

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт Информационных систем и геотехнологий
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

БАКАЛАВРСКАЯ РАБОТА

На тему Проектирование ГИС по контролю над транспортом в компании

Оглавление

ВВЕДЕНИЕ	3
ГЛАВА 1. ПРЕДПРОЕКТНЫЙ АНАЛИЗ.....	5
1.1 Анализ предметной области	5
1.2 Анализ аналогов и возможный функционал ИС.....	7
1.3 Анализ объекта исследования.....	16
1.4 Экономическая эффективность	19
1.5 Сроки окупаемости	24
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИС	26
2.1 Концептуальное проектирование	26
2.2 Объектно- ориентированное проектирование.....	27
2.3 Диаграмма вариантов использования (USE-CASE)	31
2.3 Диаграммы последовательности	33
2.4 Диаграмма классов.....	38
2.5 Диаграмма развертывания.....	40
2.6 Диаграмма компонентов.....	43
ГЛАВА 3. РЕАЛИЗАЦИЯ.....	46
3.1 Описание типовых проектных решений.....	47
3.2 Описание реализованной системы	48
3.3 Описание базы данных	54
ЗАКЛЮЧЕНИЕ	56
СПИСОК ЛИТЕРАТУРЫ.....	58
ПРИЛОЖЕНИЕ 1	59

ВВЕДЕНИЕ

Основной характеристикой нынешнего этапа развития общества является его информатизация, цель которой - интегрировать компьютеры и средства связи во все области человеческой деятельности.

Сегодня многие люди осознают ценность и необходимость использования приложения для отслеживания поставок в компаниях и организациях. Раньше большинство компаний размещали свои важные данные на бумажных носителях, но в настоящее время информация хранится на электронных носителях в специализированных для этого программах.

Транспортный план предоставляет информацию, которая позволяет нам планировать стратегии организаций по использованию ресурсов и оценке работы.

Отношения в транспортной логистике изменились, стали более сложными, поэтому нам нужна более важная форма контроля.

Поскольку многие стартапы имеют системы передачи ресурсов, мы должны знать, что не все устоявшиеся решения подходят для использования.

Важной частью этого проекта является создание службы логистики для наземного транспорта, целью которой является деятельность частных компаний.

Основное внимание в проекте уделяется веб-приложению для отслеживания движения транспортных средств внутри предприятия.

Конечная цель этой работы - систематизировать знания для решения практических задач, а также способность работать самостоятельно.

Для достижения цели необходимо решить следующие задачи:

- Произвести теоретический и практический анализ на тему отслеживающих информационных систем.

- Проанализировать исходные данные проекта;
- Исследовать существующий и выбор подходящей платформы для разработки.

- Разработать функционирующее web – приложение на выбранной платформе.

Используемый инструментарий.

Для выполнения практической части выпускной квалификационной работы были использованы следующие инструменты:

- Среда разработки Visual Studio Code;
- Библиотека Bootstrap;
- Инструмент моделирования FlowCharts;
- Использования языков программирования: HTML, CSS, Javascript, PHP;
- База данных MySQL.

ГЛАВА 1. ПРЕДПРОЕКТНЫЙ АНАЛИЗ

1.1 Анализ предметной области

С тех пор, как Карл Бенц установил первый двигатель внутреннего сгорания на грузовик в конце 1800-х годов, грузоперевозки стали неотъемлемой частью транспортного сектора. Раньше грузовики приводились в движение паровыми двигателями.

С непрерывным развитием автомобильной промышленности на противоположной стороне также есть разработки в области технологий транспортной отрасли для грузовых автомобилей.

В отрасли грузоперевозок произошли технологические достижения, сопоставимые с достижениями в автомобильном секторе. Двигатели стали более мощными и эффективными. У водителей грузовиков теперь более комфортные кабины. Кроме того, прицепы усовершенствовались, чтобы дальнобойщики могли перевозить практически любой тип материала.

Развитие грузовых автомобилей и грузоперевозок на протяжении всей истории интригует. Приведены примеры на рисунках 1 и 2.



Рисунок 1 - Первые автомобили для перевозки



Рисунок 2 - Автомобили для перевозок в современном мире

Главные проблемы прежде чем приступить к анализу этого предмета, вы должны сначала определить любые потенциальные проблемы.

Логично, что безопасность на первом месте. Любое приложение, используемое в корпоративных целях, должно соответствовать всем стандартам безопасности. Кража информации, несанкционированное использование личной информации обычных работников и утечка данных с участием водителей - все это довольно распространено.

Требование о заключении договора с собственниками организаций – второй вопрос. Предприниматели очень не любят посетителей, которые предлагают свои услуги; это особенно заметно при инициировании контакта, потому что неназванные разработчики не внушают доверия. Чем больше информации он должен ввести, тем больше доказательств надежности клиент должен видеть. В первые несколько минут после перехода вам не нужно регистрироваться или вводить личную информацию и контакты, потому что пользователь еще не установил уровень доверия и не определил, что сайт заслуживает доверия. Это связано с тем, что пользователь постепенно привыкает к программе.

1.2 Анализ аналогов и возможный функционал ИС

Геоинформационная система (ГИС) - это информационная система, основное предназначение которой сбор, хранение, обработка, отображение и распространение данных, а также получение на их основе новой информации и знаний о пространственно-координированных объектах и явлениях.

Основной особенностью ГИС, отличающей ее от других информационных систем, является то, что все моделируемые в ГИС объекты и явления имеют пространственную привязку, позволяющую анализировать их во взаимосвязи с другими пространственно-определенными объектами. ГИС кардинально отличаются от большинства других информационных систем тем, что вся информация в них очень наглядно представляется в виде электронных карт, позволяя человеку получать новые знания.

В простейшем варианте геоинформационные системы - сочетание обычных баз данных (атрибутивной информации) с электронными картами, то есть мощными графическими средствами. Основная идея ГИС - связь данных на карте и в базе данных. ГИС - это и аналитические средства для работы с любой координатно-привязанной информацией. В принципе, ГИС можно рассматривать как некое расширение концепции баз данных. В этом смысле ГИС фактически представляет собой новый уровень и способ интеграции и структурирования информации.

Географические информационные системы представляют собой компьютерные средства для отображения и анализа данных, которые позволяют интегрировать и анализировать пространственные данные (информацию о местоположениях). Географические данные могут включать таблицы с информацией, изображения или данные о реальных объектах, например, линии для дорог, точки для деревьев и полигоны для

зданий. Преимущество ГИС состоит в том, что вы можете запрашивать все таблицы и слои, затем выполнять анализ всех этих форм данных. ГИС предоставляет мощные инструменты для визуальных и аналитических выводов, которые практически невозможно интерпретировать из простых таблиц данных.

Важность ГИС как интегрирующей технологии также очевидна в ее происхождении. Развитие ГИС основывается на инновациях, сделанных по различным дисциплинам: география, картография, фотограмметрия, дистанционное зондирование, геодезия, статистика, компьютерные науки, искусственный интеллект, демография и многие другие отрасли науки. Географическая информационная система является автоматизированной системой, имеющей большое количество графических и тематических баз данных, соединенная с модельными и расчетными функциями для манипулирования ими и преобразования в пространственную картографическую информацию, на основе которой принимаются разнообразные решения и осуществляется контроль.

ГИС представляет собой организованный набор аппаратного обеспечения, программного обеспечения, географических данных, предназначенный для эффективного ввода, хранения, обновления, обработки, анализа и визуализации всех видов географически привязанной информации и связанных с ней непространственных данных. ГИС рассматривают как многоаспектную автоматизированную интегрированную информационную систему с пространственной локализацией данных. В настоящее время геоинформационными системами называют различные информационные системы, решающие широкий круг задач. В связи с этим существует несколько классификаций, позволяющих более полно понять сущность ГИС:

1. Виды ГИС по пространственному охвату:

- глобальные (планетарные);

- субконтинентальные;
- межнациональные;
- национальные (государственные);
- региональные (областные, краевые, республиканские);
- субрегиональные (районы внутри регионов);
- локальные (городские);
- ультралокальные (отдельные ограниченные территории).

2. Виды ГИС по используемой модели данных:

– векторные ГИС работают с различными моделями данных, а также иногда с триангуляционными моделями поверхностей;

– растровые ГИС позволяют работать только с растровыми моделями данных;

– гибридные совмещают в себе возможности векторных и растровых ГИС.

3. Виды ГИС по компьютерной платформе, на базе которой они функционируют:

– настольные ГИС. К этой категории относятся большинство известных ГИС. Как правило, используемые в них данные сохраняются в локальных файлах. – клиент-серверные ГИС. В этих системах пространственные данные хранятся полностью в базе данных сервера. Этот сервер обычно является высокоуровневой надстройкой над некоторой промышленной системой управления базами данных (СУБД типа Microsoft SQL Server, Oracle, DB2, Sybase и др.).

– ГИС для интернета. Такие системы бывают двух видов:

а) самостоятельные программы, обеспечивающие полные функции HTTP-сервера, либо

б) наборы программных компонент (обычно ActiveX-объектов), которые могут быть интегрированы в существующий Html-код и

произвольным образом настроены. Первый подход позволяет очень быстро выполнить публикацию карт в интернете, а второй подход более гибок.

Выделяют следующие территориальные уровни использования ГИС в России [11, с.167]:

– глобальный уровень - Россия на глобальном и европейском фоне, масштаб 1:45 000 000 - 1:100 000 000;

– всероссийский уровень - вся территория страны, включая прибрежные акватории и приграничные районы, масштаб 1:2 500 000 - 1:20 000 000;

– региональный уровень - крупнейшие природные и экономические регионы, субъекты Федерации, масштаб 1:500 000 - 1:1 000 000(ОСХ):

– локальный уровень - области, районы, национальные парки, ареалы кризисных ситуаций, масштаб 1:50 000 - 1:1 000 000:

– муниципальный уровень - города, городские районы, пригородные зоны, масштаб 1:50 000 и крупнее.

К счастью, технология разработала несколько умных методов для наблюдения за поездами, автобусами и метро. Эти приложения для отслеживания транспортных средств хорошо известны (рис. 4-9).

Citymapper

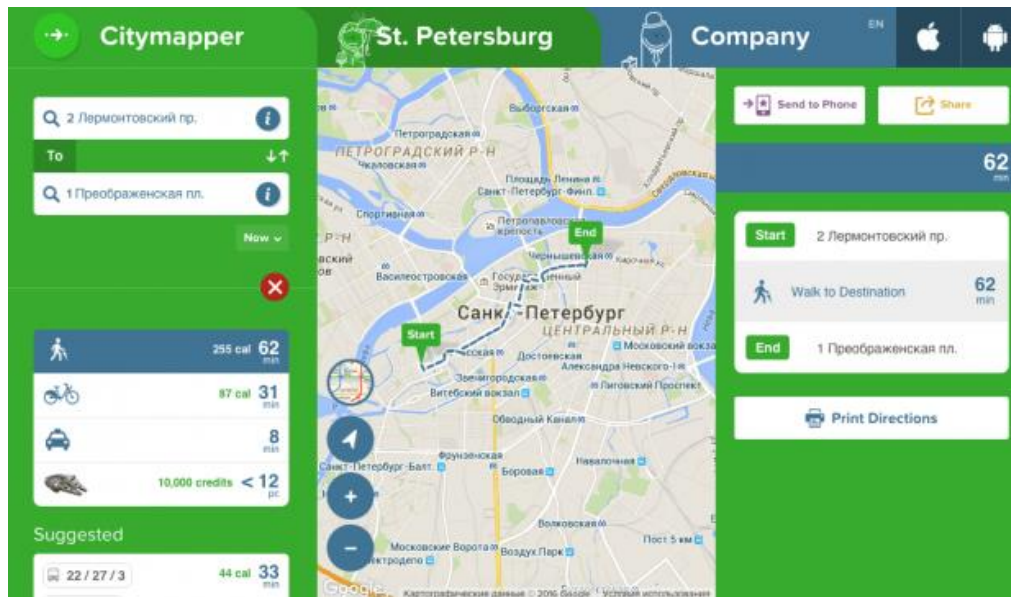


Рисунок 4 - Приложение «Citymapper»

Сильной стороной сопоставимых приложений является доступность и применимость информации. Языки программирования, такие как Java, Kotlin и Javascript, в основном использовались для создания веб-сайтов.

Citymapper предоставляет информацию о доступных такси, Ubers, Lyfts, велосипедных дорожках и даже паромах в дополнение к поездам, автобусам и метро.

К недостаткам приложения можно отнести плохое функционирование, запутанный пользовательский интерфейс, отсутствие всплывающих окон, быструю прокрутку и место для отображения обнаруженных транспортных маршрутов. Сайт создан по известному макету.

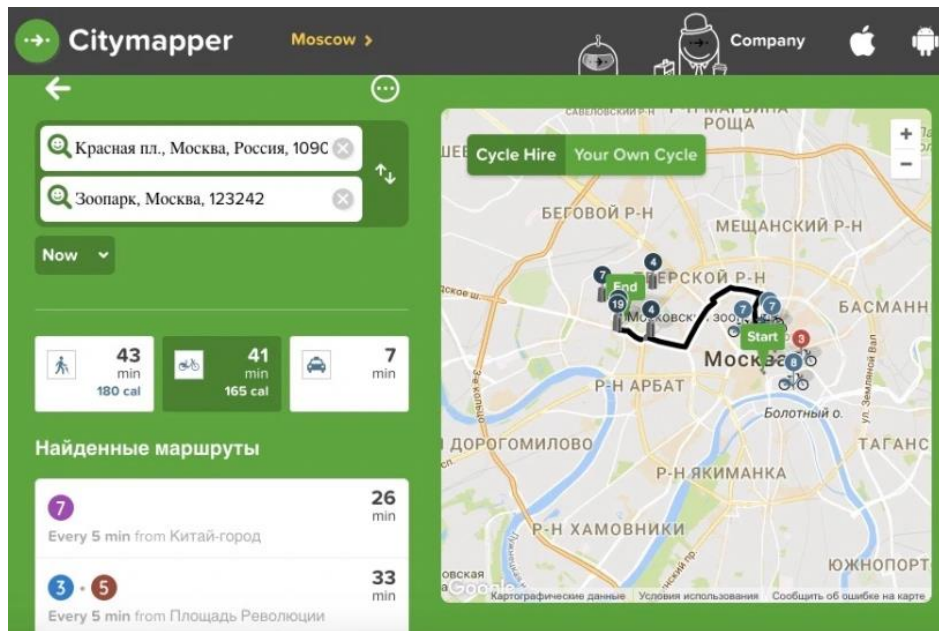


Рисунок 5 - Пример построения маршрута в «Citymapper»

Rome2rio

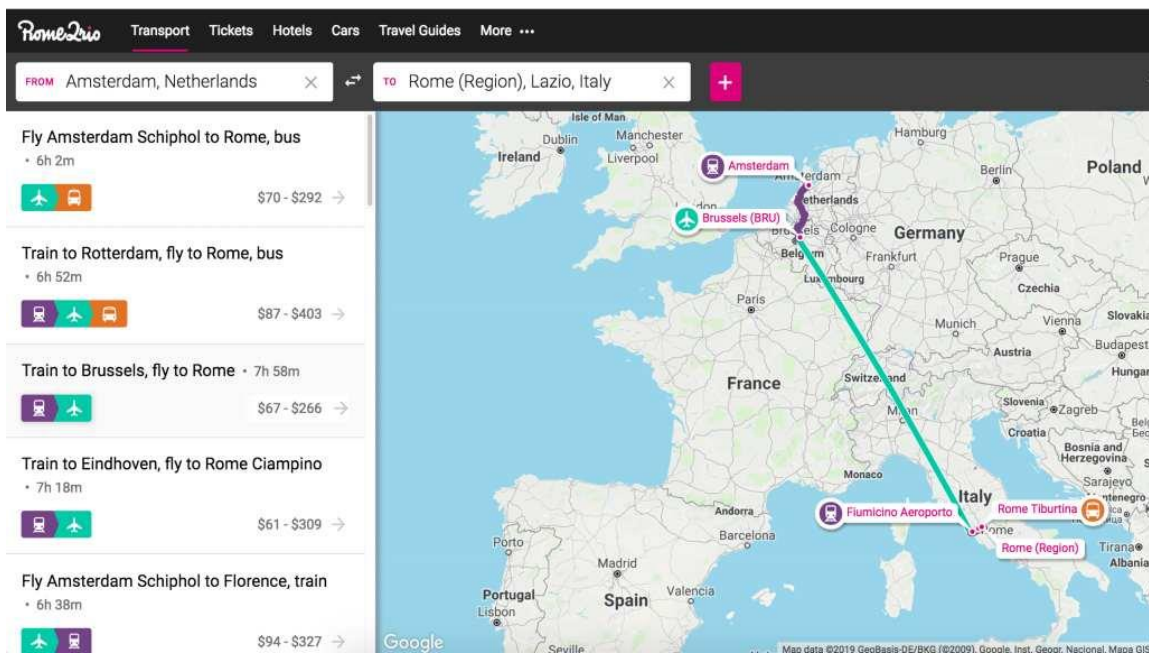


Рисунок 6 - Приложение «Rome2rio»

Сильной стороной приложения «Rome2rio» является обновленный дизайн и простая навигация внутри программы. Эта программа включает в себя не только информацию о транспорте, но и расписание рейсов.

Rome2rio предлагает доступ к графикам и стоимости 5000 различных предприятий в более чем 160 странах.

Проблемы с покупкой билетов на транспорт и шаблонный интерфейс - примеры слабых мест.

Как работает Rome2rio

Rome2rio находит любой город, населенный пункт, ориентир, достопримечательность или адрес на всем земном шаре с тысячами мультимодальных маршрутов, чтобы вы могли легко добраться из пункта А в пункт В.

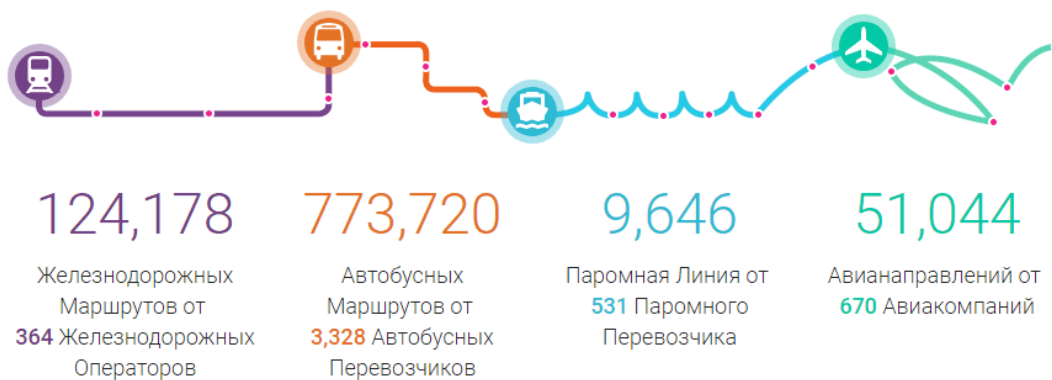


Рисунок 7 - Принцип работы приложения «Rome2rio»

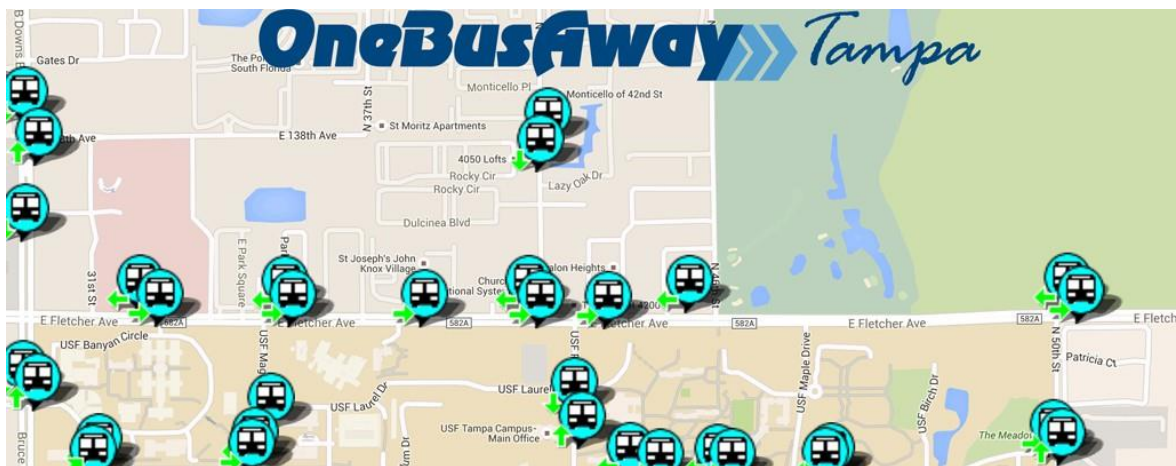


Рисунок 8 - Сервис «OneBusAway»

Статистика **OneBusAway** - пользователи и аналитика загрузок, конкуренты **OneBusAway** и доля рынка, ежедневный и исторический рейтинг в Google Play Store, ключевые слова и многое другое здесь. ... За последние 10 лет компания Similarweb разработала уникальный подход к

измерению цифрового мира, не имеющий себе равных по масштабу и точности.

OneBusAway - это приложение с открытым исходным кодом, которое позволяет сохранять любимые автобусные остановки, **определять** местонахождение ближайших остановок и получать напоминания о наиболее популярных маршрутах.

Слабое место третьего продукта - широта существующих объявлений и разнообразие тематики добавляемых объявлений.

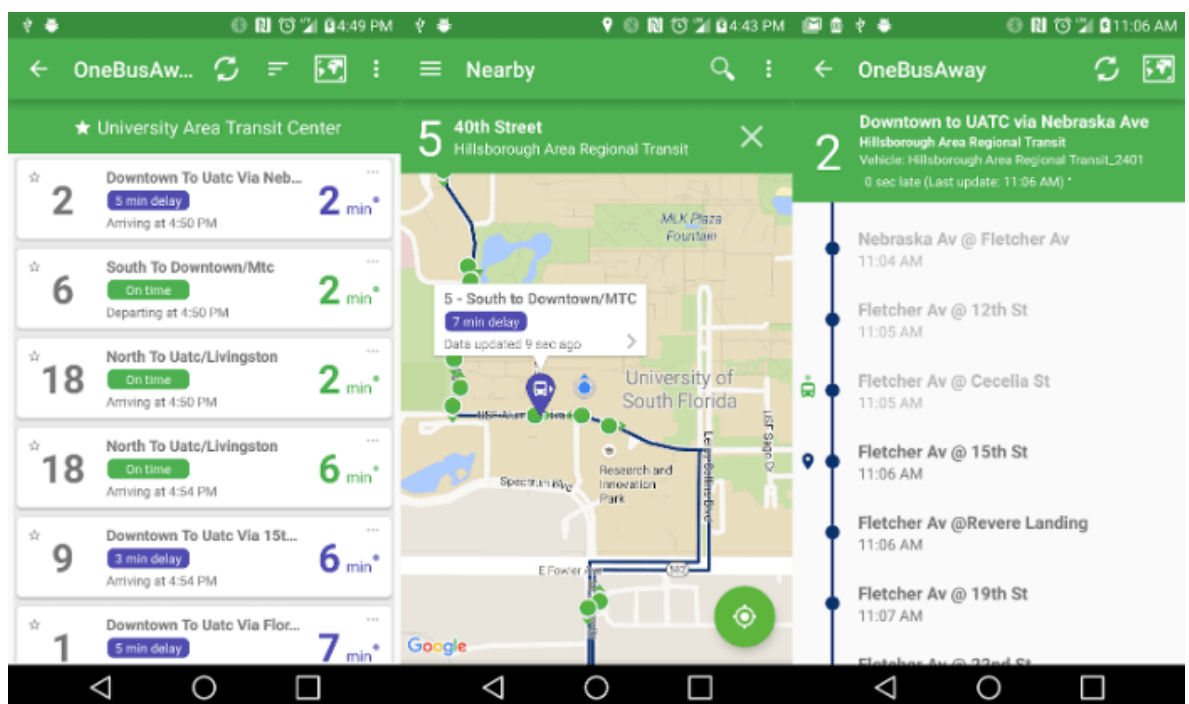


Рисунок 9 - Пример маршрута в «OneBusAway»

Таблица 1.1 - Сравнение аналогов по функциональным ВОЗМОЖНОСТЯМ

Функционал	Разрабатываемое	Citymapper	Rome2rio	OneBusAway
Авторизация и регистрация аккаунта	+	+	+	+
Посмотреть карту	+	+	+	+
Включение геолокации	+	+	-	+

просмотр маршрута на карте	+	+	+	+
маршрута текущего дня	+	-	+	+

Таблица 1.2. - Функции ИС

Функция	Список
Необходимые	Авторизация и регистрация аккаунта водителя; Просмотр Яндекс карты; геолокации водителя; Постройка маршрута на карте; Просмотр маршрута текущего дня; Список водителей.
Желательные	Возможность увидеть маршрута, который был пройден в предыдущие дни;
Возможные	Возможность связаться с водителем посредством чата; отменить маршрут программно;
Отсутствующие	Также можно учитывать товары, которые перевозятся из одного места в другое.

Работа в геоинформационной системе отнесена к обязательным работам в дополнение к обязательным рабочим местам.

Организация рабочих мест помогает нам распределить рабочие места в соответствии с предпочтениями и сосредоточиться на желаемом, в то время как остальные рабочие места будут предоставляться по мере необходимости.

1.Описание заданий:

- Водителя.
- Создание аккаунта.
- Открытие карты.
- Расположение на карте.

- Отображение маршрута.
- 2. К администратору:
 - Просмотр текущих дневных маршрутов.
 - местоположений водителей.
 - поиска водителей.

1.3 Анализ объекта исследования

Давайте представим этот SWOT-опрос, который является показателем сильных и слабых сторон компании в конкретной области.

Сила и слабость конкретного случая определяются внешним видом.

Таблица 1.3. – SWOT - анализ.

SWOT-анализ разрабатываемой системы
1. Простота использования. 2. возможность следовать за водителем. 3. возможность напрямую общаться с водителем.
1. Плохая безопасность в системе. 2. возможность технических сбоев. 3. низкая цена продукта.
1. Очень мало прямых аналогов в систему 2. возможность добавления разработки от организации 3. тайм-трекинг
1.Высокая конкуренция. 2.мы можем закончить разработку и продажу компаниями. 3. законодательство.

Оценку внутренней среды фирмы ее силу и слабость а также внешних возможностей и угроз обычно называют SWOT-анализ. Это легкий в применении инструмент быстрой оценки стратегического положения компании.

А	В	С	D	D	Е	Ф
Планировщик	Интерфейс с Server Характеристика Browser	Проложить маршрут	Server 7	Пользователь	В момент нахождения	Развитие систем отслеживания
Владелец	Частота использования Качество поиска	Найти водителя	Server аренда		Дата Запуска	Прибыль, Получение прибыли
Проектировщик	Гр. System ВН. Поток	Найти сервисный центр, Reg/авт + email.	CXD-server, СУБД, Сервер электронной почты	Администратор, Пользователь	Программирование, отладка, тест, сопровождение.	Создание типового проекта
Конструктор	Юр. Документы, Документы ЖЦ	Оформление документов	Server-юр	Пользователь	Документ сопровождение ЖЦ	Юр. Грамотно вывести ИС на рынок
Субподрядчик	Частота использования Качество поиска	Найти водителя	Server ИС		Дата Запуска	Прибыль, Получение прибыли

Проведем анализ метода ISA

Наконец, сделайте диаграмму ИС (Рисунок 10).

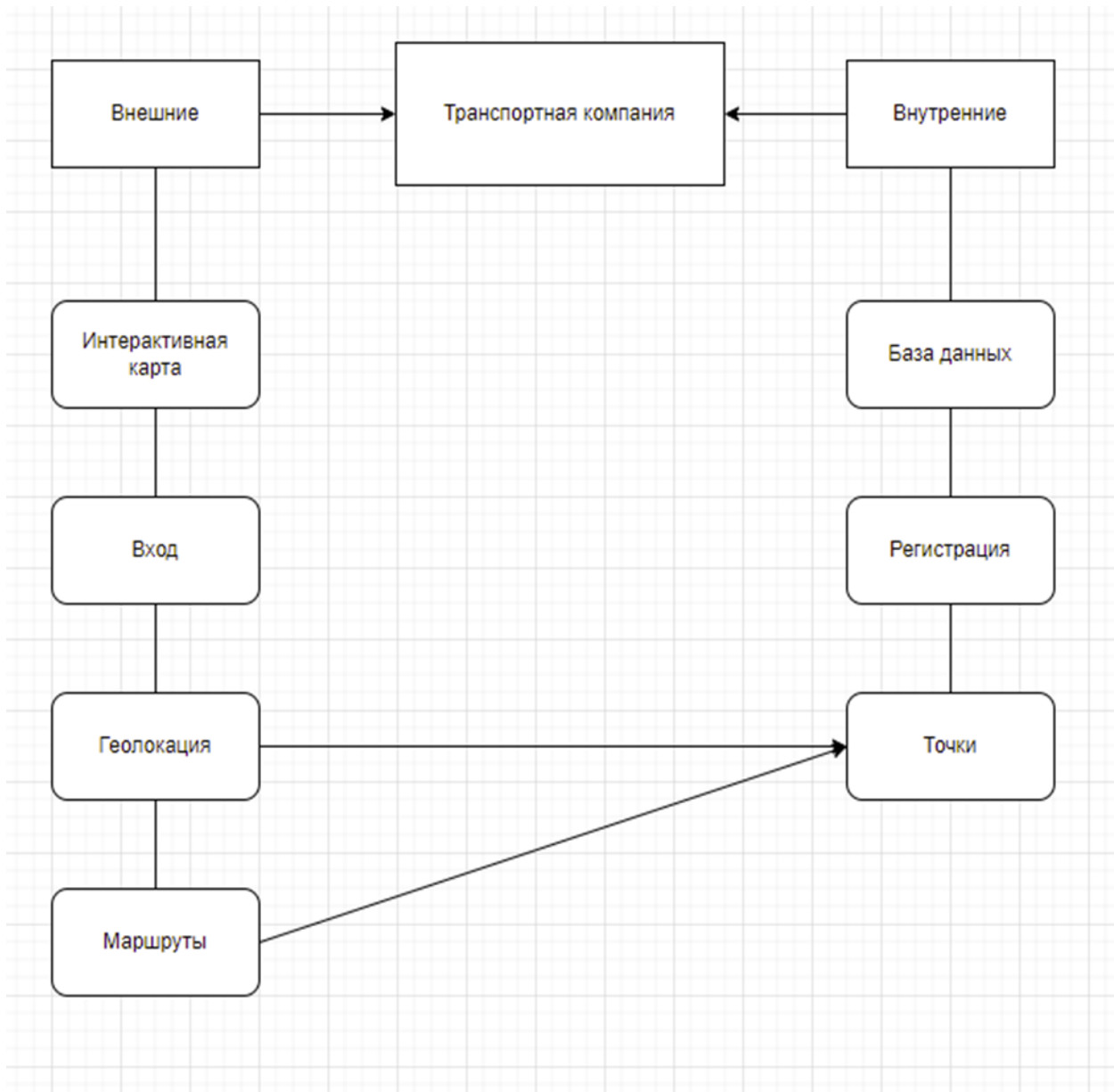


Рисунок 10 – Функционирование ИС

С помощью этой информационной системы полезно повысить эффективность и сократить количество затрат во времени.

Некоторые функции, такие как отслеживание транспортных средств и построение маршрута, будут перенесены в следующие функции. ИС использует географическую информацию система для отслеживания водителя, определяя его географическое положение.

То, что было сделано относительно сильных и слабых сторон, дает нам четкое представление о разработанных геоинформационных системах, а также помогает четко определить путь дальнейшего развития информационной системы с целью устранения текущих проблем.

1.4 Экономическая эффективность

Без сомнения, фантастический веб-сайт может помочь в расширении вашей компании. Создание хорошего первого впечатления может помочь увеличить доход и удержать клиентов.

Всем известно, что веб-сайты являются одним из самых эффективных маркетинговых инструментов для любой компании.

Веб-сайты являются одним из маркетинговых инструментов для любого бизнеса.

Поэтому важный вопрос заключается в том, сколько стоит разработка интегрированного веб-сайта, отвечающего потребностям любой компании.

В двух словах, веб-приложение - программа с определенным набором функционала, использующая в качестве клиента браузер. Другими **словами**, если приложению для осуществления бизнес-логики требуется сетевое соединение и наличие на стороне пользователя браузера, то его относят к **веб-приложению**.

Бесспорно, стоимость разработки веб-приложения равна количеству затраченных на него рабочих часов, умноженному на почасовую ставку команды веб-разработчиков.

Команда разработчиков зависит от количества рабочих часов для разработки приложения в зависимости от степени сложности, а под сложностью понимаются настраиваемые функции и способ использования

приложения, в дополнение к этому будет значение для расходы после выпуска приложения.

Итак, мы рассмотрим три фактора на этой кривой:

- 1- сложность.
- 2- цена за час.
- 3- стоимость после создания приложения.

Самый простой вариант приложения - создать сайт, которым может управлять один человек без необходимости проведения комплексных операций, например: сайт, который показывает нам, что представляет собой компания и какие услуги компании нам предлагают.

Программное обеспечение для Интернета усложняется:

1-Возможность пользователей приложения регистрироваться и добавлять

- 2-контент пользователям.
- 3-покупка и продажа в Интернете.
- 4-возможность покупки и продажи на сайте (C2C).
- 5-слияние приложения с другими приложениями.
- 6-создание сайта онлайн для других географических областей.

Стоимость заявки зависит от:

- 1-Функции.
- 2-факторы использования.

Интеграция со сторонними системами

Если мы хотим интегрировать веб-программу с другим сайтом, нам понадобится API для сторонней интеграции.

Простой пример: Предположим, мы хотим передать данные пользователя в ERP или CRM компании. Интеграция занимает несколько минут, а в остальных случаях работа занимает несколько дней.

Этот процесс усложняется, когда системный / сторонний сайт, который интегрирует пользовательский сайт, не взаимодействует

должным образом или другое приложение имеет плохую информацию об API, поэтому существует много ожиданий.

Производительность веб-приложения

Мы должны быть осторожны, если одновременно много пользователей приложения. Каждая часть приложения имеет свои ограничения для пользователей с возможностью одновременной работы с ней. Поэтому, поскольку веб-сайт будет использоваться многими людьми одновременно, он должен быть сильно оптимизирован, чтобы выдерживать большее количество пользователей.

Обновление данных в режиме реального времени

В том случае, если сайту необходимо обновлять контент каждую секунду, для этого требуется другая методология разработки. Другими словами, нам понадобятся совсем другие технологии, со сравнением сайта приложения или другого сайта, который может обновляться за день или за неделю. Такими приложениями могут быть обменные системы, мгновенное общение, Google-документы (если работа над документом ведется одновременно с другими людьми), онлайн-игры.

Что касается интенсивных мер безопасности, то за счет правильной методологии написания кода разработчики выполняют основные и важные требования к безопасности программы.

Поэтому в различных приложениях требуются дополнительные меры безопасности.

И приложение будет надежно защищено с точки зрения личной информации для пользователей, например: онлайн-оплата будет затруднена, если группа пользователей может продавать товары другим людям в веб-программе.

Это изображение дает нам представление о рыночных ценах. Мы можем посмотреть последние обновления цен для разработчиков.

Offshore software development rates by region

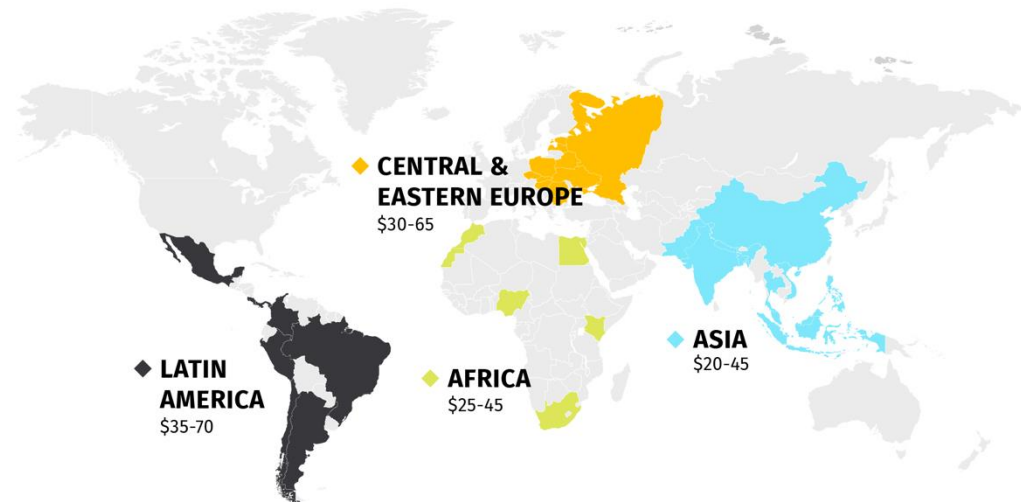


Рисунок 11 - Представление о рыночных ценах

По мнению некоторых пользователей сети, есть возможность получения подходящей прибыли для малых и средних проектов по веб-разработке, примерно от 30 до 50 долларов.

Если работа над проектом рассчитывается по часам, разработчик может умножить сумму на среднее количество отработанных часов, чтобы получить окончательную сумму. Поэтому есть указание на то, что у одних разработчиков одна почасовая оплата, включающая в себя работу всей команды, а с другой стороны, у других совершенно разные почасовые ставки для разных разработчиков и специалистов, участвующих в одном проекте.

Расходы после выпуска проекта:

- 1-исправление и исправление ошибок если таковые имеются.
- 2-инфраструктура приложения такая как базы данных и другие.

Предполагается, что при выдаче программы или приложения будут возникать ошибки, даже если тот, кто создал эту программу, является

лучшим разработчиком. При тестировании приложения используются идеи, согласованные клиентом и разработчиками.

В более понятном смысле перед тестированием приложения выбираются все устройства, на которых мы можем использовать приложение, и приложение также отображается более чем одному другому пользователю вместе с заказчиком.

Затем разработчики устанавливают гарантию качества в соответствии с этими положениями.

Очень сложно ожидать, что приложение будет полнофункциональным без дефектов. Например, может быть запущено обновление для одной из программных систем, которые были объединены с приложением до новой версии, и в связи с произошедшим изменением возникнут проблемы в функциях приложения. Было бы разумно отложить часть денег на непредвиденные ошибки, которые могут возникнуть после запуска. Часто стоимость исправления ошибок после релиза приложения составляет от 10 до 30 % от первоначального бюджета разработки приложения.

В инфраструктуру приложения входит:

1- установка ориентировочного имени для приложения от 5 до 50 долларов.

2- хостинг приложения от 4 долларов до 500 долларов, учитывая место для хранения и особенности приложения.

То, что обсуждалось в этой части проекта, не было всеобъемлющим, так как каждый проект уникален. Поэтому можно считать, что эти причины играют важную роль в определении стоимости приложения.

1.5 Сроки окупаемости

Во-первых, давайте вместе определим затраты на разработку. Пример: Человек пользуется бесплатной версией дополнительных утилит, то что разработчик берет с зарплаты это стоимость и ничего более (при условии, что программа не будет выдавать ему никаких обновлений и она будет доступна бесплатно, другими словами, ответственного за программу нет).

Затем мы определим, что мы выиграем от использования такой системы.

Цель программы - облегчить переводы в компаниях, поэтому, если компания предоставляет фиксированную сумму за продукт и определенный период в дополнение к бесплатным дням, то мы можем предоставить рабочих и начать еще один новый проект. Мы можем найти возмещение на основе среднего тестового запроса (если человек примет во внимание этот метод оценки, ожидается, что период возврата составит от 3 до 12 месяцев). В заключении важно заметить, что после создания каждого метода анализа веб-приложений дальнейшая разработка становится более приемлемой для применения и понимания.

Раньше они делали сравнительный анализ аналогичных информационных систем в Интернете, и собирали все преимущества подобных сервисов для веба, чтобы сделать разработку приложения значительно улучшенной, но плохая сторона этих аналогов заключается в отсутствии некоторых и большое количество других категорий приложений. Определена предполагаемая категория веб-приложения и освоены современные концепции пользователей.

Важно, чтобы у продукта была простая идея, но мы выиграли от него в нескольких функциях, чтобы сделать приложение более успешным и быть в первом порядке, а не иметь для него большого конкурента в

отделе грузовых перевозок. Также рассматривается стратегия развития приложения и ожидаемые обновления для этого приложения.

Следующим шагом будет создание функционального дизайна системы.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИС

Основное внимание в области информационных систем уделяется методологии, используемой при проектировании информационных систем. Ознакомимся с системами и методами управления и методами информационных систем. Функция построения, пользователи, операторы, аналитики, программисты и другие лица создают документы, которые являются предметом дискуссий о разделении и выравнивании, управлении процессами и контроле, уровнях управления, открытых системах и проектировании информационных систем управления. Далее следует обсуждение функционального анализа и спецификации, правил и процессов, а также моделирования и анализа данных. Проблемы включают в себя выбор наилучших проектных решений, передачу решений сообществу пользователей, а также создание схем и концепций для моделирования данных, доменов и целостности домена. В статье обсуждаются вопросы оценки системы, управления проектами, участия пользователей и человеко-компьютерного интерфейса. Потоки данных, анализ данных и модели данных, стратегии участия пользователей, оценка после внедрения и аудит - вот некоторые из затронутых тем. Программисты и ученые, интересующиеся работой информационных систем, могут найти в тексте полезную информацию.

2.1 Концептуальное проектирование

Дизайн информационной системы устанавливается на этом этапе проверки перед проектом и концептуальным представлением информационной системы. Концептуальное представление технологической системы является лишь первым этапом представления, поскольку в нем определяется решение, которое иллюстрирует

следующий рисунок, проводятся исследование и систематизация эталонов технических решений с их компоновкой и формированием. Другим объяснением концептуального представления является начало проектирования, которое включает в себя создание копии элемента, и оно показывает нам передний край средств разработки.

Концептуальное планирование основано на распределении потребностей в информатике, а также данных, которые включают эти требования. Процесс концептуального планирования возникает при изучении деятельности организации. Сначала определяются результаты, которых хочет достичь организация, а затем анализируются текущие рабочие процессы, и благодаря им гарантируются эти результаты, которые должны быть достигнуты.

Затем эти процессы распределяются по подпроцессам более низкого уровня, чтобы получить доступ как к программам, так и к функциям без необходимости разделения. Затем требования каждой программы настраиваются на основе информации, которую она будет запускать автоматически. Таким образом, было проанализировано начало работы и определены результаты, которые должны быть достигнуты.

2.2 Объектно-ориентированное проектирование

Метод организации системы взаимосвязанных объектов для решения программной проблемы известен как объектно-ориентированный дизайн. Вот один из методов разработки программного обеспечения.

Объект - это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.

Взаимодействие этих объектов описывает объектно-ориентированная программа. Практика определения объектов и их взаимодействий для решения проблемы, которая была распознана и зарегистрирована с помощью объектно-ориентированного анализа, известна как объектно-ориентированное проектирование.

Описание объектов в объектно-ориентированном проектировании на основе классов, которое следует ниже, исключает подходы, основанные на прототипах объектов, где объекты обычно создаются путем клонирования других (прототипов) объектов, а не путем создания копий классов.

Объектно-ориентированное проектирование - Это фундаментальный закон познания, когда мы открываем новый домен, типа проектирования взаимодействия, сперва поражены уникальностью, но по прошествии времени начинаем находить параллели с уже существующим в нашем мире.

Результаты объектно-ориентированного анализа служат исходными данными для объектно-ориентированного проектирования. Отметим что выходной артефакт не обязательно должен быть полностью разработан, чтобы его можно было использовать в качестве входных данных в объектно-ориентированном проектировании; анализ и проектирование могут происходить одновременно, и на самом деле результаты одного действия могут подпитывать результаты другого в быстрой петле обратной связи в рамках итеративного процесса. Разработка артефактов может выполняться постепенно, а не полностью, и таким же образом можно выполнять как анализ, так и проектирование.

Следующие примеры входных артефактов для объектно-ориентированного проектирования:

Концептуальная модель: она инкапсулирует концепции предметной области и является результатом объектно-ориентированного анализа. Он

выбран специально для того, чтобы концептуальная модель не зависела от особенностей реализации, таких как параллелизм или хранение данных.

Вариант использования: описание ряда действий, которые в сочетании приводят систему к выполнению значимого действия. Каждый вариант использования предлагает один или несколько сценариев, которые объясняют, как система должна взаимодействовать с пользователями, называемыми действующими лицами, для выполнения определенной задачи или функции. Конечные пользователи или другие системы могут использовать субъектов дела. Диаграммы вариантов использования часто создаются путем уточнения вариантов использования. Идентификация действующих лиц (пользователей или других систем) и процессов, которые они выполняют, осуществляется с помощью диаграмм вариантов использования.

Диаграмма последовательности систем. Диаграмма последовательности систем (SSD) - показывает взаимодействия процессов, расположенные во временной последовательности в области разработки программного обеспечения. Он изображает задействованные процессы и последовательность сообщений, которыми обмениваются процессы, необходимые для выполнения этой функции. (Если применимо)

Документация по пользовательскому интерфейсу документ, который иллюстрирует и обсуждает, как будет выглядеть и ощущаться пользовательский интерфейс готового продукта. Хотя это и не обязательно, это помогает дизайнеру визуализировать готовый продукт.

Проект реляционной базы данных (если применимо). Модель данных (или datamodel) - это абстрактная модель, которая организует элементы данных и стандартизирует их связь друг с другом и со свойствами объектов реального мира. Например, модель данных может указывать, что элемент данных, представляющий автомобиль, состоит из

ряда других элементов, которые, в свою очередь, представляют цвет и размер автомобиля и определяют его владельца.

Понятия, связанные с объектами

Характеристики языка программирования, реализованные на уровне реализации, поддерживают пять основных принципов объектно-ориентированного проектирования. Для описания этих функций часто используются следующие общие имена:

Объект/класс: тесная связь или принадлежность между структурами данных и методами или операциями, влияющими на данные. Это называется классом или объектом (объект создается из класса). Каждый предмет имеет определенную функцию. Его характеристики - что это такое и что он может делать - определяют его. Класс, представляющий собой группу связанных объектов, может включать в себя объект.

Соккрытие информации: способность скрывать определенные части объекта от посторонних. Это реализуется с помощью языковых терминов, которые позволяют обозначить переменную как частную или общедоступную.

Способность класса изменять или заменять функциональность другого класса называется наследованием. Так называемый подкласс имеет существенную часть, которая является производной (унаследованной) от суперкласса в дополнение к собственному набору функций и информации.

Интерфейс объектно-ориентированного программирования позволяет отложить применение метода. возможность указывать сигнатуры методов или функций без их фактической реализации.

Подтипирование, тип полиморфизма, представляет собой возможность замены объекта его подобъектами. способность объектной переменной содержать как главный объект, так и все его дочерние элементы.

Понятия дизайна

Создание диаграммы классов из концептуальной диаграммы и идентификация материнских элементов Сущность часто присваивается классу.

2.3 Диаграмма вариантов использования (USE-CASE)

Модель вариантов использования состоит из диаграмм вариантов использования (вариантов использования). Системная функция, которая позволяет пользователю получить какое-то лично значимое, конкретное и поддающееся количественной оценке последствие, является прецедентом.

Каждый вариант использования описывает отдельную услугу, которую смоделированная система предлагает в ответ на запрос пользователя, т. е. описывает использование системы.

Из-за этого в отечественной терминологии прецеденты часто называют прецедентами. Варианты использования чаще всего используются для описания внешних спецификаций разрабатываемой системы или для описания рабочего поведения существующей системы.

Сценарии использования также косвенно определяют обычные способы взаимодействия пользователей с системой, что позволяет пользователям эффективно использовать возможности системы.

На диаграмме прецедентов показана каждая функция, которая была создана и стала доступной для определенной группы пользователей информационной системы мониторинга грузоперевозок. Применялись отношения обобщения, ассоциации, включения и расширения.

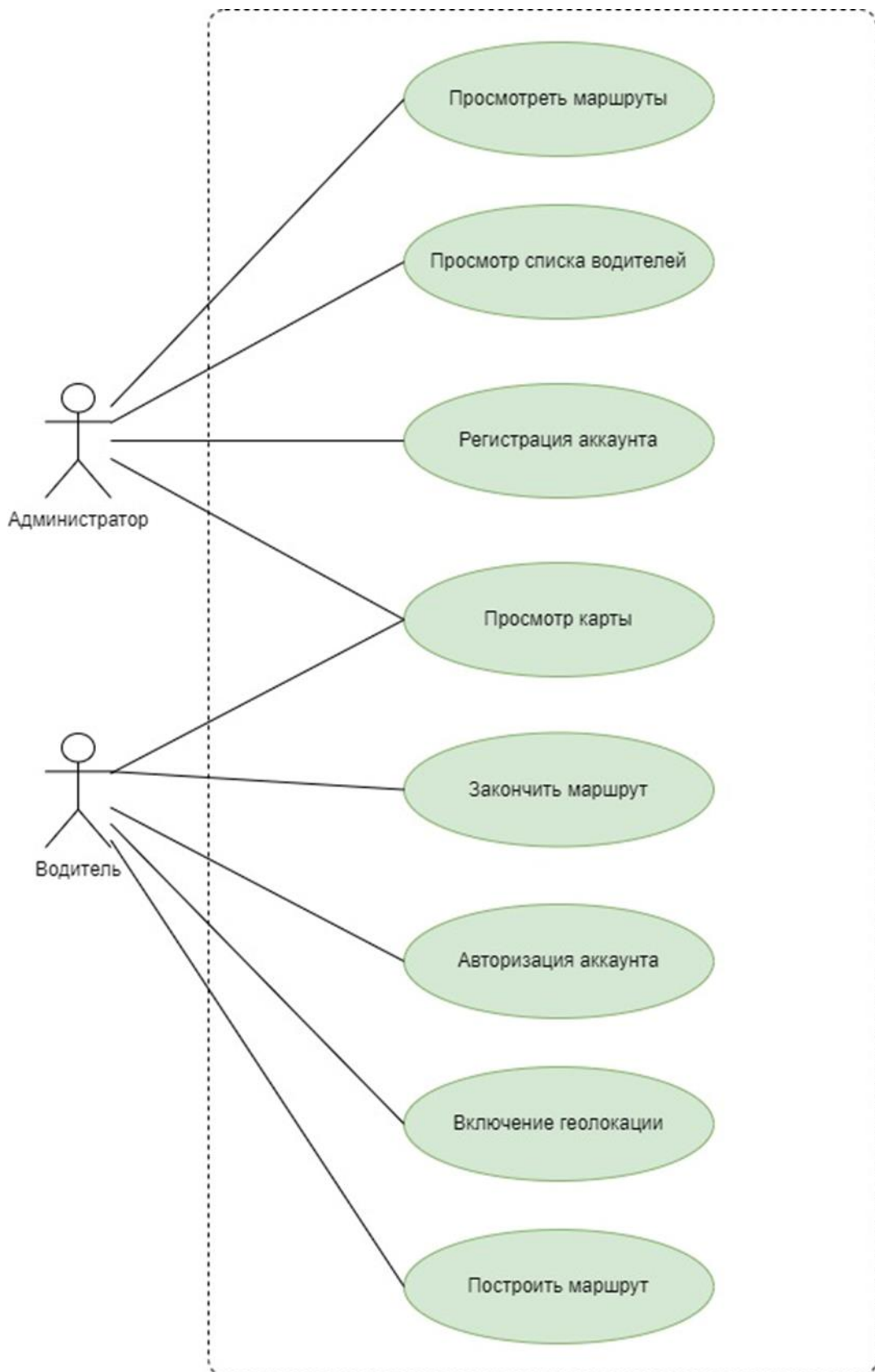


Рисунок 12 – Диаграмма тематического исследования (USE-CASE)

Операционные процессы информационной системы изображены на этой диаграмме. В качестве действующих лиц на рисунке подчеркнуты «Администратор информационной системы» и «Водитель». На диаграмме также показаны восемь вариантов использования, которые определяют последовательность действий, предпринимаемых в ответ на действия акторов. Следующее действие - нарисовать диаграмму последовательности.

2.3 Диаграммы последовательности

Поскольку она показывает, как и в каком порядке взаимодействует набор объектов, диаграмма последовательности является формой диаграммы взаимодействия. Инженеры-программисты и бизнес-эксперты используют эти диаграммы для понимания спецификаций новой системы или для записи существующей процедуры. Диаграммы событий и сценарии событий - это Сценарием называется последовательность событий, которая может иметь место при конкретном выполнении системы. Сценарии могут иметь разные области влияния: они могут включать все события, происходящие в системе, либо только события, возникающие и влияющие только на определенные объекты системы.

Это позволяет графически отображать простые исполняемые сценарии.

Система делает положительное определение на диаграмме, которую необходимо обновить и утвердить:

- Внешние действующие лица
- Сообщения (методы), вызываемые этими акторами
- Возвращаемые значения (если есть), связанные с предыдущими сообщениями
- Индикация любых циклов или области итерации

Схемы системной последовательности часто используются экспертами по разработке проектов, чтобы показать, как выполняются определенные задачи между пользователями и системой. Эти обязанности могут быть повторяющимися, легкими или трудными. Цель состоит в том, чтобы визуально представить вариант использования. Для создания блок-схемы системы (UML) необходимо знание унифицированного языка моделирования. Схемы системной последовательности часто используются экспертами по разработке проектов, чтобы показать, как выполняются определенные задачи между пользователями и системой. Эти обязанности могут быть повторяющимися, легкими или трудными. Цель состоит в том, чтобы визуально представить вариант использования. Для создания блок-схемы системы (UML) необходимо знание унифицированного языка моделирования. Решение или последующее действие находится под предыдущим действием после завершения каждого действия. Читая строки, вы можете подробно наблюдать, как выполняются те или иные действия в поставляемой модели и в каком порядке.

Следует отметить, что существует два вида диаграмм последовательности: диаграммы на основе кода и диаграммы UML. Последний получен из исходного кода и в данной работе не учитывается.

В вышеупомянутом разделе используются следующие диаграммы. (Рисунки 13-15):

1-Логин - используется в информационной системе для обработки запроса на доступ для выполнения определенных действий, связанных с системой. На рисунке 13 показано, как работает авторизация драйвера веб-приложения.

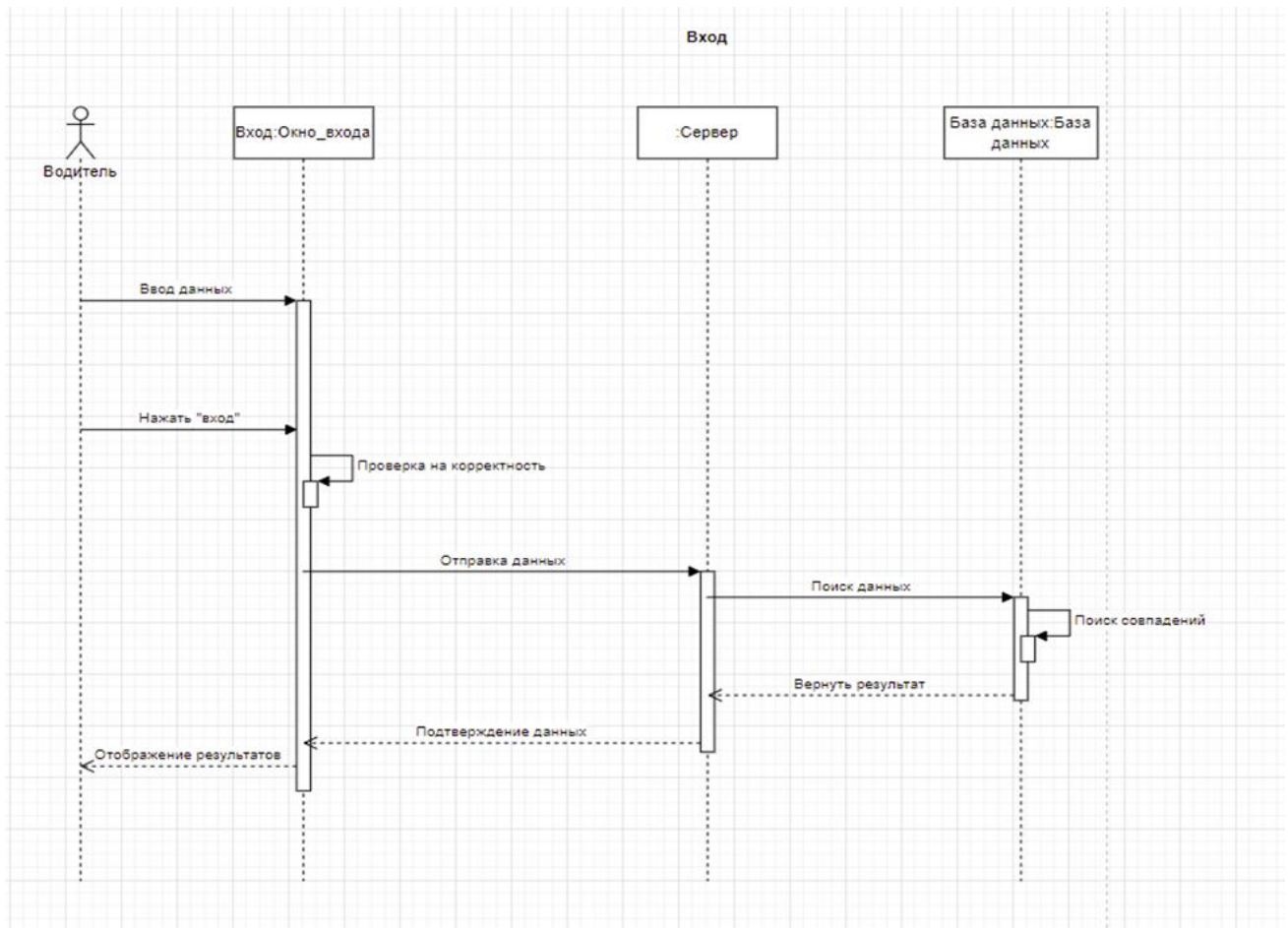


Рисунок 13 – Диаграмма последовательности «входа» для системного драйвера

Схема работы системного драйвера «Авторизация» выглядит следующим образом:

1. Водитель нажимает кнопку входа в систему после ввода информации в поле входа.

2. Начинается проверка данных .

3. Данные передаются на сервер

4. В базе данных выполняется поиск .

5. Сервер получает поток данных обратно и проверяет запись.

6. Отображение результатов

2. Регистрация - используется в информационной системе для обработки запроса на создание учетной записи, предоставляющей доступ для выполнения определенных системных действий. Добавить что-либо в

приложение без регистрации невозможно. Диаграмма на рис. 14, на которой изображена регистрация администратора приложения, относится к этому процессу.

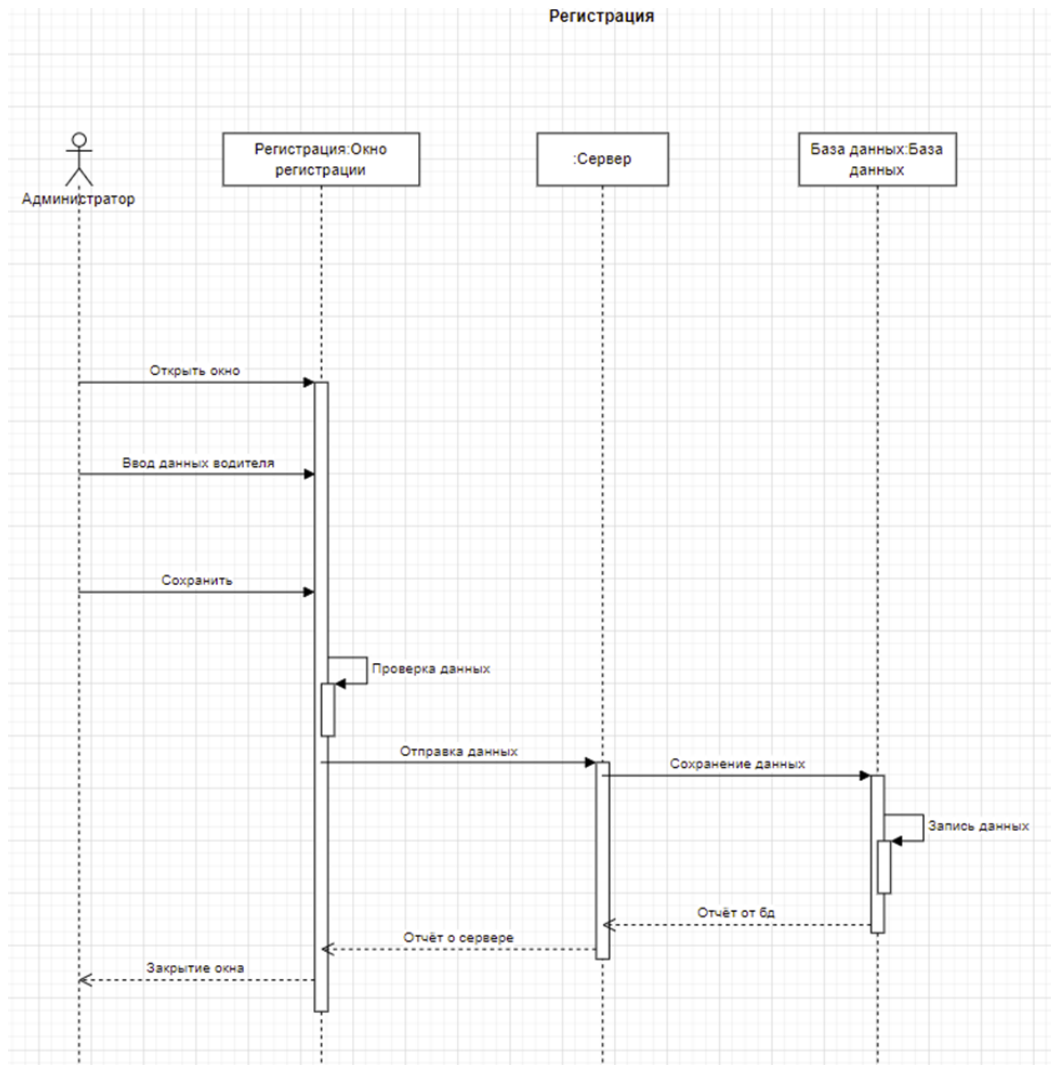


Рисунок 14 – Схема последовательности действий системного администратора «Регистрация»

Алгоритм работы схемы «Авторизация» для системного администратора выглядит следующим образом:

1. Администратор вводит информацию о водителе и сохраняет ее в окне регистрации.
2. Запускается проверка данных.
3. Данные передаются на сервер
4. Сейчас происходит сохранение базы данных.

5. Сервер получает поток данных обратно и проверяет сохранение.
6. Закрытие окна
3. Добавление маршрута используется для предоставления драйверам этого продукта нового пути. Этот проект, показанный на рис. 15, учитывает включение маршрута драйвером приложения.

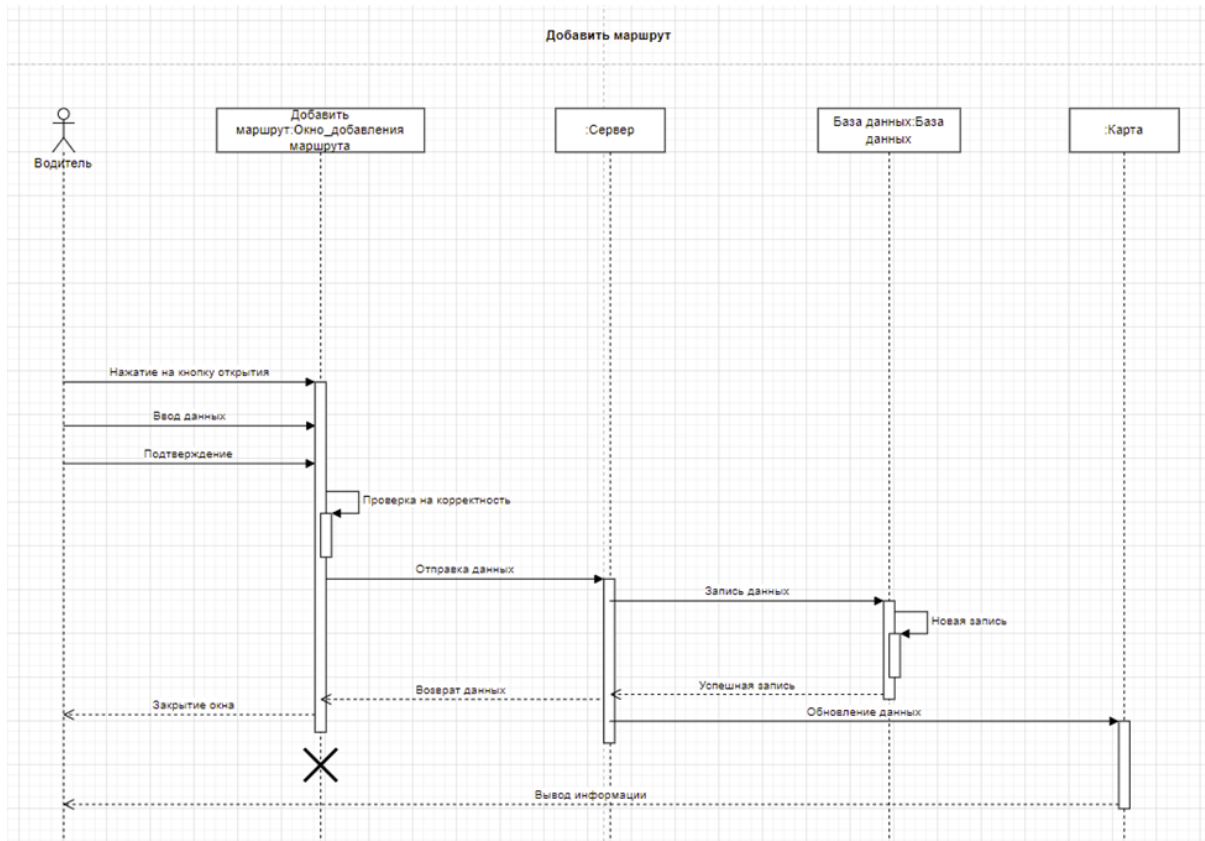


Рисунок 15 – Схема последовательности «Добавить маршрут» для системного драйвера

Алгоритм работы схемы «Авторизация» для системного администратора выглядит следующим образом:

1. Водитель нажимает кнопку запуска окна добавления маршрута, вводит необходимую информацию, а затем нажимает кнопку «Ввод».
- Начинается проверка данных 2.
3. Данные передаются на сервер
4. Создается запись базы данных (новая запись)

5. Данные через сервер возвращаются драйверу с ответом после успешной записи.

6. Закрытие окна

7. Вывод информации с карты на дисплей (после их обновления)

Данные диаграммы последовательности основаны на том, как события обычно выполняются в информационной системе. Дополнительные схемы приведены в Приложении 1.

Рассмотрите диаграммы классов при описании структуры информационной системы.

2.4 Диаграмма классов

Диаграмма статической структуры, известная как диаграмма классов унифицированного языка моделирования (UML), используется в разработке программного обеспечения для представления структуры системы путем отображения классов, свойств, действий (или методов) и отношений между объектами.

В объектно-ориентированном моделировании диаграмма классов служит основным компонентом построения. Он используется как для технического моделирования, преобразующего модели в компьютерный код, так и для общего концептуального моделирования структуры приложения. Диаграммы классов также можно использовать для моделирования данных. На диаграмме классов классы обозначают основные компоненты, взаимодействия внутри приложения и классы, требующие программирования.

Классы изображены на диаграмме в виде прямоугольников с тремя делениями в каждом:

Название класса находится в верхнем отсеке. Начальная буква заглавная, выделена жирным шрифтом и расположена по центру.

Характеристики класса находятся в центральном отсеке. Начальная буква строчная, и они выровнены по левому краю.

Операции класса расположены в нижнем отсеке. Начальная буква строчная, и они также выровнены по левому краю.

Диаграмма классов создается на протяжении всего процесса проектирования системы, чтобы помочь прояснить статические отношения между различными классами. Классы концептуального проектирования часто подразделяются на подклассы тщательного моделирования.

Эти диаграммы классов могут быть дополнены диаграммой состояний или конечным автоматом UML для дальнейшего описания поведения систем.

Эти обозначения должны использоваться перед именем члена, чтобы указать, виден ли отдельный член класса (т. е. видим ли какой-либо атрибут или метод).

Пакет + Общедоступный - Частный # Защищенный

Производное свойство - это определяется как предел отношения приращения функции к приращению её аргумента при стремлении приращения аргумента к нулю (при условии, что такой предел существует). Имя производного свойства отображается с косой чертой «/» перед ним.

Области экземпляра и класса, которые обозначаются именами, выделенными курсивом, представляют собой два типа областей, которые UML определяет для членов.

Члены экземпляра привязаны к этому экземпляру. Значения атрибутов могут различаться между экземплярами.

Состояние экземпляра (т. е. его атрибуты) может измениться в результате вызова метода.

Большинство компьютерных языков считают членов класса «статическими». Сам класс служит областью видимости.

Все экземпляры имеют одинаковые значения атрибутов.

Вызов метода не влияет на состояние классификатора.

Название квалификации для члена должно быть подчеркнуто, чтобы показать степень классификатора. Если нет, область экземпляра берется по умолчанию.

Диаграмма представлена на рисунке 16.

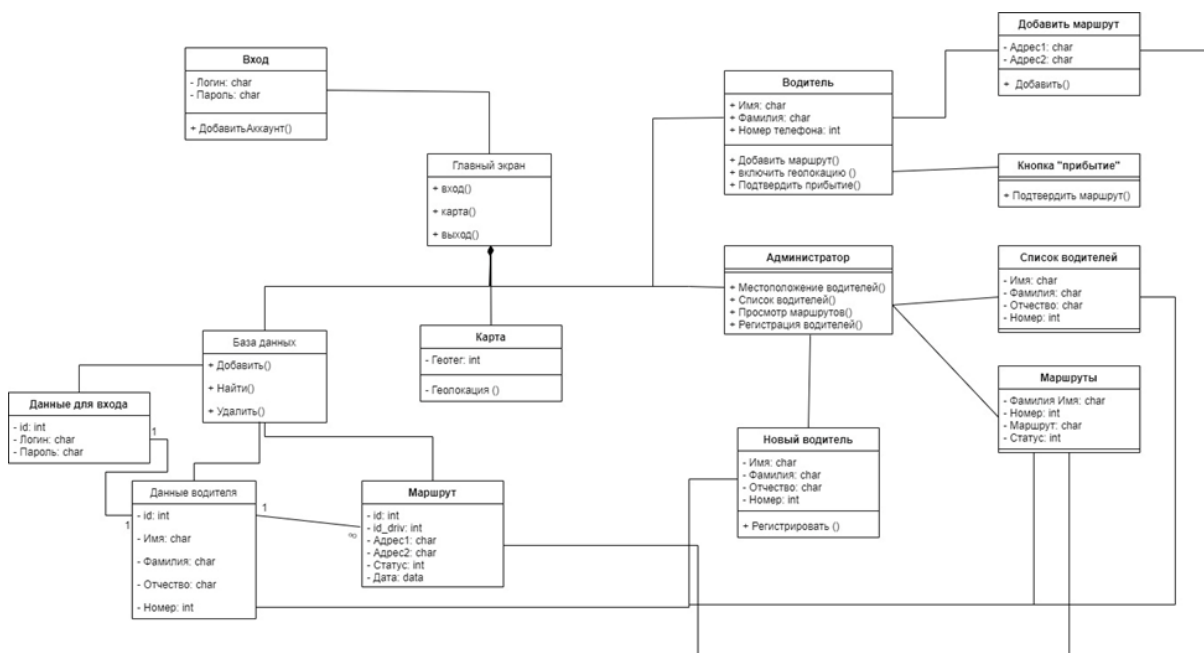


Рисунок 16 – Диаграмма классов

2.5 Диаграмма развертывания

Диаграмма развертывания – это тип UML-диаграммы, которая показывает архитектуру исполнения системы, включая такие узлы, как аппаратные или программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их.

Фактическое аппаратное и программное обеспечение системы часто представляют с помощью диаграмм развертывания. С его помощью можно понять, как система будет применяться к аппарату физически.

По сравнению с другими видами диаграмм UML, которые в основном представляют логические компоненты системы, диаграммы развертывания служат для моделирования аппаратной топологии системы. (См. рис. 17.)

Узлы бывают двух видов:

Хост среды выполнения -

Хост – узел сети, компьютер или любое другое сетевое устройство, предоставляющее сервисы формата «клиент-сервер» в режиме сервера.

Термин «хост» отражает роль устройства или программы как центра хранения данных или управления сервисами, услуги которых хост предоставляет клиентам. В более частном случае под хостом могут понимать любой компьютер, сервер, подключенный к локальной или глобальной сети.

Физический элемент моделируемой системы, существующий во время исполнения и представляющий вычислительный ресурс, который обычно обладает как минимум некоторым объемом памяти, а кроме него зачастую и процессором.

Сетевые структуры могут быть описаны каналами связи между узлами.

Экземпляр узла в моделировании UML - это элемент модели, который символизирует экземпляр или реальное появление узла. Экземпляры узлов основаны на уже существующих узлах.

Время выполнения

Операционная система или система управления базами данных являются примерами конкретных платформ выполнения, которые представлены средой выполнения в моделировании UML. Среду, в которой происходит выполнение модели, можно описать с помощью сред выполнения.

Артефакты В моделях UML артефакты - это некая физическая сущность на уровне платформы реализации, представляющая собой

замещаемую часть системы (программный скрипт, файл с исходным кодом, исполняемый модуль, таблица базы данных и т. п.).

Примеры артефактов:

Экземпляр артефакта в UML-моделировании - это элемент модели, отображающий конкретный экземпляр или реальное появление артефакта. Основой для экземпляров артефакта является существующий артефакт.

Устройства

Устройство на диаграмме развертывания - это узел, который используется для представления физического вычислительного ресурса в системе. Примером устройства является сервер приложений.

Руководство по развертыванию

Спецификация развертывания, описывающая, как артефакт развертывается на хосте, представляет собой просто файл конфигурации, такой как XML-документ или текстовый файл.

Диаграммы отношений развертывания

Отношения в UML используются для представления связи между структурными, поведенческими или группирующими вещами. Это также называется ссылкой, которая описывает, как две или более вещи могут относиться друг к другу во время выполнения системы.

Диаграмма изображена на рисунке 17.

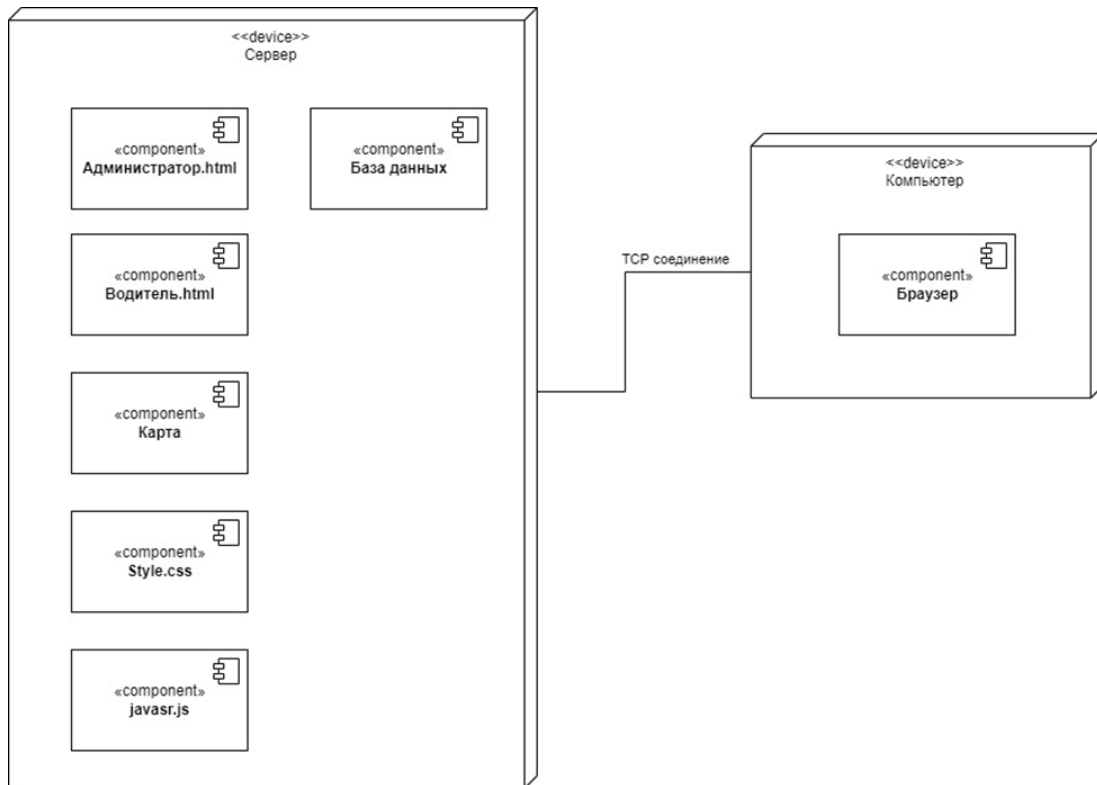


Рисунок 17 – Диаграмма развертывания

2.6 Диаграмма компонентов

Диаграмма компонентов на унифицированном языке моделирования (UML) демонстрирует связи между меньшими компонентами для создания более крупных компонентов или программных систем. Они служат примерами строения систем, которые могут быть сколь угодно сложными.

Вы можете проверить диаграмму компонентов, чтобы убедиться, что требуемая функциональность системы является удовлетворительной.

Разработчик и заинтересованные стороны системы также могут общаться с помощью этих диаграмм. Диаграммы помогают программистам и разработчикам создать дорожную карту для реализации, которая помогает им решить, как назначать задания и где сделать необходимые улучшения навыков. Диаграммы компонентов дают системным администраторам представление о логических компонентах

программного обеспечения и их связях внутри системы, которое они могут использовать для планирования наперед.

Информация, представленная в элементе символа компонента, расширена на диаграмме компонента. Прямоугольная рамка, прикрепленная к элементу компонента, может служить примером интерфейсов, которые предоставляет данный компонент и которые необходимы. Использование графического метода шар-и-гнездо - еще один типичный метод представления интерфейсов. Сплошная линия, ведущая от леденца или шарика, помеченного именем интерфейса, к компоненту, использующему интерфейс, указывает на предоставленную зависимость от компонента к интерфейсу. Полукруг или сокет, помеченный именем интерфейса и соединенный сплошной линией с компонентом, которому нужен этот интерфейс, означает необходимую зависимость использования между ними. Иллюстрацию можно увидеть на рисунке 18.

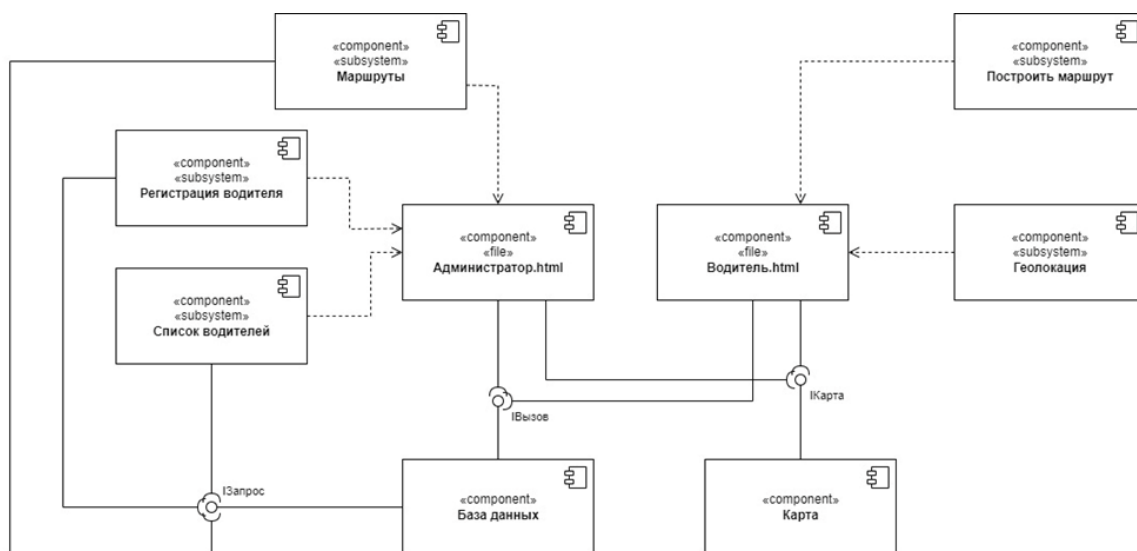


Рисунок 18 – Диаграмма развертывания

После того, как этапы разработки веб-приложения и построения диаграмм были завершены, стало намного легче понять все недостатки и

преимущества сайта. Система должна быть детально проработана еще до ее реализации с использованием разного рода анализов, схем и схем, ведь не редкость встретить красиво написанную программу, которая из-за отсутствия типовой документации приобретает ненужный второстепенный функционал, приделок, становится тягостным и теряет общий вид.

Будущая разработка системы должна учитывать знания, использованные при проектировании системы. Непосредственное создание веб-приложения - это то, что вам нужно сделать дальше.

ГЛАВА 3. РЕАЛИЗАЦИЯ

Разработанная информационная система «Транспортной компании» будет отображаться в разделе «Внедрение». Приведены функциональные возможности и интерфейс реализованной системы, а также описание типовых проектных решений.

Стоит уточнить, что спутниковый мониторинг транспорта - система мониторинга подвижных объектов, построенная с использованием GPS(ГЛОНАСС)-трекеров, оборудования и технологий сотовой и/или радиосвязи, вычислительной техники и цифровых карт. Спутниковый мониторинг транспорта используется для решения задач транспортной логистики в системах управления перевозками и автоматизированных системах управления автопарком. Принцип работы заключается в отслеживании и анализе пространственных и временных координат транспортного средства.

Существует два варианта мониторинга: online - с дистанционной передачей координатной информации и offline - информация считывается по прибытии на диспетчерский пункт.

Геоинформационное обеспечение отображения местоположения и движения контролируемых транспортных средств на электронной схеме маршрутов движения в режиме реального времени и по архивным навигационным данным

Модель транспортной ГИС, в основу которой положен функциональный принцип, состоит из следующих основных компонентов (подсистем):

- подсистема ввода и преобразования данных;
- подсистема обработки и анализа данных;
- подсистемы пространственного анализа;
- подсистема хранения данных;

- база данных (БД);
- система управления базой данных (СУБД);
- подсистема вывода (визуализации) данных;
- подсистема предоставления информации;
- пользовательский интерфейс.

Формирование структуры ГИС начинается с формирования баз данных, основанных на территориальной (географической) привязке данных. Любая ГИС работает с базами данных двух типов - графическими и атрибутивными (тематическими). В графических базах данных хранится графическая основа, атрибутивные содержат данные, которые относятся к пространственным, но не могут быть непосредственно нанесены на карту - это описания территорий или информация, содержащаяся в отчетах. Оба вида баз данных представляет собой файлы (наборы данных). Для работы с ними ГИС должна иметь систему управления базами данных. С помощью СУБД производится поиск, сортировка, добавление и исправление информации в базах данных.

3.1 Описание типовых проектных решений

Для построения геоинформационной системы применялись стандартные методы проектирования:

- Visual Studio Code - это редактор кода для разных языков программирования. Он относительно немного весит, гибкий и удобный.

- Бесплатная библиотека Bootstrap 5, содержащая ряд инструментов для создания веб-сайтов и веб-приложений. включает расширения JavaScript и шаблоны дизайна HTML и CSS для типографики, веб-форм, кнопок, меток, блоков навигации и других элементов веб-интерфейса.

- Инструмент моделирования FlowCharts, веб-служба, позволяющая графически изображать процесс и имеющая множество диаграмм, включая UML, ER и BPMN;

- Использование языков программирования: Сайт будет создан с использованием HTML, CSS, JavaScript и PHP. Важно отметить, что первые два языка отвечают за «костяк» сайта, а следующие два - за его функциональную составляющую.

- Система управления реляционными базами данных MySQL, которая предоставляется бесплатно;

- Наша геоинформационная система будет построена на модуле интерактивной карты от Яндекса. Он прост в использовании и предлагает широкий спектр функций

Упомянутые выше решения будут использованы при строительстве.

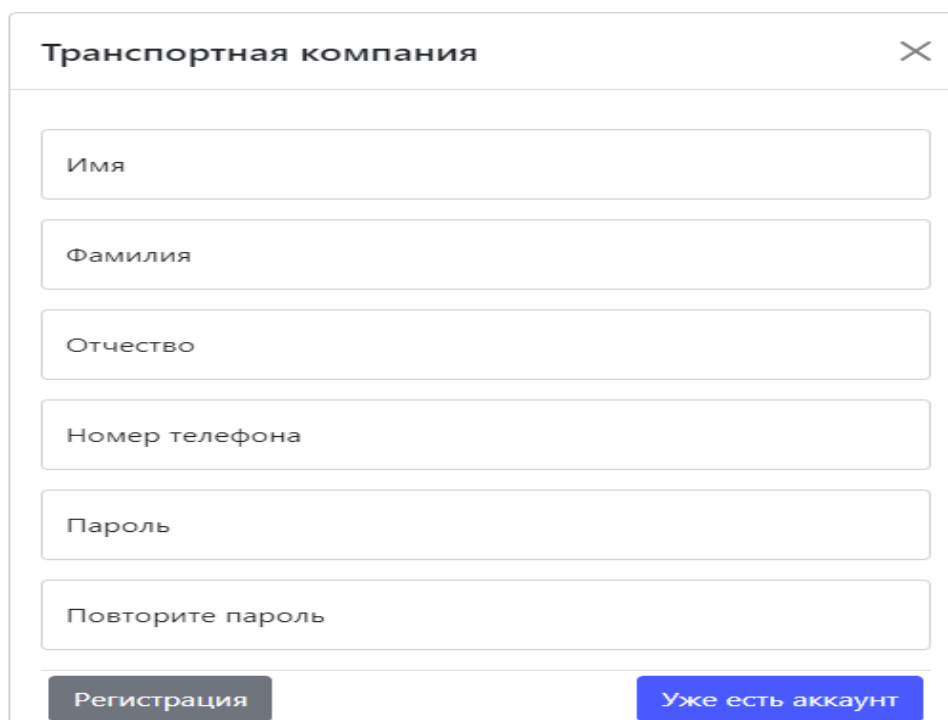
Информационные системы можно внедрять быстрее и, главное, проще при использовании стандартных проектных решений, что помогает избежать необходимости создавать с нуля и вместо этого фокусируется на разработке приложения и его функций.

3.2 Описание реализованной системы

Используя геоинформационные технологии, геоинформационная система «Транспортная компания» помогает компании решать свои логистические вопросы. В результате действий, изложенных в предыдущих главах, мы даем описание системы, которая действительно была введена в эксплуатацию.

Изначально есть только учетная запись администратора, после чего водители или сотрудники должны зарегистрироваться в системе. Для регистрации вам потребуется следующая информация: ваша фамилия,

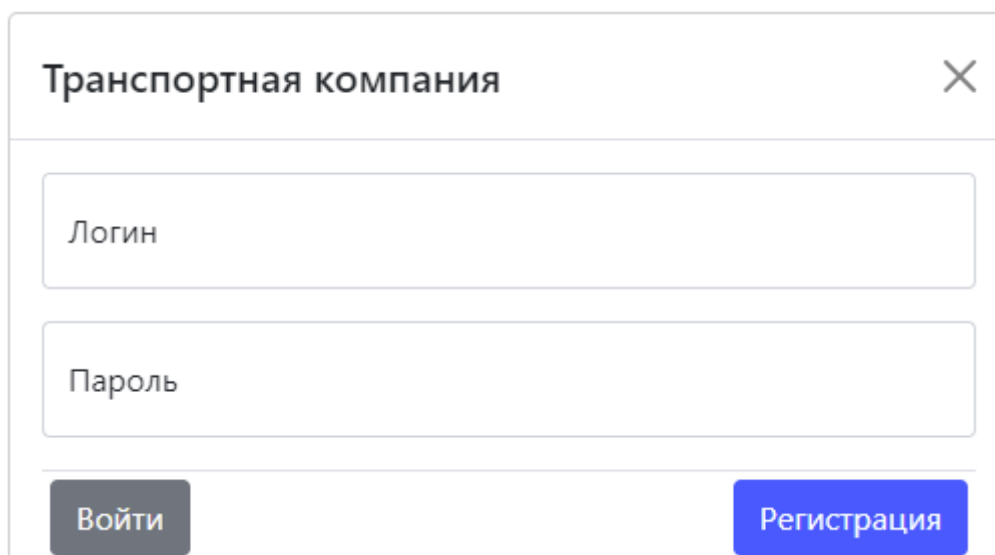
имя, отчество, номер телефона и пароль. В дальнейшем номер телефона будет служить логином. На рисунке 19 показано окно регистрации.



The screenshot shows a registration window titled "Транспортная компания" with a close button (X) in the top right corner. The window contains six input fields stacked vertically: "Имя", "Фамилия", "Отчество", "Номер телефона", "Пароль", and "Повторите пароль". At the bottom, there are two buttons: a dark grey button labeled "Регистрация" and a blue button labeled "Уже есть аккаунт".

Рисунок 19 – Окно регистрации

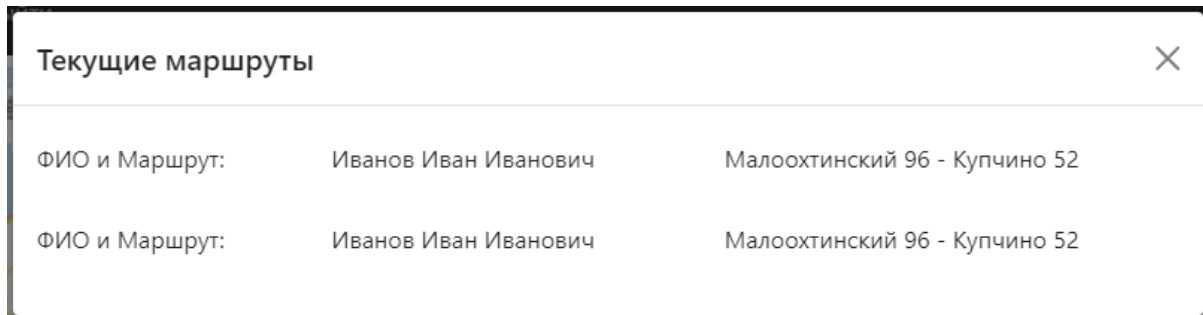
При выборе кнопки «Уже есть учетная запись» в окне останутся только поля логина и пароля, если пользователю достаточно войти в уже созданную учетную запись. (Рисунок 20)



The screenshot shows a login window titled "Транспортная компания" with a close button (X) in the top right corner. The window contains two input fields stacked vertically: "Логин" and "Пароль". At the bottom, there are two buttons: a dark grey button labeled "Войти" and a blue button labeled "Регистрация".

Рисунок 20 – Войдите в свою индивидуальную учетную запись

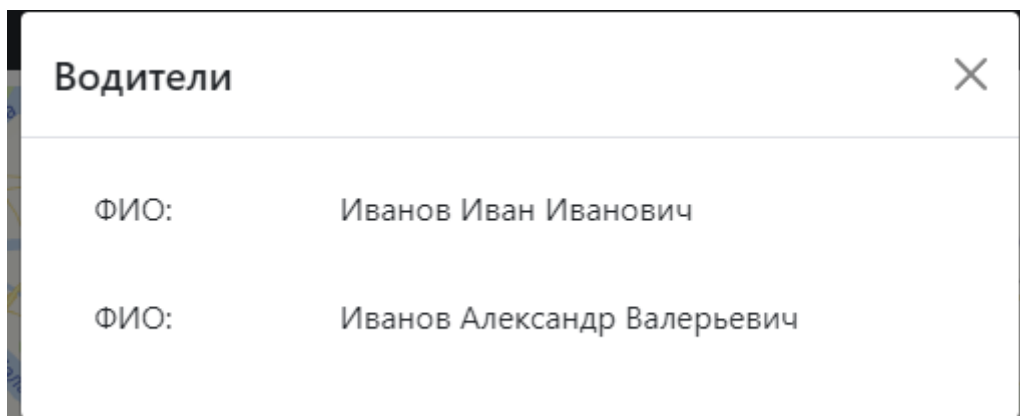
Кроме того, при нажатии на соответствующую кнопку появится окно с подробной информацией о сегодняшних маршрутах, позволяющее администратору просмотреть маршруты на текущий день. (Рисунок 23)



Текущие маршруты		
ФИО и Маршрут:	Иванов Иван Иванович	Малоохтинский 96 - Купчино 52
ФИО и Маршрут:	Иванов Иван Иванович	Малоохтинский 96 - Купчино 52

Рисунок 23 – Существующие маршруты

Разумеется, системному администратору нужен список зарегистрированных в нем водителей. Система показывает список водителей по именам, и администратор может связаться с ними по номеру телефона, если это необходимо, используя базу данных. (Рисунок 24)



Водители	
ФИО:	Иванов Иван Иванович
ФИО:	Иванов Александр Валерьевич

Рисунок 24 – Список автомобилистов

И последняя кнопка это «Выйти», что позволяет вам выйти из системы после того, как ваша задача будет выполнена.

В драйвере есть функциональные различия. Пользователю этого типа нужно только указать путевые точки, чтобы система могла направлять трафик из пункта отправления в пункт назначения; он не обязан проверять список водителей.

Добавление маршрута - первая кнопка после «Аккаунта». Когда вы нажмете кнопку, появится окно, в котором вы можете ввести желаемые адреса. (Рисунок 25)

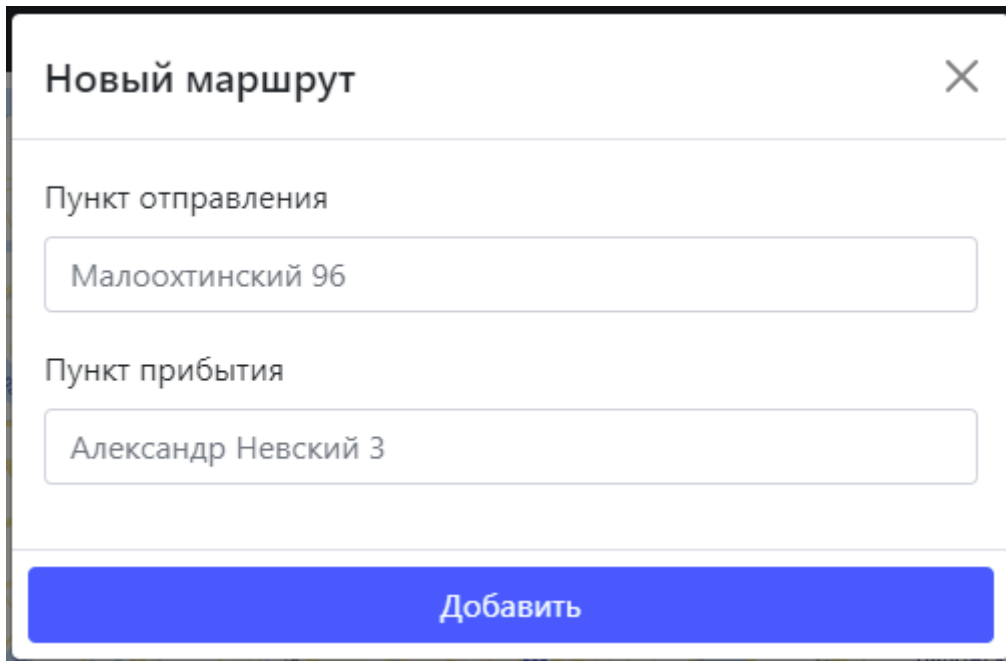


Рисунок 25 – Окно нового маршрута

Крайне важно уведомить систему о том, что доставка прошла безупречно, как только товар был успешно доставлен. Маршрут должен быть удален с карты после нажатия кнопки «Подтвердить доставку», а в базе данных будет сделана отметка о том, что маршрут пройден. (Рисунок 26)

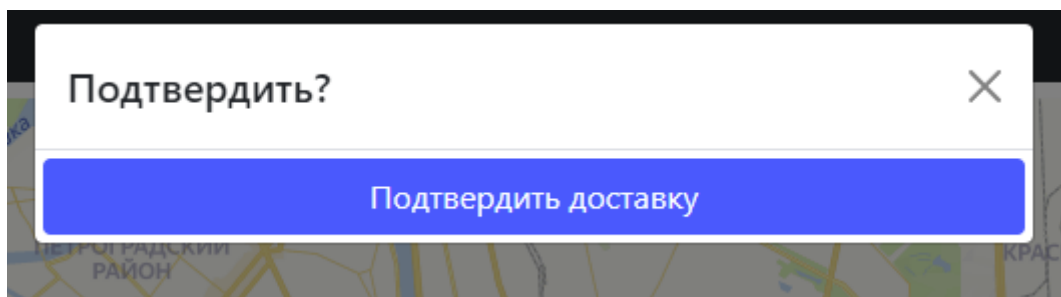


Рисунок 26 – Отслеживание доставки

На рис. 27 показан маршрут водителя.

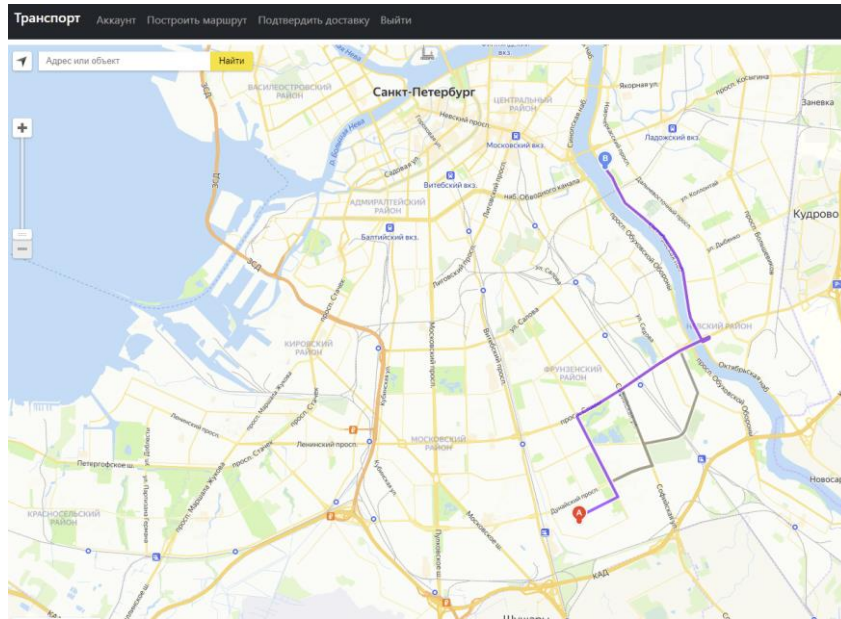


Рисунок 27 – Карта маршрута

Сама карта создана с использованием javascript и дизайнерского решения Яндекса. На рис. 28 показано фактическое применение карты.

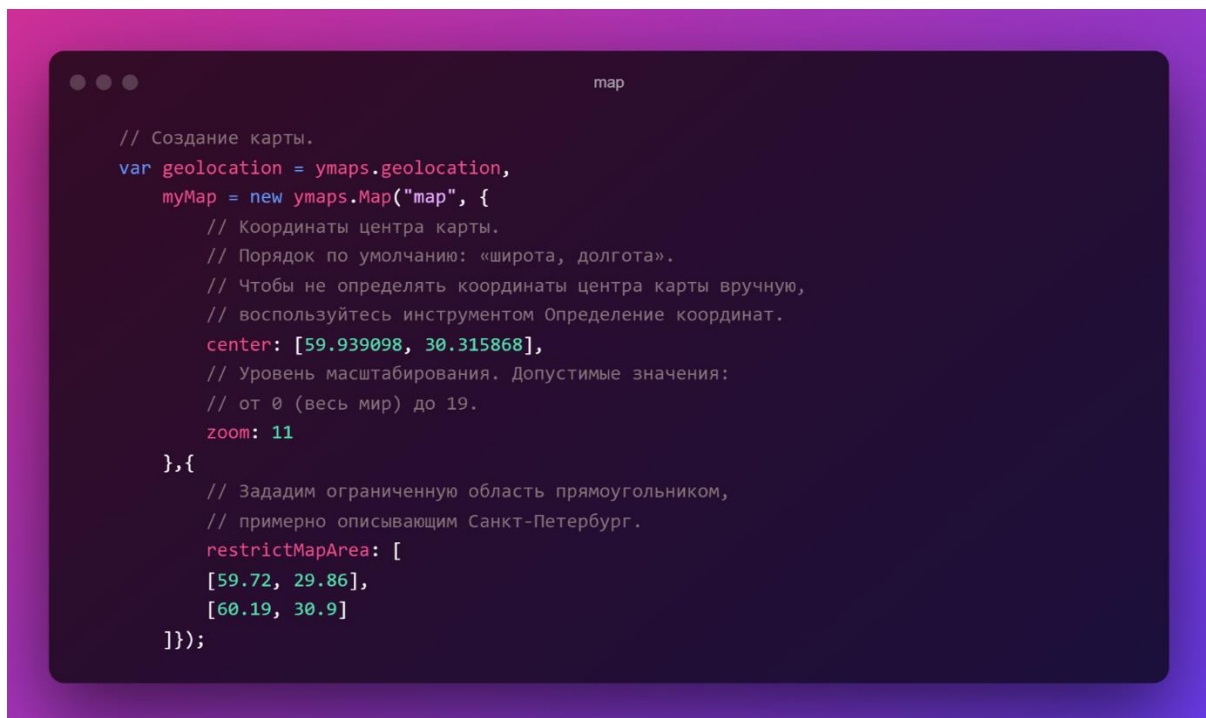


Рисунок 28 – Сделайте карту

Вам нужны две геоточки, чтобы установить маршрут, чтобы система могла вычислить путь. (Рисунок 29)

```

router

var multiRoute = new ymaps.multiRouter.MultiRoute({
  // Точки маршрута. Точки могут быть заданы как координатами, так и адресом.
  referencePoints: [
    [59.832285, 30.393959],
    [59.922251, 30.406939],
  ]
}, {
  // Автоматически устанавливать границы карты так,
  // чтобы маршрут был виден целиком.
  boundsAutoApply: true
});

// Добавление маршрута на карту.
myMap.geoObjects.add(multiRoute);

```

Рисунок 30 – Составление маршрута

3.3 Описание базы данных

Поскольку все данные хранятся в базе данных, она служит основой. Представленная реализация базы данных основана на той, что показана на диаграмме классов на рисунке 31.

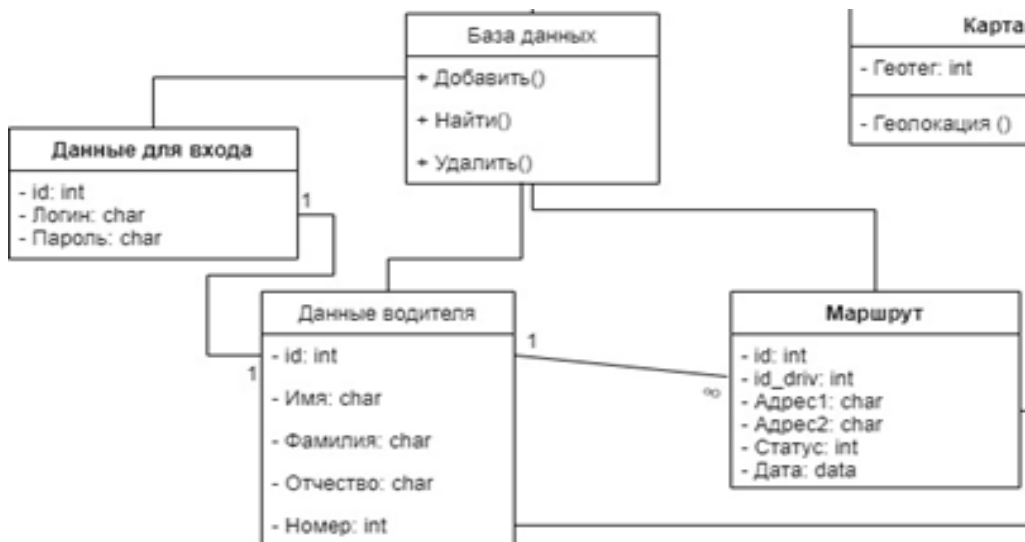


Рисунок 31 – Репозиторий ГИС

Все необходимые поля и таблицы присутствуют в базе. Для создания базы данных используется язык программирования MySQL.

Этот инструмент позволяет эффективно манипулировать базой данных.

ЗАКЛЮЧЕНИЕ

Использование автоматизированных рабочих инструментов в настоящее время проникло практически во все аспекты современной жизни и работы, от удаленных офисов и полностью автоматизированных предприятий до умных домов и персональных устройств.

Система управления плановыми перевозками с использованием ГИС-технологий является примером такого программного обеспечения, которое может значительно упростить и улучшить процесс автоматизации перевозок.

От предпроектной проработки до сопровождения готовой системы разработка ИС включает в себя все этапы создания системы.

В ходе реализации ВКР были выполнены следующие задачи:

Создал теоретические и прикладные исследования по теме следящих информационных систем.

Проанализированы исходные данные проекта; предлагает удобный интерфейс с удобной фильтрацией по категориям. проанализировали данные от работающих компаний, связанных с транспортом и сопоставимыми системами.

Соответствующие платформы для разработки были выбраны после изучения существующих платформ.

На выбранной платформе создано работающее веб-приложение, регулярно обновляется информация об услугах и возможностях приложения.

Вот какие выводы были сделаны на протяжении всей диссертации:

- Благодаря своей адаптивности среда разработки Visual Studio Code идеально подходит для создания веб-приложений.

- Благодаря простоте и удобству использования инструмент моделирования FlowCharts невероятно эффективен для рисования блок-схем и создания диаграмм.

СПИСОК ЛИТЕРАТУРЫ

1. Как ГИС помогает оптимизировать логистику. - Текст : электронный // itWeek : [сайт]. - URL: <https://www.itweek.ru/idea/article/detail.php?ID=167806> (дата обращения: 25.01.2022).
2. Геоинформационные системы в транспортной логистике. - Текст : электронный // Pandia : [сайт]. - URL: <https://pandia.ru/text/79/470/49624.php> (дата обращения: 12.01.2022).
3. Создание динамического сайта. - Текст : электронный // itProger : [сайт]. - URL: <https://itproger.com/course/php-website> (дата обращения: 20.12.2021).
4. Простое руководство по UML-диаграммам и моделированию баз данных. - Текст : электронный // Microsoft : [сайт]. - URL: <https://www.microsoft.com/ru-ru/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling> (дата обращения: 14.01.2022).
5. Новиков, Ф. А. Учебно-методическое пособие по дисциплине «Анализ и проектирование на UML» / Ф. А. Новиков. - 1-е изд. - Санкт-Петербург : ИТМО, 2007. - 286 с. - Текст : непосредственный.
6. Губенко В. К., Лямзин А. А. Городская логистика // ГВУЗ «Приазовский государственный технический университет». 2009. №19. URL: <https://cyberleninka.ru/article/n/gorodskaya-logistika> (дата обращения: 10.02.2022).
7. Карманов А.Г., Кнышев А.И., Елисеева В.В. Геоинформационные системы территориального управления: Учебное пособие – СПб: Университет ИТМО, 2015. – 121 с.

ПРИЛОЖЕНИЕ 1

Рисунки диаграмм последовательности. (Рисунок 1.1 - Рисунок 1.4)

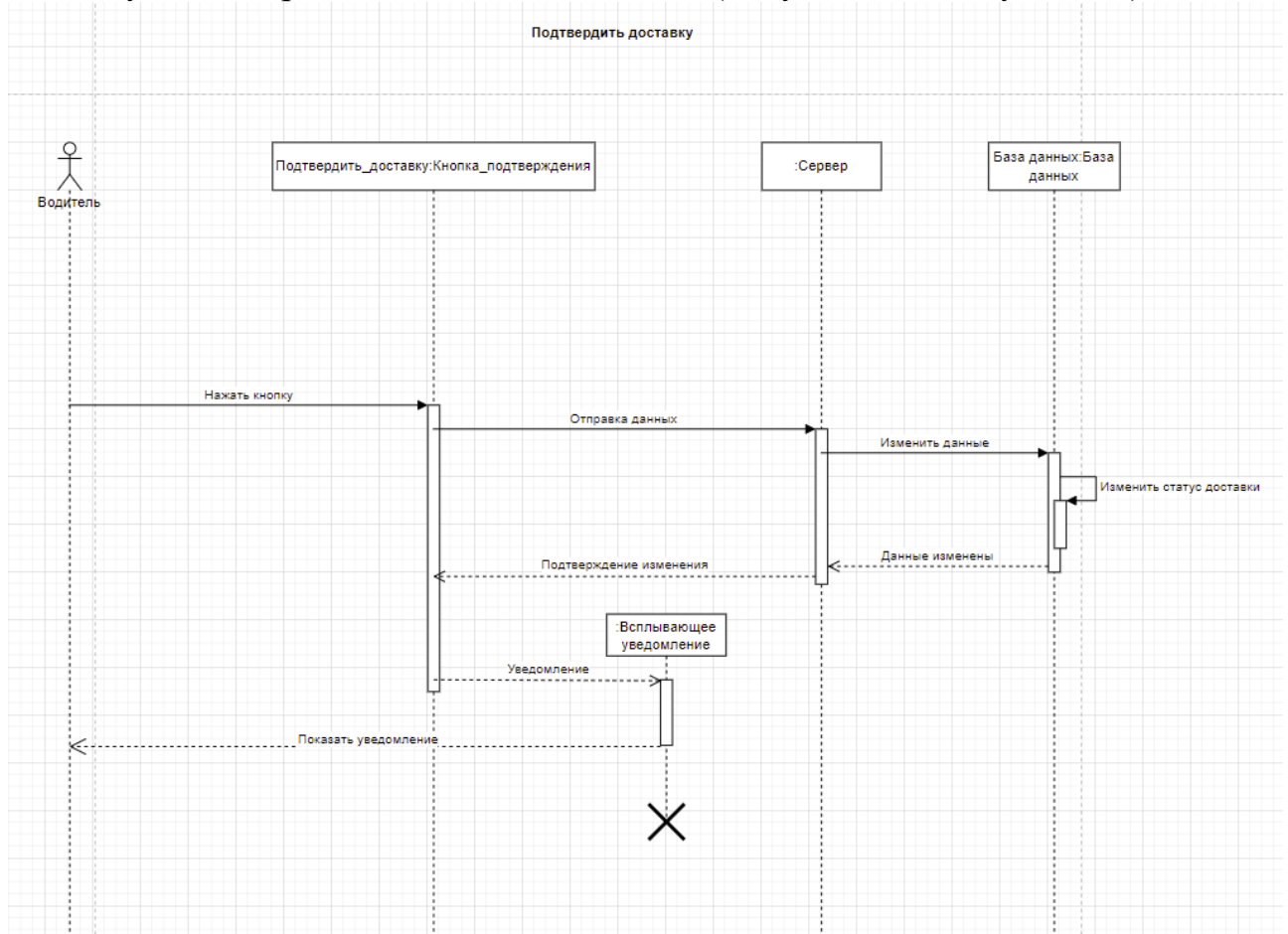


Рисунок 1.1 - Подтверждение доставки

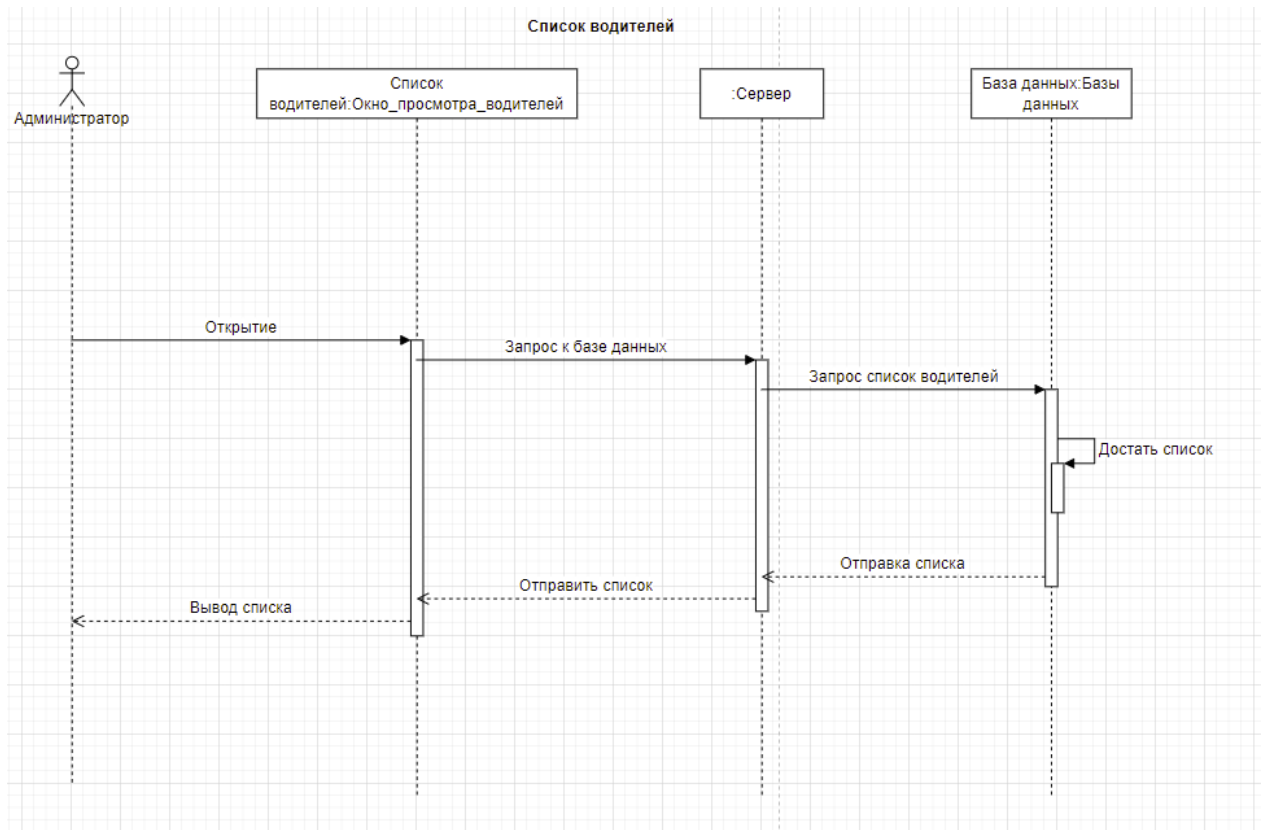


Рисунок 1.2 - Список водителей

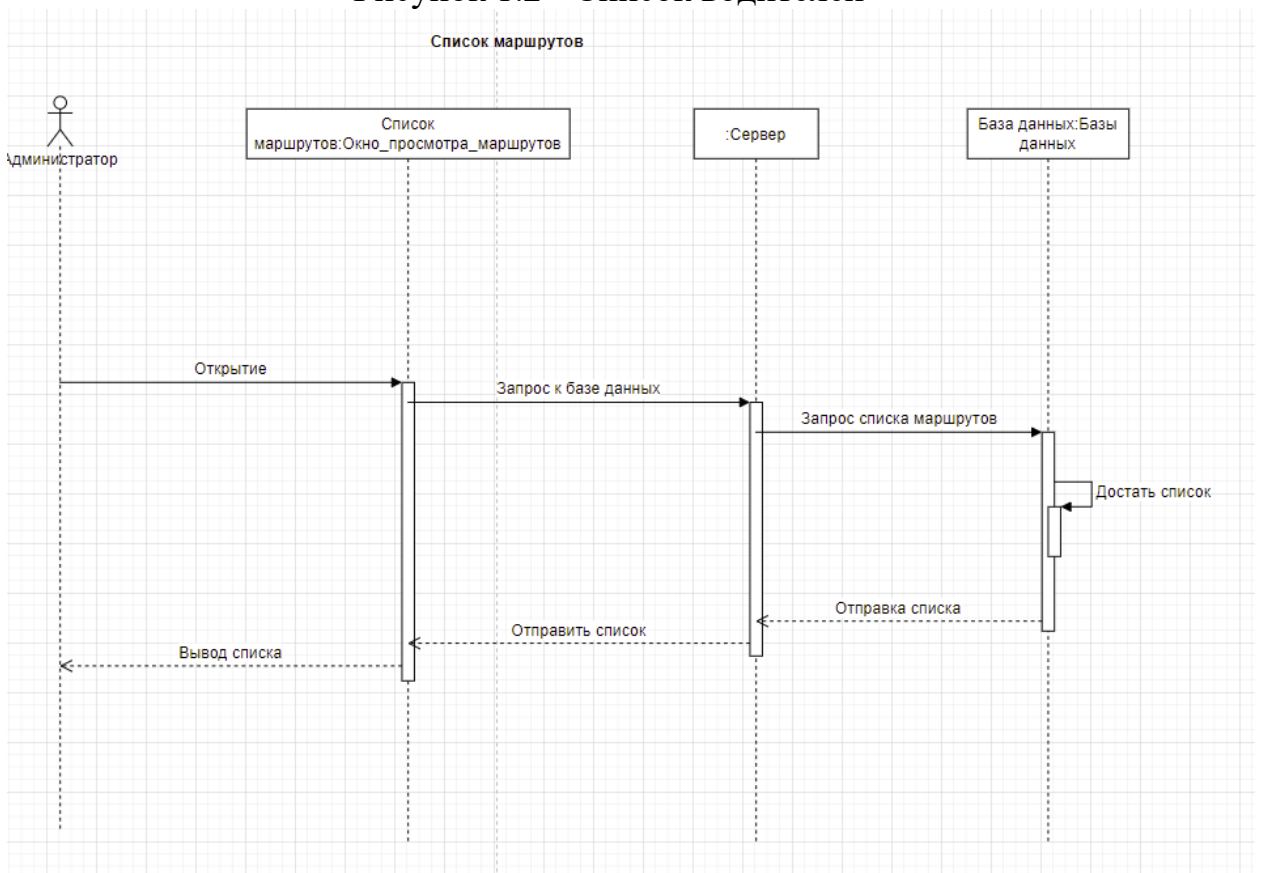


Рисунок 1.3 - Список маршрутов

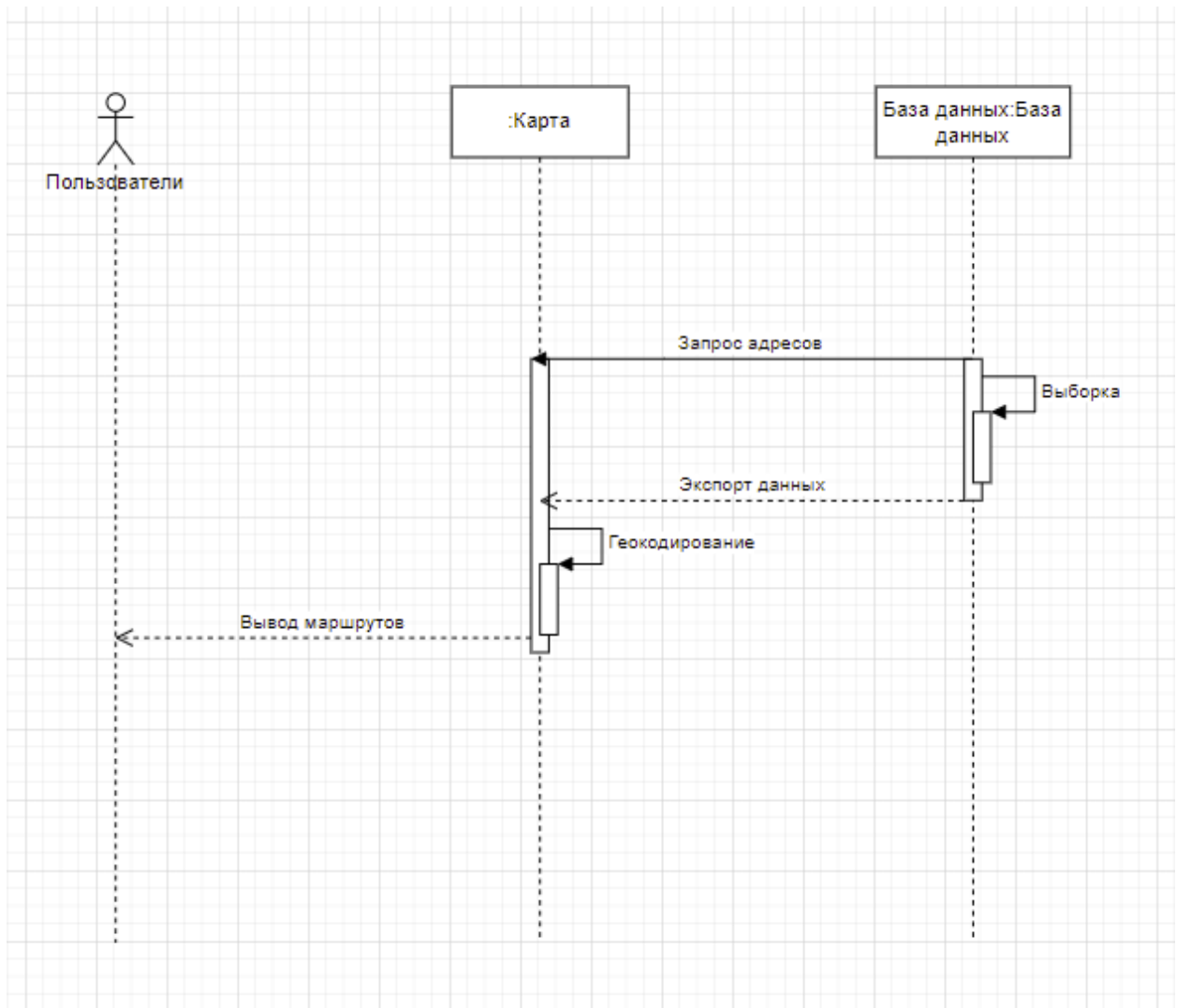


Рисунок 1.4 - Работа карты