



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 «Прикладная информатика»
(квалификация – бакалавр)

На тему «Проектирование информационной системы для управления
автотранспортным предприятием малого бизнеса»

Исполнитель Чернышов Максим Викторович

Руководитель к.г.н., доцент по кафедре информатики и прикладной математике
Полупанов Владимир Николаевич

«К защите допускаю»

Руководитель кафедры

кандидат экономических наук

Майборода Евгений Викторович

«10» 01 2025 г.

Филиал Российского государственного гидрометеорологического университета в г. Туапсе
НОРМОКОНТРОЛЬ ПРОЙДЕН
« <u>13</u> » <u>01</u> 20 <u>25</u> г.
 <u>Мартьянова Т.В.</u>
Подпись

Туапсе
2025

ОГЛАВЛЕНИЕ

Введение.....	3
1 Аналитическая часть.....	5
1.1 Анализ объекта и предмета исследования.....	5
1.2 Выбор технологии обмена информацией.....	14
1.3 Выбор средств работы с геоданными.....	18
1.4 Архитектура информационной системы.....	24
2 Проектная часть.....	33
2.1 Методология и методика проектирования.....	33
2.2 Особенности проектирования на платформе Node.....	43
2.3 Основные элементы информационной системы.....	48
2.4 Проектирование интерфейса пользователя.....	56
3 Экономическая эффективность проекта.....	65
3.1 Обоснование методики расчета экономической эффективности.....	65
3.2 Эффективность разработки.....	67
Заключение.....	75
Список литературы.....	79
Приложение.....	82

Введение

Текущее состояние предприятий транспортного комплекса показывает, что успешная работа и развитие рынка перевозок невозможны без обновления производственных процессов и внедрения современных технологий. Это особенно важно для организации движения, управления автопарком, повышения качества обслуживания, а также для реализации комплексной системы безопасности и других связанных аспектов.

Большая часть любой информации, относительно объектов, содержит в себе геоданные, в которых отражены определенные сведения о распределении объектов или процессов непосредственно на определенной местности.

Такая информация сугубо необходима для предприятий транспортной сферы по причине того, что обладание такого рода сведениями дают полную картину по определению координат, быстрого просмотра и принятию решений по сложным вопросам. Получение и обработка геоданных проходит путем считывания на ГИС[6,с.33].

Находясь в постоянном контакте с пользователями программных продуктов, ведущие разработчики ESRI пришли к решению о необходимости создания единой масштабируемой архитектуры для программного обеспечения. До сих пор программные продукты ESRI были совместимы на уровне данных, и построены на основе разных программных кодов, которые поддерживали и развивали разные команды разработчиков.

Актуальность работы обусловлена недостаточностью исследований по выбранной тематике в связи с появлением новых веб- и ГИС-технологий, которые могли бы повысить эффективность управления автотранспортом небольших предприятий.

При этом значительный эффект может быть достигнут уже за счет повышения качества управления имеющимися активами и более тщательного планирования инвестиций в развитие.

Объект исследований – автотранспортное предприятие ООО

«Транспортная компания» г. Москва

Предмет исследования - информационная система на основе открытых веб- и ГИС-технологий

Цель выпускной квалификационной работы - проектирование информационной системы на основе открытых веб-ГИС- технологий для управления автотранспортом небольшого предприятия, занимающегося перевозкой грузов.

Для достижения поставленной в работе цели определен ряд задач:

- провести анализ объекта и предмета исследования;
- изучить средства открытых веб-технологий обмена информацией и технологий ГИС для использования в проектируемой информационной системе;
- разработать проект информационной системы на основе открытых веб- и ГИС-технологий с учётом особенностей функционирования небольшого автопредприятия;
- оценить экономическую значимость предлагаемого проекта;
- подвести итоги выполненной работы с кратким изложением преимуществ использования новых средств, практической значимости проектируемой информационной системы, возможных проблем и путей их решения.

Методы решения главной задачи работы – экспериментальный, методы системного анализа, моделирования, нормализации данных, теоретический. Метод исследования - изучение текущего состояния бизнес-процессов в сфере проектирования геоинформационной системы, изучение литературы в области информационных технологий, CASE-систем и языков программирования.

Теоретико-методологической основой исследования послужили работы отечественных и зарубежных ученых в области геоинформатики, учебные пособия, монографии, научные статьи, ресурсы сети Интернет.

1 Аналитическая часть

1.1 Анализ объекта и предмета исследования

За объект исследования было выбрано предприятие ООО «Транспортная компания», осуществляющее автомобильные грузовые перевозки по Москве, Московской области и в другие регионы России.

Услуги ООО «Транспортная компания», помимо автоперевозок, включают в себя страхование и экспедирование грузов, а также оформление полного пакета документов (ТТН, счёт-фактуры и др).

На данный момент транспортная компания работает на рынке грузоперевозок уже больше 15 лет. Накопленный опыт позволяет заниматься одним из сложнейших видов автомобильных перевозок - перевозкой крупногабаритных и тяжеловесных грузов, в том числе и проектными перевозками. На сегодняшний день транспортная компания доставляет грузы по всей территории РФ (включая Крым) и территории Казахстана.

Вся используемая техника в количестве 35 единиц является собственностью компании. В 2018 году был обновлен модельный ряд, и сегодня более 50 % автопарка – это современная многофункциональная новая техника.

Деятельность компании сосредоточена на организации перевозок и сопровождении грузов и это является основным видом работы. Имея богатый опыт, транспортное предприятие постоянно увеличивает спектр предлагаемых услуг.

Парк специализированных автопоездов позволяет оказывать услуги по транспортировке грузов массой до 50 тонн и длиной до 40 метров на низкорамных полуприцепах (тралах) различных типов.

Автопарк компании, отвечающий самым высоким требованиям, представлен ведущими иностранными и российскими производителями: Газель (до 2 т); Валдай (до 3,5 т); MAN (до 5 т); Scania (до 10 т); Mercedes (до 20 т).

Линейно-функциональной моделью представлена организационная

структура управления. Линейные полномочия подразумевают передачу прав от руководителя к подчиненному, а затем к другим сотрудникам. Руководитель, обладающий такими полномочиями, имеет право принимать определенные решения без необходимости согласования с другими руководителями, в рамках, установленных организацией, законодательством и обычаями. Графическое представление линейно-функциональной структуры ООО «Транспортная компания» (рисунок 1.1). Сочетание функциональности и линейности позволяет сгенерировать универсальную организационную структуру, особенностью которой является то, что она позволяет в значительной степени устранить недостатки как линейного, так и функционального управления.



Рисунок 1.1 – Организационная структура ООО «Транспортная компания»

Отсюда наглядно видно, что при такой структуре управления, остро стоит вопрос о межфункциональной координации. Общее количество персонала на начало 2023 года составляло 57 человек.

По конкретным пожеланиям заказчиков и клиентов формируется персональная схема заказов на доставку специфичного груза или товара, на основании которой груз доставляется, не зависимо от сложности и удаленности.

Помимо осуществления непосредственной перевозки, ООО «Транспортная компания» оказывает полный спектр сопроводительных услуг:

- оформление разрешительной документации;
- осуществление погрузки - разгрузки;
- организация сопровождения автомобилями прикрытия и ГИБДД;
- изготовление не стандартизированной оснастки и оборудования;
- разработка спецпроекта и обеспечение страхования груза.

Для обеспечения качества и эффективности системы контроля необходимо проводить проверки и мониторинг, осуществляя постоянные или периодические оценки.

В этой связи важно перейти от реактивного управления, при котором управленческие решения принимаются в ответ на текущие проблемы, к управлению, основанному на анализе и прогнозировании.

При проведении анализа по состоянию деятельности транспортной компании была обнаружена такая проблема как отсутствие мониторинга за состоянием списания топливного расхода, а так же исключения из деятельности автоматизации за ходом выполнения рейсов.

На сегодняшний день весь учет ведется за счет составления таблиц в офисной программе Excel, в которой выстраивается информация по нормам топливного использования и соответственно, акта на списание остатков.

Из приведенных таблиц были получены фактические данные о списании топлива для различных емкостей.

Эти данные позволяют вести контроль за расходом топлива и

денежных средств на него.

Учет списания топлива ведется на основании накопительных ведомостей с данными путевых листов за отчетный период. Ответственным лицом (главным механиком) производится сверка данных о фактическом сливе топлива и составляется отчет.

После этого, ведомости вместе с отчетом сдаются в бухгалтерию для дальнейшей обработки. Такие отчеты обычно сдаются раз в смену.

Возвращаясь с рейсов каждый из водителей предоставляет путевые документы, в которых прослеживается отчетность перед диспетчерской службой относительно израсходованного в пути следования топлива, что в последствие заносится в единую отчетность по предприятию.

Данный алгоритм контроля и учета всех расходов по ГСМ фактически не отражает действительной картины всего слитого топлива и не фиксирует каждую единицу транспорта в отдельности.

Получается, что заправка и слив в течение всей командировки происходит нецелевым способом.

Сама учетность носит характер нерациональности и искажает полную картину всей информативности.

Форма, по которой работают на предприятии давно устарела и учет топлива согласно нормативов, установленных на самом предприятии не эффективен.

Попросту говоря, его нет и поэтому ощутима острая и глобальная нехватка информации касательно как километража по каждой автотранспортной единице, так и об объемах заправленного топлива.

Более того, назрела потребность в слежении за транспортом во время пути следования, дабы исключить сложные транспортные ситуации.

Диспетчер не имеет возможности вести учет движения транспортной единицы на холостом ходу, что предоставляет водителям прекрасную возможность искажения информации о действительном расходе горюче-смазочных средств.

Более того, нет четкого определения координат при передвижении транспортных средств и отображение автотранспорта на карте, корректировки маршрута и отклонениями от него, а так же мониторинга скоростного режима. Эти отчеты могут выгружаться в систему CRM (Customer Relationship Management), которая в случае несанкционированного слива горючего, автоматически отправит СМС-сообщение с оповещением и отразит факты в отчете.

Для контроля за состоянием топлива на всём отрезке пути можно использовать системы спутникового мониторинга. Они позволяют отслеживать расход топлива в режиме реального времени, а также получать информацию о местоположении транспорта и его движении.

Идеальным, в данном случае решением, для непрерывным контроля за транспортом для ООО «Транспортная компания» видится установка дополнительного датчика слежения за уровнем топливного средства на протяжении всей поездки с учетом прямого подключения каждого транспортного средства к системе GPS-мониторинга и контроля топлива.

Централизованная технология обработки данных, основанная на кластовых информационно-вычислительных центрах и существовавшая до начала 90-х годов, перестала удовлетворять потребности автотранспортных предприятий в условиях рынка и была ликвидирована.

Современные информационные технологии АТП основываются на следующих основных принципах [14,с.59]:

- интерактивный (диалоговый) режим работы с ПК;
- интегрированность с другими программными продуктами;
- гибкость процесса постановок задач, изменения данных, реагирование на внешние и внутренние изменения АТП.

Существует множество программ для управления информационными потоками, обработки информации на автотранспортном предприятии.

Примеры таких программ и их характеристика представлены в таблице 1.1.

Таблица 1.1 - Программы для обработки информации и управления на автотранспортном предприятии

Программа	Характеристика
Автоперевозки	Программа для учета автотранспортной техники собственного парка, документов, связанных с учетом, формирования и печати путевых листов, учета выполненных ремонтных работ и ТО, учета ГСМ, работы водителей, учета контрагентов и работы с ними, складского учета.
Умная логистика	Онлайн сервис для оперативной работы транспортно-экспедиционных компаний. Создавайте заявки на перевозки, упорядочите документы, контролируйте работу менеджеров-логистов, отслеживайте рентабельность компании.
1С Форес: Автотранспорт	Конфигурация 1С для учета автотранспорта. Программа позволяет осуществлять учет ГСМ, путевых листов, осуществлять контроль за регламентными работами, осуществлять диспетчерские функции, контролировать наличие и актуальность документов водителей. Хранение автомобилей осуществляется в автопарках и боксах, контролируется наличие автомобилей в боксах.
NovaTrans	Онлайн сервис для автоматизации всех процессов транспортной компании. Позволяет создавать и контролировать заявки, ведения актов и т.д.

Для того, чтобы в конечном итоге выбрать и установить конкретную программу, необходимо изучить ряд принципов и методов по руководству:

— системность обслуживания, учет характера деятельности потребителей, решаемых ими задач при управлении транспортно-логистическими процессами, качественном удовлетворении информационных потребностей;

— надежность обслуживания, что предполагает обеспечение информацией менеджеров и участников транспортно-логистических цепочек в нужные сроки и в наиболее удобном для них виде;

— полнота информационного обслуживания выполняемых процессов (операций) и доведение необходимой информации до конкретного потребителя;

— дифференцированность, состоящая в том, что каждый потребитель

индивидуально обеспечивается информацией, которая способствует решению поставленных задач.

Три основных принципа новой информационной технологии:

- интерактивный (диалоговый) режим работы с компьютером;
- интегрированность (стыковка, взаимосвязь) с другими программными продуктами;
- гибкость процесса изменения, как данных, так и постановок задач.

В качестве инструментария можно использовать следующие распространенные виды программных продуктов для ПК: текстовый процессор (редактор), настольные издательские системы, электронные таблицы, системы управления базами данных (СУБД), электронные записные книжки, электронные календари, информационные системы функционального назначения (финансовые, бухгалтерские, для маркетинга и пр.), экспертные системы и т.д. При внедрении новой информационной системы в организации необходимо оценить риск отставания от конкурентов в результате ее неизбежного устаревания со временем, так как информационные продукты имеют чрезвычайно высокую скорость сменяемости новыми видами или версиями. Процессы управления на автотранспортном предприятии (АТП) осуществляются циклически. Цикл управления начинается со сбора информации о состоянии управляемого объекта (АТП, цех, участок и т.п.), затем полученная информация анализируется и используется для принятия решений [17,с.24].

В первую очередь оценивается риск реального увеличения конкурентных преимуществ перед основными соперниками при разработке новой концепции системного продукта, что позволит избежать старения самой системы, так как смена версий и видовой модификации наступает очень скоро. И тут необходимо знать, что все производственные процессы, с учетом специфики деятельности транспортной сферы имеют циклический характер, что позволяет определять жизненные позиции каждого элемента. Виток развития управленческой позиции конкретного структурного подразделения, будь то

АТП или механизированный участок, следует начинать со сбора сведений, что дает основание анализа принятия решений по существу вопроса управленческих изменений [17,с.24].

Производственный учет отражает деятельность предприятия путем фиксации технических, экономических и других показателей (например, конкретные значения расхода топлива, плановых и фактических периодичностей ТО, наработок на отказ и пр.).

Сведения о выполненных производственно-хозяйственных операциях фиксируются на первичных бумажных или электронных носителях информации в виде натуральных, стоимостных или иных показателей. АТП представляет собой совокупность производственных подразделений (автоколонны, зоны ТО и ремонта, склады, участки) и служб (бухгалтерия, плановый отдел, технический отдел и т.д.), в каждом из которых решается определенный круг задач. Все подразделения АТП можно разделить на две части - выполняющие свои функции на территории и за пределами территории предприятия. Основные информационные потоки АТП формируются за счет основной деятельности каждого структурного подразделения (рисунок 1.2).

Работа подразделений и сотрудников документируется в различных формах, таких как табели учета рабочего времени, наряды на выходы на линию, путевые и ремонтные листы, а также заявки на получение запасных частей и другие документы.



Рисунок 1.2 – Схема информационных потоков управления автотранспортным предприятием

Представление самой отчетности и определенных сверок охватывают результаты деятельности организации, которые получают в ходе анализа результатов по состоянию на текущий период. Иными словами, сведения предоставляют все отделы и цеха транспортной компании и которые отражают работу всего штатного состава данного предприятия.

Результаты работы подразделений и сотрудников фиксируются в различных документах, таких как путевые листы, акты ремонта, ведомости по выдаче топлива, заявки на запасные части и другие.

Следовательно, основная цель информационного обеспечения автотранспортного предприятия состоит в обеспечении эффективного управления, контроля и комплексного планирования движения транспортно-материальных потоков.

1.2 Выбор технологии обмена информацией

Процесс автоматизации можно осуществить за счет открытых веб и ГИС технологий, что позволит покрыть территориальную распределенность и учесть индивидуальные особенности многих автотранспортных компаний.

Именно пространственный фактор является своего рода платформой для интеграции по причине своей естественности формирования транспортной инфраструктуры, решения автотранспортных задач и многих вариантов навигации.

Инновационные разработки в виде географических информационных систем позволяют создавать цифровые модели предприятий, которые значительно превосходят традиционные бумажные планы по своим функциональным возможностям. В настоящее время разработаны новые инструменты для контроля и планирования работы автотранспортных компаний.

Автоматизированные системы мониторинга автотранспорта, доступные широкому кругу пользователей, способны решать разнообразные задачи в

реальном времени. Теперь возможно сопоставление маршрутных листов с фактическим маршрутом, отображаемым на географической карте, а также получение отчетов с указанием точек маршрута или полного списка пройденных адресов.

Это позволяет легко выявлять случаи нецелевого использования транспортных средств, принадлежащих компании, а также случаи кражи и повреждения груза или топлива. Достичь этих и других возможностей можно благодаря проектированию и внедрению геоинформационных систем для контроля транспорта, таких как глобальные навигационные системы «WEB-GPS/GSM-Глонасс/GSM». [20,с.28].

Значительным прорывом стала открытая связь с водителем в пути следования, а именно и голосовой и видео контроль, что облегчает жизнь и водителю, и диспетчерской службе. Кроме того, сотрудник самого предприятия получает постоянные уведомления о расходе топлива и о состоянии транспортного средства.

Концептуально отслеживание транспортных средств осуществляется следующим образом [29,с.66]:

1) Трекер, установленный в автомобиле, получает сигнал от одной или нескольких навигационных систем, не только спутниковых, но и сервисов глобальной беспроводной сети. Для этого устройство должно иметь ГЛОНАСС/GPS/LBS-модули, которые автоматически вычисляют координаты объекта по расположению спутников и ближайших вышек сотовой связи.

2) Телеметрические и другие данные (например, аудиозапись, событие «авария») собираются в памяти трекера. Затем информация в установленные интервалы времени пакетами передается по каналам GPRS через Интернет на телематический сервер со специальным ПО и/или при помощи смс-уведомлений на мобильное устройство собственника.

За счет цифровых сообщений с определенными координатами простейшие бортовые устройства передают по выбранным каналам связи, что в настоящее время не представляет никаких трудностей.

Что касается более продвинутых устройств, то за последние несколько лет были освоены и активно применяются телеметрические параметры груза или транспорта, за счет получения которых, предприятием ведет автономную запись данных в БД и более этого строит диалог самого водителя и диспетчера.

В настоящее время любое предприятие имеет возможность правы выбора того или иного программного продукта в сфере мониторинга транспортных средств своего автопарка и контроля за формированием и доставкой груза, причем производитель может быть как отечественный, так и зарубежный.

Координаты, передаваемые бортовыми устройствами, в конечном итоге формируют базу данных единого локального характера, а те, траектории которые база формирует отображаются в виде картографического описания (рисунок 1.3).

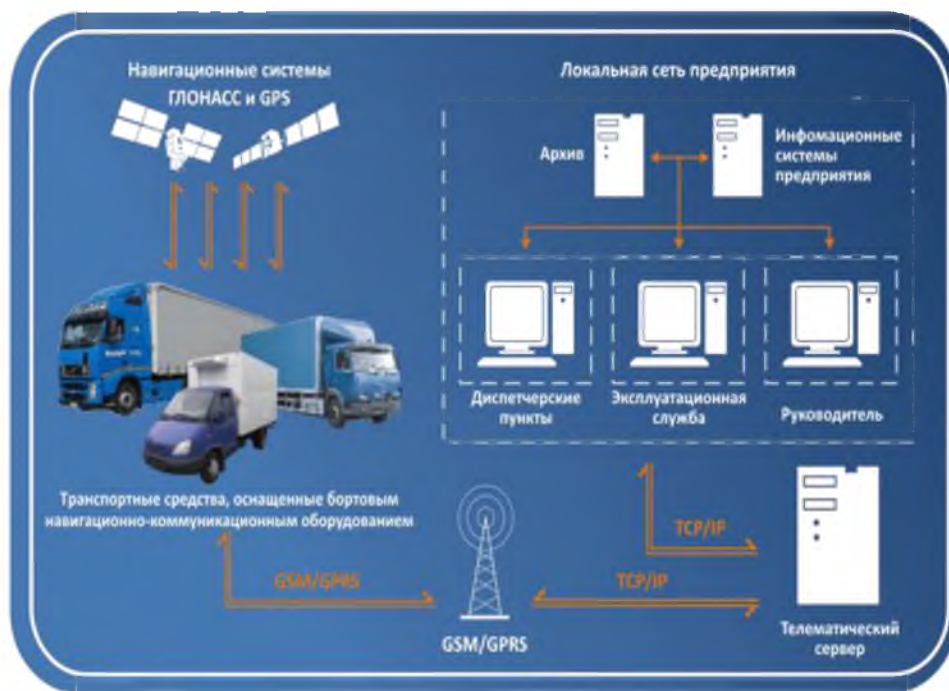


Рисунок 1.3 - Схема передачи сигнала

Шестиканальный приемник GPS-сигналов с внешней антенной использует простой ASCII-последовательный протокол и позволяет обеспечить непрерывность работы следящей системы в городских условиях. Такие данные о транспортном средстве, как широта, долгота, высота, скорость, условный номер средства и системное время могут выдаваться приемником, по крайней

мере, один раз в секунду. Различного рода стратегические направления могут осуществляться благодаря точному определению временных позиций за счет передачи сообщений либо на основании запроса, либо по установленному графику.

Информация о передвижении машины поступает на пульт через равные Временные интервалы, что выстраивает ровное выстраивание маршрута. Все незапланированные помехи в передвижении сама программа отслеживает сама и фиксирует в информационный журнал связи, поэтому сама диспетчерская группа в любой момент, не зависимо от географического месторасположения отвечает как грузоотправителю, так и получателю о состоянии товара в пути следования и времени прибытия к месту адресата. Система исключает все санкционные рейсы, не входящие в маршрут следования, а значит исключает дополнительный расход топлива или его слив. [22].

Принцип работы системы мониторинга транспорта ГЛОНАСС состоит в следующем:

- вначале определяется местоположение транспортного средства с помощью идентифицирующей способности GPS связываться через сотовую сеть с центром данных;

- далее благодаря сотовой связи трекер каждые 5-10 секунд сообщает о своем местоположении в реальном режиме времени. Помимо этого, данные, поступающие на терминал, обрабатываются;

- после чего вся приобретенная информация отправляется на сервер, на котором установлено специальное программное обеспечение. Они могут быть доступны с любого компьютера или мобильного телефона.

В результате, подключение мониторинга автотранспорта позволит оперативно наблюдать за работой собственного транспортного средства и сотрудниками, а также максимально быстро реагировать на различные внештатные ситуации и контролировать расход топлива в баке (рисунок 1.4).



Рисунок 1.4 - Бортовое навигационное оборудование ГЛОНАСС

Сегодня на рынке присутствуют десятки систем мониторинга и сотни моделей ГЛОНАСС - трекеров, многие из которых не соответствуют даже минимальным требованиям к уровню качества и надежности для коммерческой эксплуатации.

Множество образцов оборудования не имеют даже необходимой сертификации, и их использование на транспорте не может быть полностью безопасным. Резюмируя к вышесказанному, особенно важно выбрать качественную и эффективную систему мониторинга транспорта, использование которой будет приносить только положительный результат для транспортной компании, а не создавать проблемы частыми выходами из строя оборудования и спорами о достоверности данных.

1.3 Выбор средств работы с геоданными

При выборе средств для работы с геоданными стоит определить, какие задачи нужно решать. Это поможет сузить круг поиска и выбрать наиболее подходящее решение. Также стоит изучить функциональные возможности различных программ, проверить совместимость с данными и оборудованием, оценить стоимость и отзывы пользователей.

Выделим несколько средств работы с геоданными [23]:

— ГИС (геоинформационные системы). Комплекс программных и аппаратных средств для сбора, хранения, анализа и визуализации геоданных. Примеры: ArcGIS, QGIS, Mapinfo;

— Библиотеки и фреймворки. Программные библиотеки и фреймворки, предоставляющие инструменты для работы с геоданными в различных языках программирования. Примеры: GDAL/OGR, GeoPandas, Leaflet;

— Веб-сервисы и API. Онлайн-платформы и программные интерфейсы, позволяющие работать с геоданными через интернет. Примеры: GoogleMaps API, CartoDB, Mapbox.

Геоданные в контексте систем мониторинга включают в себя пространственную составляющую (географическое местоположение). Для работы с такими данными используются геоинформационные системы (ГИС). Также для мониторинга могут использоваться системы геопозиционирования, которые позволяют определять координаты объекта при движении по земле, воде или воздуху, а также скорость и направление движения объекта, оснащенного трекером или навигатором.

Система, оснащенная спутниковым навигационным оборудованием получила название системы мониторинга транспорта, что значительно упростило работу по обеспечению безопасности автопарка, а так же автоматизировало диспетчеризацию на расстоянии.

В режиме реального времени за счет применения беспроводной связи и увеличения зон покрытия телекоммуникационных сетей, сервисное оборудование получает данные по конкретным запросам. В тех компаниях, которые уже работают или пытаются перейти на GPS мониторинг транспортных средств, программа зачастую интегрирует с единой системой мониторинга транспорта, системы зачастую интегрируются с системами управления ресурсами предприятий, ERP-системами, и обычно состоят из следующих сервисов [25,с.81]:

— удаленный мониторинг транспортного средства и диагностика основных систем автомобиля на основе систем спутниковой навигации и датчиков для увеличения эффективности деятельности автопарка;

— системы диспетчеризации транспорта – удаленный мониторинг транспорта и обеспечение связи с водителями, навигация, оптимизация маршрутов, логистика, оповещение об определенных событиях.

Большинство российских компаний, осуществляющих перевозки, рано или поздно сталкивается с проблемой злоупотреблений со стороны работников, непосредственно эксплуатирующих технику. Это и «сливы топлива», и «левые рейсы», и «приписка пробега». Все эти и многие другие проблемы (контроль передвижения транспорта в реальном времени, оптимизация транспортных схем, создание полной базы данных по работе автотранспорта) поможет решить система мониторинга на основе GPS.

Рассмотрим наиболее стационарное оборудование, в которое входит:

— датчики и бортовые компьютерные системы контроля состояния транспортного средства, управление этим состоянием;

— датчики прохождения транспортных средств через контрольные зоны с фиксацией характерных признаков;

— бортовые модули навигации и связи - блок навигатора с приемником и вычислителем, радиопередатчик; радиоприемник;

— навигационные спутниковые системы, которые обеспечивают определение на электронной карте местности местоположения транспортных средств с точностью до 10 м;

— сотовые и спутниковые системы связи и обмена информацией между логистической системой и транспортным средством;

— компьютерная обработка больших массивов данных в центрах управления перевозочным процессом или дорожным движением;

— нейροкомпьютерные технологии распознавания образов;

— датчики транспортного потока на магистралях, отражающие оперативную информацию и предоставляют информацию в реальном времени

об отдельных параметрах транспортных потоков или об отдельных транспортных средства.

Мониторинг на автомобильном транспорте имеет определенную историю развития. Сначала это были диагностические стенды и комплексы, проводившие опрос долговых датчиков для поиска неисправностей. Для стандартизации средств диагностики еще в 1980 году фирма GeneralMotors реализовала фирменный интерфейс ALDL (AssemblyLineDiagnosticLink) и протокол для тестирования модулей управления двигателями (ECM). Протокол ALDL взаимодействовал при скорости 160 бит/с и следил за состоянием систем автомобиля [10].

Следующий протокол бортовой диагностики OBD-II (On-BoardDiagnostic) предоставляет полный контроль за двигателем, позволяет проводить мониторинг частей кузова и дополнительных устройств, а также диагностирует сеть управления автомобилем. В этом стандарте производители применяют различные протоколы соединения с автомобилем. Есть пять диагностических протоколов, которые регламентированы в OBD-II. Каждый из OBD-II кодов неисправностей состоит из пяти символов - буквы и четырех цифр.

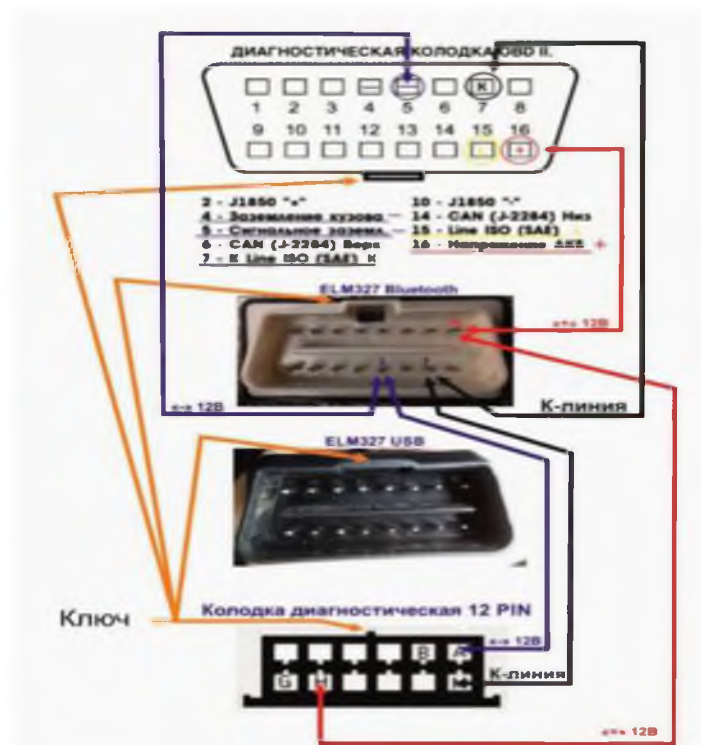


Рисунок 1.4 – Схема протокола бортовой диагностики OBD-II

Исключение составляет несколько видов транспорта, на котором устанавливается и реализует свои функции протокол модификации типа SAE J1962, что позволяет определять соответствие выводов на сам разъем и диагностической картой, как показано на рисунке 1.4.

Далее рассмотрим P2P-технологии, иными словами технологию построения сети распределенных равноправных узлов по принципу децентрализации (рисунок 1.5).

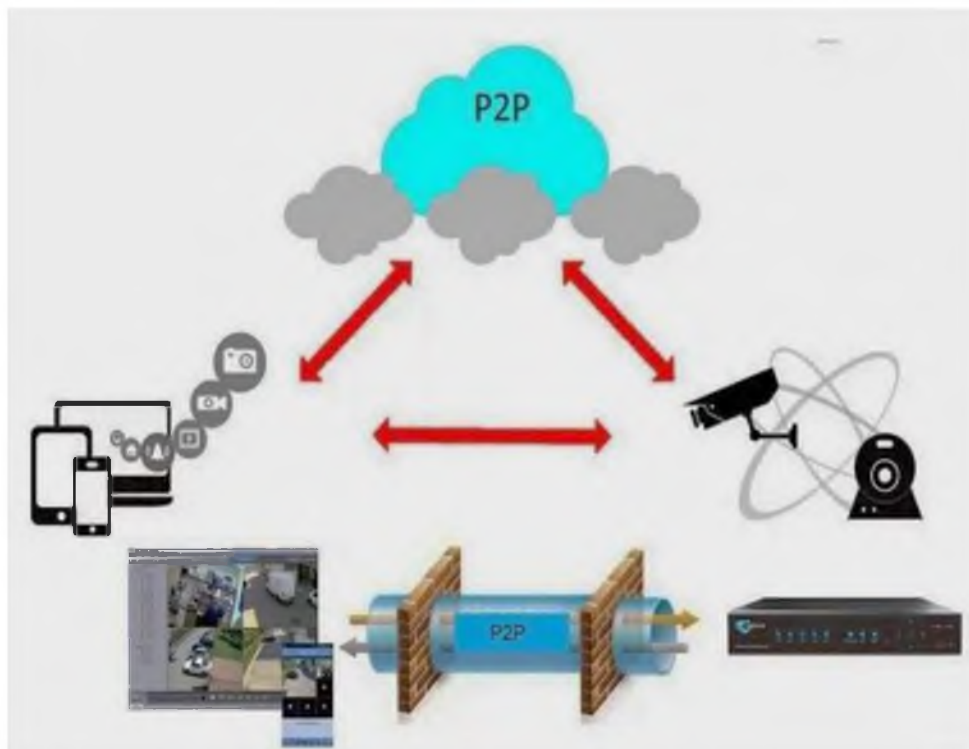


Рисунок 1.5 – Схема P2P-технологии

Совместные ГИС-приложения, основанные на P2P, создают платформы, где исследователи и аналитики могут совместно проводить пространственный анализ, выявляя закономерности распределения и взаимосвязи географических явлений. Такие инструменты становятся неоценимыми в самых разных сферах — от экологии до социологии.

Они позволяют не только улучшить качество принимаемых решений на основе пространственных данных, но и способствуют активному вовлечению сообщества в процесс исследования и принятия решений.

Таким образом, синергия ГИС и P2P-технологий открывает

новые возможности для инновационного подхода к пониманию и управлению пространственной информацией.

Также P2P-технология позволяет интегрировать веб-сервисы ГИС для предоставления размещённых пространственных данных и функциональности ГИС, что даёт возможность выполнять базовые задачи геообработки, такие как сопоставление адресов, отображение изображений карт и маршрутизация, без обслуживания инструментов ГИС или связанных географических данных.

P2P, представляя собой мощный механизм распределенной сети, облегчает взаимодействие между узлами, обеспечивая гибкость и надежность в условиях динамичного подключения и отключения.

Так как каждый узел сети выполняет функцию как клиента, так и сервера, протоколы P2P учитывают особенности маршрутизации и коррекции ошибок, создавая эффективно работающую экосистему.

Использование технологий, таких как хэш-таблицы и децентрализованные индексы, способствует быстрому поиску и обмену данными, минимизируя нагрузку на центральные серверы.

Примеры применения P2P технологий весьма разнообразны — от обмена файлами и потокового видео до децентрализованных финансовых систем и блокчейна. Важно отметить, что из-за своей природы P2P сети могут столкнуться с проблемами безопасности и анонимности, что подчеркивает необходимость разработки продвинутых механизмов шифрования и аутентификации.

Таким образом, протоколы P2P продолжают развиваться, адаптируясь к новым вызовам и расширяя свои возможности, что делает их важным элементом современного цифрового ландшафта. Клиентская программа P2P, или просто «клиент», представляет собой сложный механизм, обеспечивающий динамичное взаимодействие участников сети.

Основываясь на мощном протоколе, данный клиент служит связующим звеном между пользователями, позволяя запрашивать актуальную информацию о файлах и узлах.

Процесс начинается с отправки запроса к узлам или серверам, после чего клиент получает данные о местонахождении интересующих файлов, а также о самих узлах, которые их хранят.

1.4 Архитектура информационной системы

Архитектура информационной системы - это концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы. Некоторые виды архитектуры информационной системы [19,с.50]:

- информационная архитектура: набор методик и инструментов, описывающий информационную модель конкретной предметной области (домена), включает в себя базы и хранилища данных, информационные потоки;
- архитектура прикладных решений: совокупность прикладных программных продуктов и интерфейсов между ними;
- техническая архитектура: взаимосвязь программно-аппаратных средств, методов и стандартов, обеспечивающих эффективное функционирование приложений;
- бизнес-архитектура: модель бизнес-процессов и сквозных информационных потоков.

Технология веб-служб в архитектуре информационной системы позволяет интегрировать приложения различной природы и строить распределённые информационные системы.

Технология веб-служб, активно используемая в современном программном обеспечении, позволяет разбить любую прикладную задачу на множество небольших модулей, каждый из которых обеспечивается отдельной серверной службой.

Средства моделирования данных позволяют детально описать объекты инфраструктуры, субъектов деятельности и происходящие между ними взаимодействия. Далее необходимо обосновать необходимость разработки

информационной системы локального уровня, направленную на контроль автотранспортных процессов ООО «Транспортная компания».

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. Принцип работы заключается в отслеживании и анализе пространственных и временных координат транспортного средства.

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

Таким образом, система контроля горюче-смазочного материала ООО «Транспортная компания» должна предоставлять следующие функции:

- контроль заправки и уровня топлива по датчику (ДУТ);
- передачу данных о текущем уровне топлива и на протяжении всего маршрута следования на сервер GPS-мониторинга;
- формирование контрольно-аналитических отчётов о состоянии и месте нахождения каждой единицы автотранспортных средств и получения другой информации.

Датчик уровня топлива (ДУТ) – это датчик, позволяющий с точностью зафиксировать уровень топлива в баке автомашины или другой емкости и контролировать топливо в реальном времени через GPS и ГЛОНАСС терминалы.

Передача данных обеспечивается благодаря цифровому, аналоговому или частотному сигналу. В конструкции датчика расхода топлива предусмотрены автономные измерительные устройства емкости и температуры.

Итоговый анализ информации производится во встроенном микропроцессоре, который отвечает за выработку аналоговых сигналов и поддерживает протокол связи посредством цифрового интерфейса. Устройство может оснащаться разными интерфейсами, соответствующими его функциональному назначению (рисунок 1.6).

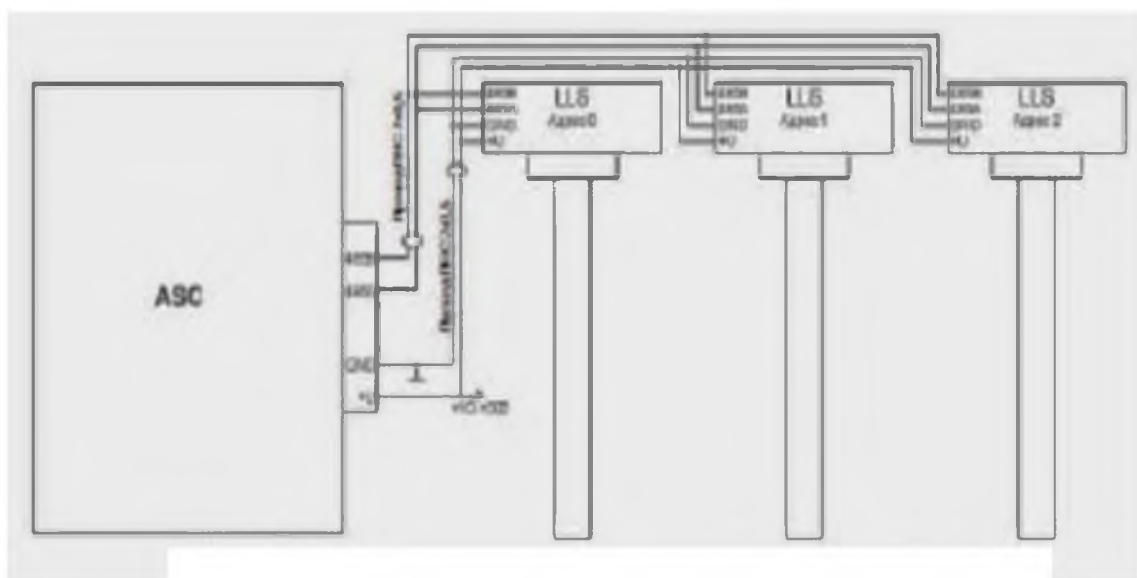


Рисунок 1.6 – Схема подключения датчиков уровня топлива LLS по интерфейсу RS485

Необходимо выделить, что необходимо точно знать марку/модель трекера, которым оборудовано автотранспортное средство. ДУТ подбирается, исходя из доступных интерфейсов. При наличии у трекера и цифрового, и аналогового интерфейса рекомендована установка ДУТ с цифровым интерфейсом (например, RS-485). Подключение к датчикам уровня топлива Omnicomm LLS по интерфейсу RS-485 производите согласно схеме, приведенной на рисунке 1.7.

Стоит отметить, что система GPS-мониторинга уровня топлива транспортных средств получает и преобразует данные от датчиков в информацию о текущем уровне топлива.

Исходя из изменений уровня топлива, определяются места остановок, дозаправок, время и координаты. Кроме того, в отчётах доступна информация о пробегах транспортного средства между загрузками и разгрузками. Диспетчер может в любой момент с персонального компьютера или мобильного устройства проверить точность выполнения заданий, соответствие загруженного и разгруженного топлива накладным и отсутствие несанкционированных сливов.

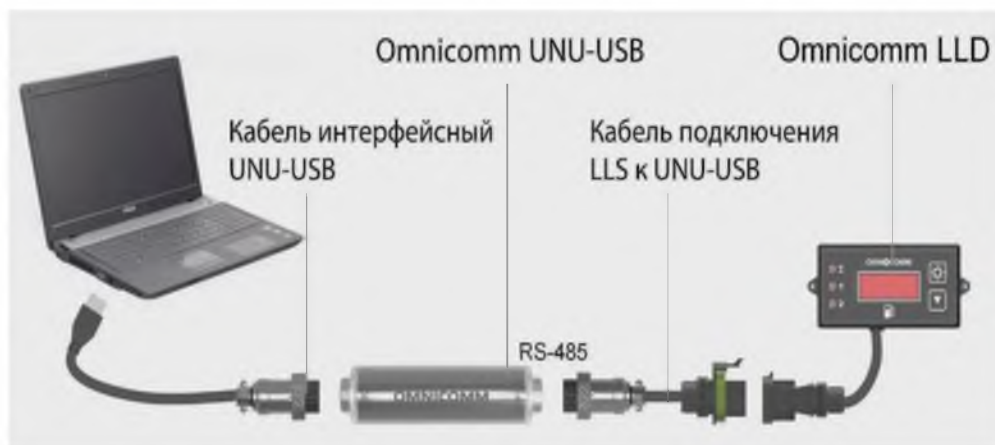


Рисунок 1.7 – Подключение ДУТ по интерфейсу RS-485

Геоинформационная система будет разрабатываться как многопользовательская для мобильных устройств на платформе Android, с уровнем API. Система предполагает консолидацию всех информационных ресурсов в едином хранилище данных на единой топографической основе, с использованием единой системы кодификации и классификации объектов учёта. Это позволяет охватить все сферы процесса управления и дать возможность пользователям синхронизировать между собой процессы своей деятельности. Одним из важных отличительных требований к проектируемой информационной системе является мобильность, так как особенность специфики предприятия заключается в потребности хранения данных под рукой не только диспетчера, но и начальника Транспортной службы - отсутствии привязки к рабочему месту.

Поэтому закономерно можно прийти к выводу о необходимости создания мобильного приложения.

Для гарантии целостности данных планируется эксплуатировать встроенные механизмы СУБД, поскольку безопасность информации необходимо построить на основе современных реляционных или объектно-реляционных СУБД. Также необходима специализированная подсистема резервного копирования и восстановления данных.

Для разработки многопользовательской ГИС используют технологию клиент-сервер. Она предполагает наличие программы-клиента для конечного пользователя и программы-сервера, который ведёт базу пространственных данных. Наличие выделенного сервера данных позволяет реализовать систему контроля доступа и разграничения прав пользователей.

Для хранения пространственных данных применяют системы управления базами данных (СУБД). Они позволяют использовать механизм транзакций, который обеспечивает целостность и корректность данных при многопользовательской работе.

В качестве вариантов для внедрения будут рассматриваться ГИС«MapView». Мобильная ГИС для ОС Android, которая позволяет подключать геопорталы пространственных данных (OpenStreetMap, Yandex и другие) и обрабатывать информацию со встроенного GPS/ГЛОНАСС-приёмника.

Для разработки и отладки ГИС-приложений в ней используются библиотеки MAPAPI и ArcGISRuntime SDK, с помощью которой можно создавать приложения на языке JavaScript, обеспечивающие доступ к функциям сервисов ArcGIS Server: картографическим, геокодирования и геообработки, а затем разворачивать эти приложения на устройствах с операционной системой Android.

Общение между водителями предполагается при помощи P2P, т.е. смартфоны, объединенные в единую систему и взаимодействующие посредством сетевого P2P протокола, который обеспечивает возможность создания и функционирования сети равноправных узлов, их взаимодействия. При выборе среды проектирования мобильной информационной системы были

рассмотрены 2 программных продукта: «Microsoft Visual Studio» и «AndroidStudio».

Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии WindowsForms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, WindowsMobile, Windows CE, NETFramework, Xbox, WindowsPhone.

В результате изучения выше перечисленных средств разработки, была выбрана платформа «AndroidStudio» так как она является специализированной средой разработки, имеет самые передовые инструменты для визуального проектирования, а также настройки пользовательского интерфейса, что позволяет легко настроить для нужд конкретной организации, предоставляет широкие возможности для формирования отчетов.

В зависимости от целевой аудитории нужно решить, на какой операционной системе будет работать приложение: iOS, Android или на обеих сразу. Отметим, что проектируемая система является локальной по пространственному охвату, так же определена спецификация проектируемой системы – компьютерный мониторинг местонахождения и основных параметров транспортных средств, перевозящих различные грузы. Определены подсистемы и требования к функциональности информационной системы.

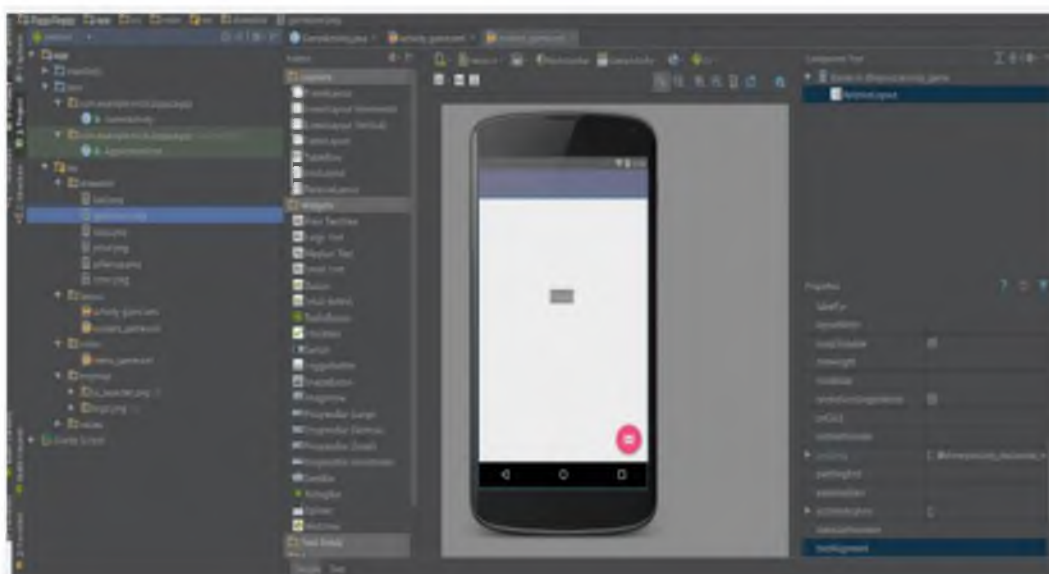


Рисунок 1.8 - Создание макета приложения в AndroidStudio

В данной работе был использован шаблон с названием «NavigationDrawerActivity», представляющий собой окно с всплывающим, по вызову пользователя, меню (рисунок 1.8).

Рассмотрим макет мобильного приложения, созданный в AndroidStudio, на примере страницы добавления информации о местонахождении транспортного средства ООО «Транспортная компания» (рисунок 1.9).



Рисунок 1.9 – Интерфейс мобильного приложения

Структура Java кода, состоит из подключаемых библиотек и описания классов и методов, которые вызываются в качестве обработки различных событий, таких как создание формы, нажатие на кнопку и другие. Отмечено, что ГИС является аналитическим средством работы с данными (в виде баз данных) и картографическим материалом, то есть компьютерные средства для отображения и анализа данных.

Как видно из текста разметки, основой для хранения элементов служат компоненты разметки LinearLayout, в которых размещаются остальные группы элементов такие как TextView - отображает текст, EditText - позволяет

осуществлять ввод текста, Button - представляет собой кнопку при нажатии на которую пользователь вызывает обработчик события. Обработка событий описывается на языке JavaScript, листинг обработки событий представлен в приложении 1 выпускной квалификационной работы.

Принято решение о том, что контроль ГСМ будет осуществляться с использованием ёмкостных датчиков уровня топлива типа. Геоинформационная система будет проектироваться для мобильных устройств на платформе Android, с использованием интегрированной среды разработки AndroidStudio. Информационная система будет эксплуатировать механизмы СУБД, и иметь возможность работать с картографическим материалом.

Таким образом, разработанная система станет важным инструментом повышения эффективности управления логистическими процессами. Основная цель – обеспечить непрерывный мониторинг местонахождения транспортных средств на протяжении всего срока их эксплуатации, а также отслеживание ключевых параметров, таких как скорость, время работы и уровень топлива. Подсистемы, включенные в проект, будут охватывать аспекты геолокации, диагностики технического состояния, а также управления грузоперевозками.

2 Проектная часть

2.1 Методология и методика проектирования

Методология проектирования информационных систем охватывает широкий спектр аспектов, начиная от определения требований пользователя до разработки архитектуры системы и выбора подходящих технологий. Важными элементами этой методологии являются системный подход и логическая структура, позволяющие четко формулировать проблемы и эффективно их решать. Система управления проектом играет ключевую роль в координации всех этапов разработки.

Концептуальная модель базы данных включает в себя [9,с.55]:

- описание информационных объектов или понятий предметной области и связей между ними;
- описание ограничений целостности, то есть требований к допустимым значениям данных и к связям между ними.

В построении общей концептуальной модели данных выделяют ряд этапов:

- выделение локальных представлений, соответствующих обычно относительно независимым данным;
- формулирование сущностей, описывающих локальную предметную область проектируемой БД, и описание атрибутов, составляющих структуру каждой сущности;
- определение ключевых атрибутов;
- спецификация связей между сущностями. Удаление избыточных связей;
- анализ и добавление неключевых атрибутов;
- объединение локальных представлений.

Важно отметить, что концептуальное планирование не является разовым мероприятием, а представляет собой циклический процесс. Регулярное пересмотрение целей и адаптация стратегий на основании полученных

результатов позволяет организации оставаться конкурентоспособной и гибкой в условиях быстро меняющейся внешней среды. Таким образом, концептуальное планирование становится эффективным инструментом в достижении долгосрочного успеха и устойчивого развития организации.

Созданная концептуальная модель данных предприятия является источником информации для фазы логического проектирования базы данных.

В последствие 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. На основе полученных данных была построена IDEF-модель (рисунок 2.1).

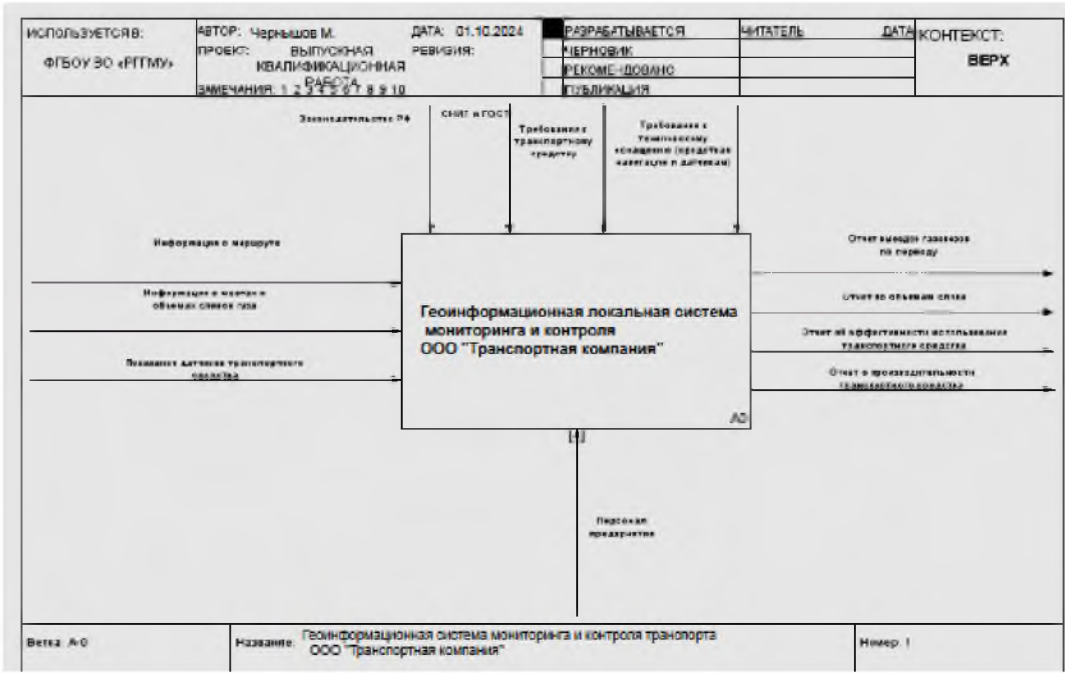


Рисунок 2.1 - Модель IDEF0

На данном уровне представлена общая модель представления потоков данных о входных и выходных данных. Содержит данные о том, какими документами руководствуются функции системы. Декомпозиция модели IDEF0 по учету ГСМ представлена на рисунке 2.2. В данном блоке, описывается реализация функции «Учет объемов ГСМ», в качестве входных данных

поступает информация о месторасположениях, маршрутах, на выход поступает информация о выездах, данные о маршруте, а также, на основе введенных данных формируется отчет заказов по периоду.

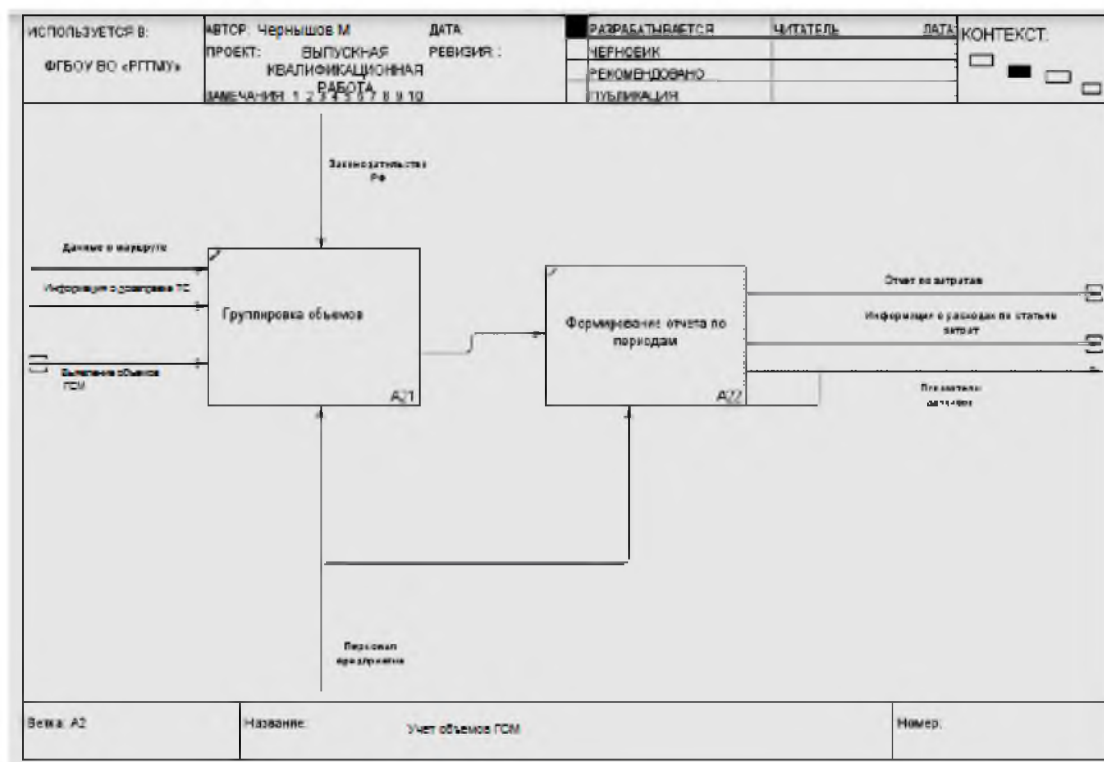


Рисунок 2.2 - Декомпозиция блока «Учет объемов ГСМ»

На основании собранной информации о маршрутах, объемах заправки и установленных нормах топлива осуществляется глубокий анализ использования автотранспортных средств. В частности, важным элементом является процесс учета показателей использования каждого транспортного средства, представленный на рисунке 2.3.

В данном блоке на вход поступают данные, полученные с датчиков, которые фиксируют различные параметры работы автомобилей, такие как пробег, расход топлива, время простоя и другие ключевые показатели.

Объектно-ориентированный подход к концептуальному проектированию использует объектную декомпозицию: структура системы определяется множеством объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Основные принципы

объектно-ориентированного подхода: абстрагирование, ограничение доступа, модульность, иерархичность, типизация, параллелизм и устойчивость.

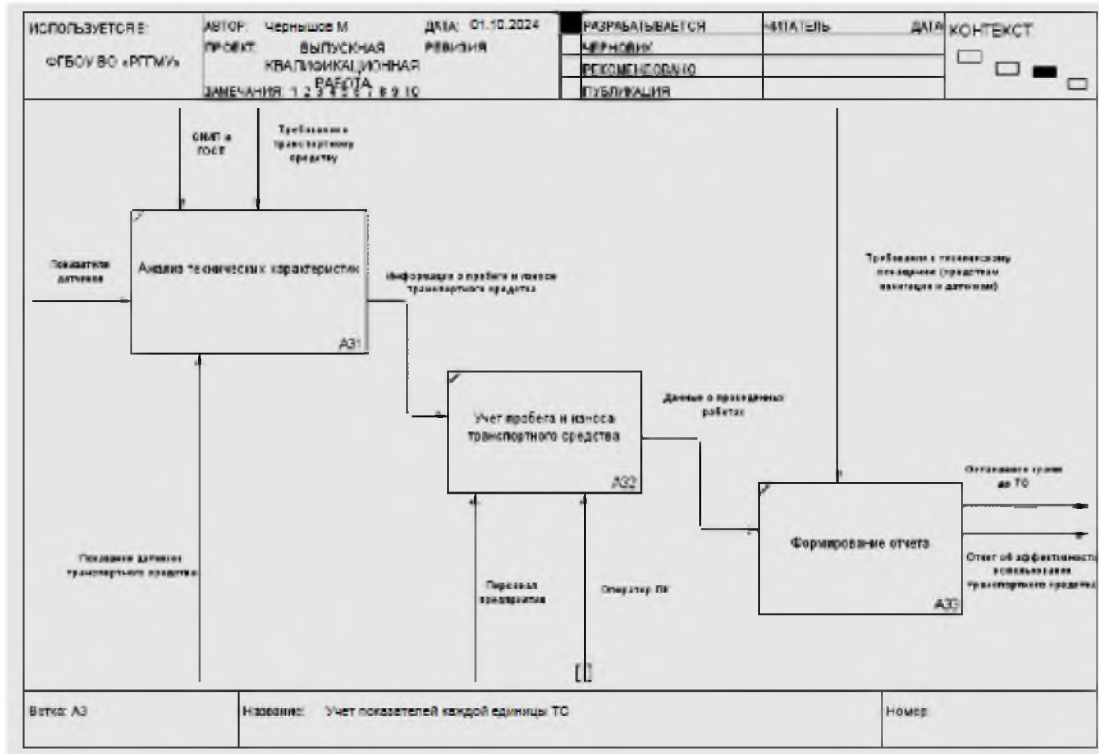


Рисунок 2.3 - Декомпозиция блока «Учет показателей использования каждой единицы ТС»

В ходе разработки реляционной модели базы данных будет создана структура, позволяющая эффективно управлять данными и их взаимосвязями. Каждая таблица в реляционной базе будет представлять собой сущность, где строки соответствуют записям, а столбцы — атрибутам. Первичный ключ каждой таблицы будет обеспечивать уникальность записей, в то время как внешние ключи будут служить связующими элементами между таблицами, формируя необходимые отношения.

Важным этапом является нормализация данных, которая поможет избежать избыточности и повысит целостность информации. Процесс нормализации включает разбивку данных на множество связанных таблиц, что способствует упрощению управления и обновления информации. Установление отношений предок-потомок будет способствовать более глубокой иерархической организации данных, облегчая навигацию по ним.

Кроме того, использование механизмов, таких как индексы, триггеры и хранимые процедуры, не только оптимизирует запросы, но и значительно улучшает общую производительность системы. Автоматизация процессов обработки данных становится ключевым элементом, обеспечивая интерактивность и отзывчивость системы к пользовательским запросам.

Первая линия защиты целостности данных обеспечивается первичным ключом, представляющим собой уникальное значение для каждой записи, даже если он состоит из нескольких столбцов. Это гарантирует, что каждая запись в таблице может быть однозначно идентифицирована, что критично для надежного управления данными.

Внешний ключ, с другой стороны, служит связующим звеном между таблицами, обеспечивая соответствие значений первичного ключа другой таблицы. Такое отношение создает структуру «предок/потомок», аналогичную иерархическим базам данных. В реляционных СУБД эта связь позволяет сохранять связи между данными, обеспечивая целостность и облегчая сложные запросы, что в конечном итоге ведет к более эффективному использованию ресурсов и улучшению общего опыта пользователей.

Диаграмма сущность-связь представлена на рисунке 2.4.

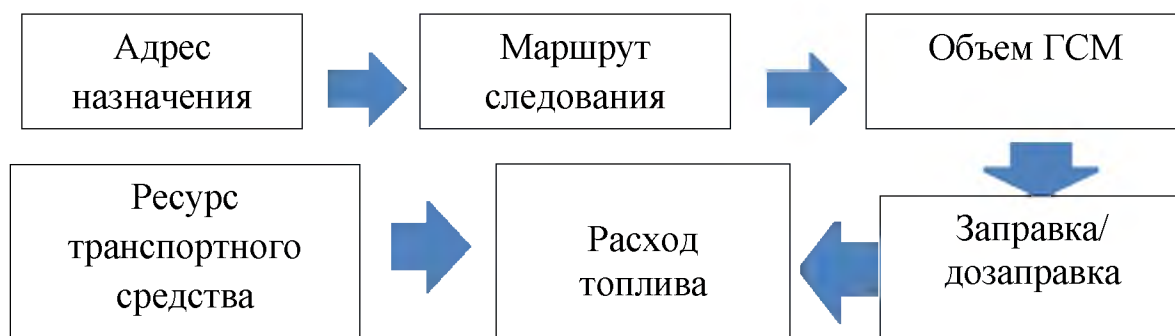


Рисунок 2.4 - Диаграмма сущность - связь (ERD)

Модель данных, основанная на ключах (KB), для автотранспорта — это более подробное представление данных. Она включает описание всех сущностей и первичных ключей и предназначена для представления структуры данных и ключей, которые соответствуют предметной области.

Основная цель модели, основанной на ключах — широкий обзор структур данных и ключей, нужных для поддержки определённой области. Эта модель определяет контекст, в котором могут быть созданы подробные модели, пригодные для конкретного воплощения (рисунок 2.5).

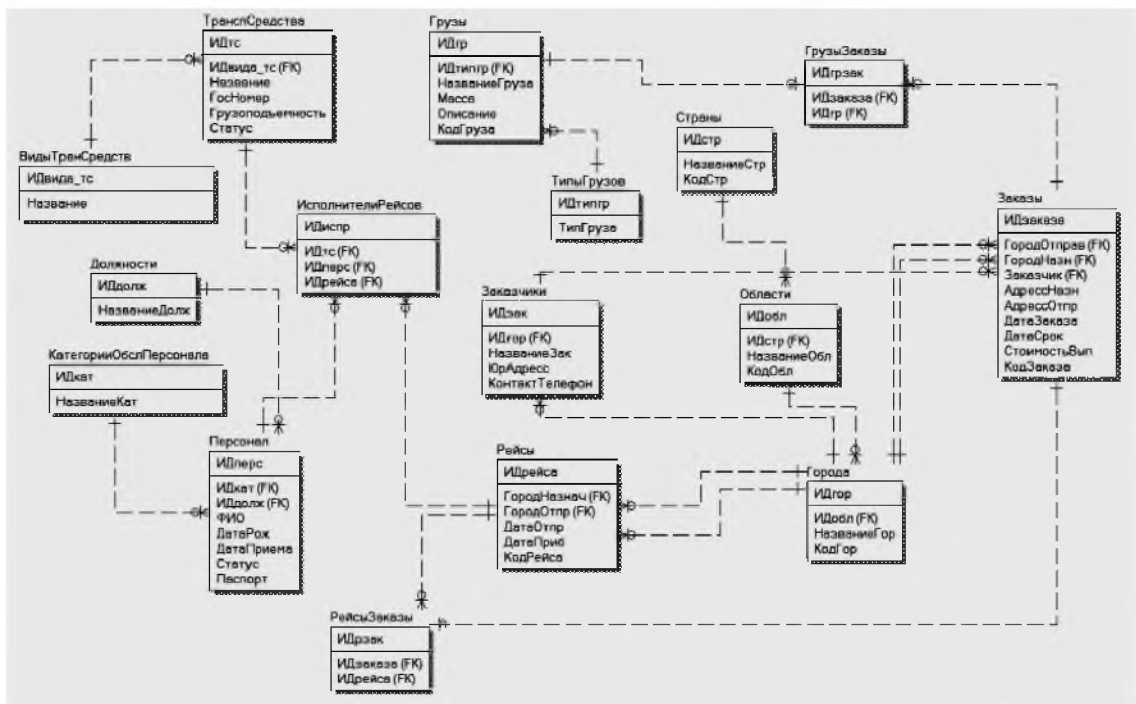


Рисунок 2.5 - Модель данных, основанная на ключах (КВ)

Диаграмма КВ - уровня отражает логическую структуру связей сущностей, составляющих предметную область деятельности. На уровне атрибутов представлены все атрибуты сущностей. Эта диаграмма содержит полные определения структуры создаваемой системы. Любая информационная система включает некоторую базу данных, для того, чтобы работать с информацией, нужно работать с данными. Информация получается из данных, если над ними произведена некоторая обработка, повышающая их ценность. Данные - более низкий уровень агрегации и сопоставления, информация — более высокий.

В информационной системе должны храниться данные о клиентах, поставщиках, видах услуг, материалах, комплектующих. Данные будут храниться в связанных между собой таблицах, состав и наименования которых указаны в таблице 2.1.

Таблица 2.1 - Сущности системы

Сущность	Атрибуты сущности описание	Тип данных	Описание
Адрес назначения	№ Резервуара	Счетчик	Первичный ключ. Уникальный номер для каждого резервуара
	Адрес	Текст	Адрес доставки газа
	Наименование резервуара	Текст	Название резервуара
Маршрут	№ Маршрута	Счетчик	Первичный ключ. № Маршрута
	Адрес отправления	Текст	Название пункта, из которого будет осуществлена отправка груза
	Адрес назначения	Текст	Название пункта, в который будет осуществлена доставка груза
	Километраж	Число	Пройденное расстояние
Объем ГСМ	№ Заказа	Счетчик	Первичный ключ № заказа
	№ Маршрута	Число	Внешний ключ № маршрута
	Дата	Дата'Время	Дата выполнения заказа
	Стоимость топлива	Денежный	Стоимость полной заправки
	Расход топлива	Число	Объем топлива
Расходы	№ Расхода	Счетчик	Первичный ключ № расходной операции
	Сумма	Денежный	Сумма расчета
	Дата время	Дата-время	Дата и время расчета
	Цель расчета	Текст	Комментарий об осуществлённой отгрузке груза
	№ заправки	Число	Внешний ключ № заправки
	№ ТС	Число	Внешняя сущность № ТС
	№ заказа	Число	Внешняя сущность № заказа

Продолжение таблицы 2.1

Ресурс ТС	№ ТС	Счетчик	Первичный ключ № ТС
	Стоимость ТО	Денежный	Стоимость технического обслуживания
	Стоимость закупки оборудования	Денежный	Стоимость закупки оборудования
	Комментарий	Текст	Комментарий об осуществлённой покупке
	Дата начала	Число	Внешняя сущность № технического обслуживания
	Дата окончания	Число	Внешняя сущность № технического обслуживания
Заправка	№ заправки	Счетчик	Первичный ключ № заправки
	Цена за литр	Денежный	Цена за полную заправку
	Количество заправленного топлива	Число	Количество сделанных маршрутов

Таким образом, выходной информацией будут являться отчеты:

— отчет заказов по периоду - содержит информацию о заказах, совершенных в указанных пользователем временных рамках;

— отчет по затратам - содержит информацию о количестве расходов определенном временном промежутке;

— отчет об эффективности деятельности - в отчете содержится информация о количестве выполненных заказов, информация о прибыли и убытках, о пройденном расстоянии;

— отчет о произведённом обслуживании транспортного средства - в отчете содержится информация о сроках проведения ремонтных работ, о количестве затраченных средств на ремонтные работы, а также на закупку оборудования.

В ходе определения основных элементов информационной системы была

создана диаграмма прецедентов. Применялись отношения обобщения, ассоциации, включения и расширения. На рисунке 2.6 отображена диаграмма тематического исследования (USE-CASE).

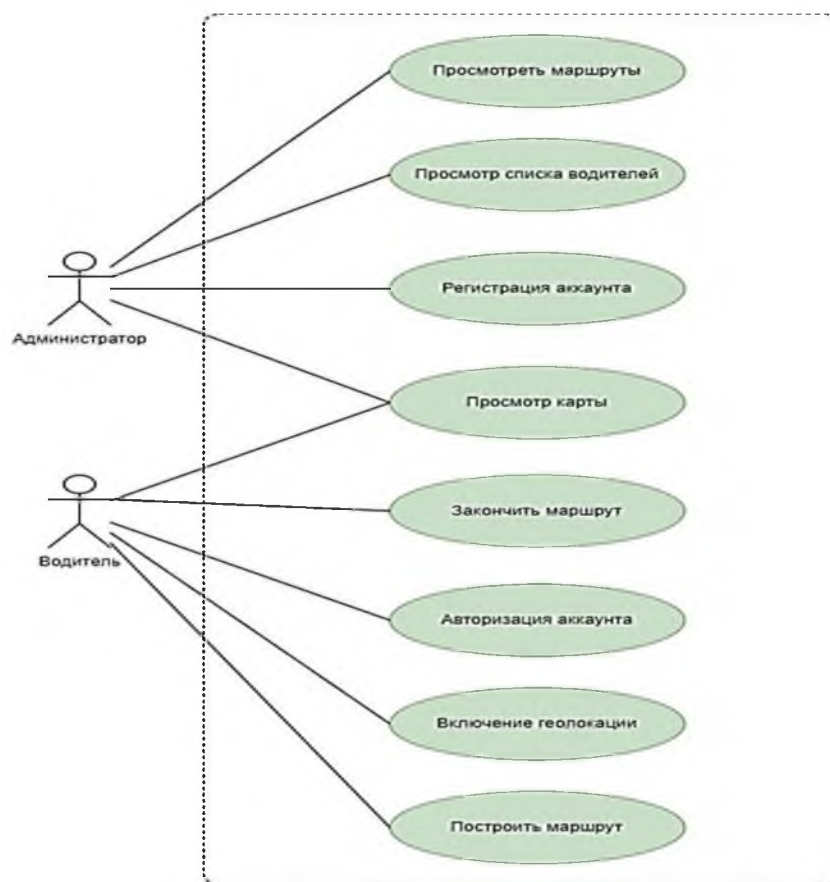


Рисунок 2.6 - Диаграмма тематического исследования (USE-CASE)

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 актёров.

В приложении 2 показано, как будет работать авторизация драйвера веб- приложения.

Схема работы системного драйвера «Авторизация» выглядит следующим образом [30]:

— водитель нажимает кнопку входа в систему после ввода информации в поле входа;

- начинается проверка данных;
- данные передаются на сервер;
- в базе данных выполняется поиск;
- сервер получает поток данных обратно и проверяет запись;
- отображение результатов.

Регистрация - используется в информационной системе для обработки запроса на создание учетной записи, предоставляющей доступ для выполнения определенных системных действий. Диаграмма, на которой изображена регистрация администратора приложения, относится к этому процессу (приложение 3).

Алгоритм работы схемы «Авторизация» для системного администратора выглядит следующим образом [30]:

- администратор вводит информацию о водителе и сохраняет ее в окне регистрации;
- запускается проверка данных;
- данные передаются на сервер;
- происходит сохранение базы данных;
- сервер получает поток данных обратно и проверяет сохранение;
- закрытие окна.

Добавление маршрута используется для предоставления драйверам этого программного продукта нового пути. Этот проект учитывает включение маршрута драйвером самого мобильного приложения (приложение 4).

Алгоритм работы схемы «Авторизация» для системного администратора выглядит следующим образом [30]:

- водитель нажимает кнопку запуска окна добавления маршрута, вводит необходимую информацию, а затем нажимает кнопку «Ввод»;
- начинается проверка данных 2;
- данные передаются на сервер;
- создается запись базы данных (новая запись);

- данные через сервер возвращаются драйверу с ответом после успешной записи;

- закрытие окна и 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 5.

С помощью диаграмм развертывания можно визуализировать, как компоненты системы взаимодействуют друг с другом и какие ресурсы необходимы для их работы. Это позволяет разработчикам не только фирменно проектировать архитектуру, но и оценивать техническую осуществимость системы. Одним из основных преимуществ использования ClassDesigner является возможность последовательно уточнять детали проектируемой системы, а также вносить изменения по мере необходимости. Тем самым, этот инструмент способствует адаптивности и гибкости разработки, позволяя командам легче реагировать на новые требования. Важно отметить, что UML-диаграммы служат не только для разработчиков, но и могут быть полезны для заказчиков, помогая им получить более четкое представление о структуре и функциональности системы.

Диаграмма классов создается на протяжении всего процесса проектирования системы, чтобы помочь прояснить статические отношения между различными классами. 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 (приложение 6).

Таким образом, интеграция инструментов моделирования в профессиональную среду разработки, такую как VisualStudio, выступает как важный элемент современного подхода к построению качественного программного обеспечения. Это не только упрощает процесс коммуникации внутри команды, но и повышает общее качество создаваемого продукта.

2.2 Особенности проектирования на платформе Node

Node.js- это недавно появившаяся платформа, которая выводит язык JavaScript за пределы браузера и позволяет использовать его в серверных приложениях. В основе платформы лежит исключительно быстрый движок JavaScript, заимствованный из браузера Chrome, V8, к которому добавлена быстрая и надежная библиотека асинхронного сетевого ввода/вывода. Основной упор в Node делается на создании высокопроизводительных, хорошо масштабируемых клиентских и серверных приложений для «веб реального времени».

Цель Node - предложить «простой способ построения масштабируемых сетевых серверов». При проектировании за образец были взяты такие системы, как EventMachine (Ruby) и каркас Twisted (Python) [28].

Тем не менее, стоит отметить, что такая трансформация в архитектуре веб-приложений не является панацеей. На первый взгляд, возможности неблокирующего ввода/вывода предлагают значительное преимущество в производительности и масштабируемости. Подобный подход позволяет серверам обрабатывать множество соединений одновременно, не ожидая завершения каждого из них. Таким образом, в реальном времени появляется возможность улучшить взаимодействие с пользователем, обеспечивая быструю реакцию на запросы.

Платформу Node не имеет смысла напрямую сравнивать ни с языками программирования, которые обычно используются для создания веб-приложений: PHP/Python/Ruby/Java и пр., ни с контейнерами, реализующими протокол HTTP :Apache/Tomcat/Glassfish и т.д. В то же время многие считают, что потенциально она может заменить традиционные стеки веб-приложений.

Существующие модули Node.js предлагают множество возможностей, однако, несмотря на пламенные надежды разработчиков, возможности создания настольных GUI-приложений всё еще ограничены. Библиотеки, работающие с такими интерфейсами, остаются на стадии разработки, и лишь несколько

проектов стремятся интегрировать Node.js с такими инструментами, как GTK. Это создает ожидания появления кросс-платформенных решений, которые смогут расширить функционал Node.js.

Тем не менее, его сильные стороны все же проявляются в других областях. Удобный сетевой уровень позволяет легко создавать приложения, использующие HTTP и другие протоколы, при этом обеспечивая разработчикам возможность тщательно настраивать необходимые параметры, включая заголовки запросов. Наличие модуля для работы с двоичными данными и мощные средства управления процессами делают Node.js идеальным выбором для сервисов, требующих обработки большого объема данных в реальном времени. Таким образом, хоть Node.js и не предлагает готовых решений для GUI, его богатый функционал и возможности интеграции делают его мощным инструментом для веб-разработки и серверного программирования.

Таким образом, Node.js предоставляет разработчикам возможность эффективно обрабатывать большое количество одновременных соединений с минимальными затратами ресурсов. Это стало возможным благодаря неблокирующей модели выполнения, где каждый запрос обрабатывается асинхронно. Вместо того чтобы ждать завершения операций ввода/вывода, Node.js позволяет выполнять другие задачи, что значительно повышает производительность приложений. Кроме того, архитектура событийного цикла делает его особенно подходящим для создания высоконагруженных веб-приложений, таких как чаты или онлайн-игры, где скорость реакции и масштабируемость критически важны.

JavaScript, будучи унифицированным языком для фронтенда и бэкенда, упрощает процесс разработки, позволяя разработчикам использовать одни и те же принципы и методы на обеих сторонах. Это также снижает Learning Curve для новых разработчиков, так как они могут работать с одним языком вместо нескольких. В результате, использование JavaScript и Node.js открывает новые горизонты в разработке сетевых приложений, создавая более гибкие, быстрые и

масштабируемые решения, что в свою очередь влияет на общее развитие веб-технологий в 21 веке.

Составной модуль может включать несколько внутренних модулей, файлы данных, файлы шаблонов, документацию, тесты и прочее. Все это можно поместить в хорошо продуманную структуру каталогов, которую Node будет рассматривать как модуль, и загружать командой `require('moduleName')`. Для этого следует добавить в каталог файл модуля `index.js` или файл с именем `package.json`. Файл `package.json` должен содержать данные, описывающие модуль, в формате, очень похожем на формат файла `package.json`, используемого менеджером пакетов `npm`.

В обоих случаях для совместимости с Node достаточно очень небольшого набора полей, распознаваемых `npm`. `Npm` позволяет быстро находить пакеты для решения конкретной задачи, загружать и устанавливать их, а также управлять уже установленными пакетами. В `npm` определен формат пакета для Node, основанный на спецификации `CommonJS`.

Формат `npm`-пакета представляет собой структуру каталогов, описанную в файле `package.json`. Для взаимодействия Node.js с базами данных SQL используется модуль `sequelize`. Он поддерживает MySQL, PostgreSQL, MSSQL и MariaDB. Модуль имеет единое API для всех перечисленных СУБД, поскольку они используют единый язык описания запросов SQL.

Для работы с базой данных подходит библиотека `Sequelize`. Она предоставляет удобный и интуитивно понятный способ моделирования данных и выполнения запросов из приложения Node.js. Более того, `Sequelize` предоставляет удобный способ взаимодействовать с базой данных SQL, абстрагируя сложности прямых запросов.

Некоторые возможности библиотеки прослеживаются в поддержке создания, обновления, удаления сущностей, вложенных сортировок, сложных условий, лимитов, подзапросов, кастомных запросов, а также защита от SQL-инъекций и отмена транзакций. Библиотека позволяет определить схемы для данных, которые нужно хранить в базе данных, установить и поддерживать соединение,

выполнять различные операции с ней (поиск, вставка, обновление и удаление документов).

Для подключения Node.js к базе данных MySQL можно использовать пакет `mysqlnpm`. После установки нужно настроить учётные данные базы данных и установить соединение.

Для создания бэкенд - части приложения на Node.js можно использовать фреймворк Android JS. Он позволяет разрабатывать приложения Android с использованием фронтальных и серверных компонентов, изначально разработанных для веб-приложений: среды выполнения Node.js для бэкэнда и `AndroidWebview` для внешнего интерфейса.

Node.js становится все более популярным выбором для разработки мобильных приложений благодаря своей способности обеспечивать высокую производительность и эффективность.

В частности, проект Node.js forMobileApps предлагает разработчикам мощный инструмент для интеграции серверной логики на основе Node.js непосредственно в мобильные приложения для платформ Android и iOS. Это предоставляет уникальные возможности для создания приложений, способных обрабатывать большие объемы данных и реализовывать сложные бизнес-логики в реальном времени [26].

2.3 Основные элементы информационной системы

В составе обеспечивающей части информационной системы объектами проектирования служат элементы и их комплексы информационного, программного и технического обеспечения системы [3,с.16].

В наиболее общем виде, информационную систему можно представить состоящей из следующих элементов [7,с.85]:

- источника информации;
- аппаратной части ИС;
- программной части ИС;

— потребителя информации.

Структуру информационной системы составляет совокупность отдельных ее частей, называемых подсистемами. Для построения информационной системы применялись стандартные методы проектирования[15]:

— бесплатная кросс-платформенная среда Node.js – позволяющая создавать приложения, где нужна обработка больших объёмов информации;

— VisualStudioCode- 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: мобильное приложение будет создано с использованием HTML, CSS, JavaScript и PHP. Важно отметить, что первые два языка отвечают за «основу» системы, а следующие два - за его функциональную составляющую;

— система управления реляционными базами данных MySQL, которая предоставляется бесплатно;

— проектируемая информационная система будет построена на модуле интерактивной карты от Яндекса, так как он прост в использовании и предлагает широкий спектр функций.

Для начала проектирования мобильного приложения для Android с Node.js необходимо выполнить следующие шаги:

Установить необходимые инструменты. На компьютере должны быть Node.js и npm.

Установить глобально сборщик проектов Android JS.

Для этого нужно выполнить команду:

```
npm install -g androidjs-builder@2.0.3
```

Создать папку для проекта.

Внутри неё нужно открыть терминал.

Создать обычный Node.js-проект. Для этого выполнить команду `npm init`.

Создать необходимые папки и файлы в приложении.

Создаем файл `main.js` в родительском каталоге приложения, файл `index.html` в каталоге `views` (это первое представление приложения, которое будет отображаться через веб-просмотрщик) и папку `assets` для хранения файлов приложения, в ней будут созданы 4 папки: `ipc`, `css`, `js`, `icon`.

Далее происходит скачивание иконки самого приложения, которая будет помещена в папку `assets/icon`.

Настраиваем приложение. В файле `package.json` необходимо определить название приложения, имя пакета, иконку, разрешения для Android и выходной каталог.

После сохранения файла установить зависимости с помощью команды `npm install`.

Упаковать приложение. Для этого нужно открыть консоль в каталоге приложения и выполнить команду `androidjs b-f`. Через некоторое время подписанный `apk` будет доступен внутри папки `dist`.

Также для создания Android-приложения с Node.js можно использовать фреймворк `AndroidJS`, который позволяет разрабатывать приложения с использованием фронтальных и серверных компонентов: среды выполнения Node.js для бэкэнда и `AndroidWebview` для внешнего интерфейса.

Изначально есть только учетная запись администратора, после чего водители регистрируются в системе самостоятельно. Для регистрации потребуется следующая информация: фамилия, имя, отчество, номер телефона и пароль. На рисунке 2.7 показано окно регистрации.

При выборе кнопки «Уже есть учетная запись» в окне остаются только поля логина и пароля, при условии, что пользователю достаточно войти в уже созданную учетную запись (рисунок 2.8).

The image 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 2.7 - Окно регистрации в системе

The image 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 2.8 – Окно входа в индивидуальную учетную запись программы

Далее открывается навигационное меню и карта с изображением Москвы после входа в собственный аккаунт. Учетная запись администратора показана на рисунке 2.9.

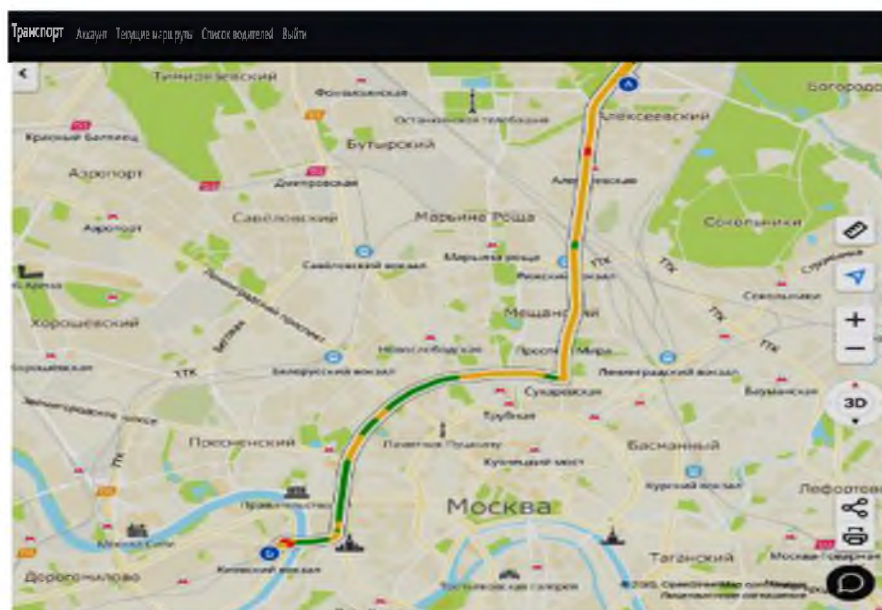


Рисунок 2.9 - Аккаунт администратора

Учётная запись администратора – это запись одного из пользователей, с помощью которой можно вносить изменения в записях других пользователей, в том числе и ограничивать их права. Как правило, ограничения накладываются на возможность устанавливать какое-либо программное обеспечение, на возможность менять параметры безопасности и иметь доступ к определённым файлам. Учётная запись администратора имеет полный контроль над файлами, каталогами, службами и другими ресурсами на локальном устройстве.

Выбрав кнопку «Учётная запись», администратор, как и водитель, может изучить информацию о своей учётной записи. Кроме того, при нажатии на соответствующую кнопку появится окно с подробной информацией о планируемых маршрутах, позволяющее администратору просмотреть маршруты на текущий день (рисунок 2.10).

Разумеется, системному администратору понадобится для работы список, зарегистрированных в системе водителей. Проектируемая система отображает список водительского состава по именам и администратор может связаться с ними по номеру телефона, используя базу данных. Последняя кнопка «Выйти», нацелена на выход из системы после того, как поставленная задача будет выполнена.

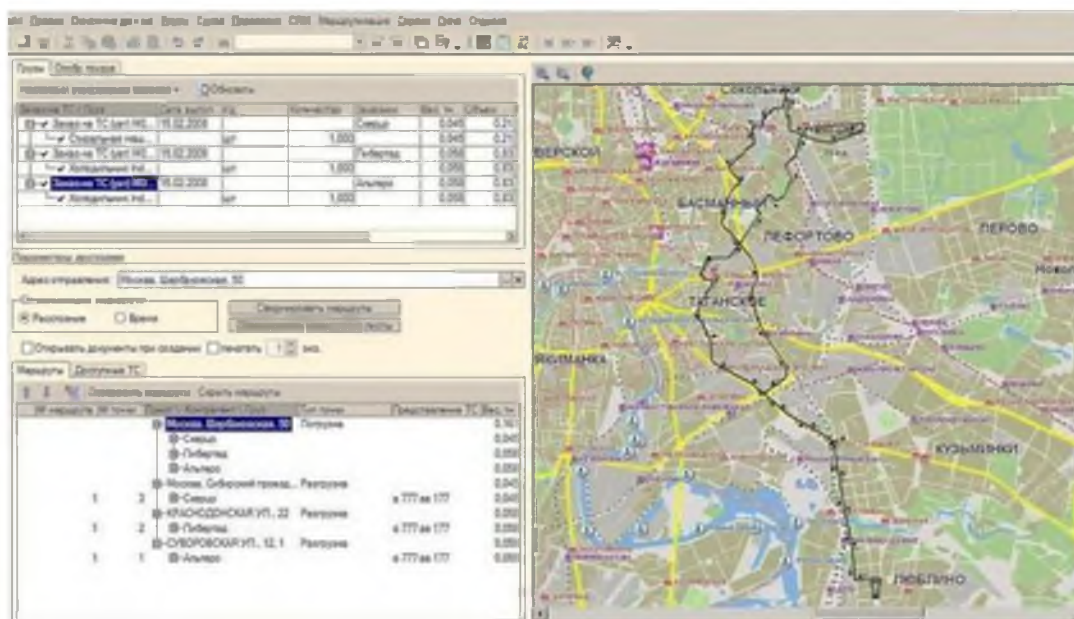


Рисунок 2.10 – Диалоговое окно «Текущие маршруты на день»

Добавление маршрута - первая кнопка после «Аккаунта». При нажатии которой, появляется окно для ввода новых адресов маршрута (рисунок 2.11). Диалоговое окно «Текущие маршруты на день» может представлять собой список точек, объединённых в единый маршрут на текущий день. Перемещаться по этому окну можно, прокручивая его вверх или вниз пальцем.

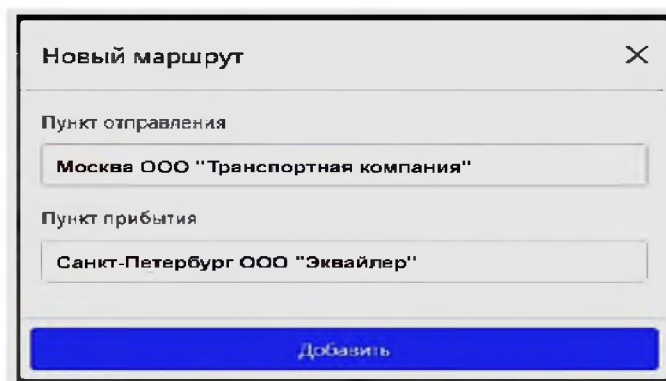


Рисунок 2.11 – Новое окно с определением маршрута следования

Крайне важно уведомить систему о том, что доставка груза прошла безупречно, как только товар был успешно доставлен в пункт назначения. Маршрут должен быть удален с карты после нажатия кнопки «Подтвердить доставку», а в базе данных будет сделана отметка о том, что маршрут пройден (рисунок 2.12).

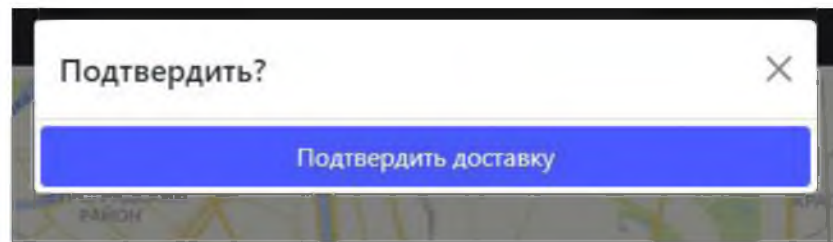


Рисунок 2.12 - Отслеживание доставки груза

На следующем рисунке 2.13 показан маршрут водителя, который детально фиксируется системой и сохраняет все данные на протяжении следования. Сама карта создана с использованием языка программирования JavaScript и дизайнерского решения Яндекса.

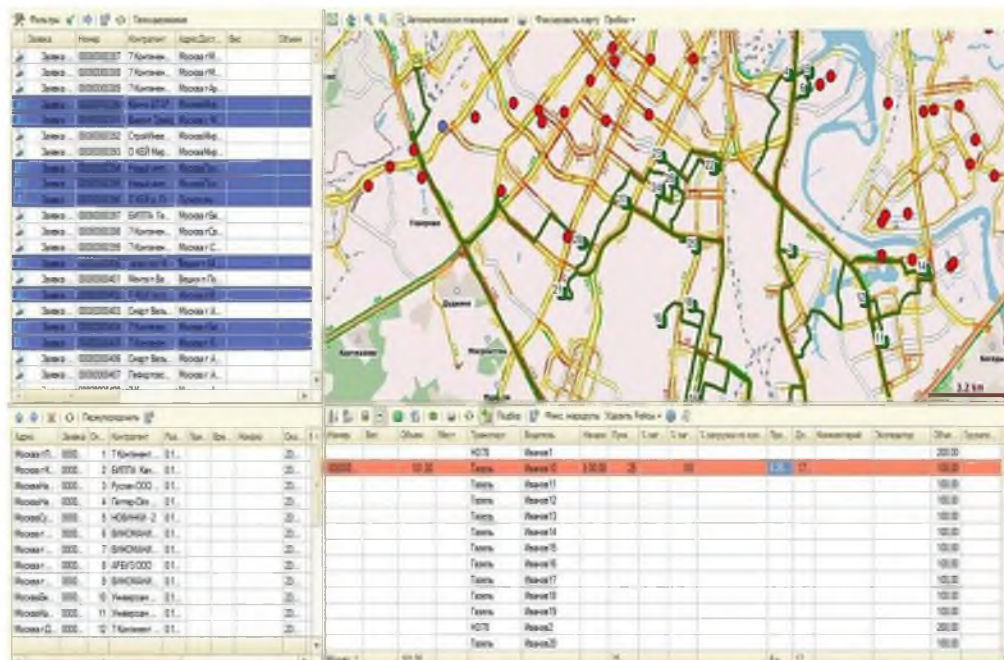


Рисунок 2.13 – Маршрут следования одной единицы транспорта

Ниже показано фактическое применение карты:

//Создание карты.

```
var geolocation = ymaps. geolocation,
```

```
myMap – new ymaps. Map("map", {
```

```
//Координаты центра карты.
```

```
//Порядок по умолчанию: "широта, долгота"
```

```
//Чтобы не определять координаты центра карты вручную, воспользуемся
```

```

//Инструментом Определение координат.
center:[59.393098, 30.315868],
//Уровень масштабирования. Допустимые значения:
//от 0 (весь мир) до 19.
zoom: 11
}, {
//зададим ограниченную область прямоугольником,
//примерно описывающим Москву
restrictMapArea:[
[59.72, 29.86]
[60.19, 30.9]
]
});

```

Далее для построения самого маршрута следования необходимо определить две геоточки с целью, вычисления пути самой системой:

```

Var multiRoute =new ymaps. multiRouter. MultiRoute (}
//Точки маршрута. Точки могут быть заданы координатами и адресом.
referencePoint:[
[59. 832285, 30.393959]
[59. 922251, 30.406939]
]
}, {
// Автоматически устанавливать границы карт так
// чтобы маршрут был виден целиком.
boudsAutoApply: true
});
//Добавление маршрута на карту.
myMap. geoObjects. Add (multi Route);

```

Поскольку вся информация хранится в базе данных, она служит основой. Представленная реализация базы данных основана на той, что показана на рисунке 2.14 в виде репозитория.

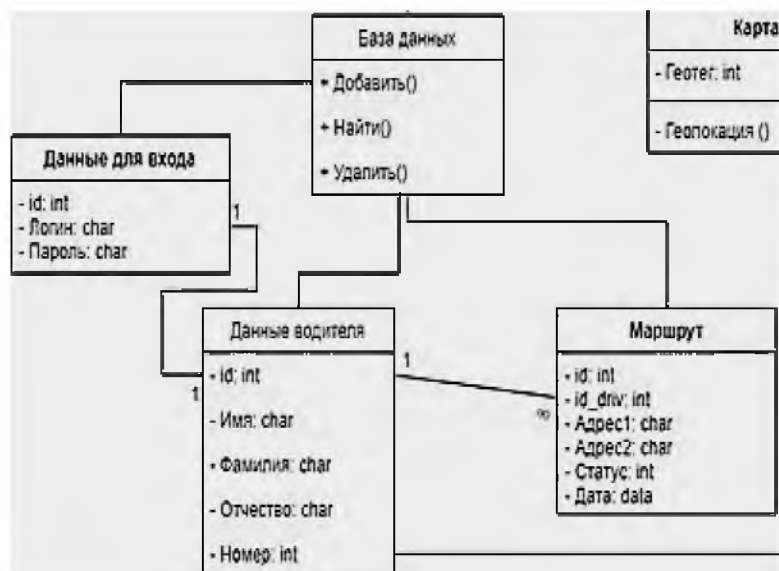


Рисунок 2.14 – Репозиторий ИС

2.4 Проектирование интерфейса пользователя

Набор определенных технических средств, с помощью которых пользователь взаимодействует с определенной цифровой версией программы принято считать интерфейсом, которым может выступать как мобильное приложение, определенная программа на компьютере или возможно веб-сайт.

Что такое интерфейс? Это все то, что помогает в первую очередь пользователю взаимодействовать в ходе работы для достижения ранее поставленной задачи и цели.

Помощь могут оказывать иконки, кнопки, формы и тд. Зачастую работая над той или иной версией информационного продукта, в первую очередь уделяется внимание сложившейся практике об определении внешнего вида, т.е. визуализации. Сама практика довольно так распространена и занимает чуть ли не первостепенное положение в процессе проектирования и модернизации пользовательских интерфейсов [12,с.28].

В нашем случае использовалось приложение Termux, которое устанавливается из магазина GooglePlay, а значит имеет форму бесплатного. Выбранная модификация требует версию Android 17.0 или более позднюю.

Изучив технические требования к разработке мобильной версии приложения был выбран потенциально используемый комплекс экранных заставок, а для каждой из них в свою очередь подобран набор элементов интерфейса. Весь процесс выбора отражен на рисунке 2.15.

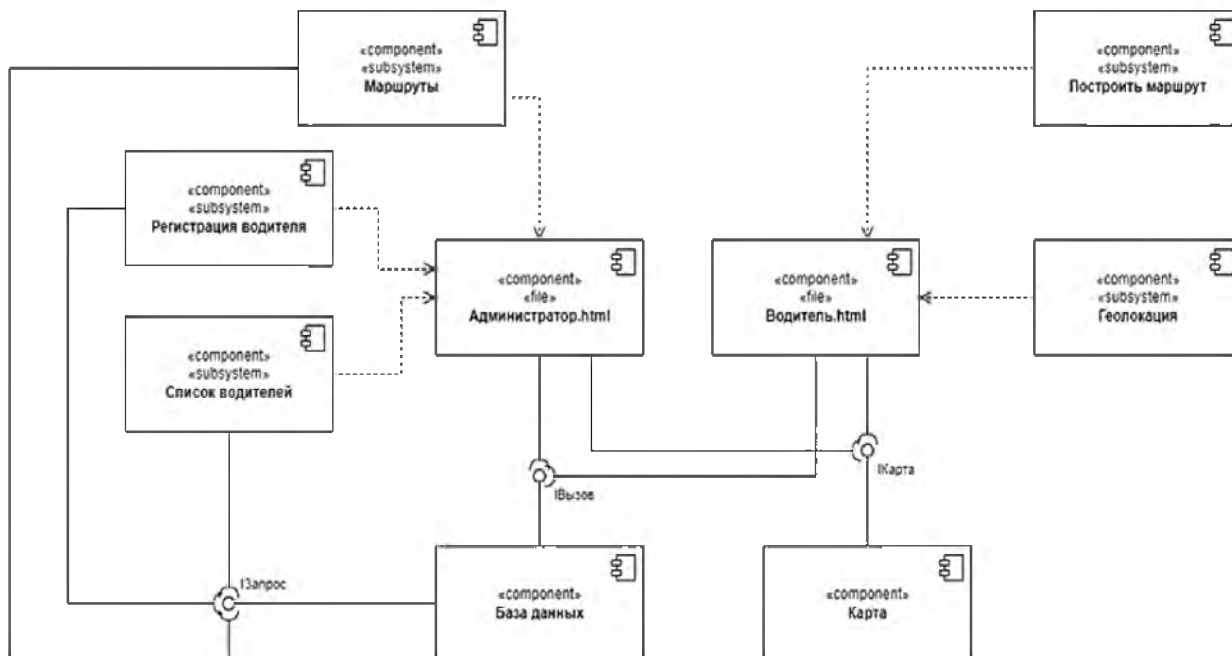


Рисунок 2.15 - Диаграмма развертывания

Проектируемая информационная система должна предоставлять возможности для анализа транспортных процессов ООО «Транспортная компания», а именно выполнение всего производственного цикла операций за смену. Система должна быть детально проработана еще до ее реализации с использованием разного рода анализов и схем. Будущая разработка системы должна учитывать знания, использованные при проектировании системы и создания веб-приложения.

Установка системы ГЛОНАСС/ (GPS) позволит осуществлять:

- контроль расхода ГСМ за счет установки ДУТ;
- контроль пробега и соблюдение заданного маршрута, сокращает приписки в путевых листах.

Абонентский терминал принимает данные от системы позиционирования ГЛОНАСС/ (GPS) и передает их в систему через GSM-сеть в диспетчерский

центр. Интерфейс системы интуитивно понятен, что способствует быстрой адаптации персонала к работе с ним. Графическое представление данных на экране (рисунок 2.16) позволяет визуализировать информацию и принимать оперативные решения. В результате эффективность работы предприятия возрастает, а затраты на логистику и мониторинг уменьшаются.

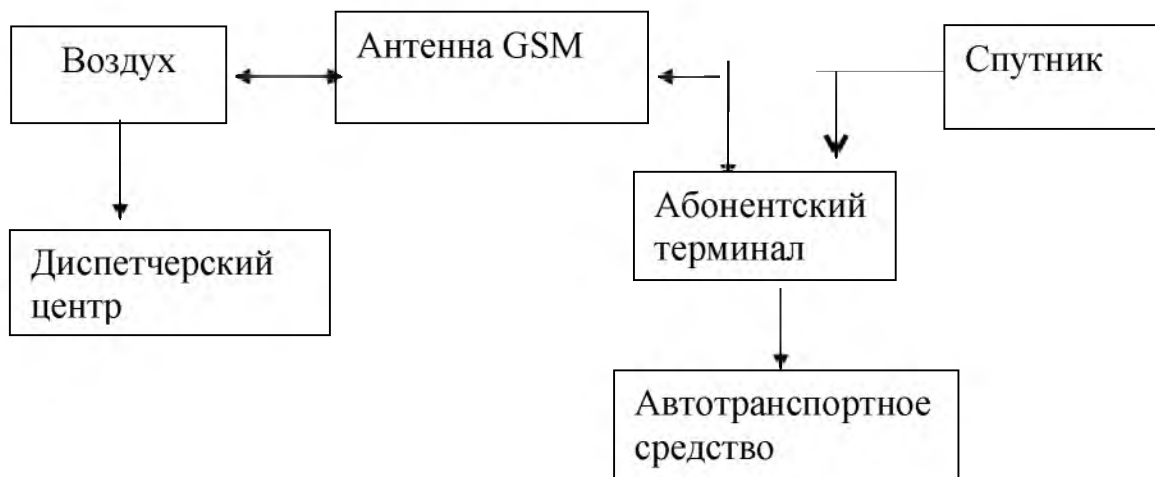


Рисунок 2.16 - Схема работы мониторинга ГЛОНАСС

Главное меню -это элемент управления рабочего стола, предназначенный для быстрого доступа к программам, документам, средствам настройки и поиска, избранным папкам и объекта (рисунок 2.17).

В данной форме пользователь может ввести необходимые данные о заказе, такие как имя клиента, адрес отправления и назначения, а также дату и время предполагаемой доставки. Подсистема «Объёмы» позволяет пользователю отслеживать текущие и прогнозируемые объёмы грузоперевозок, связывая их с поступившими заказами, что способствует оптимизации работы компании. Важным аспектом является и подсистема «Техническое обслуживание», которая обеспечивает регулярный мониторинг состояния транспортных средств, напоминая о необходимости проведения ТО и ремонта, что значительно увеличивает безопасность перевозок и снижает количество ошибок, связанных с техническим состоянием машин. Подсистема «Отчёты» генерирует аналитические сводки по выполненным перевозкам, включая информацию о затратах и прибыли, что позволяет управлению компании

принимать обоснованные решения. Наконец, подсистема «Маршрут и месторасположение» предоставляет пользователю возможность в реальном времени отслеживать местоположение транспортных средств, оптимизируя маршруты с учётом дорожной обстановки и других влияющих факторов. Таким образом, проектируемая информационная система создаёт целостный и эффективный инструмент для управления автотранспортными перевозками.

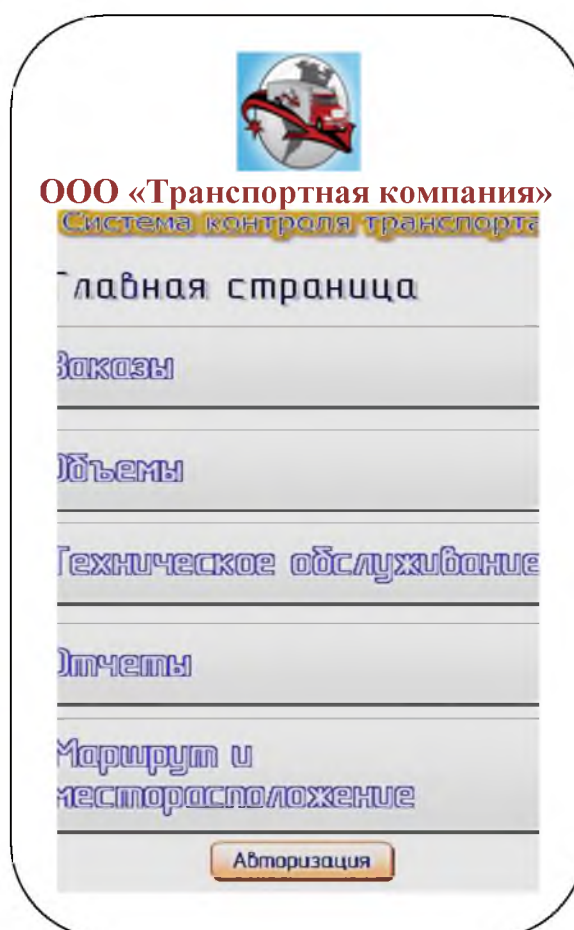


Рисунок 2.17 - Главное меню программы

Например, проектируемая ИС для автотранспортных перевозок в ООО «Транспортная компания» включает в себя подсистемы «Заказы», «Объёмы», «Техническое обслуживание», «Отчёты» и «Маршрут и месторасположение». При выборе пользователем геоинформационной системы компонента меню «Заказы», пользователю становится доступной для редактирования форма заполнения информации. Экран входа в приложение будет отображен при первом запуске приложения и при следующих запусках, если пользователь

вышел из него. Заголовок экрана входа содержит название приложения. В верхней части экрана входа находится логотип организации. В средней части находится элемент, позволяющий пользователю выбрать свою запись из предоставленного списка, не вводя данные вручную, а также поле для ввода пароля. Данные введенные в поле для ввода пароля маскируются. После входа в приложение и при повторных запусках с ранее введенными логином и паролем будет открываться экран со списком заказов назначенных для текущего пользователя прототип которого представлен на рисунке 2.18.

Информация о заказе:	
Дата:	Введите дату
Пункт назначения:	Куда
Пункт отправления:	Откуда
Груз:	Наименование
Объем отгрузки:	Введите объем
Расстояние:	Расстояние в км
Цена за литр:	Цена за 1 литр
Сумма:	Введите сумму
Транспорт:	Введите тип, модель и госномер

Добавить Отменить

Смена пользователя Выход

Рисунок 2.18 - Форма заполнения данных для учета заказа

После внесения информации данные заносятся по средствам нажатия на кнопку «Добавить», для отмены введенных в поля формы данных следует нажать на кнопку «Отменить».

В зависимости от выбора пункта меню пользователь попадает на окно формирования отчета. Например, в информационной системе учёта и анализа деятельности транспортной компании при выборе компонента меню «Отчёты» открывается окно выбора отчётов, где в зависимости от выбранного пункта

можно сформировать один из четырёх отчётов касательно либо самих заказов, либо технического состояния транспортного средства. Также имеется кнопка «Назад», возвращающая пользователя в основное меню. На данной форме пользователь вносит информацию о периоде проведения технического обслуживания, а также о стоимости проведения работ и закупки оборудования. Выполненные технические работы по ремонту отражаются в форме отчета (рисунок 2.19).



ООО «Транспортная компания»
 Система контроля транспорта
 Отчет производства
 технического обслуживания

Выберите период

От 25.12.2024 до 28.12.2024

Выберите транспортное средство:

Mercedes

Сформировать отчет по периоду

Затраты на обслуживание, руб.	Затраты на оборудование, руб.
3000	2521
2500	1845

Итого в указанном периоде:

Количество обслуживаний	Затраты на обслуживание, руб.	Затраты на оборудование, руб.
2	5500	4366

Смена пользователя

Выход

Авторизован.
 Пользователь: Ибрагим И. И.

Рисунок 2.19 - Отчет о выполненном техническом обслуживании

Данные формируют после окончательного внесения всей информации за счет нажатия кнопки «Добавить», а для корректировки и удаления соответственно нажимается кнопка «отмена». Создавая мобильное приложение изначально был определен плагин «Яндекс Карты» с учетом среды

проектирования AndroidStudio за счет которых появляется возможность четкого построения маршрута следования по каждой единице транспортных средств, на котором будет видна точка нахождения в определенный момент времени (рисунок 2.20).



Рисунок 2.20 - Окно указания маршрута транспортного средства

Нельзя забывать, что технические характеристики включаются в техническое задание на разработку приложения и служат дорожной картой в период проектирования.

Технические характеристики для функционирования приложения в нашем случае определены следующие: пользователю потребуется мобильное устройство с предустановленной операционной системой Androidc версией не

менее 4.0.3 и уровнем API не менее 17.0. Данным техническим требованиям соответствует 97% устройств.

Последним шагом в разработке мобильного приложения является его публикация, которая представляет собой процесс подготовки мобильного приложения для установки на устройства пользователей и этот процесс включает в себя две основные задачи: подготовка к публикации.

Создается релиз-версия приложения на базе Android. Релиз-версия приложения поставляется через один или несколько различных каналов распространения. Существует несколько способов, помощью которых Android приложение может быть доставлено пользователям: через web-сайт.

Файл приложения может быть доступен для загрузки на web-сайте. Загрузив файл, пользователи могут самостоятельно установить приложение. Для установки приложения самостоятельно, пользователь должен обладать некоторой квалификацией.

Среди работников ООО «Транспортная компания» возможно распространение мобильного приложения через IT отдел организации. Для установки и обновления приложения пользователям необходимо воспользоваться приложением GooglePlay. Мобильное приложение так же может быть установлено удаленно и обновлено автоматически.

Для публикации мобильного приложения был выбран сервис GooglePlay, который позволяет ограничить распространение приложения заданной группой пользователей (работников организации), наиболее быстро производить обновление приложения на мобильных устройствах пользователей, а также следить за статистикой критических ошибок приложения.

Таким образом, разработанное мобильное приложение для организации информационного обмена между водительским составом ООО «Транспортная компания» и CRM системой позволит систематизировать деятельность всего подвижного состава, а так же заметно снизить расход ГСМ.

В ходе проектирования были сформированы требования, спроектирован дизайн и создано приложение, реализующее следующие функции: оповещение

пользователя мобильного приложения о заказах, зарегистрированных в CRM системе предприятия: о поступлении новых заказов и изменении деталей ранее полученных заказов; просмотр списка заказов, привязанных к пользователю; просмотр детальной информации по заказу; быстрый набор номера клиента из данных заказа; формирование отчетов о техническом обслуживании автотранспорта.

Приложение было протестировано на стандартных эмуляторах включенных в поставку программных средств для разработки мобильных приложений Android (VisualStudioEmulatorforAndroid и Android SDK Emulator), взятых из SDK Android, так и на реальных устройствах, используемых в автотранспортной организации.

3 Экономическая эффективность проекта

3.1 Обоснование методики расчета экономической эффективности

Эффективность информационных систем можно рассматривать в узком и широком смыслах. В широком смысле - это влияние информационных ресурсов на качество принимаемых решений для достижения целей организации. В узком смысле - обеспечение информационных потребностей при помощи информационной системы для управления предприятием с наименьшими затратами [18,с.33].

В условиях современного рынка, где информация становится ключевым активом, важность качественно функционирующих информационных систем нельзя недооценивать. Эти системы не только оптимизируют внутренние процессы, но и обеспечивают конкурентные преимущества за счёт скорости и точности анализа данных. К примеру, внедрение ERP-систем позволяет интегрировать различные бизнес-процессы, что способствует повышению общей продуктивности.

Кроме того, актуальным является вопрос обеспечения безопасности данных, так как риски нарушения конфиденциальности и утечки информации могут значительно подорвать доверие клиентов и партнеров.

Следовательно, успешное руководство организацией требует комплексного подхода к разработке и внедрению информационных систем, охватывающего как технологические, так и управленческие аспекты. Таким образом, достижение гармонии между затратами и эффективностью становится неотъемлемой частью стратегического управления в современных условиях [18,с.33].

Выбор методики оценки эффективности информационной системы зависит от конкретной компании, деятельности и приоритетов. При выборе метода оценки эффективности ИС также стоит учитывать, что он должен быть максимально универсальным и оценивать не только положительный эффект от системы, но и затраты на её создание и внедрение.

Выделим некоторые популярные методы оценки эффективности программного продукта:

- котловой метод основан на определении соотношения объёмов вложений в программное обеспечение, включая внедрение и сопровождение, с размерами предприятия и направлениями его бизнеса;

- метод функциональной точки позволяет приблизительно оценить стоимость создания и внедрения ИС в зависимости от требований пользователя;

- метод ТСО (совокупная стоимость владения) даёт возможность понять структуру расходов на ИС, выявить проблемные места и обеспечить обратную связь для управления затратами;

- потребительский индекс предполагает оценку в виде индексов, которые отражают положительные изменения в работе организации;

- EVS - метод предполагает оценку положительного эффекта от использования программного обеспечения для компании. В качестве показателей для оценки используют рост доходов, увеличение производительности труда, снижение рисков для компании, сокращение времени на выполнение задания;

- EVA - метод позволяет оценить фактическую прибыль, которую приносит использование программного обеспечения для компании;

- BSC- метод предполагает оценку эффективности ИС с помощью комплексной функции, которая состоит из набора показателей, учитывающих все аспекты деятельности компании (финансовые, маркетинговые и т.д.).

Комплексный подход позволяет учитывать различные аспекты и нюансы, влияющие на экономическую эффективность, поскольку каждый метод может восполнять недостатки другого. Например, комбинируя количественные и качественные методы оценки, можно получить более целостную картину, которая будет учитывать не только финансовые показатели, но и социальные, экологические и организационные аспекты и эксплуатации, включая расходы на обслуживание, обновления и обучение пользователей. Это создает более

полное представление о финансовом влиянии системы на предприятие и позволяет принимать более обоснованные решения.

Кроме того, важно учитывать не только количественные, но и качественные аспекты. Например, необходимо продумать, как система будет восприниматься конечными пользователями, какой будет степень ее интуитивной понятности и удобства в использовании. Удовлетворение пользователей напрямую влияет на эффективность работоспособности системы и, как следствие, на повышение производительности бизнеса.

В дополнение к затратам на разработку и эксплуатацию следует обратить внимание на риски, связанные с проектом. Оценка рисков поможет избежать непредвиденных ситуаций и даст возможность заранее разработать стратегии их минимизации. Найдя баланс между затратами, качеством и рисками, предприятие сможет создать систему, которая не просто отвечает текущим потребностям, но и готова к возможным изменениям в будущем, обеспечивая долгосрочную ценность и устойчивость в условиях динамично меняющегося рынка [11,с.43].

Принимая оптимальные стратегические решения о внедрении информационной системы (ИС), необходимо учитывать множество факторов, среди которых первостепенное значение имеет анализ рисков. Риски, связанные с проектом, могут варьироваться от неисполнения сроков и превышения бюджета до несоответствия системы требованиям организации.

Эффективное внедрение ИС может существенно повысить производительность труда, автоматизировав процессы и улучшив качество услуг. Поэтому организациям необходимо установить приемлемый уровень риска, взвешивая возможные потери от неоправданных затрат и влияние на долгосрочную стратегию развития. Обретение понимания того, что наличие рисков — это неотъемлемая часть любого инновационного проекта, позволит предприятию создать сбалансированный подход к внедрению ИС, обеспечивая устойчивый рост и адаптацию к меняющимся условиям рынка. В конечном итоге, грамотный риск-менеджмент становится залогом успешного перевода

бизнеса на новый уровень технологической зрелости.

В таких условиях предприятиям следует сосредоточиться на минимизации рисков и оптимизации затрат. Введение модульного подхода позволит начать с основного функционала, который способен решить наиболее критичные задачи бизнеса, обеспечивая тем самым быстрое возвратные инвестиции. Ключевым аспектом на этом этапе станет выбор надежного вендора, способного предложить гибкое решение, соответствующее текущим требованиям.

В процессе реализации проекта важно не только жестко контролировать бюджет и сроки, но и регулярно проводить анализ достигнутых результатов. Периодические ревизии позволят оперативно корректировать курс проекта, устранять возможные проблемы и избегать значительных отклонений от первоначального плана. Не менее важным будет вовлечение сотрудников в процесс внедрения, что способствует приемлемости новой системы и снижает потенциальное сопротивление изменениям[11,с.43].

Таким образом, даже в условиях ограниченного бюджета и ресурсов, можно добиться значительных результатов, создавая основу для дальнейшего развития информационной инфраструктуры компании и постепенно внедряя дополнительные модули, которые сделают систему более комплексной и эффективной.

3.2 Эффективность разработки

Для оценки объема ресурсов, затрачиваемых на проектирование информационной системы контроля транспортных процессов в ООО «Транспортная компания», необходимо учитывать несколько ключевых факторов. Во-первых, это начальные инвестиции, включающие стоимость программного обеспечения, аппаратных средств и закупку лицензий. Кроме того, следует выделить затраты на обучение персонала, которые составляют важную часть бюджета, так как успешное внедрение системы зависит от уровня

квалификации сотрудников.

Во-вторых, важным аспектом является определение эксплуатационных затрат, которые включают в себя расходы на техническую поддержку, обновления системы и регулярное обслуживание оборудования. Эти затраты должны быть прогнозируемыми и учтенными в долгосрочной перспективе.

Кроме того, экономическая эффективность разработки системы может оцениваться через такие параметры, как уменьшение затрат на логистику, улучшение управления запасами и увеличение скорости обработки заказов. Эти показатели позволяют не только оценить непосредственный эффект от внедрения системы, но и определить её влияние на общий финансовый результат предприятия. Таким образом, комплексная оценка ресурсов и их сопоставление с экономическими результатами реализует стратегию адаптации и оптимизации процессов внутри компании.

Ключевым аспектом анализа является выявление конкретных показателей, таких как уменьшение времени на выполнение задач, сокращение числа ошибок и улучшение качества обслуживания клиентов. Сравнительный анализ может осуществляться на основе предварительных данных, собранных до внедрения, и последующих метрик, отражающих работу подразделения после интеграции системы. В нашем случае целью является оценка объема ресурсов, затрачиваемых на проектирование системы контроля транспортных процессов в транспортной компании [27,с.102]. В первую очередь проведем расчет трудоемкости работ по проектированию, учитывая срок выполнения каждого этапа (таблица 3.1).

Таблица 3.1 – Структура работ по созданию информационной системы

Этап	Содержание работ	Исполнители	Длительность, дней	Загрузка, дней	Загрузка, %
1.	Определение направления проектирования				
1.1	Постановка задачи	1	1	1	0,1
1.2	Обзор программ-аналогов	1	3	3	0,4

Продолжение таблицы 3.1

1.3	Подбор и изучение методической базы	1	9	9	0,7
	Итого по этапу	1	13	13	1,2
2	Научно-исследовательская работа				
2.1	Определение методик проектирования	1	6	6	0,7
2.2	Изучение структуры входных и выходных данных проекта	1	3	3	0,4
2.3	Обоснование необходимости проектирования	1	7	7	0,7
	Итого по этапу	1	16	16	1,8
3	Постановка технического задания				
3.1	Выявление требований к информационному обеспечению	1	7	7	1,0
3.2	Выявление требований к программному обеспечению	1	7	7	1,0
3.3	Выбор программных средств реализации проекта	1	3	3	0,4
3.4	Написание технического задания	1	2	2	0,1
	Итого по этапу	1	19	19	2,5

Далее проведем оценку трудоемкости проектирования ($Q_{\text{проект}}$) [18,с.19]:

$$Q_{\text{проект}} = \frac{Q_a * n_{\text{сп}}}{n_{\text{KB}}} \quad (3.1)$$

где, Q_a —уровень сложности проектирования программного продукта

(принимаем за 280 человеко-часов);

$n_{\text{сн}}$ - коэффициент сложности проектирования информационной системы (принимаем значение 0,8);

$n_{\text{КВ}}$ - коэффициент квалификации разработчика (студента) (принимаем значение 0,8, так как стаж менее 2х лет).

Таким образом, получим следующие данные:

$$Q_{\text{проект}} = \frac{280 * 0,8}{0,8} = 280 \text{ чел/час.}$$

Затраты труда на программирование определяют время выполнение проекта, разделяемое на следующие временные интервалы: время на разработку алгоритма, на непосредственное создание базы данных и написание программного продукта, время на написание сопроводительной документации:

$$Q_{\text{пр.}} = t_1 + t_2 + t_3 \quad (3.2)$$

где, t_1 - время на разработку алгоритма;

t_2 - время на создание базы данных и написание программного продукта;

t_3 - время для написание сопроводительной документации.

В первую очередь определим трудозатраты на алгоритмизацию задачи, используя коэффициент затрат на алгоритмизацию (n_a), равный отношению трудоемкости разработки алгоритма по отношению к трудоемкости его реализации при программировании, откуда [18,с.28]:

$$t_1 = n_a * t_2 \quad (3.3)$$

Значение коэффициента затрат на алгоритмизацию может находиться в интервале значений 0,1 до 0,5. Как правило, его выбирают равным $n_a = 0,3$. Затраты труда на создание сопроводительной документации будет определено суммой затрат труда на выполнение каждой работы каждого отдельного этапа.

Коэффициент затрат на проведение тестирования отражает отношение затрат труда на тестирование программы по отношению к затратам труда на ее разработку и может достигать значения 50%.

Коэффициент коррекции программы при ее разработке отражает увеличение объема работ при внесении изменений в алгоритм, изменения состава и структуры входной и выводимой информации. На практике, при разработке программы в среднем вносится 3-5 коррекции, каждая из которых ведет к переработке 5-10 % программы.

Коэффициент коррекции программы выбирают на уровне $n_u = 0,3$.

Коэффициент затрат на написание документации отражает отношение затрат труда на создание сопроводительной документации по отношению к затратам труда на разработку программы.

Коэффициент может составить: $n_q = 0,35$.

Объединив полученные значения коэффициентов затрат, определяют затраты труда на написание программного продукта составят [18,с.33]:

$$t_2 = \frac{Q_{\text{пр.}}}{n_a + 1 + n_m + n_u + n_q} \quad (3.4)$$

где, n_m - коэффициент затрат на проведение тестирования, принимаем значение на уровне $n_m = 0,3$;

n_u - коэффициент коррекции программы, принимаем значение на уровне $n_u = 0,3$;

n_q - коэффициент затрат на написание документации, принимаем значение на уровне $n_q = 0,35$.

Значение t_3 можно определить, если ввести соответствующие коэффициенты к значениям затрат труда на непосредственно программирование. Проведем расчеты:

$$t_2 = \frac{280}{0,3 + 1 + 0,3 + 0,3 + 0,35} = 112 \text{ чел. час.}$$

$$t_3 = 112 * (0,3 + 0,3 + 0,35) = 110 \text{ чел. час.}$$

$$t_1 = 0,3 * 110 = 34 \text{ чел. час.}$$

Затраты труда на внедрение программного обеспечения зависят от времени на осуществление опытной эксплуатации, которое согласовывается с заказчиком и, зачастую имеет значение в один месяц или 22 человеко-дня.

При 8-ми часовом рабочем дне этап внедрения может потребовать 176 чел-часов. Формулу для нахождения затрат труда на программирование можно записать следующим образом [18,с.46]:

$$Q_{\text{пр.}} = t_2 * (n_a + l + n_m + n_u + n_q) \quad (3.5)$$

Подставляя полученные данные, определяют общее значение трудозатрат для выполнения проекта:

$$Q_p = Q_{\text{пр.}} + t_i \quad (3.6)$$

где, t_i - затраты труда на выполнение i -го этапа проекта.

Подставив данные в формулу расчета, получаем:

$$Q_p = 280 + 34 + 112 + 110 = 536 \text{ чел/час.}$$

Затраты на проектирование включают в себя затраты на заработную плату разработчикам, затрат на закупку или аренду требуемого оборудования, затрат на организацию рабочих мест, а также затрат на накладные расходы, которые рассчитываются по формуле вида:

$$C = C_{\text{зп}} + C_{\text{об}} + C_{\text{орт}} + C_{\text{эл}} + C_{\text{накл}} \quad (3.7)$$

где, $C_{\text{зп}}$ - заработная плата исполнителей;

$C_{\text{об}}$ - затраты на обеспечение необходимым оборудованием;

$C_{\text{орт}}$ - затраты на организацию рабочих мест;

$C_{\text{эл}}$ - затраты на электроэнергию;

$C_{\text{накл}}$ - накладные расходы.

Затраты на выплату разработчику заработной платы определяется следующей формулой [18,с.59]:

$$C_{\text{зп}} = C_{\text{зп.осн.}} + C_{\text{зп.доп.}} + C_{\text{зп.отч.}} \quad (3.8)$$

где, $C_{\text{зп.осн.}}$ - основная заработанная плата;

$C_{\text{зп.доп.}}$ - дополнительная заработная плата;

$C_{\text{з.отч}}$ - отчисление с заработной платы.

$$C_{\text{зп.осн.}} = T_{\text{пр}} * \text{Окл}_{\text{дн}} \quad (3.9)$$

где, $T_{\text{пр}}$ - число дней, отработанных исполнителем проекта на реализации проекта;

$\text{О}_{\text{дн}}$ - дневной оклад исполнителя.

При 8-и часовом рабочем дне он рассчитывается по формуле[18,с.70]:

$$\text{Окл}_{\text{дн}} = (\text{Окл}_{\text{мес}} * 8) / T_{\text{мес}} \quad (3.10)$$

где, $\text{О}_{\text{мес}}$ - месячный оклад разработчика;

$T_{\text{мес}}$ - фонд месячного времени.

Подставим данные для расчетов:

$$\text{Окл}_{\text{дн}} = \frac{32000 * 8}{162} = 1580 \text{ руб.}$$

$$C_{\text{зп.осн.}} = 67 * 1580 = 105860 \text{ руб.}$$

$$C_{\text{зп.доп.}} = C_{\text{зп.осн.}} * 0,2 = 105860 * 0,2 = 21172 \text{ руб.}$$

$$C_{\text{зп.отч.}} = (C_{\text{зп.осн.}} + C_{\text{зп.доп.}}) * \text{Отч}_{\text{соц}} = 38109,6 \text{ руб.}$$

$$C_{\text{зп}} = 105860 + 21172 + 38109,6 = 165141,6 \text{ руб.}$$

Расчет затрат на электроэнергию производится по действующим для ООО «Транспортная компания» тарифам, согласно выбранной формулы:

$$C_{эл} = N_i * t_i * g_i * T_o \quad (3.11)$$

где, N_i - установленная мощность i -го вида технических средств, кВт;

t_i - время работы i -го вида технических средств, час (67*8=536 час.);

g_i - коэффициент использования установленной мощности оборудования;

T_o - тариф на электроэнергию, руб./кВтч.

Подставив в формулу значения, получаем:

$$C_{эл} = 0,75 * 536 * 0,7 * 5,45 = 1533,63 \text{ руб.}$$

$$C_{накл} = C_{зп.осн.} * 0,35 = 105860 * 0,35 = 37051 \text{ руб.}$$

Затраты, которые пойдут на обеспечение необходимым оборудованием по предварительным расчетам экономического отдела транспортного предприятия составят порядка 570 000 руб., а затраты на организацию рабочих мест (аренда рабочего места) составляет 500 руб./день, тогда получается из расчетов всего за период – 33 500 руб.

Таким образом, подставив все значения, найдём конечную сумму:

$$C = 165141,6 + 570000 + 33500 + 1533,63 + 37051 = 807\,226,23 \text{ руб.}$$

Прибыль предприятия (Pr), полученная из-за ликвидации потерь топлива, а также оптимизации маршрута, вследствие внедрения программного продукта, а именно простейшего мобильного приложения будет только возрастать и составит примерно 15 % от выручки, что примерно будет равно 1 020 000 руб.

Таким образом, срок окупаемости составляет:

$$Окуп = \frac{C}{Pr} = \frac{807226,23}{1020000,0} = 0,79 \text{ года} \quad (3.12)$$

Финансовая эффективность внедрения проекта составит:

$$\text{Эф} = \text{Пр} - \text{С} * 0,15 \quad (3.13)$$

Подставив значения в формулу расчета, получаем:

$$\text{Эф} = 1\,020\,000,0 - 807\,226,23 * 0,15 = 319\,160,6 \text{ руб.}$$

Произведенные расчеты показывают, что эффективность от внедрения информационной системы с элементами геоинформационных технологий для транспортной компании составит более 319 тысяч рублей, а срок окупаемости менее одного года.

Заключение

Система управления плановыми перевозками с использованием ГИС-технологий - это комплекс мер, включающий анализ существующих и планирование новых маршрутов доставки, а также управление активами.

В ходе изучения теоретического материала было выделено следующее, а именно то, что информационно-навигационная система позволяет:

- планировать маршруты, т.е. система работает с электронной картой, обеспечивает поиск объектов и определение оптимального маршрута. При составлении расписания можно учесть такие факторы, как поломки техники, ограничения на проезд в определённое время суток, время погрузки и выгрузки и другие;

- управлять активами, иными словами, навигационная система обеспечивает визуализацию точного местонахождения транспортных средств, персонала и событий в любой момент времени и информирует о возникающих задержках в осуществлении доставки;

- выполнять интеграционную функцию посредством того, что информационная система с элементами геоинформационных технологий увязывает системы управления проектами, CRM и финансово-учётные системы с местоположением объектов, что позволяет принимать решения с учётом финансовых и других факторов.

Система управления плановыми перевозками с использованием навигационных-технологий является примером востребованного на сегодняшний день решения программного обеспечения, которое может значительно упростить и улучшить процесс автоматизации перевозок.

ГИС-системы обеспечивают работу с электронной картой, поиск объектов, определение оптимального маршрута. При составлении расписания ГИС позволяет учесть такие факторы, как поломки техники, ограничения на проезд в определённое время суток, время погрузки и выгрузки и другие.

Современный человек не представляет своей жизни без информационных

систем и технологий, они проникли во все сферы современного общества и отрасли производства. Одна из таких отраслей – транспорт, где ГИС можно эффективно применять для управления транспортной инфраструктурой. Основанием для применения ГИС в транспортной отрасли является то, что почти все типы данных обладают пространственной составляющей [2,с.91].

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель - проектирование ГИС по контролю над транспортом компании.

За объект исследований и разработки мобильного приложения было выбрано ООО «Транспортная компания». В рамках этого были решены ранее определенные задачи, по каждой сделаны выводы: проведен анализ объекта и предмета исследования, проведено описание бизнес - процессов, представлена основная информация о рассматриваемой компании. Так как предприятие испытывает проблему в отчетности, как по использованию ГСМ, так и по техническому контролю, то рассмотрены именно эти бизнес процессы и их информационное обеспечение.

Рассмотрена информационная схема реализации бизнес-процесса контроля за транспортом, прописана последовательность действий, осуществляемых диспетчерской службой. Детальное внимание уделено основным проблемам бизнес - процесса технического состояния подвижного состава предприятия, среди которых основные - слабое информационное взаимодействие среди служб и отделов предприятия, а также отсутствие централизованного контроля, за процессами транспортировки и расходу топлива в автоматизированном режиме, которую можно решить созданием геоинформационной системы. Были выделены основные требования к системе.

Проведено описание предмета, анализ аналогов и обоснование выбора средств разработки ГИС контроля транспорта. Определена спецификация проектируемой системы – компьютерный мониторинг местонахождения и основных параметров транспортных средств, работающих на линии. Определены подсистемы и требования к функциональности информационной

системы. Решено, что контроль топлива будет осуществляться с использованием ёмкостных датчиков уровня топлива, а так как программной версии мобильного приложения, подходящей под требования предприятия не существует, была создана собственная версия на платформе Android, с использованием интегрированной среды разработки AndroidStudio.

В дальнейшем информационная система будет эксплуатировать механизмы СУБД и иметь возможность работать с картографическим материалом. Разработана методология проектирования геоинформационной системы. Уточнены требования к технической части проекта – к датчикам уровня топлива. Представлена схема работы системы мониторинга с использованием мобильного терминала.

Следующим этапом явилось проектирование макета геоинформационной системы с помощью среды разработки AndroidStudio. Система ориентирована на однопользовательскую работу с использованием мобильного устройства. Также система имеет возможности интеграции с облачными хранилищами, и интеграции с навигационными системами.

Неоспоримыми достоинствами разработанной геоинформационной системы является то, что она, в первую очередь, является универсальной для любых мобильных устройств, поддерживающих оперативную систему Android, во-вторых, защищенной от несанкционированного доступа и внешнего вмешательства, в-третьих, достаточно дешева для успешного внедрения даже в сложной экономической обстановке на каждое транспортное средство предприятия.

В качестве вариантов для внедрения была рассмотрена ГИС«MapView». Мобильная ГИС для ОС Android, которая позволяет подключать геопорталы пространственных данных (OpenStreetMap, Yandex и другие) и обрабатывать информацию со встроенного GPS/ГЛОНАСС-приёмника. Для разработки и отладки ГИС-приложений в ней будет использована библиотека Sequelize, с помощью которой было создано приложение на языке JavaScript, что в итоге обеспечило доступ к основным функциям сервисов.

Общение между водителями будет осуществляться при помощи P2P, т.е. смартфоны, объединенные в единую систему и взаимодействующие посредством сетевого P2P протокола, который обеспечивает возможность создания и функционирования сети равноправных узлов, их взаимодействия. При выборе среды проектирования мобильной информационной системы были рассмотрены 2 программных продукта: «Microsoft Visual Studio» и «Android Studio».

Проведено планирование комплекса работ по проектированию, проведен расчет трудоемкости работ, учитывая срок выполнения каждого этапа работ. В упрощенном варианте к разработке привлекается два человека: руководитель (преподаватель) и программист (студент). Суммарная длительность реализации проекта составляет 280 дней. Проведен анализ затрат, себестоимости и экономического эффекта разработки информационной системы.

Прибыль предприятия (Пр), полученная из-за ликвидации потерь топлива и технического обслуживания, а также оптимизации маршрута, вследствие внедрения программного продукта еще раз подчеркивает целесообразность разработки.

Таким образом, все поставленные задачи в ходе работы были выполнены, а цель работы можно считать достигнутой. Геоинформационная система может быть внедрена в деятельность транспортной компании.

Список литературы

1. Авторизованный справочник VisualStudio. [Электронный ресурс] URL: <https://code.visualstudio.com> (дата обращения: 12.11.2024).
2. Бабич, А.В. Эффективная обработка информации: учеб. пособие. - М.: ИнтернетТехнологий, БИНОМ, 2020. - 223 с.
3. Балдин, К.В. Информационные системы: учеб. / К.В. Балдин, В.Б. Уткин. - 7-е изд. - М.: Дашков и Ко, 2021. - 395 с.
4. Бирюков, А.Н. Лекции о процессах управления информационными технологиями: учеб. пособие. - М.: ИНФОТЕХ, 2020. - 215 с.
5. Браженская, Е.О., Бородина, Н.А. Плюсы и минусы информационных технологий // Научное сообщество студентов XXI столетия. Технические науки: сб. ст. по мат. XV междунар. студ. науч.-практ. конф. [Электронный ресурс]. URL: <http://sibac.info/archive> (дата обращения: 03.11.2024).
6. Геоинформатика: учеб./ под ред. А.Д. Иванников, В.П. Кулагин, А.Н. Тихонов, В.Я. Цветков. – М.: МАКС Пресс, 2021. – 349 с.
7. Гребенюк, Е.И. Технические средства информатизации: учеб. - 6-е изд., перераб. и доп. - М.: Академия, 2022. - 352 с.
8. Денисенко, А.Н. Компьютерная обработка информации. - М.: Медпрактика, 2024. - 252 с.
9. Заботина, Н.Н. Проектирование информационных систем: учеб. пособие. – М.: НИЦ Инфра-М, 2021. - 331 с.
10. Работа с конструктором классов [Электронный ресурс]. URL: <http://habr.com/company/trinion/b> (дата обращения: 01.12.2024).
11. Информационные технологии в экономике и управлении/ под ред. В.В. Трофимова. - М.: Юрайт, 2020. - 478 с.
12. Карчагина, Л.П. ГИС - проектирование: учеб. пособие. – М.: Информ, 2022. – 151 с.
13. Коновалова, Н.А., Капралов, Е.Г. Введение в ГИС. – М.: ООО «Библион», 2024. – 160 с.

14. Коноплева, И.А. Информационные технологии на транспорте: учеб. пособие для вузов. - 2-е изд. - М.: Проспект, 2021. -328 с.
15. Кроссплатформенная разработка в VisualStudio. [Электронный ресурс]. URL: <https://msdn.microsoft.com/ru> (дата обращения 22.10.2024).
16. Кроссплатформенная разработка приложений. [Электронный ресурс]. URL: <https://msdn.microsoft.com/ru>(дата обращения 27.10.2024).
17. Лисицкий, Д.В. Методические основы веб-картографии транспортных средств: справочное пособие. – М.: Геодезия и аэрофотосъемка, 2022. – 409 с.
18. Мезенцев, К.Н. Расчет экономической эффективности от внедрения автоматизированных информационных систем: учеб. пособие. - М.: Академия, 2021. - 208 с.
19. Михеева, Е.В. Практикум по информационным технологиям: учеб.пособие. - 9-е изд., стереотип. - М.: Академия, 2020. - 256 с.
20. Монахова, Г.Е. Информационные системы и технологии. Визуализация многомерных пространственных данных средствами геоинформационных систем: учеб. пособие / под ред. проф. М.Ю. Монахова. – Владимир: Изд-во ВлГУ, 2023. – 392 с.
21. Новожилов, О.П. Геоинформатика: учеб. - М. Юрайт, 2022. - 564 с.
22. Диаграммы классов VisualStudio.[Электронный ресурс] URL: <http://www.businessstudio.ru> (дата обращения: 25.10.2024).
23. Основы проектирования ГИС. [Электронный ресурс] URL: <http://citforum.ru/cfm/idef/idef3.shtml> (дата обращения: 08.09.2024).
24. Полуэктова, Н.Р. Разработка веб-приложений: учеб. пособие. – М.: Юрайт, 2021. - 204 с.
25. Просветов, Г.И. ГИС - программирование: учеб. пособие. - М.: Альфа-Пресс, 2020.-194 с.
26. Руководство по Node.js [Электронный ресурс] URL: [https://ruvds.com!\[\]\(9c85d1f27a3226615be8cfa1d77d77d6_img.jpg\)other/ee86eb4f-db9f-48d3-8094](https://ruvds.com/img/other/ee86eb4f-db9f-48d3-8094) (дата обращения 28.11.2024)
27. Синаторов, С.В. Информационные технологии в экономике: учеб.пособие. - М.: Дашков и Ко, 2022. – 456 с.

28. Трепачев, К.Е. Руководство по Node.js.[Электронный ресурс] URL: <https://metanit.com/web/nodejs> (дата обращения 21.11.2024).

29. Ширяев, С.А. Информационные технологии на транспорте: учеб.пособие. – Волгоград, 2022. – 372 с.

30. AndroidStudio - среда разработки приложений. [Электронный ресурс] URL: <http://computerologia.ru/android-studio/> (дата обращения 21.10.2024).

Приложение 1

Листинг программы

```
jdoodle.java
1 package com.example.leonxfire.ver001;
2 import android.app.FragmentTransaction;
3 import android.os.Bundle;
4 import android.support.design.widget.NavigationView;
5 import com.jjoe64.graphview.Graphview;
6 import com.jjoe64.graphview.series.BarGraphSeries;
7 import com.jjoe64.graphview.series.DataPoint;
8 import com.jjoe64.graphview.series.Series;
9 import android.support.v4.view.GravityCompat;
10 import android.support.v4.widget.DrawerLayout;
11 import android.support.v7.app.ActionBarDrawerToggle;
12 import android.support.v7.app.AppCompatActivity;
13 import android.support.v7.widget.Toolbar;
14 import android.view.Menu;
15 import android.view.MenuItem;
16 import android.app.Fragment;
17 import android.app.FragmentManager;
18 import android.view.View;
19 import android.widget.AdapterView;
20 import android.widget.EditText;
21 import android.widget.GridView;
22 import android.app.DatePickerDialog;
23 import android.widget.DatePicker;
24 import java.util.Calendar;
25 import android.app.TimePickerDialog;
26 import android.widget.TimePicker;
27 import com.example.leonxfire.ver001.Fragments.Fragment_New;
28 import com.example.leonxfire.ver001.Pages.Otchets.Otchet_Effect_Deyat; import com.example.leonxfire.ver001.Pages.Otchets.Otchet_Grafik_Zakaz; import com.example
29 import com.example.leonxfire.ver001.Pages.Otchets.Otchet_Zakaz_period;
30 import com.example.leonxfire.ver001.Pages.Otchets.Otchet_Zatrast;
31 import com.example.leonxfire.ver001.Pages.PageOtchet;
32 import com.example.leonxfire.ver001.Pages.PagePriceFuel;
33 import com.example.leonxfire.ver001.Pages.PageRashod;
34 import com.example.leonxfire.ver001.Pages.PageRemont;
35 import com.example.leonxfire.ver001.Pages.PageZakaz;
36 public class MainActivity extends AppCompatActivity
37 implements NavigationView.OnNavigationItemSelectedListener{ //Фрагменты страниц
38     PageOtchet P0tchet;
39     PageRashod PRashod;
40     PageRemont PRemont;
41     PageZakaz PZakaz;
42     Fragment_New FN;
43     PagePriceFuel PPrice;
```

Продолжение приложения 1

```

jdoodle.java
44 //Отчет
45 Otchet_Zakaz_period ZPeriod;
46 Otchet_Effect_Deyat Effect_Deyat;
47 Otchet_Grafik_Zakaz GZakaz;
48 Otchet_TO TO;
49 Otchet_Zatrati Zatrati;
50 private int mYear, mMonth, mDay;
51 @Override
52 protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
53 setContentView(R.layout.activity_main);
54 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar); setSupportActionBar(toolbar);
55 POtchet = new PageOtchet();
56 PRashod = new PageRashod();
57 PRemont = new PageRemont();
58 PZakaz = new PageZakaz();
59 FN = new Fragment_New(); PPrice = new PagePriceFuel();
60 ZPeriod = new Otchet_Zakaz_period();
61 Effect_Deyat = new Otchet_Effect_Deyat();
62 GZakaz = new Otchet_Grafik_Zakaz();
63 TO = new Otchet_TO();
64 Zatrati = new Otchet_Zatrati();
65 Calendar dateAndTime = Calendar.getInstance();
66 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
67 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
68 this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
69 drawer.setDrawerListener(toggle); toggle.syncState();
70 NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
71 navigationView.setNavigationItemSelectedListener(this);
72 }
73 @Override
74 public void onBackPressed() { DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout); if (drawer.isDrawerOpen(GravityCompat.START)) { drawer.closeDrawer(GravityCompat.START);
75 super.onBackPressed();
76 }
77 @Override
78 public boolean onCreateOptionsMenu(Menu menu) {
79 // Inflate the menu; this adds items to the action bar if it is present.
80 getMenuInflater().inflate(R.menu.main, menu);
81 return true;
82 }
83 public void Otchet_Grafik_Zakaz(View view){
84 FragmentTransaction ft = getFragmentManager().beginTransaction();
85 ft.replace(R.id.action_container, GZakaz);
86 ft.commit();
87 }

```

Продолжение приложения 1

```

jdoodle.java
87 }
88 public void Otchet_Zakaz(View view){
89     FragmentTransaction ft = getFragmentManager().beginTransaction();
90     ft.replace(R.id.action_container, ZPeriod) ;
91     ft.commit();
92 }
93 public void Otchet_TO(View view){
94     FragmentTransaction ft = getFragmentManager().beginTransaction();
95     ft.replace(R.id.action_container, TO);
96     ft.commit();
97 }
98 public void Otchet_Deyat(View view){
99     FragmentTransaction ft = getFragmentManager().beginTransaction();
100    ft.replace(R.id.action_container, Effect_Deyat);
101    ft.commit();
102 } public void Otchet_Uchet_Rashod(View view){
103     FragmentTransaction ft = getFragmentManager().beginTransaction();
104     ft.replace(R.id.action_container, Zatrats);
105     ft.commit();
106 }
107 // отобразим диалоговое окно для выбора даты
108 public void setDate(View v) {
109     final EditText currentDateTime = (EditText) findViewById(v.getId());
110     final Calendar cal = Calendar.getInstance() ;
111     mYear = cal.get(Calendar.YEAR) ;
112     mMonth = cal.get(Calendar.MONTH) ;
113     mDay = cal.get(Calendar.DAY_OF_MONTH) ;
114     DatePickerDialog datePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener () { @Override public void onDateSet(DatePicker view, int
115     String editTextDateParam = dayOfMonth + «.» +
116     (monthOfYear +1) + «.» + year;
117     currentDateTime.setText(editTextDateParam);
118 }
119 }, mYear, mMonth, mDay);
120 datePickerDialog.show();}
121 @SuppressWarnings(«StatementWithEmptyBody»)
122 public boolean onNavigationItemSelected(MenuItem item) {
123     // Handle navigation view item clicks here. int id = item.getItemId();
124     FragmentTransaction ftrans= getFragmentManager().beginTransaction();
125     if (id == R.id.nav_zakaz) {
126         ftrans.replace(R.id.action_container, PZakaz);
127     } else if (id == R.id.nav_Rashod) { ftrans.replace(R.id.action_container, PRashod);
128     } else if (id == R.id.nav_Remont) {
129         ftrans.replace(R.id.action_container, PRemont);

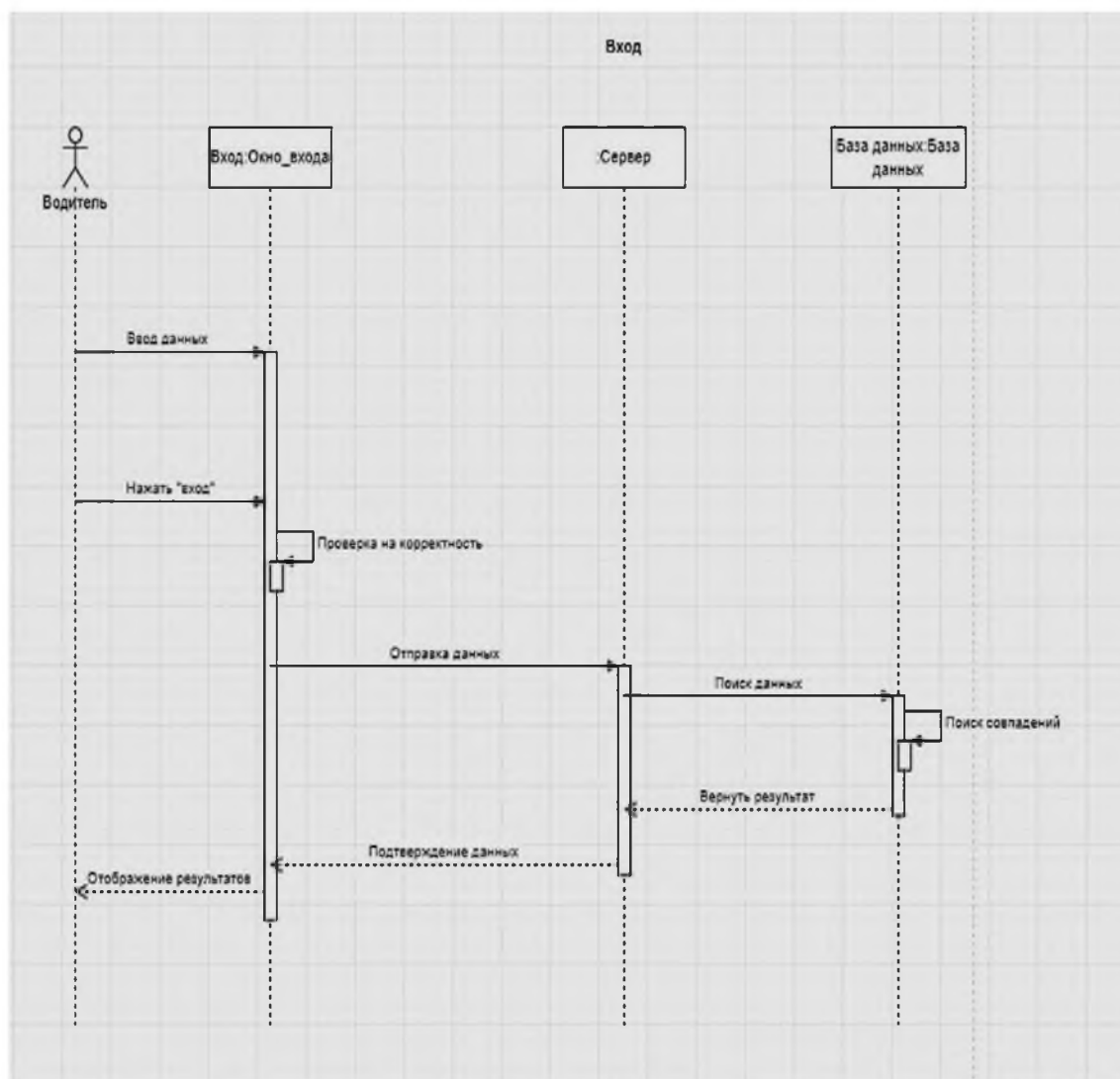
```

Продолжение приложения 1

```
jdoodle.java
118 public void setDate(View v) {
119     final EditText currentDateTime = (EditText) findViewById(v.getId());
120     final Calendar cal = Calendar.getInstance();
121     mYear = cal.get(Calendar.YEAR);
122     mMonth = cal.get(Calendar.MONTH);
123     mDay = cal.get(Calendar.DAY_OF_MONTH);
124     DatePickerDialog datePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() { @Override public void onDateSet(DatePicker view, int
125     String editTextDateParam = dayOfMonth + «.» +
126     (monthOfYear + 1) + «.» + year;
127     currentDateTime.setText(editTextDateParam);
128 }
129 }, mYear, mMonth, mDay);
130 datePickerDialog.show();
131 @SuppressWarnings(«StatementWithEmptyBody»)
132 public boolean onNavigationItemSelected(MenuItem item) {
133     // Handle navigation view item clicks here. int id = item.getItemId();
134     FragmentTransaction ftrans= getSupportFragmentManager().beginTransaction();
135     if (id == R.id.nav_zakaz) {
136         ftrans.replace(R.id.action_container, PZakaz);
137     } else if (id == R.id.nav_Rashod) { ftrans.replace(R.id.action_container, PRashod);
138     } else if (id == R.id.nav_Remont) {
139         ftrans.replace(R.id.action_container, PRemont);
140     } else if (id == R.id.nav_Otchet) { ftrans.replace(R.id.action_container, POtchet);
141     }
142     ftrans.commit();
143     DrawerLayout drawer = (DrawerLayout)
144     findViewById(R.id.drawer_layout);
145     drawer.closeDrawer(GravityCompat.START);
146     return true;
147 }
148 public void onTarget(View view) {
149     EditText edtTxt = (EditText) findViewById(view.getId());
150     // if (edtTxt.getText().length() > 0) { edtTxt.setText («»);
151     // }
152 }
153 }
154 }
155 }
```

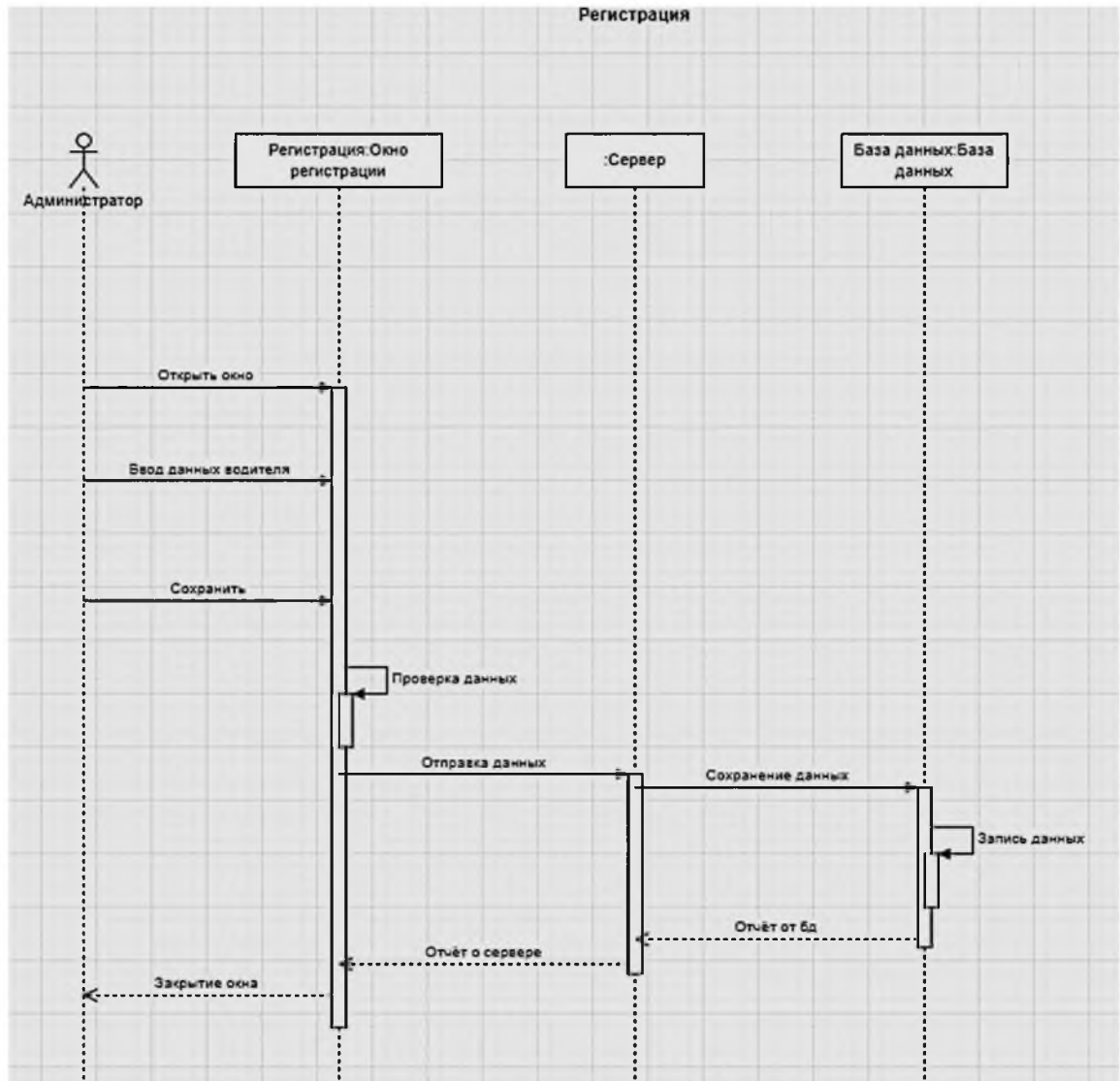
Приложение 2

Диаграмма последовательности «входа» для системного драйвера



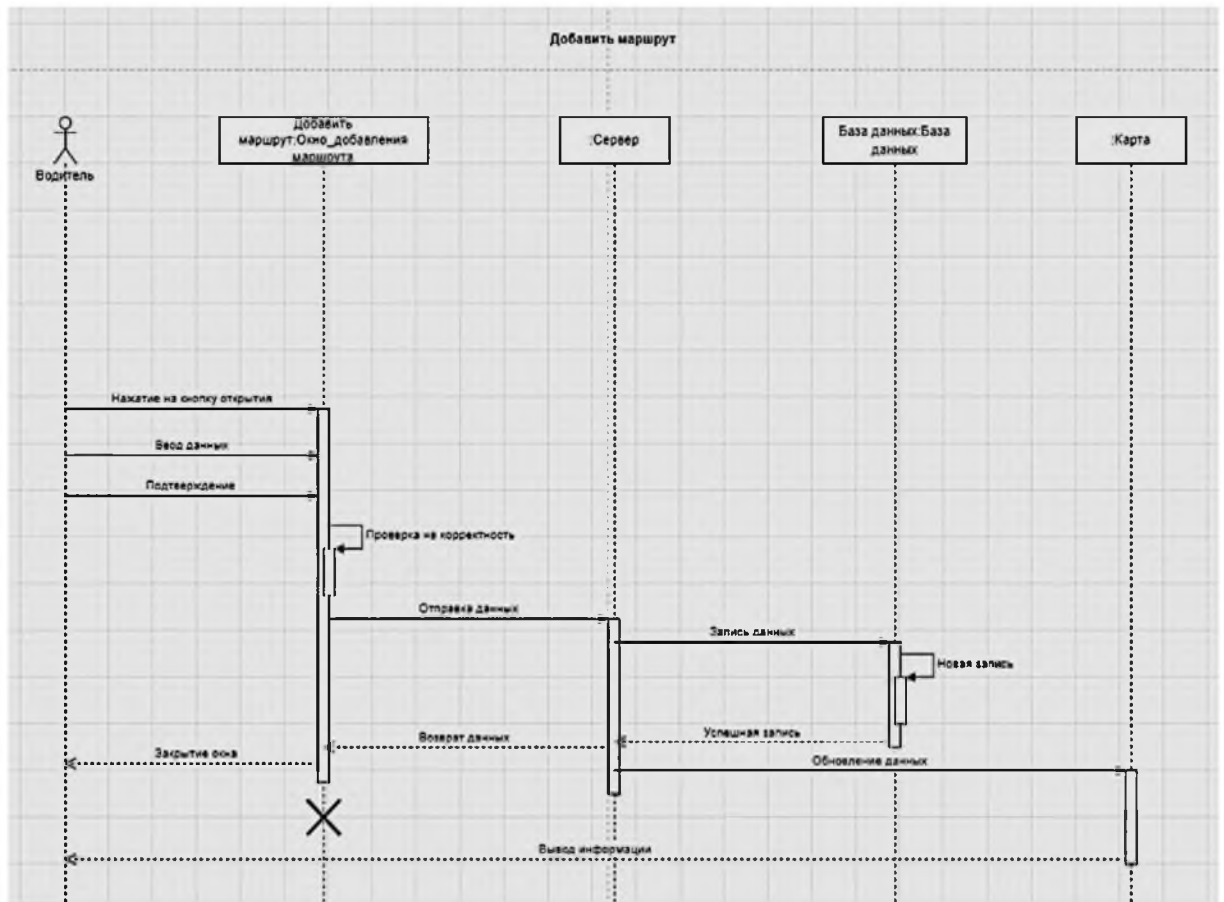
Приложение 3

Схема последовательности действий системного администратора «Регистрация»



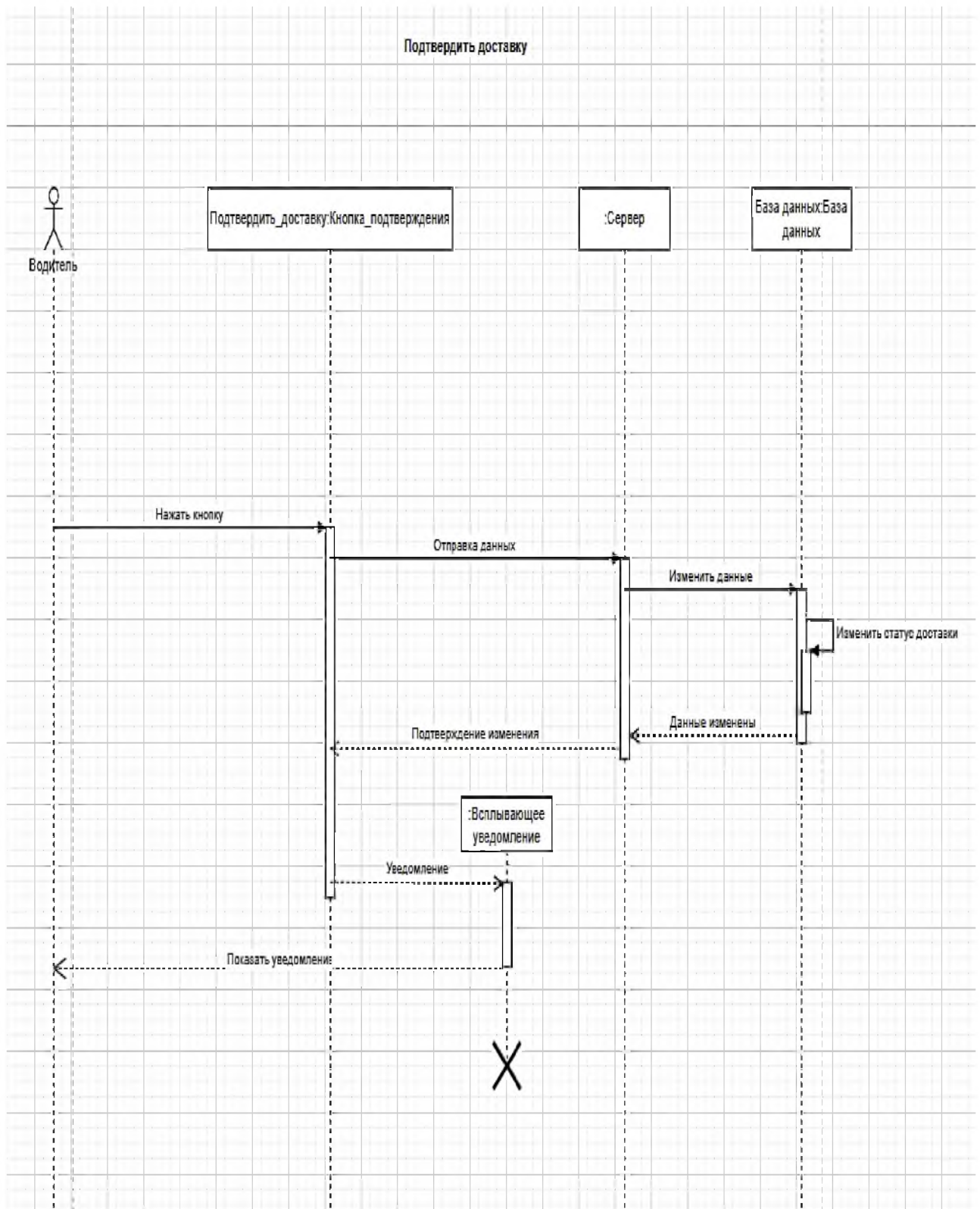
Приложение 4

Схема последовательности «Добавить маршрут» для системного драйвера

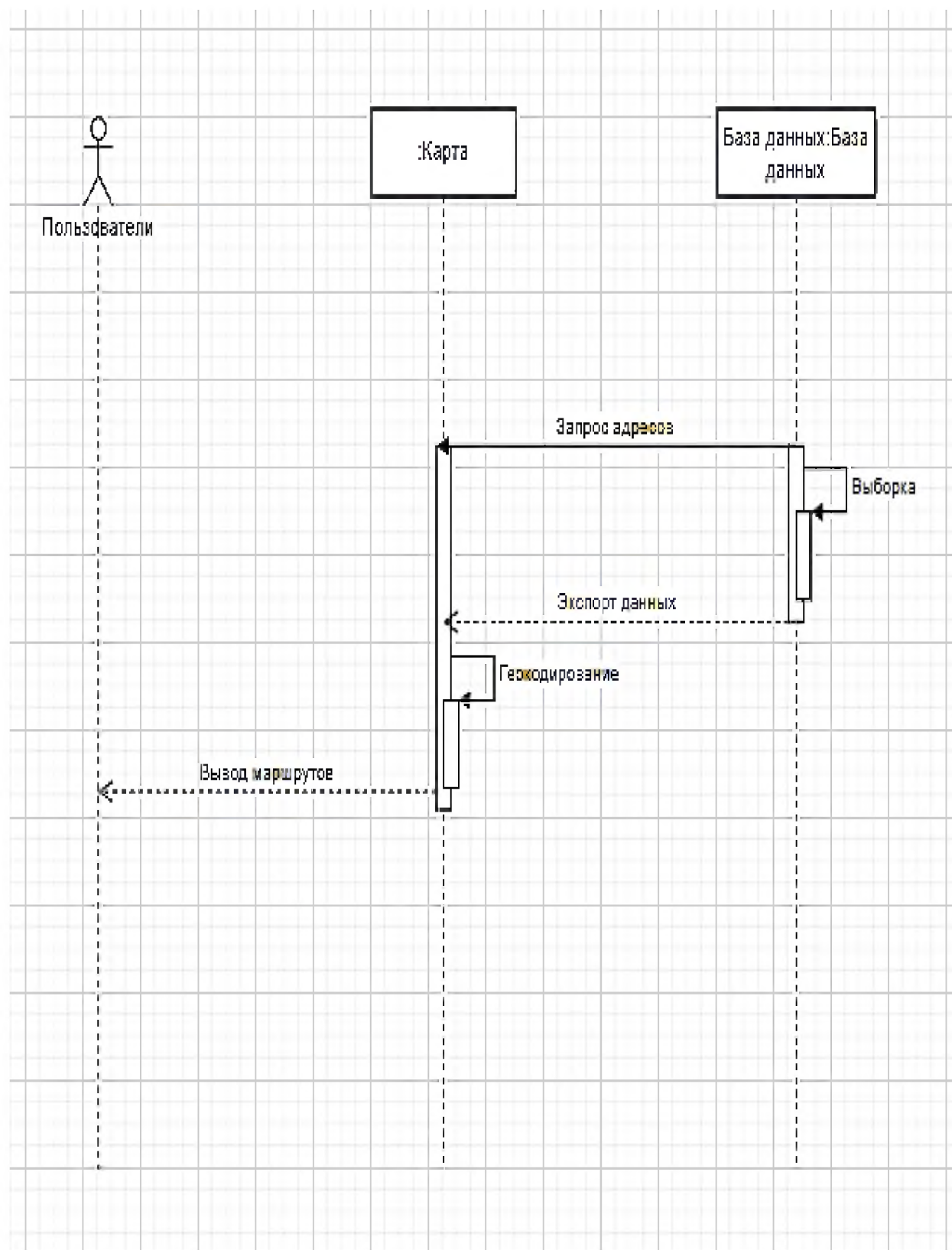


Приложение 5

Диаграммы последовательности

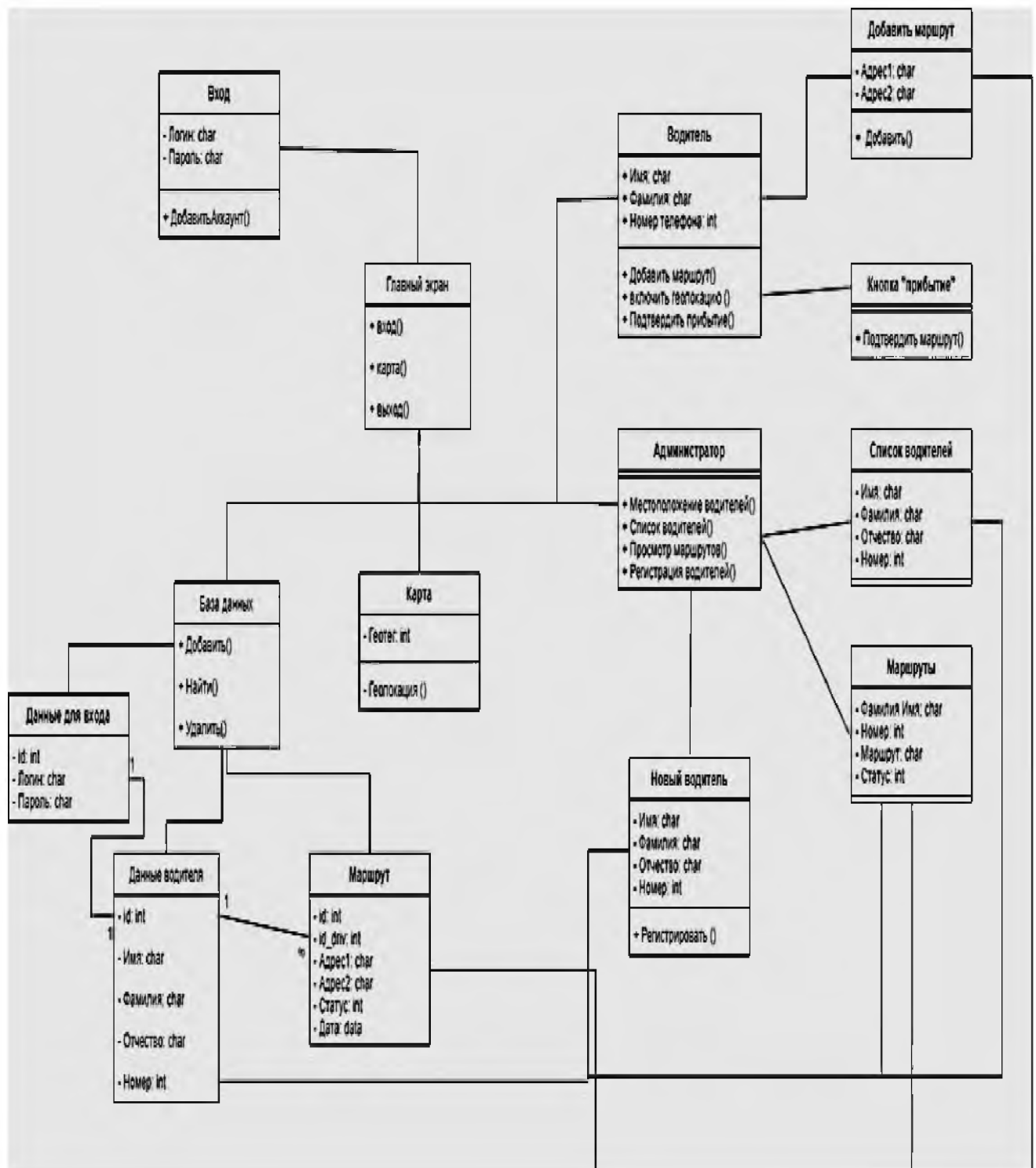


Продолжение приложения 5



Приложение 6

Диаграммы классов и развертывания



Продолжение приложения 6

