

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, доцент, к. т. н.

_____ К.А. Домбровский

« ____ » _____ 2017 г

Разработка компонентов автоматизированной системы управления работой
автотранспортного предприятия.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ- 09.03.01.2017.886 ПЗ ВКР

Руководитель проекта, доцент, к. т. н.

_____ Е. С. Ярош

« ____ » _____ 2017 г.

Автор проекта

студент группы КЭ-445

_____ А. С. Игнатов

« ____ » _____ 2017 г.

Нормоконтролер, ст. преп. каф.

«Электронные вычислительные
машины»

_____ С. В. Сясков

« ____ » _____ 2017 г.

Челябинск 2017

АННОТАЦИЯ

Игнатов А. С. Веб-приложение «Разработка компонентов автоматизированной системы управления работой автотранспортного предприятия». – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2017, 60 с., 39 ил., библиогр. список – 14 наим., 7 слайдов.

Анализ рынка приложений по автоматизации автотранспортного предприятия показал необходимость разработки дополнительных компонентов для автоматизации управления работой автотранспортного предприятия. В ходе выполнения выпускной квалификационной работы были сформулированы требования, которым должно удовлетворять приложение, выбрана технология его создания, разработана архитектура приложения, спроектирована база данных, осуществлена программная реализация.

Результатом выполнения работы является веб-приложение для автоматизации автотранспортного предприятия. Клиентская часть написана с помощью языков разметки HTML и CSS, серверная часть написана на языке программирования PHP. База данных создана с помощью MySQL.

					09.03.01.2017.886.00 ПЗ ВКР			
Изм.	Лист	№ докум.	Подпись	Дата	<i>Разработка компонентов автоматизированной системы управления работой автотранспортного предприятия</i>	Лит.	Лист	Листов
Разраб.		А. С. Игнатов						
Провер.		Е. С. Ярош					3	60
Реценз.						ФГАОУ ВО «ЮУрГУ (НИУ)»		
Н. контр.		С. В. Сяськов						
Утверд.		К.А.Домбровский						

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ.....	6
1.1 Назначение и цели создания	6
1.2 Требования к web-приложению	7
1.3 Обзор существующих приложений.....	8
2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ ДЛЯ WEB-ПРИЛОЖЕНИЯ	13
2.1 Выбор технологии для разработки web-приложения	13
2.2 Выбор аппаратного обеспечения	25
3 ПРОГРАММНАЯ РАЗРАБОТКА ДЛЯ WEB-ПРИЛОЖЕНИЯ	27
3.1 Архитектура и реализация.....	27
3.2 Организация работы web-приложения.....	29
4 ОРГАНИЗАЦИЯ БАЗЫ ДАННЫХ.....	31
5 РАЗРАБОТКА СОСТАВНЫХ ЧАСТЕЙ.....	41
5.1 Серверная часть	41
5.2 Клиентская часть	49
ЗАКЛЮЧЕНИЕ	58
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	59

									Лист
									4
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.886.00 ПЗ				

ВВЕДЕНИЕ

Сейчас сложно представить любой город без общественного транспорта и спецтранспорта. В больших городах чаще всего бывает не одно автотранспортное предприятие. Чтобы иметь конкурентное преимущество, автотранспортная компания должна предоставлять свои услуги так, чтобы именно ей заинтересовался заказчик. Зачастую при поступлении большого количества заказов компания не справляется с ними из-за обычной неорганизованности. Поэтому важно отслеживать график работы водителей и составлять максимально прибыльный план действий. Транспортные услуги являются одним из самых затратных бизнес-процессов. Поэтому крайне важно, чтобы внутренние процессы проходили чётко и без лишних затрат. Для этого используют современную вычислительную технику, которая позволяет увеличить производительность труда, повысить эффективность работы с документами и ускорить обмен информацией.

Изучение опыта работы автотранспортных предприятий показало, что одной из основных задач является уменьшение финансовых издержек. Эти издержки связаны с отсутствием чёткого графика работы водителей, не выполнением договоров между заказчиками и поставщиками, не эффективным использованием средств, выделяемых на ремонт и обслуживание, отсутствием оперативной информации. Решение этих проблем возможно с помощью приложения, которое позволяет:

- облегчить работу по учёту;
- быстро найти нужный документ;
- сократить время формирования документов.

В результате заказчику будет предоставлен более качественный сервис, повысится оперативность обслуживания.

					<i>09.03.01.2017.886.00 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ

1.1 Назначение и цели создания

Онлайн-приложение предназначено для оперативной и удобной работы различных пользователей.

Одна из отличительных особенностей системы – это возможность пользователя не только самостоятельно просмотреть список существующих услуг в фирме, но и сделать заказ. Приложения, находящиеся на рынке, предполагают, что клиент будет звонить, и ему будет назначаться менеджер, через которого будет происходить дальнейшее взаимодействие. Данная система предлагает, что клиент будет сам просматривать весь доступный парк автотранспорта и выбирать нужное ему. А так как система предполагает использование с мобильных платформ, то сделать заказ становится очень быстрым и простым действием.

Автотранспортное предприятие предлагает не только заказ автобусов и спецтехники, но и другие услуги, такие как прохождение техосмотра для автомобилей граждан.

Для заказа пользователю предоставляется возможность самому зайти в онлайн-приложение и выбрать то, что его интересует. Под заказом подразумевается, что пользователь выбирает автобус или спецтехнику по характеристикам, которые его устраивают. Например, для поездки 25 человек в другой город нужно выбрать автобус на 30 мест, и он будет дешевле, чем 40 мест.

Также на предприятии предусмотрено прохождение ТО для личного автомобиля. В большинстве станций ТО есть большие очереди. В приложении есть возможность записаться на определенное время. Для этого пользователь также должен зайти в приложение и выбрать удобное для себя время.

За каждым водителем закреплена определённая техника. В случае поломки машины водитель сообщает об этом менеджеру, который, в свою очередь, должен иметь свой интерфейс, позволяющий учесть факт ремонта

					<i>09.03.01.2017.886.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		6

машины. Это необходимо, чтобы заказчик не выбрал тот автомобиль, который по техническим причинам заказ выполнить не сможет.

Для достижения поставленных целей, необходимо решить следующие задачи:

- выявление необходимого функционала и требований к web-приложению;
- анализ существующих приложений;
- выбор технологий для веб-приложения;
- создание базы данных и серверной части;
- разработка интерфейса пользователя;
- разработка интерфейса менеджера;
- разработка интерфейса администратора.

1.2 Требования к web-приложению

Для приложения должно быть реализовано 4 интерфейса, в которых решаются различные задачи.

Интерфейс пользователя:

- возможность сделать заказ на автобус/спецтехнику;
- просмотр автопарка машин;
- запись на техосмотр.

Интерфейс менеджера:

- ввод графика работы водителей;
- обработка заказов;
- просмотр графика записи на техосмотр;
- добавление машин;
- проставление статусов автомобилей (занят, не исправен, ТО).

Интерфейс руководителя:

- возможность добавления новых пользователей и указание им прав;
- возможность добавить организацию/машину;
- редактирование заявок.

Интерфейс водителя:

- просмотр графика работы;
- просмотр отработанных смен и количество часов;
- декомпозиция контекстной диаграммы выглядит следующим образом.

1.3 Обзор существующих приложений

Программа «**КиберЛог**» запущена в 2011 году (рисунок 1.1) и на данный момент является одной из наиболее популярных программ, предлагаемых на рынке. Данное приложение является онлайн-системой управления бизнесом в сфере грузоперевозок. Цель системы – обеспечить своевременный обмен информацией, обеспечить документооборот между заказчиками, грузоперевозчиками и транспортно-экспедиционными компаниями в едином информационном пространстве, используя интернет-портал [1].



Рисунок 1.1 - Логотип «КиберЛог»

Достоинства:

- возможность создания договора, что при последующей работе позволяет упростить составления условий для заказа;
- ведение полного бухгалтерского учёта, что позволяет не приобретать дополнительных программных продуктов для расчёта зарплаты;

					<i>09.03.01.2017.886.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		8

- сама программа автоматически подтягивает в свою базу все законодательные документы для грузоперевозок;
- в программе присутствует чат между сотрудниками, это помогает быстро связаться друг с другом.

Недостатки:

- программа ориентирована на грузоперевозки, тяжело настроить на другой вид транспорта, из-за чего возникают проблемы с отчётностью;
- не ведётся технический учёт техники.

Отсутствуют возможности:

- для пользователя самостоятельно сделать заказ по электронным каналам связи;
- нет интерфейса для водителя.

Программа «АвтоПеревозки 4 AutoSoft» (рисунок 1.2) является десктопным приложением и рассчитана на учет автотранспортной техники собственного парка, получение документов, связанных с учетом, формирование и печать путевых листов, учет выполненных ремонтных работ и ТО, учет ГСМ, работы водителей, учет контрагентов и работы с ними, складской учет. Создаваемая на начальном этапе база данных включает в себя информацию о единицах техники, водителях, клиентах. С помощью программы «АвтоПеревозки» можно рассчитать и установить нормы расхода топлива для каждой единицы техники. Отчеты имеет гибкую систему настроек, и могут выдаваться за любой период времени и по заданным пользователем позициям: расходу топлива для конкретных автомобилей, выработке часов и т.д. [2].



Рисунок 1.2 - Логотип «АвтоПеревозки 4 AutoSoft»

Достоинства:

- ведётся учёт собственного парка транспортных средств с большим количеством возможностей. Возможна отчётность по каждому автомобилю отдельно;
- Присутствует учёт работ, выработки оборудования и расхода материалов, что позволяет вести учёт фактически выполненной работы и планирования заказов, также ведётся учёт общей выработки, расхода материала и простоя;
- Есть планирование ТО собственного парка на основе расчёта предыдущих пунктов.

Недостатки:

Приложение является только десктопным и нужно подбирать компьютер для работы с программой.

Отсутствуют возможности:

- записи на техосмотр автомобиля, не принадлежащего АТП;
- для пользователя самостоятельно сделать заказ по электронным каналам связи.

Приложение «ПК АвтоПлан» (рисунок 1.3) является комбинированным решением между онлайн-приложением и десктопным. С целью защиты данных программный комплекс и база данных устанавливаются локально на компьютеры заказчика под любую версию Windows: XP/7/8/10. У программы

									Лист
									10
Изм.	Лист	№ докум.	Подпись	Дата	09.03.01.2017.886.00 ПЗ				

- на начальном этапе эксплуатации в программу нужно корректно ввести все исходные данные, при ошибке могут выявиться серьёзные проблемы.

Отсутствуют возможности:

- просмотра заказчиком предоставляемых услуг без дополнительной авторизации;
- нет интерфейса водителя.

Краткие выводы по разделу один

Внедрение web-приложения эффективно скажется на: систематизации учета и хранения заказов на выполнение заявок на транспорт и ТО, оперативном доступе к документам, сокращении времени процедур согласования документов и принятия решений, повышении исполнительской дисциплины, сокращении непроизводственных затрат рабочего времени сотрудников и минимизации финансовых затрат на документооборот.

					<i>09.03.01.2017.886.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		12

2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ ДЛЯ WEB-ПРИЛОЖЕНИЯ

2.1 Выбор технологии для разработки web-приложения

Разрабатываемое приложение должно являться кроссплатформенным, доступ к нему должен осуществляться с устройств, имеющих подключение к интернету.

Один из плюсов веб приложений – отсутствие необходимости установки на устройство клиента объемного программного обеспечения, возможность работать под любыми операционными системами.

Для разработки клиентской части приложения используется связка языков HTML, CSS и JavaScript.

Для разработки серверной части приложения используют разные языки программирования. Для сравнения будут рассмотрены 3 языка (рисунок 2.1), такие как Python, Ruby, PHP, так как они являются основными конкурентами на рынке онлайн приложений.

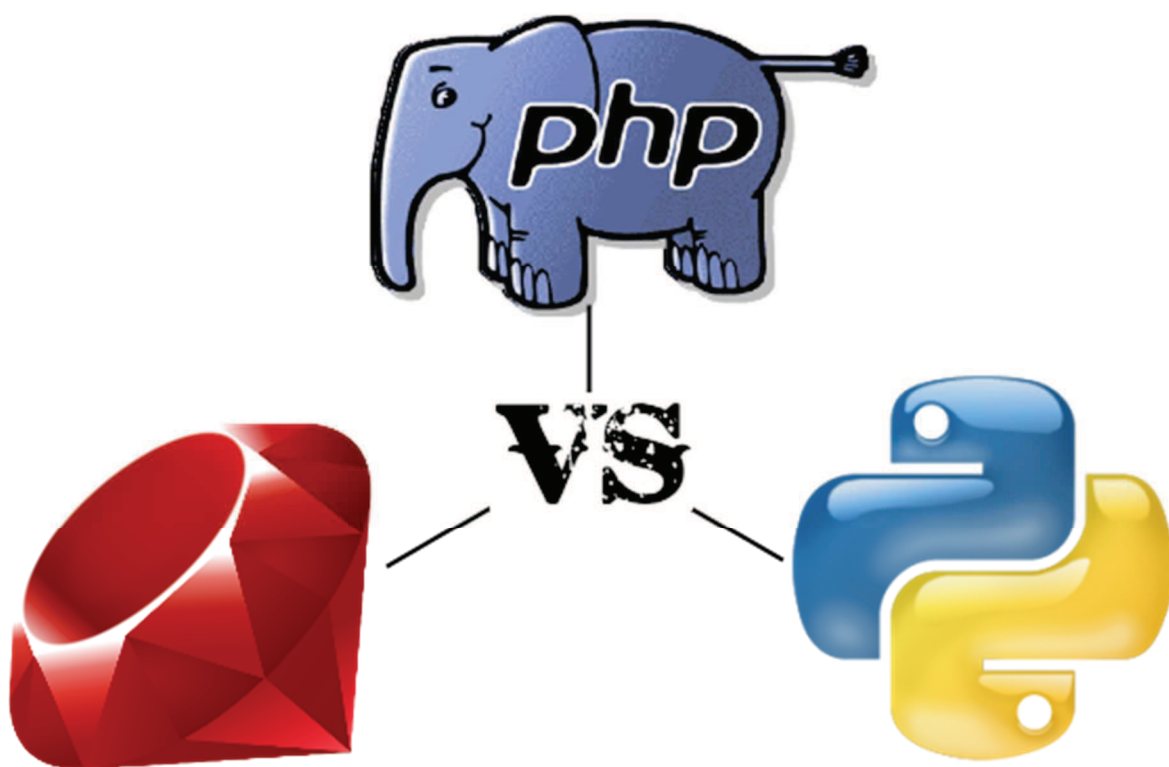


Рисунок 2.1 - Языки программирования PHP, Python, Ruby

Изм.	Лист	№ докум.	Подпись	Дата

09.03.01.2017.886.00 ПЗ

Лист

13

Python – это интерпретируемый, интерактивный, объектно-ориентированный язык программирования. Он включает в себя модули, исключения, динамическую типизацию, высокоуровневые динамические типы данных и классы. Python сочетает в себе отличную производительность с понятным синтаксисом. В нем реализованы интерфейсы ко многим системным вызовам и библиотекам, а также различным оконным системам и он расширяем с помощью C и C++. Python используется как язык расширения для приложений, которым нужен программный интерфейс. И наконец, Python – это кроссплатформенный язык: он работает на многих версиях Unix, на Mac и на компьютерах под управлением MS-DOS, Windows, Windows NT и OS/2.

Python является языком общего назначения, поэтому может применяться практически в любой области разработки ПО (standalone, клиент-сервер, Web-приложения) и в любой предметной области. Кроме того, Python легко интегрируется с уже существующими компонентами, что позволяет внедрять Python в уже написанные приложения.

Python имеет модули расширения, как стандартные, так и специфические. Стандартные модули расширения Python – это отлично спроектированная и неоднократно проверенная функциональность для решения задач, возникающих в каждом проекте по разработке ПО: обработка строк и текстов, взаимодействие с операционной системой, поддержка Web-приложений. Эти модули также написаны на языке Python, поэтому обладают его важнейшим свойством – кроссплатформенностью, позволяющей безболезненно и быстро переносить проекты с одной операционной системы на другую.

Достоинства Python:

- открытая разработка;
- простота в изучении, особенно на начальном этапе;
- особенности синтаксиса стимулируют программиста писать хорошо читаемый код;

- наличие средств быстрого прототипирования и динамической семантики;
- наличие большого сообщества, позитивно настроенного по отношению к новичкам;
- множество полезных библиотек и расширений языка можно легко использовать в своих проектах благодаря предельно унифицированному механизму импорта и программным интерфейсам;
- механизмы модульности хорошо продуманы и могут быть легко использованы;
- абсолютно всё в Python является объектами в смысле ООП, но при этом объектный подход не навязывается программисту.

Недостатки Python:

- не слишком удачная поддержка многопоточности;
- на Python создано не так уж много качественных программных проектов по сравнению с другими универсальными языками программирования;
- отсутствие коммерческой поддержки средств разработки (сейчас это решается при помощи покупки дополнительного программного обеспечения);
- изначальная ограниченность средств для работы с базами данных.

Ruby – интерпретируемый язык высокого уровня для быстрого и удобного объектно-ориентированного программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией и многими другими возможностями. Ruby близок по особенностям синтаксиса к языкам Перл и Эйфель, по объектно-ориентированному подходу к Smalltalk. Также некоторые черты языка взяты из Python, Лисп, Dylan и CLU. Кроссплатформенная реализация интерпретатора

языка является полностью свободной. Последней является версия 2.4.1, вышедшая в декабре 2016.

Ruby – полностью объектно-ориентированный язык. В нём все данные являются объектами, в отличие от многих других языков, где существуют примитивные типы. Каждая функция – метод.

У Ruby есть специальный веб-фреймворк, который очень облегчает работу с web-приложениями. Этот фреймворк реализует архитектурный шаблон Model-View-Controller для веб-приложений, а также обеспечивает их интеграцию с веб-сервером и сервером баз данных. Является открытым программным обеспечением и распространяется под лицензией MIT.

Достоинства:

- возможность прямых системных вызовов;
- мощная поддержка операций со строками и правилами (регулярными выражениями);
- мгновенное проявление изменений во время разработки;
- отсутствие стадии компиляции;
- неограниченный диапазон значений целых чисел;
- модель обработки исключений;
- все операторы возвращают значения, даже управляющие структуры;
- динамическая загрузка;
- механизм перехвата исключений;
- может внедряться в HTML-разметку;
- легко интегрирует в свои программы высокопроизводительные серверы баз данных (DB2, MySQL, Oracle и Sybase);
- поддержка потоков.

Недостатки:

- очень мало документации;

- неуправляемость некоторых процессов (таких, как выделение памяти), невозможность задания низкоуровневых структур данных или подпрограмм;
- невозможность компиляции и сопутствующей ей оптимизации программы;
- открытость исходного кода даже в готовой программе (есть средство упаковки исходного кода в .exe-файл под Windows);
- следствие двух первых недостатков – весьма низкая скорость запуска и выполнения программ.

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

Для PHP существует 3 области применения [4]:

- создание скриптов для выполнения на сервере, и это является его традиционным использованием. Для того чтобы использовать (просматривать результаты выполнения PHP-скриптов в браузере), нужен интерпретатор PHP, веб-сервер и браузер;
- создание скриптов для выполнения в командной строке. Есть возможность создать PHP-скрипт, способный запускаться без сервера или браузера. Всё, что нужно, – это парсер-PHP;
- создание оконных приложений, выполняющихся на стороне клиента. В данном виде его использую крайне редко.

PHP – это язык сценариев, который используется на стороне сервера и разработан специально для Web. Он позволяет непосредственно в HTML-страницу внедрять код, выполняемый при каждом ее посещении. При этом

программа на PHP интерпретируется Web-сервером, а результат ее работы в виде HTML-страницы отображается браузером пользователя.

PHP отличается наличием ядра и подключаемых модулей, «расширений»: для работы с базами данных, сокетами, динамической графикой, криптографическими банками, документами формата PDF и т. п.

Интерпретатор PHP подключается к веб-серверу либо через модуль, созданный специально для этого сервера (например, для Apache или IIS), либо в качестве CGI-приложения [5].

PHP в связке с Web-сервером Apache позволяет разработчику создать полноценное приложение автоматизированной информационной системы.

Процессор гипертекста PHP – это серверный язык, с помощью которого можно создавать скрипты, созданный специально для Web.

Решения на базе PHP предлагают организации:

- универсальность;
- прозрачную интеграцию;
- гибкость;
- ценовую эффективность;
- безопасность;
- высокую производительность.

Достоинства PHP:

- является свободным программным обеспечением, распространяемым под особой лицензией (PHP license);
- прост в освоении на всех этапах;
- поддерживается большим сообществом пользователей и разработчиков;
- имеет развитую поддержку баз данных;
- имеется огромное количество библиотек и расширений языка;
- может использоваться в изолированной среде;

Laravel считается мощным фреймворком, имеющий простую структуру кода, благодаря которой любой пользователь при создании web-проекта может оценить возможности модульной компоновки кода. Также данный фреймворк отличается гибкостью, которая позволяет вносить изменения в приложение, чему особенно способствует краткость и легко читаемость кода.

Отличительные особенности:

- поддерживает Composer для управления пакетами;
- хорошо делает модульное тестирование;
- предлагает тонны пакетов для расширения функциональности фреймворков;
- имеет большое сообщество с большим количеством учебных ресурсов.

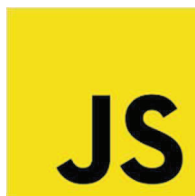
Учитывая все вышеперечисленное, был выбран фреймворк Yii2, благодаря его надежности, безопасности, а также быстродействию.

На рисунке 2.2 приведен итоговый перечень средств, с помощью которых реализована система:

PHP 5.6



JavaScript



Yii2



HTML



Изм.	Лист	№ докум.	Подпись	Дата

09.03.01.2017.886.00 ПЗ

Лист

24



Рисунок 2.2 Перечень используемых средств

2.2 Выбор аппаратного обеспечения

Для web-приложения, в котором присутствует база данных, необходим сервер для обработки запросов серверной части. Для этого нужен хостинг.

У хостинга выделяются основные характеристики, такие как:

- поддерживаемые языки программирования;
- поддержка СУБД;
- пропускная способность;
- объём самого хостинга.

Данное приложение будет разработано на языке программирования PHP. Этот язык поддерживают 99% всех хостингов, так как PHP считается одним из лучших языком написания для web-приложений.

База данных, построенная с помощью MySQL, также является очень распространённым вариантом для хостингов.

Пропускная способность рассчитывается по формуле:

*средний_вес_страницы*ожидаемое_количество_просмотров_за_месяц*

Пусть средний вес равен 20КБ. Так как город, для которого делается приложение, является маленьким, то ожидаемое количество можно ограничить 10000 посещений. В итоге получается пропускная способность равная $0.02 \text{ МБ} * 10000 = 200 \text{ МБ}$

Общий объём хостинга рассчитывается исходя из того, что город небольшой и поэтому база данных не будет большой. Пусть объём базы 1GB.

Для сравнения будут рассмотрены два популярнейших: Hostinger (рисунок 2.3) и HostiMan (рисунок 2.4).

3 ПРОГРАММНАЯ РАЗРАБОТКА ДЛЯ WEB-ПРИЛОЖЕНИЯ

3.1 Архитектура и реализация

В качестве архитектуры системы выбрана клиент-серверная. Она бывает двухуровневой и многоуровневой (в частности, трёхуровневой).

Для двухуровневой системы взаимодействие между клиентом и сервером происходит с помощью запросов клиента серверу и принятия ответов от сервера [12].

Двухуровневая архитектура представлена на рисунке 3.1.

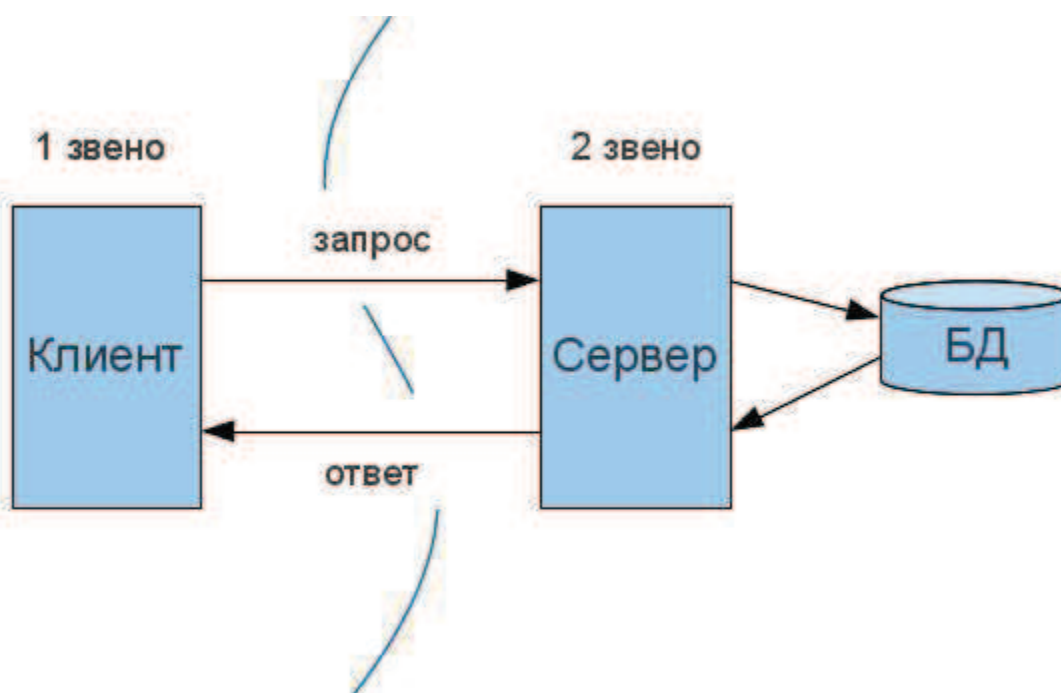


Рисунок 3.1 - Двухуровневая клиент-серверная архитектура

Двухуровневая архитектура состоит из клиента, сервера и базы данных. У данной архитектуры есть свои недостатки и достоинства. К достоинствам относят:

- многие вычисления выполняются на сервере, поэтому требования к устройствам, на которых установлен клиент, снижаются;
- все данные хранятся на сервере, который, как правило, защищён гораздо лучше клиентов;
- возможно централизованное администрирование функций.

Есть и свои недостатки у данной архитектуры:

При неработоспособности сервера будет неработоспособна вся система.

Трёхуровневая архитектура используется, когда есть потребность в больших вычислениях. Они реализуются на основе модели сервера. Сервер делится на две части, каждая из частей выполняется на отдельном компьютере. Третье звено в такой архитектуре называется сервером приложений. Компоненты трёхуровневой архитектуры представлены на рисунке 3.2.

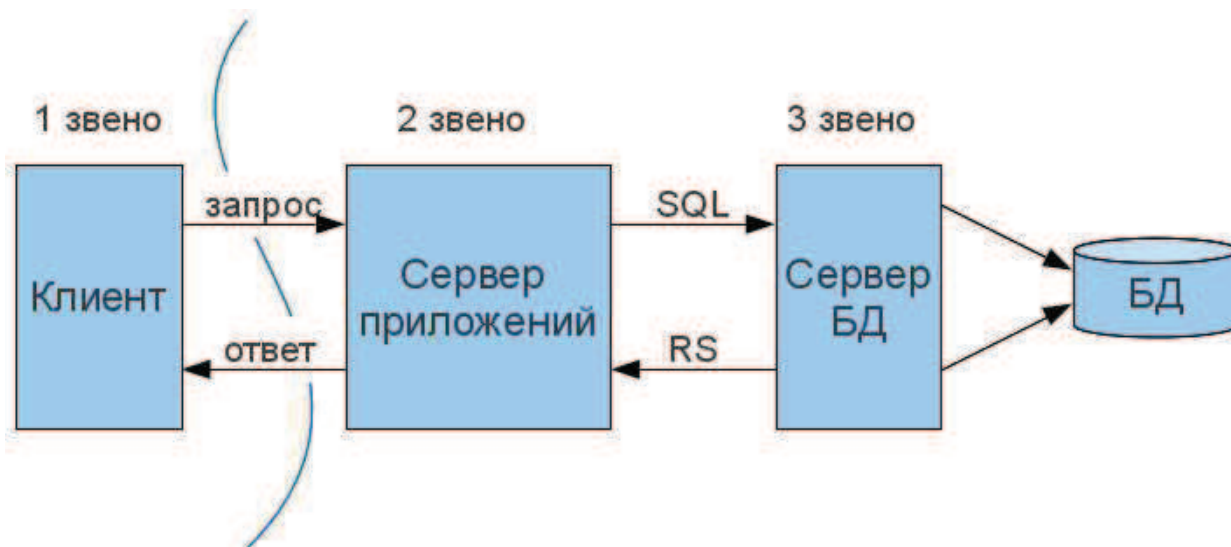


Рисунок 3.2 - Трёхуровневая клиент-серверная архитектура

Представление данных происходит на стороне клиента. Прикладной компонент находится на выделенном сервере приложений (как вариант, выполняющем функции промежуточного ПО). А управление ресурсами – на сервере БД, который и предоставляет запрашиваемые данные.

Трёхуровневая клиент-серверная архитектура, имеет следующие достоинства:

- высокую степень гибкости и масштабируемости;
- высокую безопасность (т.к. защиту можно определить для каждого сервиса или уровня);
- высокую производительность (т.к. задачи распределены между серверами).

Недостатком такой архитектуры считается сложность реализации и увеличение ресурсоёмкости.

В качестве архитектуры системы, была выбрана двухуровневая (или двухзвенная) клиент-серверная архитектура, потому что в данном случае нет необходимости в больших вычислениях на стороне сервера [13].

3.2 Организация работы web-приложения.

Необходимый функционал показан на UML-диаграммах Use Case. На рисунке 3.3 диаграмма для зарегистрированного и незарегистрированного пользователя, на рисунке 3.4 для менеджера и руководителя, на рисунке 3.5 для водителя.

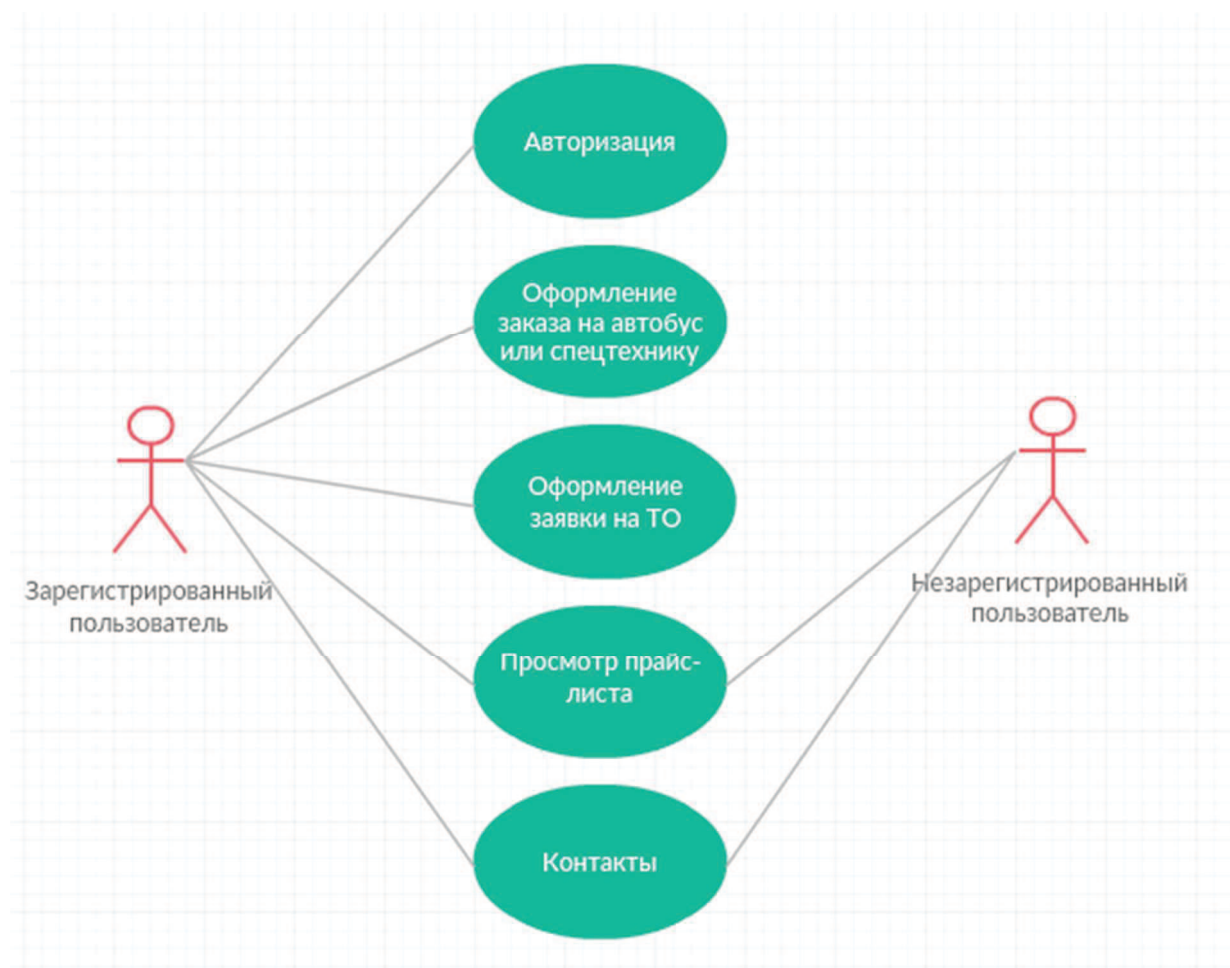


Рисунок 3.3 - Use Case для зарегистрированного и незарегистрированного пользователя

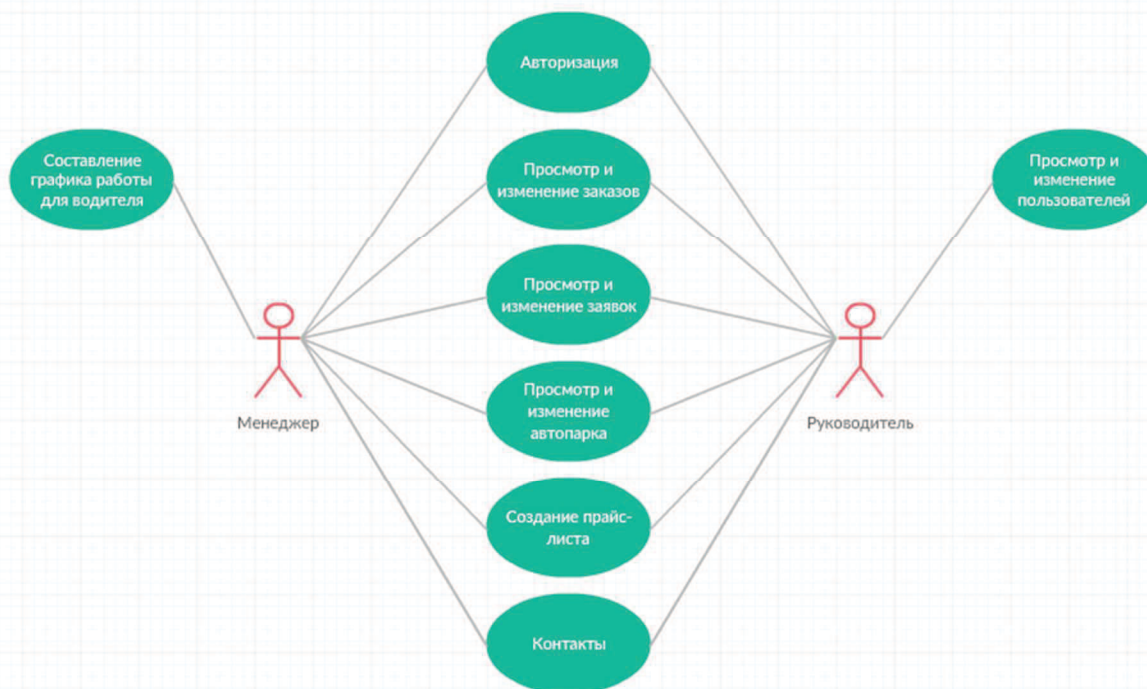


Рисунок 3.4 - Use Case для менеджера и руководителя



Рисунок 3.5 - Use Case для водителя

Краткие выводы по разделу три

Была выбрана архитектура приложения и составлены диаграммы использования для пользователей системы.

Изм.	Лист	№ докум.	Подпись	Дата

09.03.01.2017.886.00 ПЗ

Лист

30

4 ОРГАНИЗАЦИЯ БАЗЫ ДАННЫХ

Результатом построения концептуальной модели является диаграмма сущностей, представленная на рисунке 4.1.

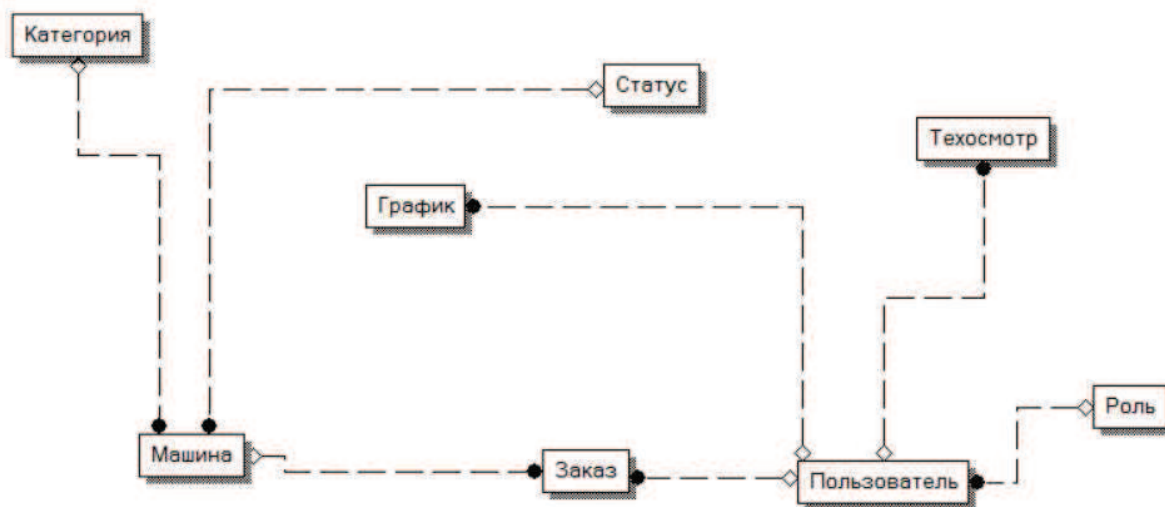


Рисунок 4.1 - Диаграмма сущностей

Описание логической модели данных лучше всего представить в виде диаграммы «сущность-связь» (ER - диаграммы).

На рисунок 4.2 представлена ER-диаграмма данных.

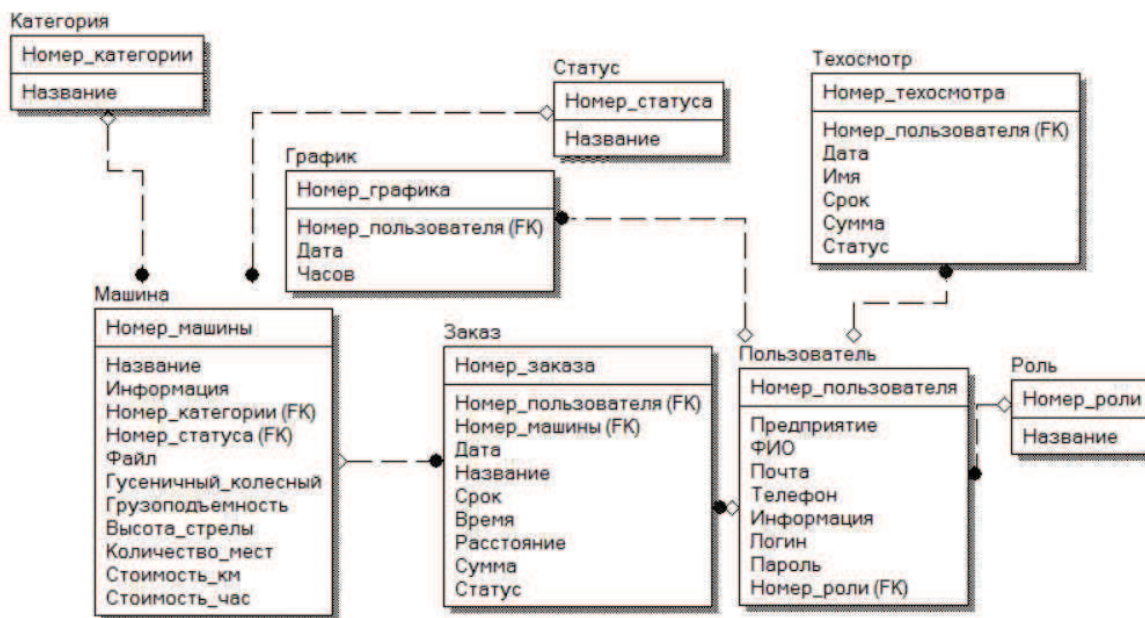


Рисунок 4.2 - ER-диаграмма

Связи между сущностями не идентифицирующие, отношение 1 к n. Все внешние ключи не входят в составной первичный ключ сущности.

Для обеспечения нормальной работы БД, устранения избыточности и других аномалий необходимо произвести нормализацию БД и привести ее, как минимум, к третьей нормальной форме (Н.Ф.) [14].

Диаграмма физической модели данных генерируется автоматически программой ERwin Data Modeler (рисунок 4.3)

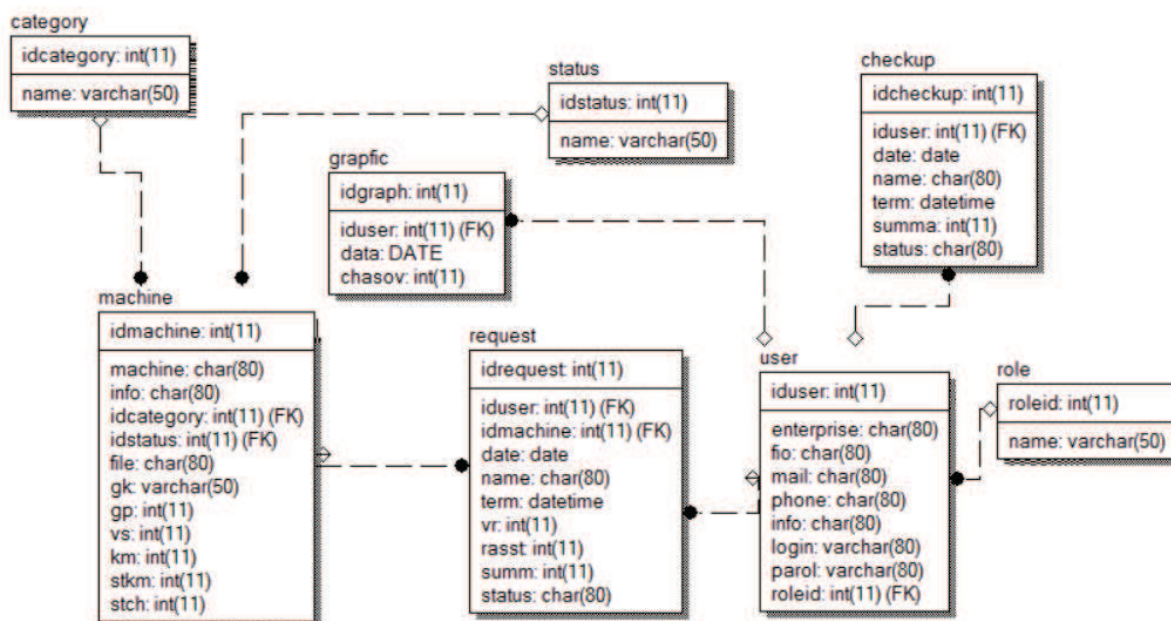


Рисунок 4.3 - Физическая модель данных

Для организации хранилища данных веб-приложения используется СУБД MySQL.

Таблица 4.1 – Назначение таблиц БД

Название	Описание таблицы
User	Таблица с указанием регистрационных данных пользователей
Request	Таблица с указанием данных по заявкам на машины
Checkup	Таблица с указанием данных по заявкам на ТО
Machine	Таблица с указанием данных по машинам
Status	Таблица с указанием возможных статусов машин
Category	Таблица с указанием возможных типов машин
Graphic	Таблица с графиком работы водителей
Role	Таблица с указанием ролей пользователей

Ниже рассмотрена каждая таблица, приведенная в таблице 4.1, более детально.

Таблица 4.2 – Состав таблицы «User»

Название поля	Тип данных	Ключ	Описание
Iduser	Integer	Первичный	Идентификатор пользователя
Enterprise	Char(80)		Предприятие пользователя
Fio	Char(80)		ФИО пользователя
Mail	Char(80)		Электронная почта
Phone	Char(80)		Телефон пользователя
Info	Char(80)		Информация о пользователе
Login	Varchar(80)		Логин пользователя
Parol	Varchar(80)		Пароль пользователя
Roleid	Integer	Внешний	Роль пользователя

```
CREATE TABLE `user` (
  `iduser` int(11) NOT NULL AUTO_INCREMENT,
  `enterprise` char(80) COLLATE cp1251_bin DEFAULT NULL,
  `fio` char(80) COLLATE cp1251_bin DEFAULT NULL,
  `mail` char(80) COLLATE cp1251_bin DEFAULT NULL,
  `phone` char(80) COLLATE cp1251_bin DEFAULT NULL,
  `info` char(80) COLLATE cp1251_bin DEFAULT NULL,
  `login` varchar(80) COLLATE cp1251_bin NOT NULL,
  `parol` varchar(80) COLLATE cp1251_bin NOT NULL,
  `role` int(11) NOT NULL DEFAULT '1',
  PRIMARY KEY (`iduser`),
  KEY `role` (`role`)
```


) ENGINE=InnoDB DEFAULT CHARSET=cp1251 COLLATE=cp1251_bin

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
<input type="checkbox"/> 1	iduser	int(11)			Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/> 2	enterprise	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/> 3	fio	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/> 4	mail	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/> 5	phone	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/> 6	info	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/> 7	login	varchar(80)	cp1251_bin		Нет	Нет	
<input type="checkbox"/> 8	parol	varchar(80)	cp1251_bin		Нет	Нет	
<input type="checkbox"/> 9	role	int(11)			Нет	1	

Рисунок 4.4 - Проект таблицы user

Таблица 4.3 – Состав таблицы «Machine»

Название поля	Тип данных	Ключ	Описание
Idmachine	Integer	Первичный	Идентификатор машины
Machine	Char(80)		Название машины
Info	Char(80)		Информация о машине
IdCategory	Integer	Внешний	Вид машины
IdStatus	Integer	Внешний	Идентификатор статуса
File	Char(80)		Файл изображения
Gk	Varchar(80)		Гусеничный/колесный
Gp	Integer		Грузоподъемность
Vs	Integer		Высота стрелы
Km	Integer		Количество мест
Stkm	Integer		Стоимость километра
StCh	Integer		Стоимость часа

```
CREATE TABLE `machine` (
  `idmachine` int(11) NOT NULL AUTO_INCREMENT,
  `machine` char(80) COLLATE cp1251_bin DEFAULT NULL,
```

```

`info` char(80) COLLATE cp1251_bin DEFAULT NULL,
`category` int(80) DEFAULT '1',
`status` int(11) NOT NULL DEFAULT '2',
`file` char(80) COLLATE cp1251_bin DEFAULT NULL,
`gk` varchar(50) COLLATE cp1251_bin NOT NULL,
`gp` int(11) NOT NULL,
`vs` int(11) NOT NULL,
`km` int(11) NOT NULL,
`stkm` int(11) NOT NULL,
`stch` int(11) NOT NULL,
PRIMARY KEY (`idmachine`),
KEY `category` (`category`),
KEY `status` (`status`)

```

) ENGINE=InnoDB DEFAULT CHARSET=cp1251 COLLATE=cp1251_bin

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
<input type="checkbox"/>	1 idmachine	int(11)			Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/>	2 machine	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/>	3 info	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/>	4 category	int(80)			Да	1	
<input type="checkbox"/>	5 status	int(11)			Нет	2	
<input type="checkbox"/>	6 file	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/>	7 gk	varchar(50)	cp1251_bin		Нет	Нет	
<input type="checkbox"/>	8 gp	int(11)			Нет	Нет	
<input type="checkbox"/>	9 vs	int(11)			Нет	Нет	
<input type="checkbox"/>	10 km	int(11)			Нет	Нет	
<input type="checkbox"/>	11 stkm	int(11)			Нет	Нет	
<input type="checkbox"/>	12 stch	int(11)			Нет	Нет	

Рисунок 4.5 - Проект таблицы machine

PRIMARY KEY (`idcheckup`),

KEY `idclient` (`iduser`)

) ENGINE=InnoDB DEFAULT CHARSET=cp1251 COLLATE=cp1251_bin

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
<input type="checkbox"/>	1 idcheckup	int(11)			Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/>	2 iduser	int(11)			Нет	Нет	
<input type="checkbox"/>	3 date	date			Да	NULL	
<input type="checkbox"/>	4 name	char(80)	cp1251_bin		Да	NULL	
<input type="checkbox"/>	5 term	datetime			Да	NULL	
<input type="checkbox"/>	6 summa	int(11)			Нет	800	
<input type="checkbox"/>	7 status	char(80)	cp1251_bin		Да	NULL	

Рисунок 4.7 - Проект таблицы checkup

Таблица 4.6 – Состав таблицы «Graphic»

Название поля	Тип данных	Ключ	Описание
Igraph	Integer	Первичный	Идентификатор графика
Iduser	Integer		Пользователь
Date	Date		Дата графика
Chasov	Integer		Количество часов

```
CREATE TABLE `graphic` (  
  `idgraph` int(11) NOT NULL AUTO_INCREMENT,  
  `user` int(11) NOT NULL,  
  `data` date NOT NULL,  
  `chasov` int(11) NOT NULL,  
  PRIMARY KEY (`idgraph`),  
  KEY `user` (`user`)  
) ENGINE=InnoDB DEFAULT CHARSET= cp1251_bin
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
<input type="checkbox"/>	1 <u>idgraph</u>	int(11)			Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/>	2 <u>user</u>	int(11)			Нет	Нет	
<input type="checkbox"/>	3 <u>data</u>	date			Нет	Нет	
<input type="checkbox"/>	4 <u>chasov</u>	int(11)			Нет	Нет	

Рисунок 4.8 - Проект таблицы graphic

Таблица 4.7 – Состав таблицы «Category»

Название поля	Тип данных	Ключ	Описание
Idcategory	Integer	Первичный	Идентификатор категории
Name	Varchar(50)		Название категории

```
CREATE TABLE `category` (
  `idcategory` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) CHARACTER SET cp1251 COLLATE cp1251_bin NOT
  NULL,
  PRIMARY KEY (`idcategory`)
) ENGINE=InnoDB DEFAULT CHARSET= cp1251_bin
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
<input type="checkbox"/>	1 <u>idcategory</u>	int(11)			Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/>	2 <u>name</u>	varchar(50)	cp1251_bin		Нет	Нет	

Рисунок 4.9 - Проект таблицы category

Таблица 4.8 – Состав таблицы «Status»

Название поля	Тип данных	Ключ	Описание
Idstatus	Integer	Первичный	Идентификатор статуса
Name	Varchar(50)		Название статуса

```
CREATE TABLE `status` (
  `idstatus` int(11) NOT NULL AUTO_INCREMENT,
```

```

`name` varchar(50) NOT NULL,

PRIMARY KEY (`idstatus`)

) ENGINE=InnoDB DEFAULT CHARSET= cp1251_bin

```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
<input type="checkbox"/>	1 <u>idstatus</u>	int(11)			Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(50)	utf8_general_ci		Нет	Нет	

Рисунок 4.10 - Проект таблицы status

Таблица 4.9 – Состав таблицы «Role»

Название поля	Тип данных	Ключ	Описание
Idrole	Integer	Первичный	Идентификатор роли
Name	Varchar(50)		Название роли

```

CREATE TABLE `role` (
  `roleid` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) CHARACTER SET cp1251 COLLATE cp1251_bin NOT
  NULL,
  PRIMARY KEY (`roleid`)
) ENGINE=InnoDB DEFAULT CHARSET= cp1251_bin

```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
<input type="checkbox"/>	1 <u>roleid</u>	int(11)			Нет	Нет	AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(50)	cp1251_bin		Нет	Нет	

Рисунок 4.11 - Проект таблицы role

Краткие выводы по разделу четыре

Была составлена ER-диаграмма. Описаны имеющиеся таблицы и приведены SQL-скрипты создания таблиц.


```

<html >

<head>

    Описательная часть HTML документа

<meta name="keywords" content="" />

<meta name="description" content="" />

<meta http-equiv="content-type" content="text/html; charset=windows-1251" />

<title>Автоматизация автотранспортного предприятия</title>

<link href="style.css" rel="stylesheet" type="text/css" media="screen" />

</head>

<body>

<div id="wrapper">

    <div id="header-wrapper1">

        <div id="logo">

            <a href="project.php">  </a>

        </div>

        <div id="header1" class="container">

            Подключается файл заголовка общий для всех модулей приложения

            <?
require "menu.php";
?>

        </div>

    </div>

    <!-- end #header -->

```

```
<div style="clear: both;">
```

```
<div class="post">
```

```
<br>
```

```
<?
```

Проверяется, происходит обновление данных, либо добавляются новые данные

```
$upd=$_REQUEST["upd"];
```

Если данные добавляются

```
if ($upd==1)
```

```
{
```

```
$Arr=$_REQUEST["Arrrequest"];
```

```
$idcheckup=$Arr[0];
```

Формируется запрос к базе данных, в который выводится текущая заявка

```
$r=mysql_query("select * from checkup where idcheckup=$idcheckup");
```

```
$f=mysql_fetch_array($r);
```

```
?> <div style="font-size:22px; color:#000" align="center">Редактирование  
заявки "<? echo $f['name']?>"</div>
```

```
<?
```

```
}
```

```
else
```

```
{
```

```
?>
```

Иначе в заголовок формы прописывается «Добавление»

					<i>09.03.01.2017.886.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		44

```
<div style="font-size:22px; color:#000" align="center">Добавление
заявки</div>
```

```
<?
```

```
}
```

```
?>
```

```
<br>
```

```
<?
```

Указывается текущая дата.

```
$date=date("Y")."-".date("m")."-".date("d");
```

```
?>
```

```
<form name="form2" method="post" enctype="multipart/form-data" >
```

```
<table width="526" border="0">
```

```
<tr>
```

```
<td width="172"><font color="#000000" > Название: </font> </td>
```

```
<td width="240"><input name="name" type="text" value="<? if
($upd==1) echo $f['name']; else echo(""); ?>" size="40" ></td>
```

```
</tr>
```

```
<tr>
```

```
<td><font color="#000000" > Дата: </font> </td>
```

```
<td><input size="8" name="date" value="<? if ($upd==1) echo
$f['date']; else echo("$date"); ?>" type="text" ></td>
```

```
</tr>
```

```
<?
```

```
$date=date("Y")."-".date("m")."-".date("d")." ".date("h").":".date("i");
```

?>

<tr>

<td> Сроки: </td>

<td><input size="16" name="term" value="<? if (\$upd==1) echo \$f['term']; else echo("\$date"); ?>" type="text" ></td>

</tr>

<tr>

<td> Статус: </td>

<td>

Статус заявки, если данные обновляются – автоматически устанавливается на нужный пункт

<select name="status" style="height:22; width:auto" >

<option value="На рассмотрении" <? if ((\$upd==1)&& (\$f['status']=="На рассмотрении")) echo "selected"; ?> > На рассмотрении </option>

<option value="Выполняется" <? if ((\$upd==1)&& (\$f['status']=="Выполняется")) echo "selected"; ?> >Выполняется </option>

<option value="Готово" <? if ((\$upd==1)&& (\$f['status']=="Готово")) echo "selected"; ?> >Готово </option>

<option value="Заказ отменен" <? if ((\$upd==1)&& (\$f['status']=="Заказ отменен")) echo "selected"; ?> >Заказ отменен </option>

</select>

</td>

</tr>

</table>

Кнопки формы

```
<input type="button" name="button"
onclick="this.form.action='updcheckmng.php?step=2&upd=<?
echo"$upd";?>&id=<? echo"$Arr[0]";?>'; this.form.submit();" value="Сохранить"
width="500">
```

```
<input type="button" name="button"
onClick="javascript:history.back();" value="Отмена">
```

Конец формы

```
</form>
```

```
</td>
```

```
</tr>
```

```
</table></td>
```

```
</div>
```

```
</div>
```

```
<!-- end #page -->
```

```
</div>
```

```
<!-- end #footer -->
```

```
</body>
```

```
</html>
```

```
<?
```

Если второй шаг, то это есть внесение данных в базу данных

```
$step=$_REQUEST["step"];
```

```
if ($step==2)
```

```
{
```

Обновляются ли данные

```
$upd=$_REQUEST["upd"];
```

Получение всех данных из переменных \$_POST

```
$name = $_POST["name"];
```

```
$term = $_POST["term"];
```

```
$date = $_POST["date"];
```

```
$status = $_POST["status"];
```

Если были введены неверные данные, то выводится сообщение об этом и данные не сохраняются

```
if ((strlen($name)>50) or (strlen($name)==0))
```

```
{
```

```
?>
```

```
<script language="javascript">
```

```
alert("Не верный ввод!");
```

```
history.back();
```

```
</script>
```

```
<?
```

```
exit();
```

```
}
```

Если данные обновляются

```
if ($upd==1)
```

```
{
```

					09.03.01.2017.886.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		48

```
$id=$_REQUEST["id"];
```

Формируется запрос на обновление

```
mysql_query("Update checkup set name='$name',status='$status', date='$date',  
term='$term' WHERE idcheckup=$id");
```

```
} else
```

```
{
```

Иначе формирование SQL-запроса на добавление данных

```
mysql_query("INSERT INTO checkup (name, status, idclient, date, term) VALUES  
('$name', '$status', $idclient, '$date', '$term')");
```

```
}
```

```
?>
```

```
<script language="javascript">
```

```
location.href='checkmng.php?filter=0';
```

```
</script>
```

```
<?
```

```
}
```

5.2 Клиентская часть

В начале работы с системой необходимо произвести вход в систему. В зависимости от того, под пользовательским или менеджерским аккаунтом вошел пользователь в систему, динамически меняется состав главного меню.

Форма авторизации показана на рисунке 5.1.

					<i>09.03.01.2017.886.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		49

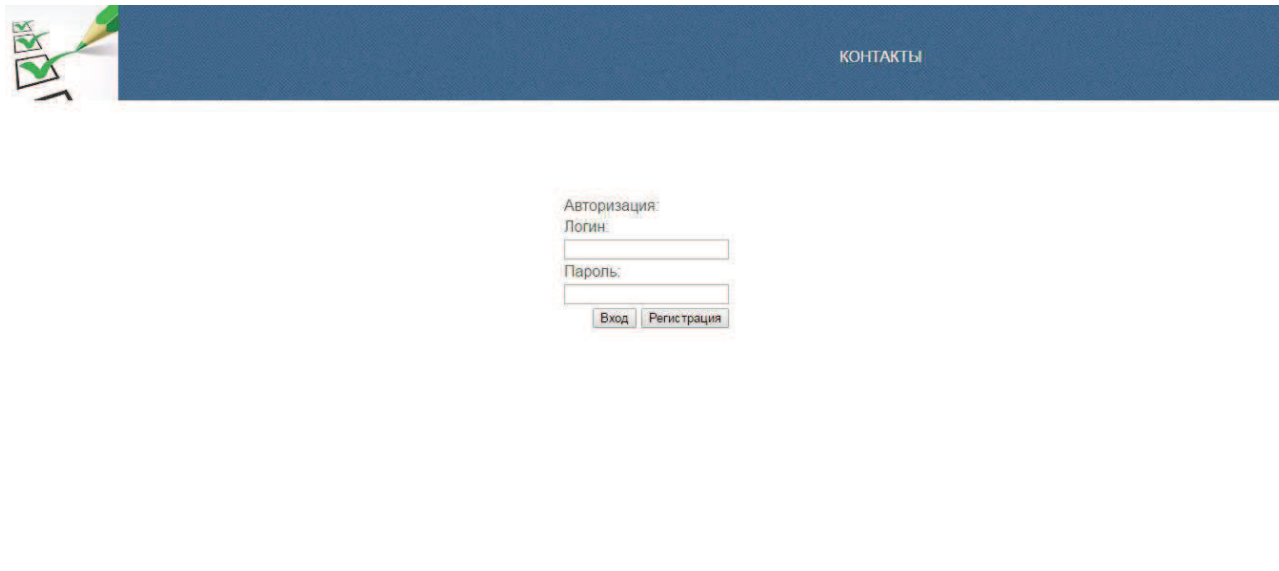


Рисунок 5.1 - Форма авторизации в системе

Для руководителя имеется возможность создавать новых пользователей системы. Форма добавление нового пользователя показана на рисунке 5.2.

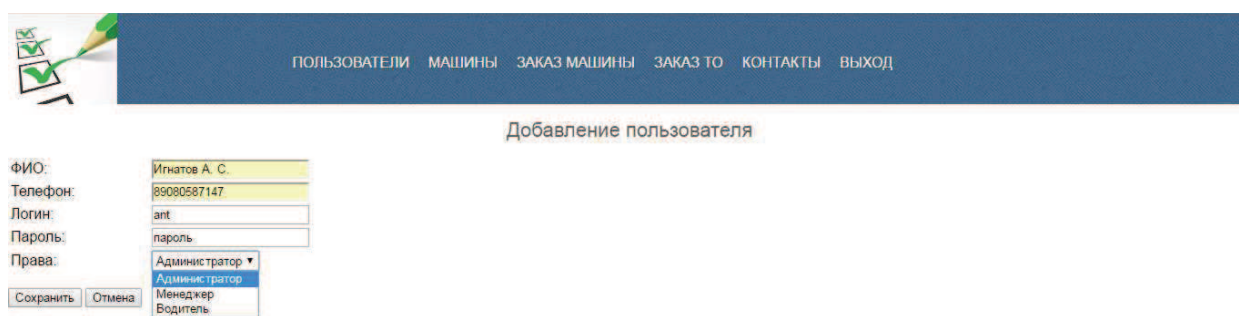


Рисунок 5.2 - Добавление пользователя

После входа в систему менеджер получает возможность редактировать имеющиеся заказы на машины и на ТО, то есть изменять их статус.

Форма списка заказов на машины показана на рисунке 5.3.

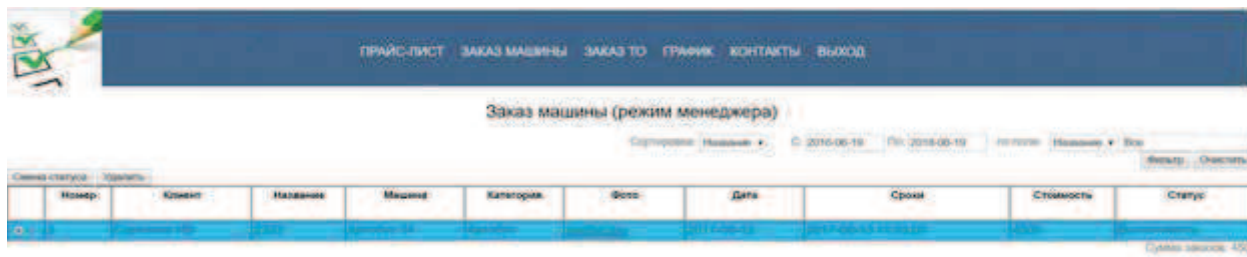


Рисунок 5.3 - Список заказов на машины

Для редактирования заказа необходимо выбрать его в списке и нажать «Смена статуса». Для удаления необходимо нажать на кнопку «Удалить».

Форма смены статуса выглядит, как показано на рисунке 5.4.

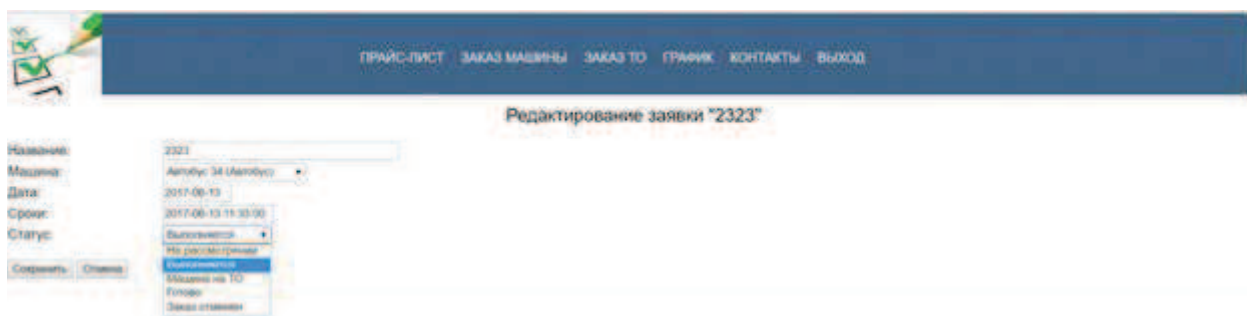
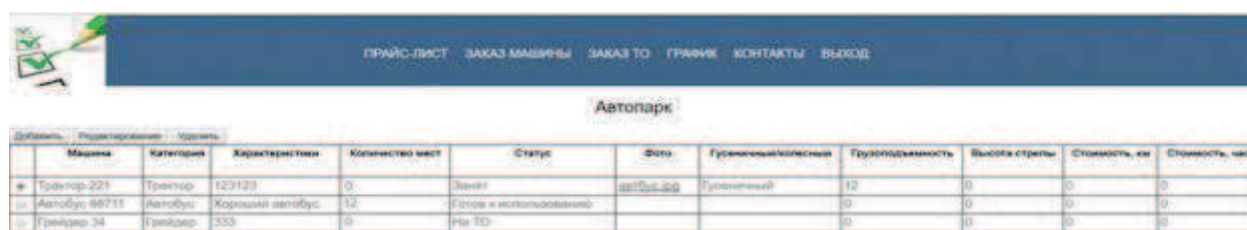


Рисунок 5.4 - Редактирование заявки

Для смены статуса необходимо выбрать статус и нажать на кнопку «Сохранить».

Менеджер не имеет возможности добавлять новые заявки, в отличие от клиента.

Также менеджеру доступно редактирование прайс-листа машин. Для этого необходимо нажать на кнопку «Прайс-лист» в верхней части страницы (рисунок 5.5).



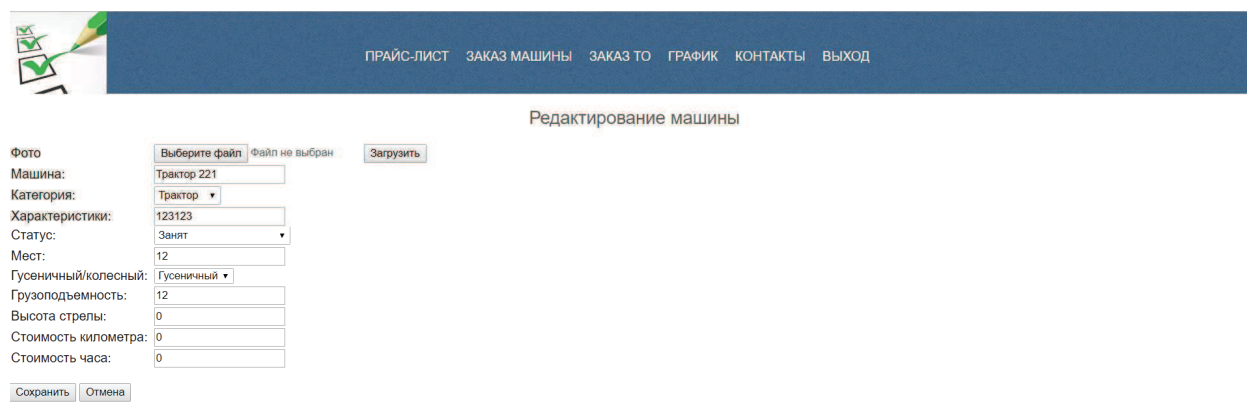
ПРАЙС-ЛИСТ ЗАКАЗ МАШИНЫ ЗАКАЗ ТО ГРАФИК КОНТАКТЫ ВЫХОД

Автопарк

Машина	Категория	Характеристики	Количество мест	Статус	Фото	Гусеничный/колесный	Грузоподъемность	Высота стрелы	Стоимость, км	Стоимость, час
Трактор 221	Трактор	123123	0	Занят	авто.ру	Гусеничный	12	0	0	0
Автобус 88711	Автобус	Хороший автобус	14	Готов к использованию			0	0	0	0
Грейдер 34	Грейдер	333	0	На ТО			0	0	0	0

Рисунок 5.5 - Список машин

В списке приводится краткая информация по каждой из машин. Для редактирования машины необходимо выбрать ее в списке, после чего нажать на кнопку «Редактировать». На экране отобразится форма редактирования автомобиля (рисунок 5.6).



ПРАЙС-ЛИСТ ЗАКАЗ МАШИНЫ ЗАКАЗ ТО ГРАФИК КОНТАКТЫ ВЫХОД

Редактирование машины

Фото:

Машина:

Категория:

Характеристики:

Статус:

Мест:

Гусеничный/колесный:

Грузоподъемность:

Высота стрелы:

Стоимость километра:

Стоимость часа:

Рисунок 5.6 - Редактирование машины

Также в кабинете менеджера доступно редактирование статусов заказов на ТО.

Для перехода к списку заявок на ТО – необходимо выбрать пункт «Заказ ТО» в верхней части страницы.

На экране отобразится список заказов на ТО (рисунок 5.7).

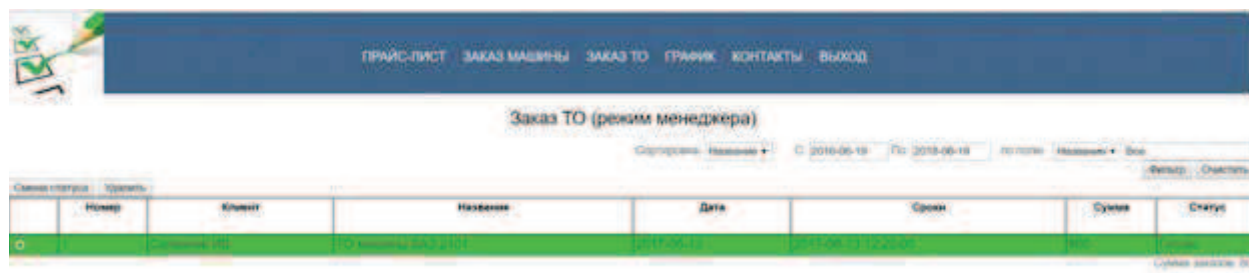


Рисунок 5.7 - Список заказов на ТО

Для добавления или редактирования заказа необходимо выбрать его в списке и нажать на соответствующую кнопку.

Форма редактирования заказа показана на рисунке 5.8.

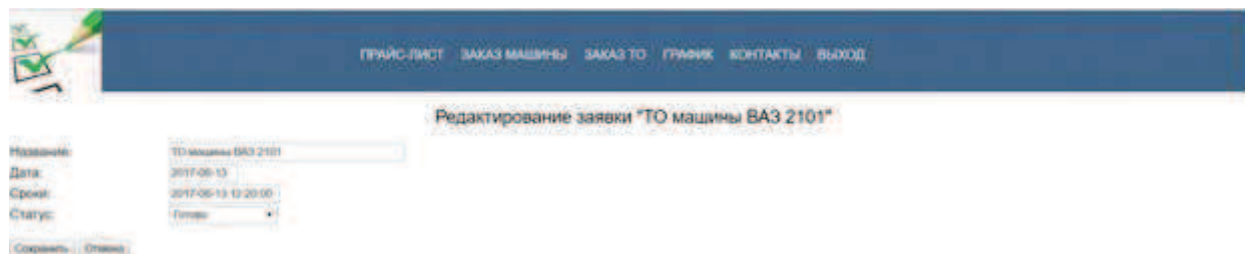


Рисунок 5.8 - Форма редактирования заказа

Также менеджеру доступно редактирование графиков работы водителей. Для этого необходимо зайти в соответствующий раздел. На экран будет выведен график за текущий месяц (рисунок 5.9).

Изм.	Лист	№ докум.	Подпись	Дата

09.03.01.2017.886.00 ПЗ

Лист

53

Рисунок 5.9 - График работы водителей

Ввод данных производится непосредственно в график. Данные сохраняются по нажатию на кнопку «Сохранить».

Для регистрации пользователя можно нажать на кнопку «Регистрация» на главной форме приложения (рисунок 5.10).

Рисунок 5.10 - Форма регистрации

После ввода данных в форму и их сохранения можно войти в систему аналогично администратора.

На экране отображается список заказов на машины (рисунок 5.11).

Номер	Название	Машина	Категория	Фото	Дата	Сроки	Расстояние	Сумма	Статус
3	2323	Автобус 34	Автобус	автбус.jpg	2017-06-13	2017-06-13 11:33:00	100	4500	Выполняется

Рисунок 5.11 - Список заказов на машины

Для редактирования заказа необходимо выбрать его в списке и нажать на кнопку «Редактировать».

На экране отобразится форма редактирования заказа (рисунок 5.12).

Редактирование заявки "2323"

Название: 2323
 Машина: Автобус 34 (Автобус)
 Дата: 2017-06-13
 Сроки: 2017-06-13 11:33:00
 Расстояние: 100

Сохранить Отмена

Рисунок 5.12 - Форма редактирования заявки

Следует отметить, что система автоматически вычисляет сумму заявки на основании введенного расстояния и количества мест в автобусе, после чего вносит эти данные в базу данных.

Также пользователю системы доступен просмотр всего транспорта. Для этого необходимо нажать на кнопку «Прайс-лист» в верхнем меню (рисунок 5.13).

Машина	Категория	Характеристики	Количество мест	Статус	Фото	Гусеничный/колесный	Грузоподъемность	Высота стрелы	Стоимость, км	Стоимость, час
Трактор-221		123123	0			Гусеничный	12	0	0	0
Грейдер 34		333	0				0	0	0	0
Автобус 88711		Хороший автобус	12				0	0	0	0

Рисунок 5.13 - Список транспорта

Доступно добавление и редактирование заявок на ТО. Для этого необходимо перейти на вкладку «Заказ ТО». На экране отобразится список заказов на ТО (рисунок 5.14).

Заказ ТО клиента, Селезнев ИВ

Сортировка: Название ▼ С: 2016-06-19 По: 2018-06-19 по полю Название ▼ Все

Добавить Редактирование Удалить

Номер	Название	Дата	Сроки	Стоимость	Статус
1	ТО машины БАЗ 2101	2017-06-13	2017-06-13 12:20:00	800	Готово

Рисунок 5.14 - Форма заказов на ТО

Следует отметить, что в данной форме доступна фильтрация и сортировка по различным типам. Для этого необходимо выбрать необходимые параметры в верхней правой части, после чего нажать «Фильтр».

Для редактирования заказа необходимо выбрать его в списке, после чего нажать «Редактировать». На экран будет выведена форма редактирования (рисунок 5.15).

ЗАКАЗ МАШИНЫ ЗАКАЗ ТО ТРАНСПОРТ КОНТАКТЫ ВЫХОД

Редактирование заявки "ТО машины ВАЗ 2101"

Название:

Дата:

Сроки:

Рисунок 5.15 - Форма редактирования заявки

Для водителя имеется свой интерфейс, в котором он может просмотреть свой график работы (рисунок 5.16).

ПРАЙС-ЛИСТ КОНТАКТЫ ВЫХОД

График работы водителей

Сортировка: С: По: по полю:

Добавить	Редактирование	Удалить	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
			1	2	3	4	5																									
			Иванов И. И.	1	2	3	4	5																								

Рисунок 5.16 - График работы

ЗАКЛЮЧЕНИЕ

В рамках дипломной работы было выполнено следующее:

1. изучены бизнес-процессы предметной области, сформулирован набор требований к приложению;
2. рассмотрены родственные разработки, обоснована необходимость создания приложения;
3. определена среда реализации приложения;
4. разработана архитектура системы;
5. спроектирована база данных;
6. реализована серверная и клиентская части, произведено тестирование приложения.

Возможными путями развития приложения являются:

1. разработка мобильного приложения;
2. улучшение интерфейса;
3. добавление нового функционала.

					<i>09.03.01.2017.886.00 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		58

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Официальный сайт KiberLog. - <http://www.kiberlog.ru/>
2. Официальный сайт АвтоПеревозки AutoSoft. - <http://www.autosoft.ru/products/transportation/>
3. Официальный сайт ПК АвтоПлан. - <http://avtoplan.ru/>
4. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Б. Хоган. – СПб.: Изд-во Питер, 2014 г. - 320 с.
5. Хопкинс К. PHP. Быстрый старт / К. Хопкинс. – М.: Изд-во Эксмо-Пресс, 2014. - 160 с.
6. Кумскова И. А. Базы данных / И. А. Кумскова. – М.: Изд-во КноРус, 2014. - 488 с.
7. Дунаев В. HTML, скрипты и стили / В. Дунаев. – М.: Изд-во ВHV, 2015. - 824 с.
8. Голдстайн А. HTML5 и CSS3 для всех/ А. Голдстайн, Л. Лазарис, Э. Уэйл. – М.: Изд-во Эксмо-Пресс, 2013.- 368 с.
9. Роббинс Д. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Д. Роббинс. – М.: Изд-во Эксмо, 2014. - 528 с.

					<i>09.03.01.2017.886.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		59

10. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Джонс. – М.: Изд-во Эксмо, 2013. - 480 с.
11. Вигерс К. Разработка требований к программному обеспечению /К. Вигрес, Д. Битти. – М.: Изд-во BHV, 2014. - 736 с.
12. Кузин А. В. Базы данных /А. В. Кузин, С. В. Левонисова. – М.: Изд-во Академия, 2014. - 320 с.
13. Астахова И. Ф. «СУБД: язык SQL в примерах и задачах» / И. Ф. Астахова, А. П. Толстобров, В. М. Мельников. – М.: Изд-во «Физматлит», 2013. – 171 с.
14. Астахова И. Ф., SQL в примерах и задачах: учебное пособие / И. Ф. Астахова, А. П. Толстобров, В. М. Мельников. – М.: Новое знание, 2014. – 176 с.
15. Вендров А.М. Проектирование программного обеспечения экономических информационных систем / А. М. Вендров. – М.: Финансы и статистика, 2012. – 352 с.
16. Глушаков С. В., Ломотько Д. В. Базы данных / С. В. Глушаков, Д. В. Ломотько. - Харьков: Фолио 2014. - 415 с.
17. Голицына О. Л., Максимов Н. В., Попов И. И. Базы данных: Учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – М.: ФОРУМ: ИНФРА-М, 2013. - 268 с.