

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись
«____ » _____ 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка автоматизированной системы приема заказов для ООО
«Шинный двор»

Руководитель _____ доцент, канд. физ.-мат. наук А. Н. Таскин
подпись, дата

Выпускник _____ А. В. Жалковский
подпись, дата

Консультанты
по разделам:

Экономический _____ Е. Н. Скуратенко
подпись, дата

Нормоконтролер _____ В. И. Кокова
подпись, дата

Абакан 2022

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись
«_____» _____ 2022 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Абакан 2022

Студенту Жалковскому Александру Вячеславовичу

Группа ХБ 18-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка автоматизированной системы приема заказов для ООО «Шинный двор»

Утверждена приказом по институту № 208 от 14.04.2022 г.

Руководитель ВКР: А.Н Таскин, доцент, канд. физ.-мат. наук, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ООО «Шинный двор»

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Описание разработки веб-приложения.
3. Расчёт затрат и оценка экономической эффективности реализации проекта.

Перечень графического материала: нет

Руководитель ВКР

А. Н. Таскин

подпись

Задание принял к исполнению

А. В Жалковский

подпись

«14» апреля 2022 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка автоматизированной системы приема заказов для ООО «Шинный двор»» содержит 70 страниц текстового документа, 32 иллюстраций, 16 таблиц, 10 использованных источников, 8 формул.

СИСТЕМА, ВЕБ-ПРИЛОЖЕНИЕ, БД, ПРОГРАММА, ОБЪЕКТ, ИНФОРМАЦИЯ, ОТЧЕТ, ИНФОРМАЦИОННАЯ СИСТЕМА, СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД), IDEF0.

Объектом выпускной квалификационной работы является деятельность сотрудников ООО «Шинный двор» по передаче информации.

Цель работы – повышение скорости принятия и передачи заказа клиента, а также составления отчетов по материалам и работам, за счет внедрения веб-приложения на предприятии ООО «Шинный двор».

Задачи:

- провести анализ работы ООО «Шинный двор»;
- провести обзор современных технологий разработки web-приложений;
- выбрать метод проектирования web-приложения;
- разработать структуру web-приложения «Шинный двор»;
- создать web-приложение «Шинный двор»;
- рассчитать показатели экономической эффективности проекта.

В результате моделирования бизнес-процессов организации ООО «Шинный двор» был выявлен недостаток, а именно, принятие и передача заказа происходит посредством бумажного носителя, также отсутствуют отчеты о проделанных работах и использованных материалах.

Созданная система позволит сократить временные затраты ООО «Шинный двор» на оформление заявок, а также создать учетные документы по работам и материалам.

SUMMARY

The theme of the graduation thesis is «Development of Automated System for Company ‘Tire Yard’ to Accept Orders ». It contains 70 pages, 26 figures, 14 tables, 10 reference items, 6 formulae.

SYSTEM, WEB APPLICATION, DATABASE, PROGRAM, OBJECT, INFORMATION, REPORT, IT SYSTEM, DATABASE MANAGEMENT SYSTEM (DBMS), IDEF0.

The object of the thesis is the framework of the employees of the “Tire Yard” company on transmission of information.

The purpose of the graduation paper is to increase the speed of acceptance and transmission of customer’s orders, as well as to make reports via the introduction of a web application of the “Tire Yard”.

Objectives:

- to analyze the framework of the “Tire Yard” company;
- to review modern web application development technologies;
- to select a web application design method;
- to develop the structure of the “Tire Yard” web application;
- to create a web application for the “Tire Yard”;
- to calculate the economic efficiency of the project.

After modeling the business processes of the “Tire Yard” company, a drawback has been identified – the acceptance and transfer of orders are fulfilled in hard copy; there are also no reports on the work done and the materials used.

The suggested system will reduce time spent by the “Tire Yard” company on processing applications, as well as will help to create accounting documents.

СОДЕРЖАНИЕ

Введение	8
1 Анализ предметной области. Выбор средств проектных решений.	9
1.1 Анализ предметной области	9
1.2 Основная деятельность предприятия ООО «Шинный двор»	10
1.3 Структурная схема предприятия ООО «Шинный двор», обоснование необходимости проекта для деятельности предприятия	13
1.4 Характеристика ИТ-инфраструктуры предприятия ООО «Шинный двор» в контексте взаимосвязи с выбранной темой проекта	15
1.5 Цель и задачи разработки проекта	16
1.6 Анализ типовых проектных решений подобных проектов.....	17
1.7 Выбор средств проектных решений.....	18
1.7.1 Выбор архитектуры информационной системы	18
1.7.2 Выбор инструментальных средств разработки	20
1.7.3 Выбор системы управления базами данных для разработки ИС.....	22
1.8 Разработка модели проектируемой информационной системы	24
1.9 Выводы по разделу «Анализ предметной области и выбор средств проектных решений»	27
2 Описание разработки веб-приложения приема заказов для ООО «Шинный двор»	28
2.1 Установка необходимого ПО	28
2.2 Описание процесса создания веб-приложения приема заказов для ООО «Шинный двор»	31
2.3 Выводы по разделу «Описание разработки веб-приложения приема заказов для ООО «Шинный двор»»	45
3 Расчёт затрат и оценка экономической эффективности реализации	46
3.1 Анализ состава и стоимости ресурсов, необходимых для реализации проекта	46
3.2 Расчет проектных затрат	49
3.3 Расчет капитальных затрат	52
3.4 Расчет эксплуатационных затрат.....	55
3.5 Расчет совокупной стоимости владения системой.....	58

3.6 Оценка рисков реализации проекта	60
3.7 Анализ рынка продуктов-аналогов. Установление стоимости программного продукта	61
3.8 Экономическая эффективность реализации проекта	63
3.9 Выводы по разделу «Расчёт затрат и оценка экономической эффективности реализации проекта»	65
Заключение	67
Список использованных источников	68

ВВЕДЕНИЕ

Данная выпускная квалификационная работа представляет собой реализацию разработки автоматизированной системы приема заказов для ООО «Шинный двор». В настоящее время почти у любой организации имеется приложение, которое обладает различными функциями, поэтому важно, чтобы данное приложение подходило этой организации, упрощала и ускоряла процесс обслуживания клиентов.

Объектом выпускной квалификационной работы является деятельность сотрудников ООО «Шинный двор» по передаче информации.

Предметом работы является процесс автоматизации выполнения заказа на предприятии.

Цель работы – повышение скорости принятия и передачи заказа клиента, а также составления отчетов по материалам и работам, за счет внедрения веб-приложения на предприятии ООО «Шинный двор».

Задачи выпускной квалификационной работы:

- провести анализ работы ООО «Шинный двор»;
- провести обзор современных технологий разработки web-приложений;
- выбрать метод проектирования web-приложения;
- разработать структуру web-приложения «Шинный двор»;
- создать web-приложение «Шинный двор»;
- рассчитать показатели экономической эффективности проекта.

1 Анализ предметной области и выбор средств проектных решений

1.1 Анализ предметной области

Главной задачей, стоящей перед любой торговой компанией, является увеличение продаж, ускорение товарооборота на предприятии, а также качество и скорость обслуживания клиентов. Данные показатели позволяют фирме развиваться и расширяться, делают бизнес более доходным и рентабельным. На сегодняшний день нелепо выглядит поход клиента с бумажным заказ нарядом по предприятию, все компании стараются автоматизировать процесс выполнения заказа, для увеличения скорости обслуживания и ведения точного учета работ и материалов.

Автоматизация обычно ведется в двух направлениях деятельности, а именно:

— автоматизация основных бизнес-процессов: например, управление продажами или работой с клиентами. В этом случае она проводится для непосредственного увеличения объема продаж, количества выпускаемой продукции и повышения доходности всего бизнеса в целом;

— автоматизация поддерживающих процессов, таких как бухгалтерский учет, отчетность, делопроизводство. Напрямую на увеличение доходов такая автоматизация не влияет, но помогает сократить время и издержки на ведение рутинной работы.

В результате автоматизации управления бизнес-процессами руководитель предприятия получает больше информации для анализа бизнес-процессов в виде отчетов и имеет возможность качественно управлять компанией, также увеличивается скорость принятия, обработки и передачи заказа.

Одним из популярных приложений по автоматизации предприятия является «РемОнлайн», оно позволяет автоматизировать важные бизнес-процессы, но если требуется автоматизировать только один процесс, то функционал данного приложения будет излишним. Также стоимость подобных

приложений достаточна высока на рынке. Для таких случаев намного проще на сегодняшний день создать веб-приложение, которое будет отвечать поставленным требованиям. Создание веб-приложения позволит существенно сократить растраты на приобретение дорогостоящего приложения, но все также позволит осуществить автоматизацию бизнес-процесса. На предприятии ООО «Шинный двор» клиенты вынуждены получать заказ на бумажном носителе и передавать его от кассира механику, обслуживающему автомобиль. Данные о выполненных работах на бумажном носителе нередко теряются. Создание веб-приложения позволит ускорить процесс принятия и передачи заказа, а также позволит сохранять отчеты, необходимые директору для отслеживания выполненных работ и использованных материалов в электронном виде, что позволит избежать потерь информации.

1.2 Основная деятельность предприятия ООО «Шинный двор»

ООО «Шинный двор» осуществляет свою деятельность на территории республики Хакасия.

Юридическое название организации – Общество с Ограниченной Ответственностью «Шинный двор», находящееся по адресу: 655000, респ. Хакасия, г. Абакан, ул. Советская, д. 148.

Организационно-правовая форма: общество с ограниченной ответственностью.

Предметом деятельности общества является финансовая и производственно-хозяйственная деятельность, направленная на удовлетворение общественных потребностей в услугах производственно-технического назначения, выполнение иных работ и оказание услуг.

Компания ООО «Шинный двор» работает с 2014 года. За это время она добилась отличных результатов в области замены автозапчастей.

Целью ООО «Шинный двор» является подбор и установка подходящих шин для автомобиля. Также компания специализируется на ремонте протектора шин, замене покрышек и профессиональной балансировке колес.

ООО «Шинный двор» ставит перед собой выполнение следующих задач:

- 1) обеспечение клиентов лучшими шинами для их автомобилей;
- 2) повышение рентабельности бизнес-процессов;
- 3) расширение точек продаж на всей территории РФ.

Одним из основных направлений в деятельности предприятия является обслуживание колес автомобилей клиента в сервисе. Материалы, предлагаемые компанией, тщательно проверяются. За 8 лет компания накопила внушительный опыт в подборе шин и шиномонтажу.

В процессе работы сотрудникам ООО «Шинный двор» приходится заполнять заказ клиента в бумажном виде. Нередко бумажный носитель теряется или портится, что сказывается на качестве и скорости оказания услуги. (усложняется учет процесса оказания услуги и запасных частей).

В связи с чем, заказчиком принято решение о разработке веб-приложения для автоматизации процесса выполнения заказа. Разрабатываемая информационная система должна максимально упростить взаимодействие кассира с механиками и клиентами. Также информационная система сможет выводить дополнительные сведения по работе шиномонтажа, необходимые для управлческого учета и директора ООО «Шинный двор», а именно можно будет использовать для ведения отчетов, расходных материалов, о том, как работают сотрудники, о клиентах, как часто они пользуются услугами.

Для детального разбора проблемных бизнес-процессов, строятся функциональные модели «как есть» и «как будет». Использована графическая нотация IDEF0, так как она позволяет создать функциональную модель, отображающую структуру и функции взаимодействия кассира с клиентами и шиномонтажном, а также потоки информации, связывающие эти функции.

Процесс оформления заказ наряда для выполнения работ начинается с заполнения данных о клиенте и автомобиле. Кассир узнает у клиента имя,

марку и модель автомобиля, гос. номер авто, услуги, необходимые клиенту. После оформления заказа клиент с заказ нарядом направляется в шиномонтаж, где передает бумагу механику. Как только механик получает заказ наряд, он начинает обслуживание автомобиля, попутно заполняя данные о выполненных работах и использованных материалах. При окончании работ, механик заполненную бумагу отдает клиенту. Клиент направляется обратно к кассиру, передает лист с заполненными данными. Кассир производит расчет стоимости услуг и принимает оплату от клиента. Далее на рисунке представлена декомпозиция функциональной диаграммы (рисунок 1.2.1).

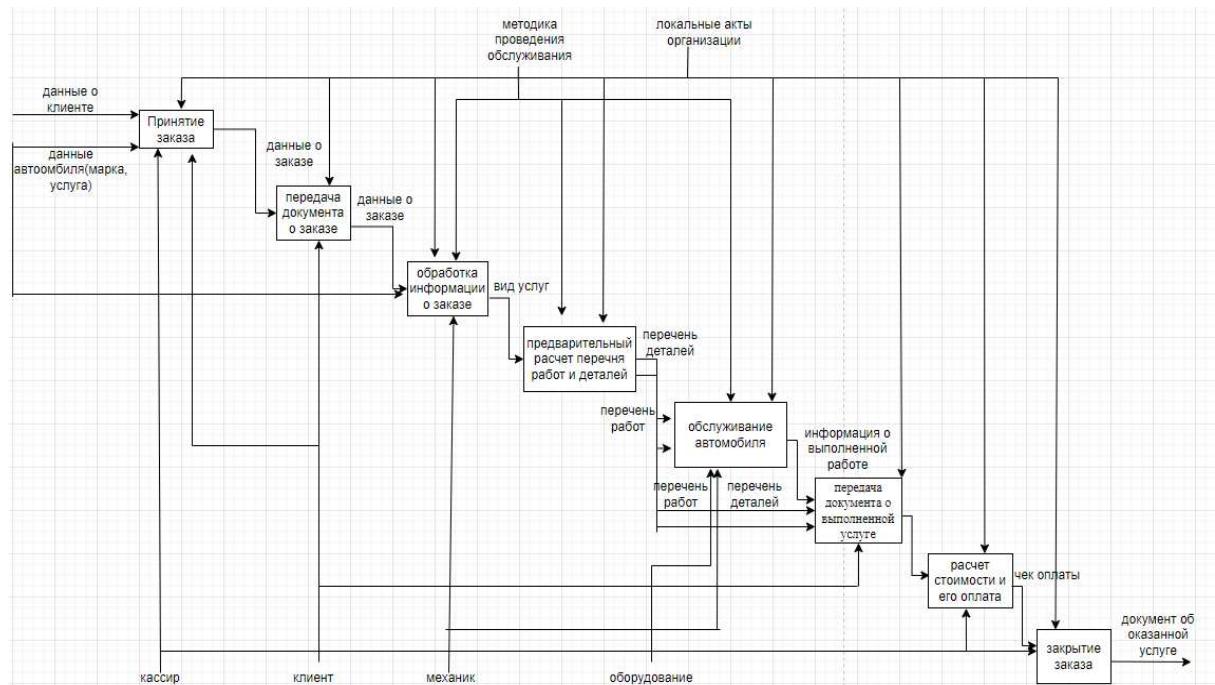


Рисунок 1.2.1 – Декомпозиция функциональной диаграммы ООО
«Шинный двор» «как есть»

Основной недостаток действующего бизнес-процесса заключается в том, что клиенту приходится передавать данные о заказе в шиномонтаж и обратно кассиру. Данные находятся на бумажном носителе и часто теряются или портятся.

Далее рассмотрим диаграмму «как будет». Диаграмма представлена на рисунке 1.2.2.

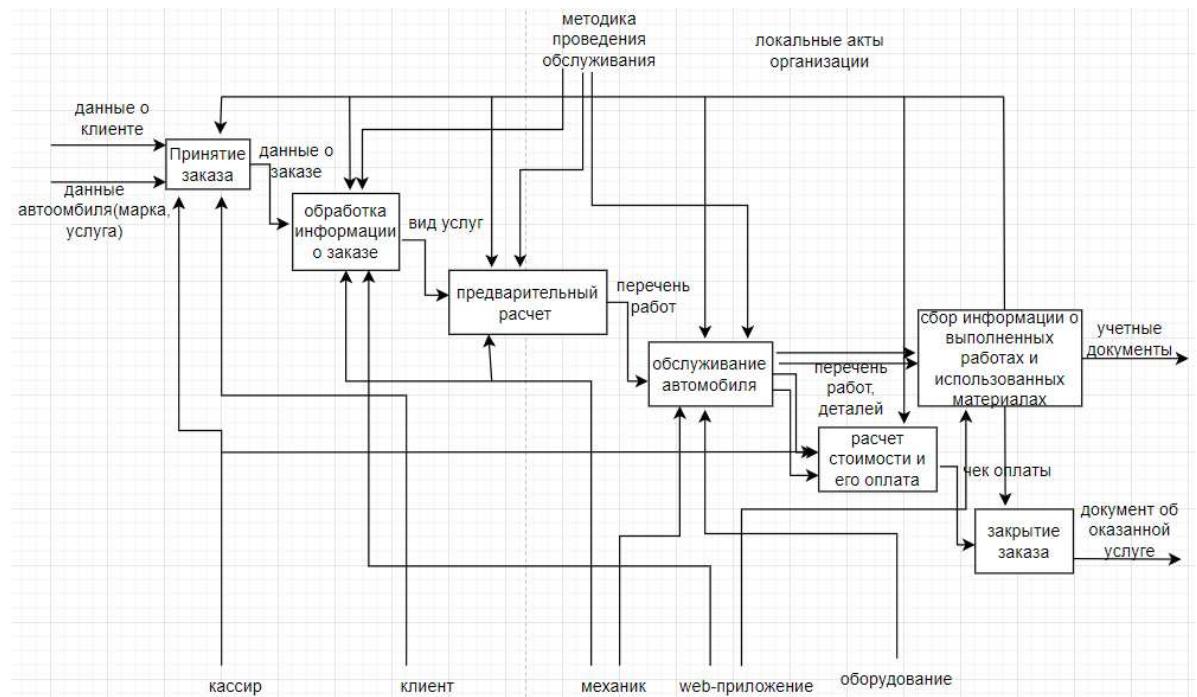


Рисунок 1.2.2 – Функциональная диаграмма ООО «Шинный двор» «как будет»

Главная задача – сокращение времени на этапах принятия заказа клиентов, а также при передаче информации в шиномонтаж. Кроме автоматизации принятия заказа, также есть возможность создания отчетов по выполненным работам и использованным материалам.

Единое информационное пространство предприятия создается из таких составляющих, как хранение необходимых сведений о деятельности шиномонтажа, что позволяет получить максимально полную информацию о деятельности организации.

1.3 Структурная схема предприятия ООО «Шинный двор», обоснование необходимости проекта для деятельности предприятия

Магазин находится под управлением индивидуального предпринимателя. Штат магазина состоит из директора, продавцов консультантов, бухгалтера, кассира, а также работников склада и механиков. Структурная схема предприятия представлена на рисунке 1.3.1.

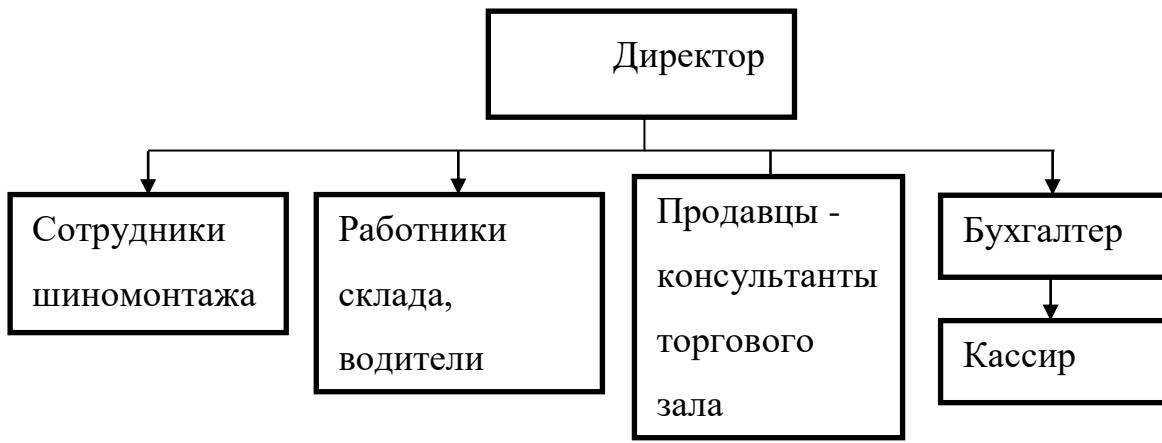


Рисунок 1.3.1 – Организационная структура ООО «Шинный двор»

Директор осуществляет контроль за деятельностью предприятия. Бухгалтер занимается документальным ведением финансово-хозяйственного учета предприятия. Продавцы – консультанты помогают клиенту выбрать товар, отвечает на вопросы. Работники склада выполняют сборку и отгрузку товара покупателям, приём нового товара, инвентаризацией. Кассир производит операции с денежными средствами, формирует кассовые чеки, а также заполняет документы, необходимые для сервисных услуг. Механики обслуживают автомобили, заменяют или ремонтируют шины, производят балансировку колес.

Пользоваться системой будут не все работники ООО «Шинный двор», а только кассиры, механики и директор. Кассир будет заносить данные о клиенте и заказе в бланк заказа, и передавать его напрямую механикам через веб-приложение. Механик сможет выбрать автомобиль и занести дополнительную информацию в бланк заказа. Директор сможет просматривать отчеты по выполненным работам и использованным материалам.

1.4 Характеристика ИТ-инфраструктуры предприятия ООО «Шинный двор» в контексте взаимосвязи с выбранной темой проекта

В ООО «Шинный двор» установлены компьютеры в кабинете директора, бухгалтера, на постах продавцов-консультантов и кассира. Конфигурации компьютеров имеют незначительные отличия. Характеристика компьютеров: - процессор IntelCorei5 – 2400; - оперативная память 4 GbDDR2 1250MHZ; - жесткий диск. HDD 500 Gb. На месте кассира также используется многофункциональное устройство, включающее в себя принтер и сканер. Установленные на предприятии компьютеры объединены в локальную вычислительную сеть, используя маршрутизатор фирмы (Eltex), коммутатор фирмы (d-link). Схема сети предприятия представлена на рисунке 1.4.1

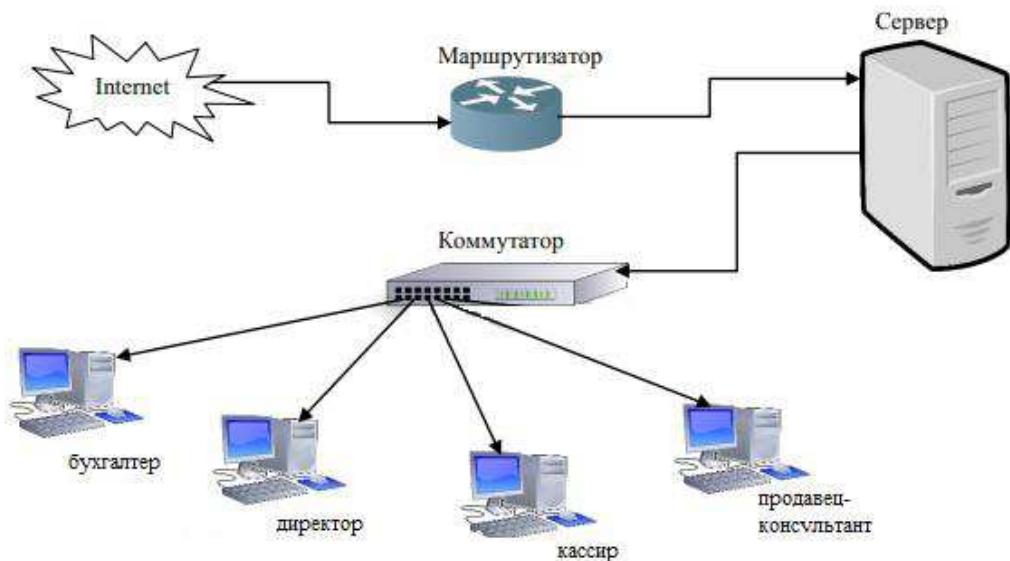


Рисунок 1.4.1 – Схема сети ООО «Шинный двор»

В ООО «Шинный двор» также существует автоматизация бизнес-процессов, построенная на типовом программном продукте «1С: Предприятие». Данная программа автоматизирует бухгалтерский и управленческий учет, работу персонала с базой данных товаров.

В качестве автоматизации процесса принятия и передачи заказа на предприятии будет использоваться веб-приложение, так как в перспективе ООО «Шинный двор» планирует открытие нескольких мастерских и в целях облегчения добавления новых пользователей в информационную систему и хранение всех отчетов в одном месте.

Также у механиков отсутствует локальное подключение к сети, в связи с чем им необходим планшет или смартфон для работы с веб-приложением. Планшет позволит механикам открыть веб-приложение в браузере устройства и принимать заказы, которые заполняет кассир.

1.5 Цель и задачи разработки проекта

Цель работы – повышение скорости принятия и передачи заказа клиента, а также составления отчетов по материалам и работам, за счет внедрения веб-приложения на предприятии ООО «Шинный двор». На данный момент в процессе работы сотрудникам ООО «Шинный двор» приходится заполнять заказ клиента в бумажном виде. Нередко бумажный носитель теряется или портится, что сказывается на качестве и скорости оказания услуги, и усложняется учет процесса оказания услуги и запасных частей.

Для достижения поставленной цели были сформулированы следующие задачи:

- провести анализ работы ООО «Шинный двор»;
- провести обзор современных технологий разработки web-приложений;
- выбрать метод проектирования web-приложения;
- разработать структуру web-приложения «Шинный двор»;
- создать web-приложение «Шинный двор»;
- рассчитать показатели экономической эффективности проекта.

1.6 Анализ типовых проектных решений подобных проектов

Проведем анализ предметной области, для этого найдем аналогичные решения и сравним их между собой. В процессе сравнения можно выявить, какое оформление и функционал используют другие информационные системы и как они реализуют свой программный продукт.

Начнём разбирать аналогичные решения с информационной системы «Help Desk».

Таблица 1.6.1 – Анализ приложения «Help Desk»

Плюсы	Минусы
регистрация клиентов и хранение информации о них (единая база данных)	работа в системе доступна только административному персоналу
бронирование работ на определенную дату и время	основной способ размещения приложения – локально на компьютере
существуют панели руководителя, для управления работой	имеет очень большой функционал, который не будет использоваться
интеграция с большим количеством сервисов и компаний	слишком трудна в освоении персоналом

Разберем другое аналогичное решение, программу для учета заказов «РемОнлайн».

Таблица 1.6.2 – Анализ приложения «РемОнлайн»

Плюсы	Минусы
контроль сотрудников, выполненных работ и сроков реализации	имеет очень большой функционал, который не будет использоваться
автоматизация полного цикла заказа от заявки до повторного визита	цена программы
учет клиентов, товаров, финансов	слишком трудна в освоении персоналом

Теперь подведём итог – два приложения «Help Desk» и «РемОнлайн». Они имеют общее назначение, но разную реализацию, стоит выделить важное среди их функционала. В данном проекте нужна мультиплатформенность, которой нет в «Help Desk». Исходя из аналитических действий, данные предложения не соответствуют заказу, так как имеют очень большой функционал, а также высокую цену. Разработка web-приложения по автоматизации процесса выполнения заказа необходима, нужно будет реализовать заявленный функционал с максимальной эффективностью.

1.7 Выбор средств проектных решений

1.7.1 Выбор архитектуры информационной системы

Для разработки web-приложений наиболее популярны такие языки, как JavaScript, PHP и Python. Рассмотрим каждый язык подробнее.

JavaScript – прототипно-ориентированный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Область его применения обширна и практически безгранична. На JavaScript пишут серверные, мобильные и компьютерные приложения. Любой браузер и любая операционная система хорошо знакома с JavaScript. Все сценарии выполняются непосредственно в браузере устройства, пользователю не нужно предпринимать каких-либо действий. JavaScript используется в клиентской части веб-приложений: клиент-серверных программ, в котором клиентом является браузер, а сервером — веб-сервер, имеющих распределённую между сервером и клиентом логику. Обмен информацией в веб-приложениях происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами.

Основные плюсы JavaScript:

- скорость работы и производительность;
- мощная инфраструктура (экосистема);
- простота и рациональность применения;
- удобство пользовательских интерфейсов.

Проанализируем достоинства и недостатки следующего языка программирования – PHP.

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.

Достоинства языка:

- мощный и гибкий;
- дополняемость. PHP имеет большое количество дополнительных расширений и библиотек, которые увеличивают его функциональность;
- не требователен. Этот язык можно применять на всех известных видах серверов;
- простота редактирования. Писать и редактировать этот код можно в любом редакторе текста.

Однако язык PHP не идеальный, и это подтверждают его следующие минусы:

- на нем невозможно создать десктопное приложение или какой-нибудь системный компонент;
- у приложений на PHP более низкая защищенность, чем с использованием других языков;
- слабая возможность работы с глобальными исключениями.

Далее проанализируем язык программирования – Python.

Python – это универсальный современный язык программирования, к преимуществам которого относят высокую производительность программных решений и структурированный, хорошо читаемый код. Синтаксис Питона максимально облегчен. Ядро имеет очень удобную структуру, а широкий

перечень встроенных библиотек позволяет применять внушительный набор полезных функций и возможностей. Язык программирования может использоваться для написания прикладных приложений, а также разработки WEB-сервисов.

Достоинства языка:

- прост в освоении;
- множество доступных сред разработки;
- универсальный;
- быстрорастущий.

Из минусов можно выделить:

- не самый быстрый среди языков программирования. Скорость выполнения программ может быть ниже;
- не самый удобный язык для мобильных разработок;
- из-за гибкости типов данных потребление памяти Python не минимальное.

Для разработки данной системы был выбран язык JavaScript и PHP. Одна из основных причин – возможность создания web-приложения с удобным интерфейсом.

1.7.2 Выбор инструментальных средств разработки

Определившись с языком программирования, необходимо выбрать среду, в которой будет разрабатываться web-приложение. Для разработки наиболее популярны три варианта: Visual Studio Code, Sublime Text и WebStorm. Сравним эти три среды и выберем наиболее подходящую из них. Сравниваемые характеристики сред разработки представлены в таблице 1.7.1.

Таблица 1.7.1 – Среды разработки

Visual Studio Code	Sublime Text	WebStorm
Интерфейс можно разделить на две панели для сравнения кода	Кликабельность ссылок отсутствует	Поддерживается множественная вложенность
Написание кода для конкретной задачи с его последующей интеграцией в проект (с надстройкой или напрямую)	Позволяет настраивать множество функций, включая: привязки клавиш, меню, фрагменты, макросы и многие другие	Данная среда разработки является платной, пробный период также не поддерживается производителем
Обширная библиотека шаблонов, готовых фрагментов кода и сниппетов с возможностью добавления своих элементов	В редакторе доступно множество плагинов, которые интегрируются в одном месте	Среда обеспечивает автодополнение, анализ кода на лету, навигацию по коду, отладку, и интеграцию с системами управления версиями
Одновременная работа с несколькими проектами (в нескольких окнах)	Нет графического интерфейса	Полная поддержка всех известных языков программирования.

Проведем анализ, сначала – Visual Studio Code, редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как редактор кода для кроссплатформенной разработки веб и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса. Имеет широкие возможности для настройки: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом.

Проанализируем Sublime Text, кроссплатформенный текстовый редактор, разработанный для пользователей, которые ищут эффективный, но минималистский инструмент для редактирования кода. Редактор, в котором отсутствуют панели инструментов или диалоговые окна. Данная среда разработки поддерживает различные языки программирования, в ней можно создавать различные веб-сайты при помощи доступных языков.

Далее рассмотрим следующую среду разработки – WebStorm.

WebStorm — это интегрированная среда для разработки на JavaScript и связанных с ним технологиях. Среда обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами

управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами. Поддерживается множественная вложенность. Главным минусом данной системы является ее цена 2000 рублей в месяц.

Для разработки данной системы была выбрана среда разработки Visual Studio Code. Одни из основных причин – удобный интерфейс, поставляется бесплатно.

1.7.3 Выбор системы управления базами данных для разработки ИС

Выбор системы управления баз данных (СУБД) представляет собой одну из важных этапов при разработке приложений баз данных. Выбранная информационная система должна удовлетворять как текущим, так и будущим потребностям предприятия. Проведя анализ можно сделать вывод, что наиболее популярными являются Oracle, MySQL и PostgreSQL. Рассмотрим каждую СУБД подробнее.

Oracle Database – объектно-реляционная система управления базами данных компании Oracle. Одна из отличительных особенностей сервера Oracle — возможность хранения и обработки различных типов данных. Данная функциональность интегрирована в ядро СУБД и поддерживается модулем interMedia в составе Oracle Database. Он обеспечивает работу с текстовыми документами, включая различные виды поиска, в том числе:

- контекстного;
- работу с графическими образами более двадцати форматов;
- работу с аудио и видеоинформацией.

Минусами данной СУБД является цена, она может быть непомерно высокой, особенно для небольших организаций. Так же требуется модернизация оборудования, для внедрения СУБД.

Проанализируем достоинства и недостатки следующей СУБД – MySQL.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей.

Основные преимущества MySQL:

- многопоточность, поддержка нескольких одновременных запросов;
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- гибкая система привилегий и паролей;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- быстрая работа, масштабируемость;
- бесплатна в большинстве случаев.

Эта серверная система способна эффективно функционировать во взаимодействии с интернет-сайтами и веб-приложениями. При этом она проста в освоении.

Далее рассмотрим следующую СУБД – PostgreSQL.

PostgreSQL – свободная объектно-реляционная система управления базами данных.

Одной из сильных сторон PostgreSQL является ее архитектура. Как и многие коммерческие СУБД, может применяться в среде клиент-сервер, что дает массу преимуществ как пользователям, так и разработчикам. Основа PostgreSQL составляет серверный процесс базы данных. Он выполняется на

одном сервере. Доступ из приложений к данным базы осуществляется посредством процесса базы данных. Клиентские программы не могут получить доступ к данным самостоятельно, даже если они работают на том же компьютере, на котором выполняется серверный процесс.

Недостатки СУБД:

- не очень хороша в плане производительности;
- что касается скорости, PostgreSQL не заслуживает внимания по сравнению с другими инструментами;
- делать репликацию сложнее;
- установка не легка для новичка.

Для разработки данной системы была выбрана система управления базами данных MySQL. Одна из основных причин – гибкость СУБД, наличие большого количества типов таблиц, а также, что СУБД является бесплатной.

1.8 Разработка модели проектируемой информационной системы

Теперь разберем, как пользователи «Механик», «Директор» и «Кассир» будут работать с моделируемой информационной системой. Для того чтобы это смоделировать, воспользуемся нотацией IDEF3. Данная нотация позволит описать процесс работы пользователя с информационной системой. На рисунках 1.8.1 — 1.8.3 изображены смоделированные работы пользователей с использованием нотации IDEF3.

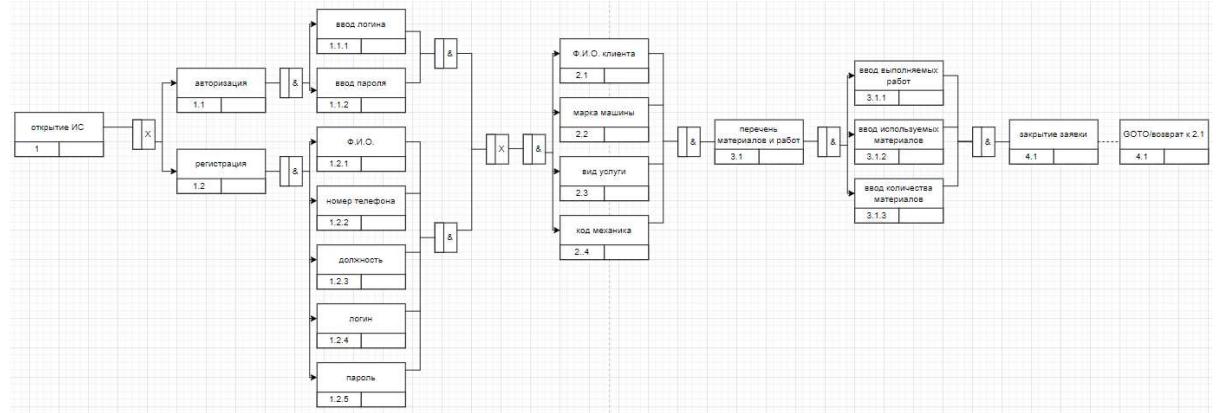


Рисунок 1.8.1 – Смоделированная работа пользователя «Механик» с использованием нотации IDEF3

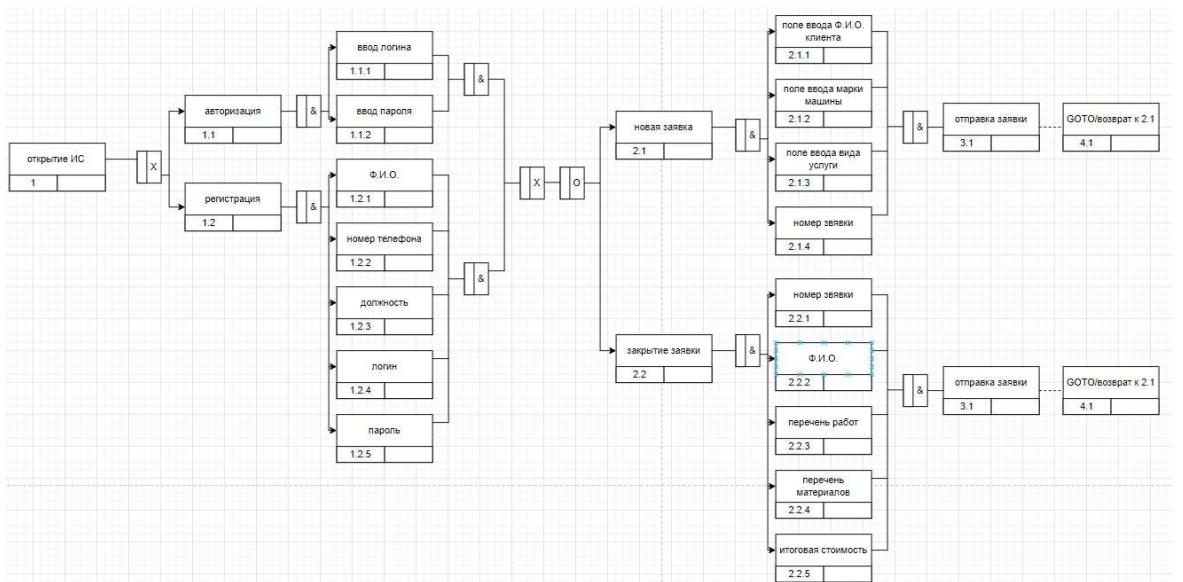


Рисунок 1.8.2 – Смоделированная работа пользователя «Кассир» с использованием нотации IDEF3

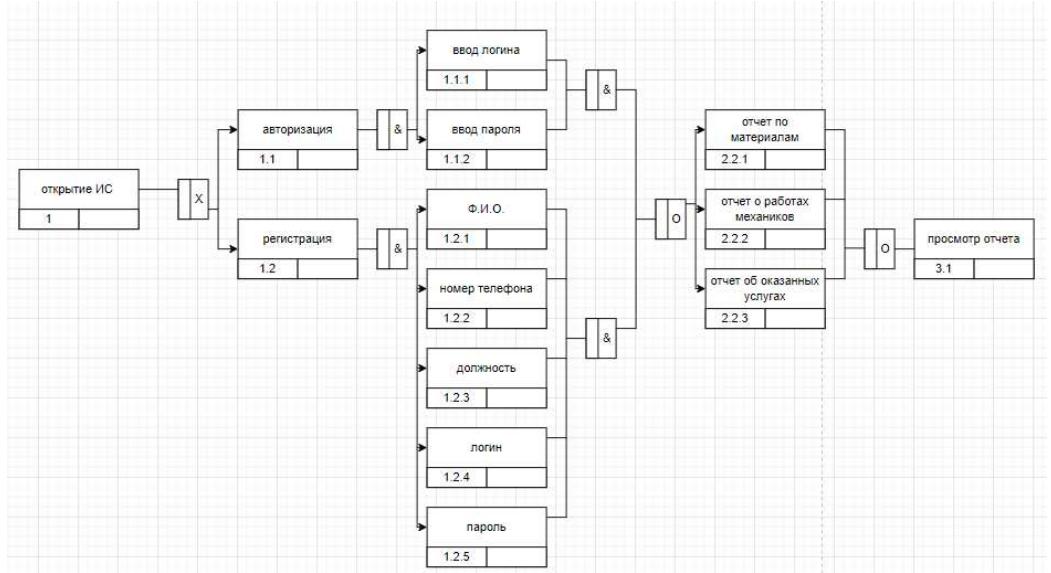


Рисунок 1.8.3 – Смоделированная работа пользователя «Директор» с использованием нотации IDEF3

Опишем диаграммы, изображенные на рисунках 9.1-9.3. Изначально работнику ООО «Шинный двор» необходимо зарегистрироваться для работы с ИС. Следующим шагом пользователь авторизуется и заходит в систему под своей учетной записью.

Далее разберем, какой функционал доступен пользователям в системе. Пользователь «механик» может просматривать доступные для исполнения заказы, указывать какие работы были выполнены и материалы использованы, так же передавать эти сведения кассиру для формирования документа об оказанной услуге.

Пользователь «Кассир» заполняет бланки заказа, вводит имя пользователя, марку автомобиля, телефон и услуги необходимые клиенту. Так же кассир закрывает заказы, рассчитывает стоимость выполненных услуг, из сведений, переданных механиком, и выставляет счет клиенту.

Пользователь «Директор» имеет возможность просматривать отчеты по выполненным работам, затраченным материалам и работам механиков за период времени. В отчете будет отображаться Кассир, создавший и заполнивший заказ, ФИО клиента, марка и модель автомобиля, предоставленные услуги и используемые для этого материалы, ФИО Механика,

выполнявшего работы, и стоимость комплекса выполненных работ.

Была смоделирована работа пользователя в информационной системе, были основаны связи между единицами работ, их перекрестки. Благодаря этому можно понять, куда пользователь может перейти, с чем он сможет взаимодействовать и при каком условии пользователь сможет воспользоваться соответствующей функцией информационной системы.

1.9 Выводы по разделу «Анализ предметной области и выбор средств проектных решений»

В разделе была проанализирована основная деятельность организации ООО «Шинный двор», а также ее сотрудников и функций, которые выполняет каждый из них в частности. Также построены функциональные диаграммы «как есть» и «как будет». Анализ эффективности применяемых средств показал, что процесс принятия и передачи заказа посредством бумажных носителей малоэффективен, частота выполнения заказов снижается. В качестве цели проекта выбрана разработка веб-приложения по автоматизации процесса выполнения заказа на предприятии. Веб-приложение проще ввести в существующее предприятие, также компания планирует расширять точки продаж и шиномонтажа, из-за этого гораздо удобнее предоставить доступ к веб-приложению новым пользователям. Веб-приложение будет состоять из блоков, а именно:

1. блок личного кабинета для распределения ролей в системе;
2. база данных, содержащая информацию о кассирах, механиках и директоре, а также информацию о выполненных работах, использованных материалах, клиентах и автомобилях;
3. блок заполнения заказа кассиром;
4. блок подтверждения выполнения заказа механиком, а также дополнение выполняемых работ и используемых материалов;
5. блок отчетов для директора, где будет возможность в реальном

времени просматривать информацию о завершенных заказах и сохранять готовые отчеты в Excel файле.

Выбраны языки программирования PHP и JavaScript, так как имеют большой функционал, и большинство сайтов разработаны с их помощью. Сформулирован и аргументирован список программных средств для создания веб-приложения. Необходимо разработать веб-приложение, позволяющее быстро заполнять и передавать информацию между работниками ООО «Шинный двор» и позволяющее составлять отчеты, с помощью которых будет отслеживаться работа персонала и поток клиентов.

2 Описание разработки веб-приложения приема заказов для ООО «Шинный двор»

2.1 Установка необходимого ПО

Перед началом непосредственно разработки программного продукта, нужно определить необходимое программное обеспечение, с использованием которого будет произведена разработка.

В качестве основного инструмента разработки будет использован ReactJS, так как он соответствует основным современным требованиям к разработке, а также различными разработчиками сконструировано множество дополнительных библиотек для расширения функционала и улучшения процесса разработки.

Для использования ReactJS необходимо наличие установленного «Node.js» на компьютере разработчика.

«React.js» - программная платформа, позволяющая превращение JavaScript кода в машинный код, исполняемый на стороне сервера.

В комплекте с React-ом устанавливаются:

1. Babel - бесплатный транс компилятор JavaScript с открытым исходным кодом, который в основном используется для преобразования кода ECMAScript 2015+ в обратно совместимую версию JavaScript.
2. Webpack - сборщик модулей, анализирующий модули приложения, создает граф зависимостей, затем собирает модули в правильном порядке.

Для установки React в командной строке нужно использовать команду «`npx create-react-app название проекта`», после чего загруженный шаблон проекта открыть в VS Code.

Установить VS Code можно с официального сайта разработчика [«code.visualstudio.com»](https://code.visualstudio.com) после загрузки установочного файла произвести стандартную установку, не изменяя какие-либо параметры, после чего среда разработки будет установлена и готова к работе, пример установленного VS Code на рисунке 2.1.

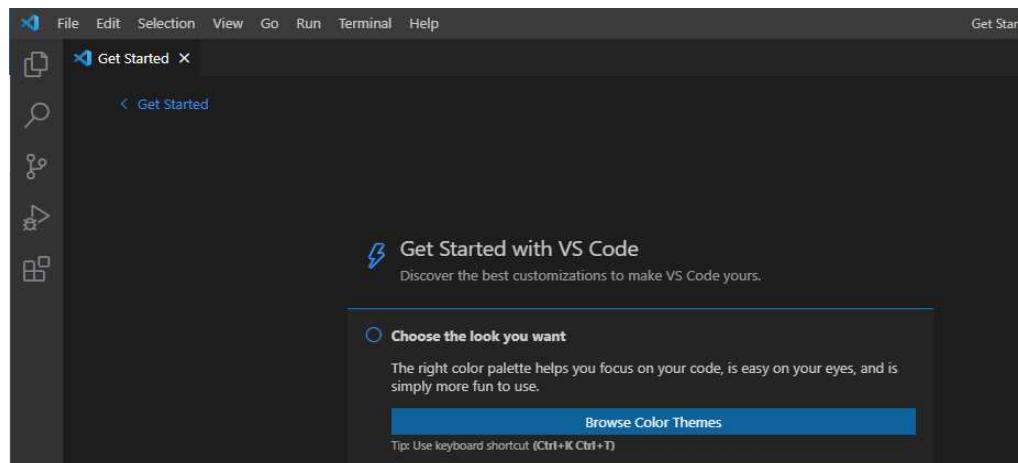


Рисунок 2.1 – VS Code

Для проверки работоспособности проекта в терминале VSCode необходимо ввести команду «`npm start`», после чего будет запущен проект ReactJS, пример запущенного проекта на рисунке 2.2.

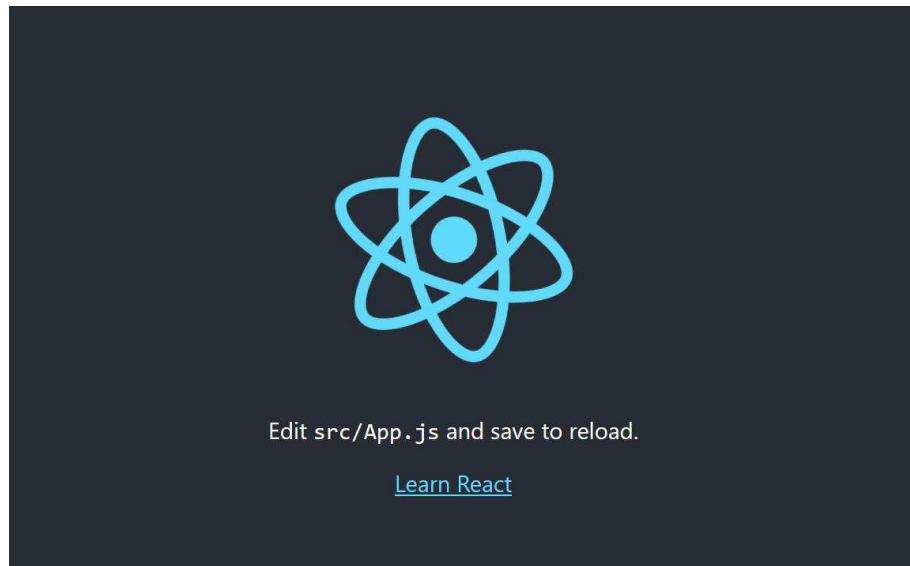


Рисунок 2.2 – Запущенный проект React App

Далее необходимо установить необходимые библиотеки, используемые при разработке программного продукта, например, для установки react-icons, позволяющего использовать различные иконки нужно использовать команду «npm install react-icons --save».

После установки всех библиотек, нужно открыть файл «package.json», в котором будут указаны все установленные библиотеки и описаны настройки приложения, результат установки библиотек изображен на рисунке 2.3.

```
{
  "name": "tires-v1",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@emotion/react": "^11.9.0",
    "@emotion/styled": "^11.8.1",
    "@mui/lab": "^5.0.0-alpha.76",
    "@mui/material": "^5.6.0",
    "classnames": "^2.3.1",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-final-form": "^6.5.9",
    "react-icons": "^4.3.1",
    "react-intersection-observer": "^8.33.1",
    "react-router-dom": "^6.3.0",
    "react-scripts": "5.0.0",
    "sass": "^1.50.0",
    "web-vitals": "^2.1.4"
  }
}
```

Рисунок 2.3 – Package.json

Для работы с БД будет использован OpenServer.

OpenServer – набор программ, необходимых для запуска локального сервера - программная среда, созданная специально для веб-разработчиков.

Программный комплекс OpenServer включает в себя тщательно подобранный набор серверного программного обеспечения, а также невероятно удобную и продуманную управляющую утилиту, которая обладает мощными возможностями по администрированию и настройке всех доступных компонентов.

После загрузки установочного пакета и последующей установки на ПК, разработчику нужно открыть панель сервера и открыть папку с проектами, в которую нужно загрузить свой проект.

Убедившись в наличии и правильной настройке всех необходимых библиотек, можно начать разработку программного продукта.

2.2 Описание процесса создания веб-приложения приема заказов для ООО «Шинный двор»

После установки всех необходимых программ и дополнительных модулей, для начала разработки программного продукта нужно создать в корне проекта папки «components» и «assets».

В первой папке будет находиться программный код и код CSS – стилей, во второй будут храниться дополнительные необходимые файлы, например – изображения, на рисунке 2.4. отображены добавленные папки в проекте.

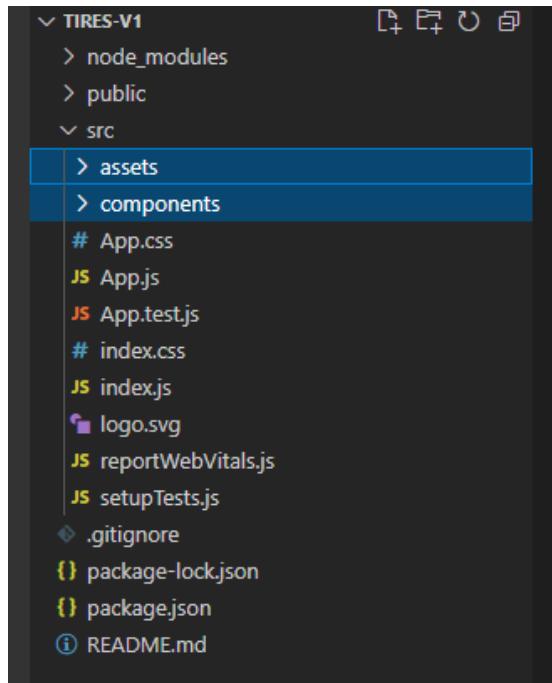


Рисунок 2.4 – Дополнительные папки проекта

В первую очередь нужно изменить и настроить главную страницу приложения.

Для этого необходимо добавить папку «Home» и создать файлы home.js, hero.js и hero.css, файлы «hero.js и css» использованы для добавления и настройки главного изображения приложения, home.css получает информацию из hero.js и добавляет 2 слайдера с дополнительной информацией. Вся информация из функции Home передается на выход – данные из функции можно отобразить в другой функции, например, App.tsx – основной файл, с помощью которого происходит вывод информации, скрипты файлов home.js и hero.js представлены на рисунках 2.5, 2.6.

```
Help Herojs - tires - Visual Studio Code

JS Herojs JS Home.js

src > Features > Home > JS Hero.js > ...
1 import React from "react";
2 import "./hero.css";
3
4 const Hero = ({imageSrc}) =>{
5   return <div className="hero">
6     <img src={imageSrc} alt="Tires" className="hero_image"></img>
7     <h1 className="hero_title">Шиномонтаж Абакан</h1>
8   </div>
9 };
10
11 export default Hero;
```

Рисунок 2.5 – Hero.js

```
Help Homejs - tires - Visual Studio Code

JS Herojs JS Home.js X

src > Features > Home > JS Home.js > Home
1 import bg_intro from "../../Common/assets/bg-intro.jpg";
2 import tires1 from "../../Common/assets/tires1.jpg";
3 import tires2 from "../../Common/assets/tires2.jpg";
4 import Hero from './Hero';
5 import Slider from "../../Common/Components/slider";
6 import React from "react";
7
8 export default function Home() //Функция, выводящая главное изображение и изображения с описанием
9
10 return <>
11   <Hero imageSrc={bg_intro}>/> /* Основное изображение */
12
13 /* Дополнительные изображения */
14 <Slider
15   imageSrc={tires1}
16   title={"Лучший шиномонтаж в Абакане"}
17   subtitle={"Лучшие специалисты, цены и условия работы"}
18 />
19 <Slider
20   imageSrc={tires2}
21   title={"Оформляйте заказы"}
22   subtitle={"Выполним все качественно и в кратчайшие сроки"}
23   flipped = {true}
24 />
25 </>
26 ]]
```

Рисунок 2.6 – Home.js

Использование выходных данных из разных файлов реализовано с помощью импорта – функция `import` позволяет получить данные из сторонней функции при использовании оператора `export`.

Итог работы над скриптами представлен на рисунке 2.7.



Рисунок 2.7 – Главная страница

Для сохранения, изменения и просмотра записей необходимо использовать хранилище данных – базу данных (БД), в данном проекте БД будет разработана в PhpMyAdmin. После авторизации в качестве администратора, нужно добавить новую БД, для этого в левом меню нужно нажать «Создать БД», пример на рисунке 2.8.

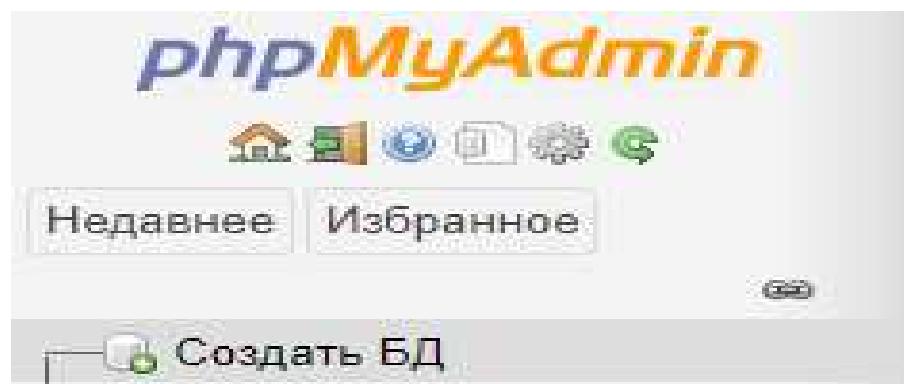


Рисунок 2.8 – PhpMyAdmin

После чего нужно ввести название БД, и выбрать «Создать». После добавления новой БД, для полноценной функциональности используются таблицы, в которых хранятся записи в зависимости от типа данных.

Добавление новых таблиц производится в директории БД, для этого необходимо:

- ввести название таблицы;
- указать количество строк;
- тип данных;
- определить основную кодировку, используемую для работы.

После настройки таблиц, необходимо настроить первичные и вторичные ключи и связи между таблицами, после чего спроектировать схему данных БД.

Итоговая схема данных представлена на рисунке 2.9.

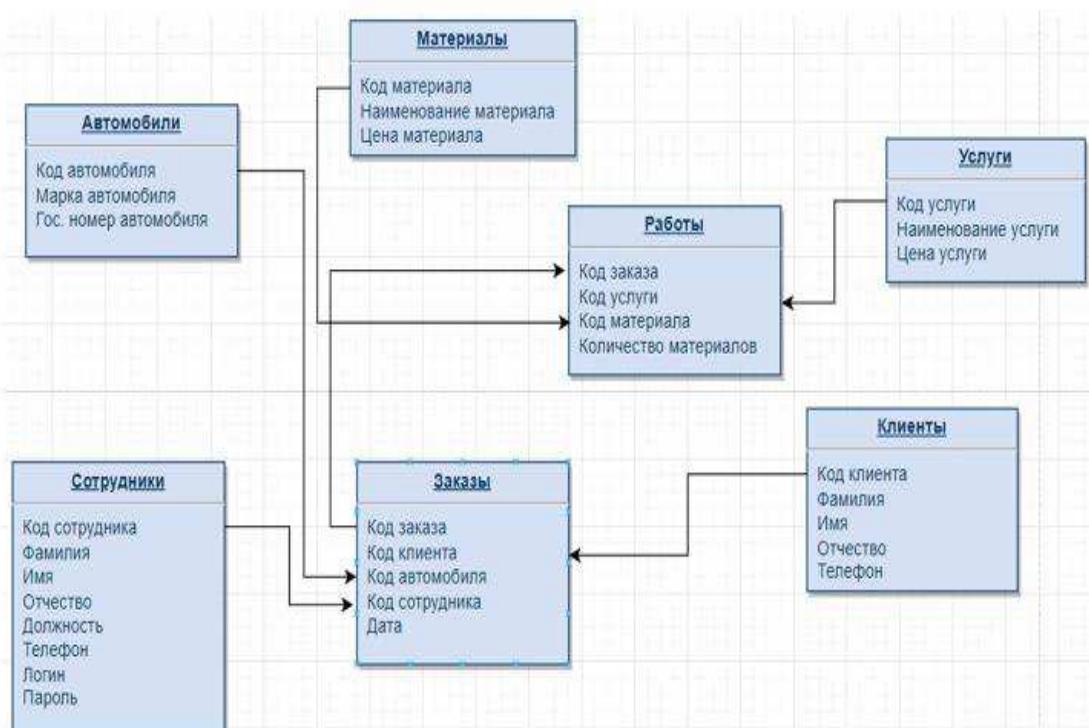


Рисунок 2.9 – Схема БД

Основной функционал работы сайта реализован с помощью роутинга – инструмента, отвечающего за отображение компонентов по маршруту из адресной строки.

Роутер позволяет удобно настраивать и описывать переходы между компонентами без многократного дублирования информации.

Также с помощью ДОМ дерева, реализуемого с помощью роутера проще ориентироваться в иерархии проекта, пример на рисунке 2.10.

```
function App() {
  return <div className="App">
    <ProfileProvider>
      <Router>
        <Routes>
          <Route
            element={
              <AuthCommonPage />
            }
          <Route
            path="authorization"
            element={
              <AuthorizationPage />
            }
          />
          <Route
            path="registration"
            element={
              <RegistrationPage />
            }
          />
        </Routes>
        <Route element={<ProtectedPage><CommonPage /></ProtectedPage>}>
          <Route
            path='/'
            element={<Home />}
          />
          <Route
            path='/cashier'
            element={<CashierProtectedPage><CashierPage /></CashierProtectedPage>}
          />
          <Route
            path='/tire_fitter'
            element={<TireFitterProtectedPage><TireFitterPage /></TireFitterProtectedPage>}
          />
        </Route>
      </Router>
    </ProfileProvider>
  </div>
}
```

Рисунок 2.10 – React Router

В данном программном коде прописаны основные компоненты сайта и организован переход между ними – route element.

Отображение дочерних маршрутов реализовано с помощью outlet – подход, в котором маршруты определяются вне компонента, а содержимое добавляется в главный компонент с помощью данного компонента – outlet.

ProtectedPage – параметр роутинга, в котором при обращении по маршруту, который входит в ProtectedPage, проверяется авторизация

пользователя, если авторизация успешна, то отображает дочерние компоненты согласно маршрутам, иначе отправляет на страницу авторизации.

Страница авторизации умный компонент, который является посредником между формой и сервером, пример на рисунке 2.11.



```
import Card from "@mui/material/Card";
import CardContent from "@mui/material/CardContent";
import { useNavigate } from "react-router-dom";
import LoginDto from "../../Data/Models.Dto/LoginDto";
import { useProfile } from "../../System/hooks/useProfile";
import AuthorizationForm from "./AuthorizationForm";

export default function AuthorizationPage() {
  const profileHook = useProfile();
  const navigateHook = useNavigate();

  const onSubmit = (
    formData: LoginDto,
    form: any,
    showErrors: (errors: any) => void
  ) => {
    return profileHook.login(formData)
      .then(() => navigateHook('/', { replace: true }))
  }

  return <div className="authorization-page">
    <Card>
      <CardContent>
        <AuthorizationForm onSubmit={onSubmit} />
      </CardContent>
    </Card>
  </div>
}
```

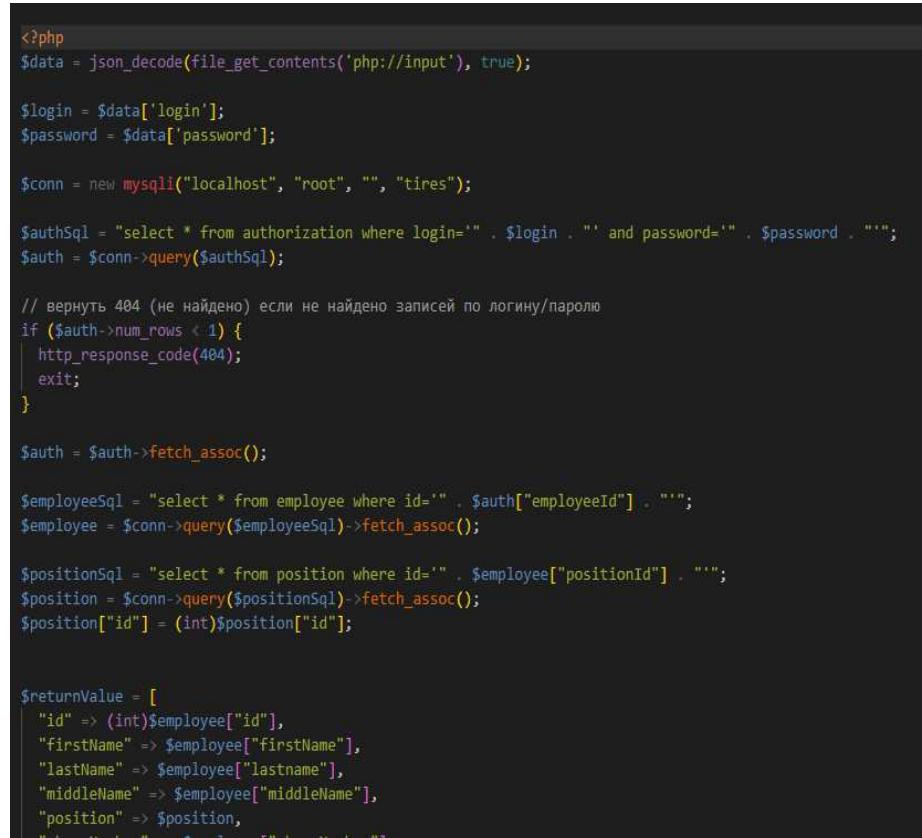
Рисунок 2.11 – Код страницы авторизации

В данном коде при нажатии кнопки «Отправить» вызывается функция onSubmit, которая получает данные из формы и отправляет запрос к серверу, где происходит проверка соответствия.

Данные передаются в php файл через POST запрос, который в дальнейшем производит запись в БД.

POSITIONS.find((x) => x.id === formData.position)!, отвечает за указанную должность.

С помощью LoginDto данные передаются в php файл login.php, в котором происходит соединение с БД и проверка введенных данных, пример кода на рисунке 2.12.



```
<?php
$data = json_decode(file_get_contents('php://input'), true);

$login = $data['login'];
$password = $data['password'];

$conn = new mysqli("localhost", "root", "", "tires");

$authSql = "select * from authorization where login='".$login."' and password='".$password."'";
$auth = $conn->query($authSql);

// вернуть 404 (не найдено) если не найдено записей по логину/паролю
if ($auth->num_rows < 1) {
    http_response_code(404);
    exit;
}

$auth = $auth->fetch_assoc();

$employeeSql = "select * from employee where id='".$auth["employeeId"]."'";
$employee = $conn->query($employeeSql)->fetch_assoc();

$positionSql = "select * from position where id='".$employee["positionId"]."'";
$position = $conn->query($positionSql)->fetch_assoc();
$position["id"] = (int)$position["id"];

$returnValue = [
    "id" => (int)$employee["id"],
    "firstName" => $employee["firstName"],
    "lastName" => $employee["lastname"],
    "middleName" => $employee["middleName"],
    "position" => $position,
];
```

Рисунок 2.12 – PHP файл авторизации

В представленном программном коде идет получение данных из формы предыдущего кода, после чего происходит подключение к БД.

При установлении соединения, выполняется выборка из таблицы и запись данных в переменную \$auth, в случае если получен не пустой набор данных, выполняется изменение переменной в ассоциативный массив, в котором выполняется проверка id пользователя и должности.

После выполнения условий данные отправляются на страницу.

Программный код регистрации выполнен идентичным образом, имея основное отличие в виде inserta данных вместо select.

Страницы для различных типов пользователей реализованы отдельно, в зависимости от типа должности пользователю доступны определенные формы.

Например, форма кассира выполнена следующим образом.

Тип формы зависит от типа состояния формы – close, create, update.

При открытии страницы проверяется запрос на проверку списка заказов – useEffect, если возвращается 200 код(положительный), в этом случае заказы сохраняются в локальную переменную.

onCreate onUpdate выполняются в зависимости от состояния формы, вызов этих функций выполняется из onSubmit. onListItemClick - сохраняет в переменную выбранный заказ и открывает форму в режиме обновления.

onNewClick - открывает форму в режиме создания.

onPaidClick - обновление статуса заказа.

При нажатии на кнопку «Новый заказ», добавляется новая форма, в которую добавляются данные о заказе, после записи которого по умолчанию устанавливается статус созданного заказа, примеры программного кода представлены на рисунках 2.13, 2.15.

```
enum FORM_MODE {
  CLOSE,
  CREATE,
  UPDATE
}

export default function CashierPage() {
  const [orders, setOrders] = useState<Array<Order>>([[]]);
  const [carTypes, setCarTypes] = useState<Array<CarType>>([[]]);
  const [selectOrder, setSelectOrder] = useState<Order | undefined>(undefined);
  const [formMode, setFormMode] = useState(FORM_MODE.CLOSE);
  const [isLoading, setIsLoading] = useState(false);
  const profile = useProfile()

  useEffect(() => {
    setIsLoading(true)
    OrderService.getAll()
      .then((orders) => setOrders(orders))
      .finally(() => setIsLoading(false))
  }, [])

  useEffect(() => {
    CarTypeService.getAll()
      .then((carTypes) => setCarTypes(carTypes))
  }, [])

  const onCreate = (formData: CashierFormFormata) => {
    const carType = carTypes.find((x) => x.id === formData.carTypeId)!
    const order = new CreateOrderRequest(
      carType,
      formData.carNumber,
      formData.clientName,
      formData.clientPhoneNumber,
      formData.description,
      ORDER_STATUS.OPEN,
      profile?.profile?.id
    )
    OrderService.create(order)
      .then((order) => {
        const newOrders = [...orders, order]
        setOrders(newOrders)
      })
      .then(() => setFormMode(FORM_MODE.CLOSE))
  }
}
```

Рисунок 2.13 – Форма кассира, лист 1

```

const onUpdate = (formData: CashierFormFormData) => {
  if (!selectOrder) return;
  OrderService.update(updateOrderDto.fromObject({ ...formData, id: selectOrder.id }))
    .then((order) => {
      const newOrders = orders.map((x) => x.id === order.id ? order : x)
      setOrders(newOrders)
    })
    .then(() => setFormMode(FORM_MODE.CLOSE))
}

const onSubmit = (formData: CashierFormFormData) => {
  if (formMode === FORM_MODE.CREATE) {
    return onCreate(formData)
  }
  if (formMode === FORM_MODE.UPDATE) {
    return onUpdate(formData)
  }
}

const onListItemClick = (order: Order) => {
  setSelectOrder(order)
  setFormMode(FORM_MODE.UPDATE)
}

const onCloseForm = () => {
  setFormMode(FORM_MODE.CLOSE)
  setSelectOrder(undefined)
}

const onNewClick = () => {
  setFormMode(FORM_MODE.CREATE)
  setSelectOrder(undefined)
}

const onPaidClick = (order: Order) => {
  OrderService.updateStatus(order, OrderStatus.DONE, profile.profile?.id)
    .then((order: Order) => {
      const newOrders = orders.map((x) => x.id === order.id ? order : x)
      setOrders(newOrders)
    })
}

```

Рисунок 2.13, лист 2

```

if (isLoading) {
  return <div className="cashier-page">
    <div className="cashier-page__content">
      <Loader />
    </div>
  </div>
}

return <div className="cashier-page">
  <div className="cashier-page__top-panel">
    <Button variant="outlined" onClick={onNewClick}>Новый заказ</Button>
  </div>
  <div className="cashier-page__content">
    <OrderList
      orders={orders}
      onSelect={onListItemClick}
      listItemComponent={CashierOrderListItem}
      onPaidClick={onPaidClick}
    />
    {formMode !== FORM_MODE.CLOSE && <div className="cashier-page__form-wrapper">
      <CashierForm
        initialValues={{ ...selectOrder, carTypeId: selectOrder?.carType.id }}
        carTypes={carTypes}
        onSubmit={onSubmit}
        onCancel={onCloseForm}
      />
    </div>
  </div>
}

```

Рисунок 2.13, лист 3

Итоговое окно авторизации и регистрации продемонстрировано на рисунке 2.16.

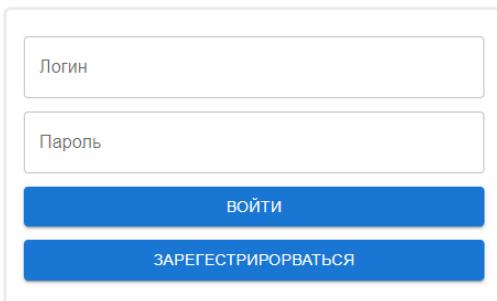


Рисунок 2.14 – Окно авторизации и регистрации

В части React-а нужно добавить верхнее меню для перемещения между формами, а также настроить сами формы отображения оформления заказа для кассира, принятия заказа и возможности добавления дополнительных работ к нему со стороны механика, а также возможность просмотра всех существующих заказов.

Для добавления верхнего меню созданы файлы navbar, в которых прописано отображение логотипа приложения, ссылок для перехода между страницами, а также в CSS части реализован адаптивный вариант меню, со скрытием ссылок в панель при использовании приложения на устройствах с экранами маленького соотношения сторон. Итоговый вариант отображен на рисунках 2.17, 2.18.

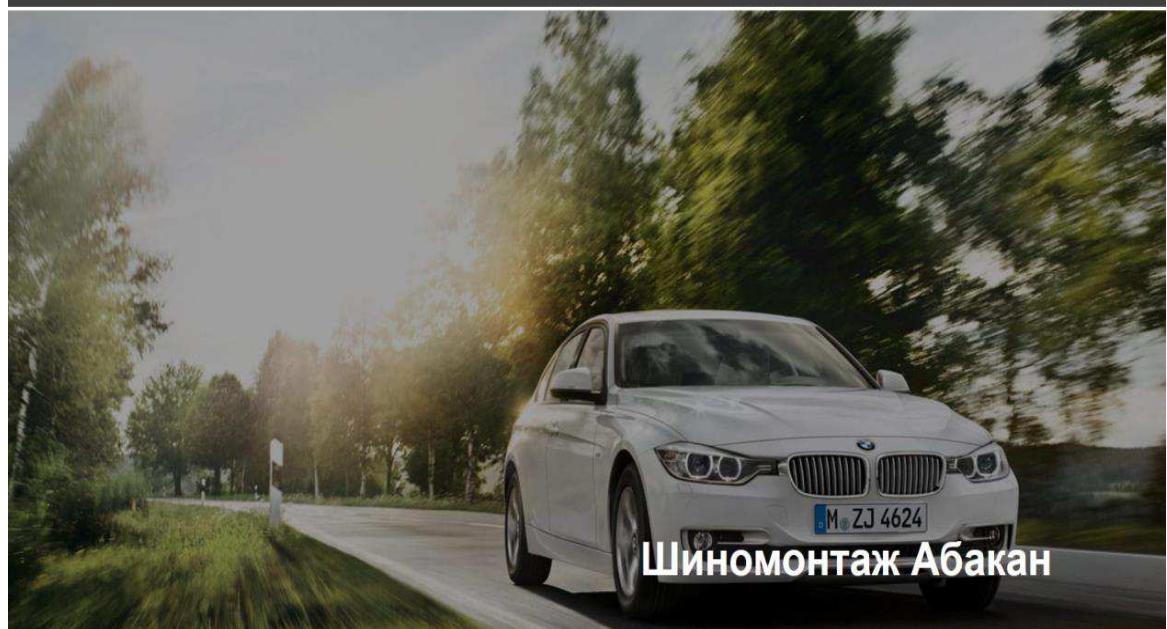


Рисунок 2.15 – Главное окно с панелью

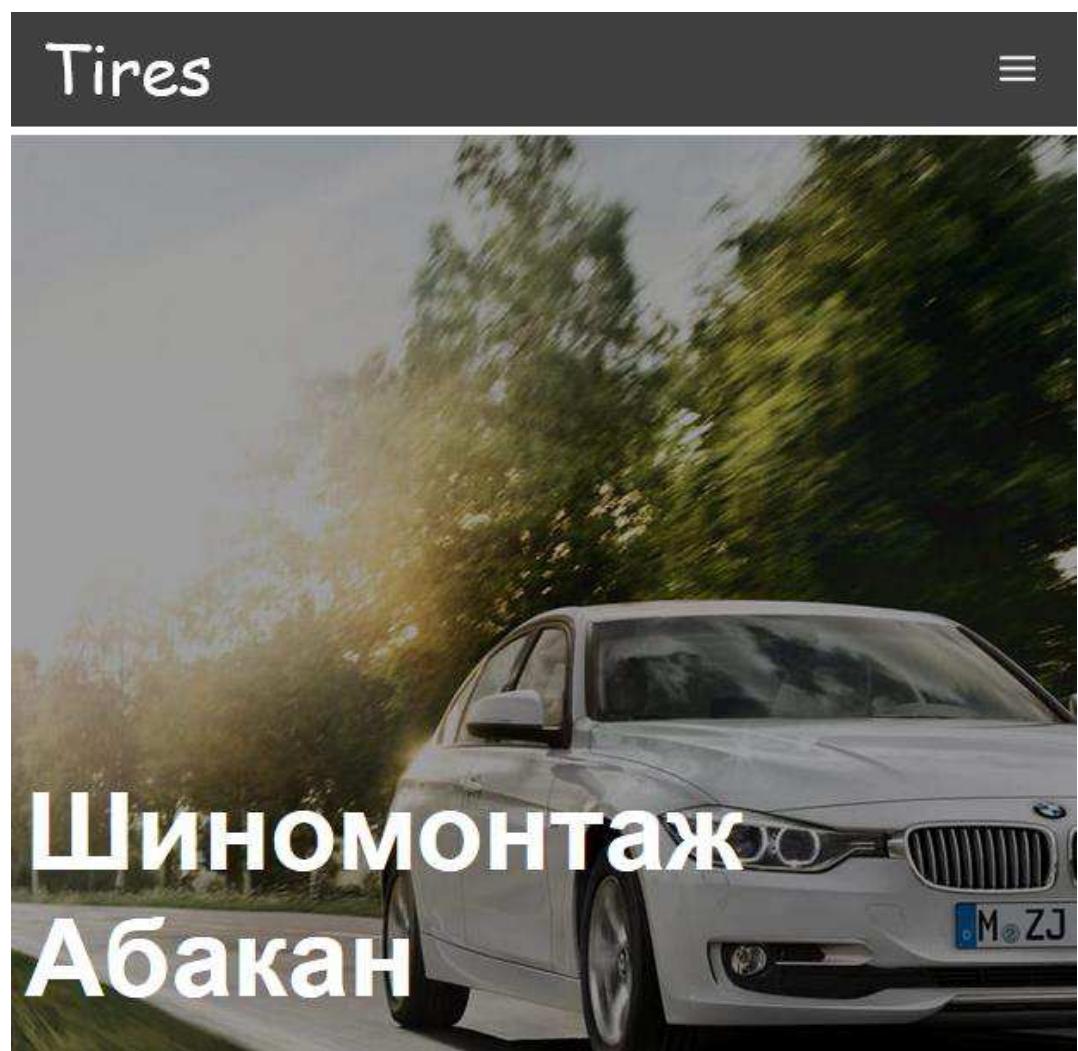


Рисунок 2.16 – Главное меню со скрытой панелью

Тип должности пользователя можно определить по тексту на верхней панели меню: кассир, директор, монтажник.

На панели имеются ссылки, ведущие к различным функциям приложения. Вкладка кассир используется сотрудниками с соответствующей должностью для предварительной записи клиентов в БД, с обобщенной информацией о нужных работах, после чего внесенный заказ переходит к работнику, выполняющему заявленные работы. Он также должен принять заказ, внести необходимую информацию и по завершению работ завершить заказ в окне приложения, пример форм для добавления заказа представлены на рисунке 2.19.

The screenshot shows a software application window. At the top is a dark header bar. Below it, on the left, is a sidebar containing client information: #0, ФИО клиента: FIO, Номер телефона: 83433423412, Марка автомобиля: BMW, Номер автомобиля: T1345TK19, Необходимые работы: Замена шин. To the right of the sidebar is a main form area. At the top right of the form is a blue button labeled 'НОВЫЙ ЗАКАЗ'. Below it, the word 'Создан' is displayed next to a timestamp. The main form consists of several input fields: 'Имя клиента' (Client Name), 'Номер телефона клиента' (Client Phone Number), 'Марка и модель автомобиля' (Car Brand and Model) with a dropdown arrow, 'Номер автомобиля' (Car Number), 'Необходимые работы' (Required Work), and a large red 'ОФОРМИТЬ ЗАКАЗ' (Place Order) button at the bottom. Below the red button is a smaller red 'ОТМЕНА' (Cancel) button.

Рисунок 2.17 – Оформление заказа

После оформления заказа, происходит запись данных в БД, которые в последующем будут отображены на странице работника, ответственного за выполнение работ.

Работнику после авторизации и перехода в меню работ будут отображены существующие заказы со всей необходимой информацией – ФИО клиента,

номер телефона, марка автомобиля, номер автомобиля, а также необходимые работы.

Пример отображения заказа на рисунке 2.20.

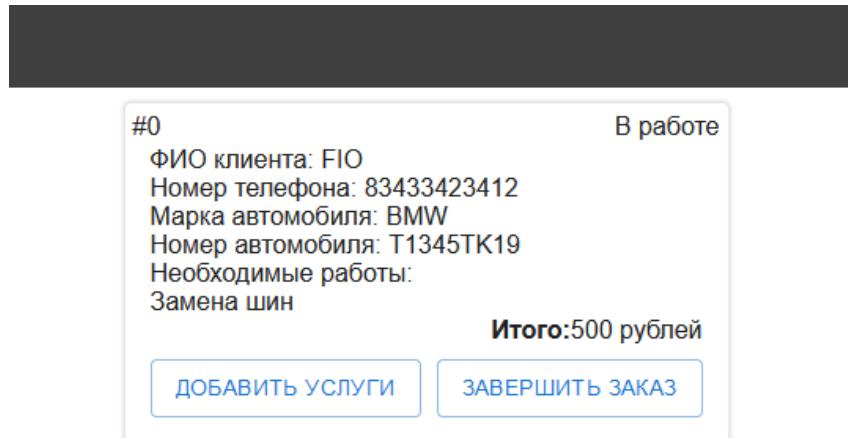


Рисунок 2.18 – Принятие заказа

После принятия заказа, устанавливается статус «в работе», означающий начало работ. Также сотрудник может добавить дополнительные услуги при необходимости. После выполнения работ, нужно завершить заказ, после чего будет указан статус «завершен».

В окне директора отображаются все существующие заказы, а также есть возможность удаления и изменения определенных заказов из списка, пример представлен на рисунках 2.21, 2.22.

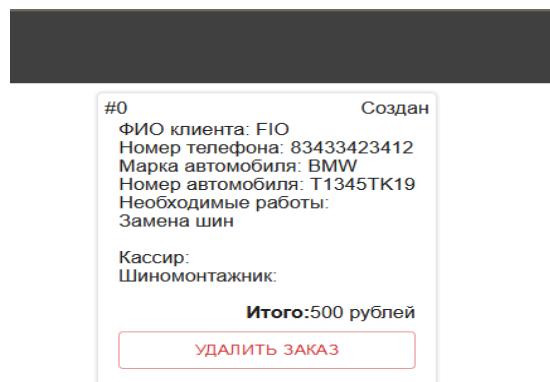


Рисунок 2.19 – Форма с информацией для директора

#0 Создан
ФИО клиента: FIO
Номер телефона: 83433423412
Марка автомобиля: BMW
Номер автомобиля: T1345TK19
Необходимые работы:
Замена шин

Кассир:
Шиномонтажник:

Итого: 500 рублей

УДАЛИТЬ ЗАКАЗ

Имя клиента
FIO

Номер телефона клиента
83433423412

Марка и модель автомобиля
▼

Номер автомобиля
T1345TK19

Необходимые работы
Замена шин

ОБНОВИТЬ ЗАКАЗ

ОТМЕНА

Рисунок 2.20 – Окно редактирования заказа

2.3 Выводы по разделу «Описание разработки веб-приложения приема заказов для ООО «Шинный двор»»

Во время работы по созданию веб-приложения были изучены различные программы и программные комплексы для разработки программных продуктов, выбраны и изучены языки программирования для определенных задач, такие как – ReactJS для разработки визуальной составляющей приложения, PHP+SQL для работы с БД.

Реализован готовый программный продукт, соответствующий необходимым требованиям.

3 Расчёт затрат и оценка экономической эффективности реализации

3.1 Анализ состава и стоимости ресурсов, необходимых для реализации проекта

В таблице 3.1 представлен перечень технических ресурсов, необходимых для реализации проекта.

Таблица 3.1 – Перечень ресурсов, необходимых на этапах создания

Название оборудования	Количество, шт.	Цена, руб.	Нормативный срок
Монитор Philips 223V5LHSB2(1920x1080)	1	13990	7 лет
Процессор INTEL Core i3 9100F	1	9590	4 года
Видеокарта Asus GeForce GTX 1060 Dual OC 3GB	1	16790	5 лет
Оперативная память AMD Radeon R3 8 ГБ	1	2650	7 лет
Блок питания Cougar XTC500(500w)	1	4299	8 лет
Клавиатура Defender Gorda GK-210L	1	990	4 года
Мышь Logitech G102 Prodigy	1	1990	1 год
Жесткий диск Toshiba HDWD110UZSVA с объемом памяти 1ТБ	1	3013	8 лет
Итого		53312	

Используя перечень ресурсов с таблицы можно вычислить годовую норму амортизации по формуле

$$A_{\text{год}} = C_T * H_a \quad (3.1)$$

где C_t – стоимость компьютера;

H_a – норма амортизации.

$$A_{год} = 53312 * 20\% = 10662$$

$$H_a = 1/\text{нормативный срок службы} * 100\% \quad (3.2)$$

$$H_a = 1/5 * 100\% = 20\%$$

Для расчета ежемесячной амортизации на проектирование нужно воспользоваться формулой

$$A_{пр} = A_{год}/12 \quad (3.3)$$

$$A_{пр} = 10662/12 = 889$$

Персонал, который будет разрабатывать проект, их срок работы и рабочее время представлен в таблице 3.2.

Таблица 3.2 – Персонал

Должность	Срок работы	Часы работы
Программист	5 месяцев	3ч в день (100ч всего)

Перечень ПО, которое понадобится на этапе создания проекта. Список ПО представлен в таблице 3.3.

Таблица 3.3 – ПО на этапе создания

Наименование ПО	Цена, руб.
visual studio code	бесплатно
dbForge Studio for Oracle	бесплатно

Продолжение таблицы 3.3

Microsoft Windows 10 Home	13500
---------------------------	-------

Срок полезного использования – 5 мес.

Годовая норма амортизация, согласно формуле (3.1): $100\%/5=20\%$

Ежегодная амортизация, согласно формуле (3.2): $13500*25\%=2670$ рублей.

Ежемесячная амортизация, согласно формуле (3.3): $2670/12=223$ рубля.

Перечень ресурсов, которые понадобятся на этапе внедрения и эксплуатации проекта представлен в таблице 3.4.

Таблица 3.4 – Перечень ресурсов необходимых на этапах внедрения и эксплуатации

Название оборудования	Количество, шт.	Цена, руб.	Нормативный срок
Планшет Lenovo M10 HD 64 ГБ	2	15999	7 лет

Срок полезного использования – 5 мес.

Годовая норма амортизация, согласно формуле (3.1): $100\%/5=20\%$

Ежегодная амортизация, согласно формуле (3.2): $31998*20\%=6400$ рублей.

Ежемесячная амортизация, согласно формуле (3.3): $6400/12=533$ рублей

Список персонала, которые будут обслуживать проект после внедрения, и их заработка платы представлен в таблице 3.5.

Таблица 3.5 – Обслуживающий персонал

Квалифицированный персонал	Заработка платы в месяц, руб.
Инженер-программист	25000

Перечень ПО, которое понадобится на этапе эксплуатации проекта представлен в таблице 3.6.

Таблица 3.6 – ПО, необходимое для эксплуатации

Наименование ПО	Цена, руб.
Браузер	бесплатно

3.2 Расчет проектных затрат

Для того чтобы вычислить затраты на проектирование ИС необходимо вычислить следующие элементы: $K_{зп}$, $K_{ипс}$, $K_{свт}$, $K_{проч}$,

где $K_{зп}$ – затраты на заработную плату проектировщиков;

$K_{ипс}$ – затраты на инструментальные программные средства для проектирования;

$K_{свт}$ – затраты на средства вычислительной техники для проектирования;

$K_{проч}$ – прочие затраты на проектирование.

Для нахождения $K_{зп}$ необходимо вычислить заработную плату всех работников, участвующих в проектировании ИС.

Минимальный оклад труда в Абакане на момент 2022 года составляет 13890 рублей. Разделив минимальный оклад на количество рабочих дней в

месяце можно определить, сколько рублей программист зарабатывает в день. Ниже представлено вычисление.

$$13890/22=631 \text{ рублей.}$$

Заработка плата программиста за месяц с учетом его рабочих дней. Программист работает по пятидневной системе. В месяц получается 20 рабочих дней. Ниже представлено вычисление.

$$631*20= 12620 \text{ рублей.}$$

Теперь известна заработка плата в месяц персонала за проект – можно вычислить переменную $K_{зп}$.

Северный и районный коэффициент составляют $3786+3786$ рублей.

Итого получается $12620+3786+3786=20192$ рублей.

Далее необходимо учесть обязательные отчисления во внебюджетные фонды – 1,302. Ниже представлено вычисление $K_{зп}$.

$$K_{зп} = 20192 * 1,302 = 26290 \text{ рублей.}$$

Вычислим значение переменной $K_{ипс}$, для этого возьмем значение ежемесячной амортизации ПО, и умножим это значение на количество рабочих месяцев. Ниже представлено вычисление переменной $K_{ипс}$.

$$K_{ипс} = 223 * 3 = 669 \text{ рублей.}$$

Вычислим значение переменной $K_{\text{свт}}$, для этого возьмём значение ежемесячной амортизации у оборудования, и умножим это на количество рабочих месяцев. Ниже представлено вычисление $K_{\text{свт}}$.

$$K_{\text{свт}} = 889 * 3 = 2667 \text{ рублей.}$$

Теперь известны почти все переменные, необходимые для вычисления $K_{\text{пр}}$, за исключением $K_{\text{проч}}$. Для ее вычисления необходимо суммировать имеющиеся коэффициенты, ниже представлено значение $K_{\text{пр}}$.

$$K_{\text{пр}} \text{ без } K_{\text{проч}} = 26290 + 669 + 2667 = 29626 \text{ рублей.}$$

Затраты на $K_{\text{проч}}$ определяются суммой $K_{\text{зп}}$, $K_{\text{ипс}}$, $K_{\text{свт}}$ от 1% до 5%. Среди затрат $K_{\text{проч}}$ входят: аренда помещения, коммунальные услуги за рабочее время программиста, электричество и интернет. Для разрабатываемого проекта хватит 5%, в таблице 3.7 выделим все имеющиеся затраты.

Таблица 3.7 – Проектные затраты

Проектные затраты	Стоимость, руб.
$K_{\text{зп}}$	26290
$K_{\text{ипс}}$	669
$K_{\text{свт}}$	2667
$K_{\text{проч}}$	1481

$$K_{\text{пр}} = 26290 + 669 + 2667 + 1481 = 31107 \text{ рублей.}$$

Отразим на диаграмме данные, которые имеет таблица 3.7. Диаграмма представлена на рисунке 3.1.



Рисунок 3.1 – Структура проектных затрат

Согласно данным диаграммы наибольшие проектные затраты составляют $K_{зп} - 85\%$, чуть меньше, $9\% - K_{свт}$, $K_{ипс} - 2\%$, $K_{проч} - 5\%$.

3.3 Расчет капитальных затрат

Расчет капитальных затрат производится по формуле

$$K = K_{пр} + K_{tc} + K_{lc} + K_{по} + K_{ио} + K_{об} + K_{оэ}, \quad (3.4)$$

где $K_{пр}$ – затраты на проектирование ИС;

K_{tc} – затраты на технические средства управления;

K_{lc} – затраты на создание линий связи локальных сетей;

$K_{по}$ – затраты на программные средства;

$K_{ио}$ – затраты на формирование информационной базы;

$K_{об}$ – затраты на обучение персонала;

$K_{оэ}$ – затраты на опытную эксплуатацию.

Для расчета затрат на технические средства управления необходимо взять значение ежемесячной амортизации оборудования в период внедрения и эксплуатации, поэтому $K_{tc} = 533$ рублей.

Затраты на линии связи отсутствуют, т.к. ООО «Шинный двор» имеет свою локальную сеть, поэтому нужды в дополнительных затратах нет. $K_{lc} = 0$ рублей.

Затраты на программные средства K_{po} отсутствуют, так как для внедрения веб-приложения понадобится только бесплатное ПО, а именно браузер. Поэтому $K_{po} = 0$.

K_{io} – затраты на формирование информационной базы. Информационная база будет формироваться в течение 1 рабочего дня. Из этого следует, что его заработка плата в день 631 рублей. С учетом районного и северного коэффициента, а также отчислением во внебюджетные фонды итоговое значение K_{io} будет следующим.

$$(631+189+189) *1,302= 1314 \text{ рублей.}$$

K_{ob} – затраты на обучение персонала равны нулю, так как инженер-программист, а также персонал, который будет работать с информационной системой уже имеет достаточный опыт с веб-приложениями, поэтому дополнительного обучения не требуется.

K_{oo} – затратами на опытную эксплуатацию будет являться заработка инженера-программиста за период эксплуатации – 2 дня. Для расчета заработной платы необходимы данные: заработка плата – 25000 рублей, среднее количество рабочих дней – 22, северный и районный коэффициент по 30 %, отчисления во внебюджетные фонды –30,2%.

$$K_{oo} = (25000 /22 *2) *1.6 *1.302= 4735 \text{ рублей.}$$

Все вычисленные затраты отображены в таблице 3.8.

Таблица 3.8 – Капитальные затраты

Капитальные затраты	Стоимость, руб.
$K_{\text{пр}}$	31107
$K_{\text{тс}}$	533
$K_{\text{лс}}$	0
$K_{\text{по}}$	0
$K_{\text{ио}}$	1314
$K_{\text{об}}$	0
$K_{\text{оэ}}$	4735

Теперь вычислим капитальные затраты согласно формуле (3.4).

$$K = 31107 + 533 + 0 + 0 + 1314 + 0 + 4735 = 37689 \text{ рублей.}$$

Теперь отразим данные из таблицы 3.8 в диаграмме капитальных затрат, которая изображена на рисунке 3.2.



Рисунок 2 –Капитальные затраты

Согласно данным диаграммы наибольшие капитальные затраты составляет показатель $K_{\text{пр}}$, занимает наибольший процент – 83%. $K_{\text{об}}$, $K_{\text{лс}}$, $K_{\text{по}}$ не имеют затрат из-за ненадобности в лишних затратах. Показатель $K_{\text{тс}}$ составил

1% от капитальных затрат, показатель $K_{ио}$ составил 3%, а показатель $K_{оэ}$ составляет 13%.

3.4 Расчет эксплуатационных затрат

Расчет эксплуатационных затрат производится по формуле

$$C = C_{зп} + C_{ао} + C_{то} + C_{гс} + C_{ни} + C_{проч}, \quad (3.5)$$

где $C_{зп}$ – зарплата персонала, работающего с веб-приложением;

$C_{ао}$ – амортизационные отчисления;

$C_{то}$ – затраты на техническое обслуживание;

$C_{гс}$ – затраты на использование глобальных сетей;

$C_{ни}$ – затраты на носители информации;

$C_{проч}$ – прочие затраты.

Из персонала ООО «Шинный двор» обслуживать и следить за работой ИС будет инженер-программист. З/п работника в месяц 25000 рублей.

$C_{зп}$ – зарплата персонала, работающего с веб-приложением. Для расчета заработной платы необходимы данные: заработка плата – 25000 рублей, среднее количество рабочих дней – 22, северный и районный коэффициент по 30 %, отчисления во внебюджетные фонды – 30,2%.

$$C_{зп} = (25000 / 22) * 1.6 * 1.302 = 2367 \text{ рублей.}$$

Переменная $C_{ао}$ будет равняться ежегодной амортизации компьютера, за которым инженер-программист будет работать. Предполагается, что работа будет занимать 1 день в году, амортизация оборудования не учитывается, так как имеет незначительные затраты менее 30 рублей. $C_{ао} = 0$.

$C_{то}$ будет являться заработка инженера-программиста на техническое обслуживание проекта в течение 1 года. Для расчета необходимы данные: заработка платы системного администратора, который будет обслуживать ИС, 24000 рублей, поделим это число на среднее количество рабочих дней – 22, количество рабочих часов в день – 8, северный и районный коэффициент по 30 %, отчисления во внебюджетные фонды – 30,2%. Инженер-программист периодически должен проводить технические работы, их длительность будет составлять примерно 6 часов за год.

$$24000/22 = 1136 \text{ рублей.}$$

Теперь вычислим заработную плату в час, для этого 1136 надо разделить на среднее количество рабочих часов (8).

$$1136 / 8 = 142 \text{ рубля.}$$

Инженер-программист периодически должен проводить технические работы, их длительность будет составлять примерно 6 часов за год.

$$142 * 6 = 852 \text{ рубля.}$$

$$C_{то} = 852 * 1.6 * 1.302 = 1775 \text{ рублей.}$$

$C_{лс}$ – затраты на использование глобальных сетей равны 0, ООО «Шинный двор» имеет свою собственную сеть, поэтому в дополнительных сетях нужды нет.

$C_{ни}$ – затраты на носители информации.

В данном случае этим носителем является виртуальный хостинг с оплатой 1300 рублей в год.

Теперь вычислим эксплуатационные затраты без $C_{\text{проч.}}$.

$$C = 2367 + 8598,6 + 1775 + 0 + 1300 = 14040 \text{ рублей.}$$

$C_{\text{проч}}$ будет равняться 5% или 702 рубля, этой суммы будет достаточно для прочих затрат. Теперь вычислим полностью эксплуатационные затраты – $14040 + 702 = 14742$ рублей.

Отразим все затраты в таблице 3.9.

Таблица 3.9 – Эксплуатационные затраты

Эксплуатационные затраты	Стоимость, руб.
$C_{\text{зп}}$	2367
$C_{\text{ао}}$	8598
$C_{\text{то}}$	1775
$C_{\text{лс}}$	0
$C_{\text{ни}}$	1300
$C_{\text{проч}}$	702

Также отразим все указанные затраты в виде диаграммы, изображенной на рисунке 3.3.

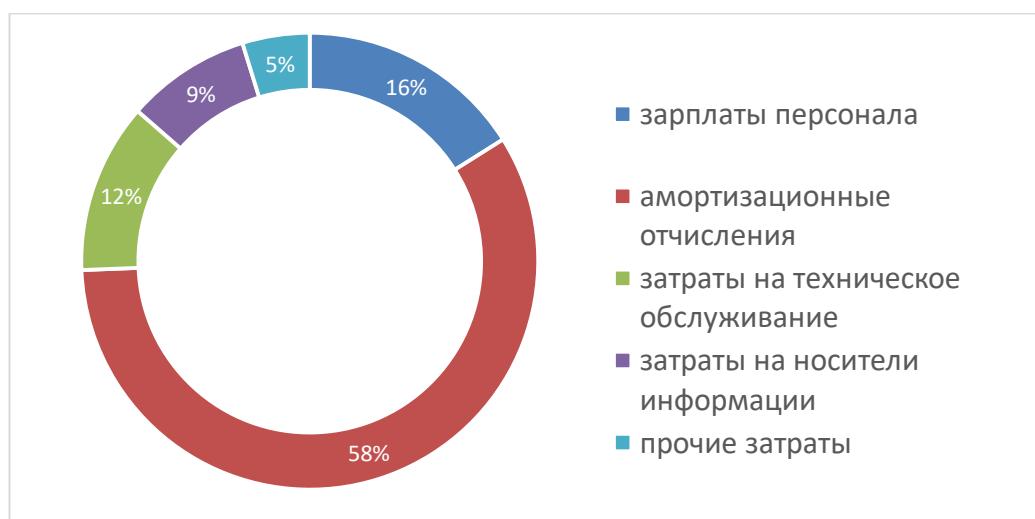


Рисунок 3.3 – Эксплуатационные затраты

Согласно данным из диаграммы показатель C_{ao} занимает 58% эксплуатационных затрат, затем показатель $C_{зп}$ с его 16%, чуть меньше показатель $C_{то}$ – 12%, $C_{проч}$ занимает незначительное количество затрат – 5%, показатель $C_{ни}$ – 9%, $C_{лс}$ не несет эксплуатационных затрат, из-за их ненадобности.

3.5 Расчет совокупной стоимости владения системой

Рассчитаем совокупную стоимость владения веб-приложения по методике ТСО. Для нахождения ТСО необходимо воспользоваться формулой

$$TCO = DE + IC1 + IC2 \quad (3.6)$$

Прямые расходы рассчитываются по формуле

$$DE = DE1 + DE2 + DE3 + DE4 + DE5 + DE6 + DE7 + DE8, \quad (3.7)$$

где $DE1$ – капитальные затраты, они были вычислены ранее и равны 37689 рублей.

$DE2$ – расходы на управление ИС, они были вычислены ранее – $C_{зп}$ 2367 рублей.

$DE3$ – расходы на техническую поддержку АО и ПО, они были вычислены ранее, в виде двух разных переменных ($C_{то}$ и C_{ao}), суммируем их. $1775 + 8598 = 10374$ рублей.

$DE4$ – расходы на разработку прикладного ПО внутренними силами, равны нулю.

$DE5$ – расходы на аутсорсинг равны нулю.

$DE6$ – командировочные расходы равны нулю, поскольку выездов в другие города не требуется.

DE7 – расходы на услуги интернет провайдера. Для работы системы потребуется интернет. Выберем провайдера Ростелеком, выберем тариф стоимостью 350 руб./мес., поскольку проект разрабатывался в течение 5 месяцев, поэтому $DE7 = 350 * 5 = 1750$ рублей. Так же нужен хостинг, стоимостью 1300 рублей.

DE8 – другие группы расходов. Все оставшиеся эксплуатационные расходы, которые были не учтены, определим в 3% т совокупной стоимости, а именно 1335 рублей.

Имея все переменные DE1-DE8, можно вычислить совокупную стоимость владения системой.

$$DE = 37689 + 2367 + 10374 + 0 + 0 + 0 + 1750 + 1300 + 1335 = 54815$$

рублей.

Данный проект подвержен некоторому числу рисков, часть из них может оказать большое влияние на выполнение проекта. Определим риски проекта – риск того, что персоналу будет неудобно использовать веб приложение, так же существует риск несоответствия разработки поставленным условиям заказчика.

На основе представленных рисков определим косвенные затраты. При риске, трудности в освоении персоналом, придется доработать проект, чтобы исправить неудобные модули для персонала. Для этого потребуется 1 рабочий день. Ранее заработка плата в день у разработчика была высчитана – 631 рублей. Поэтому затраты будут равны $(631+189+189) * 1,302 = 1314$ рублей.

Другой риск, несоответствия разработки поставленным условиям заказчика. В таком случае разработчику нужно будет переделать веб-приложение. Ранее была вычислена его заработка плата в день – 631 рубль. На доработку ему потребуется три рабочих дня, поэтому расходы на заработную плату специалиста составят $(1893+568+568) * 1,302 = 3944$ рублей.

Теперь обобщим полученные результаты. Итого, косвенные расходы первой группы IC_1 равны 1314 рублей, косвенные расходы второй группы IC_2 равны 3944 рублей.

$$TCO = 54815 + 1314 + 3944 = 60073 \text{ рублей.}$$

3.6 Оценка рисков реализации проекта

В начале проекта невозможно знать все, что случится в процессе его реализации, поэтому все инвестиции сопряжены с риском. Попытаемся определить и измерить риски, свойственные ИТ-проектам, а также неопределенности, возникающие непосредственно на этапе проведения оценки. Оценка рисков представлена в таблице 3.10.

Таблица 3.10 – Риски проекта

Пункт	Группы рисков	Перечень рисков проектов	Вероятность возникновения риска	Степень воздействия риска на проект	Возможные решения риска
1	Риск соответствия	Трудности в освоении персоналом	Средняя	Высокая	Консультация разработчика с персоналом, для уточнения деталей
2	Риск соответствия	несоответствия разработки поставленным условиям заказчика	Низкая	Средняя	Консультация разработчика с заказчиком

Теперь начнем разбирать риски проекта. Начнем с риска – трудности в освоении персоналом. Данный риск имеет среднюю вероятность возникновения, так как масштаб проекта невелик, следовательно, не должно возникнуть ситуаций, при которых у персонала возникнут трудности с освоением к веб-приложению. Однако степень воздействия данного риска высокая, так как пользователи не смогут воспользоваться всем функционалом системы, если риск произойдет. Для полного предотвращения данного риска необходимо провести консультации с персоналом, который будет работать с данной ИС.

Теперь разберем последний риск – несоответствия разработки поставленным условиям заказчика. Данный риск имеет низкий шанс возникновения, так как, заказчик и персонал, который будет работать с системой, имеет большой опыт работы, как следствие, подобные системы он уже видел и, скорее всего, с ними работал, поэтому ему не составит труда разобраться с функционалом. Однако, если все же случится так, что система действительно не понравится заказчику, то это будет критично для проекта, так как заказчику и персоналу будет тяжело работать с системой, что потребует её доработки. Для предотвращения такого развития событий можно принять следующие меры: предоставить заказчику описание функционала системы заранее и посоветоваться с ним, чтобы решить, как ему будет удобнее работать с системой.

3.7 Анализ рынка продуктов-аналогов. Установление стоимости программного продукта

Сначала определим группу товаров, в которую входит разработанный программный продукт. Программный продукт входит в следующую группу товаров – создание веб-приложения.

Проведем описание основных потребительских характеристик разработанного продукта в сравнении с найденными аналогами:

Для определения стоимости программного продукта есть несколько методов. Сравнительный метод, каждую функцию необходимо будет оценивать от 1 балла до 10 баллов. Сравнительный метод представлен в таблице 3.11.

Таблица 3.11 – Сравнение с конкурентами

Характеристики и оценки	Help Desk	РемОнлайн	Веб-приложение для автоматизации процесса выполнения заказа
Выбор дизайна	Выбрать шаблон либо прислать желаемый вариант	Разработка нового дизайна с учетом приложенного ТЗ	Будет создан шаблон, который будет реализован в соответствии с рекомендациями специалиста
Оценка	10	10	10
Реализация пожеланий клиента	Предоставляется за дополнительную оплату	Предоставляется за дополнительную оплату	Входит в ИС
Оценка	10	10	10
Функционал программы	Имеет очень большой функционал, под разные нужды	Предоставляется за дополнительную оплату	Имеет ограниченный функционал
Оценка	10	8	4
Итого баллов	30	28	24
Стоимость	30000 рублей в месяц	20000 в месяц	

. Можно было купить готовые разработки, но капитальные затраты равны $K = 28714$ рублей, а в разработанных проектах такая сумма в месяц.

Теперь определим стоимость веб-приложения, используя затратный метод.

Капитальные затраты были вычислены ранее – $K = 28714$ рублей.

Определим норму прибыли – 15% от капитальных затрат, потому что охват аудитории для программного продукта будет состоять только из работников ООО «Шинный двор». Это будет равняться $(28714 * 15) / 100 = 4307$ рублей.

Итоговый расчет – $28714 + 4307 = 33021$ рублей. Стоимость проекта по затратному методу станет равна 33021 рублей.

Проект выполняется на средства заказчика ООО «Шинный двор».

3.8 Экономическая эффективность реализации проекта

Косвенный доход от реализации проекта связан с уменьшением затрат при обслуживании клиента, эффективность проекта рассчитывается методом автоматизации.

Сначала определим коэффициенты ЗТР, для этого возьмем за основу изначальный вариант (базовый вариант), а также нашу ИС (разрабатываемый вариант). Далее вычислим ЗТР. Оценка и вычисленный коэффициент ЗТР определены в таблице 3.12.

Таблица 3.12 – Определение коэффициентов ЗТР

Коэффициенты	Базовый вариант	Разрабатываемый вариант
K_a	1	2
K_p	1	1
K_c	1	2
K_m	1	1
K_o	1	2
K_{sh}	2	2
ЗТР	3	8

Теперь определим показатели качества, весовые коэффициенты, а затем проведем оценку обоих проектов. Затем определим комплексный показатель качества (K_t). Определенные показатели K_t изображены в таблице 3.13.

Таблица 3.13 – Определение показателей K_t

Показатель качества	Весовой коэффициент, b_1	Оценка, X_i	
		Разрабатываемый проект	Базовый проект
Удобство работы (пользовательский)	0,4	4	1
Надежность (защита данных)	0,2	4	1
Функциональные возможности	0,1	2	1
Временная экономичность	0,3	4	2
Время обучения персонала	0	1	1
Комплексный показатель качества		3,8	1,3

$K_t = 3,8/1,3 = 2,9$ (показатель получился больше 1, значит, наша система будет отличной).

При базовом варианте время на составление заказ наряда и передачи его в сервис, составляет 10 минут. В среднем за рабочий день Механик обслуживает клиентов 6 часов, на 1 заказ уходит 30 минут, с учетом выше указанного времени. Средняя стоимость выполнения заказа 700 рублей. Таким образом в день, можно выполнить 12 заказов.

$$C_i \text{ баз} = 12 * 700 = 8400 \text{ рублей.}$$

При внедрении веб-приложения по время на составление заказ наряда и передачи его в сервис, составляет 2 минуты. В среднем за рабочий день

Механик обслуживает клиентов 6 часов, на 1 заказ уходит 20 минут. Таким образом в день, можно выполнить 18 заказов.

$$C_i \text{пр} = 18 * 700 = 12600 \text{ рублей.}$$

Для определения, за какой промежуток времени, проектируемая информационная система окупиться, воспользуемся формулой

$$(C_i \text{пр} - C_i \text{баз}) * X - K, \quad (3.8)$$

где Э – экономический эффект от использования разрабатываемой системы;

Х – временной промежуток;

К – капитальные затраты.

$$(12600 - 8400) * X - 28714$$

$$X = 28714 / 4200 = 7 \text{ дней.}$$

Таким образом, проект должен окупиться за 7 рабочих дней, но не каждый день количество клиентов будет максимальным. Если учесть, что во время внедрения системы сезон «смены шин» уже прошел, проект окупит себя через месяц эксплуатации.

3.9 Выводы по разделу «Расчёт затрат и оценка экономической эффективности реализации проекта»

В данном разделе была произведена оценка экономической эффективности разработки веб-приложения по автоматизации процесса выполнения заказа на предприятии ООО «Шинный двор». Рассчитаны капитальные и эксплуатационные затраты, которые входят в группу прямых

затрат. Проведён сравнительный анализ, который показал, то проект является экономически более выгодным, чем аналогичные предложения.

Капитальные затраты составили $K=37689$ рублей.

Эксплуатационные затраты составили $C=14742$ рублей.

Прямые затраты составили $DE= 54815$ рублей.

Совокупная стоимость владения (TCO) = 60083 рублей.

Кроме того, был проведен анализ рисков, которые могут возникнуть при разработке проекта, а также рассмотрены способы снижения вероятности их возникновения и влияния на процесс разработки.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы «Разработка автоматизированной системы приема заказов для ООО «Шинный двор»», была выявлена проблемная часть – взаимодействие работников. Для решения проблемы было разработано web-приложение, что позволило автоматизировать работу кассира с шиномонтажом, и взаимодействием с клиентом.

Выявлены требования заказчика к разрабатываемому программному продукту. Были созданы диаграммы в нотациях IDEF0, IDEF3.

Затем выбраны средства, язык программирования и архитектура для разработки продукта, а также произведено сравнение и выбор среды разработки.

Выполнена разработка автоматизированной системы приема заказов ООО «Шинный двор».

Web-приложение по автоматизации процесса приема заказов, разработано согласно предъявленным требованиям. Web-приложение создано при помощи языка программирования JavaScript и PHP в среде разработки Visual Studio Code.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ООО «Шинный двор» : о магазине : официальный сайт. – 2021. – URL: <https://шинныйдвор.рф/company> (дата обращения: 15.03.2022).
2. Автоматизация бизнес-процессов как необходимое условие эффективности компании : Бизнес и автоматизация : официальный сайт. – 2020. – URL: <https://www.kp.ru/guide/avtomatizatsija-biznesa.html> (дата обращения: 16.03.2022).
3. Методология IDEF0 : Знакомство с нотацией IDEF0 : официальный сайт. – 2020. – URL: <https://habr.com/ru/company/trinion/blog/322832/> (дата обращения: 17.03.2022).
4. Система учета и управления заказами OkDesk : Возможности системы : официальный сайт. – 2021. – URL: <https://okdesk.ru/ticketing> (дата обращения: 21.03.2022).
5. Программа для шиномонтажа : Программа для учета шиномонтажа РемОнлайн : официальный сайт. – 2021. – URL: <https://remonline.ru/tire-shop/> (дата обращения: 21.03.2022).
6. Методология IDEF3 - Моделирование бизнес-процессов : Методология IDEF3 : официальный сайт. – 2021. – URL: https://studme.org/87186/ekonomika/metodologiya_idef3. (дата обращения 29.03.2022).
7. Первый бит : Языки программирования для создания сайтов: официальный сайт. – 2021. – URL: <https://studiobit.ru/blog/sozdanie-web-saytov/> (дата обращения: 01.04.2022).
8. Консультант Плюс : НАЛОГОВЫЙ КОДЕКС РОССИЙСКОЙ ФЕДЕРАЦИИ : официальный сайт. – 2022. – URL: http://www.consultant.ru/document/cons_doc_LAW_19671/ (дата обращения: 02.04.2022).

9. Роструд : Государственная инспекция труда в Республике Хакасия : официальный сайт. – 2022. – URL:
<https://git19.rostrud.gov.ru/news/928937.html#:~:text=Новый%20МРОТ%20с%201%20января,он%20составляет%202020%20467%> (дата обращения: 03.04.2022).

10. СТУ 7.5–07–2021 СТАНДАРТ УНИВЕРСИТЕТА : Система менеджмента качества : Общие требования к построению, изложению и оформлению документов учебной деятельности : сайт / Сибирский Федеральный Университет. – Красноярск : СФУ, 2021 – . – URL:
<https://about.sfu-kras.ru/docs/8127/pdf/808588> (дата обращения: 04.04.2022).

Выпускная квалификационная работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной
литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 10 наименований.

Один экземпляр сдан на кафедру.

«____» _____ 2022 г.

_____ Жалковский Александр Вячеславович

подпись

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой

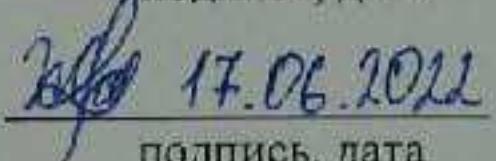
Е. Н. Скуратенко
подпись
«17 » 06 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

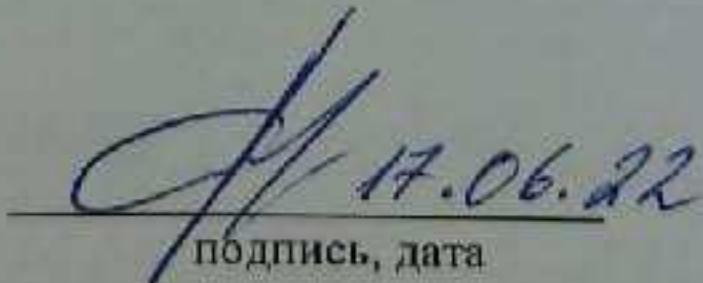
Разработка автоматизированной системы приема заказов для ООО
«Шинный двор»

Руководитель  доцент, канд. физ.-мат. наук А. Н. Таскин
подпись, дата

Выпускник  А. В. Жалковский
подпись, дата

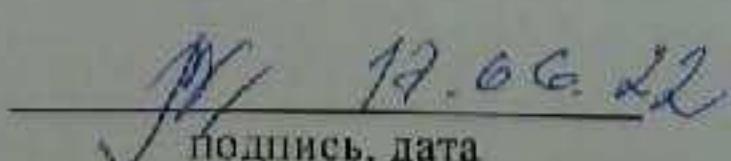
Консультанты
по разделам:

Экономический


подпись, дата

Е. Н. Скуратенко

Нормоконтролер


подпись, дата

В. И. Кокова

Абакан 2022