|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

***Разработка веб-приложения для стоматологической клиники «VivaSmile»***

по дисциплине ***Архитектура систем обработки информации и управления***

Студент гр. ИУК5-51Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Дерюгин П.А. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Смирнов М.Е. )

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Архитектура систем обработки информации и управления***

Студент\_\_\_\_\_Дерюгин П.А. ИУК5-51Б\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель\_Смирнов М.Е.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

***1. Тема курсового проекта***

***«Разработка веб-приложения для стоматологической клиники «VivaSmile»»***

***2. Техническое задание***

*Выполнить исследование и описание предметной области, произвести анализ*

*объекта автоматизации, разработать методы и способы решения технических*

*задач. Разработать и реализовать структуру базы данных, серверную компоненту,*

*интерфейс приложения и взаимодействие с базой данных.*\_

***3. Оформление курсового проекта***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г.

(подпись) (Ф.И.О.)

Примечание:

Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**СОДЕРЖАНИЕ**

1. Техническое задание………………………………………………………..4
2. Научно-исследовательская часть…………………………………………..8
   1. Постановка задачи проектирования………………………………...8
   2. Описание предметной области……………………………………...8
   3. Анализ аналогов и прототипов……………………..……………….10
   4. Перечень задач, подлежащих решению в процессе разработки….13
   5. Обоснование выбора инструментов и платформы для разработки.13
3. Проектно-конструкторская часть………………………………………….21
   1. Разработка структуры приложения…………………………………21
   2. Разработка архитектуры приложения………………………………23
   3. Разработка интерфейса взаимодействия пользователя с системой.25
4. Проектно-технологическая часть………………………………………......28
   1. Проектирование начального и тестового наполнения базы данных.28
   2. Разработка руководства администратора… ………………………37
   3. Разработка руководства пользователя …………………………….40

Заключение……………………………………………………………………..41

Список использованной литературы…………………..……………………..42

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**1.1. Общие сведения.**

**1.1.1. Наименование.**

Тема: «Разработка веб-приложения для стоматологической клиники».

**1.1.2. Исполнитель.**

Студент 3-го курса МГТУ им. Баумана по направлению «Информатика и вычислительная техника» профиль «Системы обработки информации» Дерюгин П.А.

**1.2. Назначение и цели создания системы.**

**1.2.1 Назначение системы.**

Система предназначена для пациентов стоматологической клиники. Данная система позволяет клиентам клиники просматривать расписание работы врачей, а также записываться на первичный прием к врачу. Также администратор (регистратура) может записывать клиентов на прием и взаимодействовать с данными в БД.

**1.2.2 Цель создания системы.**

Основная учебная цель выполнения разработки – повышение уровня

квалификации разработчика в области проектирования, программной

реализации и анализа сложных структур данных и алгоритмов их

обработки.

**1.3. Характеристика системы автоматизации.**

**1.3.1 Краткие сведения об объекте автоматизации.**

Объектом автоматизации являются процесс записи клиентов на прием самостоятельно или администратором.

**1.4. Требования к системе.**

**1.4.1. Требования к задачам, подлежащим решению.**

* Исследование предметной области
* Разработка системы, предназначенной для записи пациентов на прием
* Проведение тестирования алгоритма и анализ результатов
* Документирование проекта в соответствии с установленными

требованиями

**1.4.2. Требование к архитектуре системы.**

* тестируемость (возможность установления факта правильного

функционирования);

* простота эксплуатации (минимальные требования к квалификации и обучению эксплуатирующего персонала);
* защищенность системы от вандалов и неквалифицированных

пользователей;

* модифицируемость (возможность перенастройки для работы с другими технологическими процессами);
* функциональная расширяемость (возможность ввода в систему

дополнительных функциональных возможностей, не предусмотренных в техническом задании);

**1.4.3. Требования к основному функционалу системы.**

Заходя на сайт, клиент может просмотреть прайс-лист услуг клиники, информацию о врачах и записаться на первичный прием к конкретному врачу.

Администратор составляет и ведет расписание работы врачей, а также записывает клиентов на прием.

**1.5. Состав и содержание работ по созданию системы.**

**1.5.1. Этапы работ**

- 2 неделя (12 сентября). Утверждение темы, задания на разработку,

технического задания (альфа);

- 4 неделя (26 сентября). Оформление ТЗ. Описание предметной области и требований к системе. Аналоги. Концептуальная схема. Прототип и скетчи интерфейса;

- 7 неделя (17 октября). Оформление введения и исследовательской части. Обоснование выбора БД (если необходимо), логическая схема БД,

физическая схема данных. Демонстрация работы макета системы с БД;

- 10 неделя (7 ноября). Оформление проектно-конструкторской части.

Демонстрация работающего приложения. Презентация (альфа). Тестирование и отладка приложения. Разработка эксплуатационной документации.

- 14 неделя (1 декабря). Все ошибки и проблемы устранены. Демонстрация проекта. Защита.

**1.6. Порядок контроля и приемки системы.**

Установить контроль и приемку результатов работ на каждой стадии создания системы в соответствии с разделом 5. Результаты работ в течение их выполнения загружаются на сервис GitHub, где преподаватель может контролировать процесс выполнения работ. Основанием для сдачи-приёмки работ служит «Отчёт о завершении работ по стадии», представляемый Исполнителем. Для сдачи-приемки представляется также документация, перечисленная в разделе 8 настоящего ТЗ.

**1.7. Требование к составу и содержанию работ по подготовке объекта автоматизации к вводу системы действия.**

Для подготовки объекта автоматизации к вводу системы в действие должны быть проведены следующие мероприятия:

·Приведение поступающей в систему информации (в соответствии с

требованиями к информационному обеспечению) к виду, пригодному для обработки;

· Изменения, которые необходимо осуществить в объекте автоматизации;

· Создание условий функционирования объекта автоматизации, при

которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ;

**1.8. Требования к документированию.**

Расчетно-пояснительная записка по разработанной системе. Расчетно-пояснительная записка состоит из следующих частей:

• Техническое задание

• Научно-исследовательская часть.

• Проектно-конструкторская часть.

• Проектно-технологическая часть

• Доклад на защиту в формате .docx или .pptx

**1.9. Источники разработки.**

В настоящем документе использованы следующая литература и нормативные

документы:

• ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на

автоматизированные системы. Техническое задание на создание

автоматизированной системы»;

• Статья «ГОСТ 34.602-89 Информационная технология. Комплекс стандартов

на автоматизированные системы. Техническое задание на создание

автоматизированной системы» - https://internet-law.ru/gosts/gost/11254/

# 2. Научно – исследовательская часть

## **2.1 Постановка задачи проектирования**

Разрабатываемая база данных будет предназначена для реализации веб-приложения, позволяющего клиентам стоматологической клиники просматривать информацию о врачах, прайс-лист услуг, а также самостоятельно записываться на первичный прием к конкретному специалисту.

База данных будет содержать информацию о врачах, перечне и стоимости доступных услуг.

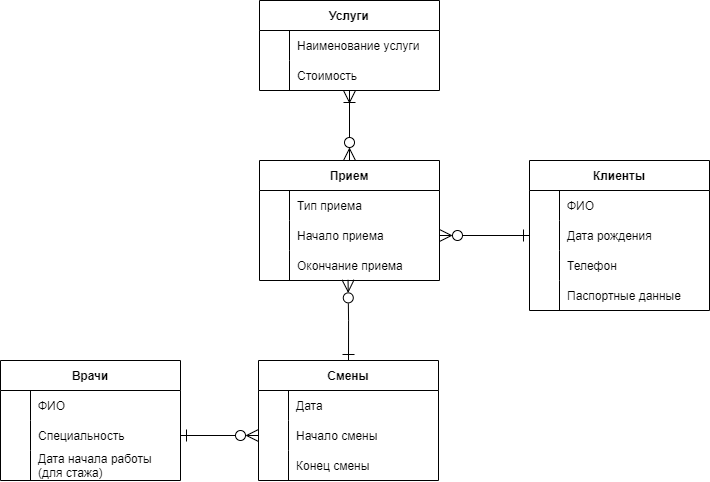
* Зарегистрированный пользователь может самостоятельно записаться на первичный прием.
* Администратор может записывать пользователей, а также заносить данные о врачах и их расписании в базу данных

## **2.2. Описание предметной области**

Клиенты приходят в стоматологическую клинику на консультацию или для получения медицинских услуг.

* Существует два типа пользователей – клиент клиники и администратор, выполняющий функции регистратуры
* Клиент записывается на первичный прием самостоятельно, выбирая специалиста и удобное для себя время
* На вторичный прием клиента записывает администратор, устанавливая продолжительность приема
* Вторичный прием включает в себя различные услуги, стоимость которых указана в прайс-листе
* Расписание работы врачей в клинике и прайс-лист услуг составляет и изменяет администратор
* Свои личные данные клиент заполняет самостоятельно

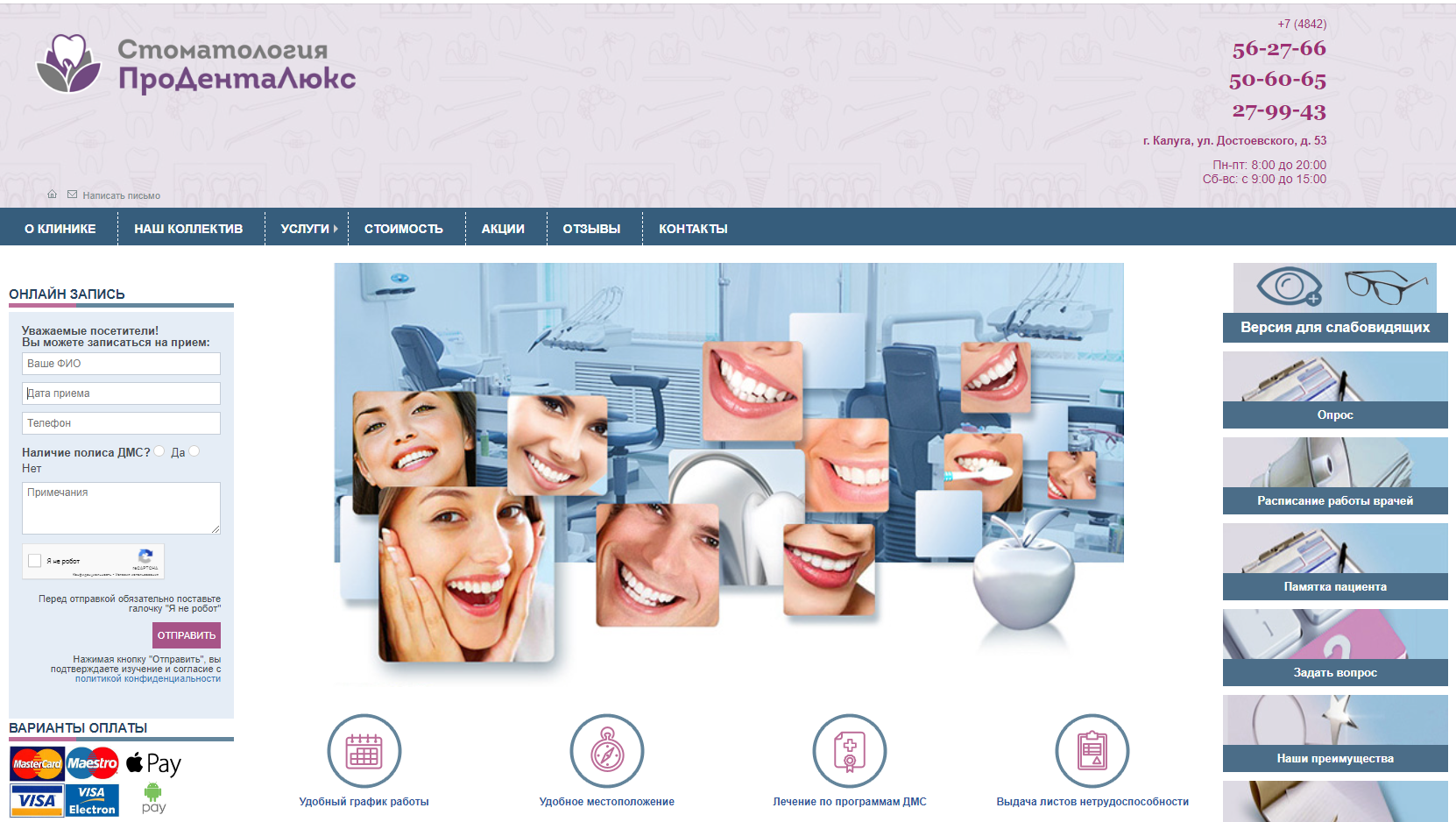
**Концептуальная схема базы данных:**



**Рисунок 1.** Концептуальная схема базы данных

## **2.3 Анализ аналоговых прототипов.**

### **«ПроДентаЛюкс»**



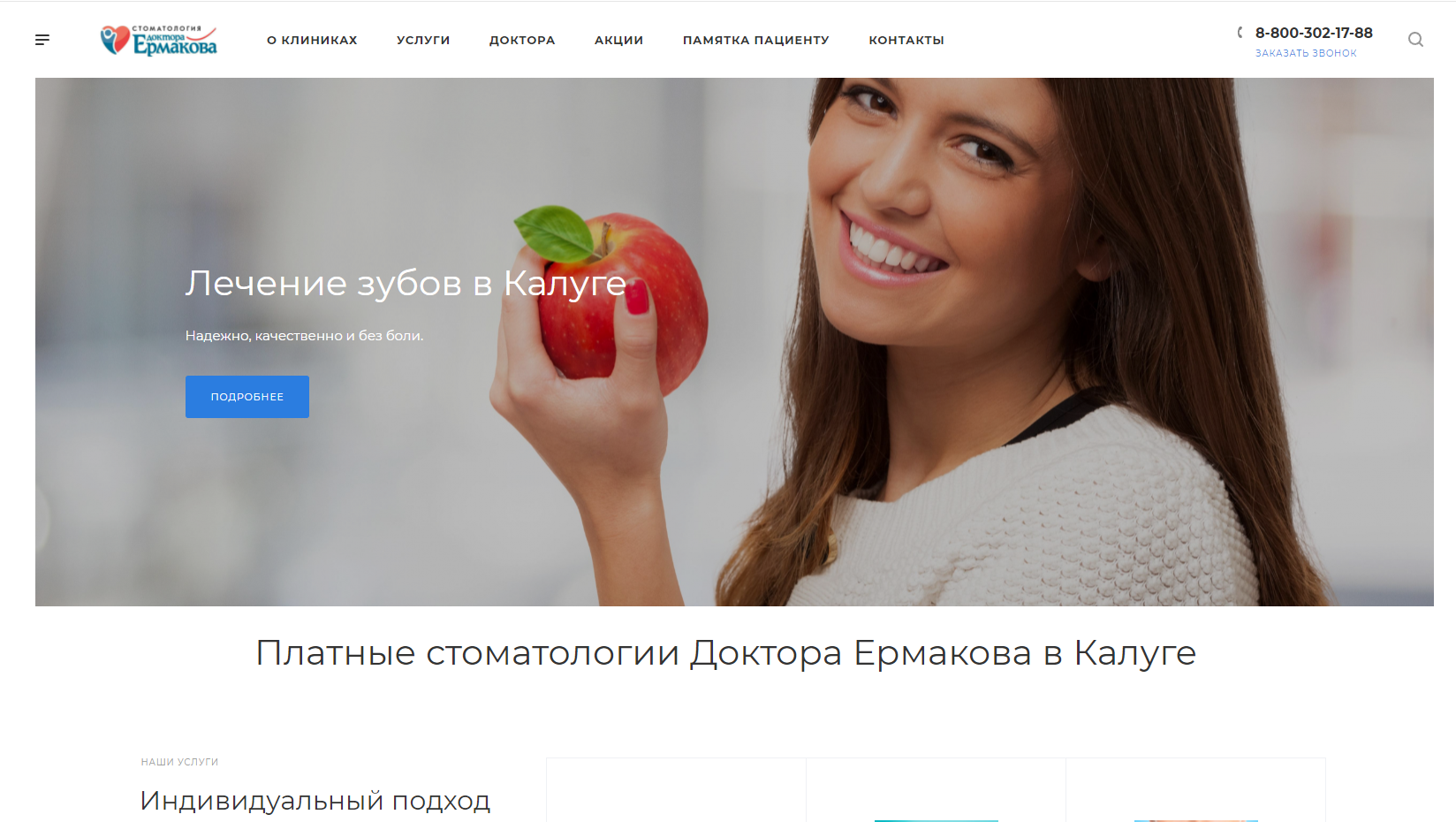
**Плюсы:**

* Возможность записи
* Наличие акций и скидок

**Минусы:**

* Отсутствие записи к конкретному врачу
* Отсутствие выбора времени записи

### **«Стоматология Доктора Ермакова»**



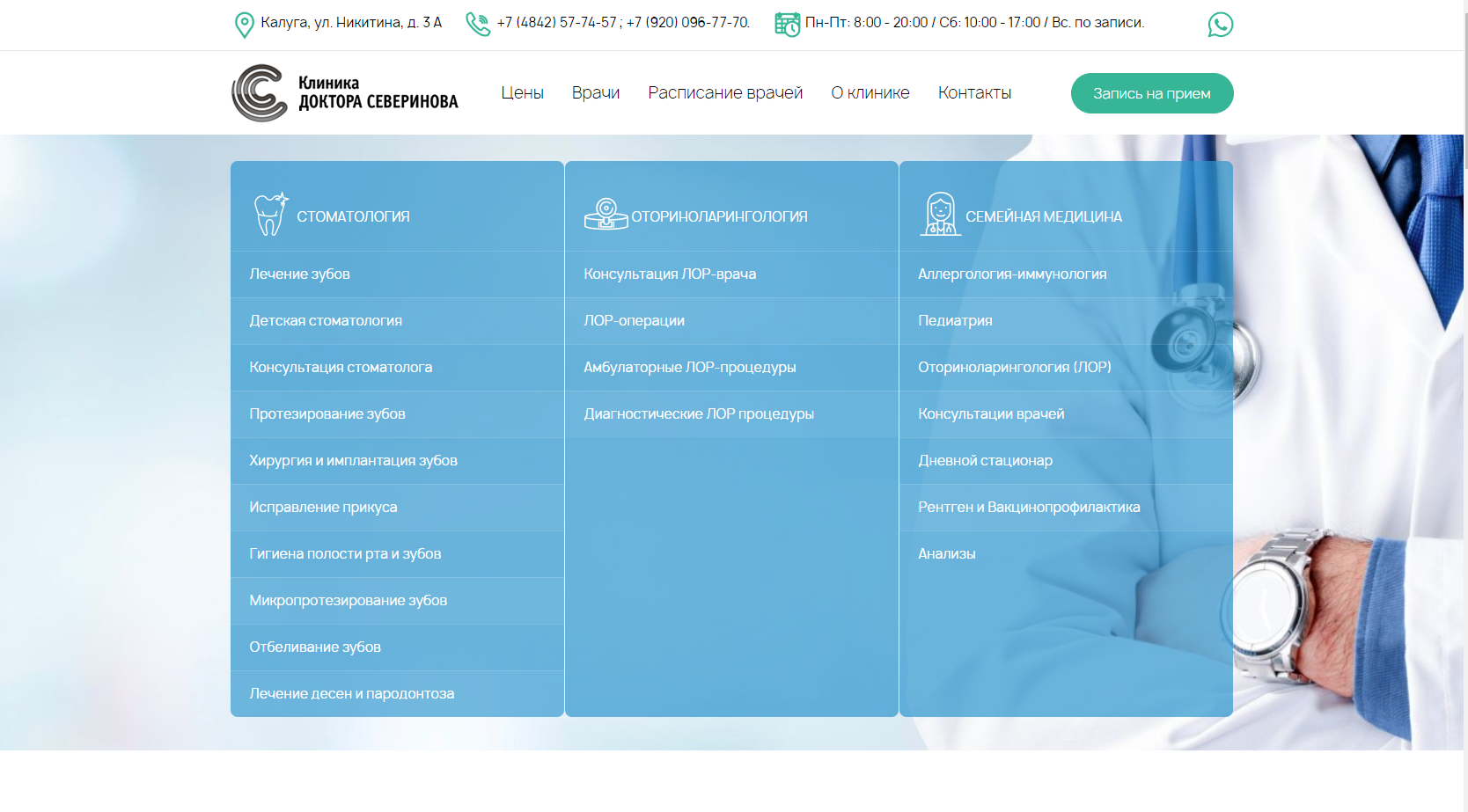
**Плюсы:**

* Удобный интерфейс
* Широкий ассортимент

**Минусы:**

* Отсутствие возможности записи
* Отсутствие информации о врачах

### **«Клиника Доктора Северинова»**



**Плюсы:**

* Возможность записи к конкретному врачу
* Большое количество информации о предоставляемых услугах и врачах

**Минусы:**

* Отсутствие выбора времени записи

Таким образом сайты имеют как различные достоинства, например, наличие акций и скидок, широкий ассортимент, так и недостатки, например, невозможность указания деталей записи. Поэтому целью данной работы является создание веб-приложения, позволяющего быстро и удобно записаться на прием в стоматологическую клинику к конкретному врачу и на определённое время.

## **2.4 Перечень задач, подлежащих решению в процессе разработки.**

1. Разработка компонента пользовательского интерфейса;
2. разработка клиентского компонента;
3. разработка серверного компонента;
4. тестирование разработанного приложения;
5. исправление выявленных ошибок.

## **2.5 Обоснование выбора инструментов и платформы для разработки.**

Для успешной разработки проекта, используется операционная система Windows 10. Windows 10 — [операционная система](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) для [персональных компьютеров](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и [рабочих станций](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F), разработанная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в рамках семейства [Windows NT](https://ru.wikipedia.org/wiki/Windows_NT).

Данная платформа предоставляет широкий набор инструментов для работы с ОС Windows, например, язык программирования Си с использованием базовых функций интерфейсов программирования приложений операционных систем Microsoft Windows – Windows API. WinApi – это набор функций, работающих под управлением ОС Windows. С помощью WinAPI можно создавать различные оконные процедуры, диалоговые окна, программы и даже игры, а также работать с операционной системой на прямую.

Согласно Statcounter.com, общий рейтинг операционных систем, показывает, что лидером является Windows, который установлен на 76,58% всех компьютеров, OS X — 18,93% и Linux — 1,62%. Платформа Windows является самой популярной и доступной, также для этой платформы создано много удобных средств, для облегчения выполнения проекта.

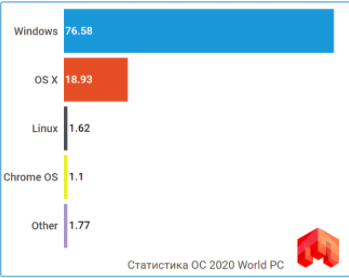


Рисунок 4. Статистика использования операционных систем в мире.

В качестве среды разработки используется Visual Studio Code. Интегрированная среда разработки Visual Studio Code — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio Code включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

Ниже перечислены некоторые популярные возможности Visual Studio Code, которые помогут вам повысить продуктивность разработки программного обеспечения.

* волнистые линии и [быстрые действия](https://docs.microsoft.com/ru-ru/visualstudio/ide/quick-actions?view=vs-2019);

Волнистые линии обозначают ошибки или потенциальные проблемы кода прямо во время ввода. Эти визуальные подсказки позволяют устранять проблемы немедленно и не ждать, пока ошибка будет обнаружена во время сборки или запуска программы. Если навести указатель мыши на волнистую линию, на экран будут выведены дополнительные сведения об ошибке. Кроме того, в поле слева может появляться значок лампочки с быстрыми действиями по устранению ошибки.

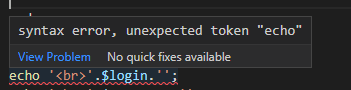


Рисунок 5. Волнистые линии и быстрые действия

* поиск в Visual Studio Code

Среда Visual Studio Code может показаться сложной, ведь там столько разных меню, параметров и свойств. Чтобы быстро находить функции интегрированной среды разработки и элементы кода, в Visual Studio Code представлен единый компонент поиска (**CTRL**+**Q**).

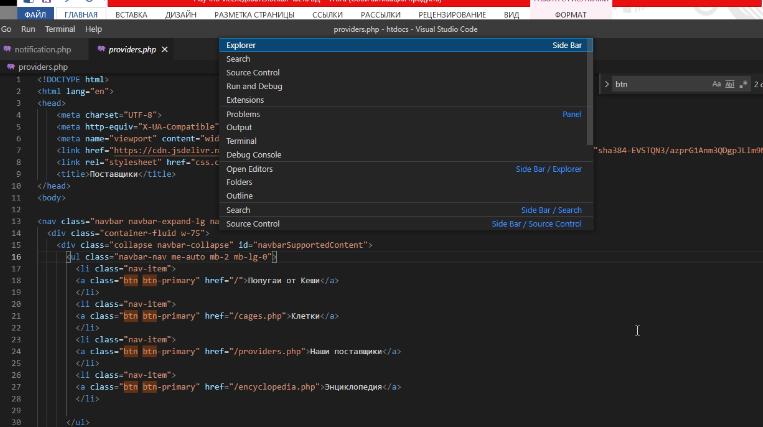


Рисунок 6. Поиск в Visual Studio Code

Согласно опросу, проведенного среди 3240 разработчиков в феврале 2018 организацией Standard, разработчики выделили такие средства разработки, как наиболее часто используемые:

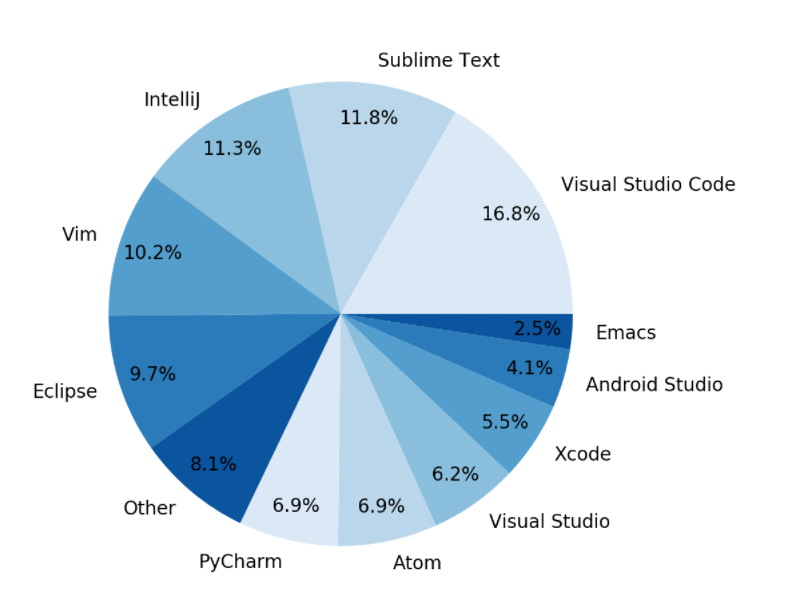


Рисунок 7. Статистика использования платформ для разработки систем в мире.

Первое место среди IDE у Visual Studio Code с 16.8% использующих его разработчиков.

Также, Visual Studio Code обладает хорошим отладчиком, с помощью которого можно легко отследить ошибки в коде и без проблем их исправить, позволяет удобно и быстро создавать проекты и работать с ними.

По сравнению со средой разработки Eclipse, Visual Studio Code имеет более простой интерфейс в использовании.

Для разработки проекта можно использовать следующие инструменты:

* язык гипертекстовой разметки HTML;

HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

Элементы HTML являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты такие как интеракивная веб-формы могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. Такие теги, как <img /> и <input />, напрямую вводят контент на страницу. Другие теги, такие как <p>, окружают и оформляют текст внутри себя и могут включать другие теги в качестве подэлементов. Браузеры не отображают HTML-теги, но используют их для интерпретации содержимого страницы.

В HTML можно встроить программый код на языке программирования JavaScript, для управления поведением и содержанием веб-страниц. Также включение CSS в HTML описывает внешний вид и макет страницы.

* каскадные таблицы стилей CSS;

CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана) или при выводе устройствами, использующими шрифт Брайля.

* язык программирования PHP;

PHP (англ. PHP: Hypertext Preprocessor — «PHP: препроцессор гипертекста»; первоначально PHP/FI (Personal Home Page / Form Interpreter), а позже названный Personal Home Page Tools — «Инструменты для создания персональных веб-страниц») — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

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

В области веб-программирования, в частности серверной части, PHP — один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET).

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств и дополнительных модулей для разработки веб-приложений. Основные из них:

* автоматическое извлечение POST- и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
* взаимодействие с большим количеством различных систем управления базами данных через дополнительные модули (MySQL, MySQLi, SQLite, PostgreSQL, Oracle Database (OCI8), Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird и InterBase, Paradox File Access, MaxDB, интерфейс PDO, Redis);
* автоматизированная отправка HTTP-заголовков;
* работа с HTTP-авторизацией;
* работа с cookies и сессиями;
* работа с локальными и удалёнными файлами, сокетами;
* обработка файлов, загружаемых на сервер;
* работа с XForms.

По состоянию на 2010-е годы используется сотнями тысяч разработчиков; согласно рейтингу корпорации TIOBE, базирующемуся на данных поисковых систем, в мае 2016 года PHP находился на 6 месте среди языков программирования.

Входит в LAMP — распространённый набор программного обеспечения для создания и хостинга веб-сайтов (Linux, Apache, MySQL, PHP).

Среди сайтов, использующих PHP — Facebook, Wikipedia, Yahoo!, Baidu.

* база данных MySql;

Первое место среди IDE у Visual Studio Code с 16.8% использующих его разработчиков.

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. 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.

MySQL имеет API и коннекторы для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk, Компонентный Паскаль, Tcl и Lua, библиотеки для языков платформы .NET.

**3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ**

**3.1 Разработка структуры приложения.**

Для разработки приложения использовалась среда Microsoft Visual Studio Code, включающая в себя широкий набор функциональных инструментов.

Для представления принципа работы приложения ниже приведена ее

UML диаграмма вариантов использования (рис. 7).

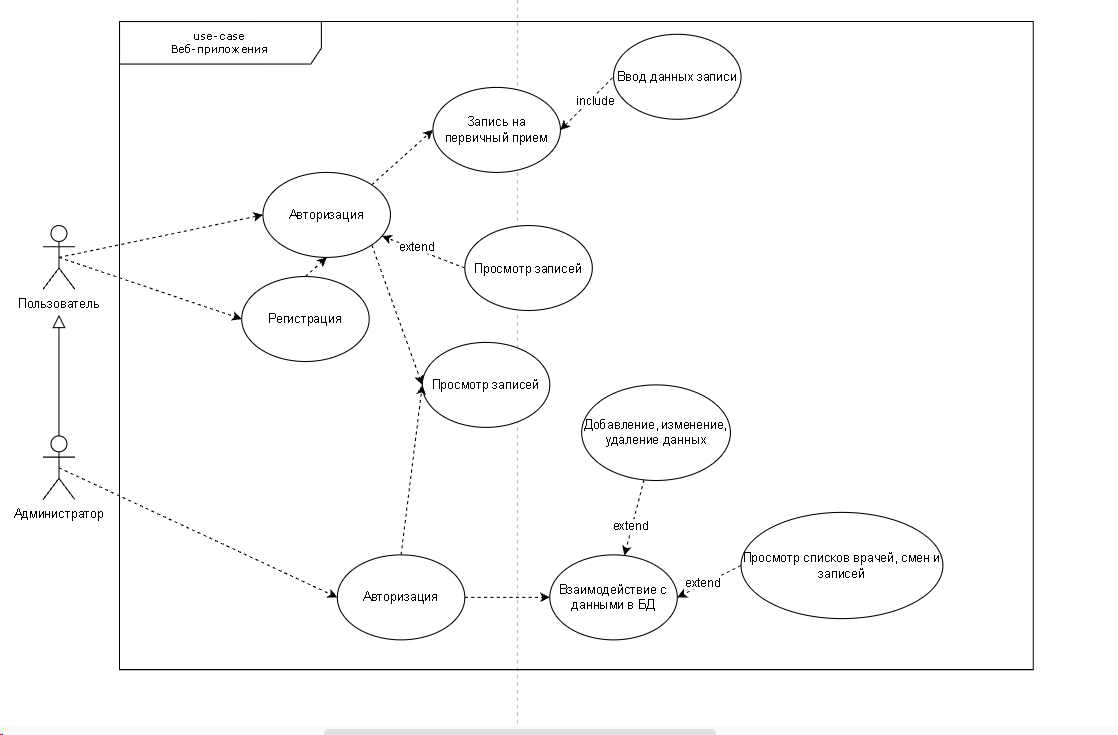


Рисунок 7 - Диаграмма использования веб-приложения

При запуске веб-приложения пользователю будет предложено зарегистрироваться или войти в аккаунт. Без этого пользователь не сможет просматривать информацию или записаться на прием.

В любой момент работы с веб-приложением, пользователь может выйти из него.

Данные на сайте добавляются, удаляются и обновляются администратором. У администратора есть все необходимые для этого возможности.

Общее назначение программного средства – предоставление возможности просмотра информации о врачах и записях.

Реализуемая задача состоит в том, чтобы хранение всей информации, необходимой для существования веб-приложения осуществлялось в базе данных, а также производилось удобное взаимодействие с пользовательским интерфейсом.

**3.2.Разработка архитектуры приложения**

Данная система представляет собой Web-приложение. Это тип программ, построенных по архитектуре «клиент-сервер». Особенность их заключается в том, что само Web-приложение находится и выполняется на сервере - клиент при этом получает только результаты работы. Работа приложения основывается на получении запросов от пользователя (клиента), их обработке и выдачи результата. Передача запросов и результатов их обработки происходит через Интернет.

За счет наличия исполняемой части, Web-приложения способны выполнять практически те же операции, что и обычные Windows-приложения, с тем лишь ограничением, что код исполняется на сервере, в качестве интерфейса системы выступает браузер, а в качестве среды, посредством которой происходит обмен данными — Интернет.

К наиболее типичным операциям, выполняемым Web приложениями, относятся:

•прием данных от пользователя и сохранение их на сервере;

•выполнение различных действий по запросу пользователя: извлечение данных из базы данных (БД), добавление, удаление, изменение данных в БД, проведение сложных вычислений;

•отображение постоянно изменяющейся оперативной информации и т.д.

При разработке данного веб-приложения в курсовой работе было принято решение использовать монолитную архитектуру.

В программной инженерии монолитная модель относится к единой неделимой единице. Концепция монолитного программного обеспечения заключается в том, что различные компоненты приложения объединяются в одну программу на одной платформе. Обычно монолитное приложение состоит из базы данных, клиентского пользовательского интерфейса и серверного приложения. Все части программного обеспечения унифицированы, и все его функции управляются в одном месте. Компоненты монолитного программного обеспечения взаимосвязаны и взаимозависимы.

Преимущества использования монолитной архитектуры:

* большим преимуществом является более простая реализация. В монолитной архитектуре можно быстро начать реализовывать свою бизнес-логику, вместо того чтобы тратить время на размышления о межпроцессном взаимодействии, кроме того, все действия выполняются с одним каталогом, что упрощает развертывание;
* улучшенная производительность. Если учитывать, что приложения были собраны правильно, то одно и то же приложение при монолитной архитектуре будет более производительным, чем, например, при микросервисной. Это обеспечивается единым кодом программы и работой из «одного» места.
* при необходимости изменить элементы программы не нужно вносить изменения по отдельности в разных местах — все делается в одном месте.

В случае данного веб-приложения все исходные «.php» файлы имеют примерно одинаковую структуру.

Если взглянуть на структуру начальной страницы (такая структура присуща всем страницам приложения), после строчки «<?php» происходит работа на серверной части веб-приложения, идет прямое обращение к БД (запрос), извлекаются необходимые данные и при помощи ключевого слова «echo» полученные данные отправляются обратно на клиентскую часть веб-приложения в виде HTML кода( рисунок 8). Никакие дополнительные методы или функции не создаются и не используются.



Рисунок 8. Пример работы серверной части.

**3.3 Разработка интерфейса взаимодействия пользователя с системой**

Для удобного взаимодействия пользователя был разработан следующий интерфейс.

Если пользователь не авторизовался, ему будет предложено создать аккаунт или войти в уже существующий. Неавторизированный пользователь не может записаться на прием к врачу. Для этого необходимо либо зарегистрироваться под новым пользователем (рис. 9), либо зайти уже под существующим (рис. 10). После авторизации, пользователь сможет записаться на прием (рис. 11).

Для обновления информации на сайте, необходим администратор с возможностями управления веб-приложением. У администратора есть доступ к страницам администрирования, при помощи нажатий на кнопки, доступные только ему (рис. 12). Он обладает рядом возможностей для успешного управления сайтом, а именно просмотр, добавление и удаление записей, смен и врачей.

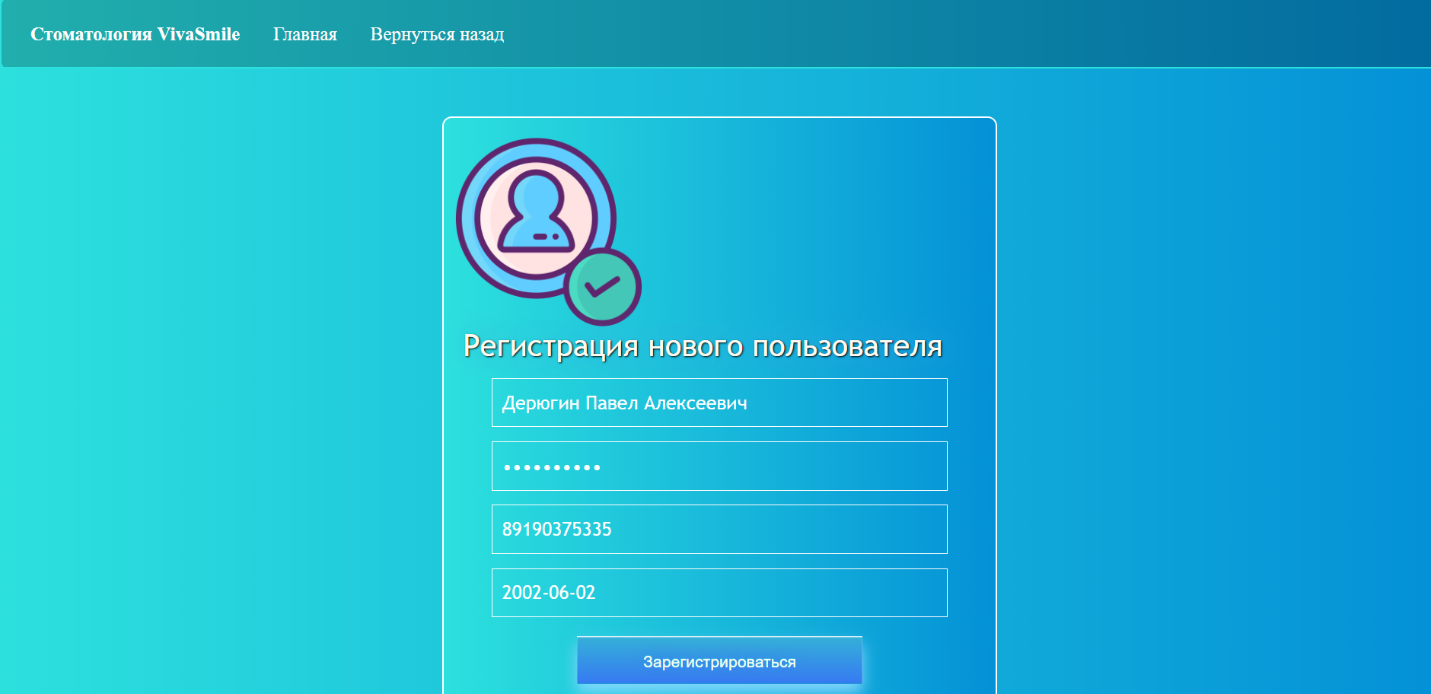


Рисунок 9 – Регистрация нового пользователя

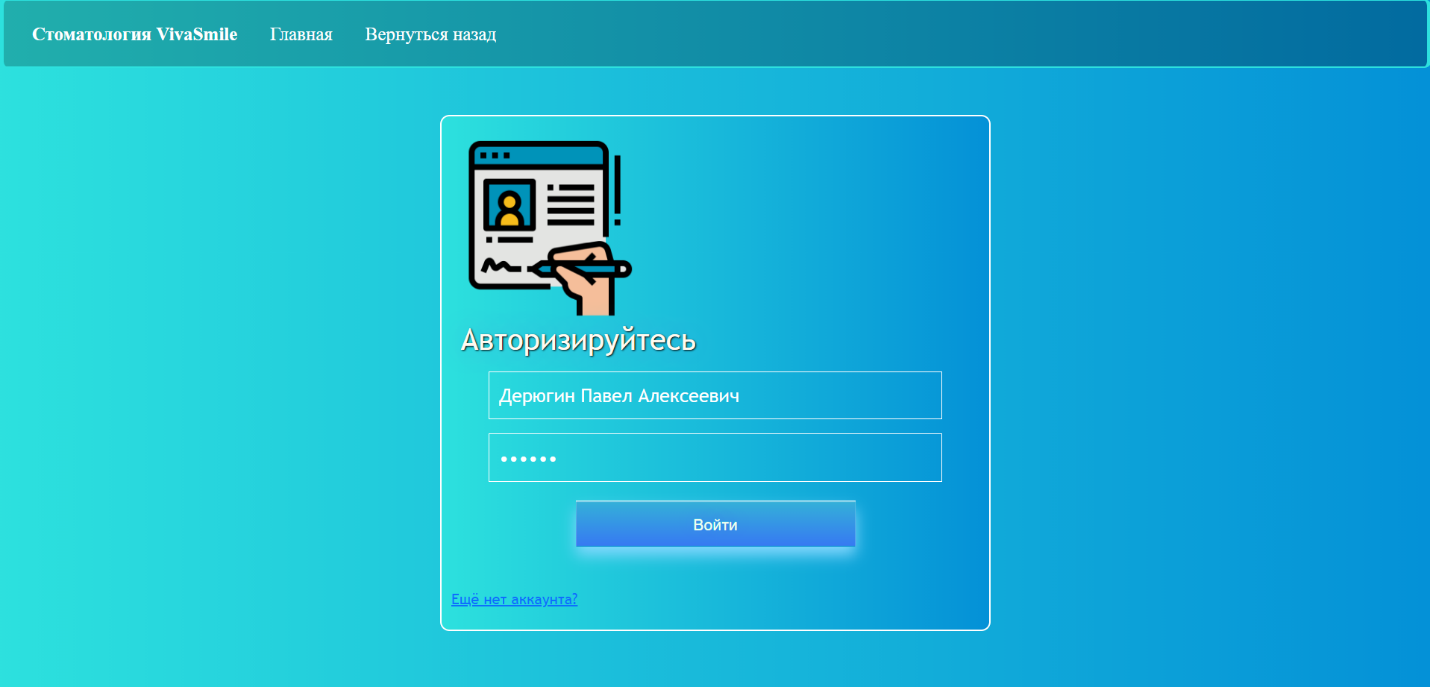


Рисунок 10 – Авторизация пользователя

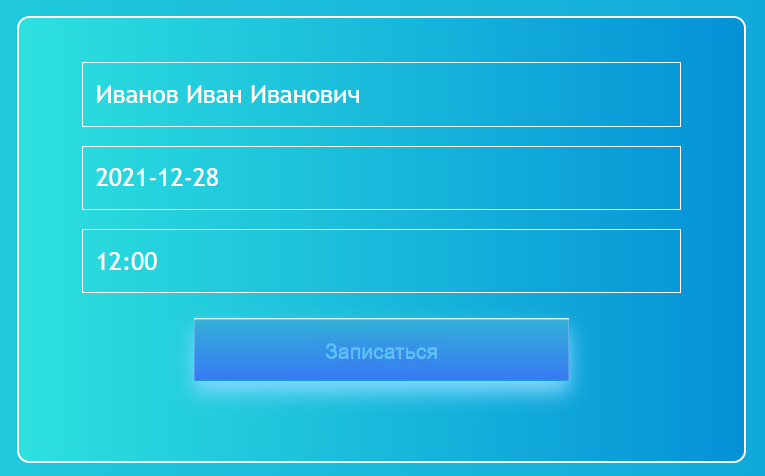


Рисунок 11 – Запись на прием

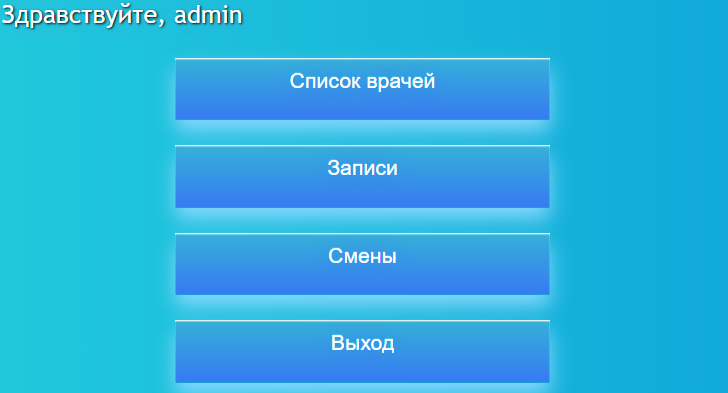


Рисунок 12 – Возможности администратора

**4.ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ**

* 1. **Проектирование начального и тестового наполнения базы данных.**

Проектирование наполнения базы данных следующее:

1) С помощью следующего SQL запроса была создана таблица `users`:

CREATE TABLE `users` (

`user\_id` int(11) UNSIGNED NOT NULL,

`name` varchar(50) NOT NULL,

`password` varchar(32) NOT NULL,

`phone` bigint(15) UNSIGNED NOT NULL,

`admin` tinyint(1) NOT NULL DEFAULT '0',

`birth\_date` date DEFAULT NULL

)

В результате в базе данных через панель управления СУБД MySQL появилась таблица, представленная на Рисунке 4.1:

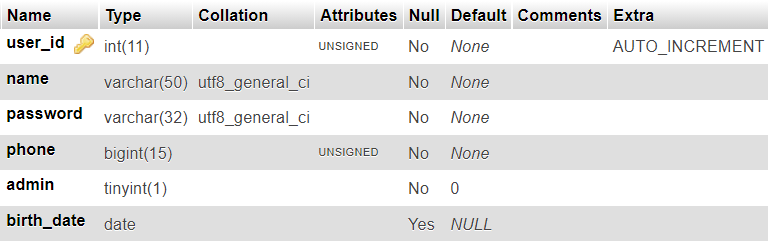


Рисунок 4.1. Таблица `users`

Таблица `users`, заполненная данными, представлена на Рисунке 4.2.

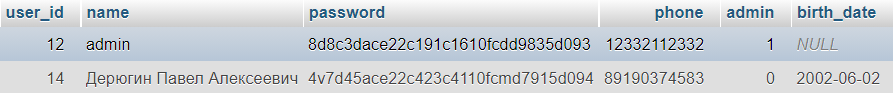


Рисунок 4.2. Заполненная данными таблица `users`.

2) С помощью следующего SQL запроса была создана таблица `doctors`:

CREATE TABLE `doctors` (

`doctor\_id` int(11) UNSIGNED NOT NULL,

`name` varchar(60) NOT NULL,

`specialization` varchar(30) NOT NULL,

`start\_work` timestamp NULL DEFAULT NULL,

`active` tinyint(1) DEFAULT '1'

)

В результате в базе данных через панель управления СУБД MySQL появилась таблица, представленная на Рисунке 4.3:

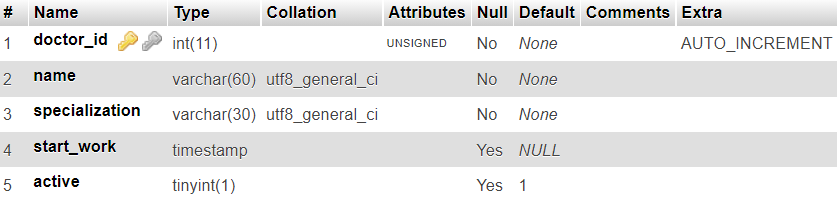


Рисунок 4.3. Таблица ` doctors `

Таблица `doctors`, заполненная данными, представлена на Рисунке 4.4.

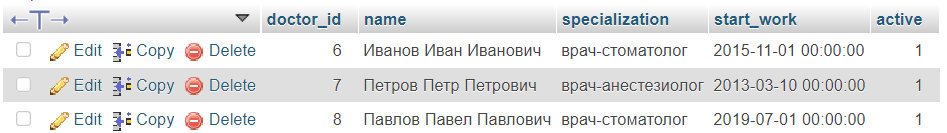


Рисунок 4.4. Заполненная данными таблица ` doctors `.

3) С помощью следующего SQL запроса была создана таблица `shifts`:

CREATE TABLE `shifts` (

`shift\_id` int(11) UNSIGNED NOT NULL,

`doctor\_id` int(11) UNSIGNED NOT NULL,

`date` date NOT NULL,

`start` timestamp NULL DEFAULT NULL,

`finish` timestamp NULL DEFAULT NULL

)

В результате в базе данных через панель управления СУБД MySQL появилась таблица, представленная на Рисунке 4.5:

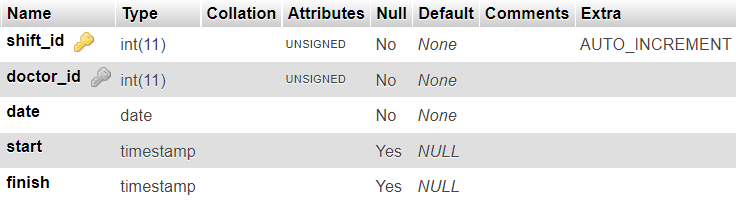


Рисунок 4.5. Таблица `shifts`

Таблица `shifts`, заполненная данными, представлена на Рисунке 4.6



Рисунок 4.6. Заполненная данными таблица `shifts`.

4) С помощью следующего SQL запроса была создана таблица `services`:

CREATE TABLE `services` (

`service\_id` int(11) UNSIGNED NOT NULL,

`service\_name` varchar(40) NOT NULL,

`price` int(10) UNSIGNED NOT NULL

)

В результате в базе данных через панель управления СУБД MySQL появилась таблица, представленная на Рисунке 4.7:

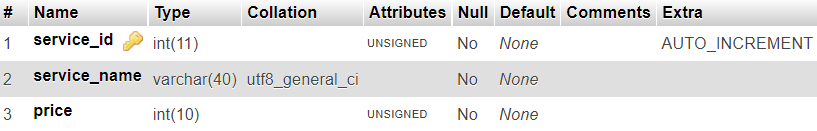


Рисунок 4.7. Таблица `services`

Таблица `services`, заполненная данными, представлена на Рисунке 4.8

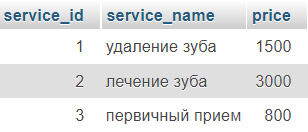


Рисунок 4.8. Заполненная данными таблица `services`.

5) С помощью следующего SQL запроса была создана таблица `session\_services`:

CREATE TABLE `session\_services` (

`ss\_id` int(11) UNSIGNED NOT NULL,

`session\_id` int(11) UNSIGNED NOT NULL,

`service\_id` int(11) UNSIGNED NOT NULL

)

В результате в базе данных через панель управления СУБД MySQL появилась таблица, представленная на Рисунке 4.9:

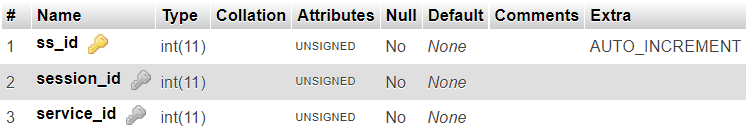


Рисунок 4.9. Таблица `session\_services`

Таблица `session\_services`, заполненная данными, представлена на Рисунке 4.10

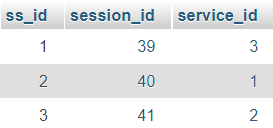


Рисунок 4.10. Заполненная данными таблица `session\_services`.

6) С помощью следующего SQL запроса была создана таблица `sessions`:

CREATE TABLE `sessions` (

`session\_id` int(11) UNSIGNED NOT NULL,

`user\_id` int(11) UNSIGNED NOT NULL,

`shift\_id` int(11) UNSIGNED NOT NULL,

`type` enum('primary','secondary') NOT NULL DEFAULT 'primary',

`start` timestamp NULL DEFAULT NULL,

`finish` timestamp NULL DEFAULT NULL

)

В результате в базе данных через панель управления СУБД MySQL появилась таблица, представленная на Рисунке 4.11:

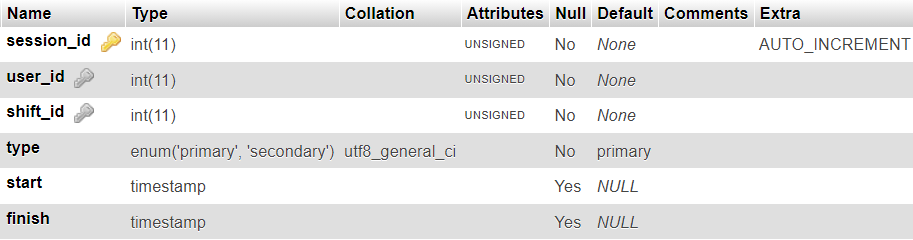


Рисунок 4.11. Таблица `sessions`

Таблица `sessions`, заполненная данными, представлена на Рисунке 4.12

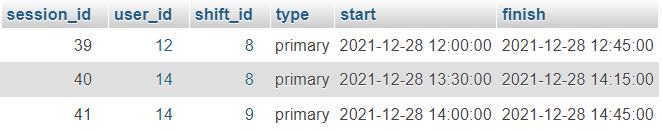


Рисунок 4.12. Заполненная данными таблица `sessions`.

Таблица 1. Основная надпись тест-кейса номер 1

|  |  |
| --- | --- |
| **Номер тест-кейса** | 1 |
| **Заголовок** | Осуществить тестирование функционала приложения, заявленного в техническом задании. |
| **Предусловие** | Запустить приложение |

Таблица 2. Тест-кейс номер 1(Тестирование «Входа пользователя в приложение»)

|  |  |  |
| --- | --- | --- |
| **Номер шага** | **Действие** | **Ожидаемый результат** |
| 1 | Нажать на кнопку «Войти», в навигационной панели. | Должно открыться окно авторизации. |
| 2 | В поля «Введите ФИО» и «Введите пароль» ввести ФИО и пароль. | В полях отобразятся введенные значения. |
| 3 | Нажать кнопку «Войти» в форме «Авторизация». | Осуществится вход под пользователем. |

Произведем тестирование приложения «Стоматология VivaSmile», по указанному тест-кейсу номер 1.

**Шаг номер 1:**

По нажатию на кнопку «Войти», в навигационной панели, должно произойти перемещение на страницу авторизации (Рисунок 4.13).

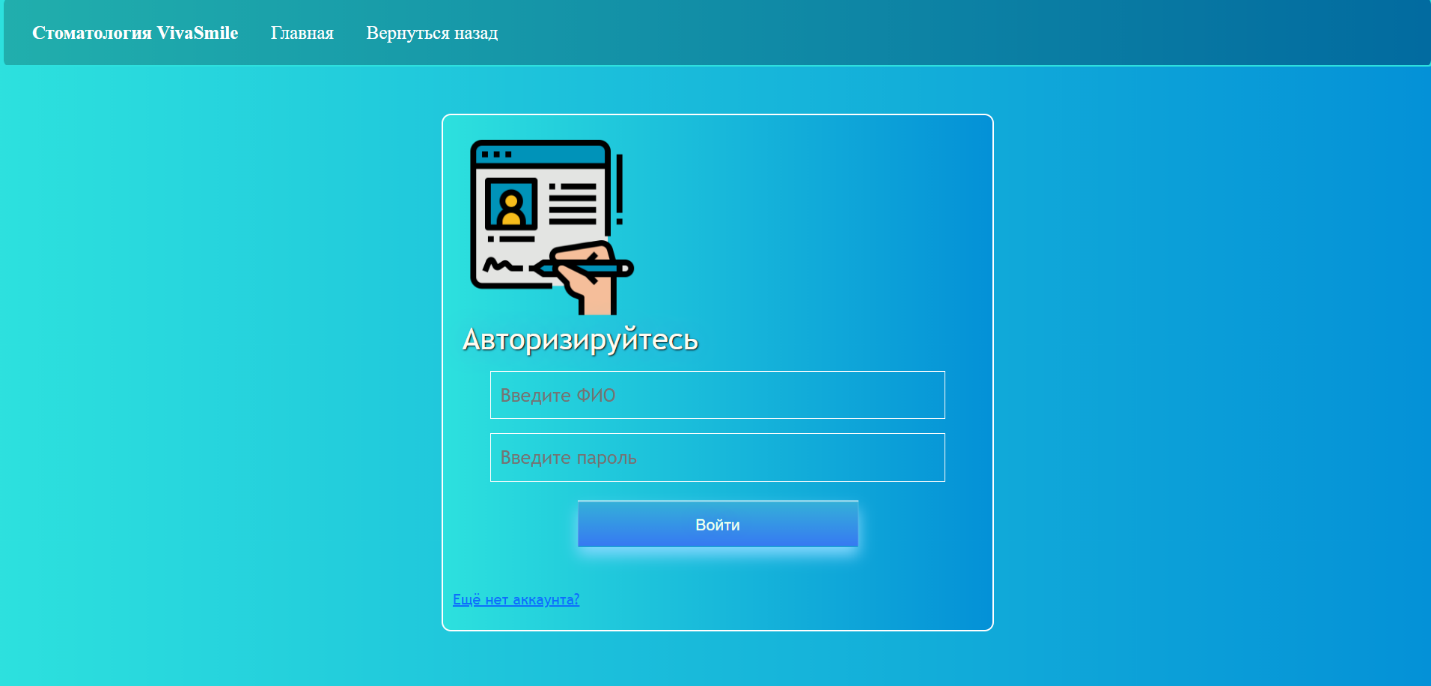


Рисунок 4.13 – Тестирование приложения, шаг номер 1

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 2:**

При выборе полей «Введите ФИО» и «Введите пароль» и вводе данных, должны отобразится введённое значения (Рисунок 4.14).

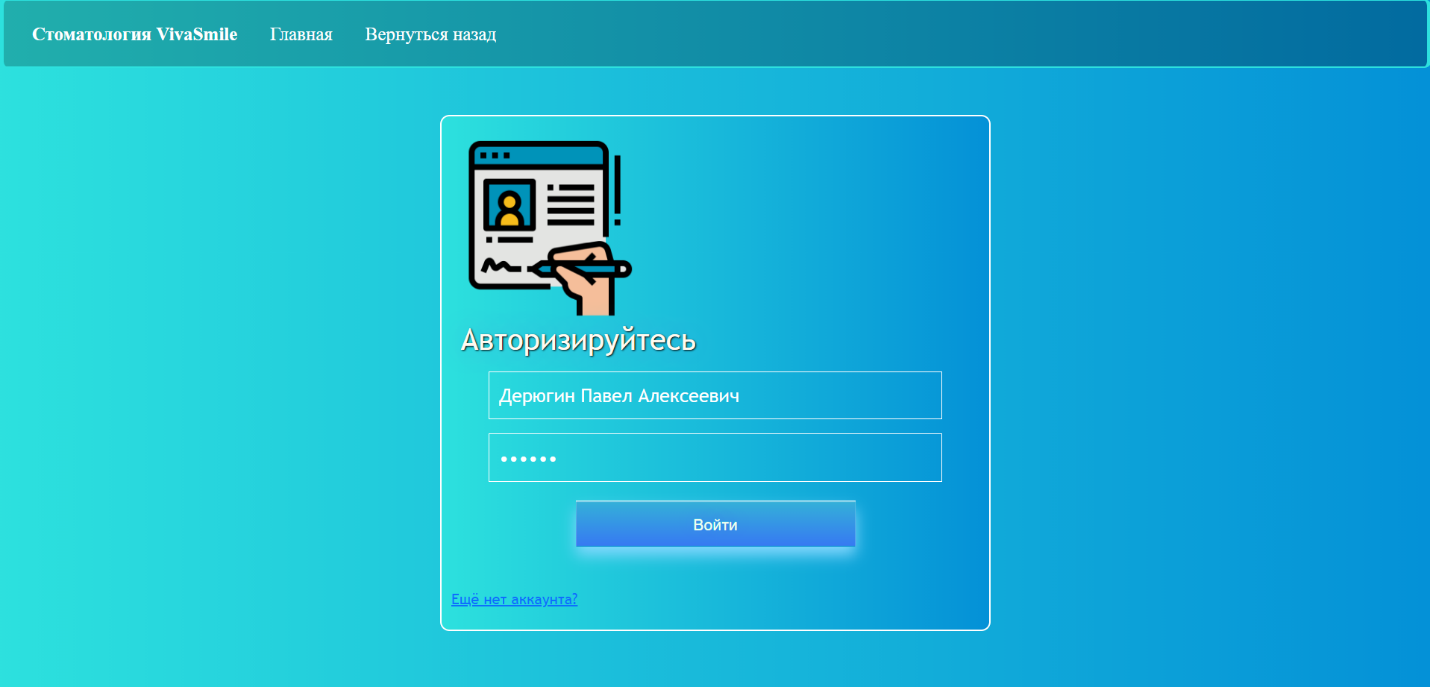


Рисунок 4.14 – Тестирование приложения, шаг номер 2

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 3:**

Нажать на кнопку «Войти», должен произойти вход под пользователем. (Рисунок 4.15).

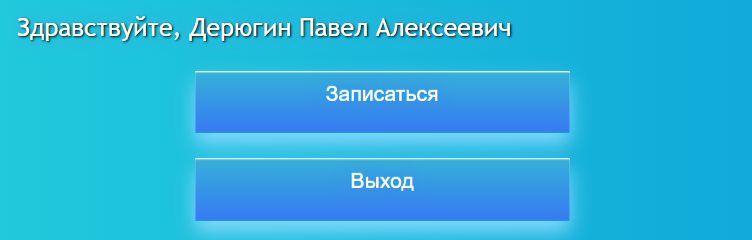


Рисунок 4.15 – Тестирование приложения, шаг номер 4

Вывод: полученный результат совпадает с ожидаемым.

Таблица 3. Тест-кейс номер 1(Тестирование «Записи на прием»)

|  |  |  |
| --- | --- | --- |
| **Номер шага** | **Действие** | **Ожидаемый результат** |
| 1 | Нажать на кнопку «Записаться», в навигационной панели. | Должен осуществится переход на страницу «Список записей». |
| 2 | В поля «Выберите врача», «Выберите дату» и «Выберите время» ввести ФИО врача, дату и время записи. | В полях отобразятся введенные значения. |
| 3 | Нажать на кнопку «Записаться». | Запись добавится в список. |

**Шаг номер 1:**

Нажать на кнопку «Записаться», должен осуществится переход на страницу «Список записей». (Рисунок 4.16).

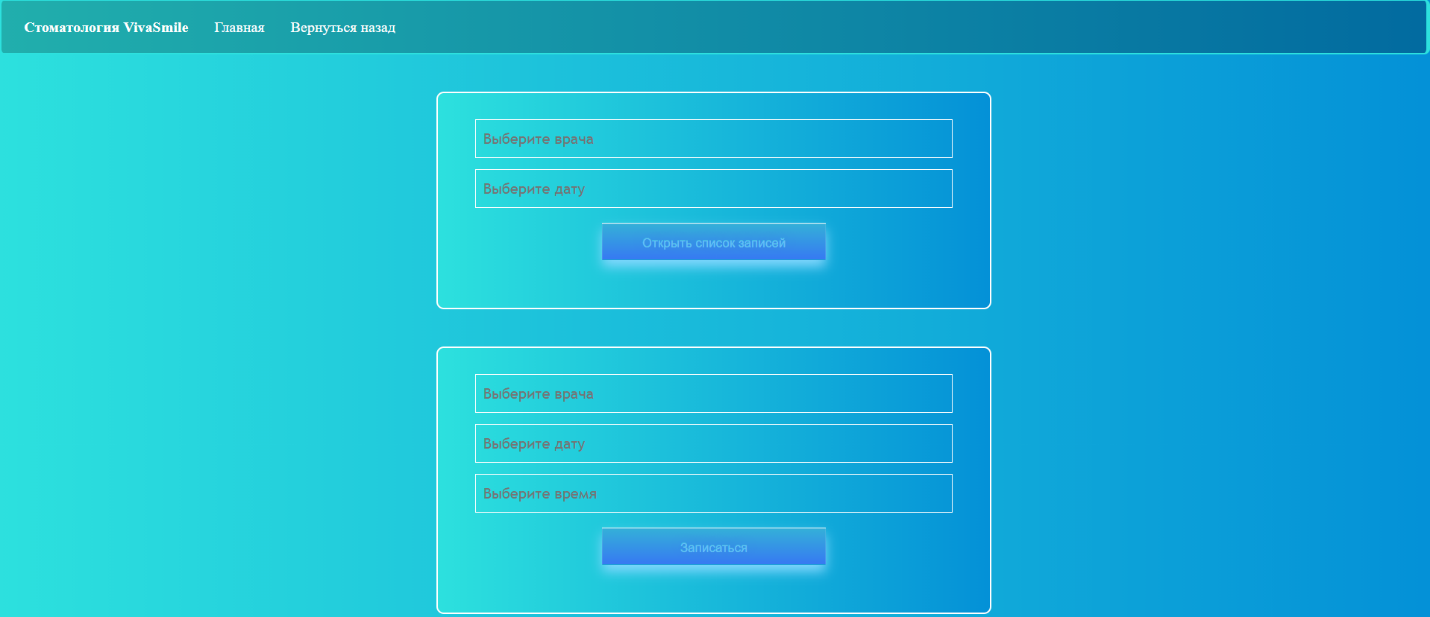


Рисунок 4.16 – Тестирование приложения, шаг номер 1

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 2:**

В поля «Выберите врача», «Выберите дату» и «Выберите время» ввести ФИО врача, дату и время записи, в полях должны отобразиться данные. (Рисунок 4.17).

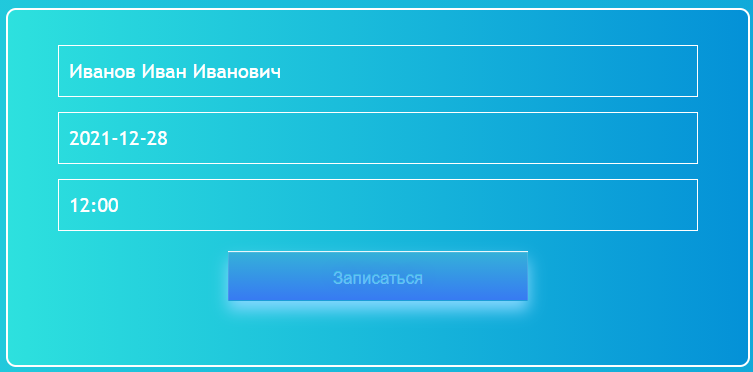


Рисунок 4.17 – Тестирование приложения, шаг номер 2

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 3:**

Нажать на кнопку «Записаться», запись должна появиться в списке. (Рисунок 4.18).



Рисунок 4.18 – Тестирование приложения, шаг номер 3

Вывод: полученный результат совпадает с ожидаемым.

**4.2. Разработка руководства администратора**

Для того, чтобы пользоваться веб-приложением, необходимо сделать распаковку архива с приложением (рисунок 32).

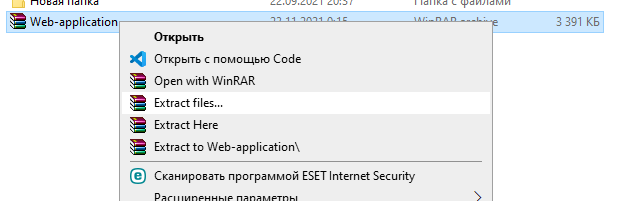


Рисунок 32 - Установка шаг №1

Необходимо произвести установку приложения MAMP для успешной компоновки файлов и работы веб-приложения (рисунок 33).

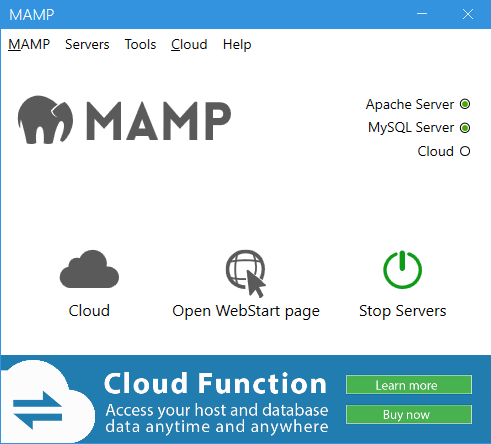
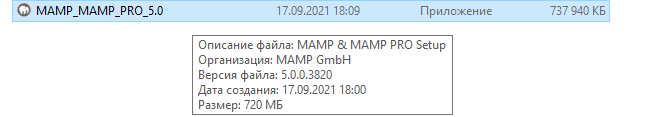


Рисунок 33 - Установка шаг №2

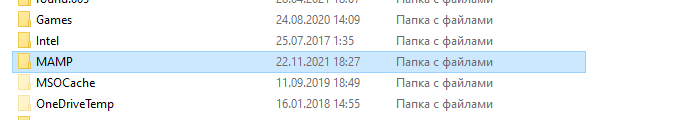
Далее переходим в папку, где было установлено приложение MAMP (рис. 34).

Рисунок 34 - Установка шаг №3

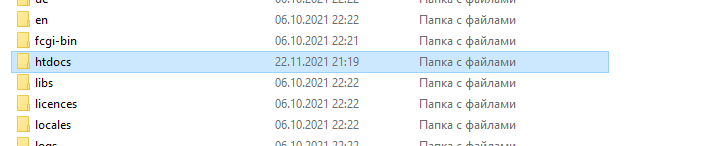
После находим папку «htdocs» и перемещаем содержимое архива в эту папку (рис. 35).

Рисунок 35 - Установка шаг №4

Далее следует импортировать базу данных со всей информацией.

<http://localhost/phpMyAdmin/server_databases.php?server=1> Переходим по этому адресу, создаем БД с именем service, жмем на кнопку «Импорт» и там выбираем файл из распакованного архива.

Веб-приложение готово к работе! Все что осталось сделать – это открыть браузер и перейти по адресу <http://localhost/> (рисунок 36).

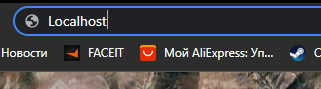


Рисунок 36 - Установка шаг №5

После исполнения вышеперечисленных действий, веб-приложение готово к работе (рис. 37).

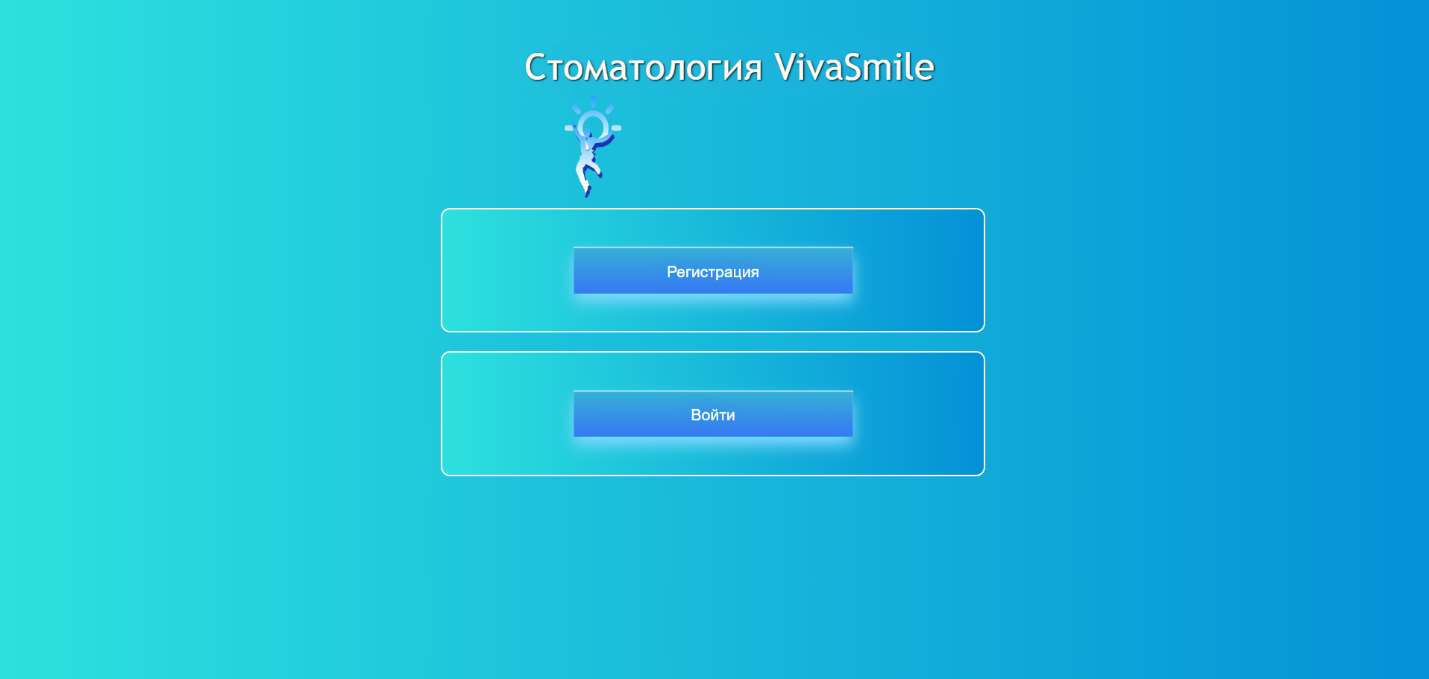


Рисунок 37 - Установка шаг №6

**4.3. Разработка руководства пользователя**

Порядок работы:

Работа с веб-приложением происходит при помощи перехода по ссылке <http://localhost/>.

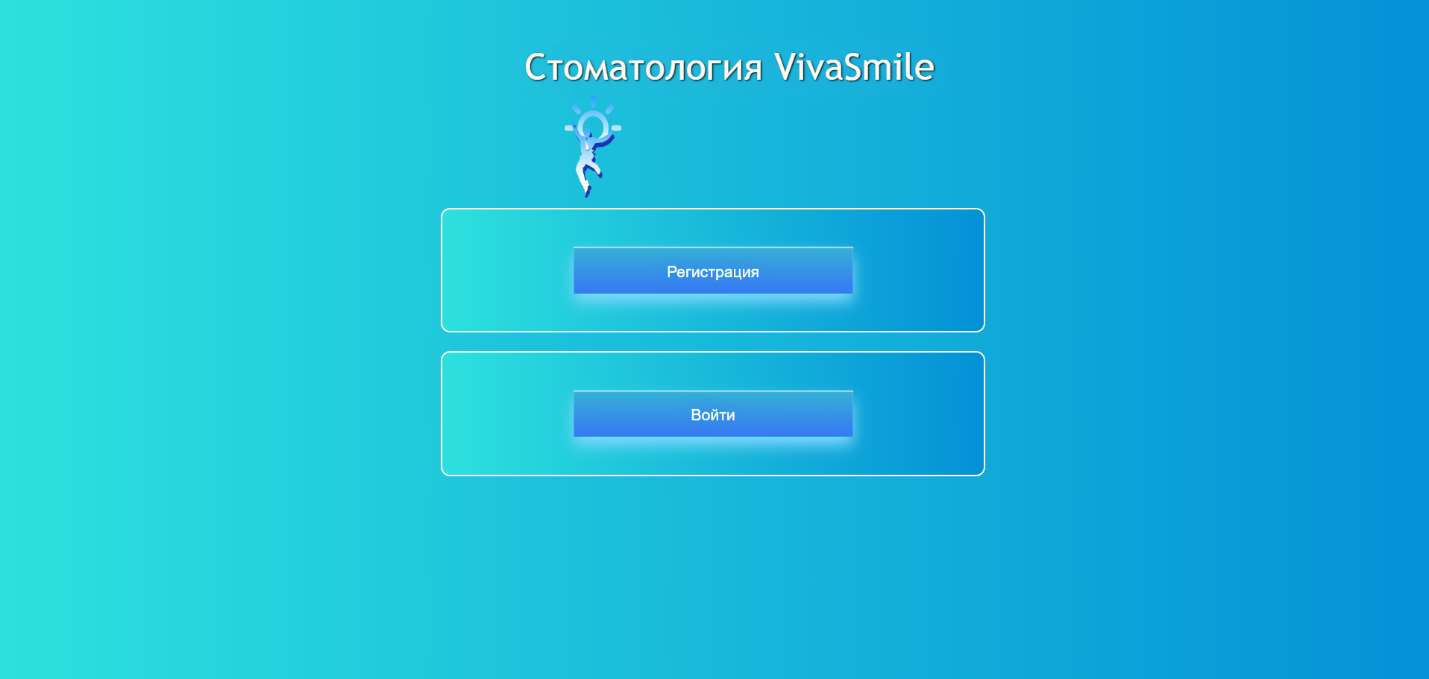
После запуска открывается страница с пользовательским интерфейсом (рисунок.38). 

Рисунок 38 – Веб-приложение после открытия

Далее пользователь выбирает сам, что ему нужно сделать. Он может: посмотреть информацию о меню, поставщиках, произвести авторизацию или регистрацию, оформить заказ.

**Заключение**

При выполнении курсовой работы на тему «Разработка веб-приложения для стоматологической клиники «VivaSmile»» была исследована и описана предметная область, проведен анализ аналогов данной системы, а также выбор инструментов и платформы для разработки.

Проведен анализ объектов автоматизации и разработаны методы решения технических задач. Также была разработана и реализована структура базы данных, серверная часть приложения и веб-интерфейс системы. База данных была наполнена тестовыми данными.

Результатом данной курсовой работы является рабочее веб-приложение «Стоматология VivaSmile».

**Список использованных источников**

1. Арно Лоре Проектирование веб-API / Пер. с англ. Д. А. Беликова.– М.:

ДМК Пресс, 2020.– 440 с

2. Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова. - СПб.:

Питер, 2013. - 240 c.

3. Tom Butler и Kevin Yank. PHP & MySQL Novice to Ninja. 2017 – 450 c.

4. <https://www.php.net/>

5. Ревунков Г.И., Ковалева Н.А., Силантьева Е.Ю. Проектирование баз

данных. [Электронный ресурс] – МГТУ им. Н.Э.Баумана, 2018. – 48 с. Режим

доступа: https://e.lanbook.com/reader/book/103499/

6. https://dev.mysql.com/

7. Руководство по программированию на JavaScript [Электронный ресурс]

– режим доступа: https://developer.mozilla.org/ru/docs/Web/JavaScript/

8. https://dev.to/

9. Learning JavaScript Design Patterns — Addy Osmani 2020 – 439c