

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»
Образовательная программа «Веб-технологии»

Отчет по курсовому проекту
по дисциплине «Инженерное проектирование»

Тема: «Разработка информационной системы для салона красоты»

Выполнил:

Студент группы 191-361

Козлов А.В.

подпись, дата

Принял:

Старший преподаватель

Данышина М.В.

подпись, дата

Москва 2020

ВВЕДЕНИЕ

В рамках данного курсового проекта была разработана информационная система для салона красоты. В состав разработанной системы входят frontend, backend и база данных. Система размещена на хостинге факультета информационных технологий. [8]

Целью данного курсового проекта является расширение информационного поля небольшого салона красоты, а именно формирование информационной системы для более удобного способа информирования клиентов об оказываемых услугах и способа ведения учёта заказов.

Задачи данного курсового проекта:

- провести анализ отечественных и зарубежных аналогов;
- составить формальные модели для формирования представления о системе;
- составить модель данных для создания на её основе базы данных;
- создать веб-приложение и серверное приложение, ведя разработку в репозитории и использованием веток;
- для каждого приложения разъяснить структуру, описать особенности функционирования и провести тестирование.

Объектом исследования данного курсового проекта является небольшой салон красоты, а предметом исследования является формируемая информационная система для салона красоты.

В данной курсовой работе содержатся 7 таблиц и 9 рисунков, состоит она из 20-ти страниц.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ аналогов

1.1.1 «White Fox»

«White Fox» – это полный спектр парикмахерских услуг, косметология лица и тела, маникюр, солярий, СПА на Рублевском и Новорижском шоссе. [3]

Оценка критериев данного сайта приведена в таблице 1.1.

Таблица 1.1 – Оценка критериев «White Fox»

Критерий	Оценка
Ключевые слова	Более 20-ти. Содержат указание категорий услуг и географическое расположение филиалов салона красоты.
Доступность	Нет специальных атрибутов для экранного диктора. Некоторые ссылки не обладают сопровождающей информацией. Интерфейс достаточно контрастен.
Структура	Структура преимущественно построена на тегах «div», «a», «p» без использования «section», «article», «button» и т.п. там, где это семантически обосновано. Все стили применяются по id элемента. Классы не используются.
Адаптивность	Сайт адаптивен под мобильные устройства и планшеты, но не под широкоформатные мониторы.
Юзабилити	Сайт интуитивно понятен, но нет возможности вывести все услуги на одной странице и применить к ним фильтр по каким-либо параметрам, что замедляет процедуру поиска в каталоге.
Функциональность	На сайте можно узнать об оказываемых услугах, мастерах, контактной информации, вакансиях, новостях, но нет возможности подать заявку на приём.
Быстродействие	Не используются современные форматы изображений. Множество файлов со стилями блокируют первоначальную отрисовку страницы. Текст не отображается до загрузки файлов со шрифтами. У статических объектов короткое время жизни в кеше. В среднем один день.

1.1.2 «Город красоты»

Сеть салонов «Город Красоты» в Москве предлагает парикмахерские услуги любой сложности, а также процедуры косметологии для лица и тела, маникюр, педикюр, различные виды массажей. [4]

Оценка критериев данного сайта приведена в таблице 1.2.

Таблица 1.2 – Оценка критериев «Город красоты»

Критерий	Оценка
Ключевые слова	«Город красоты, салон красоты, салон красоты Москва».
Доступность	Нет специальных атрибутов для экранного диктора. Некоторые ссылки не обладают сопровождающей информацией. Текст на фоне картинок иногда сливается.
Структура	Структура преимущественно построена на тегах «div», «a», «p» без использования «section», «article», «button» и т.п. там, где это семантически обосновано. В некоторых местах используется методология «БЭМ».
Адаптивность	Сайт адаптивен под мобильные устройства и планшеты, но не под широкоформатные мониторы.
Юзабилити	Главная страница перегружена элементами. Присутствует всплывающее модальное окна при открытии сайта. Удобная навигация.
Функциональность	На сайте можно узнать об оказываемых услугах, мастерах, контактной информации, вакансиях, новостях, а также произвести онлайн-запись на приём.
Быстродействие	Не используются современные форматы изображений. «Тяжёлая» кодировка изображений. Множество файлов со стилями блокируют первоначальную отрисовку страницы. У статических объектов короткое время жизни в кеше. В среднем один день. Очень высокий объём сайта: более 7-ми МБ. Огромное количество элементов в DOM.

1.1.3 «Beautify»

Салон «Beautify» в Lakelands, Stillorgan предлагает широкий спектр косметических процедур от профессионально обученных экспертов и терапевтов, чтобы вы чувствовали себя изнеженными, красивыми и свежими. [6]

Таблица 1.3 – Оценка критериев «Beautify»

Критерий	Оценка
Ключевые слова	«beautician Stillorgan , beauty salon Stillorgan»
Доступность	Нет специальных атрибутов для экранного диктора. Некоторые ссылки не обладают сопровождающей информацией. Интерфейс достаточно контрастен.
Структура	Помимо тегов «div», «a», «p» используются «main», «aside» и т.п., что помогает проще воспринимать структуру сайта. Стили заданы через классы. Методология «БЭМ» не применяется.
Адаптивность	Сайт не обладает адаптивностью под мобильные устройства, планшеты или широкоформатные мониторы.
Юзабилити	Сайт интуитивно понятен, но нет возможности вывести все услуги на одной странице и применить к ним фильтр по каким-либо параметрам, однако очень удобно списком можно вывести все услуги в рамках одной категории и сравнить их.
Функциональность	На сайте можно узнать об оказываемых услугах, контактной информации, расписании, а также подать заявку на приём. Помимо этого, есть форма связи.
Быстродействие	Не используются современные форматы изображений. «Тяжёлая» кодировка изображений. У статических объектов короткое время жизни в кеше. В среднем 15 минут. Множество фотографий без явно заданных атрибутов «width» и «height». Объём сайта: менее 1-го МБ.

1.2 Формальные модели

В качестве формальной модели была построена функциональная модель для бизнес-процесса «Записаться на приём». Для создания модели использовалась методология IDEF0. В составе IDEF0-модели была построена графическая диаграмма (Рисунок 1.1) с декомпозицией до первого уровня (Рисунок 1.2). [2]

Для построения диаграмм использовалось программное обеспечение ERwin Process Modeler. [16]

Коротко говоря, данная модель отображает структуру и функции системы, так же потоки информации и материальные объекты, преобразуемые функциями.

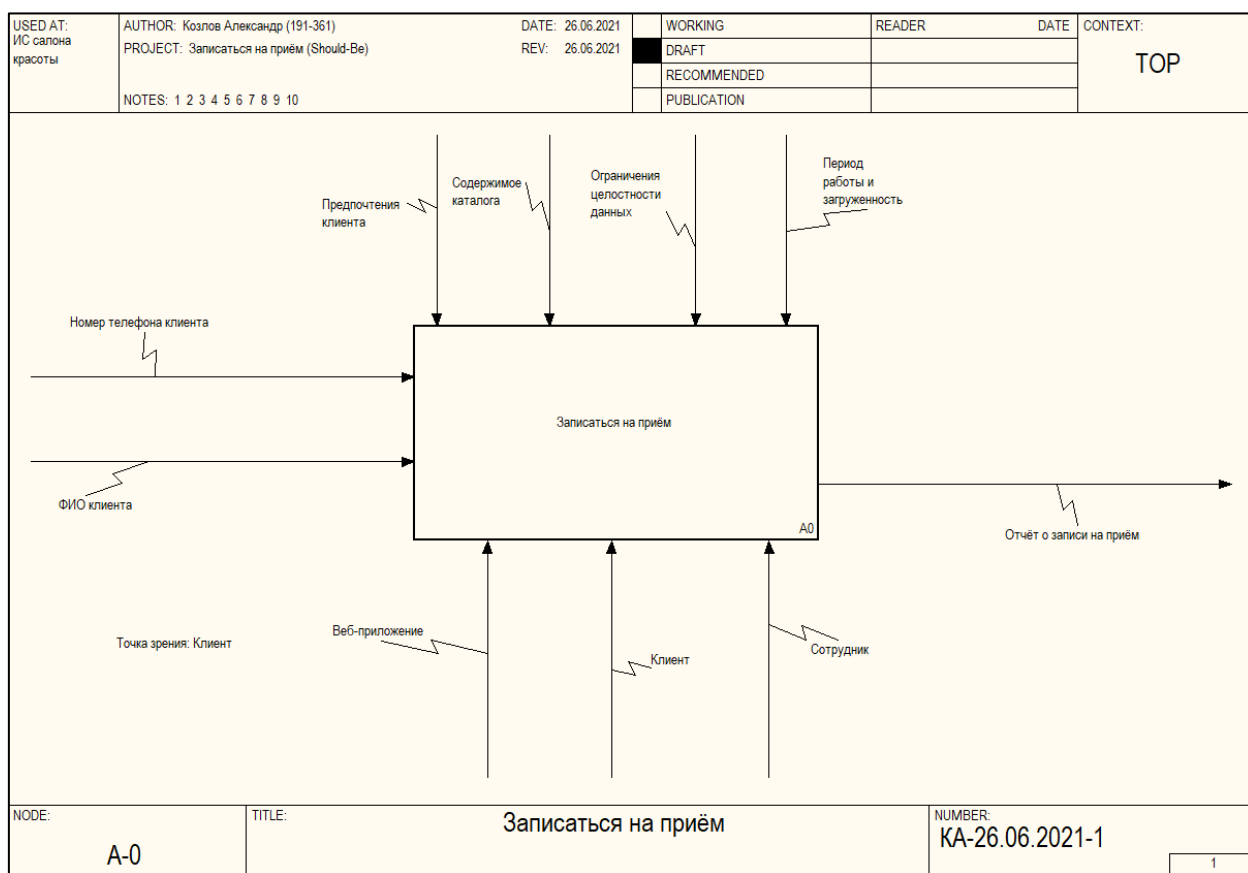


Рисунок 1.1 – Обязательная контекстная диаграмма верхнего уровня

Данная диаграмма определяет один из процессов, который должен быть внедрён в систему в конечном итоге. Сам же процесс описан с точки зрения клиента, поэтому некоторые технические подробности, например, участие backend'а, опущены.

В итоге клиент должен получить отчёт от успехе или провале подтверждения той заявки, которую он сформирует. Сам же отчёт выражен в устном подтверждении сотрудника салона красоты по телефону.

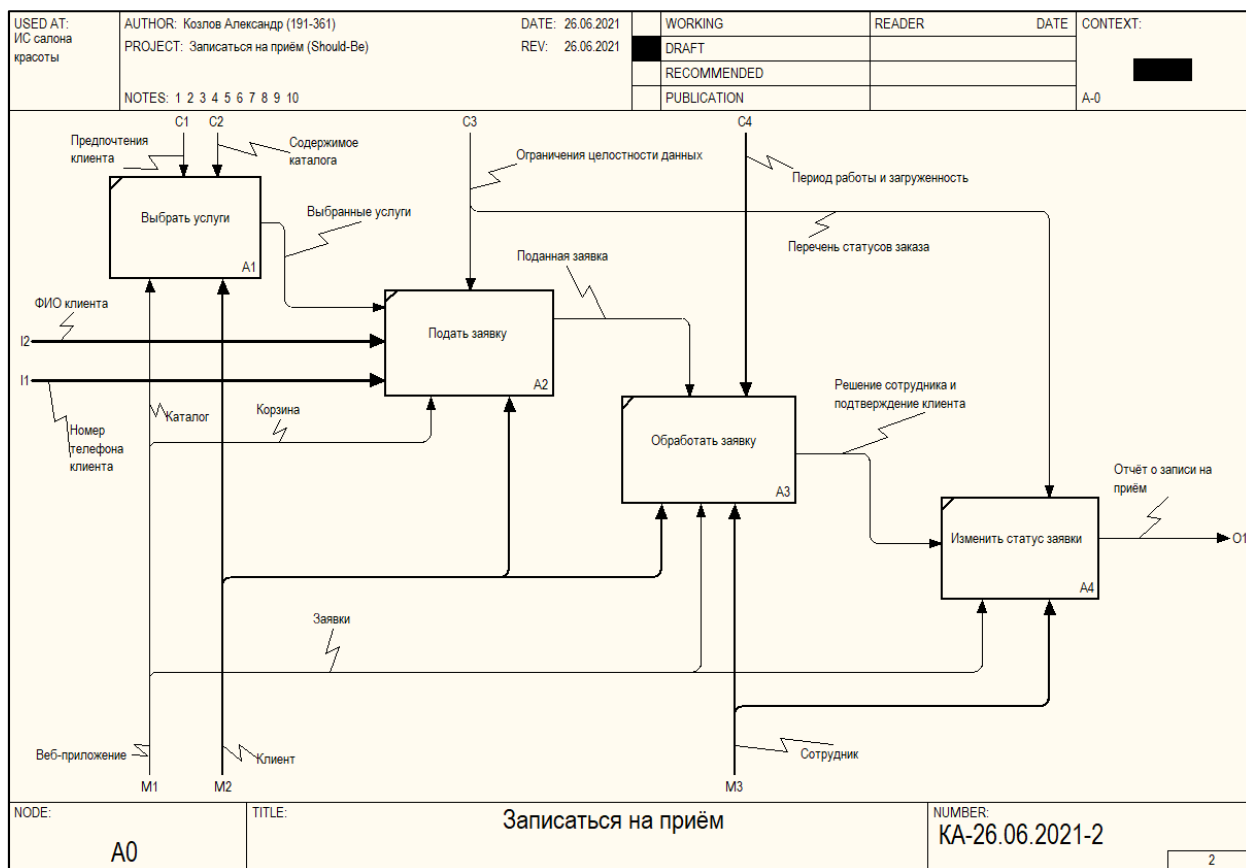


Рисунок 1.2 – Дочерняя диаграмма узла А-0

Как видно из диаграмм, все заявки должны обрабатываться сотрудниками салона красоты. Данный фактор в совокупности с накладываемыми ограничениями целостности данных позволит уменьшить число нежелательных заявок.

Прецеденты, возможные в рамках системы, определены на диаграмме прецедентов на рисунке 1.3.[15]

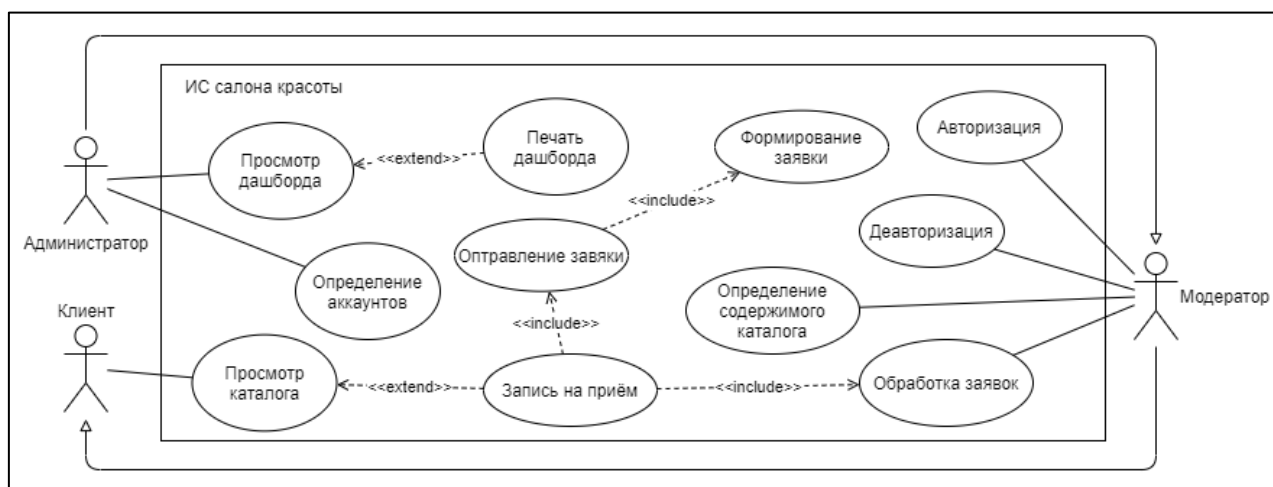


Рисунок 1.3 – Диаграмма прецедентов ИС салона красоты

На рисунке 1.4 представлена «карта пути клиента» обычного клиента салона красоты, который предпочитает производить поиск в Интернете.

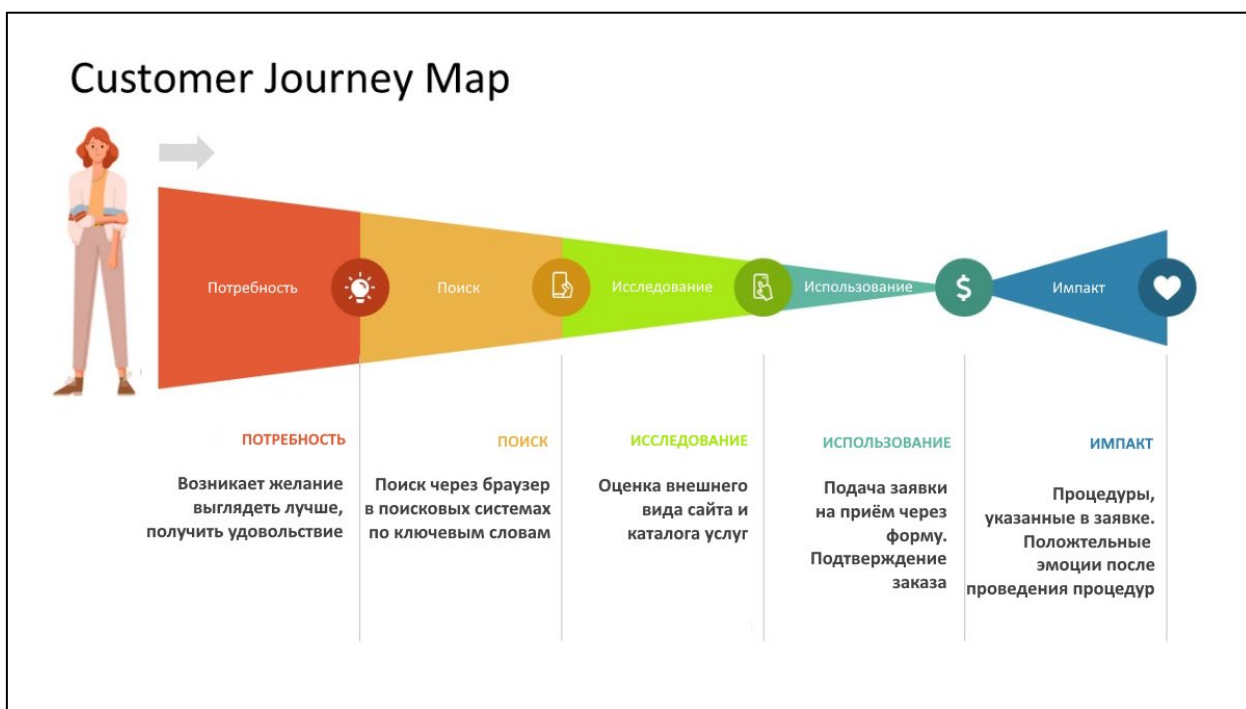


Рисунок 1.4 – Customer journey map

1.3 Выводы по аналитической части

На основе данной теоретической части можно изложить следующие выводы:

- был проведён анализ аналогов разрабатываемого веб-приложения и во внимание будут взяты результаты этого анализа: наличие ключевых слов, адаптивной вёрстки, юзабилити и прочих приведённых в таблицах 1.1-1.3 критериев;
- была представлен диаграмма прецедентов, которой необходимо следовать при реализации системы;
- была представлена IDEF0-диаграмма с декомпозицией до первого уровня для подробного описания процесса «Записаться на приём» в рамках ИС;
- была представлена CJM потенциального клиента разрабатываемой ИС.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Модель данных

2.1.1 Концептуальная модель

При концептуальном проектировании базы данных была построена ER-модель в виде ER-диаграммы, представленной на рисунке 2.1. [7]

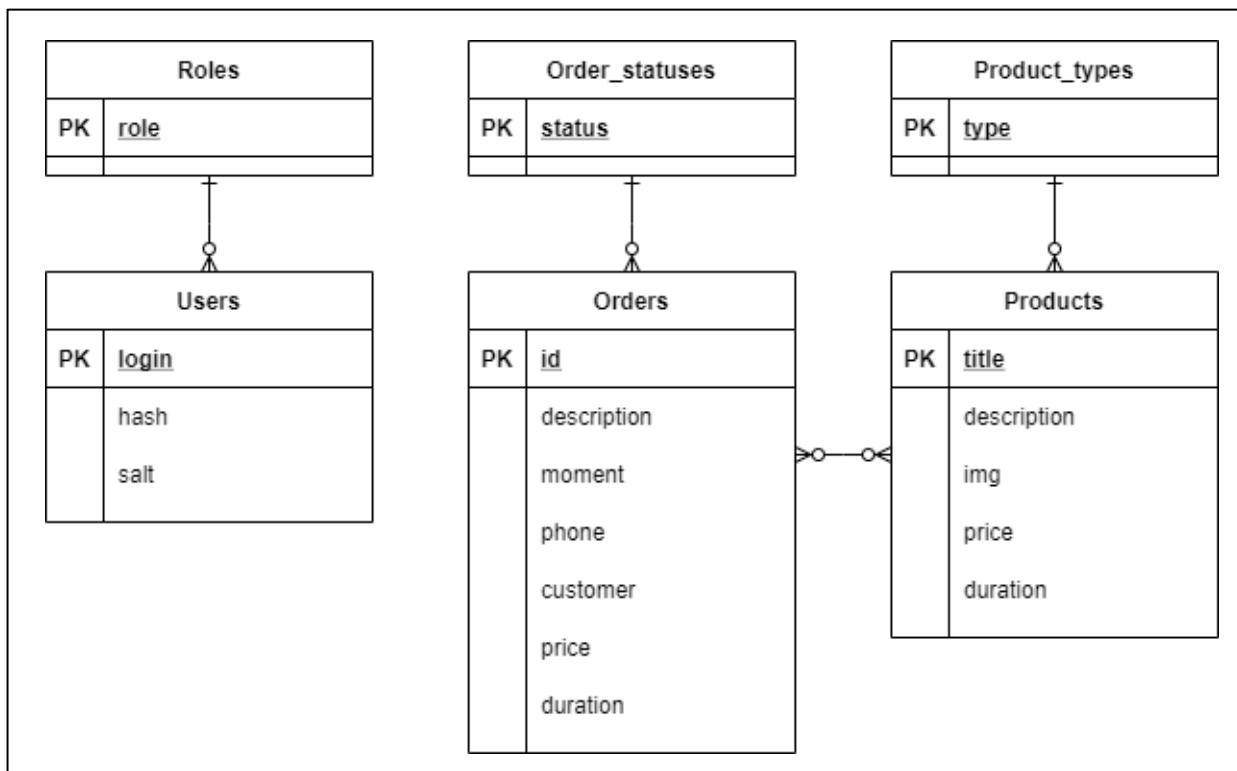


Рисунок 2.1 – Концептуальная модель данных

2.1.2 Аспект манипуляции

Хранить пароли пользователей в чистом виде не безопасно, поэтому, как видно из рисунка, вместо пароля должны храниться хэш и «соль». «Соль» единожды генерируется при создании пользователя и нужна для того, чтобы хэш у пользователей с одинаковыми паролями отличался. Например, у двух пользователей пароль «123456». В таком случае в поле «hash» в таблице «Users» будет храниться не хэш от «123456», а хэш от «123456» с приписанной на конце «солью». [13]

2.1.3 Аспект структуры

В базе данных должны быть три служебные таблицы с predetermined значениями. Одна для хранения видов ролей пользователей, другая для хранения видов статусов заказа и последняя для хранения типов услуг салона красоты.

Между сущностями «Orders» и «Products» объявлена связь «Многие-ко-многим», поэтому для реализации этой связи в базе данных необходимо создать промежуточную таблицу.

2.1.4 Аспект целостности

Для поддержания целостности данных в базе данных необходимо использовать внешние ключи между зависящими сущностями. Для простоты добавления этих ограничений у каждой сущности должно быть поле «id», по которому будет построен внешний ключ.

Логины у пользователей должны быть уникальны, поэтому необходимо добавить ограничение уникальности значений в поле «login» в таблице «Users». Аналогичное можно сказать и про названия услуг.

2.1.5 Физическая модель

На основе модели, приведённой в пункте 2.1.1, была построена физическая модель данных, описанная в виде ER-диаграммы на рисунке 2.2.

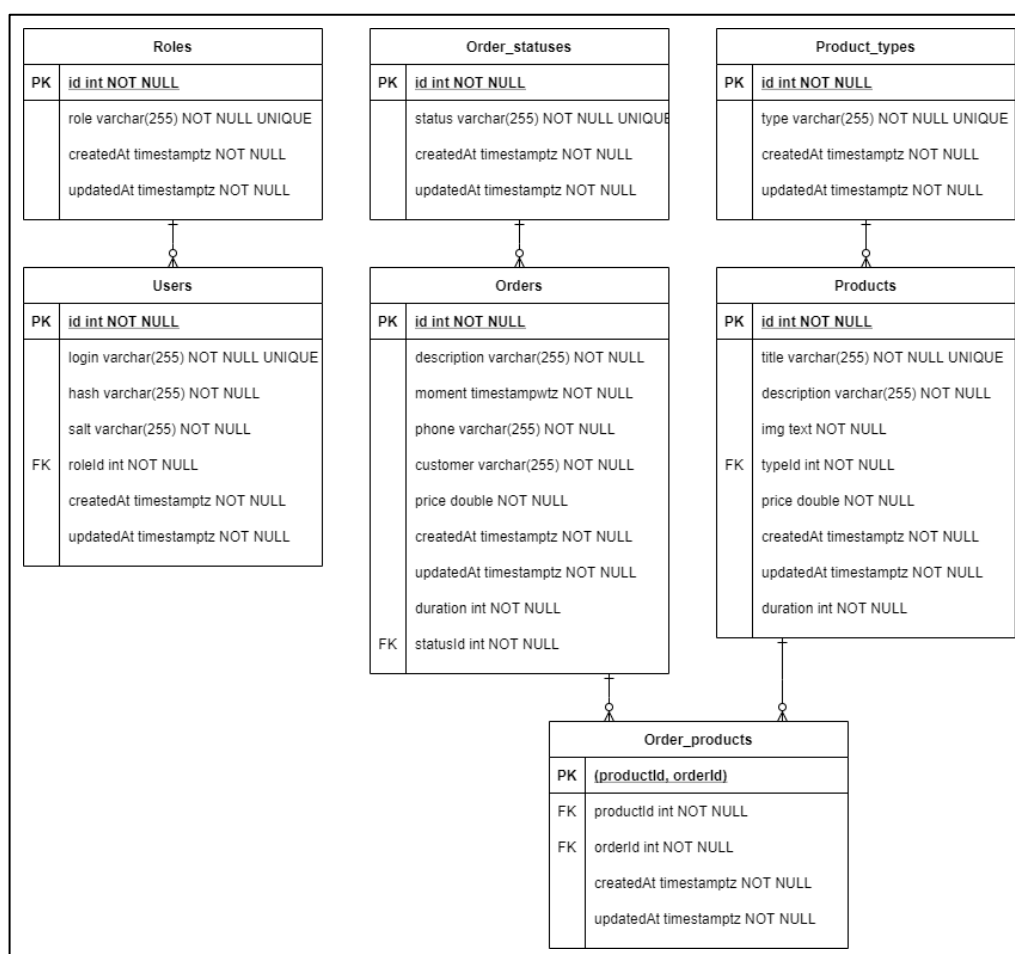


Рисунок 2.2 – Физическая модель данных

2.2 Frontend-приложение

Веб-приложение было создано на основе фреймворка «Vue» на языке программирования «JavaScript». [10]

Перечень используемых зависимостей в приложении:

- «axios»: используется для реализации API на frontend;
- «vue-cryptojs»: используется для хэширования паролей с «солью» перед отправкой на backend;
- «vue-router»: используется для поддержки общих у нескольких страниц html-элементов и удобной навигации;
- «vuex»: используется для реализации глобального хранилища в приложении;
- «node-sass», «sass-loader», «webpack»: используются для поддержки написания стилей на scss.

2.2.1 Структура проекта

Реализация API на стороне frontend находится в папке «api» и основана на «axios»: объект «axios» и его конфигурация представлены в файле «instance.js». Импорт модулей API и объекта «axios» представлены в файле «index.js». Прочие четыре файла являются модулями, между которыми разделена реализация методов API. [5]

В папке «assets» хранятся статические файлы, такие как картинки, шрифты и стили.

Компоненты Vue-приложения распределены между папками «pages» и «components». В свою очередь, в папке «pages» хранятся компоненты, описывающие собой целую страницу, если название компонента заканчивается на «Page», или обёртку для «vue-router», если название компонента заканчивается на «Layout». В папке «components» описаны отдельные компоненты, которые могут встречаться на страницах. Также, в той же папке содержится подпапка «modal», в которую вынесены все компоненты модальных окон.

В папке «router» расположен объект «VueRouter», реализующий навигацию в приложении. Навигация в нём реализована иерархически, что позволяет выводить на экране не только компонент, в котором находится пользователь, но и элементы «родителей» этого компонента.

В папке «store» реализовано глобальное хранилище приложения на основе «vuex». Во «vuex» существует четыре основных понятия: «state» - данные, «getter» - метод получения данных, «mutation» - метод размещения данных и «action» - асинхронный метод получения данных. В проекте активно используются возможности «vuex», а некоторые элементы хранилища вынесены в отдельные модули в отдельных файлах в подпапке «modules».

2.2.2 Стили

Все стили описаны в папке «assets/css» и написаны на «scss» с использованием вложенностей, миксинов и переменных. Используется методология БЭМ. [1]

В файле «variables.scss» описаны все переменные цветовых значений – ни одно цветовое значение не задано не через эти переменные, что позволяет очень просто интегрировать в приложение возможность переключения на другую тему, не меняя при этом другие свойства стилей.

Основные стили прописаны в файле «myStyle.scss». В начале файла импортируются переменные цветовых значений, затем сбрасываются некоторые стили по умолчанию, после этого задаются миксины: «form» применяется к каждой форме, «user-section» применяется к основным «обёрткам» с навигацией и «layout» для «обёрток», у которых снизу должен быть футер.

Вёрстка адаптивна как для широких экранов, так и для мобильных устройств.

На рисунке 2.3 представлен пример выбранной цветовой гаммы.

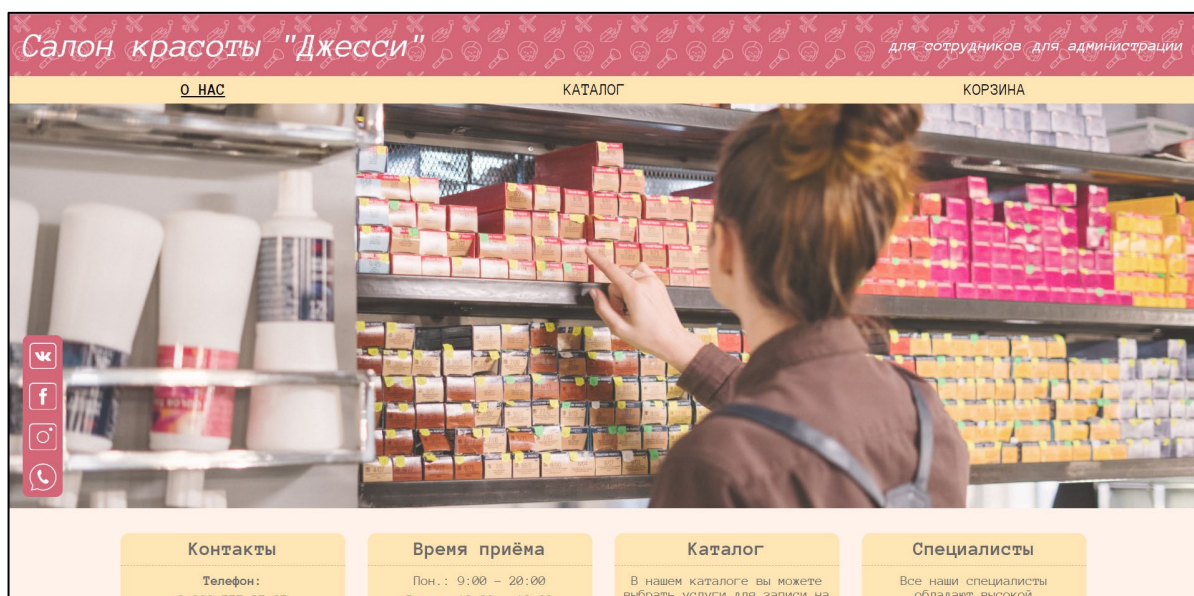


Рисунок 2.3 – Главная страница

К современным решениям дизайна можно отнести:

- Размеры шрифта и некоторых элементов в метрике «em».
- Блок фильтрации в каталоге, который прилипает к верхней части экрана.
- Сводная таблица в корзине, построенная на теге «ul».
- Футер, прилипающий к нижней части экрана и не съезжающий вверх из-за пустого пространства после основного контента.
- Использование небольшого пула сочетающихся цветов для оформления элементов.
- Оформление элементов в виде карточек для упрощения зрительного восприятия.
- Отсутствие лишних границ между элементами – резкие переходы между ними.

2.2.3 Особенности функционирования

Фильтр на странице каталога реализован динамически – фильтр по категориям показывает лишь те категории, которые содержатся в пришедших с backend данных, значения диапазона минимальной и максимальной цены рассчитывается на основании содержимого каталога, то же самое можно сказать и про значения диапазона продолжительности услуг.

В случае, если пользователь перешёл в корзину, но она пуста, отрисовывается компонент перенаправления в каталог, в панели навигации рядом со вкладкой корзины выводится количество добавленных товаров.

У администратора на дашборде выводится информация, агрегированная на сервере. Дашборд адаптирован для печати. [14]

Из-за использования модуля для удобной навигации «vue-route» путь, отображаемый в url-строке браузера, заменён не настоящим, поэтому после перезагрузки страницы веб-сервер может выдать ошибку «503», так как браузер попытается загрузить контент по этому пути. Для предотвращения подобной ситуации необходимо подредактировать конфигурационный файл веб-сервера. Но редактирование данного файла на хостинге факультета информационных технологий требует прав администратора. Прошу учесть этот независимый от меня фактор.

2.2.4 Тестирование

Для тестирования frontend`а был привлечён метод ручного тестирования на основе прохождения тест-кейсов. Сценарии тест-кейсов начинаются с главной страницы веб-приложения. Примеры тест-кейсов представлены в таблицах 2.1-2.4.

Таблица 2.1 – тест-кейс «подача заявки на приём»

Описание действий	Ожидаемый результат
<ol style="list-style-type: none"> 1. Нажать ЛКМ по надписи «Каталог» в навигационной панели в верхней части сайта; 2. путём нажатия ЛКМ по кнопкам «В корзину» на карточках в каталоге товаров выбрать как минимум одну услугу; 3. нажать ЛКМ по надписи «Корзина» в навигационной панели в верхней части сайта; 4. заполнить форму отправки заявки на приём, введя лишь ФИО, номер телефона и предполагаемое время приёма; 5. нажать ЛКМ по кнопке «Отправить заказ». 	<p>Клиент должен быть перенаправлен на страницу, где указан его номер заказа и сопутствующая информация.</p>

Таблица 2.2 – тест-кейс «подача заявки на приём: проверка валидности значений»

Описание действий	Ожидаемый результат
<ol style="list-style-type: none"> 1. Повторить первые три действия из тест-кейса в таблице 2.1; 2. заполнить форму отправки заявки на приём, введя лишь ФИО; 3. нажать ЛКМ по кнопке «Отправить заказ»; 4. заполнить форму отправки заявки на приём, введя лишь номер телефона; 5. нажать ЛКМ по кнопке «Отправить заказ»; 6. заполнить форму отправки заявки на приём, введя лишь предполагаемое время приёма; 7. нажать ЛКМ по кнопке «Отправить заказ»; 8. заполнить форму отправки заявки на приём, введя лишь ФИО, некорректный номер телефона и предполагаемое время приёма; 9. нажать ЛКМ по кнопке «Отправить заказ». 	<p>После каждого нажатия по кнопке «Отправить заказ» на форме должно появляться соответствующее предупреждение об ошибке.</p>

Таблица 2.3 – тест-кейс «Авторизация администратора»

Описание действий	Ожидаемый результат
<ol style="list-style-type: none"> 1. Нажать ЛКМ по надписи «для администрации» в верхней правой части сайта; 2. в открывшейся форме авторизации ввести случайные значения логина и пароля; 3. нажать ЛКМ по кнопке «Авторизоваться»; 4. ввести случайные пять символов в поле для ввода логина; 5. нажать ЛКМ по кнопке «Авторизоваться»; 6. в форме авторизации ввести корректные значения логина и пароля; 7. нажать ЛКМ по кнопке «Авторизоваться». 	<p>Нажатия ЛКМ по кнопке «Авторизоваться» после ввода случайных значений должны сопровождаться соответствующими сообщениями об ошибке.</p> <p>Нажатие ЛКМ по кнопке «Авторизоваться» после ввода корректных значений должно перенаправить пользователя на страницу для администрации.</p>

Таблица 2.4 – тест-кейс «Добавление нового пользователя»

Описание действий	Ожидаемый результат
<ol style="list-style-type: none"> 1. Повторить действия 1, 2 и 7 из тест-кейса в таблице 2.3; 2. нажать ЛКМ по надписи «Настройка пользователей» на навигационной панели сверху. 3. в форме добавления пользователей ввести логин уже существующего пользователя (существующие пользователи отображены в виде списка выше формы добавления пользователей), произвольный пароль, отвечающий критериям, и выбрать любую предложенную роль из выпадающего списка; 4. нажать ЛКМ по кнопке «Добавить учётную запись»; 5. изменить логин на произвольный, отвечающий критериям; 6. нажать ЛКМ по кнопке «Добавить учётную запись». 	<p>Попытка добавления пользователя с уже существующим логином должна сопровождаться соответствующей ошибкой.</p> <p>Добавление пользователя с корректными логином и паролем и выбранной ролью должно сопровождаться очищением формы и возникновением новой карточки в списке существующих учётных записей выше формы для добавления пользователей.</p>

2.3 Backend-приложение

Серверное-приложение было создано на основе фреймворка «Express» на языке программирования «JavaScript». [11]

Перечень используемых зависимостей в приложении:

- «dotenv»: используется для вынесения параметров конфигурации для подключения к базе данных;
- «jsonwebtoken»: используется для реализации получения токенов доступа и проверки токенов доступа.
- «sequelize»: используется в качестве интерфейса для обращения к базе данных, что позволило мигрировать с PostgreSQL на MySQL в середине реализации проекта с минимальными затратами по времени.

2.3.1 Структура проекта

Папка «bin» была создана по умолчанию при инициализации проекта и содержит единственный файл «www», где определены порт, на котором работает приложение, и обработчики событий.

В папке «controllers» реализованы методы добавления, удаления, редактирования и получения данных из базы данных. Методы распределены между файлами с говорящими названиями. Например, вся логика относительно заказов определена в файле «Orders.js».

В папке «db» определены модели данных и объект подключения к базе данных. Для каждой сущности определение реализовано в отдельном файле. Синтаксис реализации прост и интуитивно понятен благодаря модулю «sequelize».

В папке «middleware» определён middleware, проверяющий токен в заголовке запросов на наличие уровня доступа модератора или администратора. [12]

В папке «routes» определены маршрутизаторы, для перенаправления запросов на нужный метод из контроллеров. Как и в случае с контроллерами, каждый маршрутизатор покрывает запросы к определённой сущности из базы данных. Например, маршрутизатор «auth.js» перенаправляет на методы из контроллера, связанного с авторизацией и проверкой токена. Условия перенаправления на эти маршрутизации определены в файле «app.js» в корне проекта.

2.3.2 Особенности функционирования

В токенах содержится роль пользователя, что позволяет ограничивать доступ к методам из контроллеров, и `id` пользователя, что позволяет производить проверку, не позволяющую администратору удалить собственный аккаунт.

Связи между сущностями базы данных определены в файле «`db/index.js`». Конфигурация для подключения к базе данных содержится в файле «`.env`».

На рисунке 2.4 представлен типовой запрос к базе данных. Благодаря «`sequelize`» синтаксис запросов не зависит от базы данных, что позволило в середине разработки курсового проекта перейти на базу данных `mySQL` без модифицирования запросов.

```
const getAllOrders = async function (req, res) {  
  order.findAll( { options: { attributes: ['id', 'phone', 'description', 'customer', 'duration', 'price', 'moment'],  
    include: [{ model: orderStatus, attributes: ['id', 'status'], raw: true }, { model: product, through: { attributes: [] } } ] } } )  
    .then(orders => {
```

Рисунок 2.4 – Типовой запрос к базе данных

Поскольку внешние ключи в базе данных настроены, в запросах можно даже не указывать через какое поле совершается JOIN в запросе. В данном примере запрос возвращает список заказов с перечисленными атрибутами, статусом, взятом из другой таблицы, и списком услуг в заказе благодаря ключевому слову «`through`» без указания атрибутов.

2.3.3 Тестирование

Для тестирования `backend`'а был привлечён метод автоматического тестирования на основе приложения «`Postman`». В данном приложении была составлена коллекция запросов и для каждого запроса своя функция тестирования и свой предварительный скрипт. Предварительный скрипт, например, может позволить вставить какой-либо заголовок в запрос, основываясь на данных, пришедших из запроса ранее. Таким образом перед отправкой запросов, требующих токена, например, запрос на удаление позиции из каталога, отправляется промежуточный запрос на получение этого самого токена через логин и хэш пароля. Также к каждому из запросов применяются общие тесты на проверку кода ответа и заголовки ответа. [9]

На рисунке 2.4 представлен отчёт об автоматическом тестировании `backend`'а. Файл для импорта тестов в приложение «`Postman`» предоставлен в архиве курсового проекта.

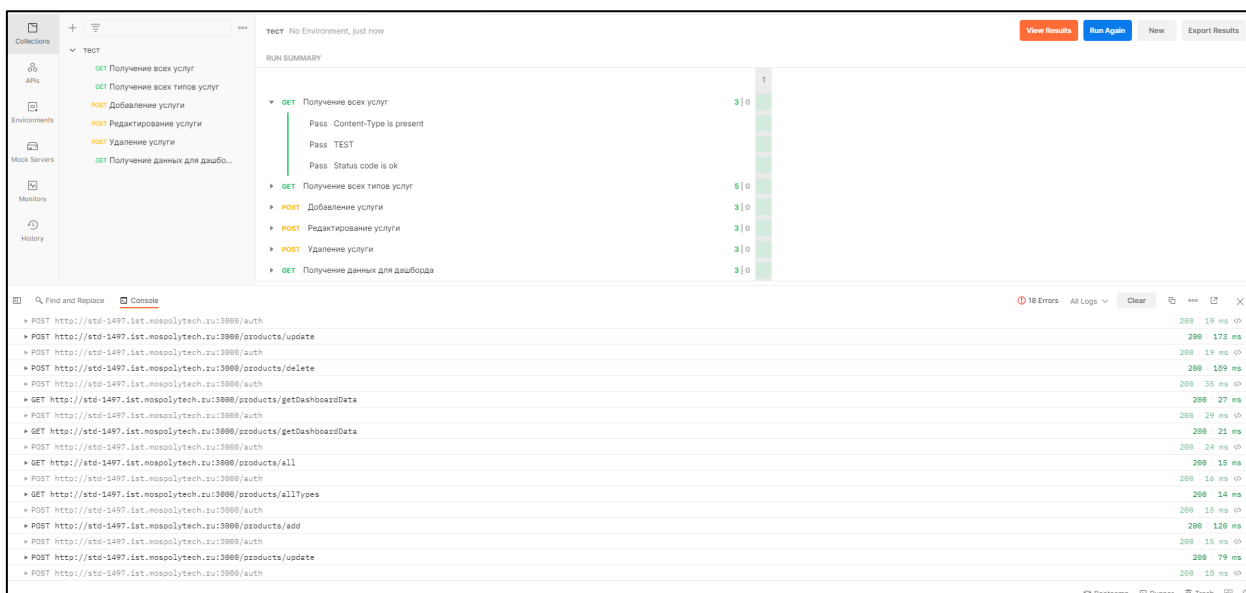


Рисунок 2.5 – Отчёт об автоматическом тестировании backend`а

2.4 Выводы по проектной части

На основе данной проектной части можно изложить следующие выводы:

- была представлена модель данных системы салона красоты — были расписаны три её аспекта: аспект манипуляции, структурный аспект и аспект целостности;
- было представлено подробное описание frontend`а: краткое описание, структура проекта, реализация стилей, особенности функционирования и ручное тестирование в виде тест-кейсов;
- было представлено подробное описание backend`а: краткое описание, структура проекта, особенности функционирования и автоматическое тестирование при помощи приложения «Postman».

ЗАКЛЮЧЕНИЕ

В аналитической части курсовой работы был проведён анализ аналогов разрабатываемой ИС, а именно frontend-части.

Также были представлены формальные модели, описывающие те или иные аспекты разрабатываемой системы: IDEF0-диаграмма с декомпозицией до первого уровня, диаграмма прецедентов и CJM.

В проектной части данной курсовой работы был представлен результат разработки модели данных разрабатываемой ИС: были представлены концептуальная и физическая ER-диаграммы, а так же описаны аспекты манипуляции, структуры и целостности модели данных.

Помимо этого, в проектной части было описано разработанное frontend-приложение: было дано кратное описание с перечнем используемых зависимостей, описана структура проекта, описаны методика применения стилей и элементы дизайна, приведены некоторые особенности функционирования и произведено ручное тестирование по составленным тест-кейсам.

Также в проектной части было описано разработанное backend-приложение: было дано кратное описание с перечнем используемых зависимостей, описана структура проекта, приведены некоторые особенности функционирования и произведено автоматическое тестирование при помощи приложений «Postman».

СПИСОК ИСТОЧНИКОВ

1. БЭМ [Электронный ресурс]. – URL: <https://ru.bem.info/> (дата обращения: 27.06.2021).
2. Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. [Электронный ресурс]. – URL: <https://gosthelp.ru/text/R5010282001Informacionnye.html> (дата обращения: 27.06.2021).
3. Салон красоты на Новорижском шоссе White Fox [Электронный ресурс]. – URL: <http://www.whitefox.ru/> (дата обращения: 27.06.2021).
4. Сеть парикмахерских салонов «Город Красоты» в Москве [Электронный ресурс]. – URL: <https://www.gorod-krasoti.com/> (дата обращения: 27.06.2021).
5. Что такое API? Простое объяснение для начинающих [Электронный ресурс]. – URL: <https://dev.by/news/chto-takoe-api-prostym-yazykom> (дата обращения: 01.12.2020).
6. Beauty Salon Lakelands, Stillorgan | Beautician Lakelands, Stillorgan [Электронный ресурс]. – URL: <https://www.beautifysalon.ie/> (дата обращения: 27.06.2021).
7. ER-диаграмма (ERD): определение и обзор | Lucidchart [Электронный ресурс]. – URL: <https://www.lucidchart.com/pages/ru/erd-диаграмма> (дата обращения: 27.06.2021).
8. Front курсового проекта [Электронный ресурс]. – URL: <http://kozlov.std-1497.ist.mospolytech.ru/> (дата обращения: 28.06.2021).
9. Postman | The Collaboration Platform for API Development [Электронный ресурс]. – URL: <https://www.postman.com/> (дата обращения: 27.06.2021).
10. Vue.js [Электронный ресурс]. – URL: <https://vuejs.org/> (дата обращения: 27.06.2021).
11. Express - фреймворк веб-приложений Node.js [Электронный ресурс]. – URL: <https://expressjs.com/ru/> (дата обращения: 31.03.2021).
12. Использование промежуточных обработчиков Express [Электронный ресурс]. – URL: <https://expressjs.com/ru/guide/using-middleware.html> (дата обращения: 27.06.2021).
13. Солим пароли [Электронный ресурс]. – URL: <https://habr.com/ru/post/145648/> (дата обращения: 27.06.2021).
14. Гридасов А. Что такое дашборд: примеры и способы применения [Электронный ресурс]. – URL: <https://blog.calltouch.ru/dashbord-chto-eto-takoe-dlya-chego-nuzhen-dashboard-i-kak-ego-ispolzovat/> (дата обращения: 27.06.2021).
15. Диаграмма прецедентов (вариантов использования) UML | Планёрка [Электронный ресурс]. – URL: <https://planerka.info/item/diagramma-precedentov-variantov-ispolzovaniya-uml/> (дата обращения: 27.06.2021).
16. AllFusion Process Modeler (Bpwin) — ERwin Process Modeler [Электронный ресурс]. – URL: <https://businessarchitecture.ru/bwin/> (дата обращения: 27.06.2021).