

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,  
к.т.н., доцент каф. ЭВМ  
\_\_\_\_\_ Д.В. Топольский  
«\_\_» \_\_\_\_\_ 2020 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ Д.А. Жернов  
«\_\_» \_\_\_\_\_ 2020 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
«\_\_» \_\_\_\_\_ 2020 г.

Челябинск-2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

## **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Жернову Дмитрию Алексеевичу  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

**Тема работы:** «Разработка Веб-приложения для сервисного центра»  
утверждена приказом по университету от 25 апреля 2020 г. №627

**Срок сдачи студентом законченной работы:** 1 июня 2020 г.

### **Исходные данные к работе:**

- используемые языки: javascript, php, css, php;
- платформа разработки visual studio code;
- необходимо что бы приложение работало с базой данных;
- информация о сервисном центре предоставляется заказчиком;
- корректная работа на разных устройствах;
- удобный интерфейс.

### **Перечень подлежащих разработке вопросов:**

- проведение анализа существующих аналогов;
- выбор средств реализации проекта;
- определение требований и основного функционала системы;
- программная организация и организация базы данных;
- проведения тестирования разработанного приложения.

**Дата выдачи задания:** 1 декабря 2019 г.

Руководитель работы \_\_\_\_\_ /Д.В.Топольский/

Студент \_\_\_\_\_ /Д.А.Жернов/

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор аналогов	01.03.2020	
Определение требований и средств реализации	01.03.2020	
Реализация	01.05.2020	
Тестирование	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	1.06.2020	
Подготовка презентации и доклада	4.06.2020	

Руководитель работы \_\_\_\_\_ /Д.В.Топольский /

Студент \_\_\_\_\_ /Д.А.Жернов /

## Аннотация

Д.А. Жернов Разработка Веб-приложения для сервисного центра – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 59 с., 39 ил., библиогр. список – 14 наим.

В рамках выпускной квалификационной работы выполнены все этапы, связанные с разработкой Веб-приложения для реально существующего сервисного центра. Описаны процессы работы с приложением для работников центра, а также для его клиентов. Рассматриваются и анализируются решения конкурентов с целью выявления положительных и отрицательных качеств для создания максимально удобного и понятного интерфейса. Произведен анализ современных технологий для создания Веб-приложений, а именно: языков программирования (Java Script, PHP), языка разметки (HTML) и языка таблицы стилей (CSS), портативная серверная платформа (Open Server Panel), Веб-интерфейс для администрирования СУБД MySQL (phpMyAdmin) и СУБД MySQL. Также в работе был использован фреймворк ReadBeanPHP для работы с базой данных.

По завершении этапа разработки, выполнено альфа-тестирование разработанного приложения для работы над поставленными задачами. Для оценки результатов работы используются специально разработанные задачи, тестирующие корректность работы приложения во всех возможных вариантах использования. В завершение оцениваются дальнейшие возможности развития созданного приложения.

# ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ .....	6
ВВЕДЕНИЕ.....	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	9
1.1 ОБЗОР АНАЛОГОВ .....	9
1.2 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ .....	14
1.2.1 Выбор среды разработки.....	14
1.2.2 Выбор языка программирования серверной части.....	15
1.2.3 Выбор типа базы данных .....	23
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ .....	28
2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	28
2.1.1 Требования к регистрации и авторизации.....	28
2.1.2 Требования к безопасности.....	29
2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	30
2.2.1 Требования к построению интерфейса системы .....	30
3 ПРОЕКТИРОВАНИЕ.....	31
3.1 АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ.....	31
3.1.1 Интерфейс пользователя .....	32
3.1.2 Сервер API.....	33
3.2 АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ.....	33
3.3 ОПИСАНИЕ ДАННЫХ .....	34
4 РЕАЛИЗАЦИЯ.....	36
4.1 ГЛАВНАЯ СТРАНИЦА ВЕБ-ПРИЛОЖЕНИЯ.....	36
4.2 АВТОРИЗАЦИЯ\РЕГИСТРАЦИЯ ПОЛЬЗОВАТЕЛЯ.....	38
4.3 АДАПТИВНОСТЬ .....	41

4.4	ЛИЧНЫЙ КАБИНЕТ ПОЛЬЗОВАТЕЛЯ .....	44
4.5	СОЗДАНИЕ И ОТОБРАЖЕНИЕ ЗАЯВОК.....	45
5	ТЕСТИРОВАНИЕ .....	47
5.1	ТЕСТИРОВАНИЕ ФОРМ ВВОДА .....	47
5.1.1	Тестирование форм ввода авторизации пользователя .....	<b>Error!</b>
<b>Bookmark not defined.</b>		
5.1.2	Тестирование формы ввода регистрации пользователя .....	49
5.1.3	Тестирование формы добавления заявки .....	54
5.1.4	Тестирование безопасности .....	56
6	ЗАКЛЮЧЕНИЕ .....	57
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	58

## **ВВЕДЕНИЕ**

На данный момент у заказчика есть страница в социальной сети «ВКонтакте» и локальная база данных на одном компьютере без поиска и с некоторыми ограничениями, так как она была взята как готовое решение, а не под заказ. Так как количество клиентов уже больше, чем 3 тысячи, поиск нужного клиента по базе занимает много времени, в случае если клиент потерял акт приемки, на котором указан его номер в базе данных. Также компания занимается скупкой и продажей бывшей в употреблении техники. Информация по ней хранится в тетрадке, что является неудобным решением

Целью представленной выпускной квалификационной работы является разработка Веб-приложения, обеспечивающего систему взаимодействия клиентов и работников с использованием современных технологий разработки.

Для достижения поставленной цели, необходимо решить следующие задачи:

1. Провести анализ существующих аналогов с точки зрения их достоинств и недостатков посредством самостоятельного и просмотра отзывов пользователей.
2. Определить основные задачи, которые должно выполнять приложение для обеспечения нужного функционала.
3. Выбор средств реализации проекта.
4. Программная организация и организация базы данных.
5. Провести тестирование разработанного приложения.



# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 ОБЗОР АНАЛОГОВ

В сети интернет можно найти множество сервисных центров по всему городу. В основном это группы во Вконтакте или «сайт-визитка», на котором можно прочитать основную информацию о сервисном центре. Мною не было найдено ни одного Веб сервиса с возможностью наблюдать за статусом ремонта без необходимости звонить в сервис. Таким образом можно заключить что большая часть решений состоит в том, чтобы дать потенциальному клиенту только основную информацию, адрес и примерные цены за ремонт, и ничего более.

Хотя количество сервисных центров на рынке действительно большое разнообразие, но все их можно свести к следующим категориям.

### 1) Сайт-визитка

В качестве примера берется сервисный центр «Сервис-К». [1]

Из плюсов можно отметить

- такое решение не требует больших затрат;
- можно сверстать без посторонней помощи, без знаний в области веб-программирования, с помощью конструкторов, как платных, так и бесплатных, которые можно найти в сети интернет;
- содержит только основную информацию о сервисном центре (описание и контакты).

Из минусов данного сайта можно выделить:

- Отсутствует удобный прейскурант, на главной странице можно увидеть блок, в который включены только 6 самых популярных работ, и все они связаны с программными проблемами. Если проблема пользователя не в списке, то узнать примерную стоимость можно только связавшись со специалистом;

### Популярные работы

Все подробности уточняйте у наших специалистов

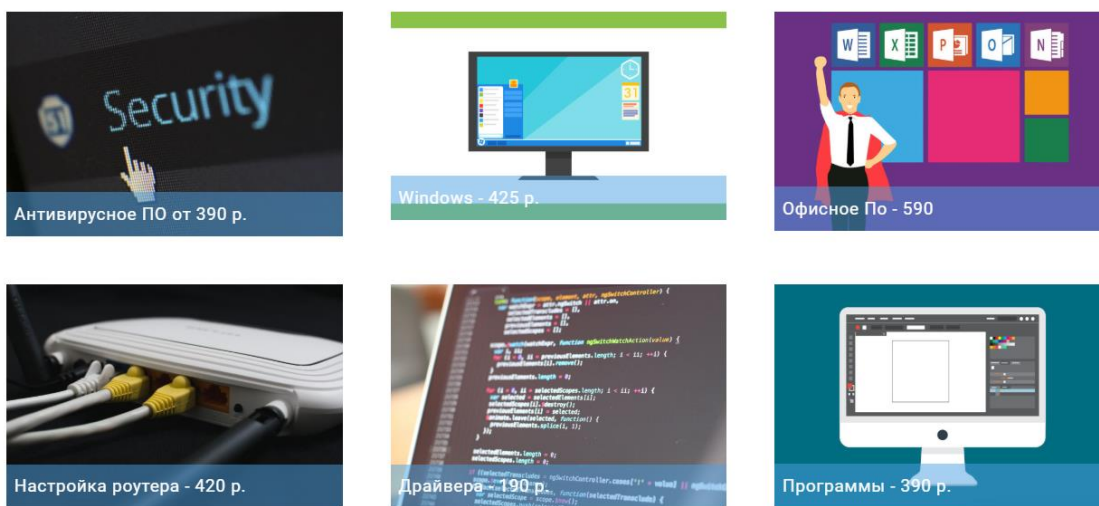


Рисунок 1 – Прейскурант Сервиса-К

- Блок отзывов не вызывает доверия, так- как не понятно с какого ресурса берутся последние отзывы. Для проверки в браузере можно открыть код страницы и обнаружить, что отзывы не динамические, а написаны сразу на HTML, что при дальнейшем разборе только подтверждается, так как у сайта отсутствуют загружаемые JavaScript файлы, которые могли бы

отвечать за парсинг отзывов с таких ресурсов как [chelyabinsk.flamp.ru](http://chelyabinsk.flamp.ru) или [2gis.ru/chelyabinsk](http://2gis.ru/chelyabinsk).

## Последние отзывы наших клиентов

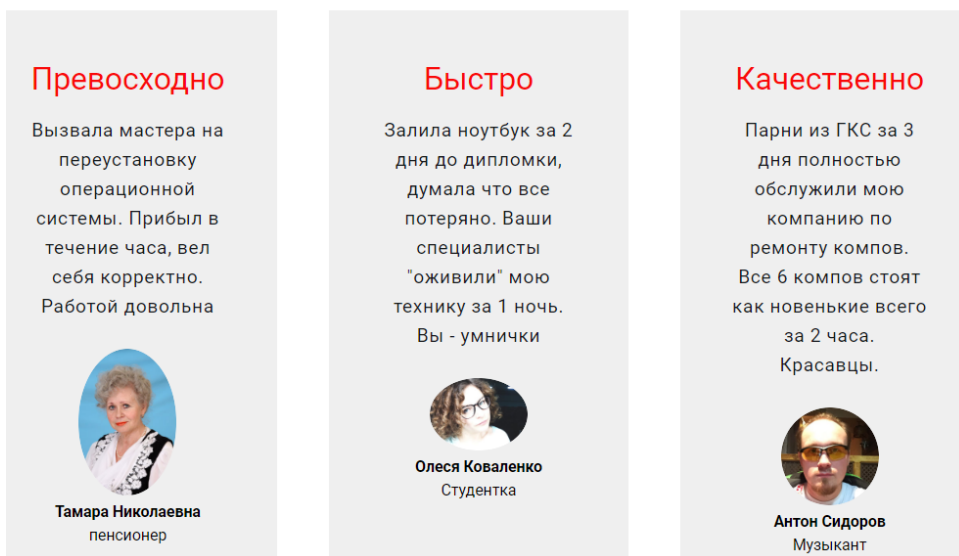


Рисунок 2 – Отзывы о компании «Сервис-К»

```
        Вызвала мастера на переустановку операционной системы.
        Прибыл в течение часа, вел себя корректно. Работой довольна"
    </p>
    <div class="card-img">...</div>
    <div class="signature">...</div>
</div>
</div>
<div class="card mbr-col-sm-12 mbr-col-md-6 mbr-col-lg-4">
  <div class="card-wrapper">
    <h3 class="mbr-fonts-style card-title align-center display-5">Быстро</h3>
    <p class="mbr-text mbr-fonts-style align-center display-7">
      "
      Залила ноутбук за 2 дня до дипломки, думала что все
      потеряно. Ваши специалисты "оживили" мою технику за 1 ночь. Вы - умнички"
    </p>
    <div class="card-img">...</div>
    <div class="signature">...</div>
  </div>
</div>
<div class="card mbr-col-sm-12 mbr-col-md-6 mbr-col-lg-4">
  <div class="card-wrapper">
    <h3 class="mbr-fonts-style card-title align-center display-5">Качественно</h3>
    <p class="mbr-text mbr-fonts-style align-center display-7">
      "Парни из ГКС за 3 дня полностью обслужили мою компанию по ремонту компов. Все 6
      компов стоят как новенькие всего за 2 часа. Красавцы."
```

Рисунок 3 – Исходный код блока с отзывами

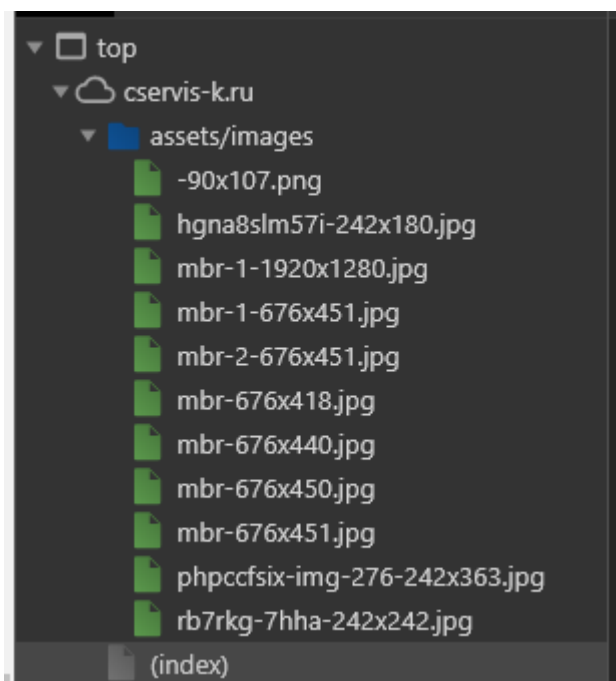


Рисунок 4 – Подключаемые на странице файлы

Из всего вышеперечисленного можно сделать вывод, что такое решение подходит развивающимся компаниям, но стоит уделить больше внимания клиентоориентированности. Из хороших примеров можно выделить сеть сервисных центров по ремонту телефонов «Pedant».

## 2)Страница на ресурсе «Вконтакте»

Для примера возьмем страницу компании «ITService».[2]

По своей структуре страница напоминает сайт визитку. Так же отображается только основная информация: Описание, адрес, номер для связи.

Соответственно к плюсам можно отнести:

- простой функционал;
- понятный интерфейс;
- не требует затрат;
- хорошо подходит для маленьких компаний;
- возможность обратной связи через комментарии.

Из минусов можно выделить следующее:

- строго ограниченный функционал;
- нет кастомизации отдельных элементов;
- возможность рекламы только внутри социальной сети;
- нет возможности добавить своих элементов.

Хорошее решение для маленьких компаний, так как не требует никаких затрат и умений, а также простой функционал и понятный интерфейс. Так же подходит для больших компаний, но ориентирован только на рекламу в социальной сети, а не как основное решение.

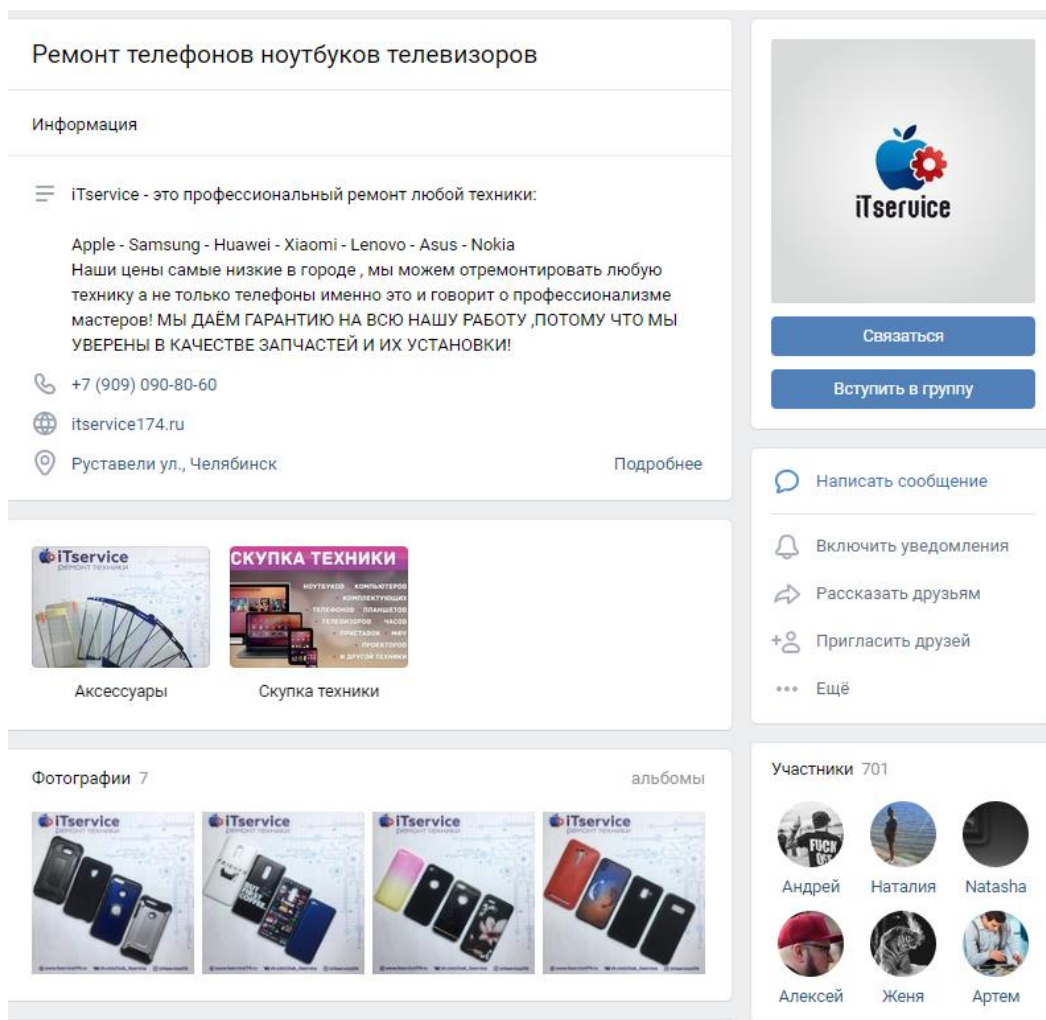


Рисунок 5 – Страница «ITService» в социальной сети «ВКонтакте»

По итогу ничего из этого не подходит для решения нашей проблемы, нужно Веб-приложение с большой структурой и широким функционалом, которое будет в себя включать и работу с базой данных и работу с клиентами.

## **1.2 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ**

### **1.2.1 Выбор среды разработки**

Так как для Веб-разработки не нужно компилировать приложение для его тестирования, а все что нужно это парочка разных браузеров, то для написания приложения будет достаточно обычного блокнота. Остановим свой выбор на Visual Studio Code.

**Visual Studio Code** – «легкий» редактор исходного кода для кроссплатформенной разработки Веб-приложений от компании Microsoft. Visual Studio Code имеет огромное количество инструментов для отладки, и анализа приложений и является свободно распространяемым. Включает в себя такие инструменты [3]:

- Отладчик;
- Инструменты для работы с Git;
- Подсветку синтаксиса, IntelSense;
- Средства для рефакторинга;
- Имеется большой выбор дополнительных плагинов.

## 1.2.2 Выбор языка программирования серверной части

**PHP — скриптовый язык общего назначения, часто применяемый для разработки Веб-сайтов.**

PHP — интерпретируемый язык программирования, позволяющий создавать программы в процедурном и объектно-ориентированном стиле. PHP — хороший язык для разработки проектов на CMS. И с технической точки зрения современный PHP весьма неплох. Но экосистема языка и особенности кадрового рынка делают PHP не самым лучшим выбором для разработки сложных проектов [4][5].

### **Достоинства**

1. Ориентация на Веб-разработку. PHP создавался, развивался и поддерживается как язык для создания Веб-сайтов. Многие конструкции и решения в нем созданы для удобства работы в Веб-среде.
2. Кроссплатформенность. PHP перенесен на все основные операционные системы: можно разрабатывать сайт в Windows, Mac OS X, а эксплуатировать на Linux-сервере. Сложности переноса будут минимальны и нивелироваться языком.
3. Бесплатность. PHP является разработкой из мира свободного программного обеспечения, не потребуется платить ни за сам язык, ни за большинство сопутствующих программ (редакторы, Веб-серверы, базы данных). Вдобавок большинство программных продуктов, с которыми придется иметь дело, будут иметь доступный для изучения и модификации исходный код. Вложения могут потребоваться при аренде доменного имени и сервера для публикации сайта в Интернете. Однако изучать PHP можно, не вкладывая ни копейки.

4. Универсальность PHP делает его популярным среди хостинг-провайдеров. Любой надежный провайдер поддерживает PHP и предоставляет хостинговые услуги для сайтов на нем. Как правило, хостинговые пакеты услуг предоставляют поддержку PHP без дополнительной оплаты включая либо бесплатные тарифные планы, либо дешевые с неограниченным объемом и бесплатным доменным именем;
5. PHP изначально создавался для разработки динамических Веб-страниц, поэтому его скрипты справляются с этой задачей лучше, чем другие языки программирования. Его код легко встраивается в HTML, что позволяет легко конвертировать статические страницы в динамические. По сути, это лучший язык для создания полноценных сайтов на основе HTML-стандарта.

### **Недостатки**

1. Отсутствие лидера — многие технологии и языки имеют лидера, архитектора, который определяет облик технологии, задает вектор развития, принимает решение о том, что должно быть обязательно, а чего не будет никогда (Linux, Python, Ruby и т. п). В PHP лидера нет, многие решения и конструкции — это компромисс заинтересованных групп и исторически сложившихся реалий.
2. Непоследовательный синтаксис — при изучении языка PHP, особенно старой части, основанной на функциях, можно заметить, что часть функций имеет префиксы array\_, str\_, часть не имеет. Параметры функций могут быть расположены не совсем логично и не так, как в другой функции этой же группы.
3. PHP уже довольно долго живущий язык. Когда язык только появляется, он довольно элегантный и внутренне согласованный. По мере жизненного



цикла язык обрастает дополнительными ключевыми словами, артефактами, устаревшими конструкциями, которые вроде есть, работают, но которыми не рекомендуется пользоваться. У РНР была довольно бурная молодость, в ходе которой была отменена масса директив и приемов, которые на первый взгляд должны были облегчать разработку, а на практике оборачивались серьезными проблемами безопасности. Сам РНР, стартовавший как неobjектно-ориентированный язык, в настоящий момент стал полноценным объектно-ориентированным языком.

## **Ruby**

В противовес машинно-ориентированным языкам, программы на которых пишутся так, чтобы компьютеру было максимально удобно их выполнять, Ruby — язык, ориентированный на удобство человека: он позволяет разработчику максимально быстро и просто выполнить задачу, хотя, возможно, это и потребует дополнительного времени работы компьютера. В современном мире вычислительные ресурсы стоят значительно дешевле, нежели время разработчиков, поэтому этот подход в большинстве случаев не только более комфортный для разработчиков, но и экономически более целесообразный с точки зрения бизнеса [6].

### **Достоинства**

- Ruby достаточно прост в освоении;
- ООП в стиле Smalltalk;
- поддержку комьюнити (сообщества программистов);
- простой и удобный синтаксис;
- большую зарплату;
- поддержку новейших перспективных подходов в программировании;
- большое количество готовых решений.

## **Недостатки**

- плохая поддержка для Windows. Так как Ruby разрабатывался, в основном, для Linux, то не в Unix-подобной системе будет сложнее работать с IDE, устанавливать дополнительные библиотеки и т.д;
- мало документации на русском;
- потребление памяти.

## **Python**

Python — высокоуровневый интерпретируемый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода [7][8].

### Достоинства

- низкий порог вхождения. синтаксис python более понятный для новичка.
- логичный, лаконичный и понятный. в сравнении с многими другими языками python имеет легкочитаемый синтаксис;
- кроссплатформенный: подходит для разных платформ: и linux, и windows;
- есть реализация интерпретаторов для мобильных устройств и непопулярных систем;
- широкое применение. используется для разработки веб-приложений, игр, удобен для автоматизации, математических вычислений, машинного обучения, в области интернета вещей. существует реализация под названием micro python, оптимизированная для запуска на микроконтроллерах;
- мощная поддержка компаний-гигантов it-индустрии. такие компании, как google, facebook, dropbox, spotify, quora, netflix, на определенных этапах разработки использовали именно python;
- высокая востребованность на рынке труда;

- в мире python много качественных библиотек.

#### Недостатки

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

### **JavaScript**

#### Преимущества [9]:

- javascript предоставляет большое количество возможностей для решения самых разнообразных задач. гибкость языка позволяет использовать множество шаблонов программирования применительно к конкретным условиям;
- популярность javascript открывает перед программистом немалое количество готовых библиотек, которые позволяют значительно упростить написание кода и нивелировать несовершенства синтаксиса;
- применение во многих областях. широкие возможности javascript дают программистам шанс попробовать себя в качестве разработчика самых разнообразных приложений.

#### Недостатки:

- язык компилируется в момент исполнения кода. каждый раз, когда вы открываете сайт, javascript код начинает компилироваться. как минимум увеличивается время выполнения программы;
- отсутствует типизация данных. проблема всех скриптовых языков. пока выполнение кода не дойдет до нужной строчки, не узнаешь работает ли она.

Существует еще язык C#, но мы не будем его даже рассматривать так как он ориентирован преимущественно на Windows платформу, а для нас важна мультиплатформенность.

Ruby – написан для людей, а не для машины из-за чего страдает скорость работы, а так же крайне мало документации на русском.

PHP и Python хорошие языки для написания серверной части приложения но остановимся мы все же на PHP так как его скорость при работе с проектами простой логики, такими как сервисный центр, крайне высока, так же будем использовать JavaScript для работы с анимацией и переходами на странице.

Немаловажным фактом будет являться то, что разработчик проекта уже знаком с PHP и JavaScript, а также необходимыми, для реализации приложения, программами, основанными на этом языке ведь для разработки нам, понадобится серверная платформа OpenServer, у которой сразу есть поддержка PHP.

## **Выбор серверной платформы**

### **OpenServer**

OpenServer – это портативная серверная платформа и программная среда, созданная специально для Веб-разработчиков для запуска и тестирования Веб-приложения без необходимости пользоваться интернет хостингами и установки дополнительного софта. Все происходит на вашем персональном компьютере [11].

В состав уже входят:

- Apache версии 2.4.43;
- MySQL версии 8.0.19;
- PHP версии 7.4.05 + расширения;
- PHPMyAdmin версии 5.0.2.

**Возможности комплекса:**

- Незаметная работа в трее Windows;
- Быстрые старт и остановка;
- Автостарт сервера при запуске программы;
- Несколько режимов управления доменами;
- Монтирование виртуального диска;
- Поддержка управления через командную строку;
- Поддержка профилей настроек;
- Удобный просмотр логов всех компонентов;
- Переключение HTTP, MySQL и PHP модулей;
- Подробная и понятная документация;
- Доступ к доменам в один клик;
- Быстрый доступ к шаблонам конфигурации;
- Мультиязычный интерфейс;
- Автозапуск программ по списку.

**Особенности комплекса:**

- Не требует установки (портативность);
- Возможность работы с USB накопителя;
- Одновременная работа с Denwer, Xampp и т.д.;
- Работа на локальном/сетевом/внешнем IP адресе;
- Поддержка SSL без всякой дополн. настройки;
- Создание домена путем создания обычной папки;
- Поддержка кириллических доменов;
- Поддержка алиасов (доменных указателей);
- Защита сервера от внешнего доступа;
- Punycode конвертер доменных имён;

- Набор популярных сторонних расширений PHP
- Планировщик заданий (cron);
- Создание локального поддомена без потери видимости основного домена в сети интернет.

### **phpMyAdmin**

PHPMyAdmin - Веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой Веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у Веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд [12].

Приложение распространяется под лицензией GNU General Public License и поэтому многие другие разработчики интегрируют его в свои разработки, например XAMPP, Denwer, AppServ, Open Server.

#### Возможности phpMyAdmin

- интуитивно понятный Веб-интерфейс;
- поддержка большинства функций MySQL;
- просмотр и удаление баз данных, таблиц, вьюшек, полей и индексов;
- создание, копирование, удаление, переименование и изменение баз данных, таблиц, полей и индексов;
- управление сервером, базами данных и таблицами, с советами по настройке сервера;
- выполнение, редакция и сохранение любого SQL-выражения, включая пакетные запросы;
- управление пользователями MySQL и их привилегиями;

- работа с хранимыми процедурами и триггерами;
- поддержка импорта данных из CSV и SQL;
- поддержка экспорта в различные форматы CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument текст и таблицы, Word, Excel, LATEX и другие
- администрирование нескольких серверов;
- генерирование наглядных схем баз данных в виде PDF;
- создание комплексных запросов с помощью функции Запрос по шаблону;
- глобальный или частичный поиск в базе данных.

### **1.2.3 Выбор типа базы данных**

#### **Реляционные базы данных**

Реляционная база данных — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Строка таблицы содержит данные об одном объекте (например, товаре, клиенте), а столбцы таблицы описывают различные характеристики этих объектов — атрибутов (например, наименование, код товара, сведения о клиенте). Записи, т. е. строки таблицы, имеют одинаковую структуру — они состоят из полей, хранящих атрибуты объекта. Каждое поле, т. е. столбец, описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них отображаются различные информационные свойства объекта.

В реляционной базе данных каждая таблица должна иметь первичный ключ поле или комбинацию полей, которые единственным образом идентифицируют каждую строку таблицы. Если ключ состоит из нескольких полей, он называется составным. Ключ должен быть уникальным и однозначно определять запись. По значению ключа можно отыскать единственную запись. Ключи служат также для упорядочивания информации в БД.

Таблицы реляционной БД должны отвечать требованиям нормализации отношений. Нормализация отношений — это формальный аппарат ограничений на формирование таблиц, который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение базы данных.

Реляционные БД находят применение повсеместно: в организациях для учёта персонала, ведения бухгалтерии, учёта товаров на складе, поставщиков, партнёров, клиентов, ведения электронного документооборота, в адресных и телефонных книгах, словарях, справочниках. Но все эти варианты использования объединяет одно свойство – строгая типизация данных, что не свойственно приложениям, осуществляющим планирование практических задач и имеющих гибкую структуру данных.

### **Нереляционные базы данных**

Базы данных NoSQL хорошо подходят для многих современных приложений, например мобильных, игровых, интернет-приложений, когда требуются гибкие масштабируемые базы данных с высокой производительностью и широкими функциональными возможностями, способные обеспечивать максимальное удобство использования.

Как правило, базы данных NoSQL предлагают гибкие схемы, что позволяет осуществлять разработку быстрее и обеспечивает возможность поэтапной реализации. Благодаря использованию гибких моделей данных БД NoSQL хорошо подходят для частично структурированных и неструктурированных данных.



Базы данных NoSQL оптимизированы для конкретных моделей данных и шаблонов доступа, что позволяет достичь более высокой производительности по сравнению с реляционными базами данных.

Базы данных NoSQL предоставляют API и типы данных с широкой функциональностью, которые специально разработаны для соответствующих моделей данных.

## **Выбор СУБД**

### **MySQL**

MySQL это система управления реляционными базами данных с открытым исходным кодом (СУРБД) с моделью клиент-сервер. СУРБД — это программное обеспечение или служба, используемая для создания и управления базами данных на основе реляционной модели. Разработку и поддержку MySQL осуществляет корпорация Oracle. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц [4][5].

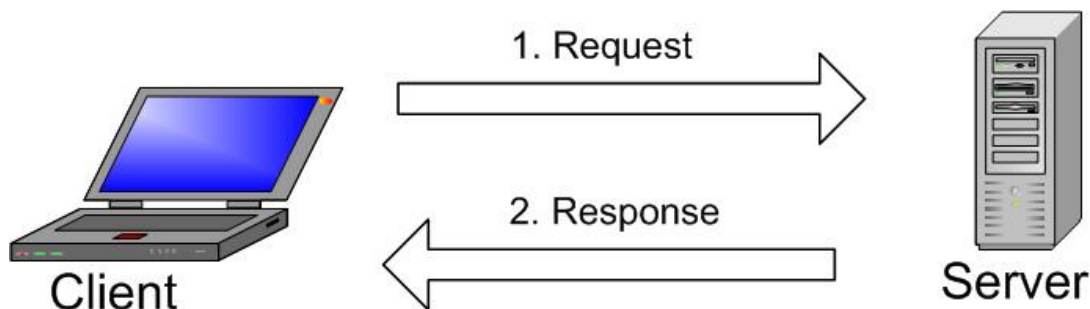


Рисунок 6 – Базовая структура клиент-сервер

Одно или несколько устройств (клиентов) подключаются к серверу через определённую сеть. Каждый клиент может сделать запрос из графического интерфейса пользователя (GUI) на своих экранах, и сервер выдаст желаемый результат, если оба конца понимают инструкцию.

## **MariaDB**

MariaDB — ответвление реляционной СУБД MySQL, разрабатываемое сообществом под лицензией GPL. MariaDB полностью совместима с приложениями, использующими MySQL. Именно с этой СУБД мы и будем работать в будущем, а переход на эту СУБД оправдан тем, что MySQL уже не так активно развивается [13].

Кроме понятной политики распространения продукта, данная СУБД отличается от MySQL более высокой производительностью, новыми возможностями и меньшим количеством ошибок в коде (в частности, за счёт лучшего покрытия тестами и более активного развития).

В MariaDB встроены улучшенный оптимизатор запросов, безопасная и быстрая репликация, более быстрые индексы для механизма хранения данных MEMORY(HEAP), более высокая производительность перекодировки символов, использование пула потоков, а также многие другие улучшения, позитивно влияющие на производительность.

В MariaDB произошёл отказ от подсистемы хранения данных InnoDB и его замена на XtraDB. Также в дополнение к MyISAM, Blackhole, CSV, Memory и Archive включены подсистемы хранения Aria, PBXT (до версии 5.5), FederatedX, OQGRAPH, SphinxSE, IBMDB2I (до версии 5.5), TokuDB, а также с версии 10.0 — Cassandra, CONNECT, SEQUENCE, Spider.

## **ВЫВОД**

Все вышеперечисленные платформы являются одними из самых популярных решений для разработки в последнее время, это обеспечивается их гибкостью, простотой и свободной лицензией. Так же во время разработки продукт будет проходить тестирование в браузере Google Chrome так как он основан на самом популярном движке chromium на котором написано большинство современных браузеров (Яндекс Браузер, Opera, Microsoft Edge, Epic Browser, Comodo Dragon и др.)

Перечислим все инструменты, которыми нам нужно будет пользоваться в процессе разработки:

- HTML;
- PHP;
- CSS;
- JavaScript;
- OpenServer;
- MariaDB;
- PHPMyAdmin;
- Google Chrome.

## 2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Для реализации данной системы необходим следующий набор подсистем:

1. *MySQL* база данных *MariaDB*, обеспечивающая хранение данных о пользователях и всей информации связанной с сервисным центром.
2. *Сервисы OpenServer*, обеспечивающих удобную работу с приложением с любого компьютера в сети.
3. *Бизнес-логика приложения* – организация обработки данных и их размещение в базе данных.
4. *Графический интерфейс клиентского приложения* – настройка и контроль работы клиентского приложения, визуальное отображение регистрации в системе, работа с базой данных без необходимости открывать ее напрямую;

### 2.1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

#### 2.1.1 Требования к регистрации и авторизации

В ходе проектирования, разработки и анализа технического задания были определены следующие функциональные требования.

Для получения доступа к функционалу пользователя, требуется регистрация. У пользователя может быть одна из трех ролей: клиент, администратор и мастер. У каждой роли свои права.

Независимо от роли, в которой выступает пользователь, он должен иметь следующие возможности.

- зарегистрироваться (для регистрации требуется ввести логин, мобильный телефон, ф.и.о и пароль);
- авторизоваться (для авторизации требуется ввести логин и пароль);
- сменить пароль;

- добавить e-mail, если пользователь не указал его при регистрации.

Роль «клиент»:

- просмотреть активные заказы;
- просмотреть завершенные заказы.

Роль «мастер»

- просматривать заявки на свое имя.

Роль «менеджер»

- добавить новую заявку;
- просмотр полного списка заявок и их редактирование;
- изменение данных заявки, назначения на него мастера и статуса;
- добавление техники на продажу;
- просмотр всего списка техник на продажу и редактирование его.

### **2.1.2 Требования к безопасности**

Система должна обладать следующими средствами безопасности:

- пользователь при входе в систему обязан указывать логин и пароль;
- пользователь при регистрации в системе обязан указывать пароль два раза (для подтверждения);
- пароль шифруется на стороне сервера-обработчика;
- важные конфиденциальные данные передаются на сервер POST запросом.

## **2.2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

### **2.2.1 Требования к построению интерфейса системы**

Приложение должно графически разделять следующие функциональные структуры:

1. Верхняя часть(header).

- имя фамилия пользователя;
- логотип название расписание и номер сервисного центра;
- меню навигации.

2. Контент.

3. Нижняя часть(footer) немного информации и правил.

Веб-приложение должно адаптироваться под мобильные устройства.

Проекты и задачи, относящиеся к пользователю, должны быть логически построены в интерфейсе сервиса.

### 3 ПРОЕКТИРОВАНИЕ

#### 3.1 АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

Рисунок 7 изображает схему модулей, используемых в приложении с графическим изображением их взаимодействия.



Рисунок 7 – Архитектура приложения

Переходя по страницам сайта, пользователь взаимодействует только с интерфейсам пользователя, но когда он выполнить действие связанное с сервером, будь то регистрация, вход, работа с базой данных и так далее, его запрос идет на обработку сервера, и если запрос корректен, сервер отправляет запрос к базе данных, которая ,в свою очередь, возвращает ответ, который сервер преобразует в JSON файл и передает обработчику, который преобразует его в HTML и отправляется на интерфейс пользователя. Если запрос не прошел проверку, то сервер так же отправляет JSON файл с ошибкой обработчику, который преобразует его в HTML и отправляется на интерфейс пользователя.

### 3.1.1 Интерфейс пользователя

Интерфейс пользователя представлен в виде HTML языка разметки, языка стилей CSS, когда пользователь совершает какое-нибудь действие, то оно обрабатывается либо внутри интерфейса, как например переход по страницам сайта, так и отправляется запрос на Backend, в случае если пользователю, посредством отправки формы, надо получить результат от БД. Так же есть JavaScript, который не входит в интерфейс пользователя, но он все же отвечает за некоторые анимации, в частности за открытие модальных окон.



Рисунок 8 – Схема отображения результата пользователю (Frontend)



### 3.1.2 Сервер API

Серверная часть реализуется на языке PHP с вспомогательным JavaScript. На рисунке 9 представлена схема его работы.

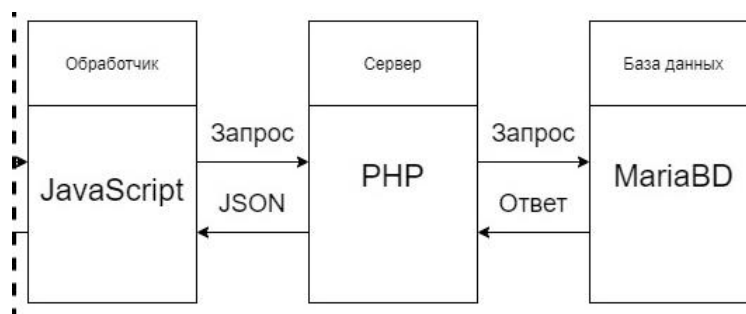


Рисунок 9 – Схема работы сервера (Backend)

Шаги работы:

1. С интерфейса пользователя приходит запрос
2. Сервер обрабатывает запрос
3. Если что-то неверно, возвращает JSON с ошибкой
4. Если все верно отправляет запрос в БД
5. БД получает запрос и выдает серверу ответ
6. Сервер получает ответ и отправляет его в виде JSON интерфейсу пользователя

## 3.2 АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

Основной задачей Веб-приложения является взаимодействие клиента с сервисным центром, посредством сайта и работников с базой данных посредством его же. Следовательно, требуется организовать удобный доступ к базе данных с помощью приложения.

1. Клиент в личном кабинете может просматривать статус своего заказа (ожидает, диагностика, ремонт, готово) без необходимости звонить и узнавать у работников, что опять же разгружает сервис от лишних звонков.
2. Мастер, как и клиент может просматривать все заявки, которые назначены ему, без необходимости записывать, или даже запоминать что он должен сделать.
3. У менеджера функционал шире, чем у остальных, так как его главная работа заключается в работе с клиентами. Он может добавлять заявки, изменять их данные, просматривать полныи список заявок, осуществлять поиск по списку, а также распределять новые заявки по мастерам, чтобы обеспечить равномерную нагрузку.

### **3.3 ОПИСАНИЕ ДАННЫХ**

#### **Входные данные**

Входные данные необходимые для регистрации:

- Логин;
- адрес электронной почты;
- ф.и.о;
- номер телефона;
- e-mail(необязательно);
- пароль.

Входные данные необходимые для авторизации:

- логин;
- пароль;
- входные данные необходимые для смены пароля:

- старый пароль;
- новый пароль.

Входные данные необходимые для создания новой заявки в том случае если клиент уже зарегистрирован в системе:

- логин клиента;
- тип техники;
- модель;
- проблема.

Входные данные необходимые для создания новой заявки в том случае если клиент не зарегистрирован в системе:

- ф.и.о клиента;
- номер телефона клиента;
- тип техники;
- модель;
- проблема.

Входные данные необходимые для создания новой позиции в таблице Б\у техники:

- тип техники;
- модель техники;
- описание техники;
- цена.

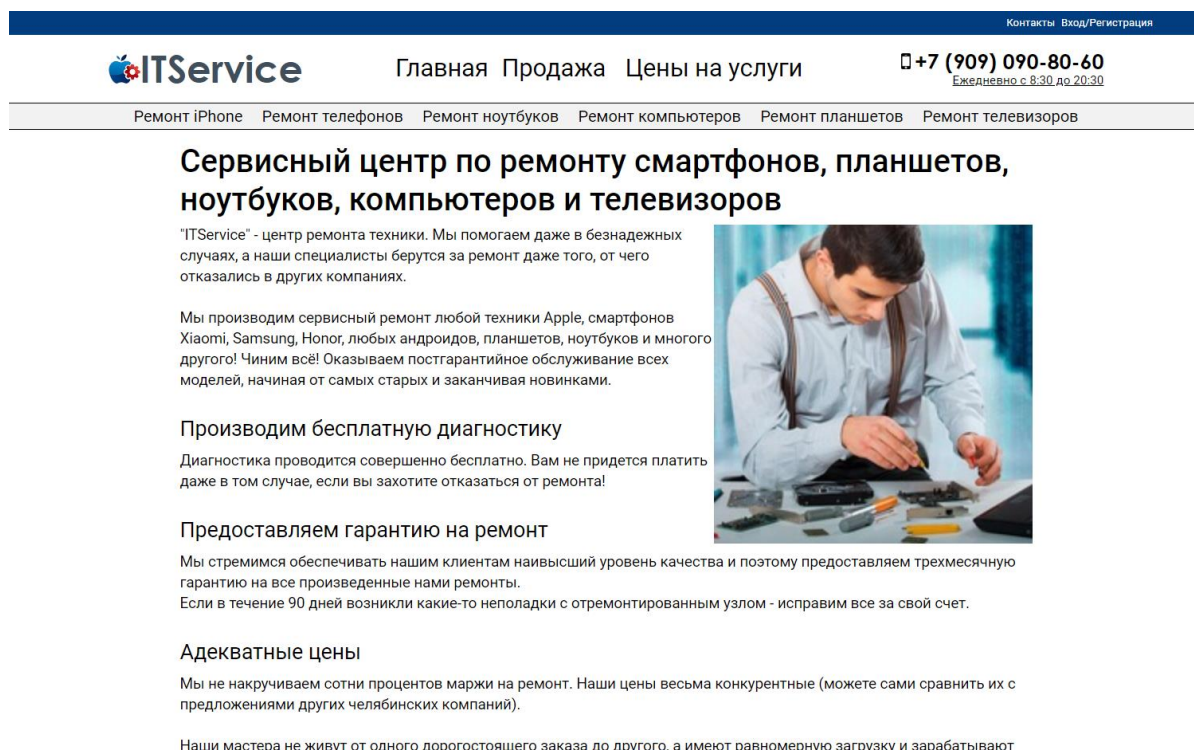
### **Выходные данные**

Выходными данными в Веб-приложении является информация из базы данных представленная в виде таблиц.

## 4 РЕАЛИЗАЦИЯ

### 4.1 ГЛАВНАЯ СТРАНИЦА ВЕБ-ПРИЛОЖЕНИЯ

При входе в приложение неавторизованный пользователь попадает на главное домашнее окно, которое содержит всю необходимую информацию о сервисном центре (Рисунок 10-11).



Контакты Вход/Регистрация

**ITService** Главная Продажа Цены на услуги ☎ +7 (909) 090-80-60  
Ежедневно с 8:30 до 20:30

Ремонт iPhone Ремонт телефонов Ремонт ноутбуков Ремонт компьютеров Ремонт планшетов Ремонт телевизоров

### Сервисный центр по ремонту смартфонов, планшетов, ноутбуков, компьютеров и телевизоров

"ITService" - центр ремонта техники. Мы помогаем даже в безнадежных случаях, а наши специалисты берутся за ремонт даже того, от чего отказались в других компаниях.

Мы производим сервисный ремонт любой техники Apple, смартфонов Xiaomi, Samsung, Honor, любых андроидов, планшетов, ноутбуков и многого другого! Чиним всё! Оказываем постгарантийное обслуживание всех моделей, начиная от самых старых и заканчивая новинками.

**Производим бесплатную диагностику**  
Диагностика проводится совершенно бесплатно. Вам не придется платить даже в том случае, если вы захотите отказаться от ремонта!

**Предоставляем гарантию на ремонт**  
Мы стремимся обеспечивать нашим клиентам наивысший уровень качества и поэтому предоставляем трехмесячную гарантию на все произведенные нами ремонты.  
Если в течение 90 дней возникли какие-то неполадки с отремонтированным узлом - исправим все за свой счет.

**Адекватные цены**  
Мы не накручиваем сотни процентов маржи на ремонт. Наши цены весьма конкурентные (можете сами сравнить их с предложениями других челябинских компаний).

Наши мастера не живут от одного дорогостоящего заказа до другого, а имеют равномерную загрузку и зарабатывают

Рисунок 10 – Главная страница приложения часть 1

На рисунке 10 можно увидеть навигационные элементы, для перемещения по страницам Веб-приложения, логотип и название компании, а также номер телефона и время работы. Все эти элементы находятся в блоке <header>, который виден на каждой странице сайта. После него идет контент страницы. В данном случае это описание с фотографией одного из наших мастеров. Все элементы

разделены между собой цветами, что бы пользователю было легче ориентироваться в приложении.

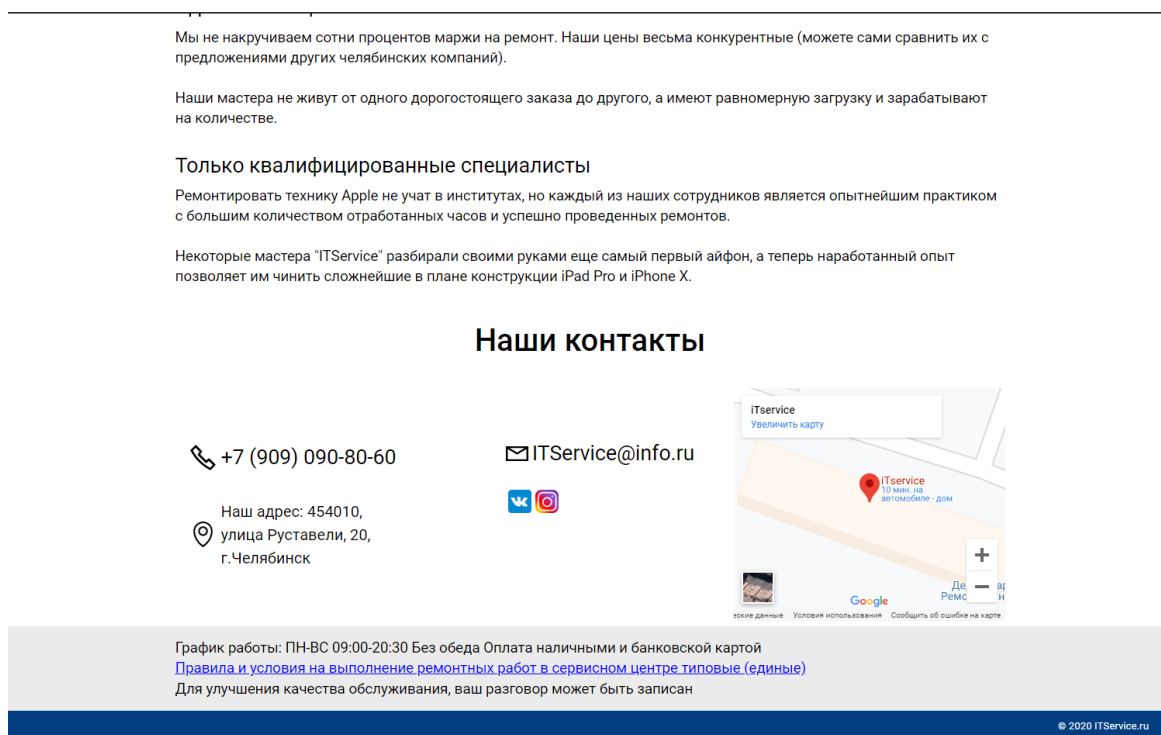


Рисунок 11 – Главная страница приложения часть 2

На рисунке 11 закрывает контент блок со всеми контактами: номер телефона, адрес, E-mail, ссылки на социальные сети «Вконтакте», «Инстаграм» и карта Google Maps. Следом идет footer. В нем содержится не такая важная информация и дублируется время работы.

На часть «Наши контакты» можно перейти из любого места на сайте нажав по кнопке «Контакты», которая находится в верхней части экрана. (Рисунок 12)

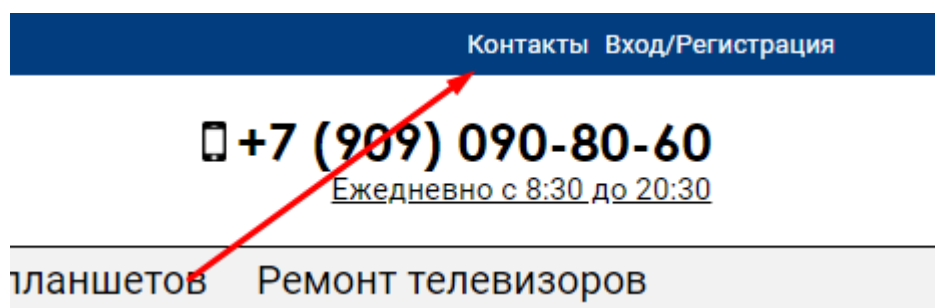


Рисунок 12 – Кнопка перехода к «Наши контакты»

## 4.2 АВТОРИЗАЦИЯ\РЕГИСТРАЦИЯ ПОЛЬЗОВАТЕЛЯ.

Неавторизированный пользователь может авторизоваться или, случае если он не зарегистрирован - зарегистрироваться. Сделать это можно нажав на соответствующую кнопку вверху сайта «Вход/Регистрация» (рисунок 13). Затем откроется Модальное окно предлагающее совершить вход или зарегистрироваться. Переключение между формами осуществляется путем нажатия на соответствующие пункты меню (Рисунок 14-15). Закрывать окно можно нажатием на затемненный участок возле него.

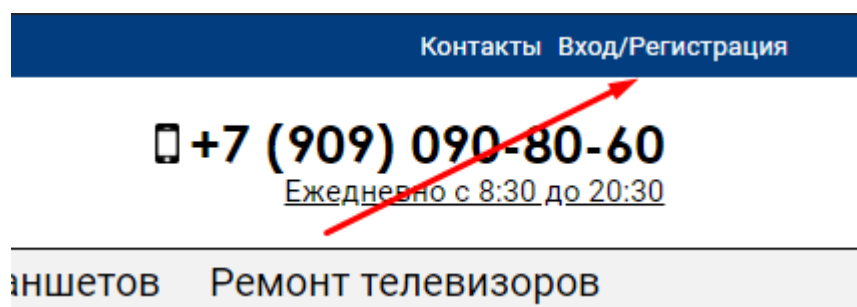


Рисунок 13 – Кнопка «Вход/Регистрация»

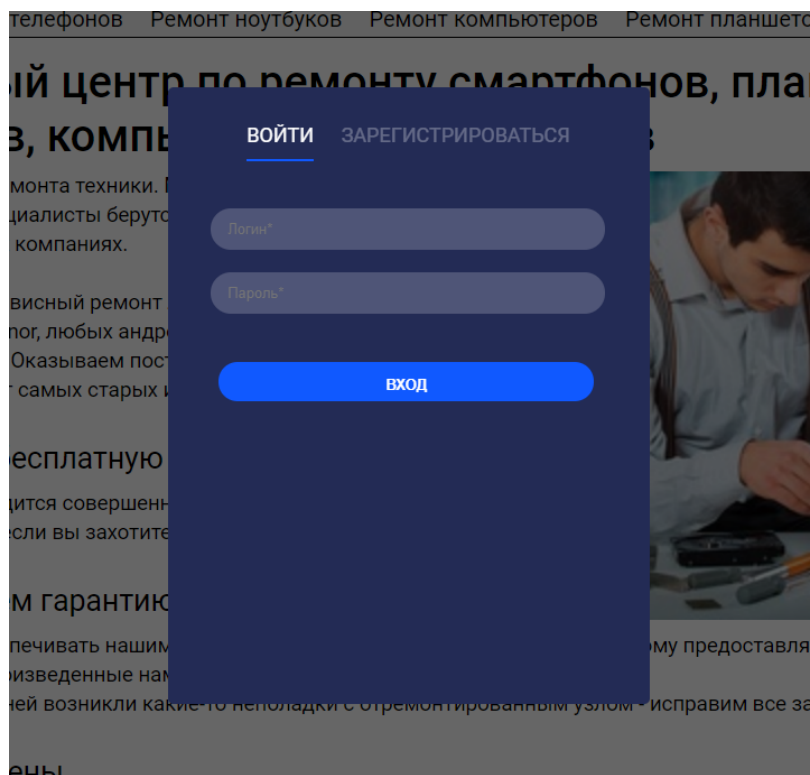


Рисунок 14 – Модальное окно с формой входа

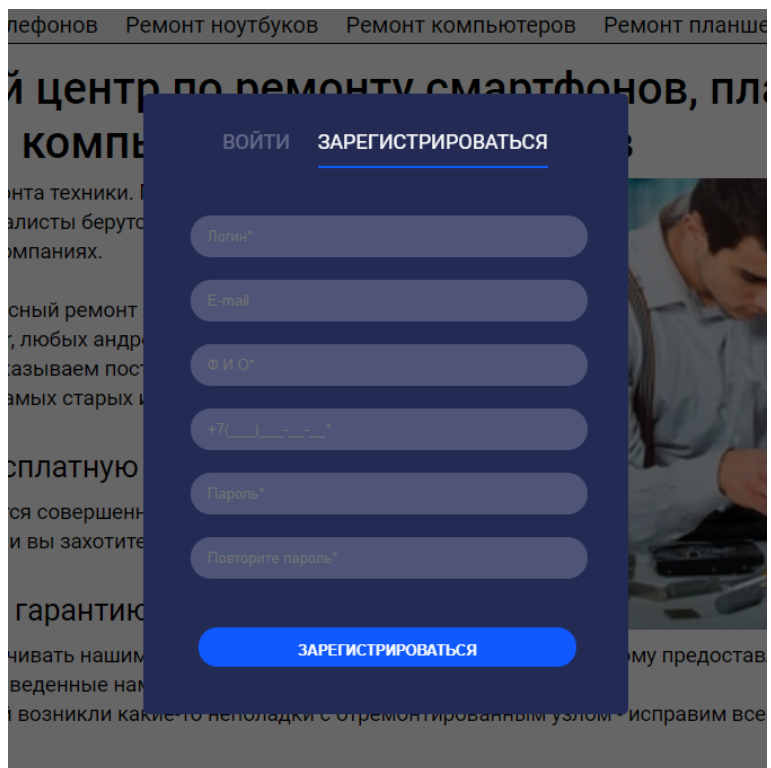


Рисунок 15 – Модальное окно с формой регистрации

После того как пользователь авторизуется, кнопка «Вход/Регистрация» заменяется на Имя и фамилию пользователя и кнопку «Выход». Так же если учетная запись менеджера, то появляется кнопка для работы с базой данных. Другими способами на эту страницу не попасть. Даже если ввести ссылку в адресной строке. Страница перед тем, как показать контент проверяет роль авторизованного пользователя, и если это не менеджер, то перенаправляет на 404 страницу.

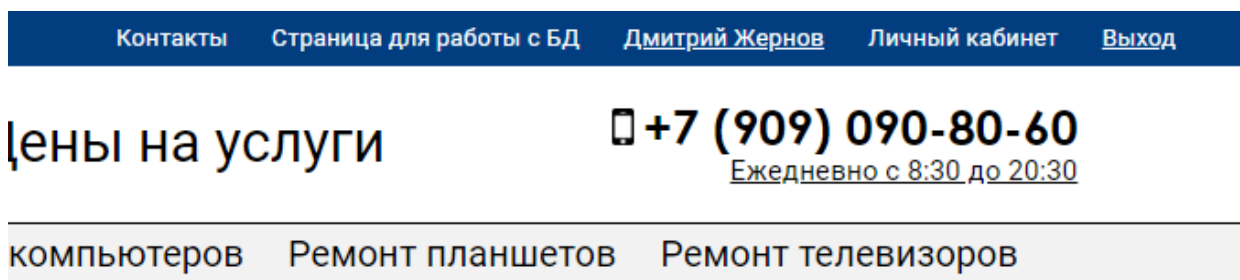


Рисунок 16 – Авторизованный менеджер

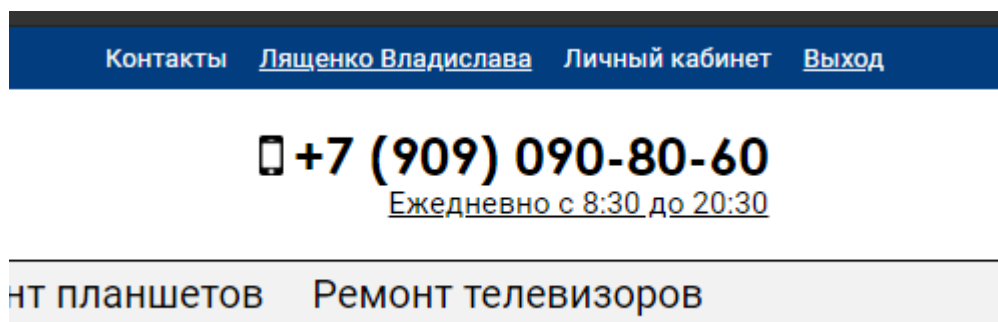


Рисунок 17 – Авторизованный клиент



## 4.3 АДАПТИВНОСТЬ

Веб-приложение адаптируется под размер экрана устройства пользователя. Непомещающиеся элементы навигации в header сворачиваются. Доступ к ним можно получить, нажав на появившуюся кнопку, которая вызовет меню слева. Рисунки с 18 по 20 являются демонстрацией как работает адаптивность на iPhone 6/7/8.

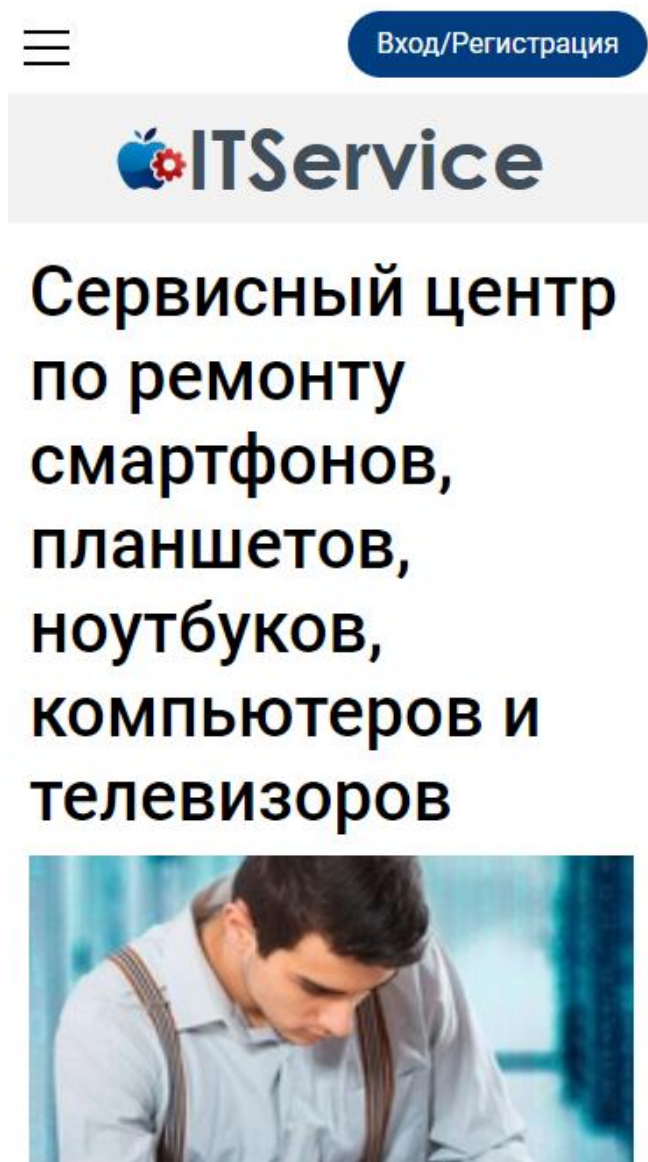


Рисунок 18 – Главная страница

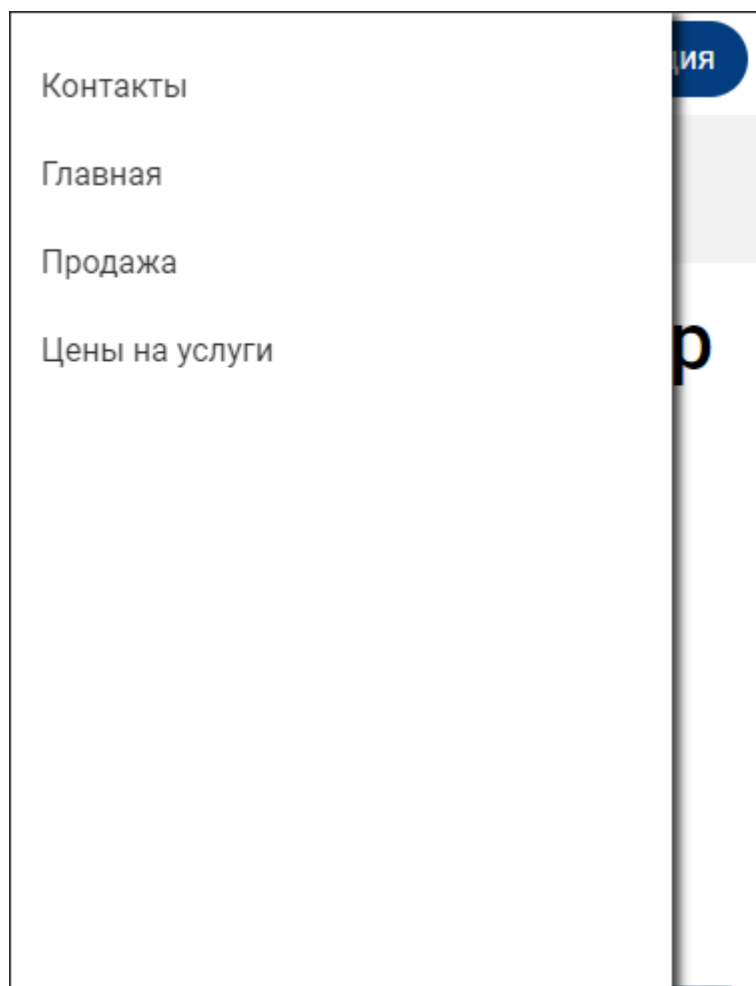


Рисунок 19 – Меню, которое вызывается при нажатии на кнопку

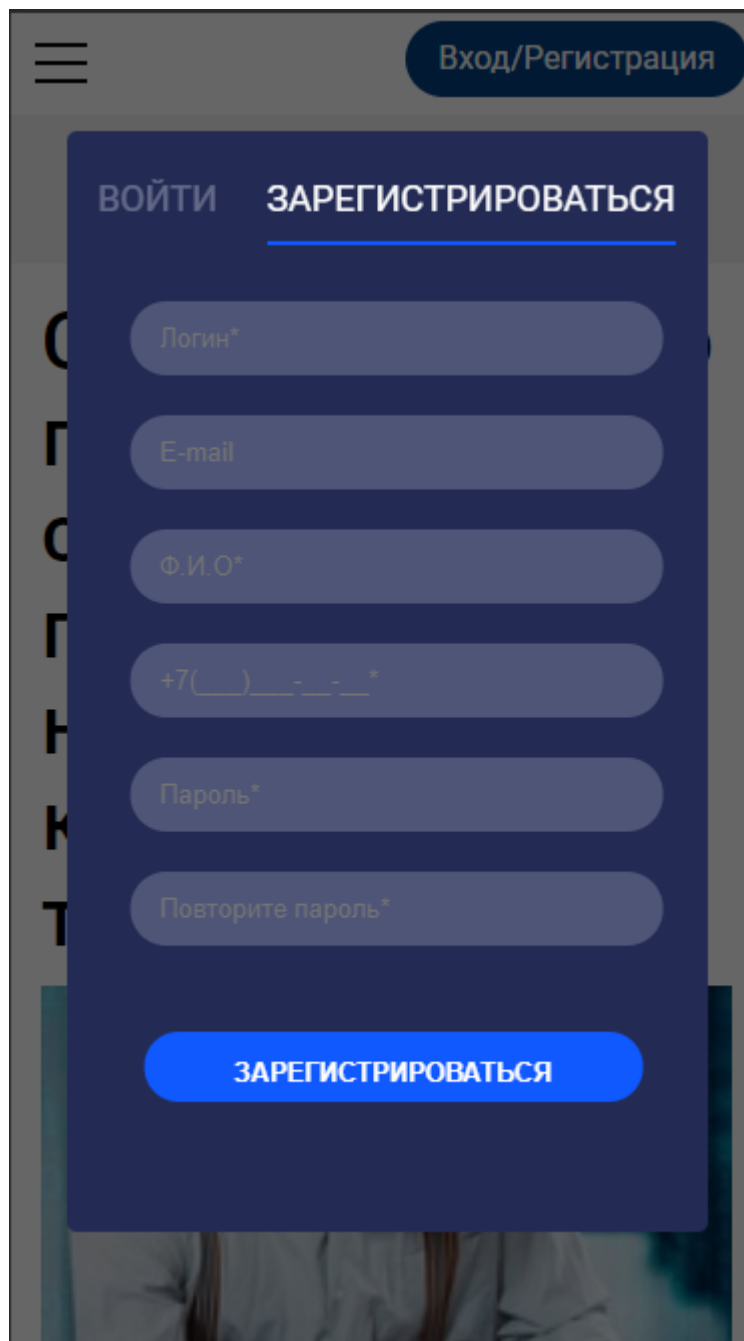


Рисунок 20 – Модальное окно

## 4.4 ЛИЧНЫЙ КАБИНЕТ ПОЛЬЗОВАТЕЛЯ

Любому авторизованному пользователю предоставляется личный кабинет, страница, на которой доступно изменение параметров его учетной записи и просмотр статусов его заказов или, если это мастер, то активные заявки. На рисунке 21 представлен скриншот личного кабинета пользователя

### Личный кабинет

Активные заказы

Завершенные заказы

Техника:	Проблема:	Статус:
Iphone 6s	Не работает сканер	Диагностика

### Контактные данные

Контактное лицо:

Лященко Владислава

Электронная почта:

vlada@info.ru

Номер телефона:

+79654758695

Логин:

Vladalych

РЕДАКТИРОВАТЬ ЛИЧНЫЕ ДАННЫЕ

### Изменение пароля

Старый пароль:

Старый пароль\*

Новый пароль:

Новый пароль\*

Повторите новый пароль:

Повторите новый пароль\*

СМЕНИТЬ

Рисунок 21 – Личный кабинет пользователя

## 4.5 СОЗДАНИЕ И ОТОБРАЖЕНИЕ ЗАЯВОК

Переход на страницу с работой базы данных доступен только менеджеру и осуществляется по нажатию на соответствующую кнопку «Страница работы с БД» в header'е страницы. На странице доступны такие функции как:

- просмотр списка зарегистрированных пользователей;
- просмотр всех заявок;
- поиск конкретной заявки по номеру телефона клиента;
- редактирование заявки;
- создание заявки;
- добавления техники на продажу;
- просмотр всего списка продажи и редактирование его.

Пример представлен на рисунке 22.

The screenshot shows the top navigation bar of the ITService website. The header includes the logo 'ITService', navigation links for 'Главная', 'Продажа', and 'Цены на услуги', and contact information: '+7 (909) 090-80-60' with a note 'Ежедневно с 8:30 до 20:30'. Below the header is a horizontal menu with categories: 'Ремонт iPhone', 'Ремонт телефонов', 'Ремонт ноутбуков', 'Ремонт компьютеров', 'Ремонт планшетов', and 'Ремонт телевизоров'. A row of buttons is displayed: 'Просмотр пользователей', 'Просмотреть все заявки', 'Добавить заявку' (highlighted in blue), 'Просмотреть весь список продажи', and 'Добавить технику на продажу'. Below this is a registration status section with 'Зарегистрирован' (highlighted in blue) and 'Не зарегистрирован'. The main form contains four input fields: 'Логин клиента\*' (with a sub-label 'Логин клиента\*'), 'Тип техники\*' (with a sub-label 'Тип техники\*'), 'Модель\*' (with a sub-label 'Модель\*'), and 'Проблема\*' (with a sub-label 'Проблема\*'). A large blue button labeled 'ДОБАВИТЬ' is positioned at the bottom of the form.

Рисунок 22 – Страница для работы с базой данных

Как можно заметить сразу открывается форма для регистрации заявки. Сделано это для быстрого создания заявки в минимальное количества кликов.

Редактирование ячеек происходит по нажатию на соответствующее поле в таблице, после чего открывается форма редактирования. Для примера воспользуемся формой поиска по номеру телефона, а затем попытаемся ее отредактировать.

Просмотр **Просмотреть все заявки** Добавить заявку Просмотреть весь список продаж Добавить технику на продажу

ПОЛУЧИТЬ ВСЕ СПИСОК

Номер телефона:  
+79654758695

ПОЛУЧИТЬ ЧАСТИЧНЫЙ СПИСОК

ID:	Мастер:	Статус:	Ф.И.О:	Номер телефона:	Тип:	Модель:	Проблема:
1	Максим Булыгин	Диагностика	Лященко Владислава	+79654758695	Смартфон	Iphone 6s	Не работает сканер

Рисунок 23 – Поиск по всем заявкам

Просмотр **Просмотреть все заявки** Добавить заявку Просмотреть весь список продаж Добавить технику на продажу

Изменение данных заявки

Логин Мастера:  
Максим Булыгин

Статус:  
Диагностика

Ф.И.О:  
Лященко Владислава

Номер телефона:  
+79654758695

Тип техники:  
Смартфон

Модель:  
Iphone 6s

Проблема:  
Не работает сканер

ПОДТВЕРДИТЬ

Рисунок 24 – Форма редактирования

## 5 ТЕСТИРОВАНИЕ

### 5.1 ТЕСТИРОВАНИЕ ФОРМ ВВОДА

Проект разрабатывался исключительно в рамках дипломного проектирования. Поэтому было проведено только альфа-тестирование продукта. Для нахождения багов и предотвращения нежелательных событий при работе программы были проведены модульные тесты.

На рисунках 25–26 представлены проверки при неправильно введенных для сервиса значениях.

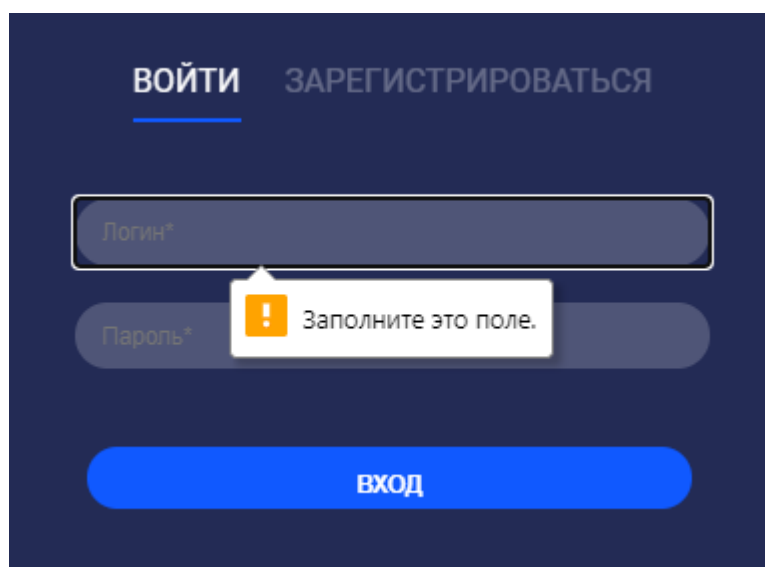


Рисунок 25 – Валидация формы на заполненность

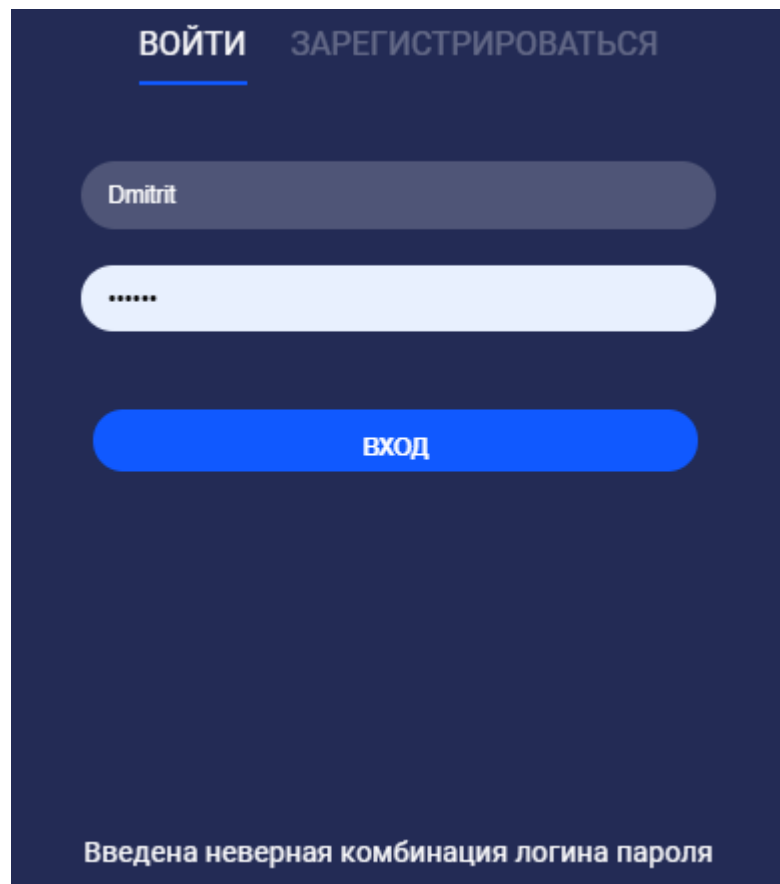


Рисунок 26 – Проверка существования комбинации логин-пароль в БД



## 5.2 ТЕСТИРОВАНИЕ ФОРМЫ ВВОДА РЕГИСТРАЦИИ ПОЛЬЗОВАТЕЛЯ

На рисунках 27– 31 представлены проверки при неправильно введенных для сервиса значениях.

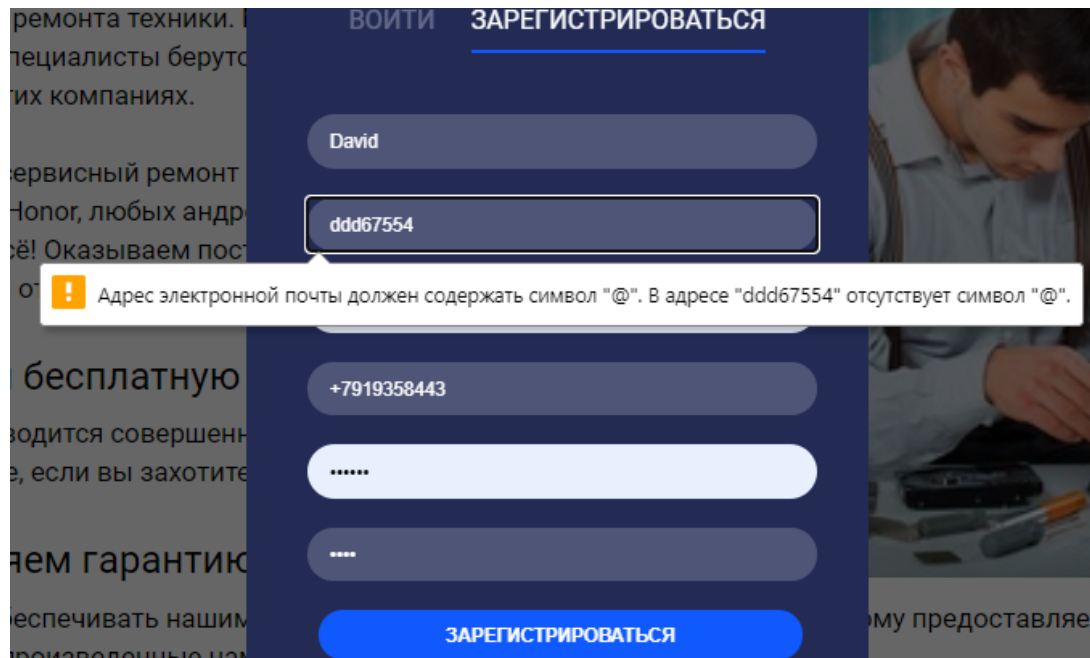


Рисунок 27 – Валидация Email

ВОЙТИ ЗАРЕГИСТРИРОВАТЬСЯ

DiZeRRenush

dima55hemov@gmail.com

Dmitr

+79164525856

\*\*\*\*\*

\*\*\*\*\*

**ЗАРЕГИСТРИРОВАТЬСЯ**

Пользователь с таким логином уже существует

Рисунок 28 – Проверка уникальности логина

ВОЙТИ ЗАРЕГИСТРИРОВАТЬСЯ

DiZol

dima55zhernov@gmail.com

Dmitr

+79164525855

\*\*\*\*\*

\*\*\*\*\*

**ЗАРЕГИСТРИРОВАТЬСЯ**

Пользователь с таким Email уже существует

Рисунок 29 – Проверка уникальности E-mail

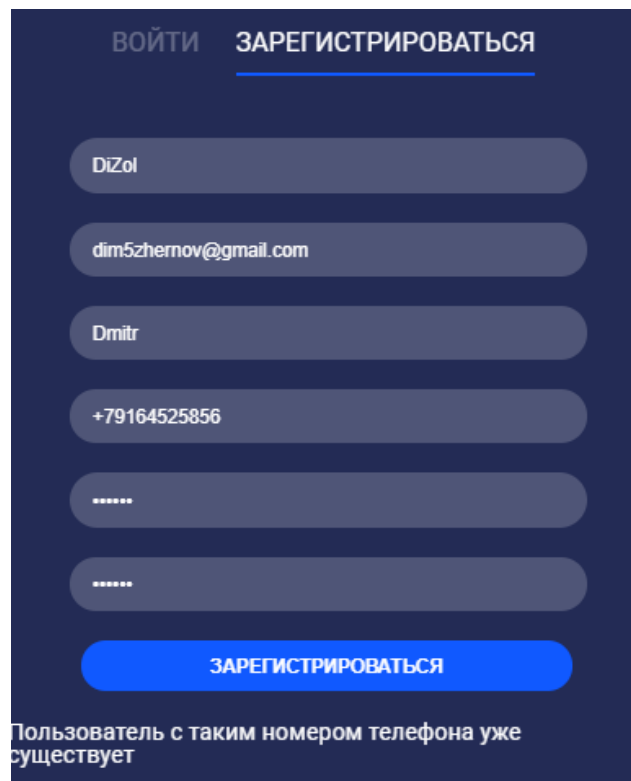


Рисунок 30 – Проверка уникальности телефона

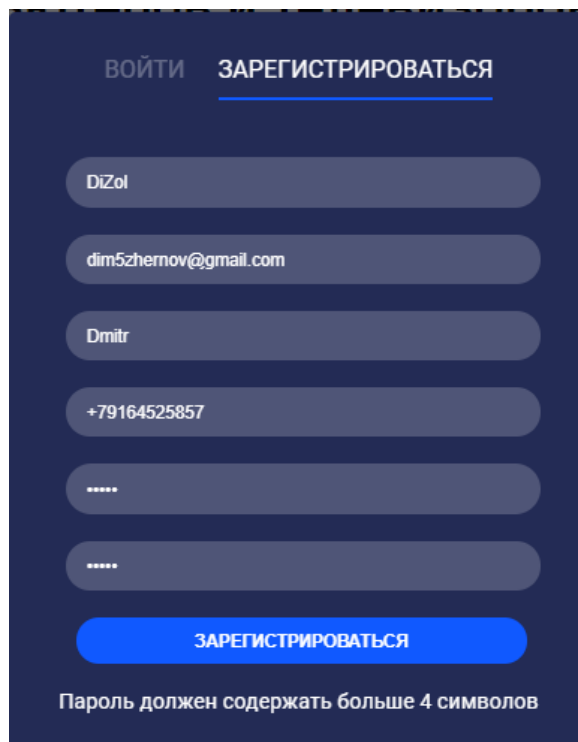


Рисунок 31 – Валидация пароля

The image shows a dark-themed registration form. At the top, there are two links: "ВОЙТИ" (Login) and "ЗАРЕГИСТРИРОВАТЬСЯ" (Register), with the latter being underlined. Below the links are six input fields: a name field containing "DiZol", an email field containing "dim5zhernov@gmail.com", a surname field containing "Dmitr", a phone number field containing "+79164525857", and two password fields, both containing six asterisks. A blue button labeled "ЗАРЕГИСТРИРОВАТЬСЯ" is positioned below the password fields. At the bottom of the form, the text "Пароли не совпадают" (Passwords do not match) is displayed.

Рисунок 32 – Валидация пары паролей

В случае успешной регистрации, пользователя перенаправляет на форму входа, где он может выполнить вход, используя данные, введенные при регистрации.

Войти    ЗАРЕГИСТРИРОВАТЬСЯ

Логин\*

Пароль\*

ВХОД

Вы успешно зарегистрировались, можете выполнить вход

The image shows a dark blue login form. At the top, there are two links: 'ВОЙТИ' (Login) which is underlined in blue, and 'ЗАРЕГИСТРИРОВАТЬСЯ' (Register) in a lighter blue. Below the links are two input fields: 'Логин\*' (Login\*) and 'Пароль\*' (Password\*). Below the input fields is a prominent blue button with the text 'ВХОД' (Login). At the bottom of the form, there is a white text message: 'Вы успешно зарегистрировались, можете выполнить вход' (You have successfully registered, you can now login).

Рисунок 33 – Форма входа с оповещением об успешной регистрации

## 5.3 ТЕСТИРОВАНИЕ ФОРМЫ ДОБАВЛЕНИЯ ЗАЯВКИ

Для начала посмотрим какие на данный момент есть заявки в сервисе, для этого во вкладке «Просмотреть все заявки» необходимо нажать на «Весь список».

The screenshot shows a navigation bar with five buttons: "Просмотр пользователей", "Просмотреть все заявки" (highlighted in blue), "Добавить заявку", "Просмотреть весь список продаж", and "Добавить технику на продажу". Below the navigation bar is a blue button labeled "ВСЕ СПИСОК". Underneath is a search field labeled "Номер телефона:" with the placeholder text "Номер телефона\*". Below the search field is another blue button labeled "ПОИСК". At the bottom is a table with the following data:

ID:	Мастер:	Статус:	Ф.И.О:	Номер телефона:	Тип:	Модель:	Проблема:
1	Максим Булыгин	Диагностика	Лященко Владислава	+79654758695	Смартфон	Iphone 6s	Не работает сканер
2	Не назначен	Ожидает	Давид	+79195354175	Смартфон	Iphone X	Замена дисплея
3	Не назначен	Ожидает	Дмитрий Жуков	+79197654765	Компьютер	Не указана	Выключается

Рисунок 34 – Список до теста

Попробуем добавить пользователя, которого не существует в системе

The screenshot shows a form with two tabs: "Зарегистрирован" (highlighted in blue) and "Не зарегистрирован". Below the tabs are five input fields: "Логин клиента:" with the value "ddd", "Тип техники:" with the value "Смартфон", "Модель:" with the value "Iphone X", and "Проблема:" with the value "Замена дисплея". At the bottom is a blue button labeled "ДОБАВИТЬ". Below the button is a red error message: "Пользователя с таким логином не существует".

Рисунок 35 – Добавление несуществующего пользователя

## А теперь добавим существующего пользователя

Логин клиента:

Тип техники:

Модель:

Проблема:

**ДОБАВИТЬ**

Новая заявка успешно добавлена

Рисунок 36 – Добавление существующего пользователя

## Проверим как изменился список

[Просмотр пользователей](#) [Просмотреть все заявки](#) [Добавить заявку](#) [Просмотреть весь список продаж](#) [Добавить технику на продажу](#)

**ВСЕ СПИСОК**

Номер телефона:

**ПОИСК**

ID:	Мастер:	Статус:	Ф.И.О:	Номер телефона:	Тип:	Модель:	Проблема:
1	Максим Булыгин	Диагностика	Лященко Владислава	+79654758695	Смартфон	Iphone 6s	Не работает сканер
2	Не назначен	Ожидает	Давид	+79195354175	Смартфон	Iphone X	Замена дисплея
3	Не назначен	Ожидает	Дмитрий Жуков	+79197654765	Компьютер	Не указана	Выключается
7	Не назначен	Ожидает	Лященко Владислава	+79654758695	Смартфон	Iphone X	Замена дисплея

Рисунок 37 – Список после теста

В базу данных добавилась только заявка, которая прошла все проверки, о чем нам сказала форма на рисунке 36. При необходимости можно также воспользоваться поиском по номеру телефона. Пример представлен на рисунке 37.

[Просмотр пользователей](#)
[Просмотреть все заявки](#)
[Добавить заявку](#)
[Просмотреть весь список продаж](#)
[Добавить технику на продажу](#)

**ВСЬ СПИСОК**

Номер телефона:

**ПОИСК**

ID:	Мастер:	Статус:	Ф.И.О:	Номер телефона:	Тип:	Модель:	Проблема:
1	Максим Булыгин	Диагностика	Лященко Владислава	+79654758695	Смартфон	Iphone 6s	Не работает сканер
7	Не назначен	Ожидает	Лященко Владислава	+79654758695	Смартфон	Iphone X	Замена дисплея

Рисунок 38 – Поиск по БД

## 5.4 ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ

При попытке перейти на страницы, которые не доступны без необходимых прав, происходит редирект на страницу с ошибкой 404 – Страница не найдена.

**Ой!**

Кажется вы потерялись.

Ошибка: 404

Нажмите на ссылку ниже что бы попасть домой:

[Главная](#)



Рисунок 39 – Страница с ошибкой 404



## **ЗАКЛЮЧЕНИЕ**

В ходе дипломного проектирования было выполнено следующее:

1. Проведен анализ решений у конкурентов и были выявлены их достоинства и недостатки.
2. Определены функциональные и нефункциональные требования к приложению.
3. Проведен анализ предметной области и рынка родственных разработок.
4. Спроектировано и разработано Веб-приложения.
5. Выполнено альфа тестирование.

В Веб-приложении были реализованы следующие функции:

1. Выполнена адаптивная верстка для корректного отображения контента на устройствах с разной диагональю экрана.
2. Регистрация и авторизация пользователей.
3. Работа с базой данных без необходимости открывать ее напрямую.
4. Настроена безопасность, что бы пользователи с недостаточными правами не могли попасть на страницы, которые их требуют.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сайт – визитка «Сервис-К». — <http://cservis-k.ru/>. (Дата обращения 05.03.2020.)
2. Группа во Вконтакте «ITService». — [https://vk.com/club\\_itservice](https://vk.com/club_itservice). (Дата обращения 05.03.2020.)
3. Редактор исходного кода Visual Studio Code. — <https://code.visualstudio.com/>. (Дата обращения 06.03.2020.)
4. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство / Б. Маклафлин 2-е изд. СПб.: Питер, 2014. — 544 с. (Дата обращения 06.03.2020.)
5. Колисниченко Д.Н. PHP и MySQL., Разработка Веб-приложений / Д.Н. Колисниченко 5-е изд., перераб и доп. СПб.: БХВ-Петербург, 2015. — 592 с. (Дата обращения 06.03.2020.)
6. Язык программирования Ruby. — <https://learn-code.ru/yazyki-programmirovaniya/ruby>. (Дата обращения 06.03.2020.)
7. Официальная страница языка Python. — <https://www.python.org/>. (Дата обращения 06.03.2020.)
8. Преимущества Python перед другими языками программирования. — <https://www.digest.pro/news/preimushhestva-python-pered-drugimi-jazykami-programmirovaniya/>. (Дата обращения 06.03.2020.)
9. Маккоу А. М15 Веб-приложения на JavaScript / А. Маккоу СПб.: Питер, 2012. — 288 с. (Дата обращения 07.03.2020.)
10. Резиг Д. JavaScript для профессионалов / Р. Фергюсон, Д. Пакстон, Д. Резиг. 2-е изд.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2016. — 240 с. (Дата обращения 08.03.2020.)

11. Официальная страница локального веб сервера «OpenServer». — <https://ospanel.io/>. (Дата обращения 09.03.2020.)
12. Официальная страница интерфейса для администрирования MySQL «PHPMyAdmin». — <https://www.phpmyadmin.net/>. (Дата обращения 09.03.2020.)
13. Официальная страница MariaDB. — <https://mariadb.org/>. (Дата обращения 09.03.2020.)
14. Жернов Д.А. исходный код Веб-приложения ITService. — <https://github.com/Dizerrenush/ITService>.