

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Вычислительные машины, комплексы, системы и сети»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА МАГИСТРА**

на тему

**Автоматизированная система для
оптимизации доставки грузов**

Студент

 3.06.2020
(подпись, дата)

Иванов Сергей Алексеевич
(ФИО полностью)

Руководитель

 3.06.2020
(подпись, дата)

А.В. Дубравин
(фамилия, инициалы)

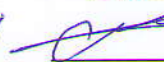
Нормоконтролёр

 4.06.2020
(подпись, дата)

А.В. Кучин
(фамилия, инициалы)

Рецензент

доцент ПензГТУ
(должность, место работы)

 5.06.2020
(подпись, дата)

С.В. Печерский
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 3.06.2020 г. № 12.)

Заведующий кафедрой


(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой

15.06, 2020 № 4 (протокол заседания ГЭК от отсутствует)

Секретарь ГЭК


(подпись)

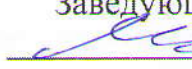
Е.И. Гурин
(фамилия, инициалы)

Пенза 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой ВТ
 М.А. Митрохин
« 8 » ноября 20 18 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
МАГИСТРА**

1. Студент Иванов Сергей Алексеевич гр. 18ВВм1 факультета ВТ

2. Тема работы «Автоматизированная система для оптимизации доставки грузов»

Тема утверждена приказом ПГУ № 1012/о от " 08 " ноября 20 18 г.

3. Руководитель магистерской диссертации к.т.н. доцент кафедры ВТ Дубравин А.В.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать автоматизированную систему для оптимизации доставки грузов, позволяющую
рассчитывать и отображать маршрут на цифровой географической карте в соответствии с
пользовательскими входными данными.

Обоснование актуальности разработки; обзор и сравнение существующих аналогов; обоснование выбора используемых при разработке технологий и программных средств; описание разработанного программного и информационного обеспечения; схема БД; схемы основных алгоритмов; описание пользовательского интерфейса приложения; исходный код приложения; скриншоты.

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ ТЗ	10%	26.01.2019	<i>Д.И.Третьяков</i>
2	Разработка ПО	50%	1.12.2019	<i>Д.И.Третьяков</i>
3	Отладка и тестирование	30%	1.03.2020	<i>Д.И.Третьяков</i>
4	Оформление ПЗ	10%	15.05.2020	<i>Д.И.Третьяков</i>
5	Нормоконтроль		30.05.2020	<i>Д.И.Третьяков</i>

Руководитель магистерской диссертации Дубравин А.В.: 3.10.2018 Дубравин А.В.
(подпись, дата)

Задание к исполнению принял студент И 3.10.2018 Иванов С.А.
(подпись, дата)

Работу к защите допустить
Декан факультета _____ Фионова Л.Р.
(подпись, дата)

Реферат

Пояснительная записка 96 листов, 1 таблица, 32 рисунка.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДЛЯ ОПТИМИЗАЦИИ ДОСТАВКИ ГРУЗОВ, ПОСТРОЕНИЕ МАРШРУТОВ, ЛОГИСТИКА, ГЕОИНФОРМАЦИОННЫЙ ВЕБ-СЕРВИС, JAVA, JAVASCRIPT.

Цель работы - разработка автоматизированной системы для оптимизации доставки грузов, позволяющей прокладывать оптимальный маршрут с учетом входных пользовательских данных, а также предоставляющей сведения о построенном маршруте.

Содержится описание и применение разработанных алгоритмов для оптимизации расчета маршрутов.

Объектами исследования являются технологии и наработки в сфере разработки геоинформационных веб-сервисов, в функционал которых входит возможность проложения маршрута с учетом входных пользовательских адресов.

В результате проделанной работы была разработана автоматизированная система для оптимизации доставки грузов, которая позволяет построить оптимизированный маршрут с использованием параметров, заданных пользователем. А также может быть использована для решения логистических задач.

Содержание

Введение.....	5
1 Анализ предметной области	6
1.1 Постановка задачи.....	6
1.2 Обзор и сравнение существующих аналогов	6
1.2.1 Google Карты	7
1.2.2 Яндекс Карты.....	7
1.2.3 2ГИС	8
1.2.4 Poncy.su	9
1.2.5 ISSA.ru.....	10
1.3 Обоснование актуальности разработки	11
1.4 Обоснование выбора используемых при разработке технологий и программных средств	13
1.5 Вывод.....	16
2 Проектирование структуры системы	17
2.1 Технология взаимодействия веб – ресурсов.....	17
2.2 Проектирование информационного обеспечения для серверной части	20
2.3 Описание пользовательского интерфейса приложения	20
2.4 Проектирование базы данных разрабатываемой системы.....	25
2.5 Вывод.....	26
3 Разработка ПО	27
3.1 Алгоритмы оптимизации маршрута.....	27
3.1.1 Алгоритм оптимизации маршрута с учетом расстояния	27
3.1.2 Алгоритм оптимизации маршрута с учетом веса грузов	29
3.2 Разработка ПО системы.....	30
3.2.1 Программное обеспечение для серверной части	30
3.2.2 Программное обеспечение для клиентской части.....	31

3.3	Вывод.....	31
4	Тестирование ПО	32
4.1	Руководство по использованию автоматизированной системы	32
4.2	Вывод.....	37
	Заключение	38
	Список использованных источников	39
	Приложение А	41
	Приложение Б	95

Введение

Современные технологии помогают в различных сферах деятельности. Благодаря им организация и оптимизация доставки грузов происходит на много быстрее. Это позволяет создавать оптимальные маршруты и эффективно планировать загрузку транспортных средств и оперативно доставлять грузы конечному потребителю, а всё благодаря автоматизации процесса при помощи программы для транспортной логистики. Отчасти, эти проблемы решают тематические веб-сервисы и веб-сайты, которые позволяют рассчитать и построить оптимальный маршрут с учетом адресов, введенных пользователем. Рассмотрим подробнее, что из себя представляют веб-сайт и веб-сервис.

Сайтом называется любая совокупность веб-страниц, объединенная общим доменным именем. Сайты, в зависимости от цели создания, функционала, типа размещаемого контента, могут быть разными: сайты-визитки, корпоративные сайты, блоги, социальные сети, видео- или фото-хостинги, интернет-магазины и др. [1].

Веб-служба, веб-сервис (англ. Web service) — идентифицируемая веб-адресом программная система со стандартизированными интерфейсами.

Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах (SOAP, XML-RPC и т. д.) и соглашениях (REST). Веб-служба является единицей модульности при использовании сервис-ориентированной архитектуры приложения.

В обиходе веб-сервисами называют услуги, оказываемые в Интернете. В этом употреблении термин требует уточнения, идёт ли речь о поиске, веб-почте, хранении документов, файлов, закладок и т. п. Такими веб-сервисами можно пользоваться независимо от компьютера, браузера или места доступа в Интернет [2].

1 Анализ предметной области

1.1 Постановка задачи

Целью работы является разработка веб-сервиса для оптимизации доставки грузов. Оптимизация достигается благодаря специальному автоматизированному алгоритму построения маршрута с учетом входных пользовательских параметров. Для расчета маршрута планируется разработка алгоритма, где основными критериями являются: адрес точки отправления, адрес(а) точек назначения, тип выбранной оптимизации и способ передвижения по построенному маршруту, а также, при выбранном типе оптимизации с учетом веса груза, в качестве необходимых параметров выступают вес посылок для каждого соответствующего адреса точки назначения и грузоподъемность средства передвижения. В результате работы этого алгоритма пользователю будет представлен маршрут передвижения, показанный на геоинформационной карте, основные сведения о нем, такие как общее расстояние и приблизительное время преодоления построенного пути, а также подробный обзор маршрута с упоминанием каждого ключевого изменения направления движения.

1.2 Обзор и сравнение существующих аналогов

Целью этой части работы является обзор аналогов, выявление недостатков у существующих систем и предложения по устранению недостатков.

Учитывая наличие современных средств связи, информационных сетей (в частности Internet), найти несколько аналогичных, по выполняемым задачам, программных средств достаточно просто.

Для поиска аналогичных систем, выполняющих схожие функции, использовались различные поисковые системы, такие как Google, Yandex.

В результате были найдены следующие наиболее схожие с разрабатываемой системой веб-сервисы: Google Карты, Яндекс Карты, 2ГИС, Poncy.su, ISSA.ru.

1.2.1 Google Карты

Google Карты – это система, представляющая собой карту и спутниковые снимки планеты Земля. Также, в данную систему интегрирован бизнес-справочник и карта автомобильных дорог с поиском маршрутов [3].

Принцип работы: Попадая на главную страницу, пользователь может начать поиск вводом названия искомого объекта или его адреса в строку поиска. После этого появится панель с результатами поиска, которые также будут отмечены на карте. Работая с появившейся панелью, пользователь может получить более детальные данные о найденных объектах. К таким данным относятся: оценка найденного объекта, его точный адрес и контакты для связи.

К достоинствам можно отнести: современный дизайн, постоянное наполнение сервиса информацией о заведениях и предприятиях.

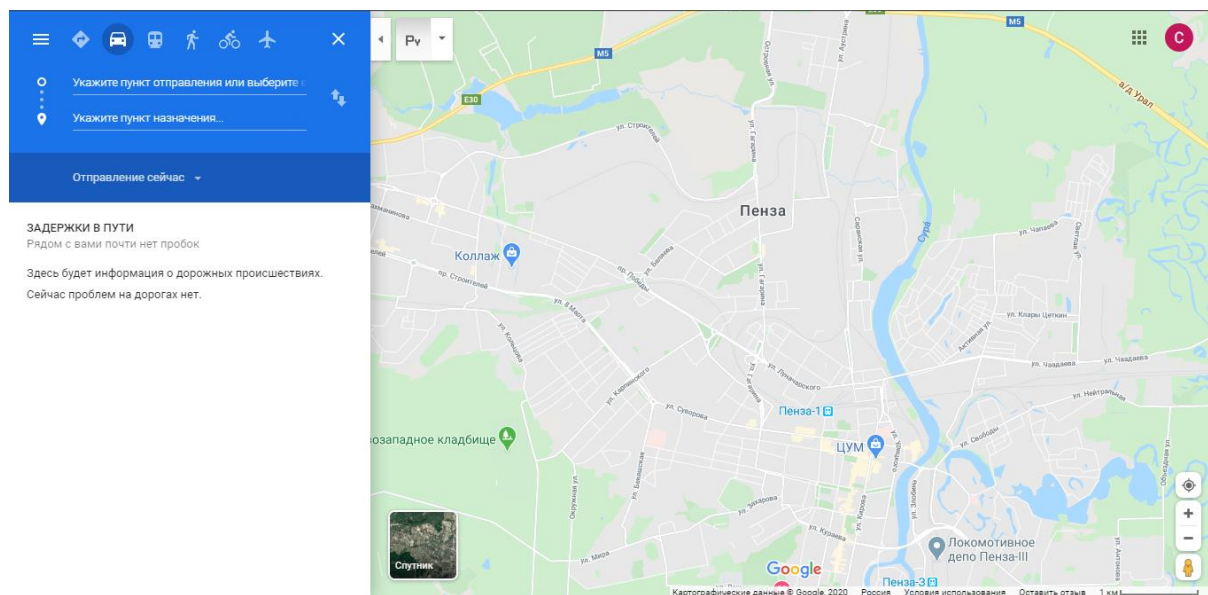


Рисунок 1 – Google Карты

1.2.2 Яндекс Карты

Яндекс Карты — поисково-информационная картографическая служба Яндекса, включающая в себя поиск по карте, информацию о пробках, прокладку маршрутов и панорамы улиц крупных и других городов [4].

Принцип работы: Попадая на главную страницу, пользователь может начать поиск вводом названия искомого объекта или его адреса в строку

поиска. После этого появится панель с результатами поиска, которые также будут отмечены на карте. Работая с появившейся панелью, пользователь может получить более детальные данные о найденных объектах. К таким данным относятся: оценка найденного объекта, его точный адрес и контакты для связи.

К достоинствам можно отнести: современный дизайн, постоянное наполнение сервиса информацией о заведениях и предприятиях.

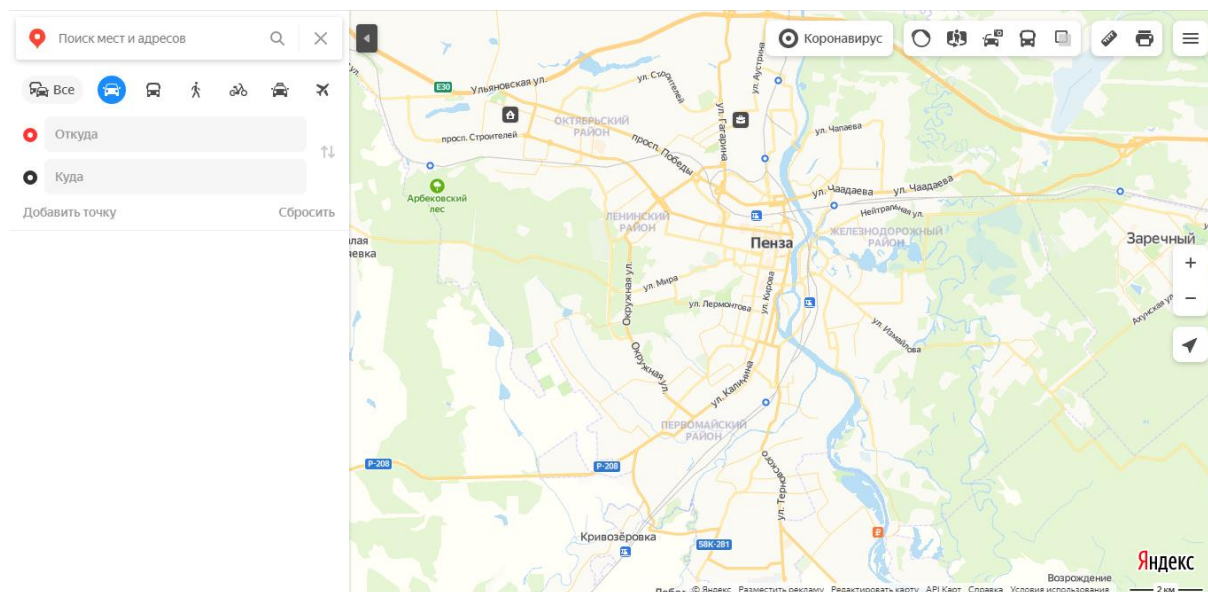


Рисунок 1 – Яндекс Карты

1.2.3 2ГИС

2ГИС — это справочник организаций и карта города, где пользователь может найти нужную ему организацию на карте или узнать, какие именно компании находятся в интересующем его здании [5].

Принцип работы: Попадая на главную страницу, необходимо выбрать город, в котором необходимо выполнить поиск объекта. Затем пользователь может начать поиск вводом названия искомого объекта или его адреса в строку поиска. После этого появится панель с результатами поиска, которые также будут отмечены на карте. Работая с появившейся панелью, пользователь может получить более детальные данные о найденных объектах. К таким данным относятся: оценка найденного объекта, его точный адрес и контакты для связи.

К достоинствам можно отнести: сервис обладает современным дизайном, постоянным наполнением сервиса информацией о заведениях и предприятиях.

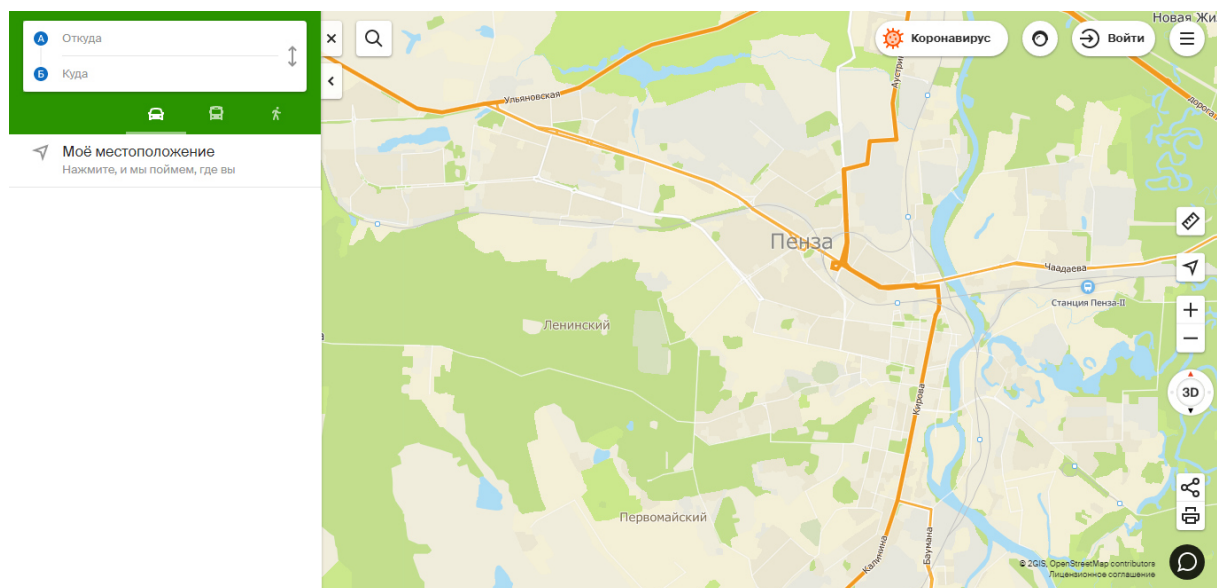


Рисунок 3 – 2ГИС

1.2.4 Poncy.su

Poncy.su – это сервис для построения оптимального маршрута по нескольким точкам на карте, полученный путь будет одним из оптимальных по расстоянию или по времени, также можно определить расстояние между несколькими точками. [6]

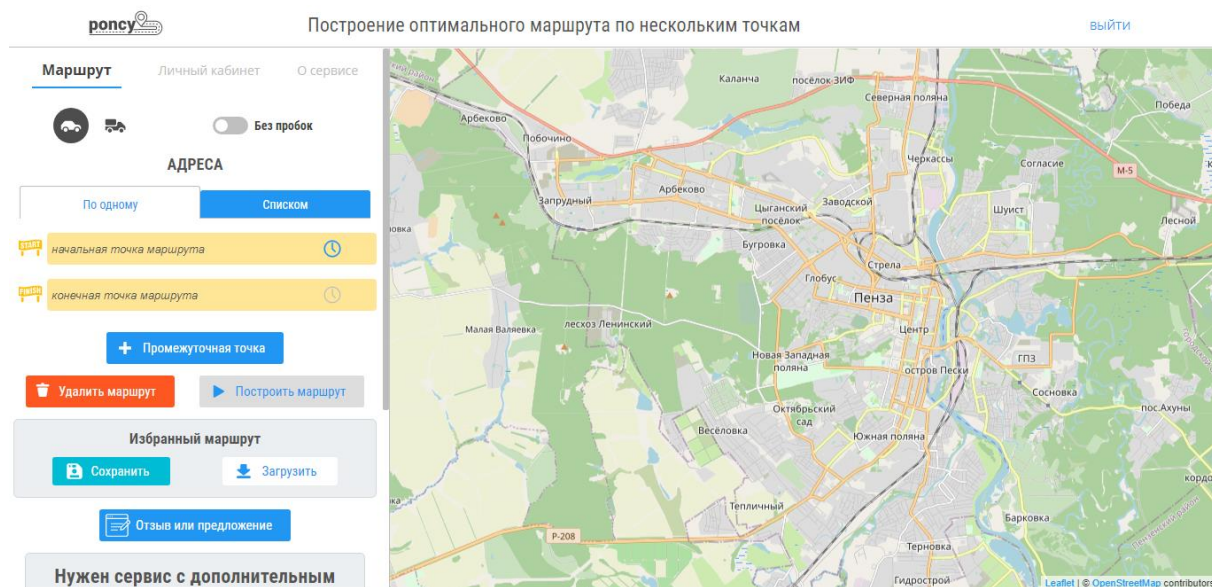


Рисунок 4 – Poncy.su

1.2.5 ISSA.ru

ISSA.ru – это бесплатный сервис расчета маршрутов и расстояний на карте.

[7]

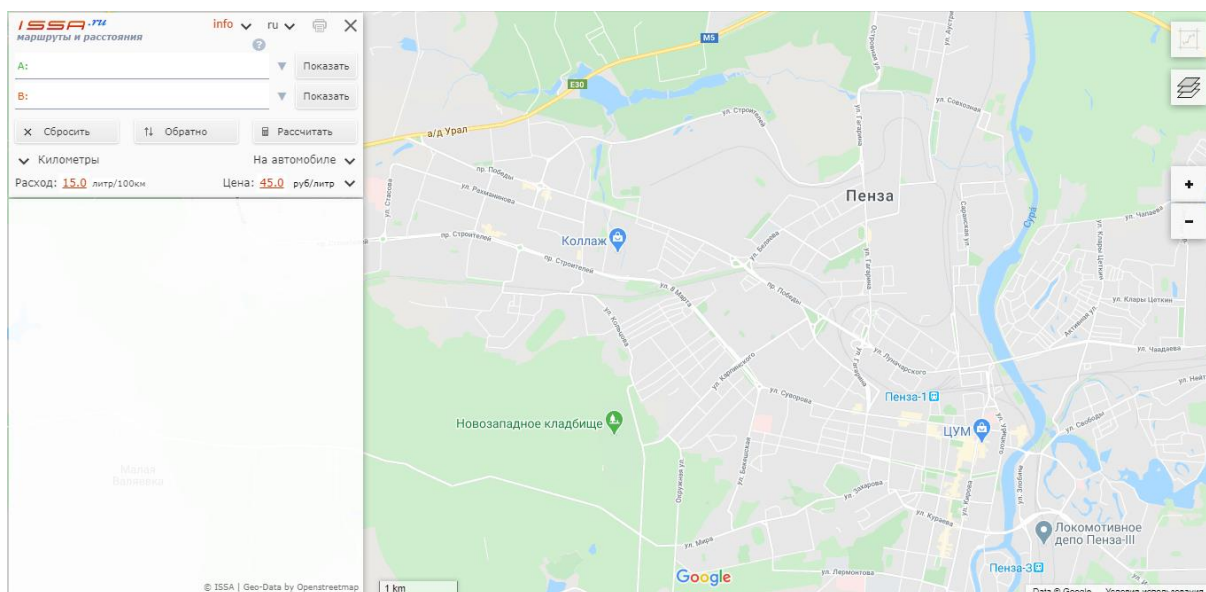


Рисунок 5 – ISSA.ru

После некоторого времени использования представленных сервисов, был выявлен ряд недочетов и недостатков. Таким образом, рассмотренные выше аналоги обладают следующими недостатками:

- Наиболее важный недостаток, которым обладает большинство рассмотренных систем – это отсутствие оптимизации промежуточных точек маршрута.

- Следующим по важности недостатком является строгое указание конечной точки маршрута, так как для пользователя затруднительно среди множества входных адресов выделить один единственный конечный адрес, при котором маршрут будет считаться по-настоящему оптимизированным по тому или иному признаку.

- Еще одним из недостатков при решении задачи логистической оптимизации, на мой взгляд, является отсутствие возможности оптимизации с учетом веса доставляемого груза на тот или иной адрес. То

есть было бы уместно добавить подобный вид оптимизации с точки зрения аварийной безопасности грузоперевозок, так как перевозка грузов близких по общему весу к максимальной грузоподъемности значительно ухудшает управляемость транспортного средства, а это в свою очередь повышает шансы возникновения дорожно-транспортного происшествия.

- Также, далеко не все из рассмотренных сервисов поддерживают функцию отправки сведений о построенном маршруте на электронную почту, что было бы полезно как для водителя, так и для работника логистической службы.

- Кроме этого, не во всех представленных системах реализована такая функция, как сохранение избранных пользовательских адресов, которая может пригодиться в случае многократного повторного использования пользователем одного и того же адреса.

- И наконец, многие рассмотренные веб-сервисы игнорируют такую функцию, как расчет приблизительного расхода и затрат на топливо, необходимое для преодоления построенного маршрута. Данная функция была бы полезна как для водителя ТС, так и для логиста.

Ниже представлена информация (см. Таблица № 1), в которой можно узнать и сравнить какими выявленными характеристиками обладает каждый из аналогов разрабатываемого сервиса.

1.3 Обоснование актуальности разработки

Геоинформационные системы это системы ориентировки во времени и пространстве, они включают в себя современные методы обработки информации. Интерфейс геоинформационных систем зачастую построен таким образом, что ее использование сводится к интуитивно понятным действиям. Современные геоинформационные системы обладают широкими функциональными возможностями.

Таблица № 1 – Сравнение характеристик аналогов

	Google Карты	Яндекс Карты	2ГИС	Poncy.su	ISSA.ru	Разрабатываемый сервис
Оптимизация промежуточных точек	-	+	-	+	-	+
Отсутствие строгого указания конечной точки	-	+	-	-	-	+
Оптимизация с учетом веса груза	-	-	-	-	-	+
Отправка сведений о маршруте на эл. почту	+	+	-	-	-	+
Сохранение избранных пользовательских адресов	+	+	+	-	-	+
Расчет расхода и затрат на топливо	-	-	-	-	+	+

В настоящее время цифровые карты и соответствующие методы их обработки занимают основную нишу в разработке картографических систем, вытесняя уже ставшие устаревшими ручные, аналоговые способы создания картографических материалов.

На сегодняшний день использование печатных топографических карт отошло на задний план, так как им на смену приходят более удобные и постоянно обновляемые веб-сервисы.

В связи с этим разрабатываемый сервис имеет актуальность для пользователей, чья деятельность так или иначе связана с построением и (или) преодолением маршрутов, включающих в себя множество промежуточных точек. Построение маршрута в режиме онлайн, это не только удобная возможность быстрой доставки груза, но и исчерпывающая информация для

всех участников логистической цепочки - водителей, логиста или руководителя.

Подводя итоги можно с уверенностью сказать, что актуальность разработки данной системы полностью оправдана.

1.4 Обоснование выбора используемых при разработке технологий и программных средств

Для разработки системы требуется определенный набор инструментов. С целью обеспечения удобства процесса написания программного кода, в качестве среды разработки была выбрана IntelliJ IDEA. Чтобы получить современный дизайн необходимо воспользоваться связкой языков программирования HTML + CSS + JS. Для отображения маршрутов на геоинформационной карте и получения информации о расстояниях между объектами, необходимо использование программных интерфейсов приложений Google Directions API, Google Distance Matrix API и Google Maps API. Для разработки алгоритмов регистрации и авторизации пользователей и алгоритма сохранения избранных адресов пользователя необходим язык Java. Чтобы сохранять полученные данные необходимо иметь базу данных, в качестве которой выступает PostgreSQL. Также необходим веб-сервер, для размещения разработанной системы и проверки её работоспособности, в качестве такого выступает веб-сервер Apache.

Рассмотрим каждый набор инструментов разработки более подробно.

IntelliJ IDEA — интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains.

Первая версия появилась в январе 2001 года и быстро приобрела популярность как первая среда для Java с широким набором интегрированных инструментов для рефакторинга, которые позволяли программистам быстро реорганизовывать исходные тексты программ. Дизайн среды ориентирован на продуктивность работы программистов,

позволяя сконцентрироваться на функциональных задачах, в то время как IntelliJ IDEA берёт на себя выполнение рутинных операций [8].

HTML (HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык разметки документов в сети Интернет. HTML-документ состоит из дерева HTML-элементов и текста. Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом (за редким исключением). Передача HTML страниц через Интернет осуществляется по протоколам HTTP или HTTPS [9].

CSS (Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка HTML. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом [10].

JavaScript – является объектно-ориентированным языком программирования, поддерживающий императивный и функциональный стили. Язык предназначен для написания сценариев для активных HTML-страниц [11].

PostgreSQL – свободная объектно-реляционная система управления базами данных.

Сильными сторонами PostgreSQL считаются:

- высокопроизводительные и надёжные механизмы транзакций и репликации;
- расширяемая система встроенных языков программирования: в стандартной поставке поддерживаются PL/pgSQL, PL/Perl, PL/Python и PL/Tcl;

дополнительно

можно

использовать PL/Java, PL/PHP, PL/Py, PL/R, PL/Ruby, PL/Scheme, PL/sh и PL/V8, а также имеется поддержка загрузки C-совместимых модулей;

- наследование;
- легкая расширяемость [12].

Google Directions API – это служба, которая вычисляет маршруты между пунктами с помощью HTTP-запроса. В службе Directions пункты отправления и назначения могут указываться в виде текстовых строк (например, «Чикаго, Иллинойс» или «Дарвин, Новый Южный Уэльс, Австралия») либо как координаты широты и долготы [13].

Google Distance Matrix API — это служба, предоставляющая информацию о расстоянии и времени пути для матрицы исходных и конечных точек. Информация предоставляется на основе данных о рекомендуемом маршруте между начальными и конечными точками, рассчитанном с помощью Google Maps API, и представляет собой строки, содержащие значения duration и distance для каждой пары точек [14].

Google Maps API — это коллекция инструментов для добавления карт Google в мобильные и веб-приложения а также создания на их основе multifunctional решений [15].

Java - сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины [16].

Apache – это полнофункциональный, расширяемый веб-сервер, полностью поддерживающий протокол HTTP/1.1 и распространяющийся с открытым исходным кодом.

Выделим основные преимущества веб-сервера Apache:

- Модульность (Возможность подключать различные модули);
- Простота настройки;
- Безопасность;
- Кроссплатформенность.

В результате рассмотрения выше описанных средств разработки можно сказать, что они обладают всеми функциями для реализации поставленной задачи.

1.5 Вывод

В результате анализа предметной области была поставлена задача для создания геоинформационного веб-сервиса для оптимизации доставки грузов. Были рассмотрены аналоги данной системы, в результате анализа выявлены достоинства и недостатки обзореваемых систем, которые будут устранены в данной работе. Так же обоснована актуальность и необходимость разработки. Выбраны инструменты разработки, которые наиболее полно подходят для данной предметной области.

2 Проектирование структуры системы

2.1 Технология взаимодействия веб – ресурсов

В основе работы веб - приложений лежит так называемая модель взаимодействия клиент-сервер, которая позволяет разделять функционал и вычислительную нагрузку между клиентскими приложениями (заказчиками услуг) и серверными приложениями (поставщиками услуг).

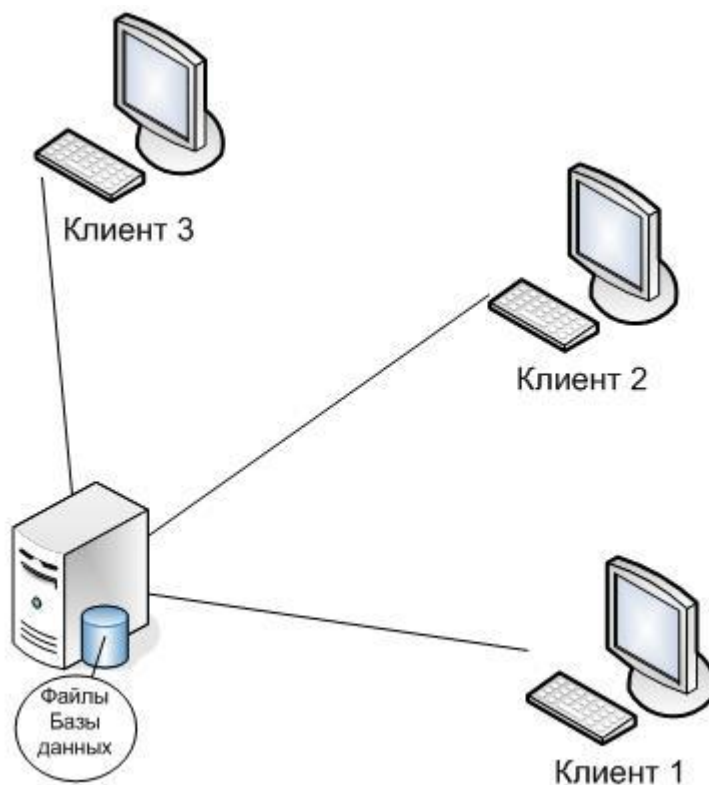


Рисунок 6 - Архитектура взаимодействия "клиент - сервер"

Не трудно догадаться из названия, что в данной архитектуре участвуют две стороны: клиент и сервер. Клиентами могут выступать обычные пользователи, имеющие доступ к сети Интернет, и соответствующее программное обеспечение, таковыми являются браузеры. Серверы - компьютеры, которые хранят веб-страницы, сайты или приложения. В качестве примера сервера можно привести веб-сервер Apache. Клиент и сервер взаимодействуют друг с другом в сети Интернет или в любой другой компьютерной сети при помощи различных сетевых протоколов, например,

IP протокол, HTTP протокол, FTP. Сообщения, которые посылают клиенты получили названия HTTP запросы. Запросы имеют специальные методы, которые говорят серверу о том, как обрабатывать сообщение. А сообщения, которые посылает сервер получили название HTTP ответы, они содержат помимо полезной информации еще и специальные коды состояния, которые позволяют браузеру узнать то, как сервер понял его запрос. Не маловажной особенностью, в данной технологии является, принцип по которому клиент всегда начинает взаимодействие, а сервер только отсылает ответы на запросы клиента на предоставлении услуг. При этом, правила по которым происходит взаимодействие описываются протоколом.

Существует два вида архитектуры клиент-сервер:

- двухзвенная архитектура клиент-серверного взаимодействия,
- многоуровневая архитектура клиент-сервер.

Принцип работы двухуровневой архитектуры взаимодействия клиент-сервер заключается в том, что обработка запроса происходит на одной машине без использования сторонних ресурсов. Двухзвенная архитектура предъявляет жесткие требования к производительности сервера, но в тоже время является очень надежной. Двухуровневая модель взаимодействия клиент-сервер представлена на рисунке 7.

Согласно рисунку, есть первое звено, клиент, который производит запрос, второе звено, сервер, обрабатывает запрос и отсылает ответ.

Подход многоуровневой архитектуры заключается в том, что запрос клиента обрабатывается сразу несколькими серверами. Такой подход позволяет значительно снизить нагрузку на сервер из-за того, что происходит распределение операций, но в то же самое время данный подход не такой надежный, как двухзвенная архитектура. Многоуровневая модель взаимодействия клиент-сервер представлена на рисунке 8.

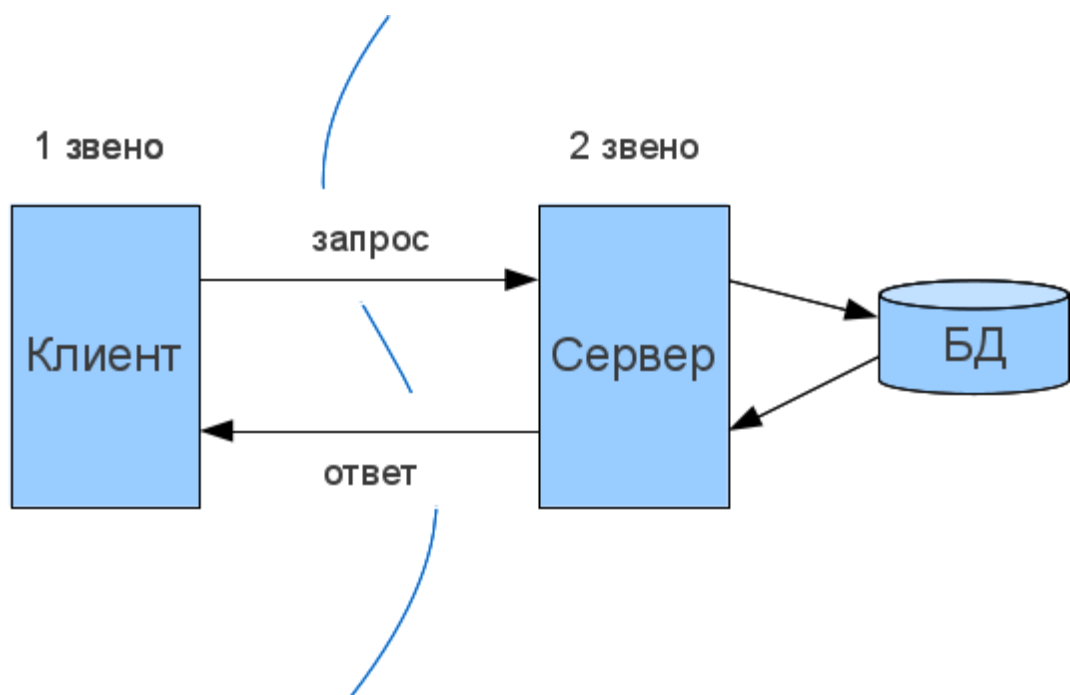


Рисунок 7 - Двухуровневая модель взаимодействия клиент-сервер

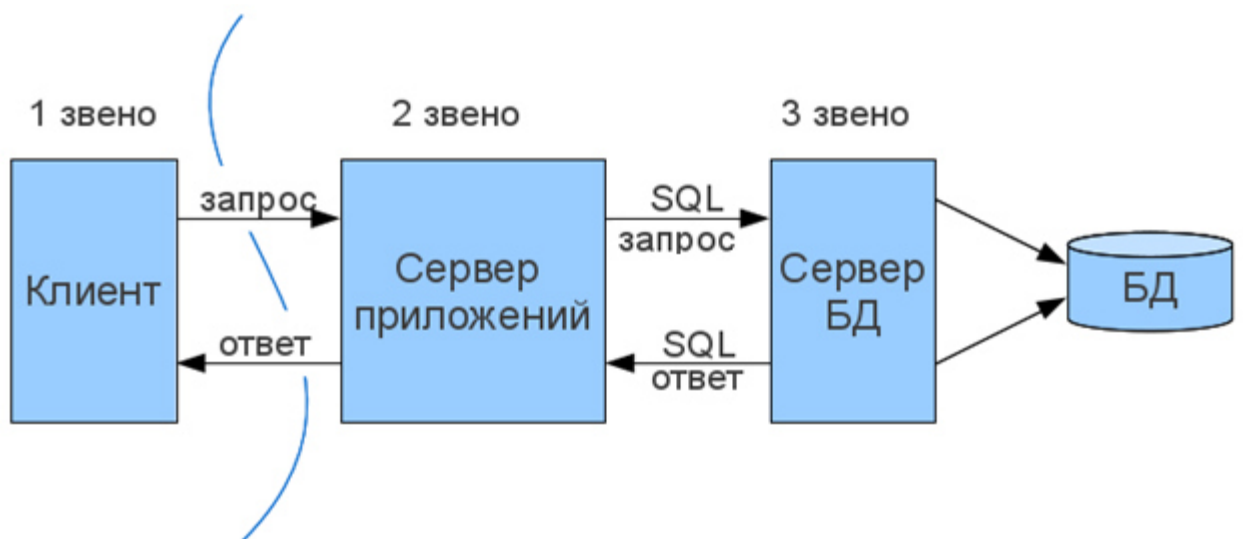


Рисунок 8 - Многоуровневая модель взаимодействия клиент-сервер

Интерпретировать рисунок выше можно с позиции сайта. На первом звене пользователь, используя браузер, выполняет запрос на посещение сайта, на втором звене веб – сервер обрабатывает запрос с помощью PHP, и отправляет 3 звену серверу с базой данных. Затем сервер базы данных отправляет ответ, веб серверу, а уже веб сервер отправляет ответ клиенту.

Преимуществом модели взаимодействия клиент-сервер является то, что программный код клиентского приложения и серверного разделен.

В результате рассмотрения клиент-серверной архитектуры проектирование автоматизированной системы будет осуществляться на серверной и клиентской части.

2.2 Проектирование информационного обеспечения для серверной части

На стороне сервера будет организована работа алгоритмов регистрации и авторизации пользователей, а также работа базы данных, в которой будет храниться информация о пользователях.

Алгоритм регистрации пользователя будет реализован на языке java. Блок схема алгоритма представлена на рисунке 9.

Алгоритм авторизации будет реализован на языке java. Блок схема алгоритма представлена на рисунке 10.

2.3 Описание пользовательского интерфейса приложения

Проектирование для клиентской части включает в себя: проектирование главной страницы сайта, обладающей современным дизайном, отсутствием назойливой рекламы, перегруженности главной страницы и удобной навигацией по сайту. Так же проектирование отдельных страниц для авторизации, регистрации и личного кабинета пользователя.

Вид спроектированной главной страницы сервиса представлен на рисунке 11.

Главная страница состоит из верхней панели, на которой расположены кнопки открытия боковой панели, регистрации и авторизации, а также название разработанной системы. Боковая панель содержит интерфейс, связанный с основным функционалом, а именно построением маршрутов. Кнопка «регистрация» предназначена для отображения модального окна регистрации нового пользователя. Кнопка «войти» предназначена для отображения специального модального окна авторизации.



Рисунок 9 - Блок схема алгоритма регистрации пользователя



Рисунок 10 - Блок схема алгоритма авторизации пользователя

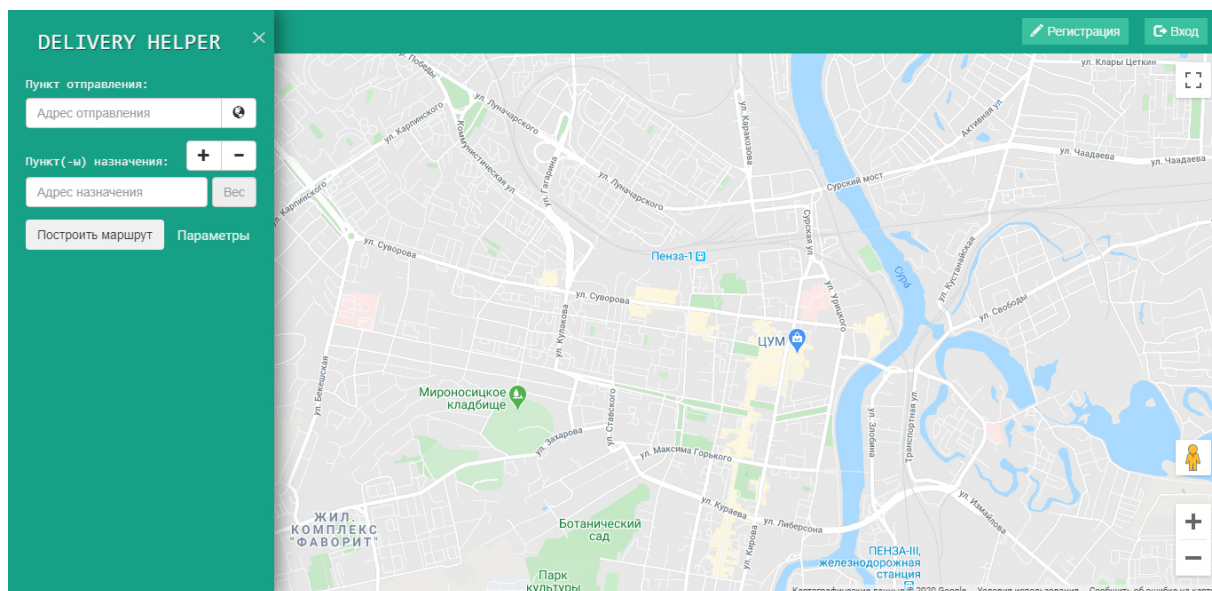


Рисунок 11 - Дизайн главной страницы

Большую часть экрана занимает цифровая географическая карта. На ней будет отображаться построенный маршрут между указанными пользователем точками.

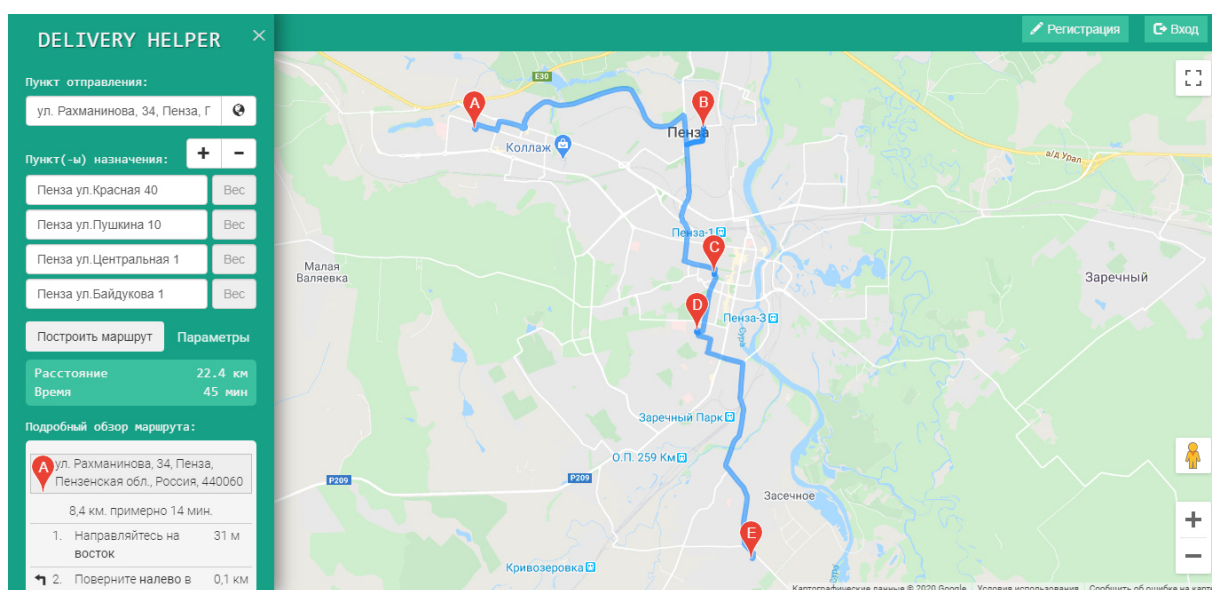


Рисунок 12 – Отображение построенного маршрута

В результате работы системы строится итоговый маршрут с использованием пользовательских входных адресов в оптимизированном порядке. Кроме этого, выводятся на экран основные сведения о маршруте такие как: расстояние и приблизительное время преодоления маршрута. При

Пенза ул. Байдукова 1

Вес

Построить маршрут

Параметры

Способ передвижения:

Пешком

☒

Автомобиль

Грузоподъемность:

Кг.

50

3500

Оптимизация маршрута:

С учетом расстояния

Цена топлива(₽/литр)

45.4

Расход топлива(л/100км)

11.2

Расстояние

22.4 км

Время

45 мин

Расход топлива

Объем

2.51 л

Стоимость

113.95 ₽

Подробный обзор маршрута:

А

ул. Рахманинова, 34, Пенза,
Пензенская обл., Россия, 440060

8.4 км. примерно 14 мин.

Регистрация

Вход

Пенза

Заречный

Засечное

Кривоzerовка

О.П. 259 км

Заречный Парк

Пенза-3

Пенза-1

Коллаж

Малая Вальевка

А

В

С

Д

Е

а/д Урал

Р-209

Р-209

8.4 км. примерно 14 мин.

Рисунок 13 – Входные параметры

Delivery helper

Регистрация

Вход

Карта

Спутник

Регистрация

Имя пользователя

Введите имя пользователя

Электронная почта

Введите адрес эл. почты

Пароль пользователя

Введите пароль

Подтвердите пароль

✓

Зарегистрироваться

Рисунок 14 – Модальное окно регистрации пользователя

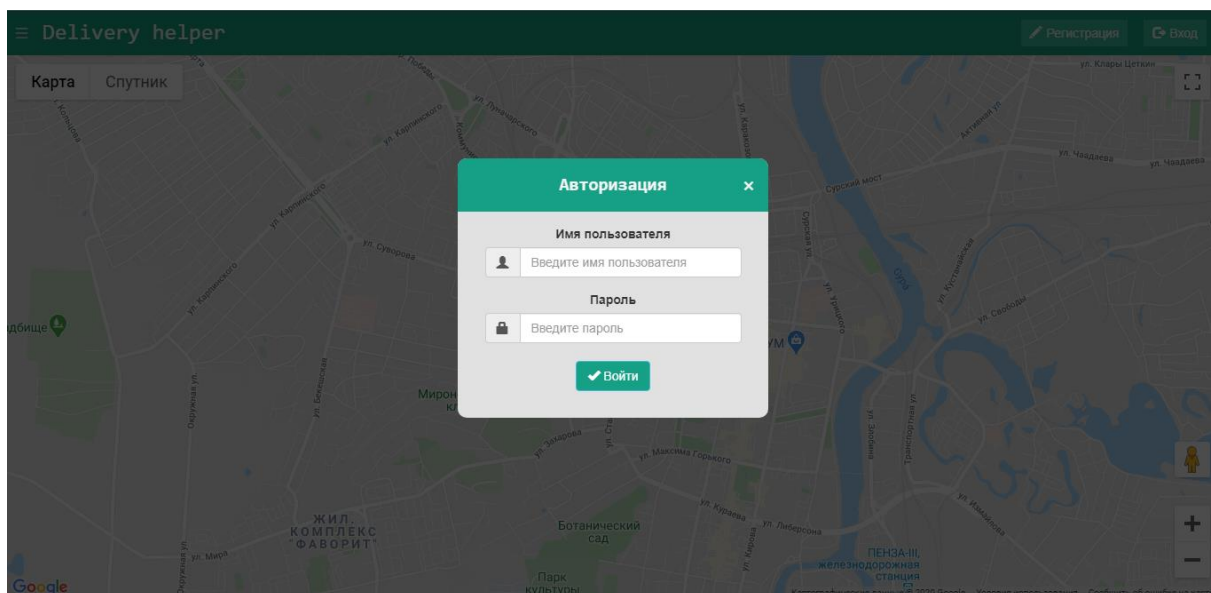


Рисунок 15 – Модальное окно авторизации пользователя

Кроме того, после прохождения авторизации, пользователю откроется возможность сохранения и использования избранных адресов, а также отправки сведений о построенном маршруте на электронную почту. Для авторизованного пользователя в системе предусмотрено специальное модальное окно, представленное на рисунке 16, с персональными настройками, где можно сохранить, изменить либо удалить один из избранных адресов, а также изменить адрес электронной почты пользователя, указанный при регистрации.

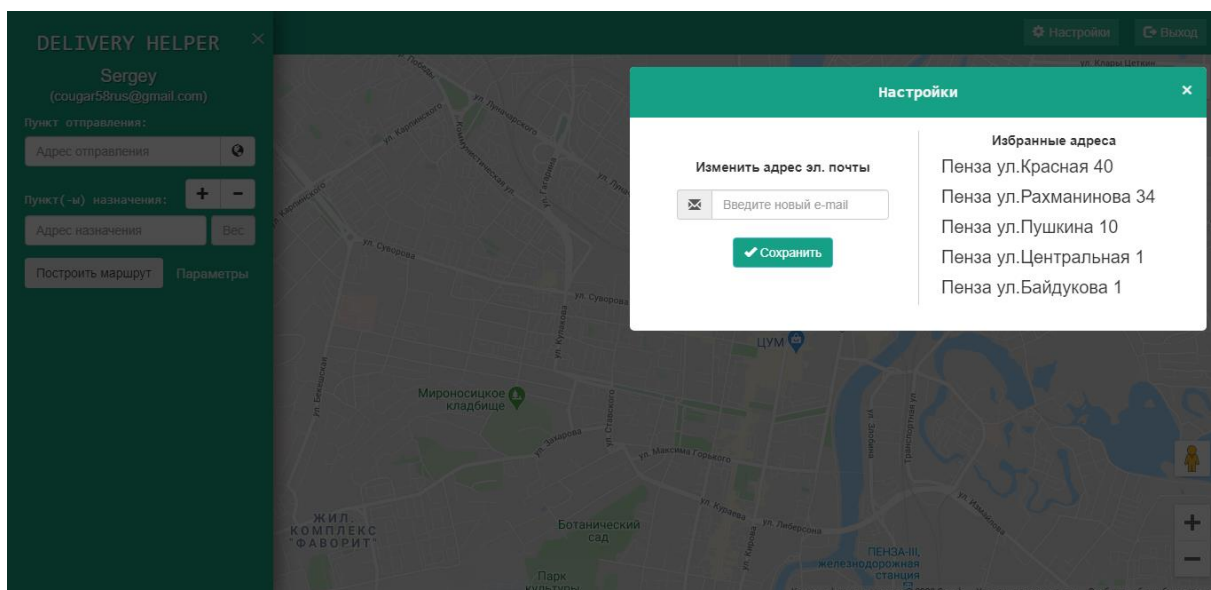


Рисунок 16 – Модальное окно персональных настроек пользователя

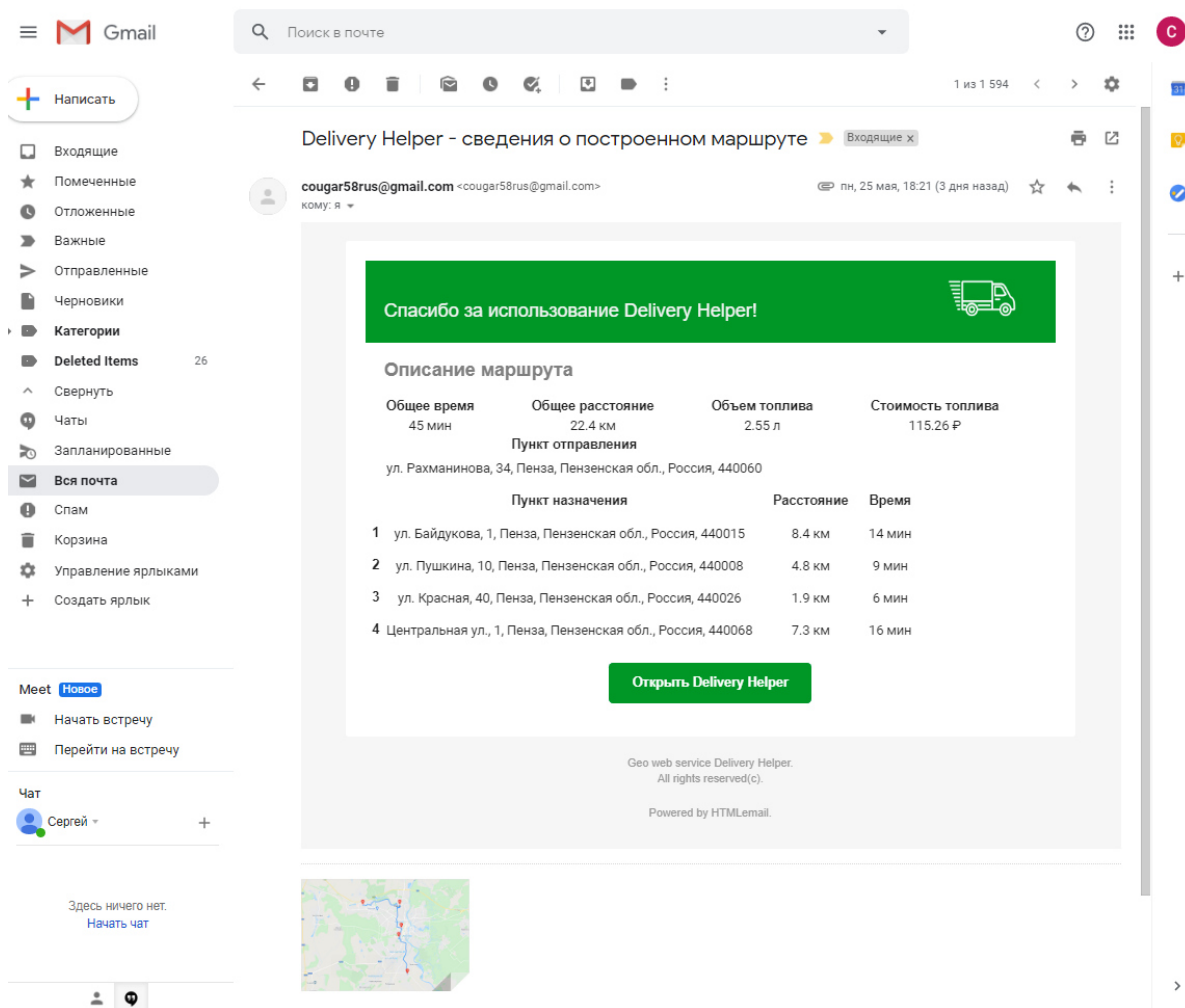


Рисунок 17 – Электронное письмо со сведениями о построенном маршруте

2.4 Проектирование базы данных разрабатываемой системы

Для создания базы данных использовалось приложение pgAdmin3. Спроектированная база данных для разрабатываемой системы представлена на рисунке 18.

Редактирование данных - PostgreSQL 9.3 (localhost:5432) - test - users								
Файл Правка Вид Инструменты ?								
Не ограниченно ▾								
	login text	password text	home text	workaddr text	fav_loc1 text	fav_loc2 text	fav_loc3 text	email text
1	TestUser	5f4d0cc3b5aa765d61d8327deb882cf99	Адрес не задан	Адрес не задан	Адрес не задан	Адрес не задан	Адрес не задан	tvinteam@yandex.ru
2	Sergey	5f4d0cc3b5aa765d61d8327deb882cf99	Пенза ул.Красная 40	Пенза ул.Рахманинова 34	Пенза ул.Пушкина 10	Пенза ул.Центральная 1	Пенза ул.Байдукова 1	cougar58rus@gmail.com

Рисунок 18 – Хранение информации в базе данных

Для разработки базы данных достаточно лишь одной таблицы:

В таблице «users» хранятся данные о зарегистрированных пользователях.

Таблица содержит следующие поля:

- login – имя пользователя;
- password – пароль для аутентификации пользователя в сервисе;
- home – избранный адрес пользователя;
- work – избранный адрес пользователя;
- fav_loc1 – избранный адрес пользователя;
- fav_loc2 – избранный адрес пользователя;
- fav_loc3 – избранный адрес пользователя;
- email – адрес электронной почты пользователя.

Для безопасности все пароли хранятся в базе данных с использованием алгоритма шифрования данных md5.

2.5 Вывод

В результате проектирования были рассмотрены особенности организации архитектуры клиент – сервер сети Интернет. Спроектированы следующие алгоритмы на серверной части: регистрация и авторизация пользователя. На клиентской части были спроектированы: дизайн главной страницы системы и модальных окон регистрации, авторизации и пользовательских настроек. Спроектирована модель базы данных. В результате проектирования были получены все необходимые сведения для разработки системы оптимизации доставки грузов.

3 Разработка ПО

3.1 Алгоритмы оптимизации маршрута

3.1.1 Алгоритм оптимизации маршрута с учетом расстояния

Смысл алгоритма оптимизации маршрута с учетом расстояния заключается в решении задачи нахождения наикратчайшего пути, включающего в себя все точки маршрута, указанные пользователем.

Иллюстрация работы алгоритма:

А, В, С, D – пункты назначения маршрута.

На 1-ом этапе работы алгоритма находится расстояние между начальной точкой и каждым пунктом назначения, маршрут строится до ближайшей точки.

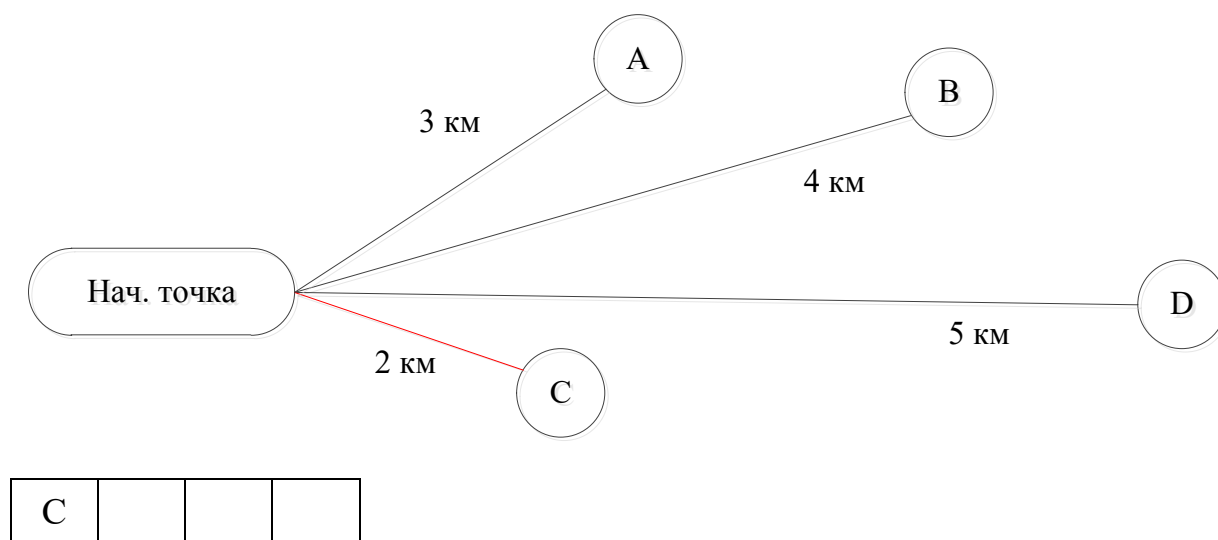


Рисунок 19 – Иллюстрация работы алгоритма

На последующих этапах алгоритма цикл повторяется с одним лишь изменением: на каждом новом этапе расстояние отсчитывается от точки, найденной на предыдущем этапе и только до тех точек, маршрут к которым еще не был построен.

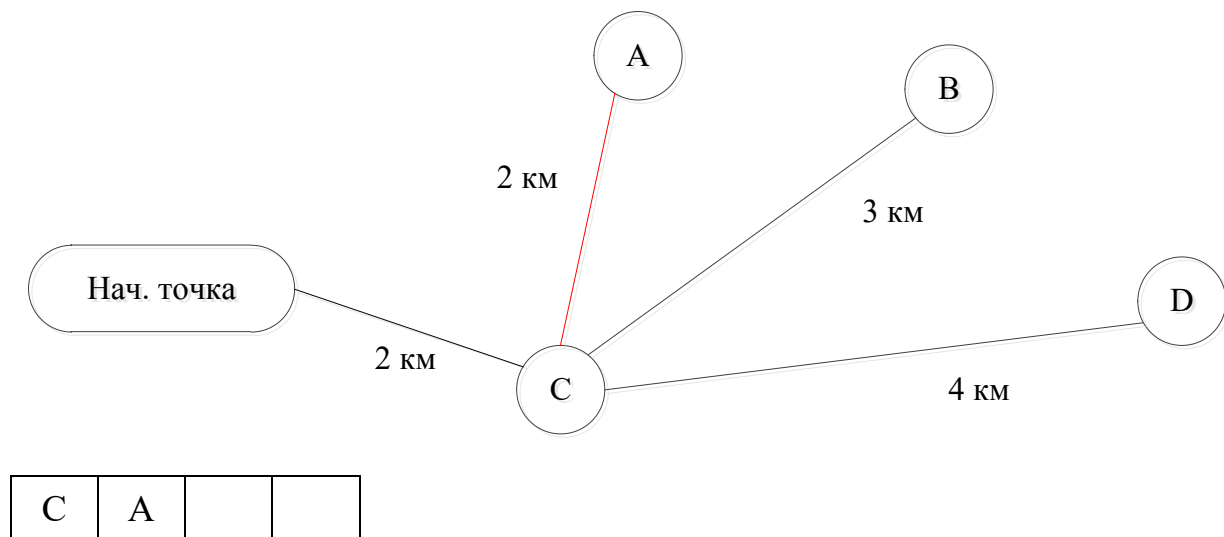


Рисунок 20 – Иллюстрация работы алгоритма

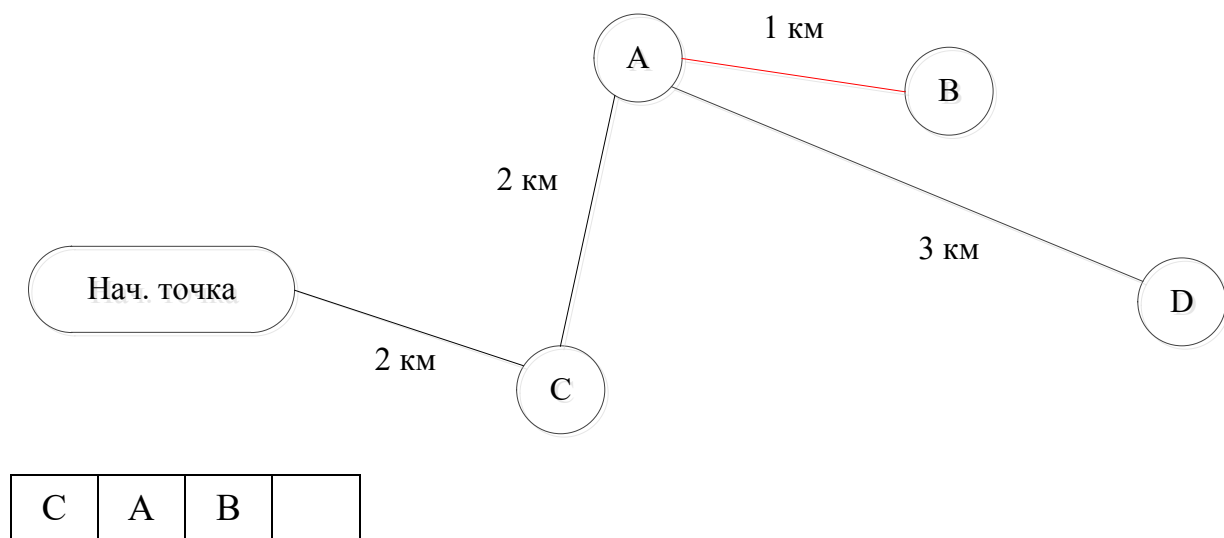


Рисунок 21 – Иллюстрация работы алгоритма

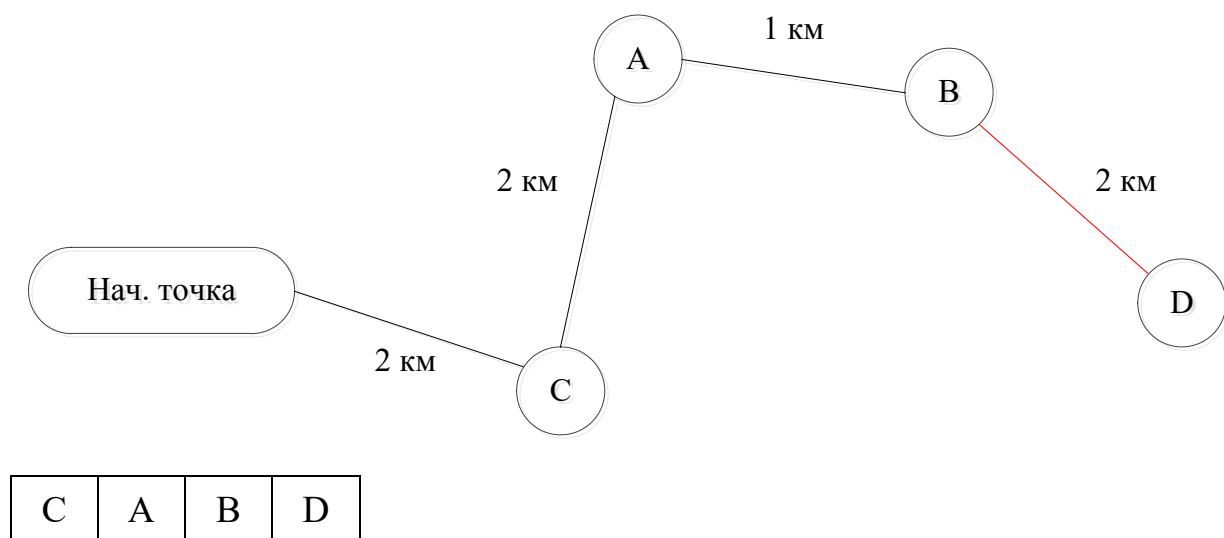


Рисунок 22 – Иллюстрация работы алгоритма

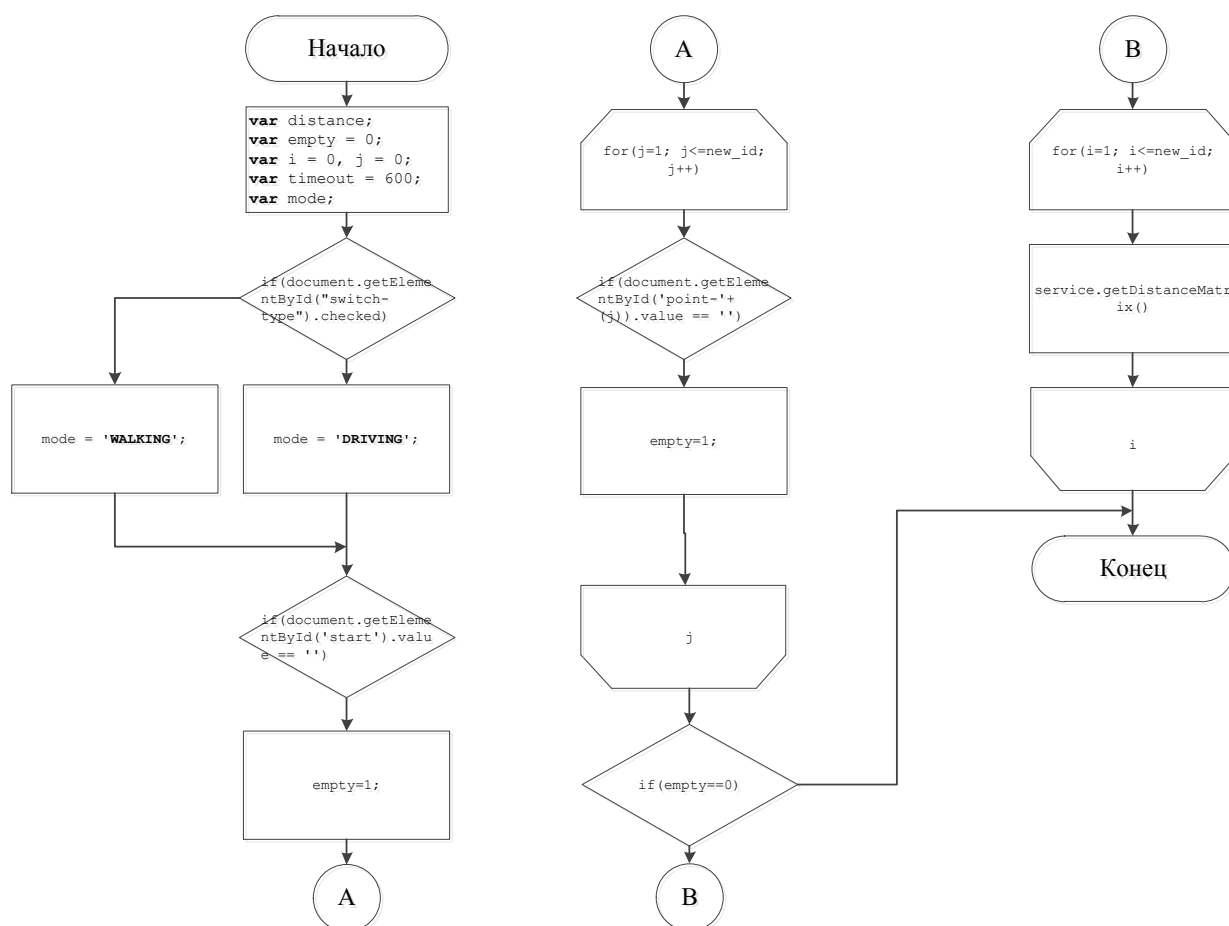


Рисунок 23 – Схема алгоритма оптимизации с учетом расстояния

3.1.2 Алгоритм оптимизации маршрута с учетом веса грузов

Разработка данного алгоритма связана с повышением безопасности доставки грузов. Использование указанной оптимизации рекомендуется, когда общий вес грузов близок или равен максимальной грузоподъемности транспортного средства, используемого для передвижения по итоговому маршруту. Смысл данного типа оптимизации построения маршрута заключается в прохождении промежуточных точек в порядке убывания веса груза, соответствующего одному из пунктов назначения.

Схема алгоритма оптимизации с учетом веса грузов представлена на рисунке 24.

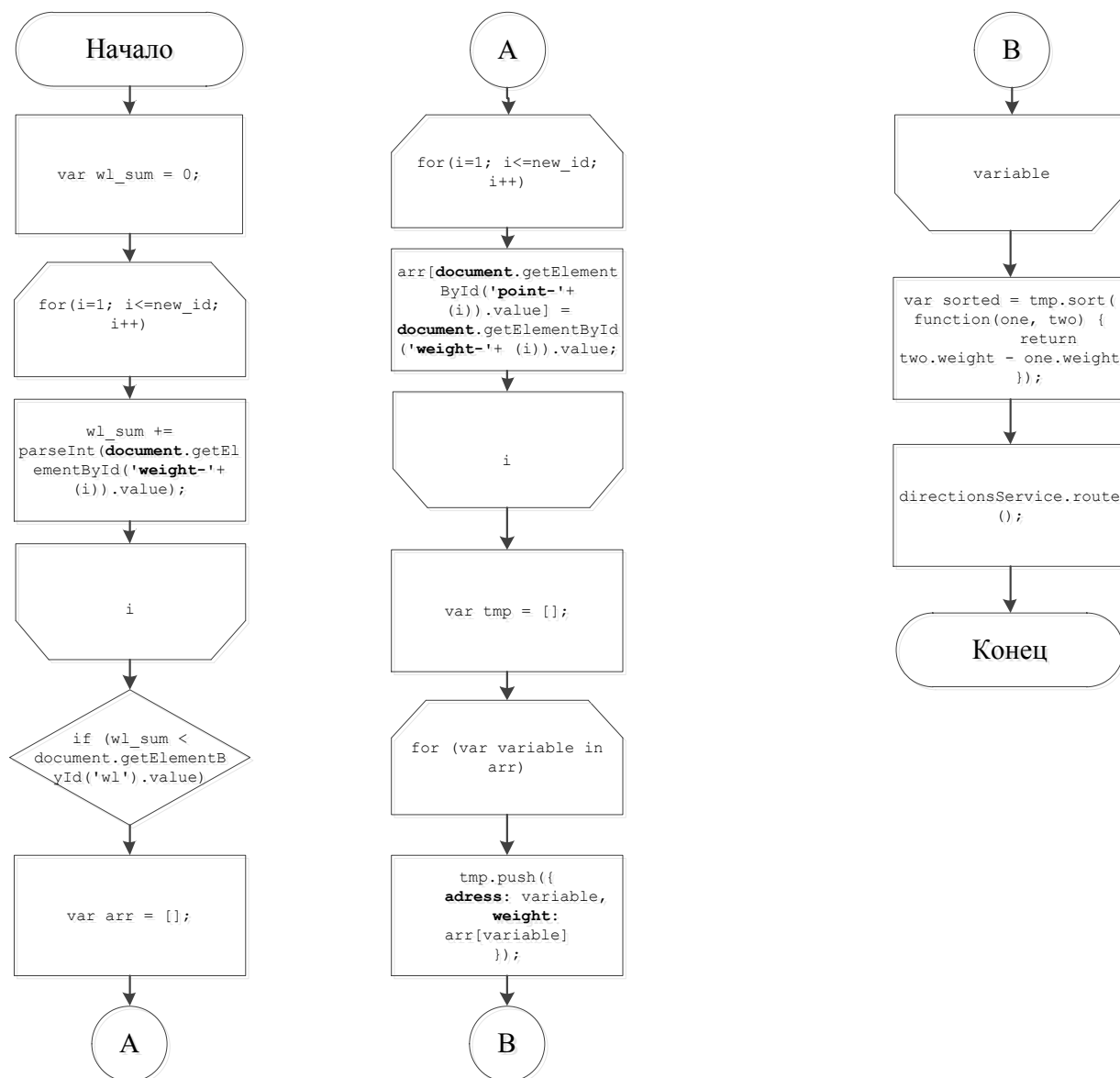


Рисунок 24 – Схема алгоритма оптимизации с учетом веса грузов

3.2 Разработка ПО системы

3.2.1 Программное обеспечение для серверной части

Код программы для серверной части состоит из нескольких файлов, написанных на языке java:

MyServlet.java – файл, в котором происходит получение сведений о пользователях сервиса и занесение этих сведений в базу данных. Данный программный код включает в себя две основные функции.

Registration() – функция выполняет регистрацию пользователей в сервисе, то есть сбор сведений о пользователях и занесение их в базу данных.

Authorization() – функция проверяет введенные пользователем данные при авторизации и сравнивает их с хранящимися в базе данных.

EmailServlet.java – файл, предназначенный для сохранения пользовательского адреса электронной почты в базе данных.

NewHAddress.java – файл, для сохранения пользовательского адреса в БД.

NewWAddress.java – файл, для сохранения пользовательского адреса в БД.

NewFavLocOne.java – файл, для сохранения пользовательского адреса в БД.

NewFavLocTwo.java – файл, для сохранения пользовательского адреса в БД.

NewFavLocThree.java – файл, для сохранения пользовательского адреса в БД.

3.2.2 Программное обеспечение для клиентской части

Код программы для клиентской части состоит из двух программных файлов index.html, index2.html написанных с использованием связки языков HTML + CSS + JS.

index.html – файл, описывающий главную страницу системы, в которой доступен основной функционал системы, однако отсутствуют возможности, доступные авторизованным пользователям.

index2.html – файл, описывающий главную страницу системы для авторизованного пользователя, в которой доступен основной функционал системы, а также возможности связанные с сохранением и взаимодействием с избранными адресами и отправкой сведений о построенном маршруте на электронную почту.

3.3 Вывод

В результате разработки было получено ПО для серверной и клиентской части, выполняющее обработку данных. Код ПО представлен в приложениях А, Б.

4 Тестирование ПО

4.1 Руководство по использованию автоматизированной системы

Для тестирования необходимо чтобы на компьютере был установлен веб сервер Apache, база данных PostgreSQL.

Запустив один из современных браузеров в строке поиска введите `http://localhost:8080/index.html`, для перехода на главную страницу. Вид главной страницы представлен на рисунке 25.

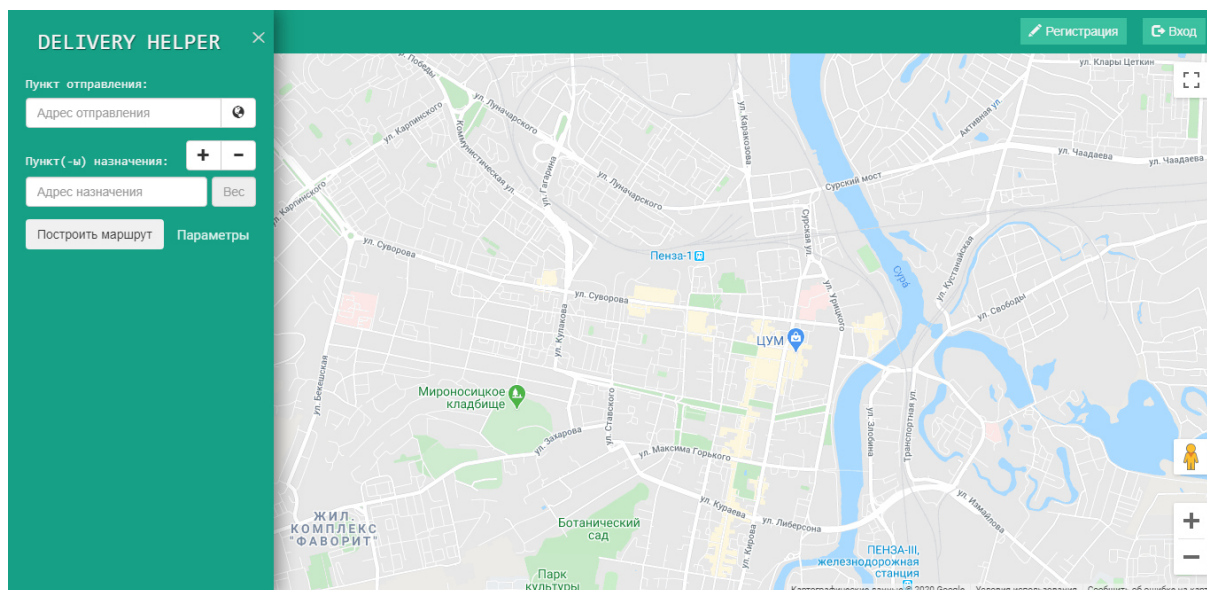


Рисунок 25 – Главная страница

При вводе адреса, для удобства пользователя, доступна возможность автоматического определения местоположения, которая активируется по нажатию специальной кнопки, расположенной рядом с соответствующим полем ввода. Результат работы представлен на рисунке 26.

Далее необходимо указать адреса пунктов назначения, по которым будет проложен итоговый маршрут. Порядок указанных адресов не влияет на результаты оптимизации маршрута. Результат ввода адресов, соответствующих пунктам назначения, представлен на рисунке 27.

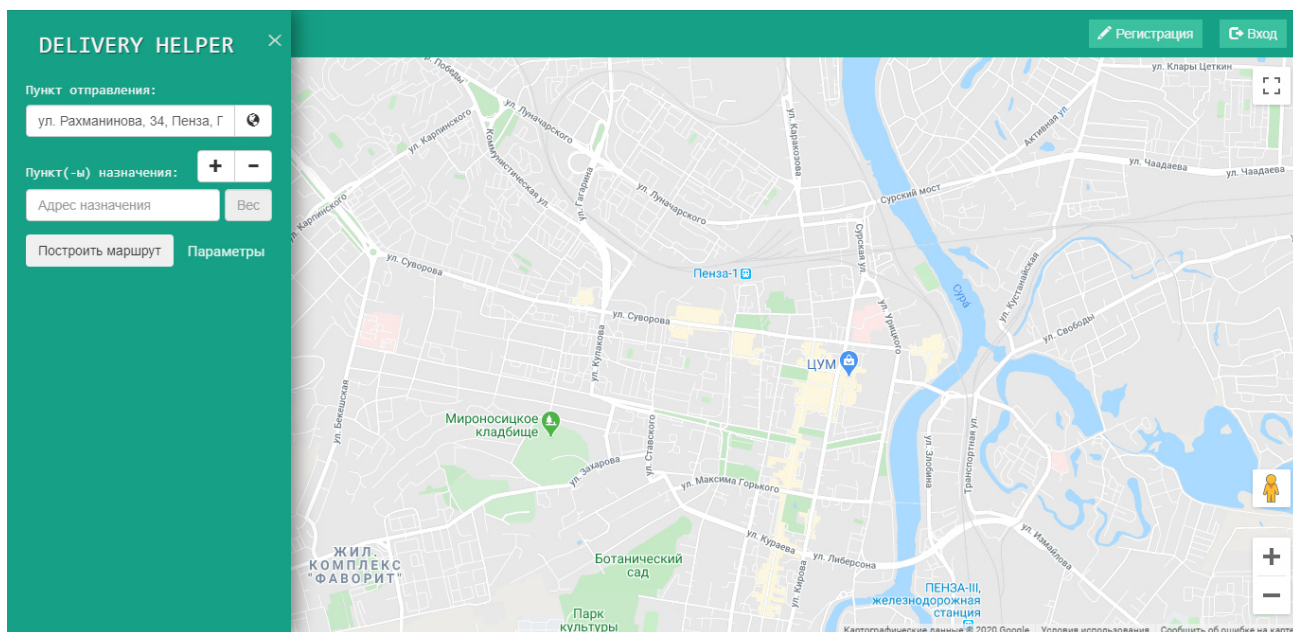


Рисунок 26 – Результат функции определения местоположения

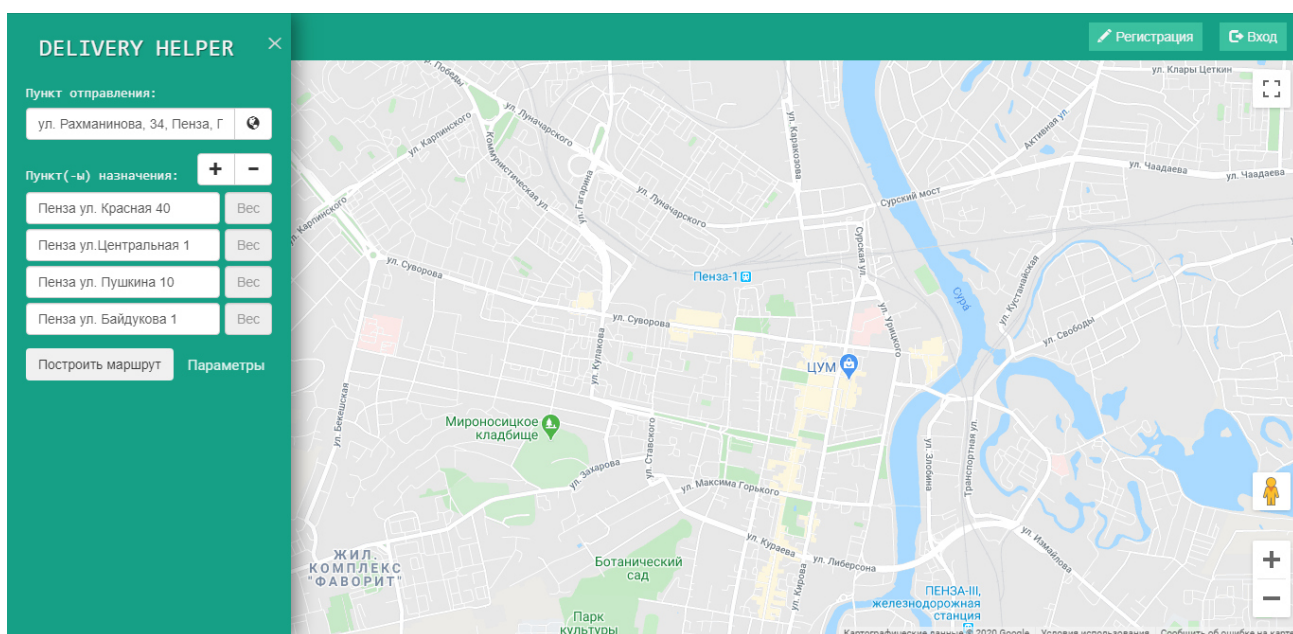


Рисунок 27 – Указание пунктов назначения

Для тестирования обработки входных параметров необходимо воспользоваться соответствующим подменю, открывающимся по клику на пункт «параметры». Указание параметров представлено на рисунке 28.

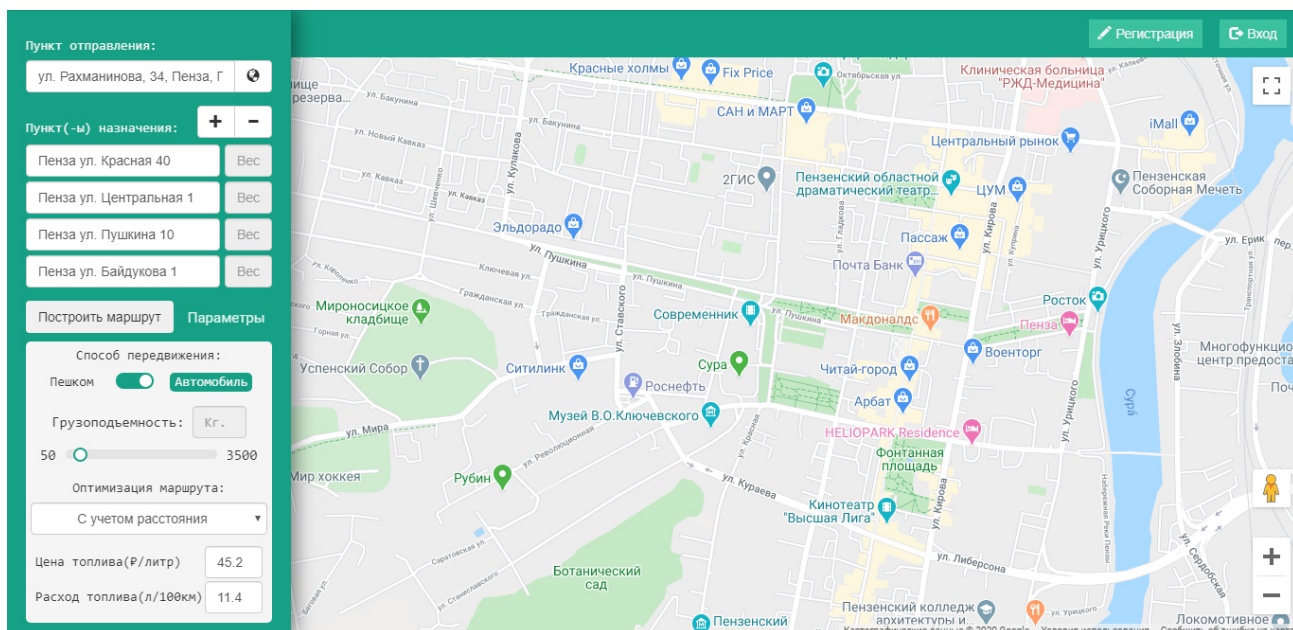


Рисунок 28 – Указание параметров

Затем, для тестирования построения маршрута, с учетом всех входных данных, необходимо нажать на кнопку «Построить маршрут». Результат построения оптимизированного маршрута представлен на рисунке 29.

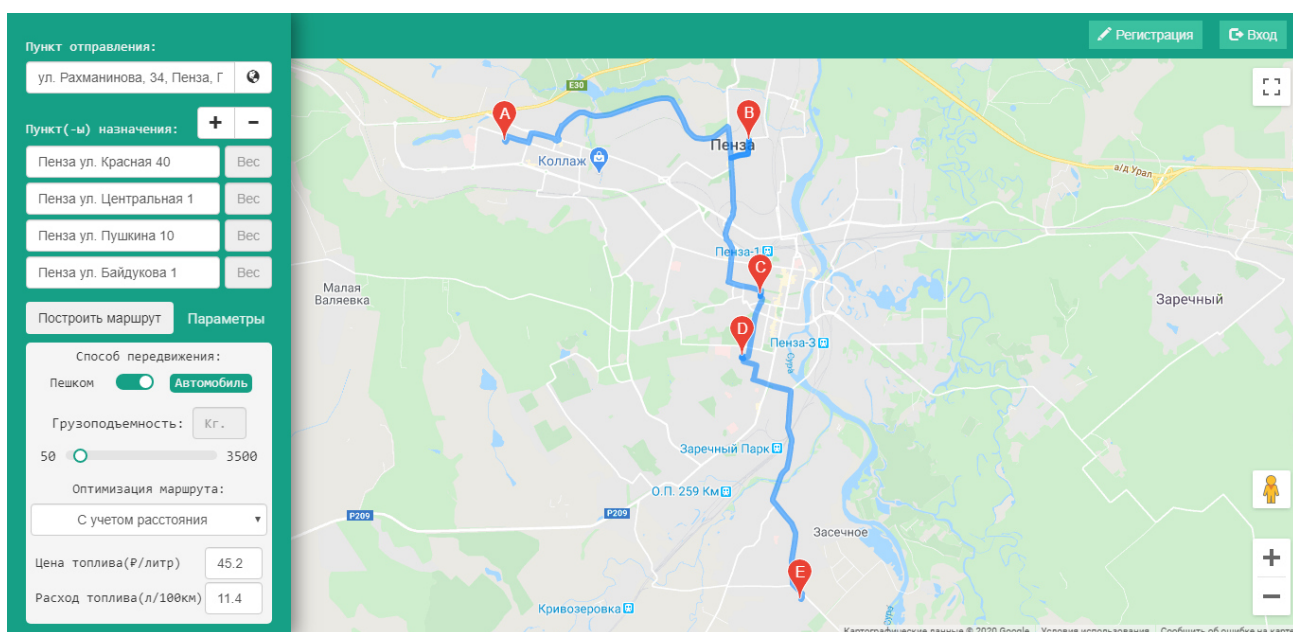


Рисунок 29 – Результат построения маршрута

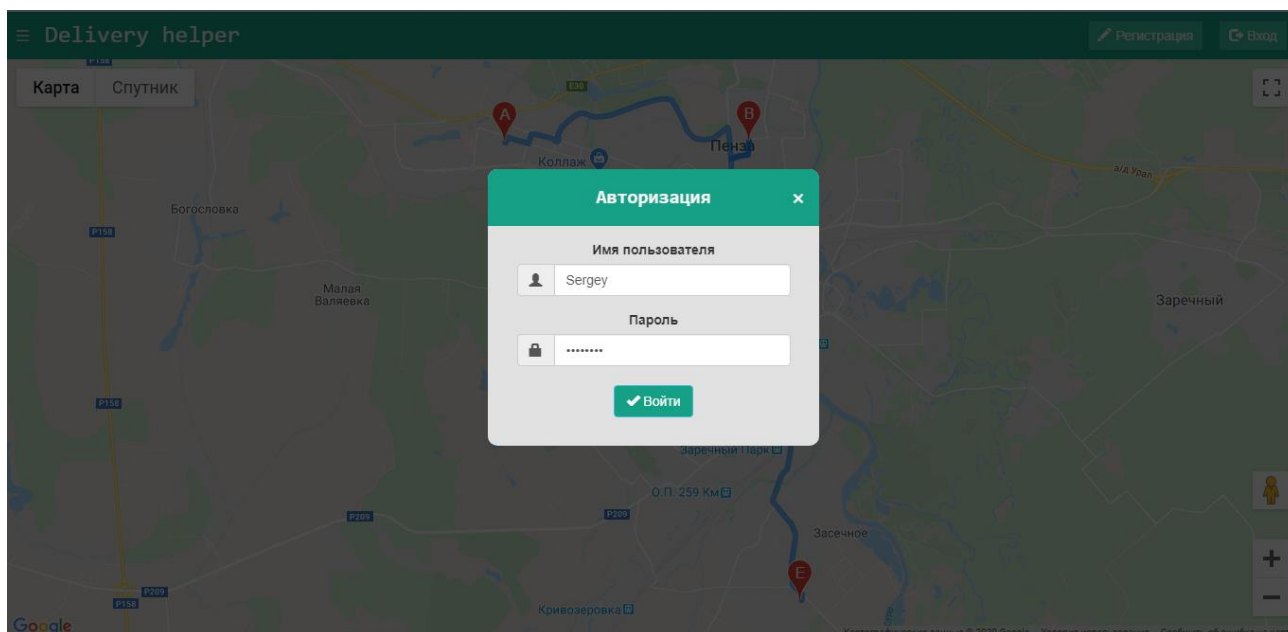


Рисунок 30 – Окно авторизации пользователя

Чтобы провести тестирование дополнительного функционала, необходимо пройти процедуру входа в систему.

На рисунке 30 показано специальное модальное окно авторизации, в котором необходимо указать имя и пароль пользователя.

Для отправления сведений о маршруте на эл. почту необходимо открыть соответствующее подменю «отправить по эл. почте», а также ввести адрес, на который будет отправлено электронное письмо (по умолчанию введен адрес эл. почты, указанный при регистрации пользователя). А затем нажать кнопку «отправить». Ввод адреса эл. почты представлен на рисунке 31.

Содержимое отправленного электронного письма представлено на рисунке 32.

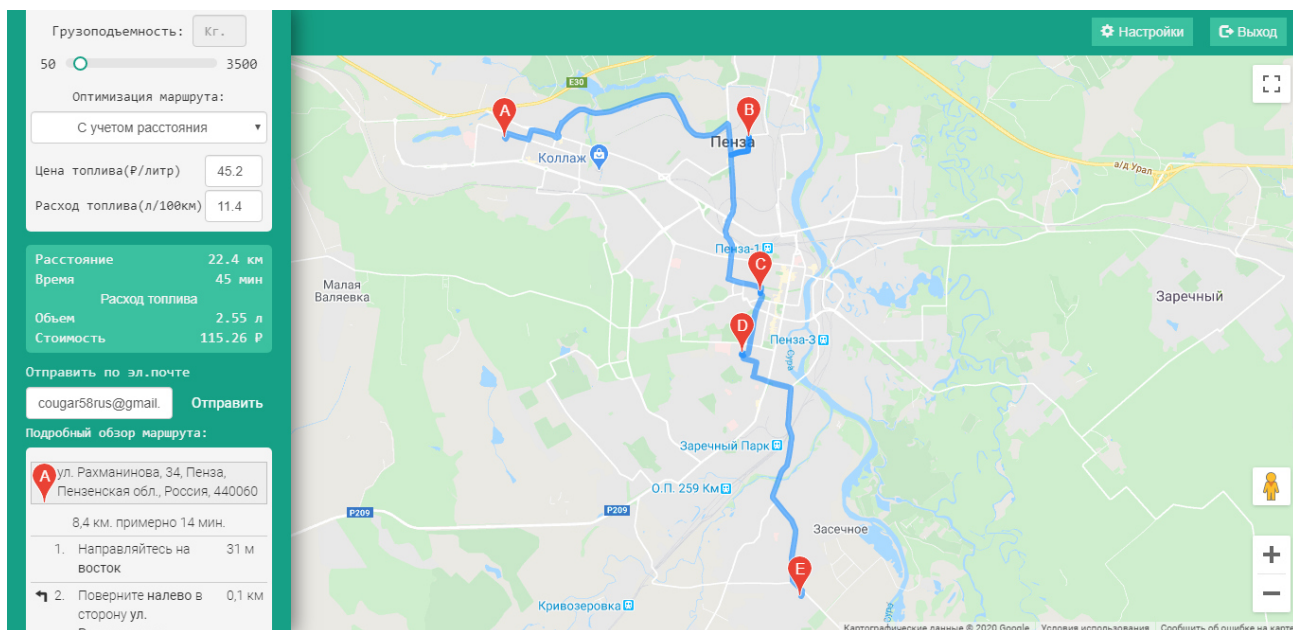


Рисунок 31 – Ввод адреса эл. почты

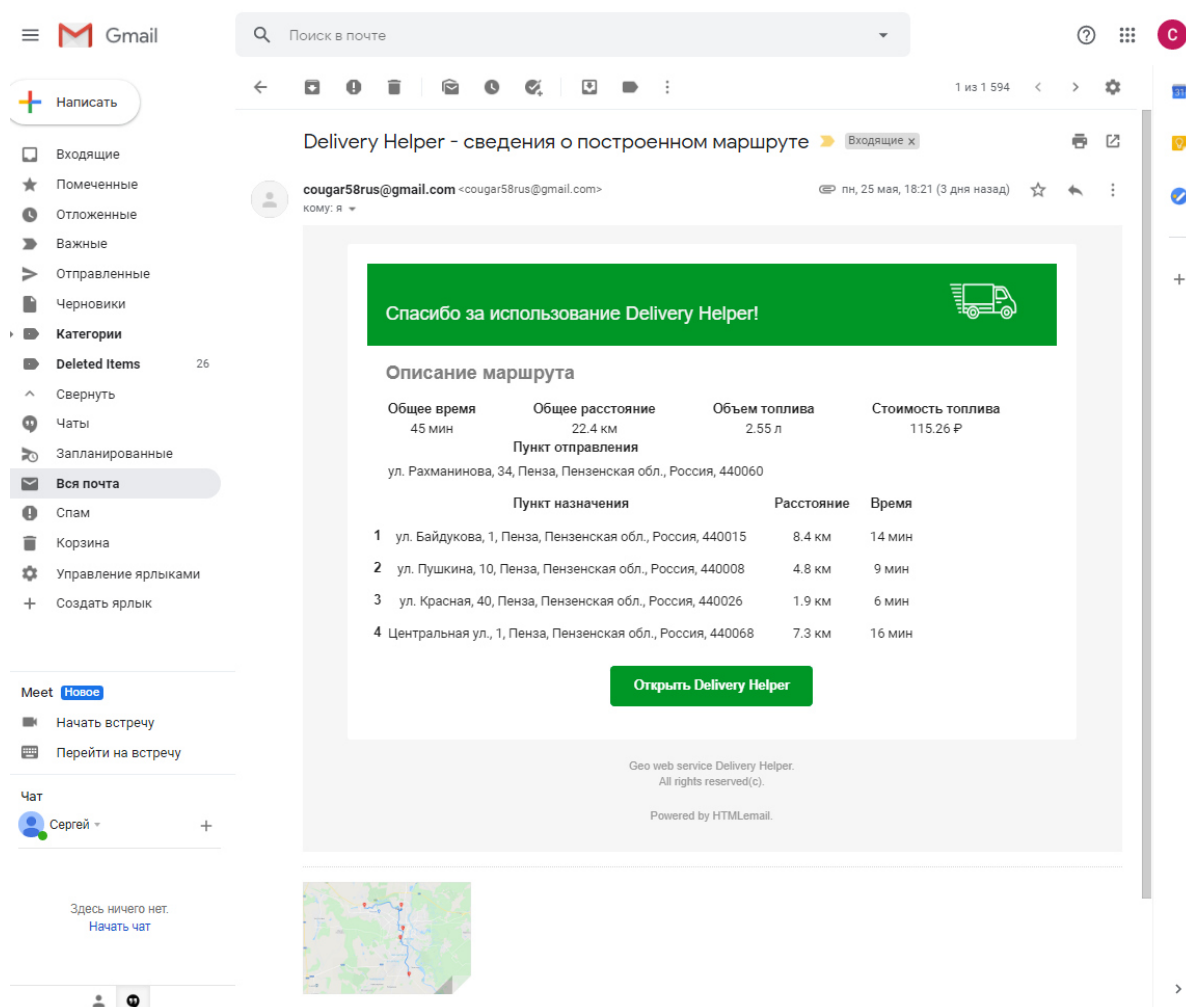


Рисунок 32 – Содержимое электронного письма

4.2 Вывод

Тестирование показало, что ПО для серверной и клиентской части работает так, как предполагалось, алгоритмы работоспособны, критических ошибок не выявлено. Проведя непрерывное тестирование в течение 1 часа, не было замечено «зависаний», потери связи с базой данных, замедления работы системы, следовательно, система готова к использованию.

Для проверки работоспособности системы требуется веб-сервер Apache, веб-браузер и база данных PostgreSQL.

Заключение

При выполнении выпускной квалификационной работы была разработана автоматизированная система оптимизации доставки грузов. Были разработаны алгоритмы сбора и сохранения пользовательских данных, создан веб-сервис, аккумулирующий все эти данные в единое целое. Также создана база данных для сохранения собранных сведений. Был реализован функционал, связанный с сохранением, редактированием и удалением избранных адресов пользователя, а также отправкой сведений о построенном маршруте на электронную почту для расширения возможностей разработанной системы.

В дальнейшем планируется разработка дополнительного функционала и алгоритмов для обработки данных внутри системы. В клиентской части программы возможно расширение функционала с помощью интеграции алгоритмов, связанных с формированием и подготовкой для печати документов, содержащих сведения о построенном маршруте.

Требования технического задания были выполнены в полном объеме и в установленные сроки.

Список использованных источников

1. Википедия [Электронный ресурс]: свободная энциклопедия. – веб - сайт. – Режим доступа: [https://ru.wikipedia.org/wiki/веб ресурсы](https://ru.wikipedia.org/wiki/веб_ресурсы) (Дата обращения 28.05.2020.)
2. Википедия [Электронный ресурс]: свободная энциклопедия. – веб - служба. – Режим доступа: <https://ru.wikipedia.org/wiki/Веб-служба> (Дата обращения 28.05.2020.)
3. Google Карты [Электронный ресурс] – Режим доступа: <https://www.google.ru/maps> (Дата обращения 28.05.2020)
4. Яндекс Карты [Электронный ресурс] – Режим доступа: <https://yandex.ru/maps/> (Дата обращения 28.05.2020)
5. 2ГИС [Электронный ресурс] – Режим доступа: <https://2gis.ru> (Дата обращения 28.05.2020)
6. Poncy.su [Электронный ресурс] – Режим доступа: <https://poncy.su> (Дата обращения 28.05.2020)
7. ISSA.ru [Электронный ресурс] – Режим доступа: <https://issa.ru/distance> (Дата обращения 28.05.2020)
8. Википедия [Электронный ресурс]: свободная энциклопедия. – IntelliJ IDEA. – Режим доступа: https://ru.wikipedia.org/wiki/IntelliJ_IDEA (Дата обращения 28.05.2020.)
9. Википедия [Электронный ресурс]: свободная энциклопедия. –HTML. – Режим доступа: <https://ru.wikipedia.org/wiki/HTML> (Дата обращения 28.05.2020.)
10. Википедия [Электронный ресурс]: свободная энциклопедия. – CSS. – Режим доступа: <https://ru.wikipedia.org/wiki/css> (Дата обращения 28.05.2020.)
11. Википедия [Электронный ресурс]: свободная энциклопедия. – javascript – Режим доступа: <https://ru.wikipedia.org/wiki/javascript>. (Дата обращения 28.05.2020.)

12. Википедия [Электронный ресурс]: свободная энциклопедия. – PostgreSQL – Режим доступа: <https://ru.wikipedia.org/wiki/PostgreSQL> (Дата обращения 28.05.2020.)
13. Google Directions API [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/148986/> (Дата обращения 28.05.2020.)
14. Google Distance Matrix API [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/148986/> (Дата обращения 28.05.2020.)
15. Google Maps API [Электронный ресурс] – Режим доступа: <http://www.scanex.ru/software/web/google-maps-api/> (Дата обращения 28.05.2020.)
16. Википедия [Электронный ресурс]: свободная энциклопедия. – Java – Режим доступа: <https://ru.wikipedia.org/wiki/Java> (Дата обращения 28.05.2020.)

ЛИСТИНГ КОДА ПРОГРАММНОГО ПРОДУКТА

Приложение А

MyServlet.java

```
package com.testervlet;

import javax.mail.Message;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.sql.*;
import java.util.ConcurrentModificationException;
import java.util.Random;
import java.util.Date;
import java.lang.ClassNotFoundException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.math.BigInteger;

@WebServlet(name = "MyServlet")
public class MyServlet extends HttpServlet {

    private String login;
    private String password;
    private String email;
    private String result;

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        System.out.println("I'm here.");
        login = request.getParameter("auth_login");
        password = request.getParameter("auth_password");
        password = md5Custom(password);
        authorization();
        response.getWriter().println(result);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        System.out.println("I'm here.");
        login = request.getParameter("reg_login");
        password = request.getParameter("reg_password");
        email = request.getParameter("reg_email");
        password = md5Custom(password);
        registration();
        response.getWriter().println(result);
    }

    public static String md5Custom(String st) {
        MessageDigest messageDigest = null;
        byte[] digest = new byte[0];

        try {
            messageDigest = MessageDigest.getInstance("MD5");
            messageDigest.reset();
            messageDigest.update(st.getBytes());
            digest = messageDigest.digest();
        } catch (NoSuchAlgorithmException e) {
            // тут можно обработать ошибку
            // возникает она если в передаваемый алгоритм в getInstance(,,,)
            не существует
        }
    }
}
```

```

        e.printStackTrace();
    }

    BigInteger bigInt = new BigInteger(1, digest);
    String md5Hex = bigInt.toString(16);

    while( md5Hex.length() < 32 ){
        md5Hex = "0" + md5Hex;
    }

    return md5Hex;
}

public void registration() {
    Connection connection;
    try {
        Class.forName("org.postgresql.Driver");
    } catch (ClassNotFoundException e) {
        System.out.println("Where is your Driver?");
        e.printStackTrace();
        return;
    }

    try {
        connection =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/test",
"postgres", "postgres");
    } catch (SQLException e) {
        System.out.println("Connection Failed");
        e.printStackTrace();
        return;
    }
    Statement statement;
    try {
        statement = connection.createStatement();
        ResultSet rs = statement.executeQuery(
            "SELECT * FROM users WHERE login = '"+ login +'"");
        if (rs.next()) {
            System.out.println(rs.getString("login"));
            result = "1";
            System.out.println("Login is founded.");
        }
        else {
            rs = statement.executeQuery(
                "INSERT INTO users(login, password, email) VALUES ('"
+ login + "','" + password + "','" + email + "') RETURNING 0"
            );
            if (rs.next()) {
                result = "y";
                System.out.println("Регистрация прошла успешно!");
            }
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}

public void authorization() {
    Connection connection;
    try {
        Class.forName("org.postgresql.Driver");

```

```

    } catch (ClassNotFoundException e) {
        System.out.println("Where is your Driver?");
        e.printStackTrace();
        return;
    }

    try {
        connection =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/test",
"postgres", "postgres");
    } catch (SQLException e) {
        System.out.println("Connection Failed");
        e.printStackTrace();
        return;
    }
    Statement statement;
    try {
        statement = connection.createStatement();
        ResultSet rs = statement.executeQuery(
            "SELECT login FROM users WHERE login='" + login + "'"
        );
        if (rs.next()) {
            System.out.println(rs.getString("login"));
            rs = statement.executeQuery(
                "SELECT password FROM users WHERE password='" +
password + "' AND login='" + login + "'"
            );
            if (rs.next()) {
                result = "y";
                System.out.println(rs.getString("password"));
            }
            else {
                result = "p";
                System.out.println("Password is not correct.");
            }
        }
        else {
            result = "l";
            System.out.println("Login is not founded.");
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}

public static String req()
{
    String str = "";
    try {
        Class.forName("org.postgresql.Driver");
    }
    catch (ClassNotFoundException e) {
        System.err.println (e);
        System.exit (-1);
    }
    try {
        // open connection to database
        Connection connection = DriverManager.getConnection(
            //"jdbc:postgresql://dbhost:port/dbname", "user",
"dbpass");
            "jdbc:postgresql://192.168.12.27:5432/test_db",

```

```

"postgres", "password");

    // build query, here we get info about all databases"
    String query = "SELECT adres FROM penza";

    // execute query
    Statement statement = connection.createStatement ();
    ResultSet rs = statement.executeQuery (query);

    // return query result
    while ( rs.next () ) {
        // display table name

        //System.out.println("PostgreSQL Query result: " +
rs.getString("adres"));

        str += rs.getString("adres");
        str += ";";
    }
    connection.close ();

}
catch (java.sql.SQLException e) {
    System.err.println (e);
    System.exit (-1);
}
return str;
}
}

```

```
<!--<%@ page contentType="text/html;charset=UTF-8" language="java" %>-->
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Delivery Helper</title>
<link rel="shortcut icon" href="delivery_logo.png" type="image/png">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link href='http://fonts.googleapis.com/css?family=PT+Sans:400,700'
rel='stylesheet' type='text/css'>

<meta name="viewport" content="initial-scale=1, maximum-scale=1, user-
scalable=no">
<link rel="stylesheet" href="https://openlayers.org/en/v3.20.1/css/ol.css"
type="text/css">

<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?libraries=places&sensor=false"></script>

<script src="https://code.jquery.com/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>

<!-- Bootstrap core CSS -->
<link href="css/bootstrap.css" rel="stylesheet">

<link rel="stylesheet" href="css/left-nav-style.css">
<link rel="stylesheet" href="css/waiting-balls.css">
<link rel="stylesheet" href="css/rounded-switch.css">

<style>
input::-webkit-outer-spin-button,
input::-webkit-inner-spin-button {
    /* display: none; <- Crashes Chrome on hover */
    -webkit-appearance: none;
    margin: 0; /* <-- Apparently some margin are still there even though
it's hidden */
}

.range-style {
    -webkit-appearance: none;
    width: 160px;
    height: 10px;
    background: #e4e4e4;
    border-radius: 5px;
    outline: none;
    margin-top: 5px;
}

.range-style::-webkit-slider-thumb {
    -webkit-appearance: none;
    width: 15px;
    height: 15px;
    border: 2px solid #16a085;
    background: #ffffff;
    border-radius: 50%;
    background-repeat: no-repeat;
}
```

```

background-position: center;
cursor: pointer;
transition: all 0.3s;
}

::-webkit-scrollbar { width: 0px; height: 3px; }
::-webkit-scrollbar-button { background-color: #666; }
::-webkit-scrollbar-track { background-color: #999; }
::-webkit-scrollbar-track-piece { background-color: #16a085; }
::-webkit-scrollbar-thumb { height: 30px; background-color: #666; border-
radius: 3px; }
::-webkit-scrollbar-corner { background-color: #999; }
::-webkit-resizer { background-color: #666; }

.text {
color: #ffffffde !important;
font-family: Roboto, Arial, sans-serif;
font-size: 15px;
text-decoration: none;
}
.text:hover{
color: #ffffff !important;
text-decoration: none;
}
.text:focus{
text-decoration: none;
}

.btn-group > .btn,
.btn-group-vertical > .btn { outline: none; background-color: #fff; }

a,a:focus,a:hover,a:active,button,
button:hover { outline: 0 !important; }
a:focus { text-decoration: none; }

.form-control:focus {
background-color: #fff;
}

#gmap_canvas {
width: 100%;
margin: 0px;
padding: 0;
border:solid 0px lightskyblue;
}

.iw-modal-btn {
background:#3bc1a0; /*цвет фона кнопки*/
color:whitesmoke; /*цвет шрифта*/
text-align:center;
display:inline-block;
padding:5px 10px; /*внутренние отступы*/
text-decoration:none;
font-size:14px; /*размер шрифта*/
margin-top:7px;
transition: all 0.5s ease;
border-radius: 0px;
}
.iw-modal-btn:hover {
color:#fff;
text-decoration:none;
background: #3dd0ae; /*цвет фона при наведении*/
}

```

```

/*CSS-свойства для заднего фона*/
.iw-modal {
  opacity: 0; /*изначально этот блок должен быть прозрачным*/
  background: rgba(0,0,0,0.7); /*задаём цвет фона*/
  pointer-events: none; /*делаем чтобы по элементу нельзя было кликнуть
когда он прозрачный*/
  position: fixed;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: 9999;
  transition: all 0.5s ease;
  margin: 0;
  padding: 0;
}
.iw-modal:target {
  opacity: 1; /*при клике блок становится видимым*/
  pointer-events: auto; /*теперь по блоку можно кликать мышкой*/
  overflow-y: auto; /*прокрутка по вертикали страницы*/
}
.iw-modal-wrapper {
  margin:auto; /*выравниваем модальное окно по центру по горизонтали*/
  margin-top:25vh; /*отступ сверху можно задавать в px, % или vh*/
}
/*CSS-свойства для блока, содержащего контент модального окна */
.iw-CSS-modal-inner {
  position: relative;
  background: #e1e1e1; /*цвет фона*/
  border: 0px solid #ccc; /*цвет и толщина рамки*/
  border-radius: 10px; /*радиус скругления углов*/
}
/*CSS-свойства заголовка модального окна */
.iw-modal-header {
  padding: 15px; /*внутренний отступ*/
  background: #16a085; /*цвет фона*/
  position: relative;
  border-radius: 10px 10px 0 0;
}
.iw-modal-title {
  font-size: 20px; /*размер шрифта*/
  color: whitesmoke; /*цвет шрифта*/
  font-weight: bold; /*жирность шрифта*/
  line-height: 1.5; /*высота строки*/
  margin-top: 0;
  margin-bottom: 0;
}
/*CSS для кнопки закрытия*/
.iw-close {
  position: absolute;
  top: 15px; /*отступ сверху*/
  right: 15px; /*отступ справа*/
  font-size: 24px; /*размер шрифта*/
  color: whitesmoke; /*цвет шрифта*/
  text-decoration: none;
}
.iw-close:hover, .iw-close:focus {
  color: white; /*цвет шрифта при наведении*/
  cursor: pointer;
  text-decoration: none;
}
/*CSS для блока с текстом*/

```

```

.iw-modal-text {
    padding: 15px 20px; /*внутренний отступ*/
}
/**MEDIA**/
@media (min-width: 550px) {
    .iw-modal-wrapper {
        max-width: 350px;
    }
}

header {
    margin-top: 0;
    position: relative;
    width: 100%;
    height: 50px;
    background: #16a085;
}
</style>

<script>
    function OnLoad() {
        var paramValue = location.search.substr(1);
        document.getElementById("log").innerHTML += ' ' + paramValue;
    }

    function reg() {
        var login = document.getElementById('login').value;
        var email = document.getElementById('email').value;
        var password1 = document.getElementById('pass1').value;
        var password2 = document.getElementById('pass2').value;

        if(login!=''&&password1!=''&&password2!=''&&email!='') {
            if(password1==password2) {
                $.ajax({
                    url: "MyServlet",
                    type: "GET",
                    dataType: "text",
                    data: ("reg_login=" + login + "&reg_password=" +
password1 + "&reg_email=" + email),
                    success: function (data) {
                        if (data[0] == "y") {
                            alert("Поздравляем, вы успешно
зарегистрировались!")
                        }
                        if (data[0] == "l") {
                            alert("Данный login уже занят!");
                            //$('#login').css("background", "pink");
                        }
                    },
                    error: function (e) {
                        alert("error");
                    }
                })
            }
            else
            {
                alert("Error: Пароли не совпадают!");
                /*$('#pass1').css("background", "pink");
                $('#pass2').css("background", "pink");*/
            }
        }
        else
        {

```

```

        alert("Error: Необходимо заполнить все поля!");
    }
}
function ClearReg(){
    $('#login').css("background", "white");
    $('#email').css("background", "white");
    $('#pass1').css("background", "white");
    $('#pass2').css("background", "white");

    document.getElementById("login").value = "";
    document.getElementById("email").value = "";
    document.getElementById("pass1").value = "";
    document.getElementById("pass2").value = "";
}
function auth() {
    var login = document.getElementById('loginauth').value;
    var password = document.getElementById('password').value;

    $('#loginauth').css("background", "white");
    $('#password').css("background", "white");

    if(login == "admin")
    {document.location.href="admin.html"}
    else {
        $.ajax({
            url: "MyServlet",
            type: "POST",
            dataType: "text",
            data: ("auth_login=" + login + "&auth_password=" +
password),

            success: function (data) {
                if (data[0] == "y")
                    document.location.href = "index2.html?" + login;
                if (data[0] == "l") {
                    alert("Не верный ввод имени пользователя.");
                    /*$('#loginauth').css("background", "pink");
                    $('#password').css("background", "pink");*/
                }
                if (data[0] == "p") {
                    alert("Не верный ввод пароля.");
                    /*$('#loginauth').css("background", "#16a085");
                    $('#password').css("background", "pink");*/
                    document.getElementById('password').value = '';
                }
            },
            error: function (e) {
                alert("error");
            }
        })
    }
}
function ClearAuth(){
    $('#loginauth').css("background", "white");
    $('#password').css("background", "white");

    document.getElementById("loginauth").value = "";
    document.getElementById("password").value = "";
}

new_id = $('.inputs').length+1;
function add_input(){
    if(new_id<9) {
        var inputs = $('.inputs input[type="text"]');
    }
}

```

```

        new_id += 1;
        $('inputs').append('<input id="point-' + new_id + '"
list="point-' + new_id + '-list" name="name[]" class="form-control"
style="width: 205px; float: left; margin-top: 5px" placeholder="Адрес
назначения"/>');
        if(document.getElementById('opt').value == 'opt_dist') {
            $('inputs').append('<input disabled id="weight-' + new_id
+ '" name="name[]" type="number" min="0" max="999" class="form-control"
style="width: 50px; display: inline-block; margin-top: 5px; margin-left: 5px"
placeholder="Бес"/>');
        }
        else if(document.getElementById('opt').value == 'opt_weight'){
            $('inputs').append('<input id="weight-' + new_id + '"
name="name[]" type="number" min="0" max="999" class="form-control"
style="width: 50px; display: inline-block; margin-top: 5px; margin-left: 5px"
placeholder="Бес"/>');
        }
    }
}

function delete_input() {
    var elem;

    if(new_id>1){
        elem = document.getElementById('point-' + new_id + '');
        elem.remove();
        elem = document.getElementById('weight-' + new_id + '');
        elem.remove();
        new_id -= 1;
    }
}

function calculateAndDisplayRoute(directionsService, directionsDisplay)
{
    var myroute;
    var i;
    var str;

    if(document.getElementById("switch-type").checked) {
        str = 'DRIVING';
    }
    else {
        str = 'WALKING';
    }

    if(new_id==1) {
        if (document.getElementById('start').value == '')
        {document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;}
        if (document.getElementById('point-1').value == '')
        {document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;}
        $('#calculations').append('<div id="1-time" style="display:
block"></div>');
        $('#calculations').append('<div id="origin_adr" style="display:
block"></div>');

        directionsService.route({
            origin: document.getElementById('start').value,
            destination: document.getElementById('point-1').value,
            travelMode: str
        }, function (response, status) {
            if (status === 'OK') {
                document.getElementById('success').style.display =

```

```

"block";

        directionsDisplay.setDirections(response);
        console.log(response);
        myroute = directionsDisplay.getDirections().routes[0];
        for (var i = 0; i < myroute.legs.length; i++) {
            var h = myroute.legs[i].duration.value/3600 ^ 0 ;
            var m = (myroute.legs[i].duration.value-h*3600)/60
            ^ 0 ;

            var s = myroute.legs[i].duration.value-h*3600-m*60
            ;

            if (h == 0){
                document.getElementById((i+1)+'-
time').innerHTML = m + ' мин';
            }
            else {
                document.getElementById((i+1)+'-
time').innerHTML = h + ' ч ' + m + ' мин';
            }
            document.getElementById('origin_adr').innerHTML =
myroute.legs[i].start_address;
        }
    } else {
        window.alert('Directions request failed due to ' +
status);
    }
});
} else {
    var adr_arr = new Array();
    var waypoints = new Array();

    if (document.getElementById('start').value == '')
{document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;} else
{adr_arr[0] = document.getElementById("start").value;}

    if(document.getElementById('opt').value == 'opt_dist')
    {
        for(i=1; i<=new_id; i++)
        {
            if (document.getElementById('point-'+ (i)).value == '')
{document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;}
            if (document.getElementById((i) + '-adr') != null) {
                adr_arr[i] = document.getElementById((i) + '-
adr').innerHTML;
            }
        }
    }
    else if(document.getElementById('opt').value == 'opt_weight')
    {
        for(i=1; i<=new_id; i++) {
            if (document.getElementById('point-'+ (i)).value == '')
{document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;}
            if (document.getElementById('weight-'+ (i)).value ==
'') {document.getElementById('wait').style.display = 'none'; alert("Не указан
вес груза!"); return;}
        }
    }

    if (document.getElementById('wl').value == '')
{document.getElementById('wait').style.display = 'none'; alert("Необходимо
указать грузоподъемность!"); return;}

```

```

        var wl_sum = 0;
        for(i=1; i<=new_id; i++) {
            wl_sum += parseInt(document.getElementById('weight-'+
(i)).value);
        }
        if (wl_sum > document.getElementById('wl').value)
{document.getElementById('wait').style.display = 'none'; alert("Суммарный вес
грузов превышает параметр грузоподъемности!"); return;}

        var arr = [];
        for(i=1; i<=new_id; i++) {
            arr[document.getElementById('point-'+ (i)).value] =
document.getElementById('weight-'+ (i)).value;
        }
        console.log('wl_sum = '+ wl_sum);
        console.log('wl = '+ document.getElementById('wl').value);
        console.log('начальный массив', arr);

        var tmp = [];
        for (var variable in arr) {
            tmp.push({
                address: variable,
                weight: arr[variable]
            });
        }

        //сортировка по убыванию
        var sorted = tmp.sort( function(one, two) {
            return two.weight - one.weight
        });

        // ИТОГОВЫЙ ВЫВОД
        console.log('вывод после сортировки', sorted);

        for(i=1; i<=new_id; i++) {
            adr_arr[i] = sorted[i-1].address;
        }
    }

    console.log(adr_arr);

    for(i=1; i<new_id; i++)
    {
        waypoints.push({location:adr_arr[i]});
    }

    for(i=1; i<=new_id; i++){
        $('#calculations').append('<div id="'+i+'-time"
style="display: block"></div>');
    }
    $('#calculations').append('<div id="origin_adr" style="display:
block"></div>');

    directionsService.route({
        origin: adr_arr[0],
        destination: adr_arr[new_id],
        waypoints: waypoints,
        /*optimizeWaypoints: true,*/
        travelMode: str
    }, function (response, status) {
        if (status === 'OK') {
            document.getElementById('success').style.display =

```

```

"block";

directionsDisplay.setDirections(response);
console.log(response);
document.getElementById('origin_adr').innerHTML =
directionsDisplay.getDirections().routes[0].legs[0].start_address;
myroute = directionsDisplay.getDirections().routes[0];
for (var i = 0; i < myroute.legs.length; i++) {
    var h = myroute.legs[i].duration.value/3600 ^ 0 ;
    var m = (myroute.legs[i].duration.value-h*3600)/60
^ 0 ;

    var s = myroute.legs[i].duration.value-h*3600-m*60
;

    if (h == 0){
        document.getElementById((i+1)+'-
time').innerHTML = m + ' мин';
    }
    else {
        document.getElementById((i+1)+'-
time').innerHTML = h + ' ч ' + m + ' мин';
    }

    if(document.getElementById('opt').value ==
'opt_weight'){
        document.getElementById((i+1)+'-adr').innerHTML
= directionsDisplay.getDirections().routes[0].legs[i].end_address;
        document.getElementById(i+1).innerHTML =
(directionsDisplay.getDirections().routes[0].legs[i].distance.value)/1000;
    }
}
} else {
    window.alert('Directions request failed due to ' +
status);
}
});
}

function calc_order(ind){
    var service = new google.maps.DistanceMatrixService();
    var distance;
    var i = 0, k = 0;
    var equal = 0;
    var str = ind;
    var mode;

    if(document.getElementById("switch-type").checked){
        mode = 'DRIVING';
    }
    else {
        mode = 'WALKING';
    }

    $('#calculations').append('<div id="'+ ind +'" style="display:
none"></div>');
    $('#calculations').append('<div id="'+ ind +'-adr" style="display:
none"></div>');
    console.log('ind = ' + ind);
    for(i=1; i<=new_id; i++) {
        service.getDistanceMatrix({
            origins: [document.getElementById((ind-1) +'-
adr').innerHTML],
            destinations: [document.getElementById('point-'+ i).value],

```

```

        travelMode: mode
    }, function (response, status) {
        if (status != google.maps.DistanceMatrixStatus.OK) {
            console.log('DistanceMatrix request failed due to ' +
status);
        } else {
            distance = response.rows[0].elements[0].distance.value
/ 1000;

            console.log('distance = ' + distance);
            console.log(response.destinationAddresses[0]);

            if ($('#'+ str).html() == "" && distance != 0 &&
(response.originAddresses[0] != response.destinationAddresses[0])) {
                equal = 0;
                for(k=1; k<str; k++)
                {
                    if(response.destinationAddresses[0] == $('#'+ k
+ '-adr').html()) {equal = 1}
                }

                if(equal == 0) {$('#'+ str).html(distance); $('#'+
str + '-adr').html(response.destinationAddresses[0]);}
            }
            else {
                if(distance<$('#'+ str).html() && distance != 0 &&
(response.originAddresses[0] != response.destinationAddresses[0])) {
                    equal = 0;
                    for(k=1; k<str; k++)
                    {
                        if(response.destinationAddresses[0] ==
$('#'+ k + '-adr').html()) {equal = 1}
                    }

                    if(equal == 0) {$('#'+
str).html(distance); $('#'+ str + '-
adr').html(response.destinationAddresses[0]);}
                }
            }
        }
    });
}

function onMouseover() {
    document.getElementById("calculations").innerHTML = '';
    console.log('-----');

    document.getElementById('wait').style.display = "block";

    var service = new google.maps.DistanceMatrixService();

    var distance;
    var empty = 0;
    var i = 0, j = 0;
    var timeout = 600;
    var mode;

    if(document.getElementById("switch-type").checked) {
        mode = 'DRIVING';
    }
    else {
        mode = 'WALKING';
    }
}

```

```

        if (document.getElementById('start').value == '') {empty=1;}
        for(j=1; j<=new_id; j++)
        {
            if (document.getElementById('point-'+ (j)).value == '')
{empty=1;}
        }
        if(empty==0) {
            $('#calculations').append('<div id="1" style="display:
none"></div>');
            $('#calculations').append('<div id="1-adr" style="display:
none"></div>');
            for(i=1; i<=new_id; i++) {
                service.getDistanceMatrix({
                    origins: [document.getElementById('start').value],
                    destinations: [document.getElementById('point-'+
i).value],
                    travelMode: mode
                }, function (response, status) {
                    if (status != google.maps.DistanceMatrixStatus.OK) {
                        console.log('DistanceMatrix request failed due to '
+ status);
                    } else {
                        distance =
response.rows[0].elements[0].distance.value / 1000;
                        console.log('distance = ' + distance);
                        console.log(response.destinationAddresses[0]);

                        if ($('#1').html() == "") {
                            $('#1').html(distance);
                            $('#1-
adr').html(response.destinationAddresses[0]);
                        }
                        else {
                            if(distance<$('#1').html())
{$('#1').html(distance);$('#1-adr').html(response.destinationAddresses[0]);}
                        }
                    }
                });
            }

            if(new_id >= 2)
            {
                for(i=2; i<=new_id; i++) {
                    setTimeout(calc_order, timeout, i);
                    timeout += 600;
                }
            }
            setTimeout(function
() {document.getElementById('wait').style.display = 'none'}, (new_id+1)*600);
        } else {console.log("Для промежуточных расчетов должны быть
заполнены все адреса!");}
    }
}

</script>

</head>
<body onload="OnLoad()">

<input type="checkbox" id="nav-toggle" hidden>

<nav class="nav" id="nav">
    <label for="nav-toggle" class="nav-toggle" onclick></label>

```

```

<h2 class="logo">
  <a style="font-family: monospace; font-size: 25px; color:
whitesmoke;">Delivery helper</a>
</h2>
<h2 id="log" align="center" style="font-size: 20px; color: #dadada;"></h2>

<label style="color:whitesmoke; margin-top: 1em; font-family: monospace;
font-weight: 500">Пункт отправления:</label>

<div class="input-group" >
  <input type="text" class="form-control" placeholder="Адрес отправления"
id="start">
  <div class="input-group-btn">
    <button class="btn btn-default" onclick="getPosition()"><i
class="glyphicon glyphicon-globe"></i></button>
  </div>
</div>

<div style="float: left"><label for="opt" style="color:whitesmoke; margin-
top: 2em; font-family: monospace; font-weight: 500">Пункт (-ы)
назначения:</label></div>
<div style="display: inline-block">
  <div class="btn-group" style="margin-top: 1em; margin-left: 19px">
    <button class="btn btn-default" onclick=add_input()><i class="glyphicon
glyphicon-plus"></i></button>
    <button class="btn btn-default" onclick=delete_input()><i
class="glyphicon glyphicon-minus"></i></button>
  </div>
</div>
<div class="inputs" style="margin-top: 0.5em" align="center">
  <input type="text" class="form-control" style="width: 205px; margin-top:
0px; float: left" placeholder="Адрес назначения" id="point-1">
  <input disabled type="number" min="0" max="999" class="form-control"
style="width: 50px; margin-top: 0px; margin-left: 5px; display: inline-block"
placeholder="Вес" id="weight-1">
</div>
<div style="margin-top: 1em; float: left"><button class="btn"
id="calculate">Построить маршрут</button></div>

<a class="text" style="display: inline-block; margin-left: 15px; margin-
top: 22px" href="#hidden1" onclick="view('hidden1'); return
false">Параметры</a>
<div id="hidden1" style="margin-top: 1em; background: whitesmoke; border-
radius: 5px; padding: 5px; display: none">

  <div align="center" style="font-family: monospace">Способ
передвижения:</div>
  <div align="center">
    <div id="walk" style="display: inline-block; font-family: monospace;
padding: 0 5px 0 5px">Пешком</div>
    <label class="switch" style="display: inline-block; margin-left: 1em;
margin-right: 1em">
      <input id="switch-type" type="checkbox" checked>
      <span class="slider round"></span>
    </label>
    <div id="auto" style="display: inline-block; font-family: monospace;
color: whitesmoke; background: #16a085; padding: 0 5px 0 5px; border-radius:
5px">Автомобиль</div>
  </div>

<script>
  document.getElementById("switch-type").addEventListener("change",

```

```

function() {
    if(document.getElementById("switch-type").checked) {
        document.getElementById("auto").style.color = "whitesmoke";
        document.getElementById("auto").style.background = "#16a085";
        document.getElementById("auto").style.borderRadius = "5px";
        document.getElementById("walk").style.color = "#333";
        document.getElementById("walk").style.background = "none";
        document.getElementById("walk").style.borderRadius = "none";
        document.getElementById("price").disabled = false;
        document.getElementById("consumption").disabled = false;
    }
    else{
        document.getElementById("walk").style.color = "whitesmoke";
        document.getElementById("walk").style.background = "#16a085";
        document.getElementById("walk").style.borderRadius = "5px";
        document.getElementById("auto").style.color = "#333";
        document.getElementById("auto").style.background = "none";
        document.getElementById("auto").style.borderRadius = "none";
        document.getElementById("price").disabled = true;
        document.getElementById("consumption").disabled = true;
    }
});
</script>

<div style="margin-top: 1em; font-family: monospace; font-size: 15px"
align="center">Грузоподъемность:
    <input disabled type="number" min="50" max="3500" class="form-control"
style="display: inline-block; width: 57px" value="" placeholder="Кг."
id="w1">
</div>

<span style="float: left; margin-left: 10px; margin-top: 0.5em; font-
family: monospace; font-size: 15px">50</span>
<div style="float: left; margin-left: 10px; margin-top: 0.5em">
    <input style="width: 160px;" id="wheightlift" type="range"
class="range-style" min="50" max="3500" step="10" value="250" disabled/>
</div>
<span style="float: left; margin-left: 10px; margin-top: 0.5em; font-
family: monospace; font-size: 15px">3500</span>

<script>
    document.getElementById("wheightlift").oninput = function() {
        document.getElementById("w1").value = this.value.toString();
    };
</script>

<div>
    <div align="center"><label for="opt" style="margin-top: 1em; font-
family: monospace; font-weight: 500">Оптимизация маршрута:</label></div>
    <select style="text-align-last: center" class="form-control" id="opt">
        <option value="opt_dist">С учетом расстояния</option>
        <option value="opt_weight">С учетом веса груза</option>
    </select>
</div>

<script>
    document.getElementById("opt").addEventListener("change", function()
{
    if(document.getElementById('opt').value == 'opt_dist') {
        document.getElementById("wheightlift").disabled = true;
        document.getElementById("w1").disabled = true;
        for(i=1; i<=new_id; i++){
            document.getElementById('weight-'+ (i)).disabled = true;

```

```

    }
  }
  else if (document.getElementById('opt').value == 'opt_weight') {
    document.getElementById("weightlift").disabled = false;
    document.getElementById("wl").disabled = false;
    for (i=1; i<=new_id; i++) {
      document.getElementById('weight-'+ (i)).disabled = false;
    }
  }
});
</script>

<div style="margin-top: 1em; padding: 5px">
  <div style="font-family: monospace; font-size: 14px; float: left">Цена
топлива (□/литр) </div>
  <input id="price" type="number" min="1" max="99" class="form-control"
style="width: 61px; margin-top: -7px; margin-left: auto" placeholder="□">
</div>

  <div style="padding: 5px">
    <div style="font-family: monospace; font-size: 14px; float:
left">Расход топлива (л/100км) </div>
    <input id="consumption" type="number" min="1" max="99" class="form-
control" style="width: 61px; margin-top: -7px; margin-left: auto"
placeholder="□ л">
  </div>

</div>

<div id="calculations" style="margin-top: 1em; display: none; color:
whitesmoke"></div>

<script>
  function view(n) {
    style = document.getElementById(n).style;
    style.display = (style.display == 'block') ? 'none' : 'block';
  }
</script>

<div id="dist_dur" align="left" style="color: whitesmoke; margin-top: 1em;
background: #3bc1a0; border-radius: 5px; padding: 5px 10px 5px 10px; display:
none">
  <div>
    <div style="font-family: monospace; font-size: 15px; float:
left">Расстояние</div>
    <div id="total" style="font-family: monospace; font-size: 15px"
align="right"></div>
  </div>
  <div>
    <div style="font-family: monospace; font-size: 15px; float:
left">Время</div>
    <div id="duration" style="font-family: monospace; font-size: 15px"
align="right"></div>
  </div>

  <div id="petrol_label" style="display: none">
    <div align="center" style="display: block">Расход топлива</div>

  <div>
    <div style="font-family: monospace; font-size: 15px; float:
left">Объем</div>
    <div id="amount" style="font-family: monospace; font-size: 15px"
align="right"></div>

```

```

    </div>
  </div>
  <div style="font-family: monospace; font-size: 15px; float:
left">Стоимость</div>
    <div id="total_price" style="font-family: monospace; font-size: 15px"
align="right"></div>
  </div>
</div>

<div id="wait" style="margin-top: 1em; color: whitesmoke; display: none">
  <div class="windows8">
    <div class="wBall" id="wBall_1">
      <div class="wInnerBall"></div>
    </div>
    <div class="wBall" id="wBall_2">
      <div class="wInnerBall"></div>
    </div>
    <div class="wBall" id="wBall_3">
      <div class="wInnerBall"></div>
    </div>
    <div class="wBall" id="wBall_4">
      <div class="wInnerBall"></div>
    </div>
    <div class="wBall" id="wBall_5">
      <div class="wInnerBall"></div>
    </div>
  </div>
</div>

<div id="success" style="display: none">
  <label style="color:whitesmoke; margin-top: 1em; font-family: monospace;
font-weight: 500">Подробный обзор маршрута:</label>
  <div id="hidden2" style="background: whitesmoke; border-radius: 5px;
margin-top: 0em; padding: 5px">
    <div id="panel" style="" align="center"></div>
  </div>
</div>

</nav>

<div class="mask-content"></div>

<div style="margin-left: 0%">
  <header>
    <h2 align="center" style="margin-left: 40px; margin-top: 10px; font-
family: monospace; font-size: 25px; color: whitesmoke; float: left">Delivery
helper</h2>
    <div style="margin-right: 1em" align="right">
      <a style="margin-top: 10px; margin-right: 1em" href="#iw-modalreg"
class="iw-modal-btn"><div class="glyphicon glyphicon-pencil" style="margin-
right: 5px"></div>Регистрация</a>
      <a style="margin-top: 10px" href="#iw-modalauth" class="iw-modal-
btn"><div class="glyphicon glyphicon-log-out" style="margin-right:
5px"></div>Вход</a>
    </div>
  </header>

  <div id="mapBox">
    <div id="gmap_canvas"></div>
  </div>

```

```

<script>
    document.getElementById("wheightlift").addEventListener("change",
function() {
    document.getElementById("wl").value = this.value;
    });

    document.getElementById("wl").addEventListener("change", function() {
        document.getElementById("wheightlift").value = this.value;
    });

    var visota = $(window).height();
    visota=visota-50;
    $( '#gmap_canvas' ).css('height', visota);

    $(window).on('resize', function(){
        var visota = $(window).height();
        visota=visota-50;
        $( '#gmap_canvas' ).css('height', visota);
    });

    var geocoder;
    var map;

    function initMap() {
        var directionsService = new google.maps.DirectionsService;
        var directionsDisplay = new
google.maps.DirectionsRenderer({draggable: true});
        geocoder = new google.maps.Geocoder();
        var coordinates = {lat: 53.200, lng: 45.004};
        map = new google.maps.Map(document.getElementById('gmap_canvas'), {
            zoom: 14,
            center: coordinates,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        });
        directionsDisplay.setMap(map);
        directionsDisplay.setPanel(document.getElementById('panel'));

        var onChangeHandler = function(){
            var timeout = 600*(new_id+1);
            setTimeout(calculateAndDisplayRoute, timeout,
directionsService, directionsDisplay);
        };

        document.getElementById('calculate').addEventListener('click',
onMouseover);
        document.getElementById('calculate').addEventListener('click',
onChangeHandler);

        directionsDisplay.addListener('directions_changed', function() {
            computeTotalDistance(directionsDisplay.getDirections());
        });
    }

    function computeTotalDistance(result) {
        var total = 0;
        var duration = 0;
        var myroute = result.routes[0];
        for (var i = 0; i < myroute.legs.length; i++) {
            total += myroute.legs[i].distance.value;
            duration += myroute.legs[i].duration.value;
        }

        document.getElementById('total').innerHTML =

```

```

(total/1000).toFixed(1) + ' км';

var h = duration/3600 ^ 0 ;
var m = (duration-h*3600)/60 ^ 0 ;
var s = duration-h*3600-m*60 ;

console.log('h = ' + h);
console.log('m = ' + m);
console.log('s = ' + s);

if (h == 0) {
    //console.log((m<10?"0"+m:m)+" мин. "+(s<10?"0"+s:s)+" сек.");
    console.log(m+" мин. "+ s +" сек.");
    document.getElementById('duration').innerHTML = m +' мин';
}
else {
    //console.log((h<10?"0"+h:h)+" ч. "+(m<10?"0"+m:m)+" мин. "+(s<10?"0"+s:s)+" сек.");
    console.log(h+" ч. "+ m +" мин. "+ s +" сек.");
    document.getElementById('duration').innerHTML = h +' ч ' + m +' мин';
}

if(document.getElementById("switch-type").checked &&
document.getElementById("price").value != '' &&
document.getElementById("consumption").value != ''){
    var dst = (total/1000).toFixed(1);
    var amount = (document.getElementById("consumption").value/100
* dst).toFixed(2);
    console.log("amount = "+ amount);
    var price = (document.getElementById("price").value *
amount).toFixed(2);
    console.log("total petrol price = "+ price);

    document.getElementById('petrol_label').style.display =
"block";

    document.getElementById('amount').innerHTML = amount +' л';
    document.getElementById('total_price').innerHTML = price +' ₺';
}
else
{
    document.getElementById('petrol_label').style.display = "none";
}

document.getElementById('dist_dur').style.display = "block";
}

function getPosition() {
    var geocoder = new google.maps.Geocoder();
    var latlng;

    navigator.geolocation.getCurrentPosition(function (position) {
        latlng = new google.maps.LatLng(position.coords.latitude,
position.coords.longitude);
        geocoder.geocode({ 'latLng': latlng }, function(results,
status) {
            $('#start').val(results[0].formatted_address);
        });
    });
}

// инициализация
google.maps.event.addDomListener(window, 'load', initMap);

```

```

</script>

</div>

<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC-
2n7pGa6eeVMXequ_pH7ypqp1nyx4tGE&libraries=places"></script>

<!--Разметка для модального окна---->
<!--Регистрация---->
<div id="iw-modalreg" class="iw-modal">
  <div class="iw-modal-wrapper">
    <div class="iw-CSS-modal-inner">
      <div class="iw-modal-header">
        <h3 class="iw-modal-title" style="font-family: monospace"
align="center">Регистрация</h3>
        <a href="#close" title="Закрыть" class="iw-close"
onclick="ClearReg() ">x</a>
      </div>
      <div class="iw-modal-text">
        <div style="margin-left: 0px" align="center">
          <label class="control-label">Имя пользователя</label>
          <div class="input-group" style="width: 30%;margin-top: 5px" >
            <span class="input-group-addon"><span class="glyphicon glyphicon-
user"></span></span>
            <input type="text" class="form-control" style="width: 250px"
placeholder="Введите имя пользователя" id="login">
          </div>
        </div>

        <div style="margin-left: 0px; margin-top: 10px" align="center">
          <label class="control-label">Электронная почта</label>
          <div class="input-group" style="width: 30%;margin-top: 5px" >
            <span class="input-group-addon"><span class="glyphicon glyphicon-
envelope"></span></span>
            <input type="email" class="form-control" style="width: 250px"
placeholder="Введите адрес эл. почты" id="email">
          </div>
        </div>

        <div style="margin-left: 0px; margin-top: 10px" align="center">
          <label class="control-label">Пароль пользователя</label>
          <div class="input-group" style="width: 30%;margin-top: 5px" >
            <span class="input-group-addon"><span class="glyphicon glyphicon-
lock"></span></span>
            <input type="password" required class="form-control"
style="width: 250px" placeholder="Введите пароль" id="pass1">
          </div>
        </div>

        <div style="margin-left: 0px" align="center">
          <div class="input-group" style="width: 30%;margin-top: 10px" >
            <span class="input-group-addon"><span class="glyphicon glyphicon-
lock"></span></span>
            <input type="password" required class="form-control"
style="width: 250px" placeholder="Подтвердите пароль" id="pass2">
          </div>
        </div>

        <div style="margin-top: 10px; margin-left: 0px" class="form-group">
          <div class="col-offset-4 col-4" align="center" style="margin-left:
0px">
            <button type="button" class="btn btn-info" style="width: auto;

```

```

background: #16a085" onclick="reg();">
    <div class="glyphicon glyphicon-ok"></div>
    Зарегистрироваться
</button>
</div>
</div>
</div>
</div>
</div>
</div>

<!--Авторизация-->
<div id="iw-modalauth" class="iw-modal">
    <div class="iw-modal-wrapper">
        <div class="iw-CSS-modal-inner">
            <div class="iw-modal-header">
                <h3 class="iw-modal-title" style="font-family: monospace"
align="center">Авторизация</h3>
                <a href="#close" title="Закрыть" class="iw-close"
onclick="ClearAuth()">x</a>
            </div>
            <div class="iw-modal-text">
                <div align="center">
                    <label class="control-label">Имя пользователя</label>
                    <div class="input-group" style="width: 33%">
                        <span class="input-group-addon"><span class="glyphicon
glyphicon-user"></span></span>
                        <input type="text" class="form-control" style="width: 250px"
placeholder="Введите имя пользователя" id="loginauth">
                    </div>
                </div>

                <div align="center">
                    <label style="margin-top: 15px" class="control-
label">Пароль</label>
                    <div class="input-group" style="width: 33%">
                        <span class="input-group-addon"><span class="glyphicon
glyphicon-lock"></span></span>
                        <input type="password" class="form-control" style="width:
250px" placeholder="Введите пароль" id="password">
                    </div>
                </div>

                <div style="margin-top: 20px" class="form-group">
                    <div class="col-offset-4 col-4" align="center" style="margin-
left: 0px">
                        <button type="button" class="btn btn-info" style="width: auto;
margin-left: 0px; background: #16a085" onclick="auth();">
                            <div class="glyphicon glyphicon-ok"></div>
                            Войти
                        </button>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

<!--end.Разметка для модального окна-->

</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
  <title>Delivery Helper</title>
  <link rel="shortcut icon" href="delivery_logo.png" type="image/png">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-
scalable=no">
  <link rel="stylesheet"
href="https://openlayers.org/en/v3.20.1/css/ol.css" type="text/css">
  <!-- The line below is only needed for old environments like Internet
Explorer and Android 4.x -->
  <script
src="https://cdn.polyfill.io/v2/polyfill.min.js?features=requestAnimationFrame,Element.prototype.classList,URL"></script>

  <script type="text/javascript"
src="https://www.google.com/jsapi"></script>
  <script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?libraries=places&sensor=false"><
/script>

  <script src="https://openlayers.org/en/v3.20.1/build/ol.js"></script>

  <script src="https://code.jquery.com/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></s
cript>

  <!-- SendMail SMTP -->
  <script src="https://smtpjs.com/v3/smtp.js"></script>

  <!-- Screenshot of div -->
  <script type="text/javascript" src="js/html2canvas.js"></script>

  <!-- Bootstrap core CSS -->
  <link href="css/bootstrap.css" rel="stylesheet">

  <link rel="stylesheet" href="css/left-nav-style.css">
  <link rel="stylesheet" href="css/waiting-balls.css">
  <link rel="stylesheet" href="css/rounded-switch.css">

  <style>
    input::-webkit-outer-spin-button,
    input::-webkit-inner-spin-button {
      /* display: none; <- Crashes Chrome on hover */
      -webkit-appearance: none;
      margin: 0; /* <-- Apparently some margin are still there even
though it's hidden */
    }

    .holder:hover .block{
      display:block;
    }
  </style>

  <script>
    function OnLoad() {
      var paramValue = location.search.substr(1);
      document.getElementById("log").innerHTML += paramValue;

      var login = document.getElementById('log').innerHTML;

```

```

function onload1() {
    $.ajax({
        url: "NewHAddress",
        type: "POST",
        dataType: "text",
        data: ("log=" + login),
        success: function (data) {
            if (data != "") {
                $('#haddr').html(data);

                if (data.substr(0,data.length-2) != "Адрес не
задан")
                {
                    var z = document.createElement("OPTION");
                    z.setAttribute("value",
data.substr(0,data.length-2));

document.getElementById("startlist").appendChild(z);
                    var z = document.createElement("OPTION");
                    z.setAttribute("value",
data.substr(0,data.length-2));
                    document.getElementById("point-1-
list").appendChild(z);
                }
            }
            else{
                document.getElementById('haddr').innerHTML =
"Адрес не задан";
            }

        },
        error: function (e) {
            alert("error");
        }
    })
}
function onload2() {
    $.ajax({
        url: "NewWAddress",
        type: "POST",
        dataType: "text",
        async: "true",
        data: ("log=" + login),
        success: function (data) {
            if (data != "") {
                $('#waddr').html(data);

                if (data.substr(0,data.length-2) != "Адрес не
задан")
                {
                    var z = document.createElement("OPTION");
                    z.setAttribute("value", data.substr(0,
data.length - 2));

document.getElementById("startlist").appendChild(z);
                    var z = document.createElement("OPTION");
                    z.setAttribute("value", data.substr(0,
data.length - 2));
                    document.getElementById("point-1-
list").appendChild(z);
                }
            }
        }
    })
}

```

```

        else{
            document.getElementById('waddr').innerHTML =
"Адрес не задан";
        }

    },
    error: function (e) {
        alert("error");
    }
    })
}
function onload3() {
    $.ajax({
        url: "NewFavLocOne",
        type: "POST",
        dataType: "text",
        data: ("log=" + login),
        success: function (data) {
            if (data != "") {
                $('#favloc1addr').html(data);

                if (data.substr(0,data.length-2) != "Адрес не
задан") {
                    var z = document.createElement("OPTION");
                    z.setAttribute("value", data.substr(0,
data.length - 2));

                    document.getElementById("startlist").appendChild(z);
                    var z = document.createElement("OPTION");
                    z.setAttribute("value", data.substr(0,
data.length - 2));

                    document.getElementById("point-1-
list").appendChild(z);
                }
            }
            else{
                document.getElementById('favloc1addr').innerHTML
= "Адрес не задан";
            }
        },
        error: function (e) {
            alert("error");
        }
    })
}
function onload4() {
    $.ajax({
        url: "NewFavLocTwo",
        type: "POST",
        dataType: "text",
        data: ("log=" + login),
        success: function (data) {
            if (data != "") {
                $('#favloc2addr').html(data);

                if (data.substr(0,data.length-2) != "Адрес не
задан")
                {
                    var z = document.createElement("OPTION");
                    z.setAttribute("value", data.substr(0,
data.length - 2));

```

```

document.getElementById("startlist").appendChild(z);
    var z = document.createElement("OPTION");
    z.setAttribute("value", data.substr(0,
data.length - 2));
    document.getElementById("point-1-
list").appendChild(z);
    }
    }
    else{
        document.getElementById('favloc2addr').innerHTML
= "Адрес не задан";
    }

    },
    error: function (e) {
        alert("error");
    }
    })
}
function onload5() {
    $.ajax({
        url: "NewFavLocThree",
        type: "POST",
        dataType: "text",
        data: ("log=" + login),
        success: function (data) {
            if (data != "") {
                $('#favloc3addr').html(data);

                if (data.substr(0,data.length-2) != "Адрес не
задан")
                {
                    var z = document.createElement("OPTION");
                    z.setAttribute("value", data.substr(0,
data.length - 2));

document.getElementById("startlist").appendChild(z);
                    var z = document.createElement("OPTION");
                    z.setAttribute("value", data.substr(0,
data.length - 2));

document.getElementById("point-1-
list").appendChild(z);
                }
            }
            else{
                document.getElementById('favloc3addr').innerHTML
= "Адрес не задан";
            }

            },
            error: function (e) {
                alert("error");
            }
            })
        }
        function onloadEmail() {
            $.ajax({
                url: "EmailServlet",
                type: "POST",
                dataType: "text",
                data: ("log=" + login),
                success: function (data) {
                    if (data != "") {

```

```

        $('#email').html('(' + data.substr(0, data.length
- 2) + ')');
        document.getElementById('reciever-email').value =
data.substr(0, data.length - 2);
    }
    else{
        document.getElementById('email').innerHTML =
"Адрес эл. почты не задан";
    }

    },
    error: function (e) {
        alert("error");
    }
    })
}

onload1();
onload2();
onload3();
onload4();
onload5();
onloadEmail();
}

function convertCanvasToImage() {
    if(document.getElementById('map-image')){
        console.log("Удалился map-image");
        document.getElementById('map-image').remove();
    }
    var canvas = document.getElementById('map-screen'),
    dataUrl = canvas.toDataURL(/"image/png"/),
    imageFoo = document.createElement('img');
    imageFoo.crossorigin = "anonymous";
    imageFoo.id = "map-image";
    imageFoo.src = dataUrl;

    var mapscr_ratio = canvas.height/canvas.width;
    console.log("mapscr_ratio = " + mapscr_ratio);

    // Style your image here
    imageFoo.style.width = '700px';
    imageFoo.style.height = mapscr_ratio*imageFoo.style.width + 'px';
    imageFoo.style.display = "none";

    // After you are done styling it, append it to the BODY element
    document.body.appendChild(imageFoo);
}

function SendMail() {
    var reciever = document.getElementById("reciever-email").value;
    var sender = "cougar58rus@gmail.com";
    var logo = '';

    var stats = '';

    if(document.getElementById("switch-type").checked &&
document.getElementById("price").value != '' &&
document.getElementById("consumption").value != ''){
        stats =
        '<table style="border-spacing: 0 5px; font-family: "Open

```

```

Sans", sans-serif;">' +
        '<tr>' +
            '<th align="center" style="padding: 0 0 0
10px;"></th>' +
            '<th align="center" style="padding: 0 0 0
10px;">Общее время</th>' +
            '<th align="center" style="padding: 0 0 0
60px;">Общее расстояние</th>' +
            '<th align="center" style="padding: 0 0 0
60px;">Объем топлива</th>' +
            '<th align="center" style="padding: 0 0 0
60px;">Стоимость топлива</th>' +
        '</tr>' +
        '<tr>' +
            '<th align="center" style="padding: 0 0 0
10px;"></th>' +
            '<td align="center" style="padding: 0 0 0
10px;">' + document.getElementById("duration").innerHTML + '</td>' +
            '<td align="center" style="padding: 0 0 0
60px;">' + document.getElementById("total").innerHTML + '</td>' +
            '<td align="center" style="padding: 0 0 0
60px;">' + document.getElementById("amount").innerHTML + '</td>' +
            '<td align="center" style="padding: 0 0 0
60px;">' + document.getElementById("total_price").innerHTML + '</td>' +
        '</tr>' +
        '</table>';
    }
    else {
        stats =
        '<table style="border-spacing: 0 5px; font-family: "Open
Sans", sans-serif;">' +
            '<tr>' +
            '<th align="center" style="padding: 0 0 0
20px;"></th>' +
            '<th align="center" style="padding: 0 0 0
60px;">Общее время</th>' +
            '<th align="center" style="padding: 0 0 0
60px;">Общее расстояние</th>' +
            '</tr>' +
            '<tr>' +
            '<th align="center" style="padding: 0 0 0
20px;"></th>' +
            '<td align="center" style="padding: 0 0 0
60px;">' + document.getElementById("duration").innerHTML + '</td>' +
            '<td align="center" style="padding: 0 0 0
60px;">' + document.getElementById("total").innerHTML + '</td>' +
            '</tr>' +
            '</table>';
    }

    var origin = document.getElementById('origin_adr').innerHTML;
    var destinations = '';

    for(var i=1; i<=new_id; i++){
        var s = parseFloat(document.getElementById(i).innerHTML);
        destinations += '<tr>' +
            '<th align="center" style="padding: 15px 0 0 0;">' + i
+ '</th>' +
            '<td align="center" style="padding: 15px 0 0 0;">' +
document.getElementById(i+'-adr').innerHTML + '</td>' +
            '<td align="center" style="padding: 15px 0 0
20px;">' + (s).toFixed(1) + ' км</td>' +
            '<td align="center" style="padding: 15px 0 0

```

```

20px;">' + document.getElementById(i+'-time').innerHTML + '</td>' +
    '</tr>';
    }

    var source = document.getElementById("map-image").src;
    var message;

    Email.send({
        SecureToken : "02bb7fb1-84e7-4e0b-a80a-99f6b40116e9",
/*02bb7fb1-84e7-4e0b-a80a-99f6b40116e9*/
        To : reciever,
        From : sender,
        Subject : "Delivery Helper - сведения о построенном
маршруте",
        Body : message,
        Attachments : [
            {
                name : "map_overview.png",
                data : source
            }
        ]
    }).then(
        alert('Данные о маршруте отправлены на эл. почту: ' + reciever
+ '!!')
    );
}

new_id = $('.inputs').length+1;
function add_input(){
    if(new_id<9) {
        var inputs = $('.inputs input[type="text"]');
        new_id += 1;
        $('.inputs').append('<input id="point-' + new_id + '"
list="point-' + new_id + '-list" name="name[]" class="form-control"
style="width: 205px; float: left; margin-top: 5px" placeholder="Адрес
назначения"/>');
        if(document.getElementById('opt').value == 'opt_dist') {
            $('.inputs').append('<input disabled id="weight-' +
new_id + '" name="name[]" type="number" min="0" max="999" class="form-
control" style="width: 50px; display: inline-block; margin-top: 5px; margin-
left: 5px" placeholder="Вес"/>');
        }
        else if(document.getElementById('opt').value ==
'opt_weight'){
            $('.inputs').append('<input id="weight-' + new_id + '"
name="name[]" type="number" min="0" max="999" class="form-control"
style="width: 50px; display: inline-block; margin-top: 5px; margin-left: 5px"
placeholder="Вес"/>');
        }
        var str = '';
        for (var i = 0; i <
document.getElementById("startlist").options.length; i++) {str += '<option
value="' +
document.getElementById("startlist").getElementsByTagName("option")[i].value
+ '"/>';}
        $('.inputs').append('<datalist id="point-' + new_id + '-
list">' + str + '</datalist>');
    }
}

function delete_input() {
    var elem;

    if(new_id>1){
        elem = document.getElementById('point-' + new_id + '');

```

```

        elem.remove();
        elem = document.getElementById('point-' + new_id + '-list');
        elem.remove();
        elem = document.getElementById('weight-' + new_id + '');
        elem.remove();
        new_id -= 1;
    }
}

function calculateAndDisplayRoute(directionsService,
directionsDisplay) {
    var myroute;
    var i;
    var str;

    if(document.getElementById("switch-type").checked) {
        str = 'DRIVING';
    }
    else {
        str = 'WALKING';
    }

    if(new_id==1) {
        if (document.getElementById('start').value == '')
        {document.getElementById('wait').style.display = 'none'; alert("Для
        построения маршрута должны быть заполнены все адреса!"); return;}
        if (document.getElementById('point-1').value == '')
        {document.getElementById('wait').style.display = 'none'; alert("Для
        построения маршрута должны быть заполнены все адреса!"); return;}
        $('#calculations').append('<div id="1-time" style="display:
        block"></div>');
        $('#calculations').append('<div id="origin_adr"
        style="display: block"></div>');

        directionsService.route({
            origin: document.getElementById('start').value,
            destination: document.getElementById('point-1').value,
            travelMode: str
        }, function (response, status) {
            if (status === 'OK') {
                document.getElementById('success').style.display =
                "block";

                directionsDisplay.setDirections(response);
                console.log(response);
                myroute =
                directionsDisplay.getDirections().routes[0];
                for (var i = 0; i < myroute.legs.length; i++) {
                    var h = myroute.legs[i].duration.value/3600 ^ 0 ;
                    var m = (myroute.legs[i].duration.value-
                    h*3600)/60 ^ 0 ;

                    var s = myroute.legs[i].duration.value-h*3600-
                    m*60 ;

                    if (h == 0){
                        document.getElementById((i+1)+'-
                        time').innerHTML = m + ' мин';
                    }
                    else {
                        document.getElementById((i+1)+'-
                        time').innerHTML = h + ' ч ' + m + ' мин';
                    }
                    document.getElementById('origin_adr').innerHTML =
                    myroute.legs[i].start_address;

```

```

    }
    } else {
        window.alert('Directions request failed due to ' +
status);
    }
    });
} else {
    var adr_arr = new Array();
    var waypoints = new Array();

    if (document.getElementById('start').value == '')
{document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;} else
{adr_arr[0] = document.getElementById("start").value;}

    if(document.getElementById('opt').value == 'opt_dist')
    {
        for(i=1; i<=new_id; i++)
        {
            if (document.getElementById('point-'+ (i)).value ==
'') {document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;}
            if (document.getElementById((i) + '-adr') != null) {
                adr_arr[i] = document.getElementById((i) + '-
adr').innerHTML;
            }
        }
    }
    else if(document.getElementById('opt').value == 'opt_weight')
    {
        for(i=1; i<=new_id; i++) {
            if (document.getElementById('point-'+ (i)).value ==
'') {document.getElementById('wait').style.display = 'none'; alert("Для
построения маршрута должны быть заполнены все адреса!"); return;}
            if (document.getElementById('weight-'+ (i)).value ==
'') {document.getElementById('wait').style.display = 'none'; alert("Не указан
вес груза!"); return;}
        }

        if (document.getElementById('wl').value == '')
{document.getElementById('wait').style.display = 'none'; alert("Необходимо
указать грузоподъемность!"); return;}

        var wl_sum = 0;
        for(i=1; i<=new_id; i++) {
            wl_sum += parseInt(document.getElementById('weight-'+
(i)).value);
        }
        if (wl_sum > document.getElementById('wl').value)
{document.getElementById('wait').style.display = 'none'; alert("Суммарный вес
грузов превышает параметр грузоподъемности!"); return;}

        var arr = [];
        for(i=1; i<=new_id; i++) {
            arr[document.getElementById('point-'+ (i)).value] =
document.getElementById('weight-'+ (i)).value;
        }
        console.log('wl_sum = ' + wl_sum);
        console.log('wl = ' +
document.getElementById('wl').value);
        console.log('начальный массив', arr);

        var tmp = [];

```

```

    for (var variable in arr) {
        tmp.push({
            address: variable,
            weight: arr[variable]
        });
    }

    //сортировка по убыванию
    var sorted = tmp.sort( function(one, two) {
        return two.weight - one.weight
    });

    // ИТОГОВЫЙ ВЫВОД
    console.log('вывод после сортировки', sorted);

    for(i=1; i<=new_id; i++) {
        adr_arr[i] = sorted[i-1].address;
    }

    console.log(adr_arr);

    for(i=1; i<new_id; i++)
    {
        waypoints.push({location:adr_arr[i]});
    }

    for(i=1; i<=new_id; i++){
        $('#calculations').append('<div id="'+i+'-time"
style="display: block"></div>');
    }
    $('#calculations').append('<div id="origin_adr"
style="display: block"></div>');

    directionsService.route({
        origin: adr_arr[0],
        destination: adr_arr[new_id],
        waypoints: waypoints,
        travelMode: str
    }, function (response, status) {
        if (status === 'OK') {
            document.getElementById('success').style.display =
"block";

            directionsDisplay.setDirections(response);
            console.log(response);
            document.getElementById('origin_adr').innerHTML =
directionsDisplay.getDirections().routes[0].legs[0].start_address;
            myroute =
directionsDisplay.getDirections().routes[0];
            for (var i = 0; i < myroute.legs.length; i++) {
                var h = myroute.legs[i].duration.value/3600 ^ 0 ;
                var m = (myroute.legs[i].duration.value-
h*3600)/60 ^ 0 ;

                var s = myroute.legs[i].duration.value-h*3600-
m*60 ;

                if (h == 0){
                    document.getElementById((i+1)+'-
time').innerHTML = m +' мин';
                }
                else {
                    document.getElementById((i+1)+'-
time').innerHTML = h +' ч ' + m +' мин';

```

```

    }

    if(document.getElementById('opt').value ==
'opt_weight'){
        document.getElementById((i+1)+'-
adr').innerHTML =
directionsDisplay.getDirections().routes[0].legs[i].end_address;
        document.getElementById(i+1).innerHTML =
(directionsDisplay.getDirections().routes[0].legs[i].distance.value)/1000;
    }
} else {
    window.alert('Directions request failed due to ' +
status);
}
});
}

function calc_order(ind){
    var service = new google.maps.DistanceMatrixService();
    var distance;
    var i = 0, k = 0;
    var equal = 0;
    var str = ind;
    var mode;

    if(document.getElementById("switch-type").checked){
        mode = 'DRIVING';
    }
    else {
        mode = 'WALKING';
    }

    $('#calculations').append('<div id="'+ ind +'" style="display:
none"></div>');
    $('#calculations').append('<div id="'+ ind +'-adr"
style="display: none"></div>');
    console.log('ind = ' + ind);
    for(i=1; i<=new_id; i++) {
        service.getDistanceMatrix({
            origins: [document.getElementById((ind-1) +'-
adr').innerHTML],
            destinations: [document.getElementById('point-'+
i).value],
            travelMode: mode
        }, function (response, status) {
            if (status != google.maps.DistanceMatrixStatus.OK) {
                console.log('DistanceMatrix request failed due to ' +
status);
            } else {
                distance =
response.rows[0].elements[0].distance.value / 1000;
                console.log('distance = ' + distance);
                console.log(response.destinationAddresses[0]);

                if ($('#'+ str).html() == "" && distance != 0 &&
(response.originAddresses[0] != response.destinationAddresses[0])) {
                    equal = 0;
                    for(k=1; k<str; k++)
                    {
                        if(response.destinationAddresses[0] == $('#'+
k +'-adr').html()) {equal = 1}

```

```

    }

    if(equal == 0) {$('#'+ str).html(distance);$('#'+
str + '-adr').html(response.destinationAddresses[0]);}
    }
    else {
        if(distance<$('#'+ str).html() && distance != 0
&& (response.originAddresses[0] != response.destinationAddresses[0])) {
            equal = 0;
            for(k=1; k<str; k++)
            {
                if(response.destinationAddresses[0] ==
$('#'+ k + '-adr').html()) {equal = 1}
            }

            if(equal == 0) {$('#'+
str).html(distance);$('#'+ str + '-
adr').html(response.destinationAddresses[0]);}
        }
    }
    });
}

function onMouseover() {
    document.getElementById("calculations").innerHTML = '';
    console.log('-----');

    document.getElementById('wait').style.display = "block";

    var service = new google.maps.DistanceMatrixService();

    var distance;
    var empty = 0;
    var i = 0, j = 0;
    var timeout = 600;
    var mode;

    if(document.getElementById("switch-type").checked) {
        mode = 'DRIVING';
    }
    else {
        mode = 'WALKING';
    }

    if(document.getElementById('start').value == '') {empty=1;}
    for(j=1; j<=new_id; j++)
    {
        if (document.getElementById('point-'+ (j)).value == '')
        {empty=1;}
    }
    if(empty==0) {
        $('#calculations').append('<div id="1" style="display:
none"></div>');
        $('#calculations').append('<div id="1-adr" style="display:
none"></div>');
        for(i=1; i<=new_id; i++) {
            service.getDistanceMatrix({
                origins: [document.getElementById('start').value],
                destinations: [document.getElementById('point-'+
i).value],
                travelMode: mode

```

```

    }, function (response, status) {
        if (status != google.maps.DistanceMatrixStatus.OK) {
            console.log('DistanceMatrix request failed due to
' + status);
        } else {
            distance =
response.rows[0].elements[0].distance.value / 1000;
            console.log('distance = ' + distance);
            console.log(response.destinationAddresses[0]);

            if ($('#1').html() == "") {
                $('#1').html(distance);
                $('#1-
adr').html(response.destinationAddresses[0]);
            }
            else {
                if (distance < $('#1').html())
                { $('#1').html(distance); $('#1-adr').html(response.destinationAddresses[0]); }
            }
        }
    });
}

if(new_id >= 2)
{
    for(i=2; i<=new_id; i++) {
        setTimeout(calc_order, timeout, i);
        timeout += 600;
    }
    setTimeout(function
() { document.getElementById('wait').style.display = 'none'; }, (new_id+1)*600);
    } else { console.log("Для промежуточных расчетов должны быть
заполнены все адреса!"); }
}
</script>

</head>
<body onload="OnLoad()" >

<input type="checkbox" id="nav-toggle" hidden>

<nav id="nav" class="nav" data-html2canvas-ignore>
    <label for="nav-toggle" class="nav-toggle" onclick></label>
    <h2 class="logo">
        <a style="font-family: monospace; font-size: 25px; color:
whitesmoke;">Delivery helper</a>
    </h2>
    <h2 id="log" align="center" style="font-size: 20px; color:
whitesmoke;"></h2>
    <h2 id="email" align="center" style="font-size: 15px; color: whitesmoke;
margin-top: -10px"></h2>

    <label style="color:whitesmoke; font-family: monospace; font-weight:
500">Пункт отправления:</label>
    <div class="input-group" >
        <input type="text" list="startlist" class="form-control"
placeholder="Адрес отправления" id="start">
        <datalist id="startlist">
        </datalist>
        <div class="input-group-btn">
            <button id="geo" class="btn btn-default"
onclick="getPosition()" ><i class="glyphicon glyphicon-globe"></i></button>

```

```

    </div>
</div>

<div style="float: left"><label for="opt" style="color:whitesmoke;
margin-top: 2em; font-family: monospace; font-weight: 500">Пункт (-ы)
назначения:</label></div>
<div style="display: inline-block">
    <div class="btn-group" style="margin-top: 1em; margin-left: 19px">
        <button class="btn btn-default" onclick=add_input()><i
class="glyphicon glyphicon-plus"></i></button>
        <button class="btn btn-default" onclick=delete_input()><i
class="glyphicon glyphicon-minus"></i></button>
    </div>
</div>
<div class="inputs" style="margin-top: 0.5em" align="center">
    <input type="text" list="point-1-list" class="form-control"
style="width: 205px; margin-top: 0px; float: left" placeholder="Адрес
назначения" id="point-1">
    <datalist id="point-1-list">
</datalist>
    <input disabled type="number" min="0" max="999" class="form-control"
style="width: 50px; margin-top: 0px; margin-left: 5px; display: inline-block"
placeholder="Вес" id="weight-1">
</div>
<div style="margin-top: 1em; float: left"><button class="btn"
id="calculate">Построить маршрут</button></div>

<a class="text" style="display: inline-block; margin-left: 15px; margin-
top: 22px" href="#hidden1" onclick="view('hidden1'); return
false">Параметры</a>
<div id="hidden1" style="margin-top: 1em; background: whitesmoke; border-
radius: 5px; padding: 5px; display: none">

    <div align="center" style="font-family: monospace">Способ
передвижения:</div>
    <div align="center">
        <div id="walk" style="display: inline-block; font-family:
monospace; padding: 0 5px 0 5px">Пешком</div>
        <label class="switch" style="display: inline-block; margin-left:
1em; margin-right: 1em">
            <input id="switch-type" type="checkbox" checked>
            <span class="slider round"></span>
        </label>
        <div id="auto" style="display: inline-block; font-family:
monospace; color: whitesmoke; background: #16a085; padding: 0 5px 0 5px;
border-radius: 5px">Автомобиль</div>
    </div>

<script>
    document.getElementById("switch-type").addEventListener("change",
function() {
        if(document.getElementById("switch-type").checked) {
            document.getElementById("auto").style.color =
"whitesmoke";
            document.getElementById("auto").style.background =
"#16a085";
            document.getElementById("auto").style.borderRadius =
"5px";
            document.getElementById("walk").style.color = "#333";
            document.getElementById("walk").style.background =
"none";
            document.getElementById("walk").style.borderRadius =
"none";

```

```

        document.getElementById("price").disabled = false;
        document.getElementById("consumption").disabled = false;
    }
    else{
        document.getElementById("walk").style.color =
"whitesmoke";
        document.getElementById("walk").style.background =
"#16a085";
        document.getElementById("walk").style.borderRadius =
"5px";
        document.getElementById("auto").style.color = "#333";
        document.getElementById("auto").style.background =
"none";
        document.getElementById("auto").style.borderRadius =
"none";
        document.getElementById("price").disabled = true;
        document.getElementById("consumption").disabled = true;
    }
    });
</script>

<div style="margin-top: 1em; font-family: monospace; font-size: 15px"
align="center">Грузоподъемность:
    <input disabled type="number" min="50" max="3500" class="form-
control" style="display: inline-block; width: 57px" value=""
placeholder="Кг." id="wl">
</div>

<span style="float: left; margin-left: 10px; margin-top: 0.5em; font-
family: monospace; font-size: 15px">50</span>
<div style="float: left; margin-left: 10px; margin-top: 0.5em">
    <input style="width: 160px;" id="wheightlift" type="range"
class="range-style" min="50" max="3500" step="10" value="250" disabled/>
</div>
<span style="float: left; margin-left: 10px; margin-top: 0.5em; font-
family: monospace; font-size: 15px">3500</span>

<script>
    document.getElementById("wheightlift").oninput = function() {
        document.getElementById("wl").value = this.value.toString();
    };
</script>

<div>
    <div align="center"><label for="opt" style="margin-top: 1em;
font-family: monospace; font-weight: 500">Оптимизация маршрута:</label></div>
    <select style="text-align-last: center" class="form-control"
id="opt">
        <option value="opt_dist">С учетом расстояния</option>
        <option value="opt_weight">С учетом веса груза</option>
    </select>
</div>

<script>
    document.getElementById("opt").addEventListener("change",
function() {
        if(document.getElementById('opt').value == 'opt_dist') {
            document.getElementById("wheightlift").disabled = true;
            document.getElementById("wl").disabled = true;
            for(i=1; i<=new_id; i++){
                document.getElementById('weight-'+ (i)).disabled =
true;
            }
        }
    });

```

```

    }
    else if(document.getElementById('opt').value ==
'opt_weight'){
        document.getElementById("wheightlift").disabled = false;
        document.getElementById("wl").disabled = false;
        for(i=1; i<=new_id; i++){
            document.getElementById('weight-'+ (i)).disabled =
false;
        }
    }
});
</script>

<div style="margin-top: 1em; padding: 5px">
    <div style="font-family: monospace; font-size: 14px; float:
left">Цена топлива (□/литр)</div>
    <input id="price" type="number" min="1" max="99" class="form-
control" style="width: 61px; margin-top: -7px; margin-left: auto"
placeholder="□">
</div>

<div style="padding: 5px">
    <div style="font-family: monospace; font-size: 14px; float:
left">Расход топлива (л/100км)</div>
    <input id="consumption" type="number" min="1" max="99"
class="form-control" style="width: 61px; margin-top: -7px; margin-left: auto"
placeholder="л">
</div>
</div>

<div id="calculations" style="margin-top: 1em; display: none; color:
whitesmoke"></div>

<script>
function view(n) {
    style = document.getElementById(n).style;
    style.display = (style.display == 'block') ? 'none' : 'block';
}
</script>

<div id="dist_dur" align="left" style="color: whitesmoke; margin-top:
1em; background: #3bc1a0; border-radius: 5px; padding: 5px 10px 5px 10px;
display: none">
    <div>
        <div style="font-family: monospace; font-size: 15px; float:
left">Расстояние</div>
        <div id="total" style="font-family: monospace; font-size: 15px"
align="right"></div>
</div>
<div>
        <div style="font-family: monospace; font-size: 15px; float:
left">Время</div>
        <div id="duration" style="font-family: monospace; font-size:
15px" align="right"></div>
</div>

<div id="petrol_label" style="display: none">
    <div align="center" style="display: block">Расход топлива</div>

    <div>
        <div style="font-family: monospace; font-size: 15px; float:
left">Объем</div>
        <div id="amount" style="font-family: monospace; font-size:

```

```

15px" align="right"></div>
    </div>
    <div>
        <div style="font-family: monospace; font-size: 15px; float:
left">Стоимость</div>
        <div id="total_price" style="font-family: monospace; font-
size: 15px" align="right"></div>
    </div>
</div>

<div id="wait" style="margin-top: 1em; color: whitesmoke; display: none">
    <div class="windows8">
        <div class="wBall" id="wBall_1">
            <div class="wInnerBall"></div>
        </div>
        <div class="wBall" id="wBall_2">
            <div class="wInnerBall"></div>
        </div>
        <div class="wBall" id="wBall_3">
            <div class="wInnerBall"></div>
        </div>
        <div class="wBall" id="wBall_4">
            <div class="wInnerBall"></div>
        </div>
        <div class="wBall" id="wBall_5">
            <div class="wInnerBall"></div>
        </div>
    </div>
</div>

<div id="success" style="display: none">
    <a class="text" style="display: block; margin-top: 10px; font-family:
monospace;" href="#hidden2" onclick="view('hidden2'); return false">Отправить
по эл.почте</a>

    <div id="hidden2" style="margin-top: 0px; border-radius: 5px;
padding: 5px 0 5px 0; display: none">
        <input type="text" class="form-control" style="width: 155px;
margin-top: 0px; float: left" placeholder="Укажите e-mail" id="reciever-
email">
        <div class="text" style="margin-top: 7px; margin-left: 20px;
display: inline-block; cursor: pointer" onclick="SendMail()">Отправить</div>
    </div>

    <label style="color:whitesmoke; margin-top: 5px; font-family:
monospace; font-weight: 400">Подробный обзор маршрута:</label>
    <div style="background: whitesmoke; border-radius: 5px; margin-top:
0em; padding: 5px">
        <div id="panel" style="" align="center"></div>
    </div>
</div>
</nav>

<div class="mask-content"></div>

<div style="margin-left: 0%">
    <header>
        <h2 align="center" style="margin-left: 40px; margin-top: 10px; font-
family: monospace; font-size: 25px; color: whitesmoke; float: left">Delivery
Helper</h2>
        <div style="margin-right: 1em" align="right">
            <a style="margin-top: 10px; margin-right: 1em" href="#iw-

```

```

modalreg" class="iw-modal-btn"><div class="glyphicon glyphicon-cog"
style="margin-right: 5px"></div>Настройки</a>
    <a style="margin-top: 10px" href="index.html" class="iw-modal-
btn"><div class="glyphicon glyphicon-log-out" style="margin-right:
5px"></div>Выход</a>
    </div>
</header>

<div id="mapBox">
    <div id="gmap_canvas"></div>
</div>

<script>
    document.getElementById("weightlift").addEventListener("change",
function() {
        document.getElementById("wl").value = this.value;
    });

    document.getElementById("wl").addEventListener("change", function() {
        document.getElementById("weightlift").value = this.value;
    });

    var visota = $(window).height();
    visota=visota-50;
    $( '#gmap_canvas' ).css('height', visota);

    $(window).on('resize', function(){
        var visota = $(window).height();
        visota=visota-50;
        $( '#gmap_canvas' ).css('height', visota);
    });

    var geocoder;
    var map;

    function initMap() {
        var directionsService = new google.maps.DirectionsService;
        var directionsDisplay = new
google.maps.DirectionsRenderer({draggable: true});
        geocoder = new google.maps.Geocoder();
        var coordinates = {lat: 53.200, lng: 45.004};
        map = new google.maps.Map(document.getElementById('gmap_canvas'),
{
            zoom: 14,
            center: coordinates,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        });
        directionsDisplay.setMap(map);
        directionsDisplay.setPanel(document.getElementById('panel'));

        var onChangeHandler = function(){
            var timeout = 600*(new_id+1);
            setTimeout(calculateAndDisplayRoute, timeout,
directionsService, directionsDisplay);

            setTimeout( function(){
                html2canvas(document.body,{
                    imageTimeout: 0,
                    allowTaint: true,
                    useCORS: true,
                    x: 50,
                    y: 100,
                    width: $( '#gmap_canvas' ).width() - 100,

```

```

        height:$('#gmap_canvas').height() - 100
    }).then (function (canvas) {
        console.log(canvas);

        if(document.getElementById('map-screen')){
            document.getElementById('map-screen').remove();
            console.log("Удалился map-screen");
        }

        canvas.id = "map-screen";
        canvas.style.display = "none";

document.getElementById("mapBox").appendChild(canvas);
    });
    }, timeout+1500);

    setTimeout(convertCanvasToImage, timeout+4500);
};

document.getElementById('calculate').addEventListener('click',
onMouseover);
document.getElementById('calculate').addEventListener('click',
onChangeHandler);

directionsDisplay.addListener('directions_changed', function() {
    computeTotalDistance(directionsDisplay.getDirections());
});
}

function computeTotalDistance(result) {
    var total = 0;
    var duration = 0;
    var myroute = result.routes[0];
    for (var i = 0; i < myroute.legs.length; i++) {
        total += myroute.legs[i].distance.value;
        duration += myroute.legs[i].duration.value;
    }

    document.getElementById('total').innerHTML =
(total/1000).toFixed(1) + ' км';

    var h = duration/3600 ^ 0 ;
    var m = (duration-h*3600)/60 ^ 0 ;
    var s = duration-h*3600-m*60 ;

    console.log('h = ' + h);
    console.log('m = ' + m);
    console.log('s = ' + s);

    if (h == 0) {
        //console.log((m<10?"0"+m:m)+" мин. "+(s<10?"0"+s:s)+"
сек.");

        console.log(m+" мин. "+ s + " сек.");
        document.getElementById('duration').innerHTML = m + ' мин';
    }
    else {
        console.log(h+" ч. "+ m + " мин. "+ s + " сек.");
        document.getElementById('duration').innerHTML = h + ' ч ' + m
+ ' мин';
    }

    if(document.getElementById("switch-type").checked &&
document.getElementById("price").value != '' &&

```

```

document.getElementById("consumption").value != ''){
    var dst = (total/1000).toFixed(1);
    var amount =
(document.getElementById("consumption").value/100 * dst).toFixed(2);
    console.log("amount = "+ amount);
    var price = (document.getElementById("price").value *
amount).toFixed(2);
    console.log("total petrol price = "+ price);

    document.getElementById('petrol_label').style.display =
"block";

    document.getElementById('amount').innerHTML = amount + ' л';
    document.getElementById('total_price').innerHTML = price + '
';
}
else
{
    document.getElementById('petrol_label').style.display =
"none";
}

document.getElementById('dist_dur').style.display = "block";

setTimeout( function() {
    html2canvas(document.body, {
        imageTimeout: 0,
        allowTaint: true,
        useCORS: true,
        x: 50,
        y: 100,
        width: $('#gmap_canvas').width() - 100,
        height: $('#gmap_canvas').height() - 100
    }).then( function (canvas) {
        console.log(canvas);

        if(document.getElementById('map-screen')) {
            document.getElementById('map-screen').remove();
            console.log("Удалился map-screen");
        }

        canvas.id = "map-screen";
        canvas.style.display = "none";
        document.getElementById("mapBox").appendChild(canvas);
    });
}, 1500);

setTimeout(convertCanvasToImage, 5500);
}

function getPosition() {
    var geocoder = new google.maps.Geocoder();
    var latlng;

    navigator.geolocation.getCurrentPosition(function (position) {
        latlng = new google.maps.LatLng(position.coords.latitude,
position.coords.longitude);
        geocoder.geocode({ 'latLng': latlng }, function(results,
status) {
            $('#start').val(results[0].formatted_address);
        });
    });
}

```

```

// инициализация
google.maps.event.addDomListener(window, 'load', initMap);

var inner1=0, inner2=0, inner3=0, inner4=0, inner5=0;
function change_addr1() {
    if($("#home").length) {return;}

    inner1 = document.getElementById('haddr').innerText;
    var value = document.getElementById('haddr').innerHTML;

    if (document.getElementById('haddr').innerText == "Адрес не
задан") {value = "";}

    var str = '' +
        '<div style="float: left"><input type="text" required
class="form-control" style="width: 180px" id="home" placeholder="Введите
адрес" value="'+ value +' "></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="addrhome()"><i class="glyphicon glyphicon-
ok"></i></button></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="del_addrhome()"><i class="glyphicon
glyphicon-remove"></i></button></div>' +
        '';
    document.getElementById('haddr').innerHTML = str;
}
function change_addr2() {
    if($("#work").length) {return;}

    inner2 = document.getElementById('waddr').innerText;
    var value = document.getElementById('waddr').innerHTML;

    if (document.getElementById('waddr').innerText == "Адрес не
задан") {value = "";}

    var str = '' +
        '<div style="float: left"><input type="text" required
class="form-control" style="width: 180px" id="work" placeholder="Введите
адрес" value="'+ value +' "></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="addrwork()"><i class="glyphicon glyphicon-
ok"></i></button></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="del_addrwork()"><i class="glyphicon
glyphicon-remove"></i></button></div>' +
        '';
    document.getElementById('waddr').innerHTML = str;
}
function change_addr3() {
    if($("#f11").length) {return;}

    inner3 = document.getElementById('favlocladdr').innerText;
    var value = document.getElementById('favlocladdr').innerHTML;

    if (document.getElementById('favlocladdr').innerText == "Адрес не
задан") {value = "";}

    var str = '' +
        '<div style="float: left"><input type="text" required
class="form-control" style="width: 180px" id="f11" placeholder="Введите
адрес" value="'+ value +' "></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="addrf11()"><i class="glyphicon glyphicon-

```

```

ok"></i></button></div>' +
    '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="del_addrfl1()"><i class="glyphicon
glyphicon-remove"></i></button></div>' +
    '';
    document.getElementById('favloc1addr').innerHTML = str;
}
function change_addr4() {
    if($("#fl2").length) {return;}

    inner4 = document.getElementById('favloc2addr').innerText;
    var value = document.getElementById('favloc2addr').innerHTML;

    if (document.getElementById('favloc2addr').innerText == "Адрес не
задан") {value = "";}

    var str = '' +
        '<div style="float: left"><input type="text" required
class="form-control" style="width: 180px" id="fl2" placeholder="Введите
адрес" value="'+ value +' "></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="addrfl2()"><i class="glyphicon glyphicon-
ok"></i></button></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="del_addrfl2()"><i class="glyphicon
glyphicon-remove"></i></button></div>' +
        '';
    document.getElementById('favloc2addr').innerHTML = str;
}
function change_addr5() {
    if($("#fl3").length) {return;}

    inner5 = document.getElementById('favloc3addr').innerText;
    var value = document.getElementById('favloc3addr').innerHTML;

    if (document.getElementById('favloc3addr').innerText == "Адрес не
задан") {value = "";}

    var str = '' +
        '<div style="float: left"><input type="text" required
class="form-control" style="width: 180px" id="fl3" placeholder="Введите
адрес" value="'+ value +' "></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="addrfl3()"><i class="glyphicon glyphicon-
ok"></i></button></div>' +
        '<div style="display: inline-block; margin-left: 5px"><button
class="btn btn-default" onclick="del_addrfl3()"><i class="glyphicon
glyphicon-remove"></i></button></div>' +
        '';
    document.getElementById('favloc3addr').innerHTML = str;
}

jQuery(function($){
    $(document).mouseup(function (e){ // событие клика по веб-
документу

        if(inner1!=0) {
            var div = $("#haddr"); // тут указываем ID элемента
            if (!div.is(e.target) // если клик был не по нашему блоку
                && div.has(e.target).length === 0) { // и не по его
дочерним элементам

                document.getElementById('haddr').innerHTML = inner1;
            }
        }
    })
}

```

```

    });
  });
  jQuery(function($) {
    $(document).mouseup(function (e) { // событие клика по веб-
документу
      if(inner2!=0) {
        var div = $("#waddr"); // тут указываем ID элемента
        if (!div.is(e.target) // если клик был не по нашему блоку
        && div.has(e.target).length === 0) { // и не по его
дочерним элементам
          document.getElementById('waddr').innerHTML = inner2;
        }
      }
    });
  });
  jQuery(function($) {
    $(document).mouseup(function (e) { // событие клика по веб-
документу
      if(inner3!=0) {
        var div = $("#favloc1addr"); // тут указываем ID элемента
        if (!div.is(e.target) // если клик был не по нашему блоку
        && div.has(e.target).length === 0) { // и не по его
дочерним элементам
          document.getElementById('favloc1addr').innerHTML =
inner3;
        }
      }
    });
  });
  jQuery(function($) {
    $(document).mouseup(function (e) { // событие клика по веб-
документу
      if(inner4!=0) {
        var div = $("#favloc2addr"); // тут указываем ID элемента
        if (!div.is(e.target) // если клик был не по нашему блоку
        && div.has(e.target).length === 0) { // и не по его
дочерним элементам
          document.getElementById('favloc2addr').innerHTML =
inner4;
        }
      }
    });
  });
  jQuery(function($) {
    $(document).mouseup(function (e) { // событие клика по веб-
документу
      if(inner5!=0) {
        var div = $("#favloc3addr"); // тут указываем ID элемента
        if (!div.is(e.target) // если клик был не по нашему блоку
        && div.has(e.target).length === 0) { // и не по его
дочерним элементам
          document.getElementById('favloc3addr').innerHTML =
inner5;
        }
      }
    });
  });

  function addrhome() {
    var homeaddr = document.getElementById('home').value;
    var login = document.getElementById('log').innerHTML;

    $('#home').css("background", "white");

```

```

    if(homeaddr!=''){
        $.ajax({
            url: "NewHAddress",
            type: "GET",
            dataType: "text",
            data: ("haddr=" + homeaddr + "&log=" + login),
            success: function (data) {
                if (data[0] == "y") {
                    window.location.reload();
                }
            },
            error: function (e) {
                alert("error");
            }
        })
    }
    else
    {
        alert("Поле ввода адреса не заполнено!");
    }
}

function del_addrhome() {
    var homeaddr = "Адрес не задан";
    var login = document.getElementById('log').innerHTML;

    $('#home').css("background", "white");

    $.ajax({
        url: "NewHAddress",
        type: "GET",
        dataType: "text",
        data: ("haddr=" + homeaddr + "&log=" + login),
        success: function (data) {
            if (data[0] == "y") {
                window.location.reload();
            }
        },
        error: function (e) {
            alert("error");
        }
    })
}

function addrwork() {
    var workaddr = document.getElementById('work').value;
    var login = document.getElementById('log').innerHTML;

    $('#work').css("background", "white");

    if(workaddr!=''){
        $.ajax({
            url: "NewWAddress",
            type: "GET",
            dataType: "text",
            data: ("waddr=" + workaddr + "&log=" + login),
            success: function (data) {
                if (data[0] == "y") {
                    window.location.reload();
                }
            },
            error: function (e) {
                alert("error");
            }
        })
    }
}

```

```

    })
  }
  else
  {
    alert("Поле ввода адреса не заполнено!");
  }
}
function del_addrwork() {
  var workaddr = "Адрес не задан";
  var login = document.getElementById('log').innerHTML;

  $('#work').css("background", "white");

  $.ajax({
    url: "NewWAddress",
    type: "GET",
    dataType: "text",
    data: ("waddr=" + workaddr + "&log=" + login),
    success: function (data) {
      if (data[0] == "y") {
        window.location.reload();
      }
    },
    error: function (e) {
      alert("error");
    }
  })
}
function addrfl1() {
  var favlocladdr = document.getElementById('fl1').value;
  var login = document.getElementById('log').innerHTML;

  $('#fl1').css("background", "white");

  if(favlocladdr!=''){
    $.ajax({
      url: "NewFavLocOne",
      type: "GET",
      dataType: "text",
      data: ("floneaddr=" + favlocladdr + "&log=" + login),
      success: function (data) {
        if (data[0] == "y") {
          window.location.reload();
        }
      },
      error: function (e) {
        alert("error");
      }
    })
  }
  else
  {
    alert("Поле ввода адреса не заполнено!");
  }
}
function del_addrfl1() {
  var favlocladdr = "Адрес не задан";
  var login = document.getElementById('log').innerHTML;

  $('#fl1').css("background", "white");

  $.ajax({
    url: "NewFavLocOne",

```

```

        type: "GET",
        dataType: "text",
        data: ("floneaddr=" + favloc1addr + "&log=" + login),
        success: function (data) {
            if (data[0] == "y") {
                window.location.reload();
            }
        },
        error: function (e) {
            alert("error");
        }
    })
}

function addrfl2() {
    var favloc2addr = document.getElementById('fl2').value;
    var login = document.getElementById('log').innerHTML;

    $('#fl2').css("background", "white");

    if(favloc2addr!=''){
        $.ajax({
            url: "NewFavLocTwo",
            type: "GET",
            dataType: "text",
            data: ("fltwoaddr=" + favloc2addr + "&log=" + login),
            success: function (data) {
                if (data[0] == "y") {
                    window.location.reload();
                }
            },
            error: function (e) {
                alert("error");
            }
        })
    }
    else
    {
        alert("Поле ввода адреса не заполнено!");
    }
}

function del_addrfl2() {
    var favloc2addr = "Адрес не задан";
    var login = document.getElementById('log').innerHTML;

    $('#fl2').css("background", "white");

    $.ajax({
        url: "NewFavLocTwo",
        type: "GET",
        dataType: "text",
        data: ("fltwoaddr=" + favloc2addr + "&log=" + login),
        success: function (data) {
            if (data[0] == "y") {
                window.location.reload();
            }
        },
        error: function (e) {
            alert("error");
        }
    })
}

function addrfl3() {
    var favloc3addr = document.getElementById('fl3').value;

```

```

var login = document.getElementById('log').innerHTML;

$('#fl3').css("background", "white");

if(favloc3addr!=''){
    $.ajax({
        url: "NewFavLocThree",
        type: "GET",
        dataType: "text",
        data: ("flthreeaddr=" + favloc3addr + "&log=" + login),
        success: function (data) {
            if (data[0] == "y") {
                window.location.reload();
            }
        },
        error: function (e) {
            alert("error");
        }
    })
}
else
{
    alert("Поле ввода адреса не заполнено!");
}
}

function del_addrfl3() {
    var favloc3addr = "Адрес не задан";
    var login = document.getElementById('log').innerHTML;

    $('#fl3').css("background", "white");

    $.ajax({
        url: "NewFavLocThree",
        type: "GET",
        dataType: "text",
        data: ("flthreeaddr=" + favloc3addr + "&log=" + login),
        success: function (data) {
            if (data[0] == "y") {
                window.location.reload();
            }
        },
        error: function (e) {
            alert("error");
        }
    })
}

function NewEmail() {
    var newemail = document.getElementById('newemail').value;
    var login = document.getElementById('log').innerHTML;

    if(newemail != ''){
        $.ajax({
            url: "EmailServlet",
            type: "GET",
            dataType: "text",
            data: ("email=" + newemail + "&log=" + login),
            success: function (data) {
                if (data[0] == "y") {
                    $('#email').html('(' + newemail + ')');
                    document.getElementById('reciever-email').value =
newemail;

                    alert("Адрес эл. почты был успешно изменен!");
                }
            }
        })
    }
}

```

```

    },
    error: function (e) {
        alert("error");
    }
})
}
else
{
    alert("Новый адрес эл. почты не указан!");
}
}
</script>

</div>
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC-
2n7pGa6eeVMXequ_pH7ypqplnyx4tGE&libraries=places"></script>

<!--Разметка для модального окна-->
<!--Регистрация-->
<div id="iw-modalreg" class="iw-modal">
    <div class="iw-modal-wrapper">
        <div class="iw-CSS-modal-inner">
            <div class="iw-modal-header">
                <h3 class="iw-modal-title" style="font-family: monospace"
align="center">Настройки</h3>
                <a href="#close" title="Закрыть" class="iw-close"
onclick="">x</a>
            </div>
            <div class="iw-modal-text">
                <div style="float: left; margin-top: 30px; width: 50%">
                    <div align="center">
                        <label style="font-size: 15px">Изменить адрес эл.
почты</label>
                        <div class="input-group" style="width: 30%;margin-
top: 10px" >
                            <span class="input-group-addon"><span
class="glyphicon glyphicon-envelope"></span></span>
                            <input type="email" required class="form-control"
style="width: 200px" placeholder="Введите новый e-mail" id="newemail">
                        </div>
                    </div>
                    <div style="margin-top: 20px; margin-left: 0px"
class="form-group">
                        <div class="col-offset-4 col-4" align="center"
style="margin-left: 0px">
                            <button type="button" class="btn btn-info"
style="width: auto; background: #16a085" onclick="NewEmail();">
                                <div class="glyphicon glyphicon-ok"></div>
                                Сохранить
                            </button>
                        </div>
                    </div>
                </div>
                <div style="margin-left: 50%; border-left: thin solid #ccc">
                    <div align="center"><label style="font-size:
15px">Избранные адреса</label></div>
                    <div style="margin-left: 5px" >

```

```


93


```

```
</body>  
</html>
```

ЛИСТИНГ КОДА БАЗЫ ДАННЫХ

Приложение Б

```
CREATE DATABASE test;
CREATE TABLE users
(
    login text NOT NULL,
    password text NOT NULL,
    home text DEFAULT 'Адрес не задан'::text,
    workaddr text DEFAULT 'Адрес не задан'::text,
    fav_loc1 text DEFAULT 'Адрес не задан'::text,
    fav_loc2 text DEFAULT 'Адрес не задан'::text,
    fav_loc3 text DEFAULT 'Адрес не задан'::text,
    email text
);
```