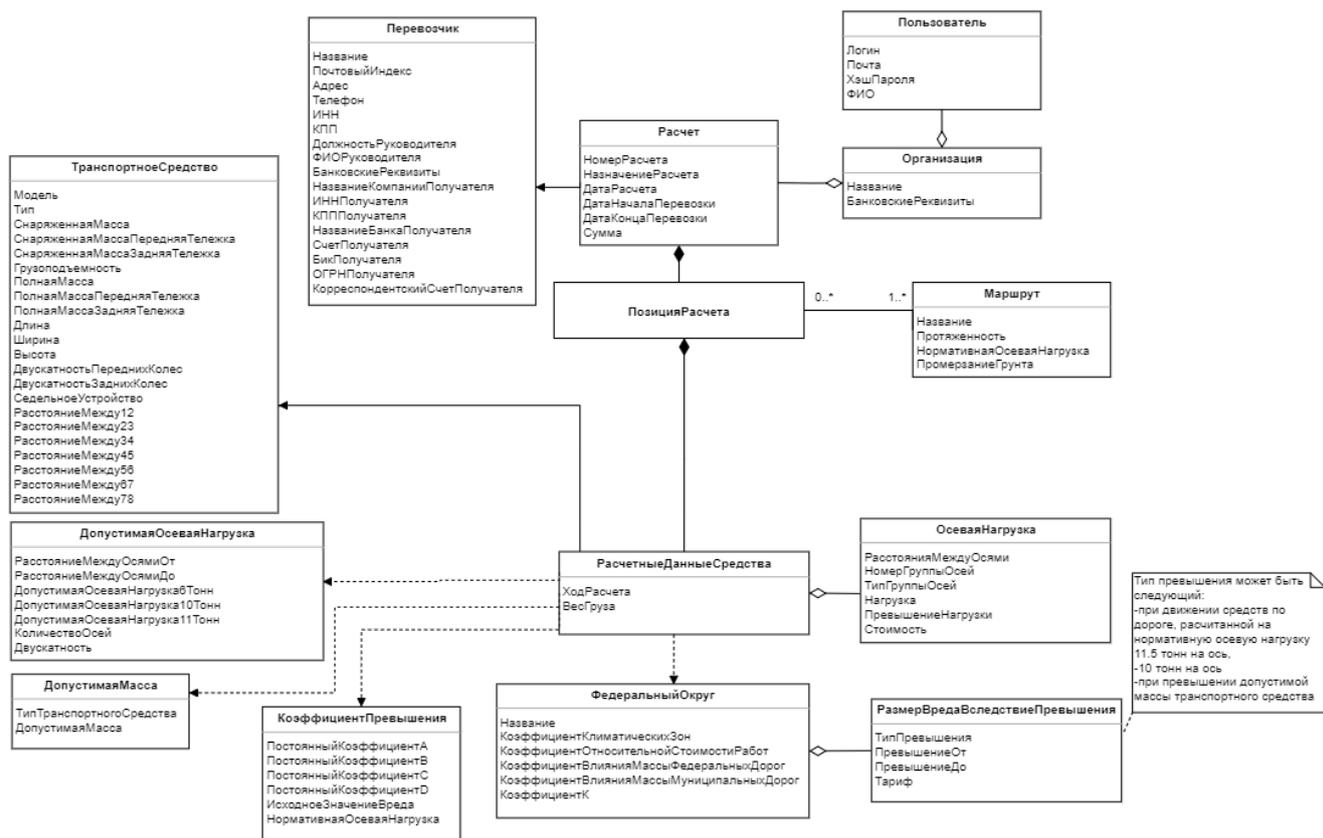


Рисунок 7 – Диаграмма классов модели предметной области

2 Инструменты

Для разработки web-приложения выбраны следующие инструменты:

1. Typescript – язык, используемый для разработки приложения. Является расширением языка Javascript, добавляющий типизацию, полноценное ООП, возможность обобщенного программирования и т.д [4].
2. Node.js – среда для серверной разработки на языке JavaScript [5].
3. Express – программная платформа для среды Node.js, предоставляющая обширный функционал для разработки web-приложений [6].
4. GraphQL – язык запросов и манипулирования данными и средой выполнения запросов [7].
5. Apollo – сервер для обработки GraphQL запросов [8].
6. MySQL – в качестве базы данных.
7. Sequelize – ORM библиотека для Node.js приложений [9].
8. Angular – фреймворк для разработки одностраничных приложений, использующий паттерн проектирования MVC [10].
9. HTML/SCSS – верстка и оформление сайта.

3 Проектирование

3.1 Архитектура системы

Важной особенностью архитектуры системы является то, что вместо наиболее распространенной и привычной REST API-архитектуры используется GraphQL API.

REST API (Representational State Transfer API) – способ организации архитектуры приложения, при котором для получения необходимых данных имеется интерфейс с множеством HTTP-запросов, которые могут содержать определенные параметры. Такие запросы в данном приложении, к примеру, могли бы быть запросами, которые отвечали бы за получение данных о конкретном перевозчике, редактирование транспортного средства или создание отчета.

GraphQL API представляет собой язык запросов, который используется на стороне клиентских приложений для работы с данными. В отличие от REST API в таких приложениях используются одна конечная точка, которая обрабатывает запросы на языке GraphQL вместо большого количества REST-запросов, каждый из которых представляет собой отдельную конечную точку. GraphQL API строится на двух основных слоях:

- Схема (schema или type definitions) – определяет типы данных и запросы, которыми может воспользоваться клиент, например:

```
type Path {
  id: Int
  name: String
  length: Int
  standardAxialLoad: Float
  soilFreezing: Boolean
  districtID: Int
}

extend type Query {
  path(id: Int): Path
  paths: [Path]
}
```

Листинг 1 – Пример схемы

В данном отрывке схемы мы определяем тип Path, его атрибуты и типы, затем возможные запросы path(id), который предоставляет клиенту данные о конкретном пути, и paths, которые

предоставляет данные о всех имеющихся в базе путях. В запросах указываются возможные атрибуты и возвращаемый тип запроса.

- Распознаватели (resolvers) – при запросе со стороны клиента, сервер GraphQL должен понять, как и где получить необходимые клиенту данные. Для этого и нужны распознаватели. Распознаватели всегда имеют название, соответствующее определенному запросу в GraphQL схеме:

```
const PathResolver = {
  Query: {
    path(_ : any, args: any) {
      return Path.findOne({ where: args });
    },
    paths(_ : any, args: any) {
      return Path.findAll();
    }
  }
}
```

Листинг 2 – Распознаватель PathResolver

В примере выше имеется объект PathResolver с вложенным объектом Query, имеющим методы path и paths, каждый из которых возвращает результат обращения к методам модели Path, и, соответственно, нужные клиенту данные.

Теперь для получения данных о путях достаточно написать следующий запрос на стороне клиента (Листинг 3):

```
query paths {
  paths {
    id
    name
    length
    standardAxialLoad
    soilFreezing
  }
}
```

Листинг 3 – Запрос Paths

Большим преимуществом данного подхода заключается в том, что клиент имеет возможность указать, какие именно ему необходимы поля и GraphQL сервер вернет только указанные в запросе данные. Более того, таким образом можно указывать связанные внешним ключом записи, при этом вложенность запрашиваемых полей неограниченна:

```
query calculate($payloads: [CalculationPositionPayload], $carrierID: String) {
  calculate(payloads: $payloads, carrierID: $carrierID) {
    calculationPositions {
      vehiclesPassports {
        vehicle {
          ...
        }
        axisLoads {
          ...
        }
        cargoWeight
      }
    }
    ...
  }
}
```

Листинг 4 – Запрос Calculate

На рисунке 8 представлен общий вид архитектуры системы в виде диаграммы пакетов:

- Express – фреймворк, на котором строится веб-приложение. В данном случае используется для запуска Apollo Server.
- Apollo Server – сервер для обработки GraphQL запросов.
- Пакет Typedefs – представляет собой GraphQL схему, разбитую для удобства на 6 файлов, содержащих определения типов и запросов.
- Пакет Resolvers – пакет обработчиков заданных в схеме запросов.
- Пакет Models – пакет ORM моделей.
- Sequelize – ORM библиотека, необходимая для создания моделей.
- Пакет Calc – пакет, содержащий в себе классы и обработчики для проведения необходимых вычислений.

- Пакет Misc – необходим для различных вспомогательных инструментов. На данный момент содержит только класс Logger, предназначенный для сохранения подробной информации о ходе расчета.
- Angular Application – отдельное приложение, которое предоставляет интерфейс пользователя. Данный пакет рассмотрен более подробно в главе 3.6.

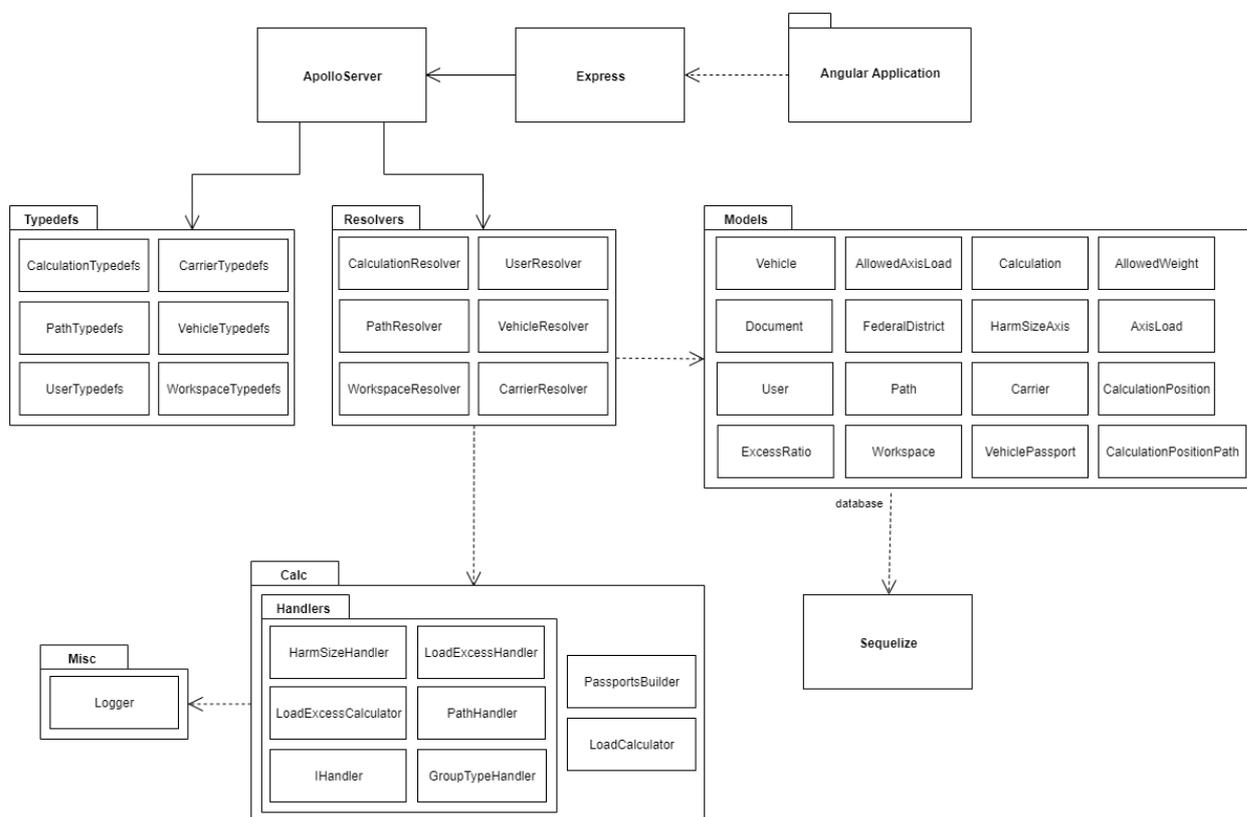


Рисунок 8 – Общий вид архитектуры системы

3.2 База данных

База данных в большинстве своем идентична модели предметной области (Рисунок 9):

- Users – представляет собой пользователей приложения.
- Workspaces – организации, в которую может входит множество пользователей.
- Calculations – созданные расчеты, принадлежащие конкретной организации.
- Carriers – перевозчики грузов.
- Calculations_positions – позиции в расчете. Т.к. в одном расчете могут быть несколько транспортных средств с разными маршрутами, то необходимо хранить внешние ключи к paths и vehicles_passports в отдельной сущности, и затем использовать их при отображение отчета.
- Calculations_positions_paths – промежуточная таблица для реализации связи M:N между таблицами calculations_positions и paths.

3.3 Пакет Models

Пакет Models представляет из себя пакет классов, которые наследуются от типа Model из пакета Sequelize. Каждая из моделей соответствует некоторой сущности в базе данных и описывает ее атрибуты и связи, например:

```
export class User extends Model {
  public id!: number;
  public username: string;
  public email: string;
  public passwordHash: string;
  public fullName: string;
}

User.init(
{
  id: {
    type: DataTypes.INTEGER.UNSIGNED,
    autoIncrement: true,
    primaryKey: true
  },
  username: {
    type: DataTypes.CHAR
  },
  email: {
    type: DataTypes.CHAR
  },
  passwordHash: {
    type: DataTypes.CHAR
  },
  fullName: {
    type: DataTypes.CHAR
  }
},
{
  tableName: "users",
  sequelize: database,
  charset: 'utf8',
```

```

    collate: 'utf8_unicode_ci'
  }
);
User.belongsTo(Workspace, {
  foreignKey: 'workspaceID',
  onDelete: 'CASCADE'
});

```

Листинг 5 – Инициализация модели User

Кроме создания класса, наследующегося от Model, также необходимо инициализировать саму модель, используя статический метод `init()` класса Model, в котором более подробно указываются необходимые типы, первичные ключи, название таблицы, кодировка и данные о подключении к самой таблице.

Такое разделение на определение класса и инициализацию самой таблицы на основе этого класса необходимо для поддержания строгой типизации в проекте. В свете того, что библиотека Sequelize изначально была разработана на Javascript, который является слабо типизированным языком, некоторые функции библиотеки все еще реализованы без использования типизации и соответственно необходимо использование таких методов, как `init()`, и в то же время использование расширения библиотеки, написанном на Typescript. Это делается для удобства написания программы, т.к. таким образом нет необходимости помнить о том, какие атрибуты есть у сущностей и точно знать, какого они типа.

3.4 Пакет Resolvers

Пакет Resolvers (Пакет распознавателей) содержит в себе весь набор необходимых распознавателей, реализующих работу запросов в GraphQL схеме (Рисунок 10).

В этот пакет входят:

- PathResolver, VehicleResolver, CarrierResolver – распознаватели запросов, которые относятся к пути, транспортному средству и перевозчику. Все три распознавателя реализуют одинаковые по смыслу запросы, которые запрашивают одну запись через ее id, все записи, вставку, редактирование и удаление записей.
- WorkspaceResolver – идентично предыдущим трем распознавателям, за исключением того, что запрос на получение всех организаций отсутствует.
- UserResolver – распознаватель, который отвечает за получение данных о пользователе, аутентификации пользователя в системе, регистрации пользователя и проверки текущей сессии.

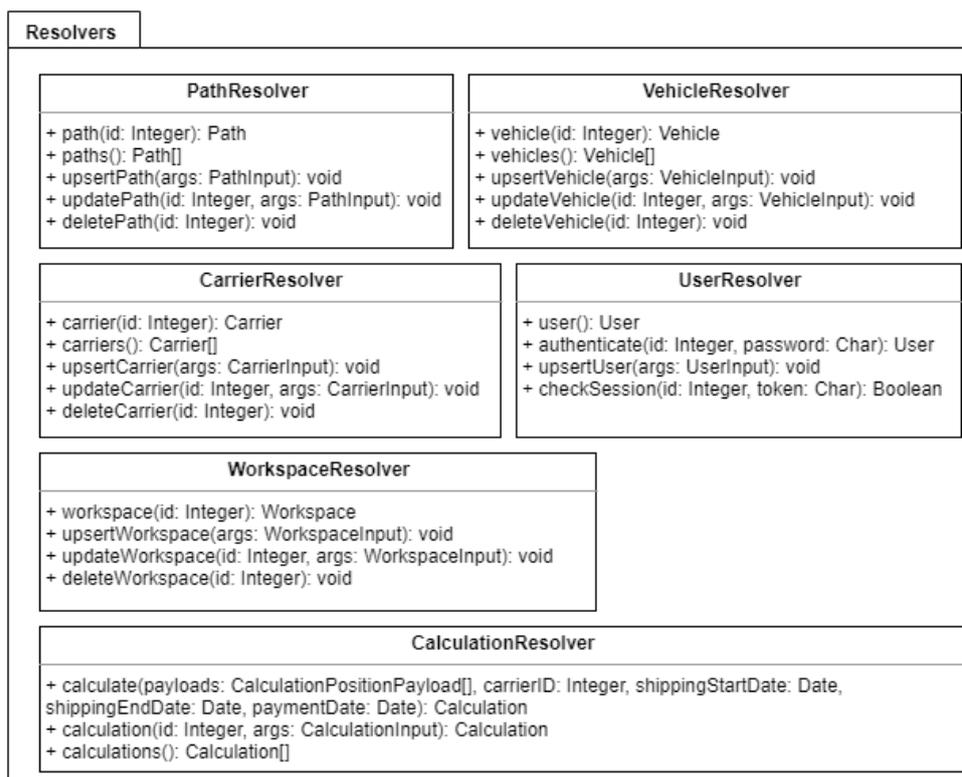


Рисунок 10 – Содержимое пакета Resolvers

- **CalculationResolver** – распознаватель, который ответственен за операции над расчетами: создание, получение данных об одном расчете по его идентификатору, получение всех расчетов.

3.5 Пакет Calc

Пакет Calc является набором классов, которые выполняют работу по вычислению осевых нагрузок, проверок превышений, расчета итогового вреда, формированию итогового отчета и созданию записей в базе данных (Рисунок 11).

Создание расчета начинается в классе **PassportsBuilder**, основной обязанностью которого является составление непосредственно самого расчета в несколько шагов с использованием калькулятора осевых нагрузок и специальных обработчиков, реализующих интерфейс **IHandler**.

Класс **LoadCalculator** предназначен для составления схемы осей и расчета осевых нагрузок транспортного средства и имеет следующие методы:

- **AddAxisScheme** – формирует схему осей. Необходимо для того, чтобы определить количество осей в передней и задней тележках и сформировать массив осевых нагрузок, которые в дальнейшем будут заполняться данными о превышении нагрузки, типу осей и причиненному вреду в рублях

- `CalculateAxisLoads` – метод, который вычисляет вес, приходящийся на каждую из осей, используя методику вычисления осевых нагрузок.

- `Calculate` – метод, который для каждой из переданных позиций расчета формирует схемы осей для каждого транспортного средства позиции с помощью приватного метода `addAxisScheme` и вычисляет для него осевые нагрузки с помощью метода `calculateAxisLoads`.

Интерфейс `IHandler` имеет единственный метод `handle`, который принимает на вход позицию расчета. Интерфейс существует для удобства реализации обработчиков, обязанностью каждого из которого будет являться только добавление нужной информации о расчете исходя из того, на каком этапе создания расчета находится приложение. Для того, чтобы иметь возможность быстро и эффективно вносить изменения в коде и в дальнейшем поддерживать приложение, решено разбить добавление такой информации в несколько этапов, каждый из которых соответствует определенному обработчику:

- `GroupTypeHandler` - определение типа расположения осей. Для каждой из осевой нагрузки транспортного средства определяется тип тележки, которой данная осевая нагрузка относится (Одиночная ось, сдвоенная ось, строенная ось, с количеством осей более трех).

- `LoadExcessHandler` - поиск допустимой осевой нагрузки для каждой фактической осевой нагрузки транспортного средства и вычисление процента превышения осевой нагрузки. На данном этапе приложение ищет запись в таблице `harm_sizes_axis`, у которого `distanceFrom` меньше расстояния от оси до оси, `distanceTo` больше расстояния от оси до оси и тип тележки соответствует типу тележки, в которую входит текущая осевая нагрузка.

- `HarmSizeHandler` - расчет размера причиненного вреда в рублях на каждую из осей транспортного средства. Здесь происходит поиск записи в таблице `federal_districts`, соответствующей федеральному округу, по которому проходит маршрут. Затем, если превышение текущей осевой нагрузки меньше 60%, то происходит поиск нужного тарифа, иначе – расчет по формулам, указанных в постановлении Правительства РФ [3].

- `PathHandler` – отвечает за расчет длины маршрута и расчета итогового вреда в рублях исходя из длины маршрута.

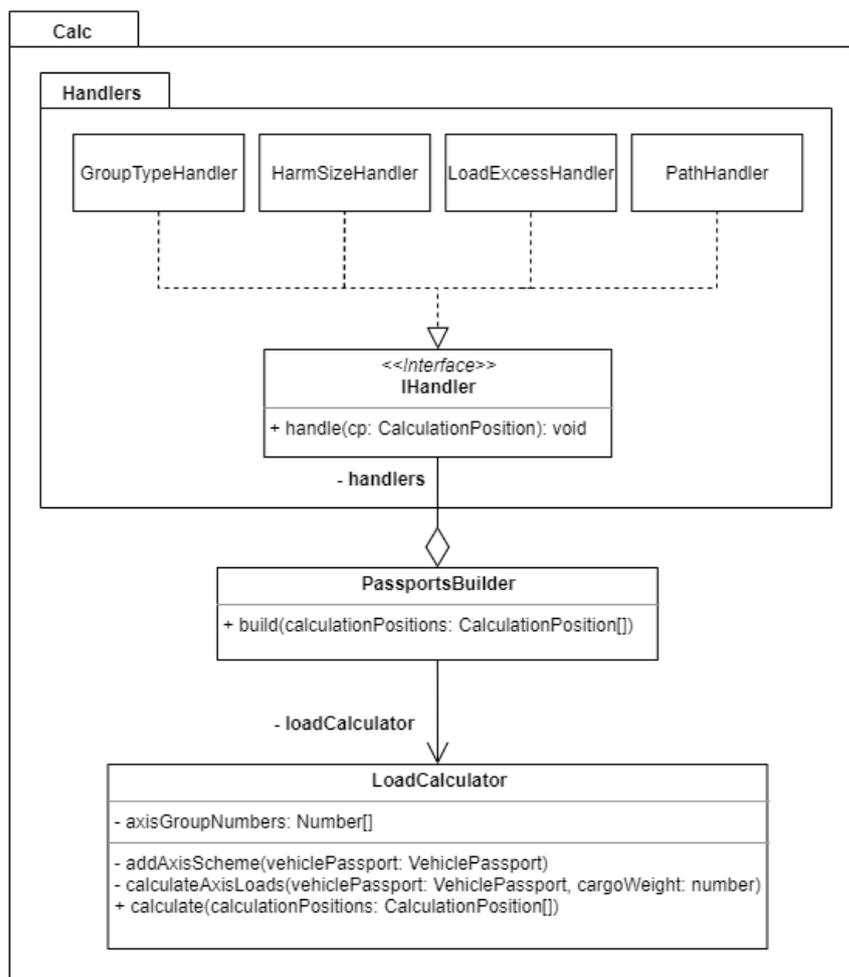


Рисунок 11 – Содержимое пакета Calc

3.6 Frontend приложение

Все, что было сказано об архитектуре системы в предыдущих главах, касалось backend – приложения, которое отвечало за расчеты, манипуляцию над данными и работой с базой данных.

В задачи данной работы также входит создание пользовательского интерфейса, которое решено сделать в виде отдельного web приложения с целью повышения эффективности разработки и удобства внесения изменений в код.

Приложение написано с использованием фреймворка Angular, который реализует шаблон проектирования MVC[12]. Angular приложение построено на модулях, главная цель которых – группирование компонентов и сервисов, которые связаны друг с другом. В компоненте реализуется логика представления и само представление в виде включаемого в атрибуты компонента html документа. Сервис реализует обмен данными с backend - приложением. Архитектура приложения представлена на рисунке 12.

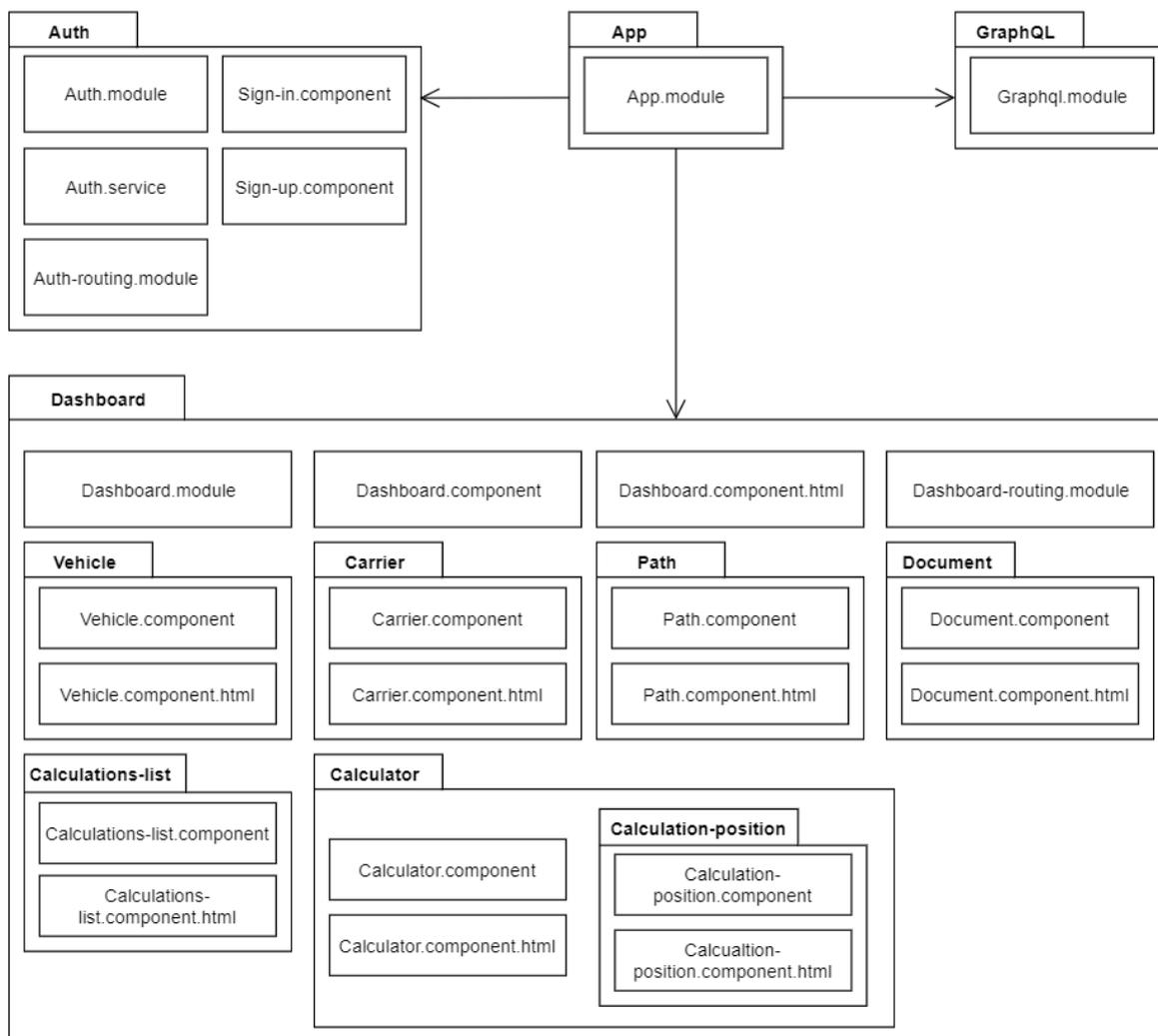


Рисунок 12 – Общий вид архитектуры Frontend приложения

- Пакет `App` – содержит в себе единственный модуль `app`, который является точкой входа в приложение и группирует все прочие модули.
- Пакет `GraphQL` – так же содержит в себе единственный модуль, необходимый для инициализации подключения к `GraphQL` серверу на стороне `backend` - приложения.
- Пакет `Auth` – содержит сервисы и представления для совершения аутентификации и регистрации в приложении.
- Пакет `Dashboard` – панель управления, включающая в себе все представления для работы с данными и созданием расчетов.

Панель управления содержит в себе следующие модули:

- `Vehicle` – компонент, ответственный за поиск, удаление, редактирование транспортных средств (Рисунок 13), имеющий следующие атрибуты:
 - `Type` – тип выбранного транспортного средства (Тягач / прицеп)

- CargoWeightInput – наличие возможности ввести вес груза. В случае, если мы имеем дело с тягачом, то своего груза он иметь не может, поэтому необходимо заблокировать соответствующее поле ввода.

- Types – массив типов транспортных средств, необходимый для инициализации элемента SelectButton в представлении.

- Vehicles – массив полученных от бэкенд – приложения транспортных средств.

- Filtered – массив отфильтрованных по введенным в строку поиска символам транспортных средств.

- Vehicle – выбранное транспортное средство.

- EditMode – режим редактирования. Необходим для изменения видимости некоторых кнопок во время редактирования (При редактировании сущности не должно быть кнопок «Удалить» и «Обновить», но должны быть «Создать» и «Отменить»).

- CargoWeight – вес груза.

Так же компонент содержит методы:

- NgOnInit – метод инициализации компонента, который вызывается сразу после инициализации его свойств. Здесь происходит предварительное получение данных о всех транспортных средствах.

- GetData – метод получения данных. Содержит в себе GraphQL запрос с необходимыми полями.

- Filter – метод фильтрации транспортных средств по уже введенным символам в поле поиска.

- Update – метод, посылающий запрос на обновление записи в базе данных.

- Upsert – метод, посылающий запрос на вставку новой записи.

- Delete – метод, посылающий запрос на удаление записи.

- EditModeOn – метод для включения режима редактирования.

- EditModeOff – метод для выключения режима редактирования.

Vehicle.component
+ type: String + cargoWeightInput: Boolean + types: SelectItem[] + vehicles: Vehicle[] + filtered: Vehicle[] + vehicle: Vehicle + editMode: Boolean + cargoWeight: Integer
+ ngOnInit(): void + getData(): void + filter(event: Object): void + update(): void + upsert(): void + delete(): void + editModeOn(): void + editModeOff(): void

Рисунок 13 – Компонент Vehicle

• Path – компонент, ответственный за поиск, удаление, редактирование путей. Во многом повторяет компонент Vehicle.component, за исключением отсутствия нескольких полей (Рисунок 14).

Path.component
+ types: SelectItem[] + paths: Path[] + filtered: Path[] + vehicle: Vehicle + editMode: Boolean
+ ngOnInit(): void + getData(): void + filter(event: Object): void + update(): void + upsert(): void + delete(): void + editModeOn(): void + editModeOff(): void

Рисунок 14 – Компонент Path

• Carrier – компонент, необходимый для работы с данными о перевозчиках. Полностью повторяет компонент Path.component за исключением отсутствия поля types.

- Document – компонент, ответственный за отображение документа (Рисунок 15).

Имеет следующие поля:

- Calculation – переданный компонентом-родителем расчет, который будет отображаться.
- CalculationPositions – отдельное поле для массива позиций расчета, необходимый для удобства формирования представления.
- Carrier – отдельное поле для данных о перевозчике.

Также имеет методы:

- NgOnInit – инициализация полей calculationPositions и carrier.
- RoundNumber – округление чисел. Используется в представлении.
- GetPathsLength – получение длины всего в маршрута. Используется в представлении.
- GetPricePerKm – получение общего вреда в рублях. Используется в представлении.

Document.component
+ calculation: Calculation + calculationPositions: CalculationPosition[] + carrier: Carrier
+ ngOnInit(): void + roundNumber(n: Integer): Integer + getPathsLength(paths: Path[]): number + getPricePerKm(vehiclePassports: VehiclePassport[]): number

Рисунок 15 – Компонент Document

- Calculations-list – компонент, предназначенный для отображения списка расчетов с полями (Рисунок 16):

- Calculations – массив расчетов, содержащих в себе только необходимые для отображения в списке поля.
- CurrentCalculation – выбранный расчет.
- Cols – массив колонок списка, соответствующих определенным полям расчета (в данном случае номеру расчета, дате создания и названию организации получателя).

Имеет методы:

- NgOnInit – предварительное получение данных о расчетах для отображения в списке.
- GetCalculation – получение данных о выбранном расчете из списка.

Calculations-list.component
+ calculations: Calculation[] + currentCalculation: Calculation + cols: Object[]
+ ngOnInit(): void + getCalculation(id: String): Calculation

Рисунок 16 – Компонент Document

• Calculator – компонент, отображающий форму для создания расчета. Данное представление включает в себя компонент Calculation-position, количество которых регулируется с помощью методов addCalculationPosition и removeCalculationPosition, которые активируются при нажатии соответствующих кнопок. Метод makeCalculation отвечает за отправку запроса на создание расчета с данными, полученных от формы (Рисунок 17). Имеет следующие поля:

- CalculationPositionsForms – массив компонентов CalculationPosition. Необходим для возможности получения данных от вложенных представлений.
- CarrierForm – компонент CarrierComponent так же повторно используется в данном представлении.
- Logs – содержит в себе подробности о процессе расчета.
- Calculation – поле, в которое возвращается результат расчета для отображения.
- PositionsCount – количество позиции расчета.

Calculator.component
+ calculationPositionsForms: CalculationPosition[] + carrierForm: CarrierComponent + logs: String[] + calculation: Calculation + positionsCount: Integer[]
+ makeCalculation(): void + addCalculationPosition(): void + removeCalculationPosition(): void

Рисунок 17 – Компонент Calculator

• Calculation-position – компонент, который отображает комбинацию представлений Vehicle и Path и предоставляет элементы интерфейса для контроля количества транспортных средств и путей с помощью методов addVehicle, removeVehicle, addPath и removePath (Рисунок 18).

Calculation-position.component
+ vehiclesForms: VeihcleComponent[] + pathsForms: PathComponent[] + vehicles: Vehicle[] + paths: Path[]
+ addVehicle(): void + removeVehicle(): void + addPath(): void + removePath(): void

Рисунок 18 – Компонент Calculation-position

4 Реализация

В данной главе на рисунках 19-25 представлен разработанный функционал приложения.

Страница создания расчета состоит из следующих разделов, каждый из которых является повторно используемым компонентом из соответствующей вкладки в меню пользователя (Рисунки 19, 20, 21, 22):

1. Добавление группы транспортных средств. Группа транспортных средств может содержать как отдельный грузовик, так и совокупность тягача (грузовика) с неограниченным количеством прицепов. Так же группа транспортных средств включает в себя массу груза. Все характеристики транспортных средств доступны для редактирования, сохранения и удаления без необходимости переходить по отдельным вкладкам.

2. Маршрут. В расчете их так же может быть неограниченное количество, т.к. маршрут может состоять из некоторой совокупности путей, которые могут использоваться повторно.

3. Грузоперевозчик.

ИмяПрофиля

Тягачи Прицепы Маршруты Грузоперевозчики Расчеты

Расчет

Группы транспортных средств

Транспортные средства

ВОЛЬВО FM TRUCK 6x6

Добавить Обновить Удалить

Название: ВОЛЬВО FM TRUCK 6x6 | Тип: Тягач | Вес снаряженного: 10819 | Макс. вес груза: 30181 | Вес полностью груз.: 41000 | Длина: 0

Ширина: 0 | Высота: 0 | Дистанция между осями: 3.7, 1.37

Двухос./л/вмю - ПЕРЕД Двухос./л/вмю - ЗАД

Нагрузки на оси:

Передняя ось снаряженного: 5815 | Задняя ось снаряженного: 5004 | Передняя ось полностью грузевого: 8600

Задняя ось полностью грузевого: 32400

Седельное устройство

Начните писать название средства

Добавить

Вес груза: 0

+

-

Маршрут

Тестовый маршрут

Добавить Обновить Удалить

Название: Тестовый маршрут | Протяженность: 1000 | 10

Проморзание грунта

+

-

Рисунок 19 – Форма создания расчета

ИмяПрофиля

Тягачи Прицепы Маршруты Грузоперевозчики Расчеты Расчет

ВОЛЬВО FM TRUCK 6x6 Добавить Обновить Удалить

Название: ВОЛЬВО FM TRUCK 6x6 Тягач Прицеп
 Вес снаряженного: 10819
 Макс. вес груза: 30181
 Вес полностью груз.: 41000
 Длина: 0

Ширина: 0
 Высота: 0
 Дистанции между осями: 3.7, 1.37
 Двускатн./пневмо - ПЕРЕД
 Двускатн./пневмо - ЗАД

Нагрузки на оси:
 Передняя ось снаряженного: 5815
 Задняя ось снаряженного: 5004
 Передняя ось полностью грузеного: 8600

Задняя ось полностью грузеного: 32400
 Седельное устройство

Рисунок 20 – Информация о тягачах

ИмяПрофиля

Тягачи Прицепы Маршруты Грузоперевозчики Расчеты Расчет

Создать Отменить

Название: Тягач Прицеп
 Вес снаряженного:
 Макс. вес груза:
 Вес полностью груз.:
 Длина:

Ширина:
 Высота:
 Дистанции между осями:
 Двускатн./пневмо - ПЕРЕД
 Двускатн./пневмо - ЗАД

Нагрузки на оси:
 Передняя ось снаряженного:
 Задняя ось снаряженного:
 Передняя ось полностью грузеного:

Задняя ось полностью грузеного:
 Седельное устройство

Рисунок 21 – Незаполненная форма с данными о тягаче

ИмяПрофиля

Тягачи Прицепы Маршруты Грузоперевозчики Расчеты Расчет

SPE Добавить Обновить Удалить

SPECPRICEP

Название: Тягач Прицеп
 Вес снаряженного:
 Макс. вес груза:
 Вес полностью груз.:
 Длина: 3

Ширина:
 Высота:
 Дистанции между осями:
 Двускатн./пневмо - ПЕРЕД
 Двускатн./пневмо - ЗАД

Нагрузки на оси:
 Передняя ось снаряженного:
 Задняя ось снаряженного:
 Передняя ось полностью грузеного:

Задняя ось полностью грузеного:
 Седельное устройство

Рисунок 22 – Информация о прицепах

Во всех разделах присутствует возможность поиска, при этом будут доступны подсказки исходя из первых введенных пользователем символов (Рисунок 23).

Рисунок 23 – Информация о грузоперевозчиках

Список созданных расчетов содержит все созданные пользователем расчеты, которые могут быть удалены либо распечатаны (Рисунок 24):

Номер	Дата создания	Название орг-ии получателя
1	31.05.2020 16:56:42	Имя компании получателя

РАСЧЕТ №1 от платы за счет возмещения вреда, причиняемого транспортным средства, осуществляющими перевозки тяжеловесных грузов по автомобильным дорогам Имя компании получателя

Получатель:	Имя компании получателя	Счет:	Счет получателя	ИНН/КПП:	ИНН получателя / КПП получателя
Банк получателя:	Банк получателя	БИК:	БИК получателя	Корр.счет:	Корреспондентский счет получателя
Назначение платежа:	Тестовая должность менеджера				
	Тестовая должность менеджера		Тестовое Полное Имя		

Тестовый перевозчик: Тестовый адрес 12, 43 +9999999999
 Перевозчик груза: Тестовый перевозчик
 Банковские реквизиты: Тестовые банковские реквизиты перевозчика
 Период перевозок: "period"
 Маршрут движения: Тестовый маршрут

№ п/п	Марка транспортного средства	Кол-во машин	Маршрут (км)	Кол-во рейсов	Масса с грузом (т)			Тариф на 100км.	Длина		Ширина		Высота		Стоимость, руб.			
					факт.	норм.	перегруз		факт	доп.	факт	доп.	факт	доп.	за ед.	* на кол-во рейсов	* на компенс. индекс	
1	ВОЛЬВО FM TRUCK 6x6 + SPECPRICER	2	Тестовый маршрут		9716	9000	8	2747	0	0	0	0	0					
					8301	15000	0	0	19.06	3	1.25							
					7750	6500	19	43920										
					7750	6500	19	43920										
					7750	6500	19	43920										
		1000		1			178427						178427	178427	178427			

Удалить Печать

Рисунок 24 – Список созданных расчетов

При генерации расчета пользователю представляется ход расчета для обеспечения достоверности и поиска несоответствий в случае ошибки. Это важно для тестирования работоспособности приложения (Рисунок 25).

+
-

Перевозчик

Тестовый перевозчик
[Добавить](#)
[Обновить](#)
[Удалить](#)

Название Тестовый перевозчик	Почтовый индекс 600000	Адрес Тестовый адрес 12, 43	Телефонный номер +9999999999	ИНН 000000000000	КПП 000000000000
Должность руководителя Тестовая должность мет	ФИО руководителя Тестовое Полное Имя	Банковские реквизиты перевозчика Тестовые банковские ре	Название компании получателя Имя компании получате	ИНН получателя ИНН получателя	КПП получателя КПП получателя
Название банка Банк получателя	Счет получателя Счет получателя	БИК получателя БИК получателя	ОГРН получателя ОГРН получателя		

Сгенерировать отчет

Схема осей тягача: 0-----0-0-0-0-
 Количество осей на передней тележке = 1
 Количество осей на задней тележке = 4
 Вес тягача с грузом: 96500
 Коэффициент нагрузки на заднюю тележку при массе 80000 = 0.6666666666666667
 Вес на заднюю тележку = 64333.333333333333
 Вес на переднюю тележку = 32166.666666666667
 Нагрузка на 0 ось = 32166.666666666667
 Нагрузка на 1 ось = 16083.333333333332
 Нагрузка на 2 ось = 16083.333333333332
 Нагрузка на 3 ось = 16083.333333333332
 Нагрузка на 4 ось = 16083.333333333332
 Схема осей тягача: 0----0-0-
 Количество осей на передней тележке = 1
 Количество осей на задней тележке = 2
 Вес тягача с грузом: 42985.666666666667
 Коэффициент нагрузки на заднюю тележку при массе 32166.66666666667 = 0.8118055187227602
 Вес на заднюю тележку = 34896.001425976996
 Вес на переднюю тележку = 8089.665240689672
 Нагрузка на 0 ось = 8089.665240689672
 Нагрузка на 1 ось = 17448.000712988498
 Нагрузка на 2 ось = 17448.000712988498
 Итоговый ущерб транспортного средства SPECPRICEP по осям

Рисунок 25 – Подробная информация о ходе расчета

ЗАКЛЮЧЕНИЕ

В рамках данной работы выполнены поставленные для достижения цели и задачи. На этапе анализа требований изучены аналоги, сформулированы функциональные требования к приложению и проведён анализ предметной области. На этапе проектирования web-приложения выбраны инструменты разработки, спроектирована архитектура системы, завершена разработка ключевого функционала приложения. В будущем планируется:

- Автоматическое тестирование
- Доработка модуля аутентификации
- Написание собственного механизма миграций
- Возможность создания маршрута с использованием онлайн карты
- Учет неравномерно распределенного груза
- Реализация графического представления транспортных средств с целью улучшения наглядности

ЛИТЕРАТУРА

1. Грузовые перевозки в России: обзор текущей статистики [Электронный ресурс] - URL: <https://ac.gov.ru/archive/files/publication/a/24196.pdf> (дата обращения 2.9.2019)
2. Постановление Правительства РФ от 15.04.2011 N 272 (ред. от 26.03.2020) "Об утверждении Правил перевозок грузов автомобильным транспортом" (Дата обращения 17.10.2019)
3. Постановление Правительства РФ от 31 января 2020 г. № 67 “Об утверждении Правил возмещения вреда, причиняемого тяжеловесными транспортными средствами, об изменении и признании утратившими силу некоторых актов Правительства Российской Федерации” (Дата обращения 1.2.2020)
4. Typescript - JavaScript that scales. [Электронный ресурс] - URL: <https://www.typescriptlang.org/> (Дата обращения 13.05.2020)
5. Node.js – JavaScript runtime built on Chrome's V8 JavaScript engine. [Электронный ресурс] – URL: <https://nodejs.org/en/> (Дата обращения 27.11.2019)
6. Express - Node.js web application framework. [Электронный ресурс] – URL: <https://expressjs.com/> (Дата обращения 27.11.2019)
7. GraphQL - A query language for your API [Электронный ресурс] – URL: <https://graphql.org/> (Дата обращения 27.11.2019)
8. The Apollo Data Graph Platform [Электронный ресурс] – URL: <https://www.apollographql.com/> (Дата обращения 27.11.2019)
9. Sequelize - promise-based Node.js ORM [Электронный ресурс] – URL: <https://sequelize.org/> (Дата обращения 27.11.2019)
10. Angular - One framework. Mobile & desktop. [Электронный ресурс] – URL: <https://angular.io/> (Дата обращения 27.11.2019)
11. Фаулер М. UML. Основы, 3-е издание / М. Фаулер : пер. с англ. Петухов А. – СПб: Символ-Плюс, 2004. – 192 с.
12. Википедия – свободная энциклопедия: Model-View-Controller. [Электронный ресурс] – URL: <https://ru.wikipedia.org/wiki/Model-View-Controller> (Дата обращения 13.05.2020)

Приложение А. Допустимая нагрузка на ось транспортного средства

Расположение осей транспортного средства	Расстояние между сближенными осями, м	Допустимая нагрузка на ось колесного транспортного средства для автомобильной дороги, тс		
		для автомобильной дороги, рассчитанной на нормативную нагрузку 6 тс на ось	для автомобильной дороги, рассчитанной на нормативную нагрузку 10 тс на ось	для автомобильной дороги, рассчитанной на нормативную нагрузку 11,5 тс на ось
Одиночная ось	свыше 2,5	5,5 (6)	9 (10)	10,5 (11,5)
Группа сближенных сдвоенных осей	до 1 (включительно)	8 (9)	10 (11)	11,5 (12,5)
	свыше 1 до 1,3 (включительно)	9 (10)	13 (14)	14 (16)
	свыше 1,3 до 1,8 (включительно)	10 (11)	15 (16)	17 (18)
	свыше 1,8 до 2,5 (включительно)	11 (12)	17 (18)	18 (20)
Группа сближенных строенных осей	до 1 (включительно)	11 (12)	15 (16,5)	17 (18)
	свыше 1 до 1,3 (включительно)	12 (13)	18 (19,5)	20 (21)
	свыше 1,3 до 1,8 (включительно)	13,5 (15)	21 (22,5)	23,5 (24)
	свыше 1,8 до 2,5 (включительно)	15 (16)	22 (23)	25 (26)
Группа сближенных осей с количеством осей более 3 (не более 2 односкатных или двускатных колеса на	до 1 (включительно)	3,5 (4)	5 (5,5)	5,5 (6)
	свыше 1 до 1,3 (включительно)	4 (4,5)	6 (6,5)	6,5 (7)
	свыше 1,3 до 1,8	4,5 (5)	6,5 (7)	7,5 (8)

Продолжение приложения А

оси)	(включительно)			
	свыше 1,8 до 2,5 (включительно)	5 (5,5)	7 (7,5)	8,5 (9)
Группа сближенных осей с количеством осей 2 и более (по 4 (включительно) и более односкатных или двускатных колеса на оси)	до 1 (включительно)	6 (6,5)	9,5 (10)	11 (11,5)
	свыше 1 до 1,3 (включительно)	6,5 (7)	10,5 (11)	12 (12,5)
	свыше 1,3 до 1,8 (включительно)	7,5 (8)	12 (12,5)	14 (14,5)
	свыше 1,8 до 2,5 (включительно)	8,5 (9)	13,5 (14)	16 (16,5)

Отчет о проверке на заимствования №1



Автор: Ivanov Vladislav ID: 7617534

Проверяющий: Ivanov Vladislav ID: 7617534

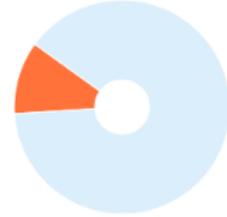
Отчет предоставлен сервисом «Антиплагиат»- <http://users.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 1
Начало загрузки: 08.06.2020 15:07:36
Длительность загрузки: 00:00:02
Имя исходного файла: Диплом.pdf
Название документа: ВКР Иванов В.Д. 1461
Размер текста: 1 кБ
Символов в тексте: 48702
Слов в тексте: 5431
Число предложений: 293

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 08.06.2020 15:07:38
Длительность проверки: 00:00:07
Комментарии: не указано
Модули поиска: Модуль поиска Интернет



ЗАИМСТВОВАНИЯ

10,54%

САМОЦИТИРОВАНИЯ

0%

ЦИТИРОВАНИЯ

0%

ОРИГИНАЛЬНОСТЬ

89,46%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.

Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.

Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общепотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Ссылка	Актуален на	Модуль поиска
[01]	3,15%	Об утверждении Правил перевозок грузов автомобильным транспортом (с изменениями на 31 янв...	http://docs.cntd.ru	01 Апр 2020	Модуль поиска Интернет
[02]	0,98%	Проект Постановления Правительства Российской Федерации "Об утверждении Правил возмещен...	http://garant.ru	22 Окт 2019	Модуль поиска Интернет
[03]	1,08%	О внесении изменений в некоторые акты Правительства Российской Федерации по вопросам пере...	http://docs.cntd.ru	17 Фев 2016	Модуль поиска Интернет

Еще источников: 17

Еще заимствований: 5,31%