

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА СЕРВИСНОГО  
ОБСЛУЖИВАНИЯ АВТОМОБИЛЕЙ**

Выпускная квалификационная работа  
обучающейся по направлению подготовки 02.03.02 Фундаментальная  
информатика и информационные технологии  
заочной формы обучения, группы 07001360  
Таран Ангелины Александровны

Научный руководитель  
к.т.н., доцент  
Румбешт В.В.

БЕЛГОРОД 2017

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ.....	5
1.1 Особенности организации сервисного обслуживания.....	5
1.2 Обзор и анализ существующих систем.....	9
1.3 Требования к автоматизированной системе.....	14
ГЛАВА 2. ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ.....	17
РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	17
2.1 Обоснование выбора средств разработки.....	17
2.2 Выбор системы управления базы данных.....	21
2.3 Проектирование структуры БД.....	25
2.4 Проектирование структуры web-приложения.....	29
ГЛАВА 3. ИСПЫТАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	35
3.1 Программа и методика испытаний.....	35
3.2 Результаты испытаний.....	38
3.2.1 Тестирования удобства использования.....	38
3.2.2 Тестирование функциональных возможностей.....	39
3.2.3 Тестирования совместимости;.....	47
3.2.4 Тестирования баз данных.....	47
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	50
ПРИЛОЖЕНИЕ.....	52

## ВВЕДЕНИЕ

В наше время достаточно много автомобилистов, а также автосервисов. Этот вид деятельности востребован. Существует задача учета сервисного обслуживания автомобилей, которая частично автоматизирована. Автоматизированная система предназначена для более удобной работы с данными. Задача важна и актуальна.

В автосервисе работает менеджер, обслуживает клиентов, и он должен обрабатывать большое количество информации, поэтому актуальна система - для организации хранения информации,

Менеджер сервисного центра перегружен работой, поэтому чтобы упростить ему задачу эту деятельность можно автоматизировать. Автоматизация сервисного обслуживания автомобилей позволит избавиться от многих рутинных процессов, что обеспечит наиболее быстрое, полное и качественное обслуживание клиентов, увеличится достоверность и скорость обработки информации.

Целью работы является создание автоматизированной системы сервисного обслуживания автомобилей. Данная автоматизированная система представляется как web-приложение, обрабатывающее информацию о клиентах, услугах, заказах.

Для достижения поставленной цели необходимо решение следующих задач:

- 1) обзор и анализ деятельности по учету сервисного обслуживания автомобилей;
- 2) обзор и анализ существующих автоматизированных систем в этой области;
- 3) формулировка требований;

4) проектирование и программная реализация автоматизированной системы;

5) проведение испытаний.

Структура дипломной работы. Дипломная работа состоит из введения, трех глав, заключения, списка литературы и приложения.

Во введении обосновывается актуальность разработки системы, ее цели и задачи.

В первой части выполняется постановка задачи на разработку.

Во второй части представлено как производилась разработка системы, обзор средств разработки и выбор СУБД, а также проектирование структуры базы данных.

В третьей части описано испытание автоматизированной системы, описан алгоритм выполнения проверки, также выполняется проверка работоспособности автоматизированной системы.

В заключении приводится общий вывод о проделанной работе.

Выпускная квалификационная работа содержит 51 страницу, 34 рисунка и приложение на 19 страницах.

## ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ

### 1.1 Особенности организации сервисного обслуживания

Автосервис - это комплекс организационно-технических мероприятий, документов и производств, обеспечивающих полезность, реализацию, работоспособность и сохранность автотранспортных средств, в соответствии с требованиями их владельцев и общества. Сервисный центр охватывает наибольший период жизненного цикла автотранспортных средств от момента выхода машин с завода - изготовителя до их окончательной утилизации. Основным видом деятельности сервисного центра является реализация ремонта, диагностики автомобилей [3].

В сервисном центре происходит техническое обслуживание (поддержание) и ремонт (восстановление) работоспособности автотранспортных средств, преимущественно легковых автомобилей на станциях технического обслуживания (СТО). Вместе с тем сфера оказания услуг потребителям, связанным с эксплуатацией автотранспортных средств значительно шире. Возрастание роли автосервиса в последние годы объясняется чисто экономическими соображениями. Значительная часть автомобилей и их владельцев участвуют в производственном процессе, в котором каждый час труда характеризуется либо величиной заработной платы, либо величиной прибыли [4].

Фирма оказывает услуги ремонта и диагностики автомобилей.

В автосервисе работают мастера, которые занимаются ремонтом, менеджеры, которые занимаются приемом заказов.

Менеджер сервисного центра — это представитель, который должен максимально качественно работать с клиентом, чтобы полностью удовлетворить все его потребности. В процессе ремонта автомобиля

менеджер обязан полностью контролировать все этапы, чтобы у клиента не возникало никаких проблем.

Менеджер - это также руководитель подразделения автосервиса, составной части крупного объединения. Несет ответственность за работу всего подразделения автосервиса. Подчиняется директору. Сервис - менеджеру подчиняются все нижестоящие сотрудники автосервиса.

Работа менеджера — сложное ремесло, особенно если менеджер планирует быть успешным. Помогает клиенту принять решение о ремонте, диагностике. Объясняет преимущества в сервисном центре, предоставляя клиенту информацию обо всех предлагаемых услугах, диагностиках, ремонте. Сообщает каждому потенциальному клиенту о возможностях услуг и приглашает воспользоваться услугами автосервиса.

Выполняет менеджер такие функции: работа с клиентами, подбор диагностики либо ремонта автомобиля, организация принятия заказа от клиента, отслеживание состояния заказа, составление отчета о заказах и оплатах этого заказа.

Отвечает на телефонные звонки, даёт полную исчерпывающую информацию клиентам о ценах, видах предлагаемых услуг.

Менеджер может ознакомиться с перечнем предоставляемых услуг, прейскурантом цен на виды ремонта, диагностик, и наличием на складе запасных частей и расходных материалов.

Может получить распечатку платежного документа, нужного заказа.

Получает сведения о состоянии своего заказа.

Изменяет прейскурант цен на услуги, корректирует записи о клиентах.

Менеджер работает как в офисе, так и на выезде.

Организация рабочего места - полное и комплектное оснащение и планировка рабочего места таким образом, чтобы улучшить условия труда менеджера с целью повышения эффективности работы.

Совершение заказа обычно проходит в три этапа. Первый этап – первое посещение автосервиса клиентом, либо вызов менеджера.

Так, одни приходят, чтобы получить необходимую услугу, другие приходят узнать, не предоставляются ли новые виды услуг. Клиент, как правило, осознает, зачем он направляется в автосервис. У него имеется вполне определенная установка на получение той или иной услуги.

Чем конкретнее представление об услуге, тем быстрее будет оформлен и сделан заказ. Расплывчатость представления о заказе требует от клиента большей активности для выбора услуги, из имеющихся в ассортименте.

Второй этап – принятие решения сделать заказ или отказаться от него.

Окончательное решение обычно принимается после повторного (иногда многократного) тщательного сопоставления клиентом "представления об услуге" с представлением обладания им.

На принятие решения определенное влияние оказывает целый ряд дополнительных факторов:

- соответствие услуги цене: невысокая цена услуги делает заказ выгодным, но вызывает настороженность и недоверие в отношении качественного выполнения; высокая цена обычно служит показателем добротности услуги, но она затрудняет принятие решения;

- особенности личности заказчика (бережливость, расточительность, доверчивость, внушаемость и другие);

- настроение клиента, которое во многом определяется атмосферой взаимоотношения в автосервисе. Менеджер имеет в виду, что конфликтная ситуация, несоответствие уровня обслуживания и ожиданиям клиентов настраивают их против совершения заказа;

- четкая организация работы автосервиса, деловитость и компетентность работников, их внимательное, заинтересованное отношение к посетителям.

Естественно, все указанные факторы воздействуют на решение клиента сделать заказ одновременно, а не разрозненно.

Третий этап – совершение заказа или отказ от него. Если клиент сделал заказ, после он будет оценивать качество полученной услуги, как она соответствует требованиям.

Если сделанный заказ полностью удовлетворяет клиента, у него возникает положительное отношение к автосервису, желание стать постоянным клиентом.

Если же заказчик разочаровался в полученной услуге, он, по всей видимости, не будет обращаться в данный автосервис [5].

После выбора услуги оформляется заказ, подготавливается и отправляется запрос механику, предоставляется товарный чек.

Менеджер оформляет заказ, в котором указываются:

- номер заказа;
- статус;
- наименование услуги;
- информация о клиенте;
- цена работ;
- дата.

В заказе имеются такие статусы как:

- 1) новый (заказ, который только поступил);
- 2) выполняется (заказ отправленный механику);
- 3) выполнен (завершенный заказ);
- 4) просрочен (за заказ не была произведена оплата);
- 5) отменен (отказ от заказа).

Менеджер просматривает, и при необходимости корректирует данные о поступлении новых заказов и выполнении уже существующих заказов. Эта информация служит в качестве справочного пособия, и обновляется по мере поступления/выполнения заказов. Также в этой системе хранятся сведения об



оказываемых услугах и расценках на них.

Очевидно, что деятельность менеджера может быть автоматизирована. Это поможет ему легче проводить свою работу.

Автоматизация приведет к следующему эффекту:

- организуется рабочее место;
- оперативное предоставление информации;
- отслеживание информации.

Поможет создание своеобразного органайзера, то есть в нем будет предварительно собрана и обработана информация.

## **1.2 Обзор и анализ существующих систем**

На сегодняшний день существует достаточно автоматизированных систем учета заказов услуг, но в большинстве случаев автоматизированная система спроектирована под конкретный автосервис, имеет ограниченные возможности или входит в состав большого комплекта программ.

Система должна обладать такими критериями:

- 1) удобный интерфейс;
- 2) низкая стоимость;
- 3) эффективная;
- 4) мобильная.

Проводится обзор и анализ существующих разработок на предмет удовлетворения необходимым критериям.

1 система «Альфа-Авто: Автосалон + Автосервис»

Описание: «Альфа-Авто: Автосалон + Автосервис» предназначен для комплексной автоматизации учета на предприятиях автобизнеса (магазинов оптовой и розничной продажи запчастей, автосервисов и станций технического обслуживания автомобилей).

С помощью «Альфа-Авто: Автосалон + Автосервис» можно:

- оформлять заказы;

- оказывать услуги по ремонту и обслуживанию автомобилей;
- организовать работу склада;
- учитывать оплаты и отслеживать состояние взаиморасчетов с покупателями и поставщиками.

Имеется детейлинг центр, в котором можно получить такие услуги как:

- мойка легковых автомобилей;
- химчистка;
- полировка;
- предпродажная подготовка;
- защитные покрытия для стекол и кузова.

Фирма предлагает охранные системы:

- автосигнализации;
- иммобилайзеры;
- секретки;
- механические противоугонные системы;
- противоугонная маркировка стекол и зеркал;
- GPS - трекер.

В Альфа-Авто можно поставить системы комфорта:

- автономные подогреватели двигателя и салона;
- системы безопасности парковки;
- улучшение шумоизоляции;
- доводчики дверей и крышки багажника;
- видеорегистраторы.

Пользователи программы имеют возможность быстро формировать необходимые документы. Руководство может оперативно получать и использовать данные о различных аспектах деятельности компании.

Система предоставляет информацию, необходимую для принятия управленческих решений. Интерфейс Альфа-Авто представлен на рис. 1.1. Цена системы: 51 000 руб.

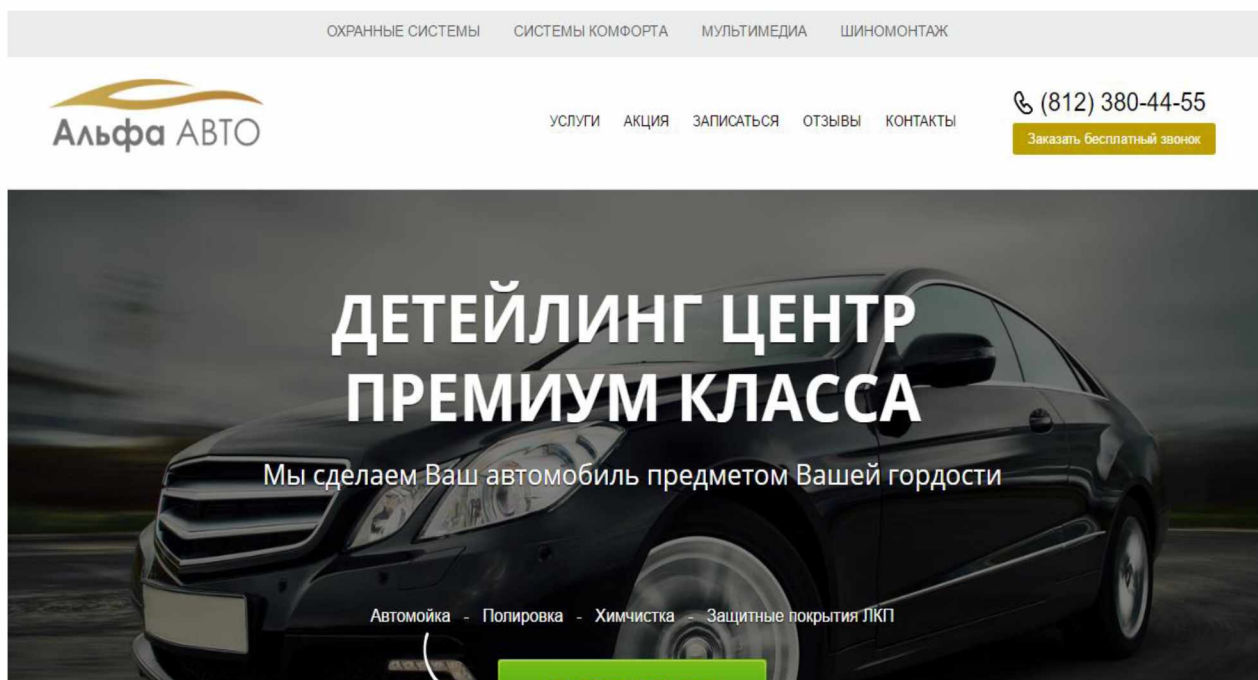


Рис. 1.1. Альфа-Авто

2 система "АВТОСЕРВИС".

Описание: Программа АВТОСЕРВИС предназначена для автоматизации деятельности автосервисов, занимающихся ремонтом и диагностикой, в части, касающейся ведения учета, предоставления услуг.

В фирме АВТОСЕРВИС можно получить такие услуги:

- автоэлектрика;
- замена автостекол;
- полировка;
- рихтовочные работы;
- покрасочные работы; подбор автоэмали.

Можно записать на услугу онлайн, необходимо только указать требуемую информацию. Интерфейс системы представлен на рис. 1.2.

Цена системы: 3 000 руб.

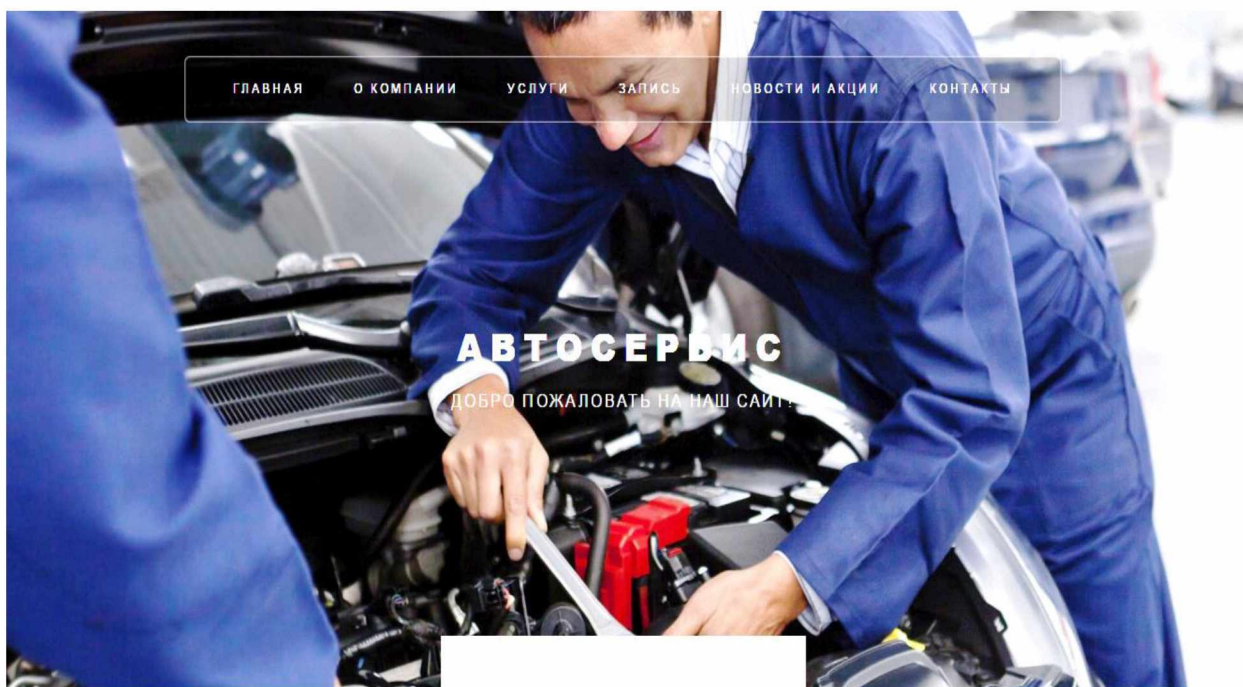


Рис. 1.2. АВТОСЕРВИС

3 система "Лаборатория 5" является единой информационной средой для всех сотрудников компании. Это обеспечивает максимально быстрое и эффективное взаимодействие между ними.

При помощи системы разделения доступа можно организовать автоматизированные рабочие места для:

- руководства компании;
- рецепшен, операторов колл-центра;
- менеджеров;
- работников автопарка;
- бухгалтерия и кадры.

Поддерживается "много-филиальный" режим работы. Возможна удаленная работа с программой через интернет.

"Лаборатория 5" автоматически формирует и печатает полный комплект документов по заказ - наряду.

В результате:

- время обслуживания клиента сокращается не менее чем в 5 раз;
- многократно сокращается объем рутинной работы;

- исключаются ошибки;
- появляется возможность использования сотрудников не высокой квалификации.

Программа помнит все: информацию о клиентах, об автомобилях и их текущем состоянии, все заказ - наряды и выполненные работы, все контакты с клиентами, все операции с деньгами.

Программа напоминает о приближении всех критических сроков: ТО, страховки и пр.

В программе ведется учет всех операций с деньгами.

Вся информация находится в программе, а сотрудники работают по простым и четко описанным правилам.

В программе есть гибко настраиваемая система разделения доступа сотрудников к различным функциям программы, отчетам. Каждый сотрудник имеет доступ только к тем операциям и функциям, которые ему разрешены. Автоматически сохраняются информация о том, кто из сотрудников и что делал в программе.

Имеются средства планирования и контроля исполнения заданий сотрудников. Интерфейс системы представлен на рис. 1.3.

Цена системы: 63 000 руб [7].

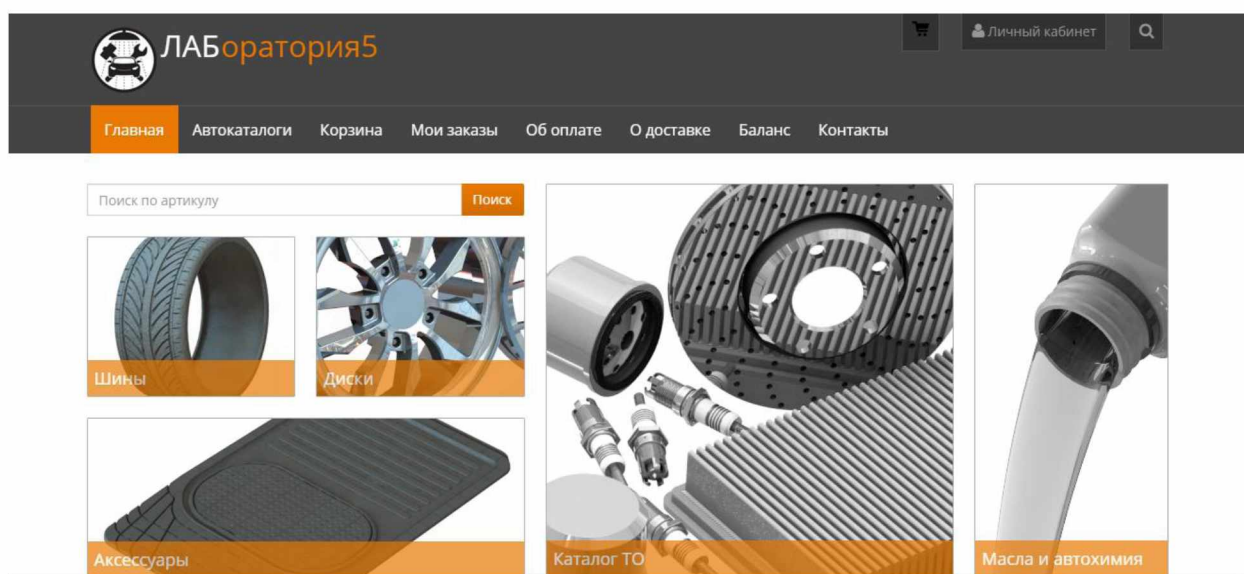


Рис. 1.3. Лаборатория 5

Подводя итоги данного обзора можно составить сводную таблицу свойств рассмотренных систем (таблица 1.1)

Таблица 1.1

Сводная таблица свойств рассмотренных систем

Система / Критерии	Удобный интерфейс	Стоимость	Эффективность	Мобильность
Альфа-Авто: Автосалон + Автосервис	+	-	-	+
АВТОСЕРВИС	-	+	-	+
Лаборатория 5	-	-	+	+

Из обзора существующих разработок следует, что ни одна из рассмотренных систем не удовлетворяет всему комплексу требуемых свойств в целом, поэтому возникает необходимость разработка своей собственной системы.

### 1.3 Требования к автоматизированной системе

Автоматизированная система предназначена для повышения эффективности работы менеджера автосервиса.

Функциональные характеристики: система предназначена для обработки данных о заказах, которые предоставляет фирма, расценках на оказываемые услуги, о клиентах и совершённых сделках.

Входные данные:

- сведения о клиентах;
- сведения об услугах;
- сведения о заказах.

Выходные документы:

- товарный чек.

Требования к надежности это обеспечение устойчивого функционирования системы.

Эксплуатироваться система должна на рабочем месте менеджера (в офисе на компьютере), либо на выездном заказе (телефоне или планшете).

Требования к информационной и программной совместимости. Должна быть реализована для работы интернет, в веб-интерфейс. Так как менеджер принимает заказы не только в офисе, но и на выезде. Ему необходима система для выхода через браузер на месте, чтобы выполнить заказ [2].

Система должна обеспечивать следующие функции:

1) Ввод, вывод, редактирование, хранение, удаление информации о клиентах:

- ФИО клиента;
- email;
- телефон;

2) Ввод, вывод, редактирование, хранение, удаление информации об услугах:

- наименование услуги;
- стоимость;

3) Ввод, вывод, редактирование, хранение, печать информации о заказах:

- номер заказа;
- дата начало/окончание работы (добавлен);
- сумма;
- услуга;
- статус;
- клиент;

4) Составление платежного документа для конкретного заказа: товарный чек;

5) Проводить авторизацию.

Использовать систему будут пользователи средней и низкой квалификации. Интерфейс системы максимально приближен к интерфейсам подобных систем (приведенных в пункте 1.2).

Для переноса программы не должны требоваться специальные программные и аппаратные средства.

В результате внедрения автоматизированной системы будет снижено число ошибок менеджера при вводе данных, повысится оперативность выполнения функций, обеспечится быстрый доступ к данным. Повысит эффективность и качество обслуживания клиентов.

Предметная область предполагает формирование товарного чека и просчет стоимости заказа. Макет сформированного товарного чека представлен на рис. 1.4.

Фирма		ТОВАРНЫЙ ЧЕК№*		Дата заказа	
Наименование услуги		Количество	Цена	Сумма	
Итого:					
Подпись продавца					

Рис. 1.4. Макет товарного чека

Платежный документ (товарный чек) должен формироваться автоматически.

Формируется номер товарного чека, добавляется дата заказа, а также заполняется таблица, состоящая из наименования услуги, количества, цены, если позиций несколько обязательно указание итоговой суммы заказа.



## ГЛАВА 2. ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

### 2.1 Обоснование выбора средств разработки

Результатом дипломной работы является разработка информационной структуры, дизайна и информационное наполнение web-приложения, которое предоставляет всю необходимую информацию для менеджера сервисного обслуживания автомобилей.

Созданное web - приложение позволит получить исчерпывающую информацию о предоставляемых услугах, будет поддерживать аутентификацию менеджера по логину и паролю, позволит добавлять информацию о клиентах, а также добавлять поступившие заказы.

Преимущества веб-приложения:

- веб-приложения не требуют установки;
- для работы необходимы только браузер и интернет — не нужно устанавливать специальное программное обеспечение для того, чтобы пользоваться веб-приложением;
- разработка клиент-серверных приложений обычно обходится дешевле, чем мобильных;
- при внесении обновлений в веб-приложение они осуществляются на сайте автоматически.

Приложение должно быть удобным в использовании – страницы должны загружаться как можно быстрее, а сам ресурс должен обладать логичной и понятной структурой. Web-приложение должно состоять из связанных между собой web-страниц. Хорошо выполненный дизайн выглядит чистым, простым и аккуратным.

Независимо от конкретности проблемы, можно выделить некоторые формальные требования, на которых защищается квалификация разработчика.

Устойчивость. Web - приложение не должно терять работоспособности. Вводимая информация там, где возможно, подвергается логическому контролю.

Функциональная полнота. Все функции должны быть реализованы.

Использование клавиатуры. На любом этапе нажатия любой клавиши (особенно функциональных) должно игнорироваться или вызвать предусмотренные действия [11].

С типовой структурой важным становится вопрос средства разработки.

Создаю web - приложение с помощью языка описания структуры веб-страниц HTML и языка стилей CSS. При этом, для придания большей функциональности и динамичности сайтов использую и другие языки программирования, такие как PHP и JavaScript.

HTML – это неотъемлемая составляющая и основа практически любой web-страницы. Язык HTML в первую очередь выступает как средство логической разметки страницы. Он позволяет наделять содержимое страницы определенным смыслом.

Поскольку HTML - документы записываются в ASCII-формате, то для ее создания может быть использован любой текстовый редактор, и это очень удобно.

Одной из отличительных особенностей HTML - документов является то, что сам документ содержит только текст, а все остальные объекты встраиваются в документ в момент его отображения Браузером с помощью специальных тэгов и хранятся отдельно [13].

CSS используется для легкого изменения внешнего вида, придаваемый HTML - документам. С помощью CSS задаются для страницы и отдельных

ее элементов различные гарнитуры и размеры шрифта, расположения, отступы элементов друг от друга, цвета элементов.

Основной целью разработки CSS (Cascading Style Sheets) являлось разделение содержимого (написанного на html) и представления документа (написанного на CSS) [12].

HTML используется для структурирования содержимого страницы, а CSS - для форматирования этого структурированного содержимого.

Придание внешнего вида документам HTML – это самый популярный случай применения языка CSS. Целью создания CSS было отделение описания логической структуры веб-страницы от ее внешнего вида. Используется HTML для описания структуры, для описания же того, как эта логическая структура будет выглядеть, отвечает CSS.

Раздельное описание логической структуры и представления документа позволяет более гибко управлять внешним видом документа и минимизировать объем повторяющегося кода, который бы неизбежно возникал для описания внешнего вида документа при использовании HTML.

PHP на протяжении многих лет стабильно находится в списке наиболее популярных языков программирования и web-разработки.

Вообще PHP прекрасно работает в паре с HTML. Более того, в HTML-код можно делать вставки PHP-кода, а с помощью PHP выводить HTML-разметку.

PHP - это серверный язык создания сценариев. Конструкции PHP, вставленные в HTML - текст, выполняются сервером при каждом посещении страницы. Результат их обработки, вместе с обычным HTML-текстом передается браузеру [14].

Не имеет значения, насколько сложен PHP - код, в конечном итоге "на выходе" он станет обычным HTML.

PHP - программы работают быстрее, чем ASP. Разработку PHP-программы можно отделить от собственно разработки web-страницы. PHP абсолютно бесплатен.

Использование языка PHP при web - разработке, позволяет создать простой, эффективный, многофункциональный, безопасный и самое главное динамичный сайт.

Простота в использовании. На распространенных языках присутствует синтаксис PHP хорошо знакомый.

JavaScript – это язык программирования, нашедший наиболее широкое применение в браузерах для придания интерактивности web-страницам. Он позволяет активно управлять структурой страницы, манипулировать ее элементами. Его применяю для создания слайдера фотографий на главной странице.

JavaScript это язык подготовки сценариев, позволяющий сделать web-страницы более интерактивными и функциональными [13].

С помощью JavaScript можно:

- динамически изменять содержимое web-страниц;
- привязывать к элементам обработчики событий (функции которые выполняют свой код только после того, как совершатся определенные действия);
- выполнять код через заданные промежутки времени;
- управлять поведением браузера (открывать новые окна, загружать указанные документы);
- создавать и считывать cookies;
- определять, какой браузер использует пользователь (также можно определить ОС, разрешение экрана, предыдущие страницы, которые посещал пользователь и так далее);
- проверять данные форм перед отправкой их на сервер и многое другое.

Фундаментом web-разработки был и остается язык HTML. Без него все остальное практически лишено смысла, ведь именно HTML - разметка преобразуется браузером в ту итоговую картину, что мы видим на экране монитора.

CSS является инструментом задания внешнего вида и позиционирования различных элементов web-страницы, что позволяет гибко управлять обликом web-приложения.

PHP позволяет создавать мощные динамические сайты, обладающие определенной логикой поведения. Он также дает практически неограниченные возможности по расширению функционала сайта.

JavaScript выступает в качестве важного, но не всегда обязательного дополнения. Он способен существенно повысить удобство использования сайта и его интерактивность, добавить в него определенную «изюминку».

## **2.2 Выбор системы управления базы данных**

Средства автоматизации – это инструментальные системы, с помощью которых создаются новые программные продукты, и готовые программные комплексные, настраиваемые на потребности конкретного пользователя.

В качестве СУБД в данной работе могут быть использованы, например: PostgreSQL, MySQL, IBM DB2.

SQL позволяет организовать взаимодействие сайта с базой данных, что и позволяет, в связке с языком PHP создавать действительно мощные, функциональные сайты.

PostgreSQL — свободная объектно - реляционная система управления базами данных.

PostgreSQL допускает использование функций, возвращающих набор записей, который далее можно использовать так же, как и результат выполнения обычного запроса.

Размер строки 656 КБ, максимальная скорость 1 поток 21974 КБ/с, пороговая скорость 5494 КБ/с, максимальное количество потоков 14, максимальная скорость 83094КБ/с [16].

PostgreSQL может быть расширен пользователем для собственных нужд практически в любом аспекте. Есть возможность добавлять собственные:

- преобразования типов;
- типы данных;
- домены (пользовательские типы с изначально наложенными ограничениями);
- функции (включая агрегатные);
- индексы;
- операторы (включая переопределение уже существующих);
- процедурные языки.

PostgreSQL представляет из себя объектно реляционную базу данных, которая работает только на одном движке - storage engine. Все таблицы представлены в виде объектов, они могут наследоваться, а все действия с таблицами выполняются с помощью объективно ориентированных функций.

PostgreSQL - это проект с открытым исходным кодом, он разрабатывается командой энтузиастов, и разработчики пытаются максимально соответствовать стандарту SQL и реализуют все самые новые стандарты. Но все это приводит к ущербу простоты. Postgresql очень сложный и из-за этого он не настолько популярен как MySQL.

MySQL удобна для проектов по веб разработке, и которые не требуют обработки большого количества информации. Проектирование баз данных MySQL не требует огромных затрат сил и времени.

MySQL - свободно распространяемая система. Кроме того, это достаточно быстрая, надежная и, главное, простая в использовании СУБД.

Преимущества:

- бесплатная СУБД;
- имеет открытый код;
- занимает мало места;
- лёгкое администрирование MySQL с набором всего необходимого функционала;
- большое сообщество разработчиков;
- работает на множестве платформ.

Недостатки: незащищённость от потери данных, поддержка только малых баз данных.

MySQL поддерживает далеко не все новые возможности стандарта SQL. Разработчики выбрали именно этот путь развития, чтобы сохранить MySQL простым для использования. Компания пытается соответствовать стандартам, но не в ущерб простоте. Если какая-то возможность может улучшить удобство, то разработчики могут реализовать ее в виде своего расширения, не обращая внимания на стандарт [19].

IBM DB2 – это наиболее высокопроизводительная и мощная СУБД в мире. Ее основное уникальное преимущество в том, что любое приложение, написанное для DB2, будет работать с серверами данных DB2, работающими на любой распределенной платформе, поддерживаемой DB2 (Windows, HP-UX, Sun Solaris, Linux, Mac OS X и AIX®).

IBM DB2 размер строки 128 КБ, максимальная скорость 1 поток, 1445 КБ/с, пороговая скорость 361 КБ/с, максимальное количество потоков 10, максимальная скорость 4439 КБ/с [10].

Функции, определяемые пользователем, позволяют достичь многократного использования кода за счет того, что операции, общие для различных приложений, хранятся на сервере, а не включаются в каждое отдельное приложение.

Использование пользовательских функций вместо непосредственного доступа к данным может обеспечить некоторую инкапсуляцию данных, что можно использовать для того, чтобы скрыть от пользователя их внутреннюю структуру.

Дополнительную гибкость функциям, определяемым пользователем, придает способность одновременно работать как с данными DB2, так и другими данными, как, например, файлами, электронной почтой.

Пользователь может выбирать оптимальный для своей задачи подход в зависимости от ее специфики.

Плюсы СУБД:

- есть хорошая бесплатная версия;
- в бесплатной версии нет ограничений на размер базы;
- в бесплатной версии нет ограничителя запросов, это позволяет одновременно обслуживать десятки пользователей;
- хорошая бесплатная техподдержка;
- в отличие от PostgreSQL есть возможность получить платную поддержку производителя, что позволяет применять в Enterprise секторе бизнеса;
- с конфигурациями 1С-Предприятие в автоматическом режиме блокировок работает лучше, чем PostgreSQL (речь о параллельности, область блокировок на уровне строк, а не таблиц);
- хорошая производительность;
- меньше проблем с неуникальностью индексов (фактически для решения проблемы рекомендуется временно базы загружать в DB2);
- лучше обрабатывает ситуации вроде "не хватает памяти для сервера 1С";
- нет ограничения на 256 таблиц, что расширяет возможности при работе с RLS.

Минусы:



- мало специалистов и высокая стоимость хороших специалистов;
- небольшая распространенность;
- в отличие от MS SQL Server для новых версий 1С выпускает "адаптированные" версии;
- размер баз больше, чем в других СУБД;
- медленная загрузка dt - файла;
- требуется "тонкая" настройка параметров СУБД, автоподстройка системы есть, но неполная;
- некоторые сообщения платформой могут неверно обрабатываться, для решения приходится "понижать уровень" логирования ошибок;
- 1С-ЦУП собирает длительные запросы, ожидания на блокировках, но не собирает взаимные блокировки [2].

В общем результате лучшим по возможностям получается PostgreSQL, но он сложен и не везде его можно применять. MySQL проще, но не поддерживает некоторых интересных функций.

MySQL часто рассматривают как быструю базу данных, более легкую в настройке и использовании, а PostgreSQL с самого начала разрабатывалась как база с большим числом настроек и более точным соответствием стандарту.

Из приведенных выше преимуществ и недостатков выбрали в качестве СУБД - MySQL.

### **2.3 Проектирование структуры БД**

В соответствии с требованиями, автоматизированная система должна обеспечивать хранение информации о клиентах, услугах, заказах.

На основании этого были выделены следующие сущности:

1) Клиенты. Данная сущность содержит информацию о клиентах и имеет следующие атрибуты:

- код клиента;

- фамилию, имя, отчество клиента;
- email;
- телефон.

2) Услуги. Данная сущность содержит информацию об услугах, представляющихся в сервисном:

- код услуги;
- название услуги;
- стоимость услуги.

3) Заказы. Данная сущность содержит всю необходимую информацию о заказах и имеет следующие атрибуты:

- код заказа;
- код клиента;
- код услуги в заказе;
- дата начал/окончания работ;
- статус.

4) Услуга в заказе:

- код услуги в заказе;
- код заказа;
- код услуги.

Ключевыми являются атрибуты код клиента, код услуги, код заказа, код услуги в заказе соответственно.

Сущность «Клиенты» находится в первой нормальной форме, так как каждый из атрибутов не имеет повторяющихся значений. Кроме этого она находится во второй нормальной форме, поскольку атрибуты Фамилия, Имя, email, телефон зависят только от ключевого поля - код клиента; и в третьей нормальной форме, поскольку, кроме этого, между не ключевыми атрибутами нет функциональных зависимостей.

Сущность «Заказы» находится в первой нормальной форме, то есть каждый ее атрибут может содержать только одно значение. Кроме этого, она

находится во второй нормальной форме, все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом. И в третьей нормальной форме, поскольку любой ее неключевой атрибут зависит только от первичного ключа, набор атрибутов не содержит повторяющихся групп.

Сущность «Услуги» находится в первой нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто. Кроме этого, она находится во второй нормальной форме, поскольку атрибуты наименование услуги, стоимость зависят только от ключевого поля – код услуги. Также она находится в третьей нормальной форме, поскольку кроме этого, между не ключевыми атрибутами нет функциональных зависимостей.

Таким образом, все указанные сущности находятся в третьей нормальной форме.

Между указанными сущностями существуют следующие отношения (см. рис. 2.1):

- между клиентом и заказом есть отношение типа один ко многим, то есть на одного клиента может быть оформлено множество заказов, но конкретный заказ соответствует только одному клиенту;

- между заказом и услугой в заказе есть отношение один ко многим, то есть заказ может содержать несколько услуг, но каждая услуга в заказе соответствует только одному заказу;

- между услугой и услугой в заказе есть отношение один ко многим, то есть одна услуга может содержаться несколько услуг в заказе, но каждая услуга в заказе соответствует только одной услуге.

В типе связей один ко многим, записи первой таблицы соответствуют нескольким записям в другой таблице.

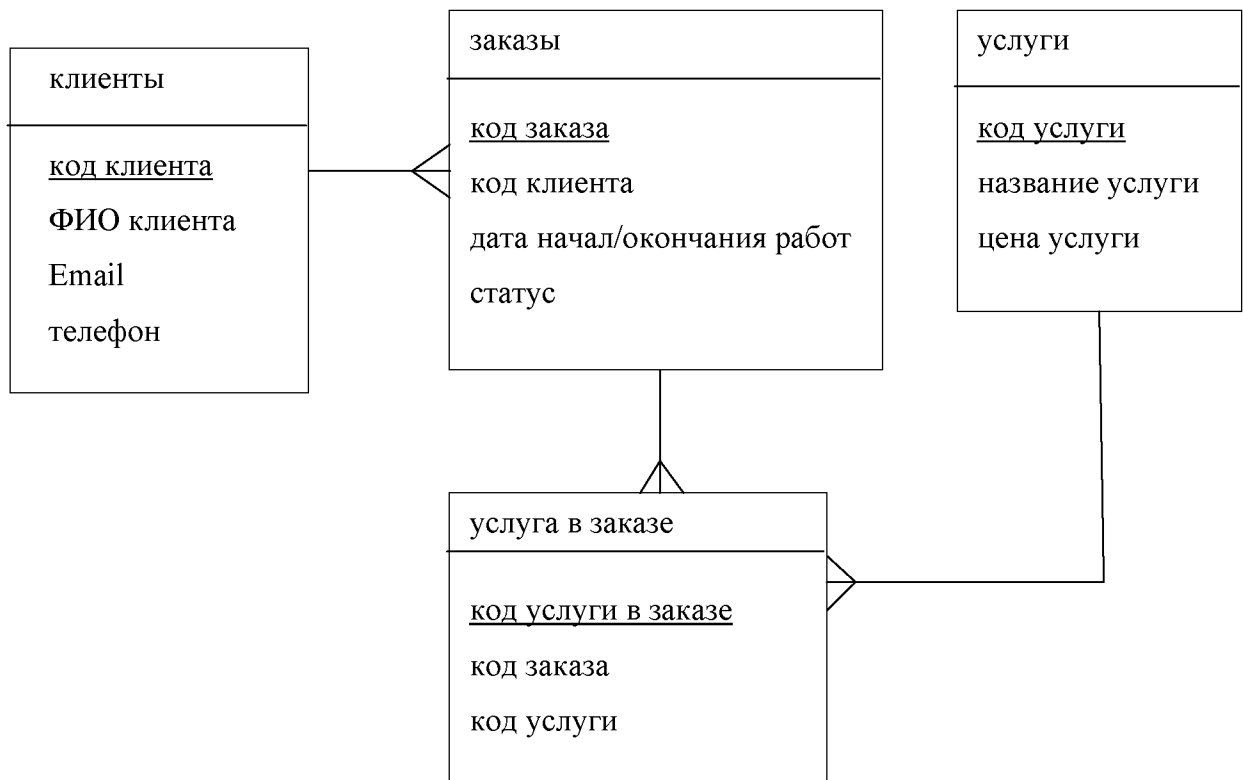


Рис. 2.1. ERD-диаграмма

Этим сущностям в БД соответствуют таблицы:

### 1) Clients – Клиенты

С атрибутами и полями:

- код клиента – client\_id, int, primary key;
- фамилия, имя, отчество - client\_name, varchar;
- телефон - client\_phone, varchar;
- email - client\_email, varchar.

### 2) Orders – Заказы

С атрибутами и полями:

- код заказа - order\_id, int, primary key;
- код услуга в заказе - order\_service, int;
- дата - order\_date, datetime;
- статус - order\_status, varchar.

### 3) Services – Услуги

С атрибутами и полями:

- код услуги - service\_id, int, primary key;

- название услуги - service\_name, varchar;
- цена услуги - order\_service\_price, float.

Структура базы данных приведена на рис. 2.2.

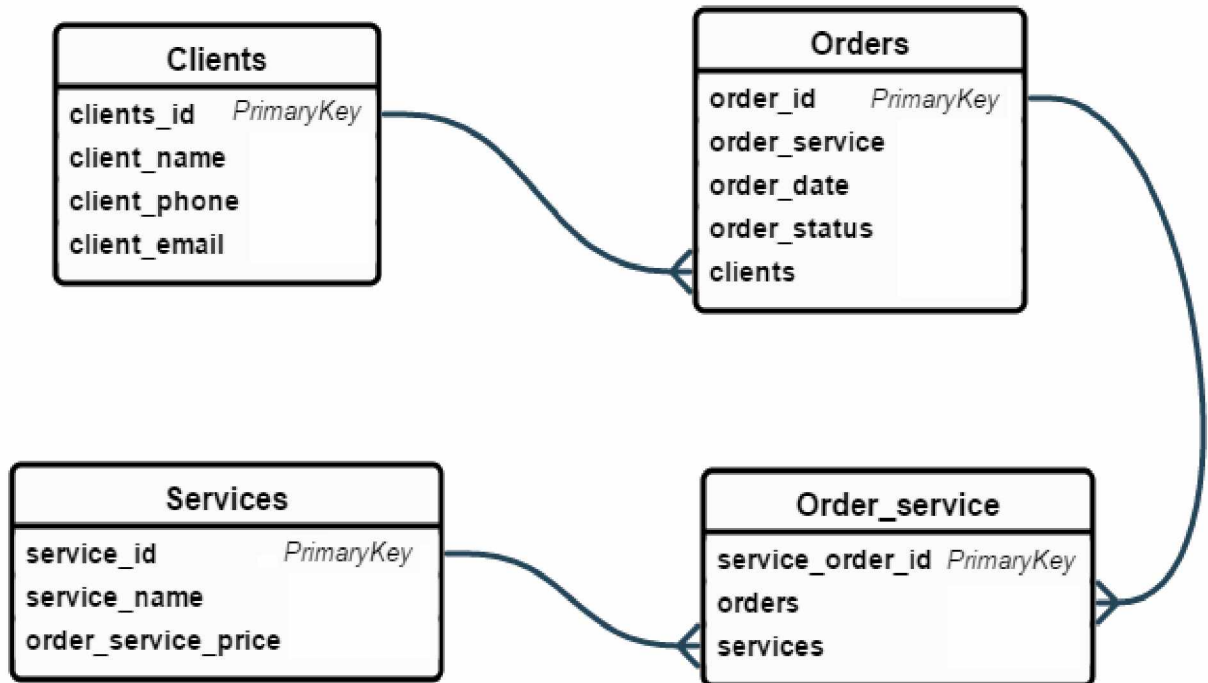


Рис. 2.2. Структура БД

## 2.4 Проектирование структуры web-приложения

Исходя из требований к интерфейсу и функциональным требованиям предлагается следующая структура, включающая следующие страницы:

- главная;
- диагностика;
- ремонт;
- запчасти;
- шиномонтаж.

На рис. 2.3 представлена структурная схема веб-приложения, которая учитывает всю специфику разрабатываемого приложения.

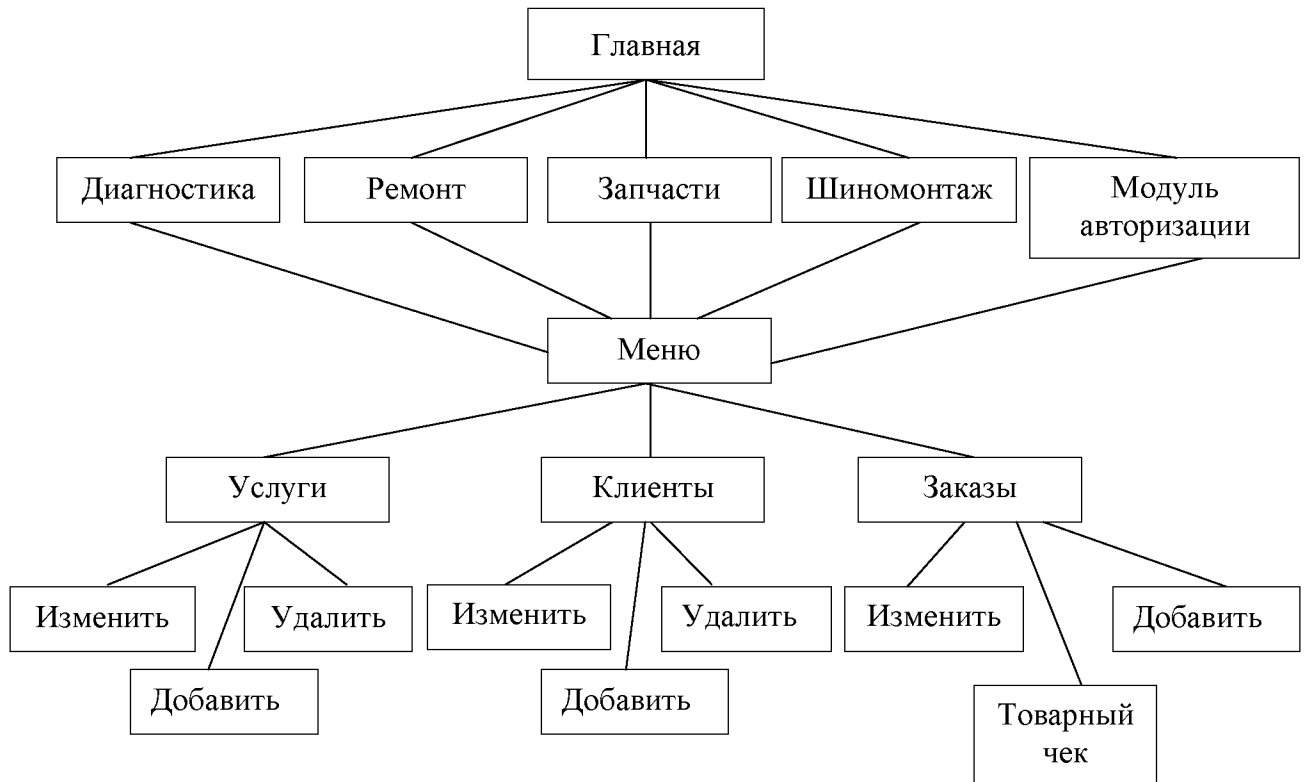


Рис. 2.3. Общая структура web-приложения

На главной странице отображается информация о сервисном центре, описано, что может предложить команда автосервиса. На макете главной страницы, видно, что содержатся такие вкладки как: диагностика, ремонт, запчасти, шиномонтаж, а также по правой стороне странице расположено меню. Макет главной страницы показан на рис. 2.4.

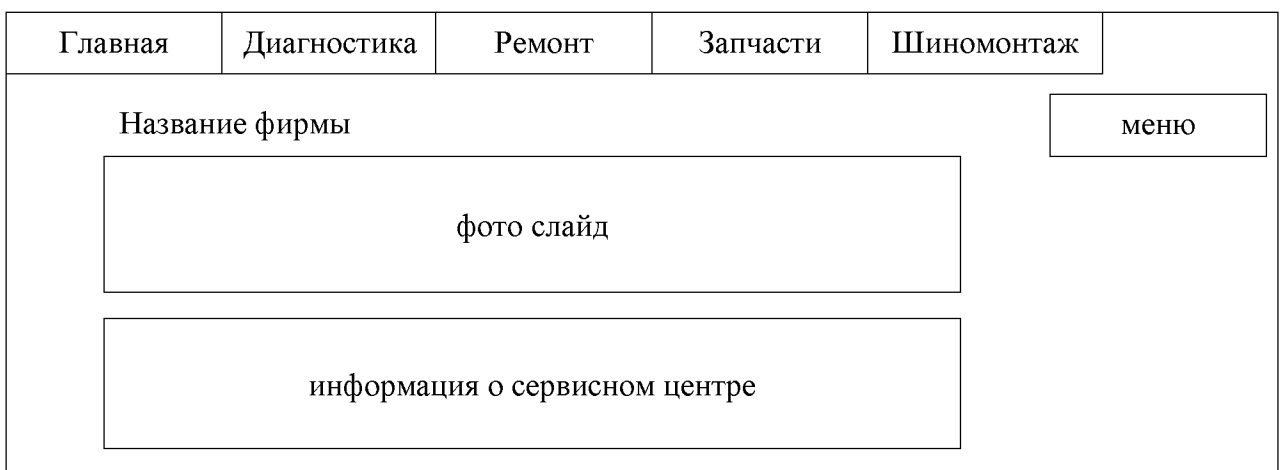


Рисунок 2.4. Макет главной страницы

Страница «Диагностика» содержит информацию о видах предоставляемых диагностик в сервисном центре. Макет страницы представлен на рис. 2.5.

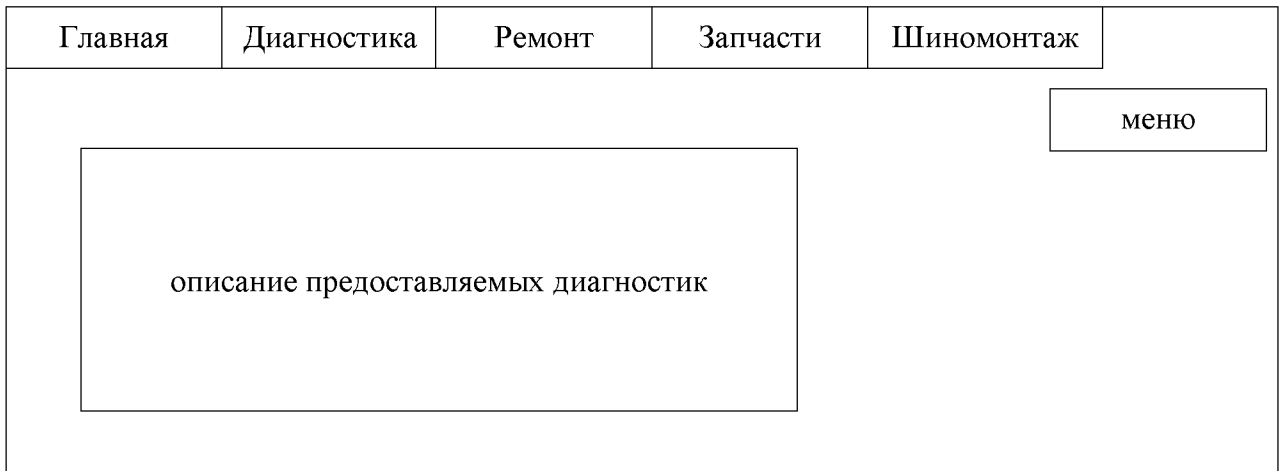


Рис. 2.5. Макет страницы «Диагностика»

При переходе на страницу «Диагностика», также остаются видны вкладки: диагностика, ремонт, запчасти, шиномонтаж и отображается справа меню.

Страница «Ремонт» содержит информацию о видах ремонта, которые можно получить в сервисном центре.

Макет страницы представлен на рис. 2.6.

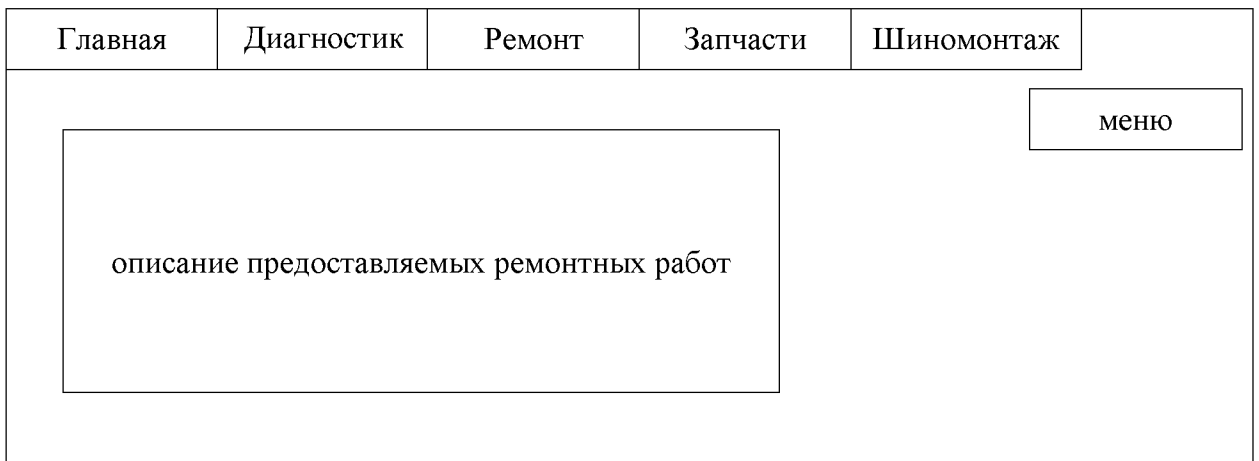


Рис. 2.6. Макет страницы «Ремонт»

Страница «Запчасти», макет которой показан на рис.2.7, информирует о том, что сервисный центр может предоставить запчасти на автомобиль

любой марки, сроки доставки. С этой страницы также можно перейти на любую страницу.

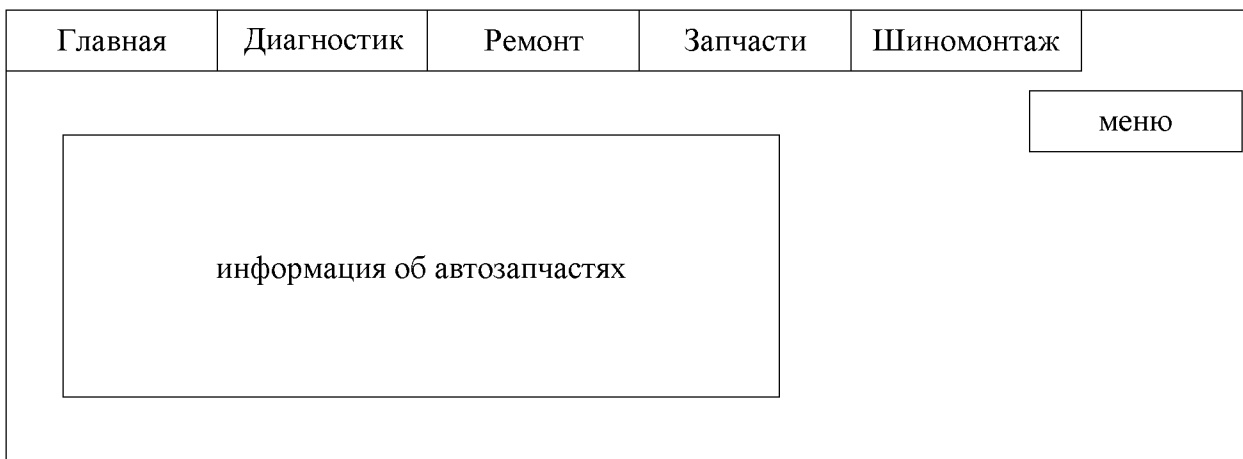


Рис. 2.7. Макет страницы «Запчасти»

Страница «Шиномонтаж» содержит информацию о предоставляемом шиномонтаже. С этой страницы можно также перейти на любую другую страницу сайта. Макет страницы представлен на рис. 2.8.

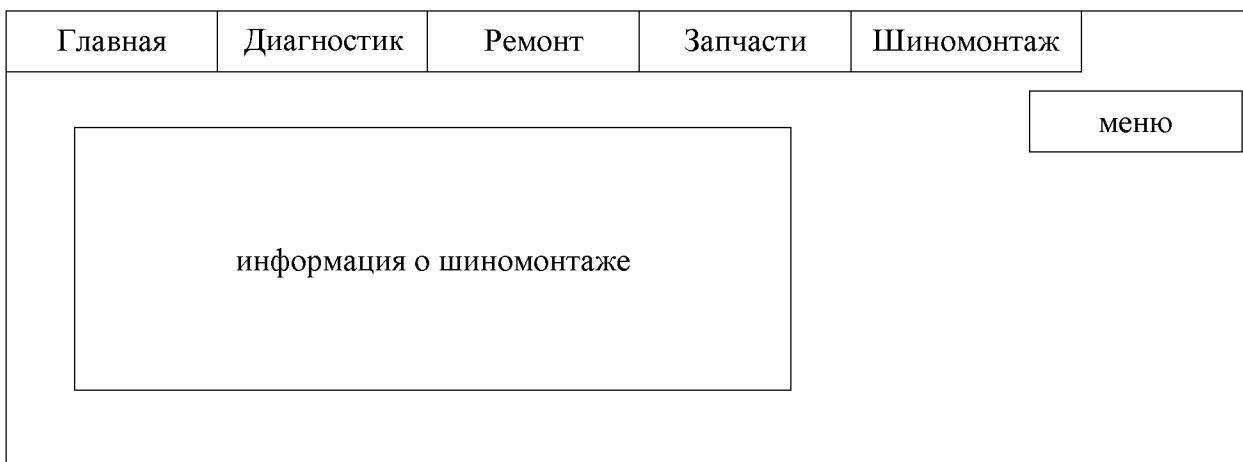


Рис. 2.8. Макет страницы «Шиномонтаж»

На странице справа имеется «Меню» в котором отображены страницы: услуги, клиенты, заказы. Макет показан на рис. 2.9.

меню
услуги
клиенты
заказы

Рис. 2.9. Макет «Меню»



Меню отображается на всех страницах сайта. Страницы «Клиенты» и «Заказы» видны только менеджеру сервисного центра (авторизованному пользователю).

Страница «Услуги» - которая содержит информацию о видах и ценах услуг. Макет страницы представлен на рис. 2.10.

Главная	Диагностик	Ремонт	Запчасти	Шиномонтаж	
				добавить услугу	меню
наименование		стоимость	управление		
			изменить	удалить	

Рис. 2.10. Макет страницы «Услуги»

Добавлять, изменять услуги, а также стоимость, может менеджер сервисного центра.

Страница «Клиенты» содержит информацию о клиентах (фамилию, имя, отчество, телефон, email).

Макет страницы «Клиенты» представлен на рис. 2.11.

Информацию о клиентах может добавлять, изменять и удалять менеджер сервисного центра. В базе занесены все клиенты, которые заказывали услугу в автосервисе.

Главная	Диагностик	Ремонт	Запчасти	Шиномонтаж	
				добавить клиента	меню
клиент	email	телефон	управление		
			изменить	удалить	

Рис. 2.11. Макет страницы «Клиенты»

Страница «Заказы» - содержит информацию о заказах, которые добавляет и изменяет менеджер сервисного центра (см. рис. 2.12).

Главная	Диагностик	Ремонт	Запчасти	Шиномонтаж	
				добавить заказ	меню
статус					
Заказ №1		чек	изменить		
Заказ №2		чек	изменить		
Заказ №3		чек	изменить		
Заказ №4		чек	изменить		

Рис. 2.12. Макет страницы «Заказы»

На основании разработанной структуры было реализовано web-приложение, исходный код которого приведен в приложении.

## ГЛАВА 3. ИСПЫТАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

### 3.1 Программа и методика испытаний

Цель испытаний: показать, что система требованиям удовлетворяет.

В соответствии с требованиями система должна выполнять такие эксплуатационные свойства:

- удобный интерфейс;
- эффективная;
- мобильная.

Для проверки этого нужно сделать тестирование:

- 1) Тестирования удобства использования;
- 2) Функционального тестирования;
- 3) Тестирования совместимости;
- 4) Тестирования баз данных.

Необходимо получить результат и проанализировать что получим.

1) Тестирование удобства использования - это не что иное, как тестирование дружелюбности приложения для пользователя. При тестировании удобства использования проверяется, легко ли разобраться новому пользователю в приложении. В целом при тестировании удобства использования тестируется системная навигация.

Тест удобства использования удостоверяется в эффективности и простоте использования продукта при использовании стандартных практик тестирования удобства использования.

Сценарии тестирования удобства использования:

- содержание web - страницы верное, без грамматических и орфографических ошибок;
- все тексты правильно выровнены;

- все сообщения об ошибках верные, без орфографических и грамматических ошибок;
- все поля правильно выровнены;
- все кнопки имеют нужный формат и размер;
- ссылка на домашнюю страницу на каждой странице сайта;
- неактивные поля выделяются отличительным цветом от активного;
- подтверждающие сообщения отображаются для всех операций обновления, добавления и удаления;
- tab правильно работает;
- заголовок отображается на каждой странице.

2) Функциональное тестирование – это тестирование функциональностей и операционного поведения приложения с целью убедиться, что они соответствуют спецификациям; тестирование, игнорирующее внутренние механизмы системы или компонента. Оно концентрируется исключительно на выходных данных, полученных в ответ на пользовательский ввод и условия исполнения сценариев.

Цель функционального тестирования – убедиться, что приложение соответствует нужной функциональной спецификации.

Сценарии функционального тестирования:

- протестировать максимальную длину каждого поля, чтобы убедиться, что данные не обрезаются;
- убедиться, что подтверждающее сообщение отображается для операций обновления, добавления и удаления;
- убедиться, что значения стоимости отображаются в нужной валюте;
- протестировать все поля ввода на спецсимволы;
- протестировать функциональность сортировки;
- протестировать функциональность доступных кнопок;
- протестировать, что при отказе функциональности пользователь перенаправляется на специальную страницу ошибки;

- протестировать, что все документы правильно открываются;
- протестировать, что java script верно работает в разных браузерах (explorer, chrome, opera).

3) Тестирование совместимости используется, чтобы убедиться, что приложение совместимо с другими элементами системы, в которой оно работает – например, браузерами, операционными системами или техническими средствами поддержки.

Цель тестирования совместимости – оценка того, насколько хорошо ПО работает в определенном браузере, под определенной ОС, с другим программным обеспечением.

Сценарии тестирования совместимости:

- протестировать сайт в различных браузерах (explorer, chrome, opera) и убедитесь, что сайт правильно отображается;
- убедиться, что используемая версия html совместима с соответствующими версиями браузеров;
- убедиться, что картинки корректно отображаются в разных браузерах;
- убедиться, что шрифты верно отображаются в разных браузерах;
- убедиться, что java script код работает в разных браузерах.

4) Тестирование баз данных. При этом тестировании проверяются backend записи, введенные через веб - приложение. Данные, которые отображаются в приложении, должны совпадать с данными, хранящимися в базе данных.

Сценарии тестирования баз данных:

- проверить название базы данных: оно должно совпадать со спецификацией;
- проверить таблицы, колонки, типы колонок и значения по умолчанию: все это должно совпадать со спецификацией;
- проверить первичный и внешний ключ каждой таблицы;
- проверить процедуры хранения;

- проверить процедуру хранения, задав простые sql - запросы;
- проверить процедуру вводом тестовых данных;
- убедиться, что данные правильно сохраняются в базе данных после каждого ввода;
- проверить данные при каждой операции обновления, удаления и добавления;
- проверить длину каждого поля;
- проверить размер базы и время отклика на каждый запрос;
- проверить триггеры.

## 3.2 Результаты испытаний

Для достижения цели испытания, в соответствии с требованиями, проводится тестирование web - приложения.

### 3.2.1 Тестирования удобства использования

По тестирование удобства была совершена проверка web-приложения. проверено содержание web-страниц, все тексты без грамматических и орфографических ошибок, все тексты выровнены (см. рис. 3.1, рис. 3.2).

**Автосервис "АвтоДоктор"** — современный автотехцентр, специализирующийся на оказании высококачественных услуг автомобилям европейских, японских, корейских, американских и отечественных производителей.

Качественный ремонт автомобиля, наличие на складе и оперативный подбор автозапчастей – и все это по умеренным ценам. СТО "АвтоДоктор" осуществляет ремонт с 2006 г. и всегда рад предложить своим клиентам гибкие условия сотрудничества. Наши механики имеют многолетний опыт работы на специализированных сервисах официальных дилеров европейских и японских автомобилей.

Специалистами нашего автосервиса осуществляются все процедуры по комплексной компьютерной диагностике всех видов транспортных средств, техническому обслуживанию и ремонту любых узлов и систем автомобилей.

Наша команда механиков осуществляет ремонт любой сложности для всех марок автомобилей с **гарантией качества проведенных работ**. Мы знаем, что автомобиль для Вас не просто роскошь, это ещё и необходимое средство передвижения, без которого долгое время просто не обойтись. Поэтому в автотехцентре «АвтоДоктор» проводятся все необходимые работы по ремонту Вашего авто оперативно.

Ремонтные боксы нашего автосервиса - оснащены новейшим первоклассным оборудованием западных и отечественных производителей для проведения необходимых ремонтных работ.

Рис. 3.1. Текст без ошибок

СТО "АвтоДоктор" предлагает все виды диагностических услуг в Белгороде и последующий ремонт.

Диагностика автомобилей производится с помощью современных компьютерных оборудований и тесторов, которые позволяют выявить всевозможные ошибки автомобилей любых марок

Виды диагностики оказываемые нашей СТО:

- Компьютерная диагностика двигателя;
- Компьютерная диагностика всех электронных систем;
- Компьютерная диагностика системы трансмиссии (АКПП и МКПП);
- Диагностика систем управления;
- Диагностика ходовой;
- Диагностика системы кондиционирования;
- Диагностика инжектора;
- Комплексная диагностика автомобиля;
- Диагностика автомобиля перед покупкой.

Мы решим Вашу проблему звоните и получите консультацию по всем интересующим вопросам.

Рис. 3.2. Текст без ошибок

Сообщения об ошибках верные, без орфографических ошибок. Ссылка на домашнюю страницу на каждой странице сайта. Неактивные поля выделяются отличительным цветом - от активного.

Подтверждающее сообщение отображаются для всех операций добавления, обновления и удаления, как показано на рис. 3.3.

Заголовок отображается на каждой странице. Tab работает правильно.

## Редактирование информации о клиенте

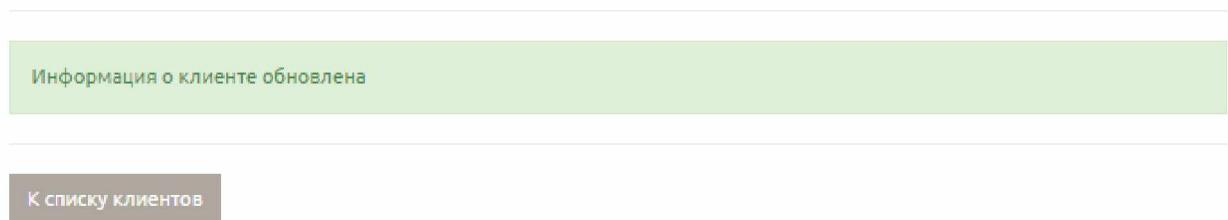


Рис. 3.3. Подтверждающее сообщение

### 3.2.2 Тестирование функциональных возможностей

Благодаря функциональному тестированию, убедились, что приложение соответствует нужной функциональной спецификации.

Были протестированы поля, на максимальную длину, для убеждения, что данные не обрезаются.

Убедились, что подтверждающие сообщения о добавлении, обновлении и удалении отображаются.

Менеджер может внести изменения информации о клиенте, после чего будет отображена страница информирующая об успешном редактировании информации (см. рис. 3.3.)

После удаления из таблицы информации о клиенте, отображается соответствующая страница, представленная на рис. 3.4, информирующая об успешном удалении.

Если информация о клиенте удалена ошибочно, ее можно восстановить, после чего будет отображена страница, как представлено на рис. 3.5, информирующая о восстановлении.

Для добавления информации о новом клиенте, после нажатия на кнопку «добавить», необходимо заполнить таблицу информацией о клиенте, и нажать кнопку «сохранить», после чего отобразится соответствующее окно об успешном добавлении клиента в базу (см. рис. 3.6).

## Удаление клиента

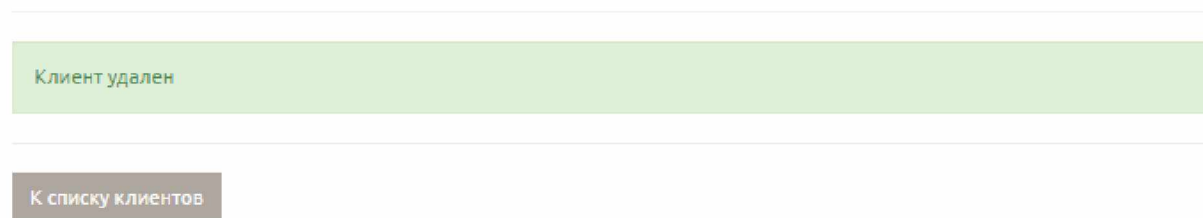


Рис. 3.4. Подтверждающее сообщение

## Восстановление клиента

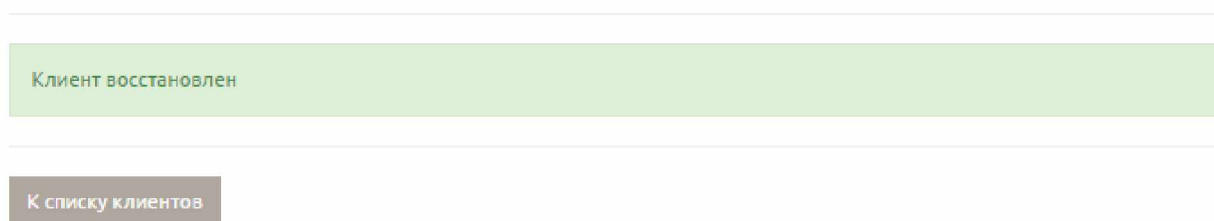


Рис. 3.5. Подтверждающее сообщение



## Добавление клиента

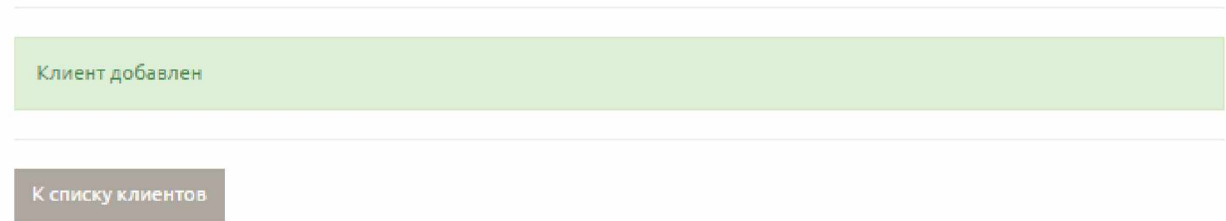


Рис.3.6. Подтверждающее сообщение

Значения стоимость отображаются в нужной валюте (см. рис. 3.7).

Стоимость
550,00 ₺
500,00 ₺
650,00 ₺
600,00 ₺
600,00 ₺
2 000,00 ₺
500,00 ₺
600,00 ₺
150,00 ₺
600,00 ₺
100,00 ₺
180,00 ₺
900,00 ₺

Рис. 3.7. Валюта полей

Протестирована функциональность доступных кнопок. Также проверено, что JavaScript верно работает в разных браузерах (см. рис. 3.8, рис.3.9, рис. 3.10).

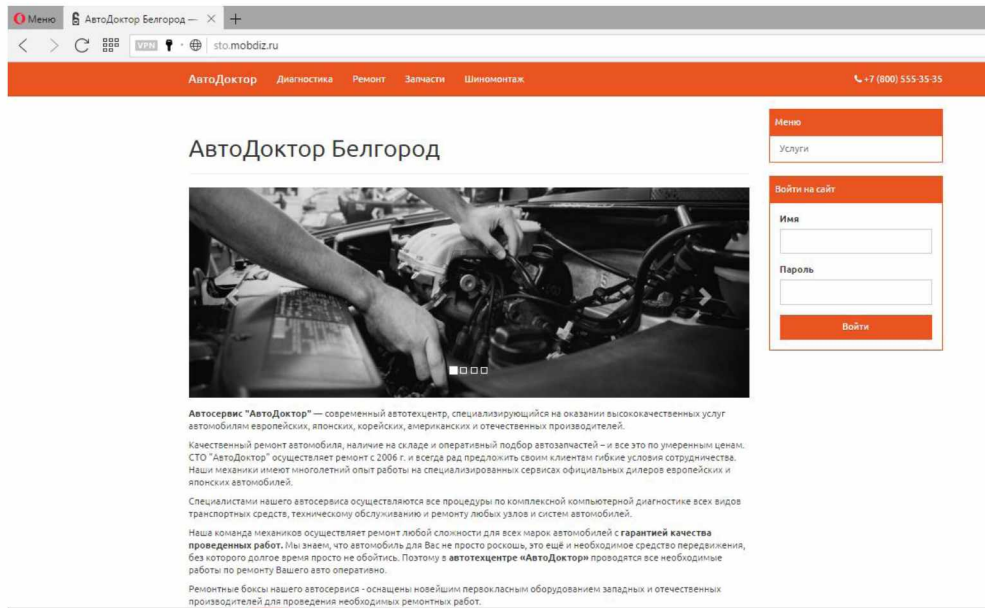


Рис.3.8. Проверка в браузере Opera

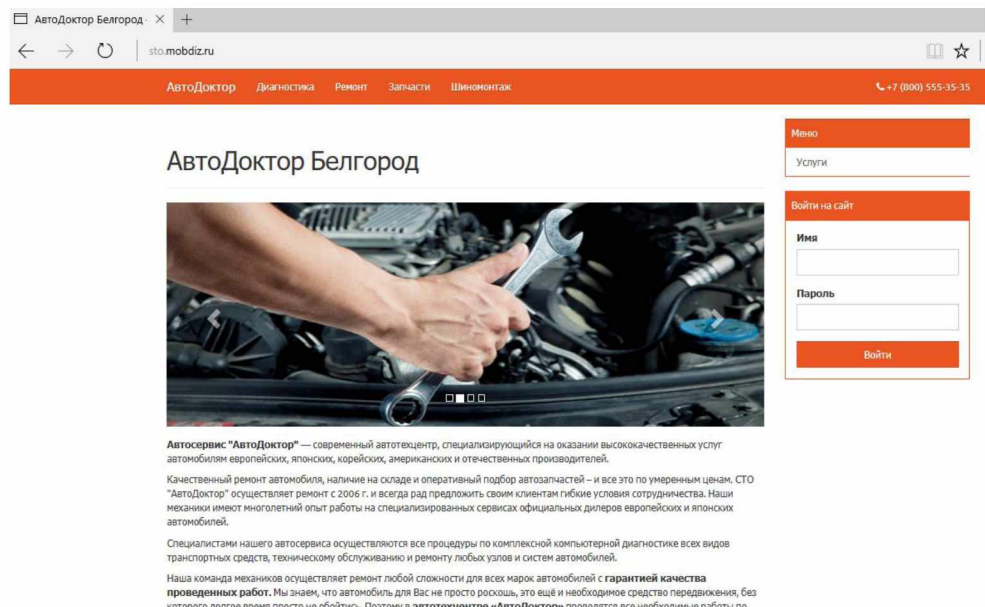


Рис. 3.9. Проверка в браузере Explorer

Протестировали сайт в таких браузерах как:

- Opera;
- Explorer;
- Google Chrome.

Сайт в этих браузерах отображается верно, картинки отображаются корректно, шрифты отображены верно. Сайт во всех браузерах отображается идентично.



Рис. 3.10. Проверка в браузере Google Chrome

Главная страничка сайта, которая показана на рис. 3.11, содержит все основные структурные элементы, переход по которым осуществляется с помощью гиперссылок.

На главной странице отображается информация о сервисном центре, описано, что может предложить команда автосервиса.



Рис. 3.11. Главная страница

По правой стороне находится панель управления, с помощью которой менеджер сервисного центра авторизовывается на сайте, для управления страницами - клиенты, заказы, услуги.

Как и планировалось на этапе постановки задачи, сайт содержит все необходимые структурные и навигационные элементы.

Работа программы основана на диалоге с пользователем через специальные экранные формы. Через форму «Главная страница» можно посредством кнопок перейти к другим формам. На главной форме расположены кнопки «Диагностика», «Ремонт», «Запчасти», «Шиномонтаж».

Через эти формы можно просмотреть информацию обо всех предоставляемых услугах сервисным центром.

При нажатии на ссылку «Услуги» можно ознакомиться с предоставляемыми услугами, а также ценами на услуги в сервисном центре (см. рис. 3.12).

АвтоДоктор   Диагностика   Ремонт   Запчасти   Шиномонтаж   +7 (800) 555-35-35

## Услуги

Как только подошло время прохождения технического обслуживания, или в Вашем автомобиле появился странный звук, запах или загорелась сигнальная лампочка, специалисты СТО АвтоДоктор изучат ситуацию, выяснят причину и предложат Вам пути ее решения.

Цены на типовые услуги Вы найдете далее. Окончательная цена, перечень работ и срок выполнения определяется индивидуально и зависит исключительно от сложности. Позвоните нам, и менеджер ответит на Ваши вопросы.

Название услуги	Стоимость
Замена жидкости гидроусилителя	550,00 ₺
Замена масла в двигателе, легковые а/м	500,00 ₺
Замена масла в двигателе, микроавтобусы и внедорожники	650,00 ₺
Замена масла в МКПП	600,00 ₺
Замена масла в раздаточной коробке	600,00 ₺
Замена топливного насоса	2 000,00 ₺
Замена топливного фильтра под днищем	500,00 ₺
Замена тормозной жидкости легковые а/м	600,00 ₺
Регулировка стояночного тормоза	150,00 ₺
Регулировка угла опережения зажигания	600,00 ₺
С/у дополнительной защиты картера двигателя	100,00 ₺
Установка номерных знаков	180,00 ₺

Меню  
Услуги

Войти на сайт

Имя  
admin

Пароль  
\*\*\*\*\*

Войти

Рис. 3.12. Услуги

Согласно программе и методике испытаний, система должна выполнять авторизацию менеджера.

Вводим имя и пароль, нажимаем кнопку «Войти», если все верно, отображается соответствующая страница и активируется меню доступное менеджеру (см. рис.3.13).

Становятся доступны страницы «Клиенты» и «Заказы» (см. рис.3.14).

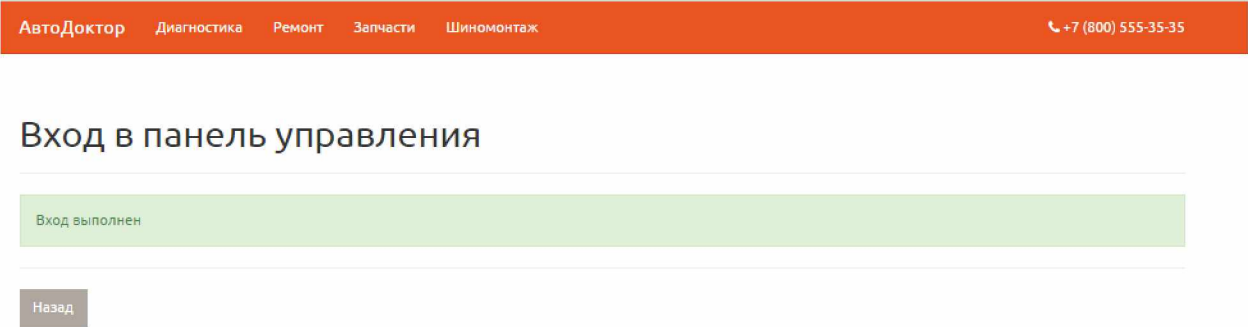


Рис.3.13. Вход в панель управления

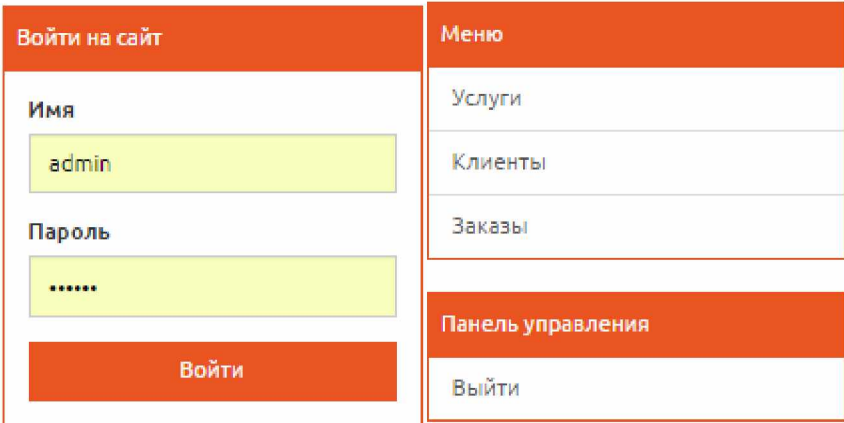


Рис. 3.14. Регистрационная форма

Если ввести логин и пароль неверно, нажать кнопку «Вход», то мы увидим соответствующее окно, оповещающее ошибку (см. рис.3.15).

## Ошибка



Рисунок 3.15. Неверный ввод авторизации

По ссылке «Клиенты» предоставляется контактная информация о клиентах обращающихся в сервисный центр, доступная менеджеру сервисного центра (см. рис. 3.16).

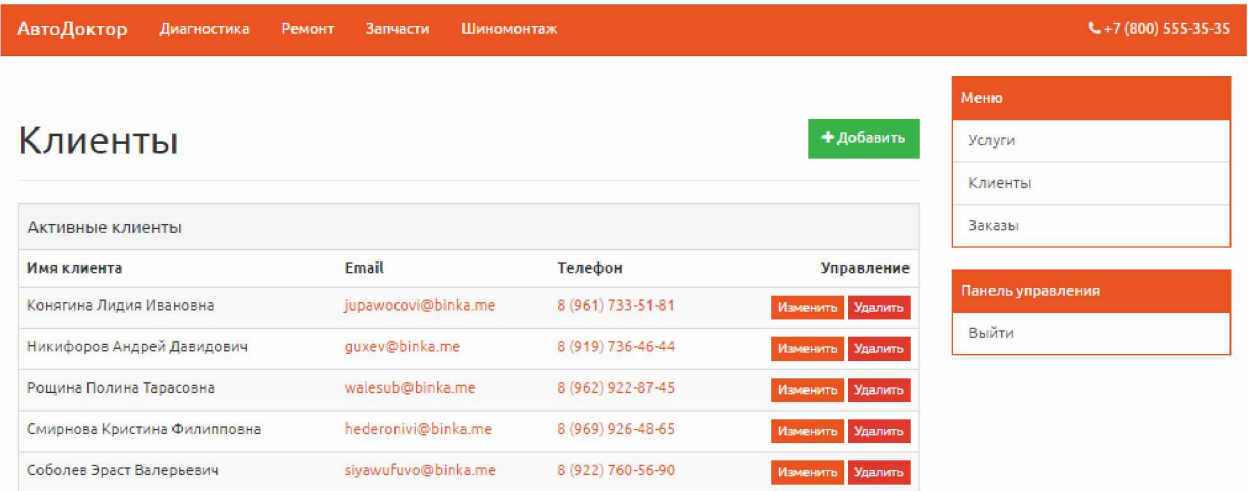


Рис. 3.16. Клиенты

При нажатии на ссылку «Заказы», менеджеру доступна информация обо всех заказах (новых, выполняемых, выполненных, просроченных и отмененных), которые он может редактировать (см. рис. 3.17).

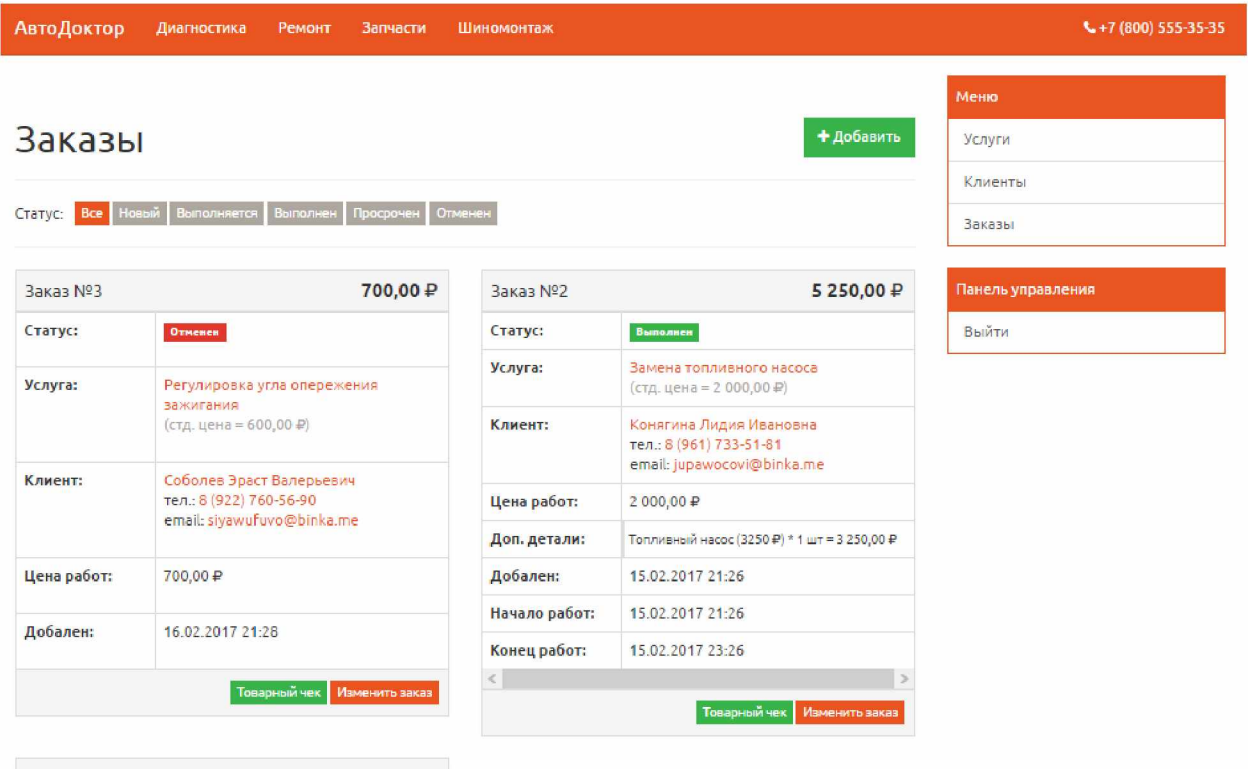


Рис. 3.17. Заказы

Менеджер, после оформления заказа, распечатывает товарный чек, который формируется автоматически (см. рис. 3.18).

Фирма: АвтоДоктор

Дата «16» февраля 2017 г.

### ТОВАРНЫЙ ЧЕК №3

Наименование товара	Количество	Цена	Сумма
Услуга: Регулировка угла опережения зажигания	1	700,00	700,00

Всего: 700 семьсот рублей 00 копеек

Подпись продавца: \_\_\_\_\_

Рис. 3.18. Товарный чек

На страницах «Услуги», «Клиенты», «Заказы» можно просмотреть информацию обо всех клиентах, заказах и услугах, добавить новые данные, изменить информацию, найти необходимую информацию.

### 3.2.3 Тестирования совместимости;

Оценка того, насколько хорошо работает программное обеспечение, в определенном браузере, помогло тестирование совместимости.

Протестировали сайт в различных браузерах (см. рис. 3.8, рис. 3.8, рис.3.10), убедились, что картинки корректно отображаются в разных браузерах. Убедились, что шрифты отображаются верно, в разных браузерах.

Также убедились, что java script код в работает в разных браузерах.

### 3.2.4 Тестирования баз данных.

Благодаря тестированию базы данных было проверено, что данные, которые отображаются в web-приложении, совпадают с данными, хранящимися в базе данных.

Проверили название базы данных, таблицы, первичные и внешние ключи, процедуры ввода текстовых данных. Убедились, что данные правильно сохраняются в базе данных после каждого ввода.

С помощью тестирования безопасности было проверено на поиск недостатков и пробелов с точки зрения безопасности приложения.

В результате испытаний автоматизированной системы, убедились, что получилась система, удовлетворяет всем требованиям.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была создана автоматизированная система сервисного обслуживания автомобилей, в виде полнофункционального web - приложения, полностью готового к применению.

Данное приложение ориентировано на менеджеров сервисного обслуживания автомобилей. Будет сокращена избыточность хранимых данных. С его помощью повысится эффективность работы сервисного центра обслуживания автомобилей, позволит избавиться от многих рутинных процессов, что обеспечит наиболее быстрое, полное и качественное обслуживание клиентов. Увеличится достоверность информации и скорость обработки информации.

Был произведен обзор и анализ деятельности по учету сервисного обслуживания автомобилей, а также существующих систем в этой области. Сформулированы требования. Спроектирована программная реализация автоматизированной системы. Проведены испытания.

Разработанное web - приложение удовлетворяет всем требованиям, поставленным на этапе постановки задачи.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению
2. ГОСТ 19.701-90 - Единая система программной документации - Схемы алгоритмов, программ, данных и систем - Условные обозначения и правила выполнения
3. Волгин, В.В. Автосервис. Маркетинг и анализ: практическое пособие / В.В. Волгин. – 2-е изд. – М.: Издательско-торговая корпорация «Дашков и К<sup>о</sup>», 2005. – 496 с.
4. Волгин, В.В. Автосервис. Создание и сертификация: практическое пособие / В.В. Волгин. – 2-е изд. – М.: Издательско-торговая корпорация «Дашков и К<sup>о</sup>», 2005. – 496 с.
5. Волгин, В.В. Автосервис. Производство и менеджмент: практическое пособие / В.В. Волгин. – 2-е изд. – М.: Издательско-торговая корпорация «Дашков и К<sup>о</sup>», 2005. – 520 с.
6. Проектирование предприятий автомобильного транспорта: учебник для студентов специальности «Техническая эксплуатация автомобилей» учреждений, обеспечивающих получение высшего Образования / М.М. Болбас [и др.]; под ред. М.М. Болбаса. – Минск: Адукацыя і выхаванне, 2004. – 528 с.
7. Туревский, И.С. Техническое обслуживание и ремонт автомобильного транспорта: учебное пособие / И.С. Туревский. – М.: ФОРУМ; ИНФРА-М, 2005. – 256 с.
8. Дейт К. Руководство по реляционной СУБД DB2. - М.: Финансы и статистика, 1988. - 320 с.
9. Когаловский М.Р. "Энциклопедия технологий баз данных" / М.Р. Когаловский. - М.: Финансы и статистика, 2002. - 800 с.

10. Рауль Ф. Чон. Начало работы с DB2 Express 9.7. Москва, 2010.- 269 с.
11. Никсон Р. Learning PHP, MySQL, JavaScript, CSS & HTML: A Step-by-Step Guide Dynamic Websites. Четвертое издание, Питер, 2016 – 768с.
12. Макфарланд Д. CSS: The Missing Manul. – Питер, 2016 – 720 с.
13. Прохоренок Н., Дронов В. HTML, JavaScript, PHP и MySQL. Джентельменский набор web- мастера. Четвертое издание. СПб.: БХВ-Петербург, 2015 – 766 с.
14. Прохоренок Н. HTML, JavaScript, PHP и MySQL. Джентельменский набор web- мастера. Четвертое издание. СПб.: БХВ-Петербург, 2010 – 900 с.
15. Бутко В. Р., Дерябкин В. П. CASE - технологии моделирования и проектирования АИС - Учебное пособие. - Самара: Изд-во Самарск. Гос. Экон. академ., 2001.-105 с.
16. Дунаев В.В. Базы данных. Язык SQL, 2006. 289с.
17. <http://htmlbook.ru/> - Самоучитель по HTML и CSS; PHP и MySQL.
18. Карпова Т.С. Базы данных. Модели, разработка, реализация/СПб.: Питер, 2002. - 304 с.
19. Web Database Application with PHP and MySQL, 2ndEdition By David Lane, Hugh E. Williams. O'Reilly, May 2004. ISB: 0-596-00543-1

## ПРИЛОЖЕНИЕ

contract.twig

```

<!doctype html>
<html lang="RU">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
scale=1.0">
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Товарный чек №{{ order.order_id }}</title>
  <link rel="stylesheet" href="/vendors/normalize-css/normalize.css">
  <link rel="stylesheet" href="/assets/css/contract.css">
</head>
<body>
  <div class="page">
    <div class="company-name">Фирма: {{ contract.company_name }}</div>
    <div class="order-date">
      Дата
      &laquo;{{ order.order_date_add_array.day }}&raquo;
      {{ order.order_date_add_array.month }}
      {{ order.order_date_add_array.year }} г.
    </div>
    <h1>Товарный чек №{{ order.order_id }}</h1>
    <table class="table">
      <thead>
        <tr>
          <th>Наименование товара</th>
          <th>Количество</th>
          <th>Цена</th>
          <th>Сумма</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Услуга: {{ order.service_name }}</td>
          <td>1</td>
          <td>{{ order.order_service_price | number_format(2, ',', ' ') }}</td>
          <td>{{ order.order_service_price | number_format(2, ',', ' ') }}</td>
        </tr>
        {% for detail in order.order_details %}
          <tr>
            <td>{{ detail.name }}</td>
            <td>{{ detail.quantity | number_format(2, ',', ' ') }} {{ detail.unit }}</td>
            <td>{{ detail.price | number_format(2, ',', ' ') }}</td>
            <td>{{ (detail.quantity * detail.price) | number_format(2, ',', ' ') }}</td>
          </tr>
        {% endfor %}
      </tbody>
    </table>
    <div class="order-total">Всего: {{ order.order_total }} {{ order.order_total_string }}</div>
    <div class="order-signature">
      <span>Подпись продавца:</span>
      <span></span>
    </div>
    <button type="button" onclick="print();">

```

```

        <svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">
          <path d="M19 8H5c-1.66 0-3 1.34-3 3v6h4v4h12v-4h4v-6c0-1.66-1.34-3-3-3zm-3 11H8v-5h8v5zm3-7c-.55 0-1-.45-1-1s.45-1 1-1 1-.45 1-1 1-.45 1-1 1zm-1-9H6v4h12V3z"/>
        </svg>
        Распечатать
      </button>
    </div>
  </body>
</html>

```

clients.twig

```

{% include 'common/header.twig' %}
{% include 'common/navbar.twig' %}

<div class="container">
  <div class="row">
    <div class="col-sm-8 col-md-9">
      <div class="page-header">
        <a href="/clients/add" class="btn btn-success pull-right">
          <i class="fa fa-plus"></i>
          Добавить
        </a>
        <h1>{{ page.page_title }}</h1>
      </div>

      <div class="panel panel-default">
        <div class="panel-heading">
          <div class="panel-title">Активные клиенты</div>
        </div>
        {% if clients %}
          <div class="table-responsive">
            <table class="table table-striped">
              <thead>
                <tr>
                  <th>Имя клиента</th>
                  <th>Email</th>
                  <th>Телефон</th>
                  <th class="text-right">Управление</th>
                </tr>
              </thead>
              <tbody>
                {% for client in clients %}
                  <tr>
                    <td>{{ client.client_name }}</td>
                    <td>
                      {% if client.client_email %}
                        <a href="mailto:{{ client.client_email }}">{{ client.client_email }}</a>
                      {% else %}
                        &mdash;
                      {% endif %}
                    </td>
                    <td>
                      {% if client.client_phone %}
                        <a href="tel:{{ client.client_phone }}">{{ client.client_phone }}</a>
                      {% else %}
                        &mdash;
                      {% endif %}
                    </td>
                    <td class="text-right">
                      <a href="/clients/edit/{{ client.client_id }}"
                        class="btn btn-primary btn-xs">Изменить</a>
                      <a href="/clients/remove/{{ client.client_id }}"
                        data-client-name="{{ client.client_name }}"

```

```

        class="btn btn-danger btn-xs js-client-remove">Удалить</a>
    </td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
{% else %}
<div class="panel-body">
    <div class="alert alert-info margin-none">Активных клиентов не найдено</div>
</div>
{% endif %}
</div>

<div class="panel panel-default">
<div class="panel-heading">
    <div class="panel-title">Удаленные клиенты</div>
</div>
{% if clients_removed %}
<div class="table-responsive">
<table class="table table-striped">
<thead>
<tr>
<th>Имя клиента</th>
<th>Email</th>
<th>Телефон</th>
<th class="text-right">Управление</th>
</tr>
</thead>
<tbody>
{% for client in clients_removed %}
<tr>
<td>{{ client.client_name }}</td>
<td>
        {% if client.client_email %}
        <a href="mailto:{{ client.client_email }}">{{ client.client_email }}</a>
        {% else %}
        &mdash;
        {% endif %}
</td>
<td>
        {% if client.client_phone %}
        <a href="tel:{{ client.client_phone }}">{{ client.client_phone }}</a>
        {% else %}
        &mdash;
        {% endif %}
</td>
<td class="text-right">
        <a href="/clients/restore/{{ client.client_id }}"
        data-client-name="{{ client.client_name }}"
        class="btn btn-success btn-xs js-client-restore">Восстановить</a>
</td>
</tr>
</tbody>
</table>
</div>
{% else %}
<div class="panel-body">
    <div class="alert alert-info margin-none">Удаленных клиентов не найдено</div>
</div>
{% endif %}

```

```

    </div>
  </div>
  <div class="col-sm-4 col-md-3">
    {% include 'common/sidebar.twig' %}
  </div>
</div>
</div>

```

```
{% include 'common/footer.twig' %}
```

form.twig

```
{% include 'common/header.twig' %}
{% include 'common/navbar.twig' %}
```

```

<div class="container">
  <div class="row">
    <div class="col-sm-8 col-md-9">
      <div class="page-header">
        <h1>{{ page.page_title }}</h1>
      </div>

      {% if form %}
        {% include form %}
      {% else %}
        <div class="alert alert-info">Форма не указана</div>
      {% endif %}
    </div>
    <div class="col-sm-4 col-md-3">
      {% include 'common/sidebar.twig' %}
    </div>
  </div>
</div>

```

```
{% include 'common/footer.twig' %}
```

message.twig

```
{% include 'common/header.twig' %}
{% include 'common/navbar.twig' %}
```

```

<div class="container">
  <div class="row">
    {% if sidebar %}
      <div class="col-sm-8 col-md-9">
        <div class="page-header">
          <h1>{{ page.page_title }}</h1>
        </div>
        <div class="alert alert-{{ message.type }}">{{ message.text }}</div>
        {% if link %} <hr>
          <a href="{{ link.href }}" class="btn btn-default">{{ link.title }}</a>
        {% endif %}
      </div>
      <div class="col-sm-4 col-md-3">
        {% include 'common/sidebar.twig' %}
      </div>
    {% else %}
      <div class="col-xs-12">
        <div class="page-header">
          <h1>{{ page.page_title }}</h1>
        </div>
        <div class="alert alert-{{ message.type }}">{{ message.text }}</div>
        {% if link %} <hr>
          <a href="{{ link.href }}" class="btn btn-default">{{ link.title }}</a>
        {% endif %}
      </div>
    </div>
  </div>

```

```

    {% endif %}
  </div>
</div>

{% include 'common/footer.twig' %}

{% include 'common/header.twig' %}
{% include 'common/navbar.twig' %}

<div class="container">
  <div class="row">
    <div class="col-sm-8 col-md-9">
      <div class="page-header">
        <a href="/orders/add" class="btn btn-success pull-right">
          <i class="fa fa-plus"></i>
          Добавить
        </a>
        <h1>{{ page.page_title }}</h1>
      </div>

      <div class="row">
        <div class="col-xs-12">
          <div class="filter">
            <span class="filter-label">Статус:</span>
            {% if current_status.slug == 'all' %}
              <a href="/orders" class="btn btn-primary btn-xs">Все</a>
            {% else %}
              <a href="/orders" class="btn btn-default btn-xs">Все</a>
            {% endif %}
            {% for status in statuses %}
              {% if current_status.slug == status.slug %}
                <a href="/orders/status/{{ status.slug }}"
                  class="btn btn-primary btn-xs">{{ status.name }}</a>
              {% else %}
                <a href="/orders/status/{{ status.slug }}"
                  class="btn btn-default btn-xs">{{ status.name }}</a>
              {% endif %}
            {% endfor %}
          </div>
        </div>
      </div>

      {% if orders %}
        <div class="row orders-list">
          {% for order in orders %}
            <div class="col-md-6">
              <div class="panel panel-default">
                <div class="panel-heading">
                  <h2 class="panel-title">Заказ №{{ order.order_id }}</h2>
                <div class="order-total">
                  {{ order.order_total | number_format(2, ',', ' ') }}
                  <i class="fa fa-rub"></i>
                </div>
              </div>
              <div class="table-responsive">
                <table class="table table-bordered">
                  <tbody>
                    <tr>
                      <th>Статус:</th>
                      <td>
                        <div class="label label-{{ order.status.type }}">

```

orders.twig



```

        {{ order.status.name }}
    </div>
</td>
</tr>
<tr>
<th>Услуга:</th>
<td>
    <a href="/services/edit/{{ order.service_id }}">{{ order.service_name }}</a>
    <div class="text-muted">
        (стд. цена = {{ order.service_price | number_format(2, ',', ' ') }}
        <i class="fa fa-rub"></i>)
    </div>
</td>
</tr>
<tr>
<th>Клиент:</th>
<td>
    <ul class="list-unstyled margin-none">
        <li>
            <a href="/clients/edit/{{ order.client_id }}">{{ order.client_name }}</a>
        </li>
        {% if order.client_phone %}
        <li>
            тел.: <a
                href="tel:{{ order.client_phone }}">{{ order.client_phone }}</a>
        </li>
        {% endif %}
        {% if order.client_email %}
        <li>
            email: <a
                href="mailto:{{ order.client_email }}">{{ order.client_email }}</a>
        </li>
        {% endif %}
    </ul>
</td>
</tr>
<tr>
<th>Цена работ:</th>
<td>
    {{ order.order_service_price | number_format(2, ',', ' ') }} <i class="fa fa-rub"></i>
</td>
</tr>
{% if order.order_details %}
<tr>
<th>Доп. детали:</th>
<td class="td-details">
    <ol class="list-group list-group-details margin-none">
        {% for detail in order.order_details %}
        <li class="list-group-item">
            {{ detail.name }}
            ({{ detail.price }} <i class="fa fa-rub"></i>)
            *
            {{ detail.quantity }} {{ detail.unit }}
            =
            {{ (detail.quantity * detail.price) | number_format(2, ',', ' ') }}
            <i class="fa fa-rub"></i>
        </li>
        {% endfor %}
    </ol>
</td>
</tr>
{% endif %}

```

```

        {% if order.order_date_add_timestamp > 0 %}
        <tr>
            <th>Добавлен:</th>
            <td>{{ order.order_date_add_formatted }}</td>
        </tr>
        {% endif %}
        {% if order.order_date_start_timestamp > 0 %}
        <tr>
            <th>Начало работ:</th>
            <td>{{ order.order_date_start_formatted }}</td>
        </tr>
        {% endif %}
        {% if order.order_date_finish_timestamp > 0 %}
        <tr>
            <th>Конец работ:</th>
            <td>{{ order.order_date_finish_formatted }}</td>
        </tr>
        {% endif %}
    </tbody>
</table>
</div>
<div class="panel-footer text-right">
    <a href="/orders/contract/{{ order.order_id }}"
        target="_blank"
        class="btn btn-success btn-xs">Товарный чек</a>
    <a href="/orders/edit/{{ order.order_id }}"
        class="btn btn-primary btn-xs">Изменить заказ</a>
</div>
</div>
</div>
</div>
    {% endfor %}
</div>
    {% else %}
    <div class="alert alert-info margin-none">Заказов со статусом "{{ current_status.name }}" нет</div>
    {% endif %}

</div>
<div class="col-sm-4 col-md-3">
    {% include 'common/sidebar.twig' %}
</div>
</div>
</div>

{% include 'common/footer.twig' %}

{% include 'common/header.twig' %}
{% include 'common/navbar.twig' %}

<div class="container">
    <div class="row">
        <div class="col-sm-8 col-md-9">
            <div class="page-header">
                <h1>{{ page.page_title }}</h1>
            </div>

            {% if slider %}
            <div id="slider" class="carousel slide" data-ride="carousel">
                <ol class="carousel-indicators">
                    {% for slide_index, slide_image in slider %}
                    {% if slide_index == 0 %}
                    <li data-target="#slider" data-slide-to="{{ slide_index }}" class="active"></li>
                    {% else %}

```

```

        <li data-target="#slider" data-slide-to="{{ slide_index }}"></li>
    {% endif %}
{% endfor %}
</ol>
<div class="carousel-inner" role="listbox">
    {% for slide_index, slide_image in slider %}
        {% if slide_index == 0 %}
            <div class="item active">
                
            </div>
        {% else %}
            <div class="item">
                
            </div>
        {% endif %}
    {% endfor %}
</div>
<a class="left carousel-control" href="#slider" role="button" data-slide="prev">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
</a>
<a class="right carousel-control" href="#slider" role="button" data-slide="next">
    <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
</a>
</div>
{% endif %}

    {{ page.page_content | raw }}
</div>
<div class="col-sm-4 col-md-3">
    {% include 'common/sidebar.twig' %}
</div>
</div>
</div>

{% include 'common/footer.twig' %}

{% include 'common/header.twig' %}
{% include 'common/navbar.twig' %}

<div class="container">
    <div class="row">
        <div class="col-sm-8 col-md-9">
            <div class="page-header">
                {% if admin.logged %}
                    <a href="/services/add" class="btn btn-success pull-right">
                        <i class="fa fa-plus"></i>
                        Добавить
                    </a>
                {% endif %}
                <h1>{{ page.page_title }}</h1>
            </div>

            {% if admin.logged %}
                <div class="panel panel-default">
                    <div class="panel-heading">
                        <div class="panel-title">Активные услуги</div>
                    </div>
                    {% if services.active %}
                        <div class="table-responsive">
                            <table class="table table-bordered table-striped">
                                <thead>
                                    <tr>

```

services.twig

```

        <th>Название услуги</th>
        <th class="text-right">Стоимость</th>
        <th class="text-right">Управление</th>
    </tr>
</thead>
<tbody>
{% for service in services.active %}
    <tr>
        <td>{{ service.service_name }}</td>
        <td class="text-right">
            {{ service.service_price | number_format(2, ',', ' ') }}
            <i class="fa fa-rub"></i>
        </td>
        <td class="text-right">
            <a href="/services/edit/{{ service.service_id }}"
                class="btn btn-primary btn-xs">Изменить</a>
            <a href="/services/remove/{{ service.service_id }}"
                data-service-name="{{ service.service_name }}"
                class="btn btn-danger btn-xs js-service-remove">Удалить</a>
        </td>
    </tr>
{% endfor %}
</tbody>
</table>
</div>
{% else %}
    <div class="panel-body">
        <div class="alert alert-info margin-none">Активных услуг не найдено</div>
    </div>
{% endif %}
</div>

<div class="panel panel-default">
    <div class="panel-heading">
        <div class="panel-title">Удаленные услуги</div>
    </div>
    {% if services.disabled %}
        <div class="table-responsive">
            <table class="table table-striped table-bordered">
                <thead>
                    <tr>
                        <th>Название услуги</th>
                        <th class="text-right" width="100">Стоимость</th>
                        <th class="text-right" width="160">Управление</th>
                    </tr>
                </thead>
                <tbody>
                    {% for service in services.disabled %}
                        <tr>
                            <td>{{ service.service_name }}</td>
                            <td class="text-right">
                                {{ service.service_price | number_format(2, ',', ' ') }}
                                <i class="fa fa-rub"></i>
                            </td>
                            <td class="text-right">
                                <a href="/services/restore/{{ service.service_id }}"
                                    data-service-name="{{ service.service_name }}"
                                    class="btn btn-success btn-xs js-service-restore">Восстановить</a>
                            </td>
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </div>

```

```

        </table>
    </div>
    {% else %}
    <div class="panel-body">
        <div class="alert alert-info margin-none">Удаленных услуг не найдено</div>
    </div>
    {% endif %}
</div>
{% else %}
{{ page.page_content | raw }}
<hr>
{% if services.active %}
<div class="table-responsive">
<table class="table table-striped table-bordered">
<thead>
<tr>
<th>Название услуги</th>
<th class="text-right" width="100">Стоимость</th>
</tr>
</thead>
<tbody>
{% for service in services.active %}
<tr>
<td>{{ service.service_name }}</td>
<td class="text-right">
{{ service.service_price | number_format(2, ',', ' ') }}
<i class="fa fa-rub"></i>
</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
{% else %}
<div class="alert alert-info margin-none">Услуг не найдено</div>
{% endif %}
</div>
<div class="col-sm-4 col-md-3">
{% include 'common/sidebar.twig' %}
</div>
</div>
</div>
{% include 'common/footer.twig' %}

```

clients\_add.twig

```

<form action="/clients/save" class="form-horizontal" method="post">
<div class="form-group">
<label for="client-name" class="col-sm-3 control-label">Имя</label>
<div class="col-sm-9">
<input type="text"
class="form-control"
id="client-name"
placeholder="Иванов Петр Иванович"
required="required"
name="client-name">
</div>
</div>
<div class="form-group">
<label for="client-email" class="col-sm-3 control-label">Email</label>

```

```

<div class="col-sm-9">
  <input type="email"
    class="form-control"
    id="client-email"
    placeholder="client@mail.ru"
    name="client-email">
</div>
</div>
<div class="form-group">
  <label for="client-phone" class="col-sm-3 control-label">Телефон</label>
  <div class="col-sm-9">
    <input type="text"
      class="form-control"
      id="client-phone"
      placeholder="+7 (800) 555-35-35"
      name="client-phone">
  </div>
</div>
<div class="form-group">
  <div class="col-sm-9 col-sm-offset-3">
    <button type="submit" class="btn btn-success">Сохранить</button>
    <a href="/clients" class="btn btn-default">Отменить</a>
  </div>
</div>
</form>

```

edit.twig

```

{% if client %}
<form action="/clients/update" class="form-horizontal" method="post">
  <input type="hidden" value="{{ client.client_id }}" name="client-id">
  <div class="form-group">
    <label for="client-name" class="col-sm-3 control-label">Имя</label>
    <div class="col-sm-9">
      <input type="text"
        class="form-control"
        id="client-name"
        placeholder="Иванов Петр Иванович"
        value="{{ client.client_name }}"
        required="required"
        name="client-name">
    </div>
  </div>
  <div class="form-group">
    <label for="client-email" class="col-sm-3 control-label">Email</label>
    <div class="col-sm-9">
      <input type="email"
        class="form-control"
        id="client-email"
        placeholder="client@mail.ru"
        value="{{ client.client_email }}"
        name="client-email">
    </div>
  </div>
  <div class="form-group">
    <label for="client-phone" class="col-sm-3 control-label">Телефон</label>
    <div class="col-sm-9">
      <input type="text"
        class="form-control"
        id="client-phone"
        placeholder="+7 (800) 555-35-35"
        value="{{ client.client_phone }}"
        name="client-phone">
    </div>
  </div>

```

```

</div>
<div class="form-group">
  <div class="col-sm-9 col-sm-offset-3">
    <button type="submit" class="btn btn-success">Сохранить</button>
    <a href="/clients" class="btn btn-default">Отменить</a>
  </div>
</div>
</form>
{% else %}
<div class="alert alert-danger">Услуга не найдена</div>
<hr>
<a href="/clients" class="btn btn-default">Вернуться</a>
{% endif %}

```

orders\_add.twig

```

<form action="/orders/save" class="form-horizontal" method="post" id="form-orders-add">
  <div class="form-group">
    <label for="order-status" class="col-sm-3 control-label">Статус</label>
    <div class="col-sm-9">
      <select name="order-status" id="order-status" class="form-control">
        {% for status in statuses %}
          <option value="{{ status.slug }}">{{ status.name }}</option>
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="form-group">
    <label for="order-service" class="col-sm-3 control-label">Услуга</label>
    <div class="col-sm-9">
      <select name="order-service" id="order-service" class="form-control">
        {% for service in services %}
          <option value="{{ service.service_id }}"
            data-price="{{ service.service_price }}"
            {{ service.service_name }} ({{ service.service_price | number_format(2, ',', ' ') }} руб.)
          </option>
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="form-group">
    <label for="order-service-price" class="col-sm-3 control-label">Своя цена услуги</label>
    <div class="col-sm-9">
      <input type="number"
        id="order-service-price"
        class="form-control"
        name="order-service-price"
        min="0"
        step="0.01"
        value="0">
    </div>
  </div>
  <div class="form-group">
    <label for="order-client" class="col-sm-3 control-label">Клиент</label>
    <div class="col-sm-9">
      <select name="order-client" id="order-client" class="form-control">
        {% for client in clients %}
          <option value="{{ client.client_id }}">{{ client.client_name }}</option>
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="form-group">
    <label for="order-details" class="col-sm-3 control-label">Детали</label>

```

```

<div class="col-sm-9 form-orders-add-details-list-wrapper">
  <div class="panel-group form-orders-add-details-list"
    id="order-details"
    role="tablist"
    aria-multiselectable="false">
    </div>
    <button type="button" class="btn btn-default btn-sm pull-right btn-detail-add js-detail-add">
      <i class="fa fa-plus"></i>
    </button>
  </div>
</div>
<div class="form-group">
  <label for="order-date-add" class="col-sm-3 control-label">Дата добавления услуги</label>
  <div class="col-sm-9">
    <input type="text"
      id="order-date-add"
      class="form-control datetime-picker"
      name="order-date-add"
      value="{{ current_date }}">
  </div>
</div>
<div class="form-group">
  <label for="order-date-start" class="col-sm-3 control-label">Дата начала работ</label>
  <div class="col-sm-9">
    <input type="text"
      id="order-date-start"
      class="form-control datetime-picker"
      name="order-date-start"
      value="">
  </div>
</div>
<div class="form-group">
  <label for="order-date-finish" class="col-sm-3 control-label">Дата окончания работ</label>
  <div class="col-sm-9">
    <input type="text"
      id="order-date-finish"
      class="form-control datetime-picker"
      name="order-date-finish"
      value="">
  </div>
</div>
<div class="form-group">
  <div class="col-sm-9 col-sm-offset-3">
    <button type="submit" class="btn btn-success">Сохранить</button>
    <a href="/orders" class="btn btn-default">Отменить</a>
  </div>
</div>
</form>

<script>
  window.details = [];
</script>

<script id="template-detail" type="text/html">
  <div class="panel panel-default"
    data-index="__index__"
    id="order-detail-__index__-panel">
    <div class="panel-heading" role="tab" id="order-detail-heading-__index__">
      <div class="panel-heading-controls">
        <span class="panel-heading-control text-primary"
          role="button"
          data-toggle="collapse"

```



```

        data-parent="#order-details"
        data-target="#order-detail-__index__"
        aria-expanded="true"
        aria-controls="order-detail-__index__">
        <i class="fa fa-edit"></i>
    </span>
    <span class="panel-heading-control text-danger js-detail-remove" data-index="__index__">
        <i class="fa fa-remove"></i>
    </span>
</div>
<div class="panel-title">
    <div role="button"
        data-toggle="collapse"
        data-parent="#order-details"
        data-target="#order-detail-__index__"
        aria-expanded="true"
        aria-controls="order-detail-__index__">
        __name__ ( __price__ <i class="fa fa-rub"></i>) * __quantity__ __unit__ = __total__ <i class="fa fa-
rub"></i>
    </div>
</div>
</div>
<div id="order-detail-__index__"
    class="panel-collapse collapse"
    role="tabpanel"
    aria-labelledby="order-detail-heading-__index__">
    <div class="panel-body">
        <div class="form-horizontal">
            <div class="form-group">
                <label for="order-detail-__index__-name" class="col-sm-4 control-label">Название</label>
                <div class="col-sm-8">
                    <input type="text"
                        id="order-detail-__index__-name"
                        class="form-control"
                        value="__name__"
                        data-name="name"
                        name="order-details[__index__][name]">
                </div>
            </div>
            <div class="form-group">
                <label for="order-detail-__index__-quantity" class="col-sm-4 control-label">Количество</label>
                <div class="col-sm-8">
                    <input type="number"
                        id="order-detail-__index__-quantity"
                        value="__quantity__"
                        class="form-control"
                        min="0.01"
                        step="0.01"
                        data-name="quantity"
                        name="order-details[__index__][quantity]">
                </div>
            </div>
            <div class="form-group">
                <label for="order-detail-__index__-unit" class="col-sm-4 control-label">Единица
измерения</label>
                <div class="col-sm-8">
                    <input type="text"
                        id="order-detail-__index__-unit"
                        value="__unit__"
                        class="form-control"
                        data-name="unit"
                        name="order-details[__index__][unit]">
                </div>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
  </div>
  <div class="form-group margin-none">
    <label for="order-detail-__index__-price" class="col-sm-4 control-label">Цена за единицу</label>
    <div class="col-sm-8">
      <input type="number"
        id="order-detail-__index__-price"
        value="__price__"
        class="form-control"
        min="0.01"
        step="0.01"
        data-name="price"
        name="order-details[__index__][price]">
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
</script>

```

edit.twig

```

<form action="/orders/update" class="form-horizontal" method="post" id="form-orders-edit">
  <input type="hidden" value="{{ order.order_id }}" name="order-id">
  <div class="hidden" id="details-json">{{ order.order_details | raw }}</div>
  <div class="form-group">
    <label for="order-status" class="col-sm-3 control-label">Статус</label>
    <div class="col-sm-9">
      <select name="order-status" id="order-status" class="form-control">
        {% for status in statuses %}
          {% if order.order_status == status.slug %}
            <option value="{{ status.slug }}" selected="selected">{{ status.name }}</option>
          {% else %}
            <option value="{{ status.slug }}">{{ status.name }}</option>
          {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="form-group">
    <label for="order-service" class="col-sm-3 control-label">Услуга</label>
    <div class="col-sm-9">
      <select name="order-service" id="order-service" class="form-control">
        {% for service in services %}
          {% if order.order_service == service.service_id %}
            <option value="{{ service.service_id }}"
              selected="selected"
              data-price="{{ service.service_price }}">
              {{ service.service_name }} ({{ service.service_price | number_format(2, ',', '') }} руб.)
            </option>
          {% else %}
            <option value="{{ service.service_id }}"
              data-price="{{ service.service_price }}">
              {{ service.service_name }} ({{ service.service_price | number_format(2, ',', '') }} руб.)
            </option>
          {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="form-group">
    <label for="order-service-price" class="col-sm-3 control-label">Своя цена услуги</label>
    <div class="col-sm-9">

```

```

    <input type="number"
      id="order-service-price"
      class="form-control"
      name="order-service-price"
      min="0"
      step="0.01"
      value="{{ order.order_service_price }}">
  </div>
</div>
<div class="form-group">
  <label for="order-client" class="col-sm-3 control-label">Клиент</label>
  <div class="col-sm-9">
    <select name="order-client" id="order-client" class="form-control">
      {% for client in clients %}
        {% if order.order_client == client.client_id %}
          <option value="{{ client.client_id }}" selected="selected">{{ client.client_name
}}</option>
        {% else %}
          <option value="{{ client.client_id }}">{{ client.client_name }}</option>
        {% endif %}
      {% endfor %}
    </select>
  </div>
</div>
<div class="form-group">
  <label for="order-details" class="col-sm-3 control-label">Детали</label>
  <div class="col-sm-9 form-orders-add-details-list-wrapper">
    <div class="panel-group form-orders-add-details-list"
      id="order-details"
      role="tablist"
      aria-multiselectable="false">
    </div>
    <button type="button" class="btn btn-default btn-sm pull-right btn-detail-add js-detail-add">
      <i class="fa fa-plus"></i>
    </button>
  </div>
</div>
<div class="form-group">
  <label for="order-date-add" class="col-sm-3 control-label">Дата добавления услуги</label>
  <div class="col-sm-9">
    <input type="text"
      id="order-date-add"
      class="form-control datetime-picker"
      name="order-date-add"
      value="{{ order.order_date_add_formatted }}">
  </div>
</div>
<div class="form-group">
  <label for="order-date-start" class="col-sm-3 control-label">Дата начала работ</label>
  <div class="col-sm-9">
    <input type="text"
      id="order-date-start"
      class="form-control datetime-picker"
      name="order-date-start"
      value="{{ order.order_date_start_formatted }}">
  </div>
</div>
<div class="form-group">
  <label for="order-date-finish" class="col-sm-3 control-label">Дата окончания работ</label>
  <div class="col-sm-9">
    <input type="text"
      id="order-date-finish"

```

```

        class="form-control datetime-picker"
        name="order-date-finish"
        value="{{ order.order_date_finish_formatted }}">
    </div>
</div>
<div class="form-group">
    <div class="col-sm-9 col-sm-offset-3">
        <button type="submit" class="btn btn-success">Сохранить</button>
        <a href="/orders" class="btn btn-default">Отменить</a>
    </div>
</div>
</form>

<script>
    window.details = JSON.parse(document.getElementById('details-json').innerHTML);
</script>

<script id="template-detail" type="text/html">
    <div class="panel panel-default"
        data-index="__index__"
        id="order-detail-__index__-panel">
        <div class="panel-heading" role="tab" id="order-detail-heading-__index__">
            <div class="panel-heading-controls">
                <span class="panel-heading-control text-primary"
                    role="button"
                    data-toggle="collapse"
                    data-parent="#order-details"
                    data-target="#order-detail-__index__"
                    aria-expanded="true"
                    aria-controls="order-detail-__index__">
                    <i class="fa fa-edit"></i>
                </span>
                <span class="panel-heading-control text-danger js-detail-remove" data-index="__index__">
                    <i class="fa fa-remove"></i>
                </span>
            </div>
            <div class="panel-title">
                <div role="button"
                    data-toggle="collapse"
                    data-parent="#order-details"
                    data-target="#order-detail-__index__"
                    aria-expanded="true"
                    aria-controls="order-detail-__index__">
                    __name__ (__price__ <i class="fa fa-rub"></i>) * __quantity__ __unit__ = __total__ <i
class="fa fa-rub"></i>
                </div>
            </div>
        </div>
        <div id="order-detail-__index__"
            class="panel-collapse collapse"
            role="tabpanel"
            aria-labelledby="order-detail-heading-__index__">
            <div class="panel-body">
                <div class="form-horizontal">
                    <div class="form-group">
                        <label for="order-detail-__index__-name" class="col-sm-4 control-
label">Название</label>
                        <div class="col-sm-8">
                            <input type="text"
                                id="order-detail-__index__-name"
                                class="form-control"
                                value="__name__"

```

```

        data-name="name"
        name="order-details[__index__][name]">
    </div>
</div>
<div class="form-group">
    <label for="order-detail-__index__-quantity" class="col-sm-4 control-
label">Количество</label>
    <div class="col-sm-8">
        <input type="number"
            id="order-detail-__index__-quantity"
            value="__quantity__"
            class="form-control"
            min="0.01"
            step="0.01"
            data-name="quantity"
            name="order-details[__index__][quantity]">
        </div>
    </div>
<div class="form-group">
    <label for="order-detail-__index__-unit" class="col-sm-4 control-label">Единица
измерения</label>
    <div class="col-sm-8">
        <input type="text"
            id="order-detail-__index__-unit"
            value="__unit__"
            class="form-control"
            data-name="unit"
            name="order-details[__index__][unit]">
        </div>
    </div>
<div class="form-group margin-none">
    <label for="order-detail-__index__-price" class="col-sm-4 control-label">Цена за
единицу</label>
    <div class="col-sm-8">
        <input type="number"
            id="order-detail-__index__-price"
            value="__price__"
            class="form-control"
            min="0.01"
            step="0.01"
            data-name="price"
            name="order-details[__index__][price]">
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</script>

```

services\_add.twig

```

<form action="/services/save" class="form-horizontal" method="post">
    <div class="form-group">
        <label for="service-name" class="col-sm-3 control-label">Название услуги</label>
        <div class="col-sm-9">
            <input type="text"
                class="form-control"
                id="service-name"
                placeholder="Замена масла"
                required="required"
                name="service-name">
        </div>
    </div>
</div>

```

```

<div class="form-group">
  <label for="service-price" class="col-sm-3 control-label">Цена услуги</label>
  <div class="col-sm-9">
    <input type="number"
      class="form-control"
      id="service-price"
      min="0"
      required="required"
      value="0"
      name="service-price">
  </div>
</div>
<div class="form-group">
  <div class="col-sm-9 col-sm-offset-3">
    <button type="submit" class="btn btn-success">Сохранить</button>
    <a href="/services" class="btn btn-default">Отменить</a>
  </div>
</div>
</form>

```

edit.twig

```

{% if service %}
<form action="/services/update" class="form-horizontal" method="post">
  <input type="hidden" value="{{ service.service_id }}" name="service-id">
  <div class="form-group">
    <label for="service-name" class="col-sm-3 control-label">Название услуги</label>
    <div class="col-sm-9">
      <input type="text"
        class="form-control"
        id="service-name"
        value="{{ service.service_name }}"
        placeholder="Замена масла"
        required="required"
        name="service-name">
    </div>
  </div>
  <div class="form-group">
    <label for="service-price" class="col-sm-3 control-label">Цена услуги</label>
    <div class="col-sm-9">
      <input type="number"
        class="form-control"
        id="service-price"
        min="0"
        required="required"
        value="{{ service.service_price }}"
        name="service-price">
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-9 col-sm-offset-3">
      <button type="submit" class="btn btn-success">Сохранить</button>
      <a href="/services" class="btn btn-default">Отменить</a>
    </div>
  </div>
</form>
{% else %}
<div class="alert alert-danger">Услуга не найдена</div>
<hr>
<a href="/services" class="btn btn-default">Вернуться</a>
{% endif %}

```