**ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ

**ДИПЛОМНЫЙ ПРОЕКТ**

по специальности СПО 09.02.04 «Информационные системы (по отраслям)»

по программе базовой подготовки

«Разработка информационной системы для парикмахерской»

Студент гр. Т9-ИС-17-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Кузнецов С.В.)

(подпись)

Руководитель проекта:

Преподаватель СПО, к.т.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Ведищев В.В.)

(подпись)

Дипломный проект рассмотрен

и допущена к защите в ИЭК «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20 \_\_ г.

и.о. заведующего кафедрой АСУ, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_ (В.А. Алексеев)

Липецк 2021

ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

|  |  |
| --- | --- |
| Университетский колледж | Зав. кафедрой Алексеев В.А. |
| Кафедра АСУ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ

ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студенту Кузнецову Степану Владимировичу группы Т9-ИС-17

Направление (специальность) Информационные системы (по отраслям)

1. Тема Разработка информационной системы для парикмахерской
2. Цель и задачи работы Цель данного проекта заключается в оптимизации работы парикмахерской и разработке сервиса регистрации сеанса стрижки пользователем. Задачи: литобзор, постановка задачи; изучение и моделирование предметной области; разработка информационной системы; анализ результатов
3. Характеристика предметной области Предметной областью является упрощение записи пользователей на сеанс стрижки за счет создания информационной системы
4. Содержание расчетно-пояснительной записки введение, постановка задачи, изучение и моделирование предметной области, проектирование системы, реализация системы, представление и анализ полученных результатов, заключение (выводы), список источников
5. Перечень графического материала иллюстрации предметной области: схема компонентов программного обеспечения информационной системы
6. Срок сдачи ВКР руководителю \_\_\_\_\_\_\_\_
7. Консультанты по ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. Дата выдачи задания \_\_\_\_\_\_\_
9. Руководитель проекта к.т.н., профессор, Ведищев Виталий Викторович
10. Представитель организации-заказчика «OCTAGON мужская парикмахерская» \_\_\_\_\_\_\_\_\_\_\_\_\_
11. Задание принял к исполнению студент Кузнецов Степан Владимирович

**Аннотация**

С. 0. Ил. 0. Табл. 0. Литература 0 назв.;

Настоящий отчет является пояснительной запиской по выпускной квалификационной работе техника «Разработка информационной системы для парикмахерской».

Цель данного проекта заключается в повышении эффективности работы парикмахерской за счет автоматизации ручных операций, таких как: уменьшение времени ожидания, увеличение числа обслуженных клиентов. В данном отчете приводится описание создания информационной системы и результаты ее работы в условиях реальной работы.

Спроектированная система позволяет осуществлять регистрацию пользователя на сеанс стрижки к определенному мастеру, в определенный день и время, также с возможностью выбора стрижки. Система предоставляет администраторам парикмахерской возможность добавления стрижек, мастеров, цен на услуги.

**Оглавление**

[1 Постановка задачи 6](#_Toc70600532)

[1.1 Обзор предметной области 6](#_Toc70600533)

[1.2 Анализ стандартных средств существующих способов решения задачи 6](#_Toc70600534)

[1.3 Постановка задачи. Цели работы, критерии оценки и ограничения 7](#_Toc70600535)

[1.3.1 Постановка задачи 7](#_Toc70600536)

[1.3.2 Критерии системы 8](#_Toc70600537)

[2 Изучение и моделирование предметной области 9](#_Toc70600538)

[2.1 Выявление основных понятий и процессов 9](#_Toc70600539)

[2.2 Теоретическое изучение предметной области 9](#_Toc70600540)

[2.3 Экспериментальное изучение предметной области 10](#_Toc70600541)

[2.4 Ролевая модель 11](#_Toc70600542)

[3 Проектирование системы 12](#_Toc70600543)

[3.1 Общая архитектура системы 12](#_Toc70600544)

[3.2 Концептуальная и физическая модели данных 13](#_Toc70600545)

[3.3 Макеты интерфейсов пользователя 15](#_Toc70600546)

[4 Реализация системы 21](#_Toc70600547)

[4.1 Аппаратное обеспечение 21](#_Toc70600548)

[4.2 Программное обеспечение 21](#_Toc70600549)

[4.3 Разработанные программные средства 21](#_Toc70600550)

[4.3.1 Структура проекта 22](#_Toc70600551)

[4.3.2 Fronted микросервис 22](#_Toc70600552)

**Введение**

Мода на прически не стоит на месте, каждый день привнося в себя новые стили. Не так давно привычное название “парикмахерская” сменилась на “барбер-шоп”. Малейший ребрендинг старого названия позволил повысить спрос на данные бытовые услуги в кругах не только молодежи, но и более старшего поколения. В связи с увеличением спроса, начали появляться проблемы с наличием ведения нечетких списков записи клиентов на услуги “барбера”. Ведение бумажных списков клиентов не актуальны, так как существует множество способов реализации хранения, передачи и получения данных на электронных ресурсах. Применение веб-сервисов облегчает работу любой сферы деятельности человека за счет своей кроссплатформенности.

Разрабатываемая система призвана упростить процесс записи клиентов барбер-шопа на сеанс стрижки волос и бороды. Кроме того, система предоставляет инструменты администрирования для редактирования информации в приложении. Администратор сможет добавлять в систему новых барберов, стрижки, цены и время сеансов. У барберов будет возможность просмотра списка текущих сеансов на день.

# **1 Постановка задачи**

## **Обзор предметной области**

В настоящее время бытовые услуги являются неотъемлемой частью жизни каждого человека. Каждый гонится за своим имиджем, ищет нужные стилевые подборки, ведь мода не стоит на месте. Не так давно в России обычные парикмахерские ушли на второй план. Их вытеснили так называемые «барбер-шопы». Это специализированные парикмахерские для мужчин, где стрижку головы и бороды осуществляет мастер-барбер, чаще всего мужчина. Из-за большого спроса на качественные работы барберов, в такие заведения часто бывают большие очереди, в связи с чем нуждаются в оптимизации рабочего времени мастеров. Процесс оптимизации проделывается путем создания кроссплатформенных веб-приложений с возможностью записи онлайн на сеанс стрижки. На сервере хранятся данные о том, свободен ли барбер на определенное время или нет.

## **Анализ стандартных средств существующих способов решения задачи**

Задачи оптимизации рабочего процесса барбер-шопов решались множество раз в различных мобильных и веб приложениях специализированных парикмахерских. Рассмотрим решение оптимизации на примере мобильного приложения барбер-шопа “Charlies”.

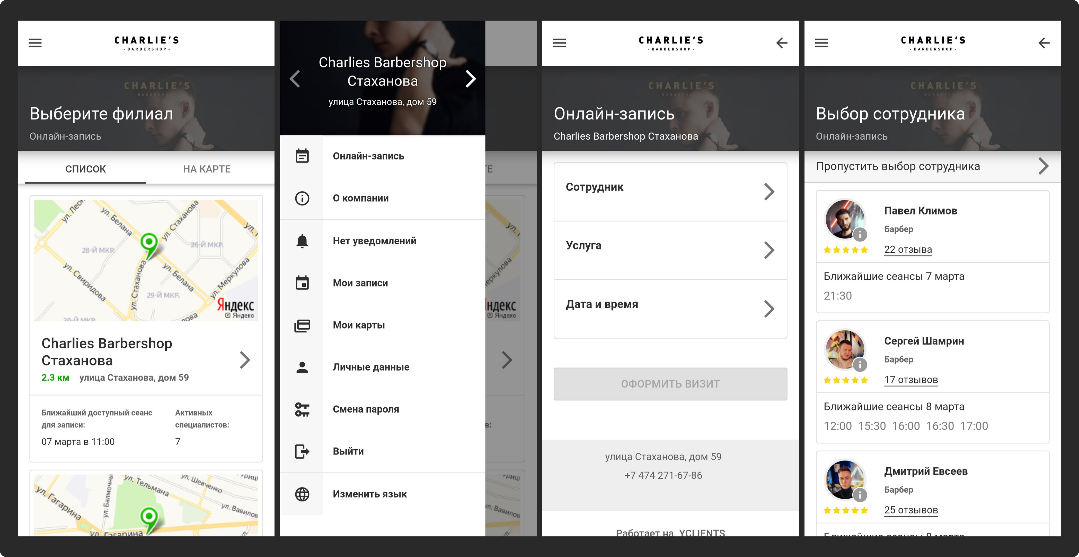


Рисунок 1 – Страницы мобильного приложения барбер-шопа “Charlies”

Как видно на рисунке 1, приложение имеет множество страниц, на которых есть возможность просмотра адресов барбер-шопов, также можно просматривать свои записи на сеансы, уведомления, личные данные. Онлайн запись является одной из главных функций приложения, которая и позволяет упростить работу барбер-шопа за счет того, что информационная система сама контролирует поток клиентов в приложении. Данный функционал логически представляет следующую последовательность: в базе данных содержится информация о барберах, о свободном времени для посещения, а также услуги. При онлайн-записи для пользователя предоставляются эти данные, из которых он выбирает самые удобные для себя. После подбора оптимальных требований, пользователь нажимает на кнопку регистрации и происходит отправка данной формы на сервер. Информация из формы обрабатывается и для выбранного мастера, выбранное время из базы данных удаляется. Логика хранения и обработки данных в разных приложениях может меняться.

## **Постановка задачи. Цели работы, критерии оценки и ограничения**

### **1.3.1 Постановка задачи**

Цель данного проекта заключается в повышении эффективности работы за счет автоматизации ручных операций в барбер-шопе “Octagon”.

Задачи, поставленные перед выполнением работы:  
1. Изучить научную литературу по теме работы.

2. Рассмотреть имеющиеся решения задачи.

3. Обеспечить повышение эффективности работы барбер-шопа.

4. Уменьшить время ожидания клиентов.

5. Увеличить число обслуженных клиентов.

Основными функциями системы являются:

1. Онлайн-запись на услугу барбера. Данная функция позволит решить задачу по уменьшению времени ожидания за счет отложенной записи клиентов. Благодаря заполнению формы регистрации на сеанс, система заносит в базу данных данные о клиенте и сеансе и автоматически распределяет поток клиентов. На выходе, функция отправляет пользователю уведомление о запланированном сеансе.
2. Оповещение пользователей о забронированной услуге. Функция позволяет напоминать пользователю о запланированном сеансе. За счет этого можно сократить случаи, связанные с человеческим фактором, когда клиент может забыть о сеансе, что негативно влияет на выручку заведения.
3. Система авторизации. Система авторизации позволяет упростить процесс бронирования сеанса благодаря автоматическому заполнению формы бронирования имеющимися данными о клиенте в базе данных.

### **1.3.2 Критерии системы**

Основными критериями оценки при построении системы являются:

* Возможность работы системы в момент, когда требуется заказать услугу.
* Валидность данных при регистрации и авторизации.
* Минимальное время обработки данных.
* Достоверность информации о свободных датах и времени на сеансы.

# **2 Изучение и моделирование предметной области**

**2.1 Выявление основных понятий и процессов**

В работе парикмахерских и барбер-шопов есть определенные услуги, которые различаются по длительности и сложности, что влияет на распределение клиентов по времени. Например, в одном сеансе клиент имеет право заказать стрижку головы и бороды, или только стрижку головы, или только стрижку бороды и т.д. В связи с этим стоит выявить степени сложности сеанса, как параметр распределения очереди.

Так как живая очередь создает помехи в работе заведений из сферы бытовых услуг, то в барбер-шопе «Octagon» будет отменено понятие живой очереди, которое заменится на понятие электронной очереди. Это означает, что даже если клиент самостоятельно пришел в барбер-шоп, администратор заносит информацию о сеансе по предпочтению клиента в систему, поэтому на данный момент его обслужить не получится в связи с тем, что электронная очередь уже распределила на текущее время поток клиентов.

## **2.2 Теоретическое изучение предметной области**

Входными данными для бронирования сеанса будет информация о клиенте и сеансе. Основными сущностями при проектировании модели данных в ИС являются: пользователь, услуга, сеанс. Пользователи будут разделены правами доступа на несколько уровней: клиент, барбер, администратор. Барбер предоставляет услуги клиенту во время сеанса. Пользователь (клиент) бронирует в приложении сеанс на определенную дату, время, барбера и услугу. Информация об этом помещается в базу данных, соответственно, забронированная дата и время у указанного барбера уже не будет доступна другому пользователю.

У каждого барбера есть определенные временные этапы рабочего дня. Это означает, что пользователь может осуществить запись только по указанному времени в системе. Например, в базе данных будет указываться, что у барбера в день может быть 6 сеансов: 10:00, 12:00, 14:00, 16:00, 18:00, 20:00. Когда пользователь бронирует дату и время, в таблице сеансов появляется соответствующая запись. Если на указанную дату, время и барбера, в независимости от выбранной услуги, уже есть запись в базе данных, то пользователю будет показано соответствующее уведомление и запись совершена не будет.

Если клиент не явился по назначенному времени, сеанс все равно считается начатым, однако прибыли этот сеанс не принесет. Решением данной проблемы может быть обзвон клиентов с вопросом о том, может ли он посетить сеанс раньше времени. В таком случае барбер обязан завершить сеанс до начала следующего.

## **2.3 Экспериментальное изучение предметной области**

Для получения максимально точных данных по длительности услуг в баребр-шопе “Octagon”, я изучил все возможные услуги и засек их длительность.

Таблица 1 – Услуги и их длительность

|  |  |  |  |
| --- | --- | --- | --- |
| **Мастер** | **Услуга** | **Длительность** | **Цена** |
| Старший мастер | Стрижка головы | 60 мин. | 600 р. |
| Старший мастер | Стрижка бороды | 60 мин. | 1000 р. |
| Старший мастер | Комплекс | 120 мин. | 1500 р. |
| Мастер | Стрижка головы | 90 мин. | 800 р. |
| Мастер | Стрижка под машинку | 45 мин. | 600 р. |
| Мастер | Стрижка бороды | 1 ч. | 600 р. |

Данная информация будет занесена в базу данных в соответствующие таблицы. В процессе проектирования базы системы были учтены следующие сущности: пользователь, услуга, сеанс.

## **2.4 Ролевая модель**

**Управление доступом на основе ролей** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Role Based Access Control, RBAC*) — развитие политики [избирательного управления доступом](https://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%B1%D0%B8%D1%80%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC), при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя [роли](https://ru.wikipedia.org/w/index.php?title=%D0%A0%D0%BE%D0%BB%D1%8C_(%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C)&action=edit&redlink=1). Формирование ролей призвано определить чёткие и понятные для [пользователей](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C) компьютерной системы правила разграничения [доступа](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%BE%D1%81%D1%82%D1%83%D0%BF&action=edit&redlink=1). Ролевое разграничение доступа позволяет реализовать гибкие, изменяющиеся динамически в процессе функционирования компьютерной системы правила разграничения доступа.

В приложении для барбершопа есть свое разделение ролей. Всего предполагается 3 роли: обычный пользователь, барбер, администратор. Такое разграничение ролей влечет за собой разделение функционала для пользователей каждой категории ролей.

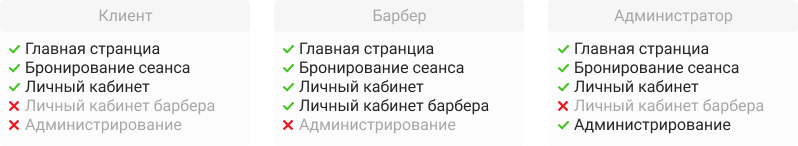


Рисунок 2 – Ролевая модель информационной системы

Как видно на рисунке 2, клиент будет иметь самый простой функционал, включающий в себя помимо системы авторизации, просмотр главной страницы, осуществление записи на сеанс и просмотр своих записей в личном кабинете. Барбер имеет дополнительную функцию в системе – просмотр своих клиентов как на все дни, так и на сегодняшний день. Администратор не будет иметь доступа к личному кабинету барбера, однако он сможет изменять данные о сеансе по требованию клиента по звонку. Также администратор может удалить запись.

# **3 Проектирование системы**

## **Общая архитектура системы**

При проектировании системы была рассмотрена следующая архитектура, которая показана на рисунке 3.

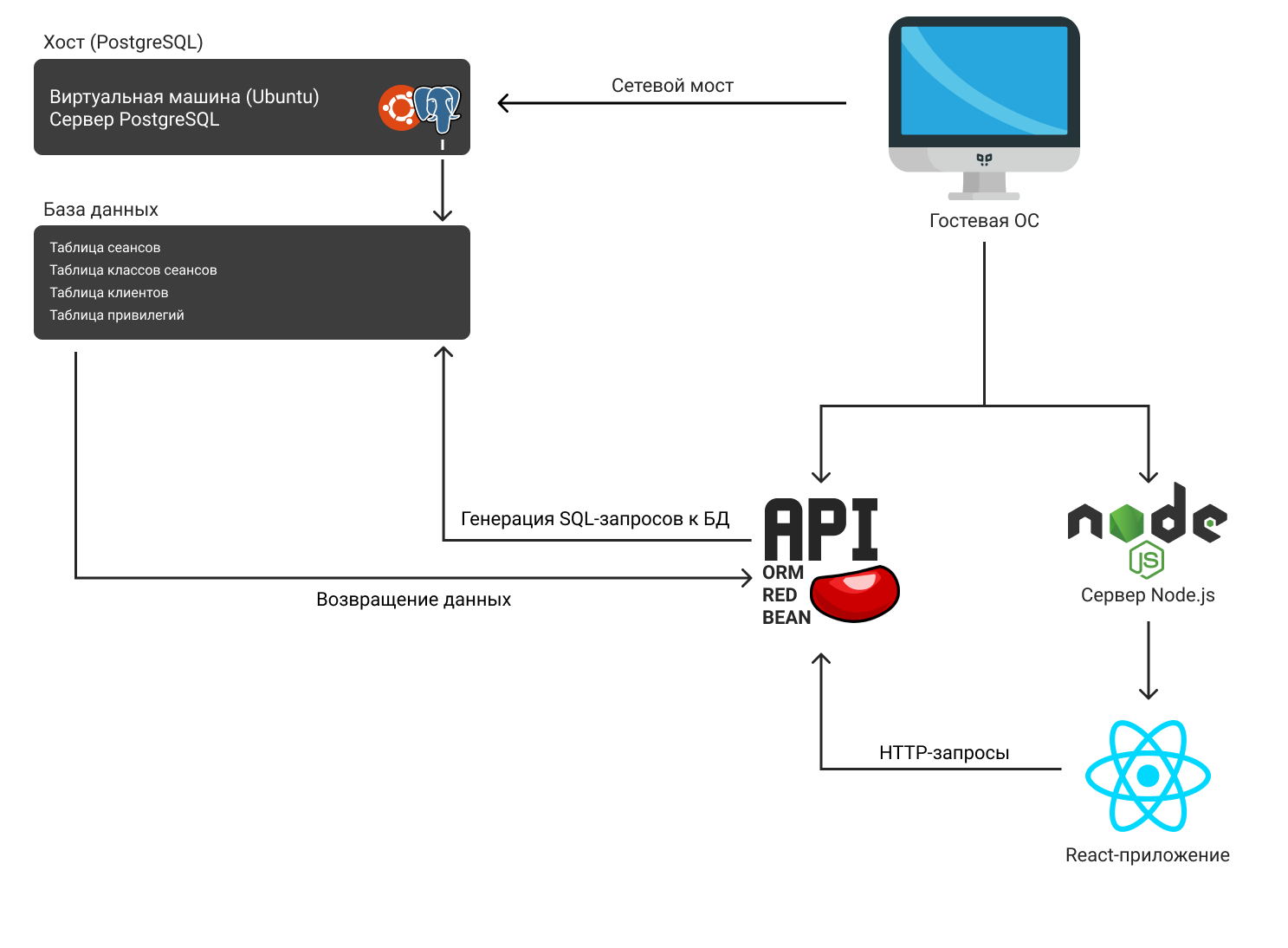


Рисунок 3 – Архитектура информационной системы

Как видно на рисунке 3, система состоит из множества составляющих, которые можно разбить на последовательный связанный перечень:

1. База данных;
2. Таблицы БД;
3. Гостевая ОС;
4. API;
5. Node;
6. React-приложения.

База данных находится на виртуальной машине вместе с сервером PostgreSQL. Связь гостевой ОС и хоста построена через сетевой мост в настройках программы Virtual Box, а также при помощи открытия портов через терминал ОС Ubuntu. На гостевой ОС работает локальный сервер, на котором работает API. API предназначено для отправки и возвращения данных в БД. Также на гостевой ОС располагается сервер Node.js, на котором работает React-приложение.

## **Концептуальная и физическая модели данных**

На рисунках 4, 5 изображены концептуальная и физическая модели данных, соответственно на таблицах 2, 3 приведены спецификации сущностей.

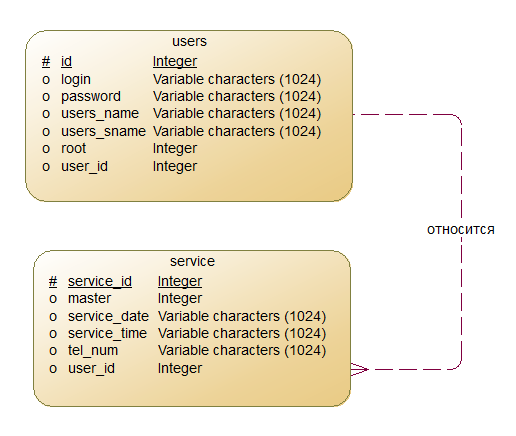


Рисунок 4 – Концептуальная схема данных

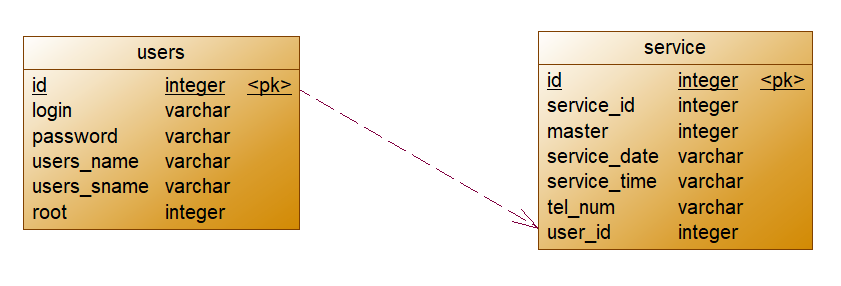


Рисунок 5 – Физическая схема данных

Таблица 2 – Спецификации сущности users

|  |  |  |
| --- | --- | --- |
| **Поле** | **Расшифровка** | **Тип данных** |
| id | id | integer |
| login | логин | varchar |
| password | пароль | varchar |
| users\_name | имя | varchar |
| users\_sname | фамилия | varchar |
| root | уровень доступа | integer |

Таблица 3 – Спецификации сущности services

|  |  |  |
| --- | --- | --- |
| **Поле** | **Расшифровка** | **Тип данных** |
| id | id | integer |
| Service\_id | Id услуги | integer |
| master | Id мастера | integer |
| Service\_date | Дата сеанса | varchar |
| Service\_time | Время сеанса | varchar |
| Tel\_num | Номер телефона клиента | varchar |
| User\_id | Id пользователя | integer |

* 1. **Макеты интерфейсов пользователя**

На рисунках 0 - 2 приведены макеты пользовательского интерфейса. Как показано на рисунке 2, для пользователей с различными уровнями прав доступны определенные страницы приложения.

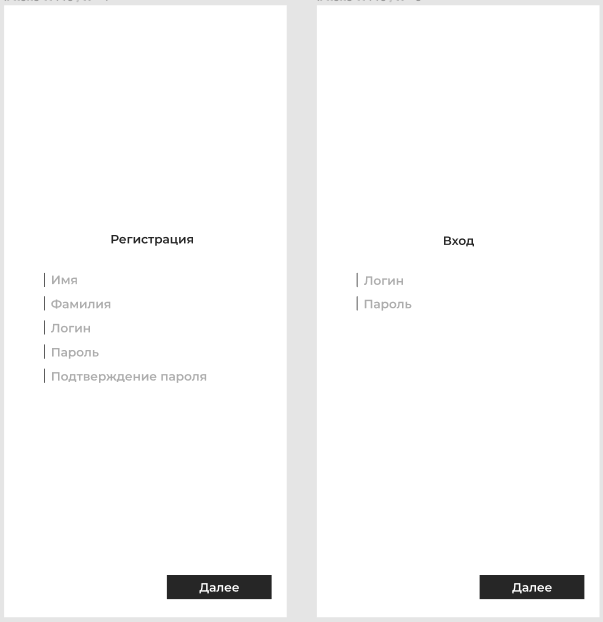


Рисунок 6 – Страницы регистрации и входа

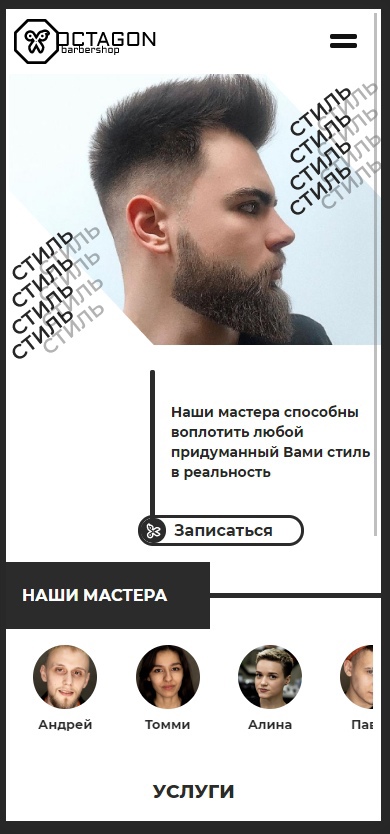


Рисунок 7 – Главная страница приложения

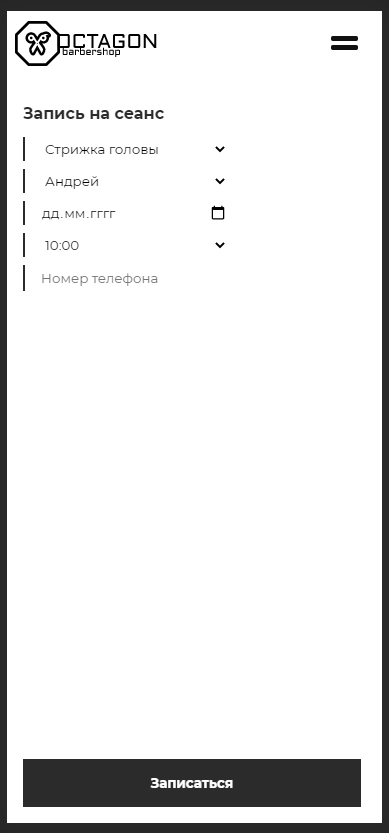


Рисунок 8 – Страница бронирования сеанса

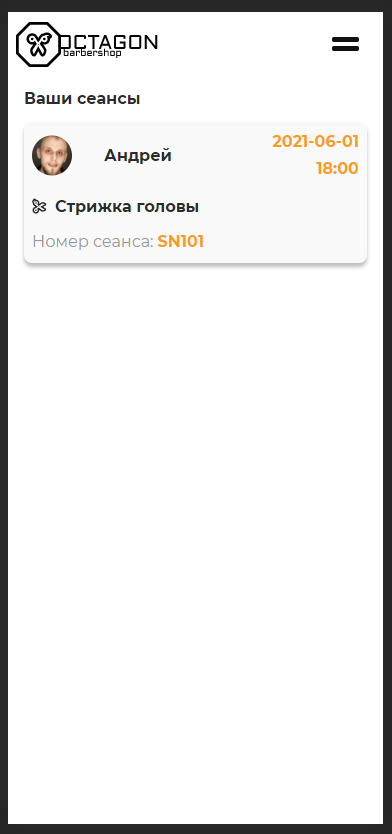


Рисунок 9 – Личный кабинет пользователя

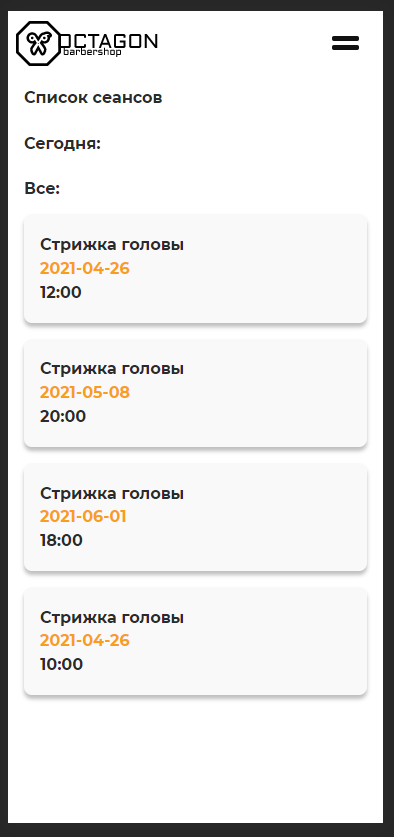


Рисунок 10 – Страница барбера

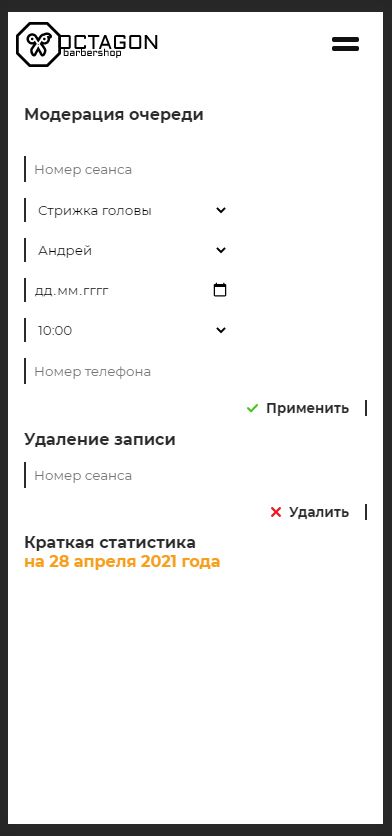


Рисунок 11 – Страница администратора

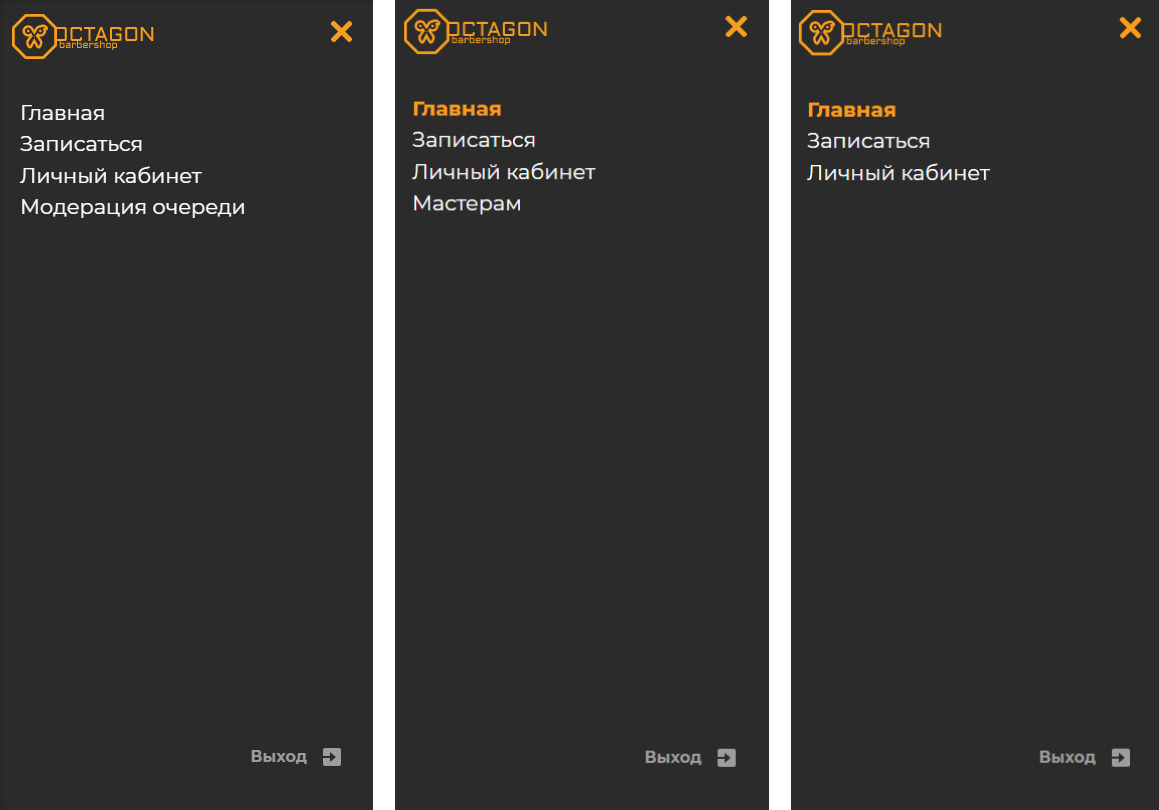


Рисунок 12 – Вариации навигационного меню

При заполнении и отправке формы, показанной на рисунке 8, пользователь получает одно из двух уведомлений, показанных на рисунке 13.

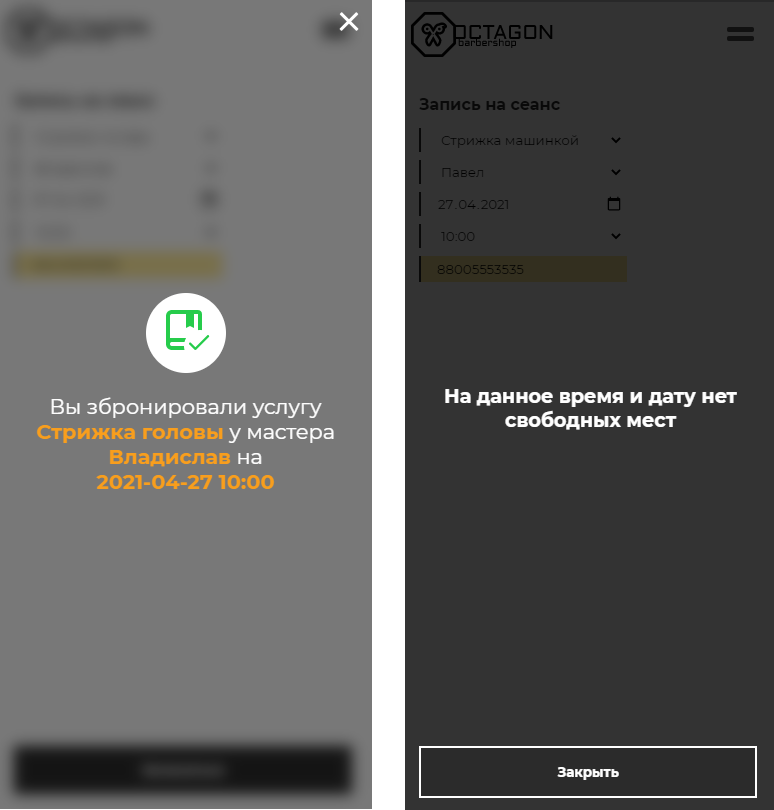


Рисунок 13 – Окна уведомлений при бронировании сеанса

Барбер может просматривать сеансы, которые он должен провести как на текущий день, так и на грядущие дни, но он не имеет доступа к странице администратора. Администратор может изменить информацию о сеансе, удались запись о сеансе и посмотреть сеансы на текущий день.

# **4 Реализация системы**

## **4.1 Аппаратное обеспечение**

Аппаратное обеспечение для разрабатываемой системы предполагает аппаратную платформу для развертывания всех компонентов системы. Опытным путем были установлены следующие требования к аппаратной платформе:

• процессор - 4 ядер;

• оперативная память - 8 Gb;

• дисковое пространство для операционной системы, виртуальной машины, Open Server, и компонентов Node.js - 256 Gb SSD.

Дисковое пространство необходимо как на ОС-хосте, так и на гостевой ОС.

## **4.2 Программное обеспечение**

Для обеспечения работы разрабатываемой системы необходима СУБД PostgreSQL версии 9 и выше. Для администрирования СУБД используется система Data Grip.

Для разработки системы предполагалась настройка СУБД. Сервер PostgreSQL был установлен на виртуальную машину с установленной ОС Ubuntu 20.04-desktop. Устанавливается связь ОС хост (Ubuntu) -> ОС гость (Windows). Для доступа БД на сервере на виртуальной машине, следует открыть порты доступа, выставив доверительные параметры в конфигурации PostgreSQL.

Для работы backend-микросервиса ИС устанавливается локальный сервер в качестве программы «OpenServer». Произведена необходимая настройка доменных имен для API.

## **4.3 Разработанные программные средства**

Разрабатываемся система имеет микросервисную архитектуру, состоящую из backend и frontend частей. Каждый микросервис представляет собой отдельный программный компонент, которые функционируют на рабочем устройстве программиста.

### **4.3.1 Структура проекта**

Согласно указанной архитектуре проекта, он разбит на два отдельных микросервиса, располагающихся на репозиториях GitHub. Итого проект состоит из двух отдельных репозиториев:

1. Octagon-react: frontend микросервис, содержащий интерфейсную часть приложения;
2. Octagon-orm: API приложения, возвращающая и отправляющая данные в БД.

Сборка проекта происходит путем использования консоли ОС (cmd) и настройки доменных имен в Open Server. Установка frontend микросервиса производится путем клонирования проекта с репозитория GitHub на устройство программиста и выполнения команды “npm install” в папке проекта. “Node packet manager” устанавливает необходимые связи и библиотеки, перечисленные в файле “package.json”. Для установки backend микросервиса, необходимо клонировать проект с репозитория GitHub и добавить в Open Server доменное имя “octagon-orm.loc”, выбрав путь к папке “public”.

### **4.3.2 Fronted микросервис**

Пользовательский интерфейс разработан на фреймворке React JS. Получение и отправка данных в проекте осуществляется путем AJAX и Fetch запросов на сервер. В качестве сервера выступает API, возвращающая данные из БД по определенно написанным методам. API создано на базе фреймворка RedBean для PHP, который является в качестве ORM. ORM (англ. Object-Relational Mapping, рус. объектно-реляционное отображение, или преобразование) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

API возвращает данные в виде строки, после чего, на стороне клиента данная строка преобразуется в JSON-объект. Распарсенные объекты циклически добавляются на страницы приложения. Формы регистрации, бронирования сеанса также работают через API, отправляя form-data на сервер.

Пользовательский интерфейс ввиду особенности разработки на React JS состоит из отдельных компонентов. Связь компонентов осуществляется через библиотеку “react-router”, “react-router-dom”. Компонент экспортируется внутри себя в другие компоненты, в которые он импортируется. После импорта компонента, его можно использовать как HTML-тэг.

### **4.3.3 Backend микросервис**

Backend архитектура проекта состоит из двух частей:

* База данных
* API

В БД созданы две таблицы (см. Пункт 3.2), выборка из которых производится через методы API. Всего создано 6 методов API:

1. rq($table) – Тип GET. Метод выборки данных по параметру, который определяет таблицу, из которой производится выборка;
2. registration – Тип POST. Метод регистрации пользователя;
3. update – Тип POST. Метод редактирования таблицы услуг. Редактирование осуществляет администратор на странице «Модерация»;
4. delete – Тип POST. Метод удаления сеанса из таблицы услуг.
5. getUserByRoot($root) – Тип GET. Метод выборки пользователей по уровню прав.
6. service – Тип GET. Метод выборки записей на сеансы из таблицы услуг.

# **5 Представление результатов**