|  |  |
| --- | --- |
| **­­** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ \_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ДОМАШНЕМУ ЗАДАНИЮ***

***НА ТЕМУ:***

***Поставки комплектующих для производства\_\_\_\_\_ электроавтомобилей\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_ИУ5-55Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_Шакиров Т.М.\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.И. Канев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

АННОТАЦИЯ

Расчётно-пояснительная содержит 32 страницы. С приложениями объем составляет 50 страниц. Работа включает в себя 8 диаграмм и 20 изображений системы. В процессе выполнения было использовано 14 источников.

Объектом разработки является сервис управления поставками комплектующих для электрических автомобилей. Этот программный модуль предназначен для оптимизации процессов учета и обработки данных о поставках, обеспечивая эффективное взаимодействие с поставщиками и заказчиками. Система позволяет отслеживать статусы поставок комплектующих, их качество, сроки доставки и другие важные параметры, что способствует улучшению логистики и снижению затрат.

Цель работы заключается в исследовании различных методов управления поставками для электрических автомобилей, а также в создании набора программного обеспечения и сопутствующих веб-сервисов для оптимизации этих процессов.

В ходе работы была разработана архитектура веб-сервиса, обеспечивающего учет поставок комплектующих, разработан интерфейс для взаимодействия с этим веб-сервисом, а также развернут веб-сервер, нативное приложение и прогрессивное веб-приложение, которые позволяют пользователям эффективно взаимодействовать с сервисом и получать актуальную информацию о поставках.

Пояснительная записка содержит 2 приложения.

Содержание

[АННОТАЦИЯ 1](#_Toc185592768)

[ВВЕДЕНИЕ 4](#_Toc185592769)

[1 ПРЕДМЕТНАЯ ОБЛАСТЬ 6](#_Toc185592770)

[2 АРХИТЕКТУРА 13](#_Toc185592771)

[3 АЛГОРИТМЫ 17](#_Toc185592772)

[4 ОПИСАНИЕ ИНТЕРФЕЙСА 19](#_Toc185592773)

[ЗАКЛЮЧЕНИЕ 30](#_Toc185592774)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc185592775)

[ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ 33](#_Toc185592776)

[1. Введение 34](#_Toc185592777)

[2. Назначение разработки 34](#_Toc185592778)

[3. Стадии и этапы разработки 34](#_Toc185592779)

[4. Требования к функциональным характеристикам 35](#_Toc185592780)

[5. Требования к составу и параметрам технических средств 39](#_Toc185592781)

[6. Требования к информационной и программной совместимости 40](#_Toc185592782)

[ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ 41](#_Toc185592783)

ВВЕДЕНИЕ

Правительство Российской Федерации уделяет особое внимание развитию отечественного производства автомобилей, в том числе электромобилей, в контексте глобальных тенденций устойчивого развития и экологической политики. В 2024 году, согласно аналитике EV-volumes, объем мирового рынка электромобилей продолжает расти с темпом в 35-40% ежегодно. В России также наблюдается значительное увеличение продаж электромобилей. Только за первый квартал 2024 года было продано более 1 000 электрокаров, что в два с половиной раза больше, чем за аналогичный период 2023 года. Важным фактором такого роста является не только увеличивающийся спрос, но и активное развитие инфраструктуры для зарядки электромобилей, а также поддержка со стороны государства.

Целью данной работы является разработка системы для автоматизации поставок автозапчастей для электромобилей, состоящей из веб-сервиса, веб-приложения, десктопного приложения и выделенного сервиса для переговоров с поставщиками.

Назначение системы – создание эффективного взаимодействия между заказчиками автозапчастей, производителями и поставщиками, что поможет улучшить процесс закупки, учета и управления запасами автокомплектующих. В рамках системы заказчики могут создавать заявки на необходимое оборудование и автозапчасти, выбирать поставщиков и отслеживать статусы своих заявок. Также система позволяет редактировать или удалять заявки, если это необходимо.

Поставщики и совет директоров компании, в свою очередь, имеют возможность просматривать и одобрять или отклонять заявки. Помимо этого, они могут вносить изменения в каталог автозапчастей, добавлять новые позиции, редактировать существующие или удалять их. Система предоставляет удобный интерфейс для управления всеми этапами взаимодействия – от создания заявки до завершения поставки.

Таким образом, предложенная система обеспечит надежную и прозрачную платформу для управления поставками автозапчастей, отвечая потребностям растущего рынка электромобилей и способствуя его дальнейшему развитию.

Нефункциональные требования к разрабатываемой системе:

1. Должна поддерживать кроссплатформенность.
2. Интерфейс системы и текст ошибок должны быть русифицированы.

В ходе работы необходимо выполнить следующие задачи:

1. Разработать дизайн приложения в Figma на основе teslashop.ru. Ознакомиться с разработкой бэкенда с использованием фреймворка Django.
2. Разработать структуру и создать базу данных PostgreSQL, подключить её к бэкенду.
3. Создать веб-сервис со всей итоговой бизнес-логикой в бэкенде системы для использования его в SPA.
4. Добавить авторизацию, использовать Redis для хранения сессий, а также внедрить Swagger в веб-сервис.
5. Разработать базовый интерфейс приложения для гостя на React.
6. Внедрить менеджер состояний Redux Toolkit для хранения значений фильтров, добавить адаптивность и PWA.
7. Завершить разработку интерфейса пользователя в React, использовать для обращений к методам веб-сервиса Axios.
8. Реализовать React интерфейс издатель, внедрить Real-time web.
9. Разработать десктопное приложение на Tauri.
10. Развернуть приложение на GitHub Pages.
11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта.
13. Реализовать пагинацию с фильтрацией по индексу в React-приложении.

1 ПРЕДМЕТНАЯ ОБЛАСТЬ

Исследование экспоненциального распространения батарейных электромобилей (Battery Electric Cars, BEC), охватывающее исторические данные их внедрения в 17 странах, включая Европу и мировую статистику, является очень значимым в современном мире. Основное внимание уделяется пассажирским автомобилям с батарейным приводом и прогнозам их доминирования в автопарках в ближайшие годы [1].

Анализ показывает, что темпы роста автопарка батарейных электромобилей во многих странах следуют экспоненциальной модели, что подтверждается данными за последние годы. В 2022 году общее количество BEC достигло 11 миллионов единиц по всему миру, что составило 1,4% мирового автопарка [2]. При этом в Европе доля BEC составила 2,3%, а в США — 0,9% [3]. Лидерами стали Норвегия и Исландия, где доля электромобилей превысила 25%, в то время как в Польше этот показатель остаётся на уровне менее 0,2%. Такие различия свидетельствуют о значительных региональных различиях в скорости внедрения электромобилей.

Согласно прогнозам, автопарк будет существенно трансформирован в ближайшие годы. Для моделирования будущего роста использовались три подхода: экспоненциальная модель, логистическая модель и модель диффузии Басса. Экспоненциальная модель подтвердила, что годовые темпы роста BEC составляют около 55%, что эквивалентно времени удвоения менее чем в 15 месяцев. Средний темп роста BEC соответствует времени удвоения менее чем в 1,26 года, что подтверждается стандартным отклонением темпов роста, равным σa = 0,09. Эти данные указывают на то, что доля BEC в мировом автопарке достигнет 31% к 2030 году даже при самых осторожных прогнозах, а в Европе половина автопарка станет электрической между 2030 и 2033 годами [4].

Эти прогнозы имеют ключевое значение для компаний, занимающихся поставками комплектующих для электрических автомобилей. Увеличение автопарка BEC непосредственно влияет на спрос на детали, такие как аккумуляторные батареи, электродвигатели и зарядные устройства. Это создаёт условия для роста объёмов заказов, что требует от компаний гибкости в адаптации логистических цепочек. Увеличение количества заказчиков (car manufacturers), таких как крупные производители электромобилей, предполагает более высокие стандарты качества и сокращение сроков поставок. Например, в статье подчёркивается, что развитие зарядной инфраструктуры тесно связано с ростом заказов на зарядные станции и сопутствующее оборудование, что становится важным аспектом взаимодействия компаний с клиентами.

Кроме того, решения совета директоров Европейской комиссии (Directorate-General of the European Commission) играют критическую роль в стратегическом развитии компаний. Инвестиции в расширение производственных мощностей, использование инновационных технологий и обеспечение устойчивых цепочек поставок становятся ключевыми элементами, направленными на удовлетворение растущего спроса на электротранспорт. Такие решения также способствуют увеличению рыночной доли компаний и укреплению их позиций в глобальной цепочке добавленной стоимости [5].

Практическая значимость исследования подчёркивает, что экспоненциальный рост электромобилей оказывает влияние на всю цепочку поставок. Автопарк изменится в сторону значительного увеличения доли электрических автомобилей, что потребует модернизации производственных мощностей и инфраструктуры [6]. Например, в Европе прогнозируется, что к 2029 году количество BEC будет в три тысячи раз больше, чем автомобилей на водородных топливных элементах, даже с учётом их высоких темпов роста (69% в год).

Государственная политика играет ключевую роль в стимулировании роста BEC и изменении автопарка. Европейский Союз обязал все новые автомобили, продаваемые с 2035 года, быть нулевыми по выбросам. Кроме того, ускоренное развитие зарядной инфраструктуры, включая увеличение числа быстрых зарядных станций, делает BEC более привлекательными для потребителей. Положительные эффекты обратной связи, такие как снижение цен благодаря масштабам производства и внедрение новых технологий, также способствуют ускоренному росту.

Альтернативные технологии, такие как водородные топливные элементы или синтетические топлива, обладают потенциалом, однако их вклад в автопарк остаётся незначительным. Например, в 2029 году парк BEC в Европе будет в тысячу раз больше, чем количество автомобилей на водородных топливных элементах. Даже при высоких темпах роста последних они вряд ли смогут конкурировать с батарейными электромобилями в ближайшие годы [7].

Результаты исследования подчёркивают, что экспоненциальный рост BEC обусловлен не только экономическими факторами, но и политической поддержкой, развитием инфраструктуры и эффектами масштаба. Эти факторы, вероятно, будут усиливаться, ускоряя изменения в автопарке и переход к доминированию электромобилей в мировом автопарке.

Таким образом, компании в области электротранспорта и поставок должны быть готовы к активной адаптации к изменениям на рынке. В условиях роста спроса на инновационные детали и увеличения объёмов заказов от производителей электромобилей важно наладить устойчивую и гибкую систему поставок, которая будет соответствовать ожиданиям заказчиков и стратегическим целям советов директоров.

В ответ на высокий спрос был разработан сервис, который позволяет облегчить поиск деталей для электромобилей. Вместо того чтобы искать детали вручную, пользователи могут оставлять заявки, которые автоматически направляются к поставщикам, а затем выбрать наиболее подходящие детали. Каждая заявка начинается как черновик, в который можно добавлять или удалять детали. В рамках одной заявки можно указать несколько позиций.

Для каждой детали можно указать необходимое количество. После того как заказчик окончательно определится с выбором деталей, он подтверждает заявку, и она отправляется на проверку. После модерации заявку уже нельзя редактировать. Также есть возможность отслеживать историю своих заявок.

Чтобы гарантировать, что количество и тип деталей соответствуют ожиданиям, заявка проходит проверку советом директоров, который принимает или отклоняет заявки для обеспечения корректности процесса. В случае, если заказчик решит отменить заявку, эта функция также предусмотрена.

За рассмотрение заявки отвечает выделенный сервис, который проверяет информацию о поставщиках и её возможности поставлять заданные детали. Функции пользователей с разными ролями описаны на диаграмме прецедентов (рис. 1).

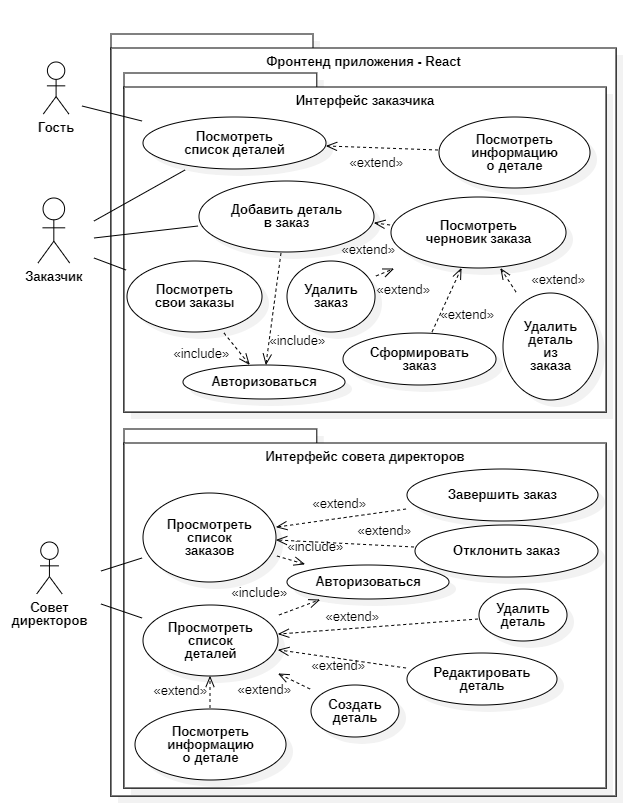


Рисунок 1 – Диаграмма прецедентов

Гостям доступен просмотр деталей. Зарегистрированные гости – заказчики. Они могут добавлять детали в заявку, просматривать список своих заявок и сформировывать текущую заявку. Заявки обрабатываются советом директоров. В результате обработки заявки её либо одобряют, либо отклоняют. Совету директоров также доступны уникальные функции для работы с деталями, а именно: просмотр всех деталей, редактирование, создание и удаление деталей. Процесс оформления заявки отражен на диаграмме деятельности (рис. 2).

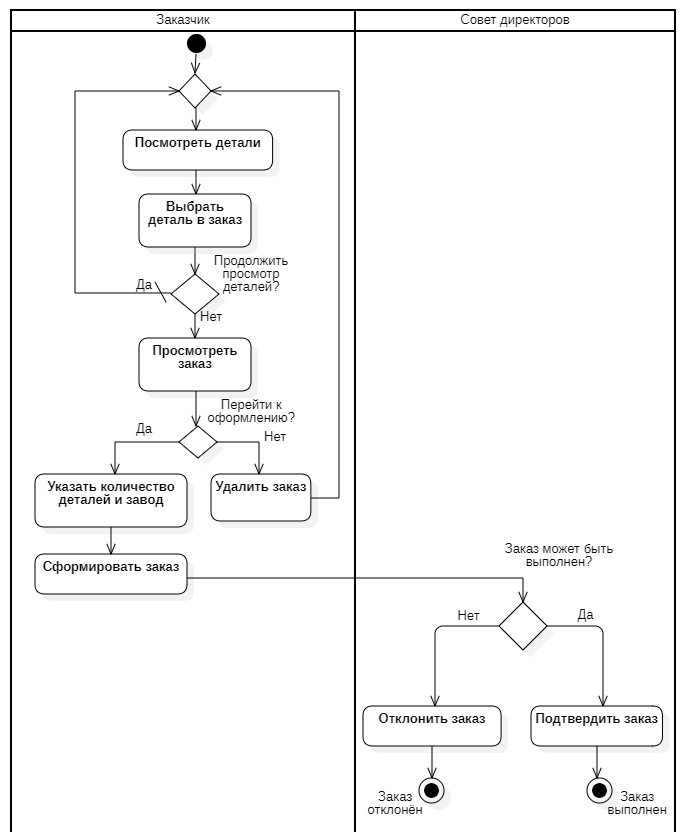


Рисунок 2 – Диаграмма деятельности

Заказчик выбирает детали, затем формирует на основе выбранных деталей заявку. Эту заявку обрабатывает совет директоров. Возможные состояния заявки отражены на диаграмме состояний заявки (рис. 3).

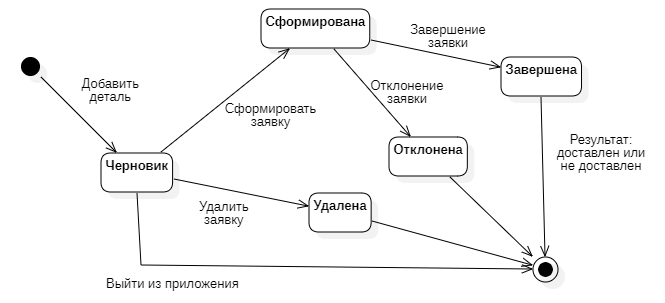


Рисунок 3 – Диаграмма состояний заявки

При выборе первой автозапчасти формируется черновик. Последующие выбранные автозапчасти добавляются в этот черновик. Заказчик затем формирует заявку, удаляет её или выходит из приложения. Сформированную заявку обрабатывает совет директоров. Они могут одобрить или отклонить её.

2 АРХИТЕКТУРА

Архитектура системы отображена на диаграмме развертывания (рис. 4).

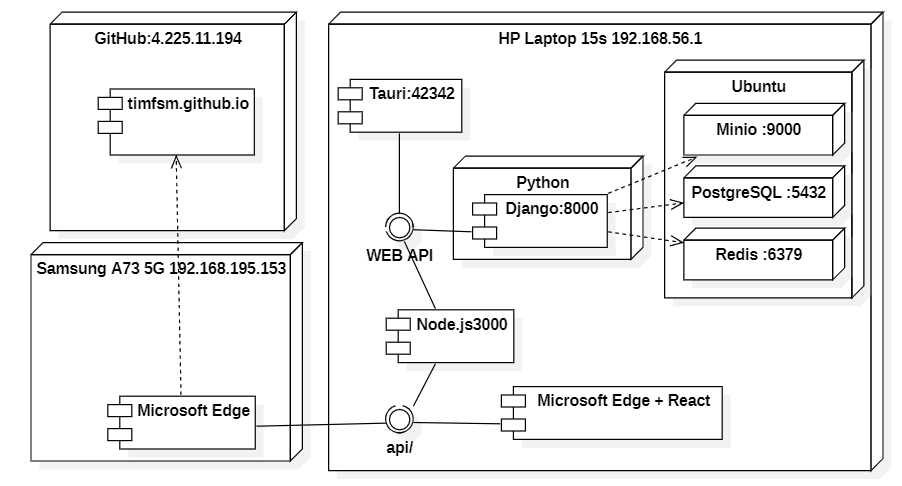


Рисунок 4 – Диаграмма развертывания

Веб-сервис, реализованный на фреймворке Django [8] связан с серверами Redis [9], Minio [10] и СУБД PostgreSQL [11]. В Redis осуществляется хранение активных сессий пользователей, а при выходе пользователя из системы – их удаление.

В качестве языка программирования был выбран Python [12], отличающийся простотой восприятия и популярностью. Использование фреймворка Django обусловлено широким спектром его возможностей: ORM [13], административная панель и наличие библиотек для решения специализированных задач, например, для упрощения работы с API. Веб-сервис на Django является общим для веб-сервера и для десктопного приложения Tauri [14].

Согласно с текущей политикой импортозамещения, для хранения данных была выбрана СУБД PostgreSQL. Выбранная СУБД является стандартом индустрии.

Структура данных приведена на ER-диаграмме (рис. 5). Она довольна проста. Сведения о деталях хранятся в таблице Товар. Данные о заявках хранятся в таблице Заказ. Для хранения в одной заявке нескольких деталей используется промежуточная таблица Товар\_заказ, с помощью которой реализуется связь «многие ко многим». Данные о пользователях системы – заказчиках и совете директоров – хранятся в таблице Пользователь.

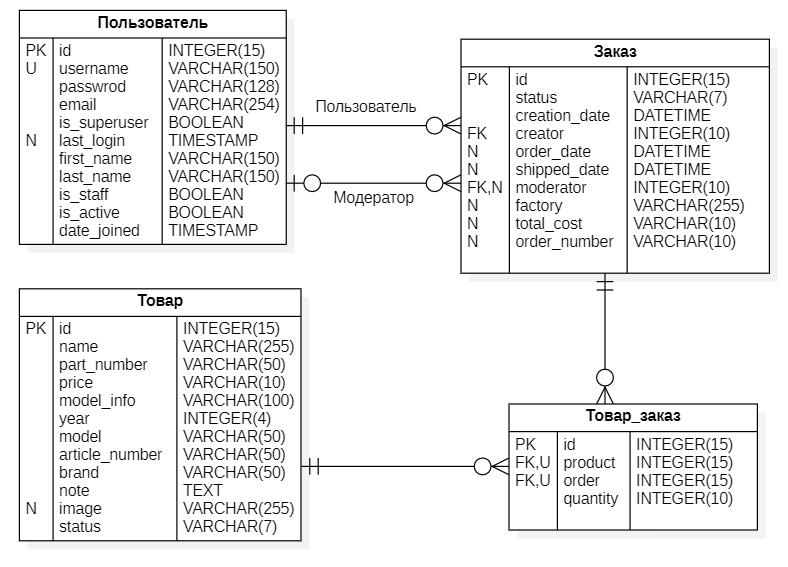


Рисунок 5 – ER-диаграмма

Устройство бэкенда приложения разработанной системы отражено на диаграмме классов бэкенда (рисунок 6). Пользователи взаимодействуют с доменами, которые, в свою очередь, связаны с моделями. Модели имеют связи с таблицами в базе данных.

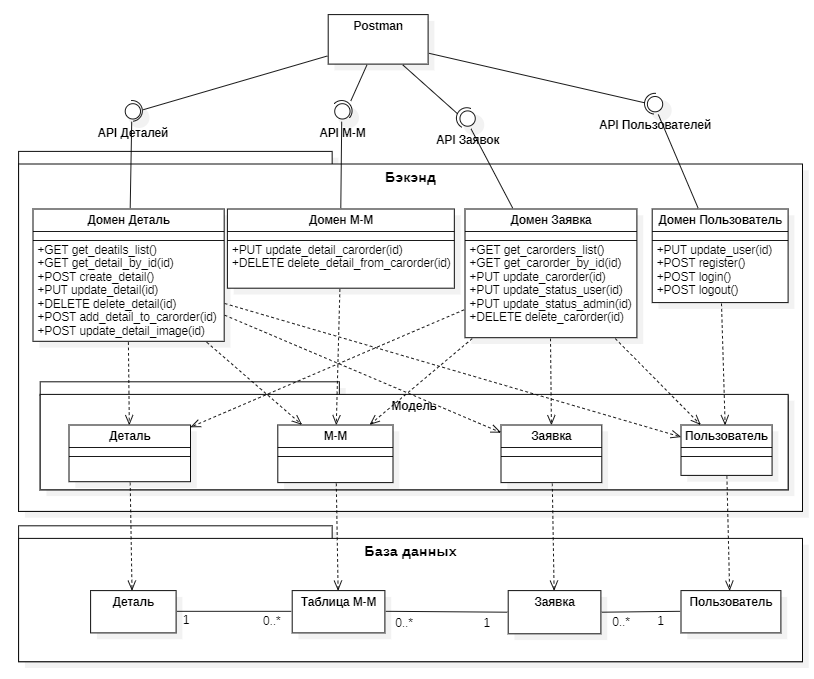


Рисунок 6 – Диаграмма классов бэкенда

Связь фронтенда и бекенда отражена на диаграмме классов фронтенда (рис. 7). Ключевые страницы имеют связь с API аутентификации, т.к. доступ к ним осуществляется только для авторизированных пользователей с определенными правами (ролями).

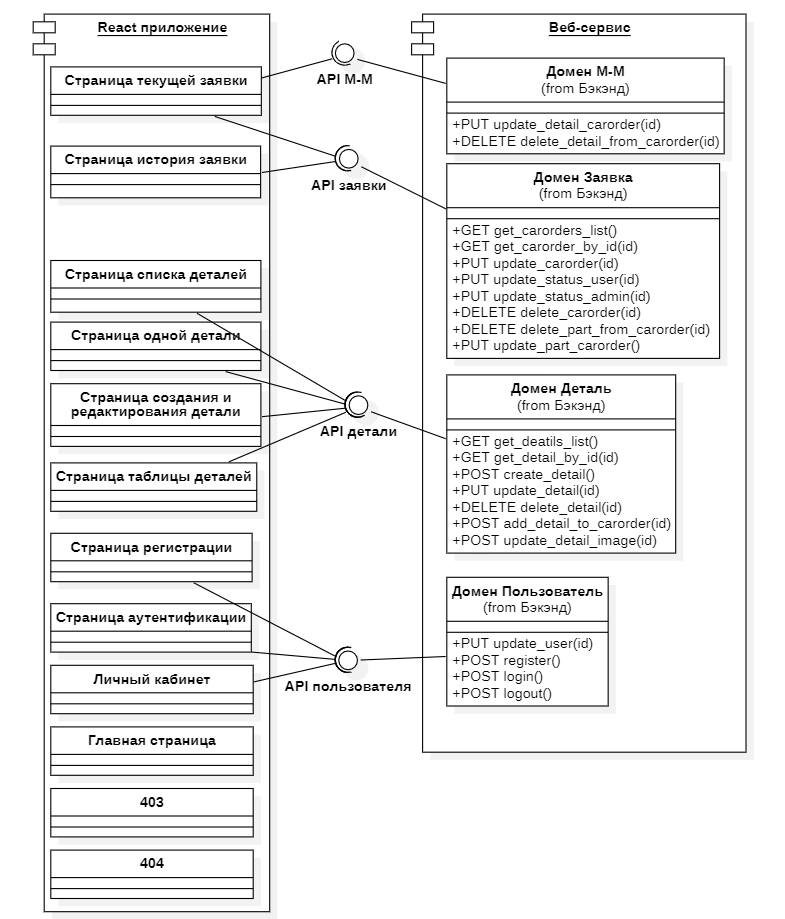


Рисунок 7 – Диаграмма классов фронтенда

3 АЛГОРИТМЫ

Алгоритм работы разработанной системы отображен на диаграмме последовательности (рис. 8). В основе системы лежит веб-сервис, внутри которого реализована вся бизнес-логика. Он предоставляет доступ к методам следующих доменов: детали, заявки, пользователи. Методы следуют правилам REST API.

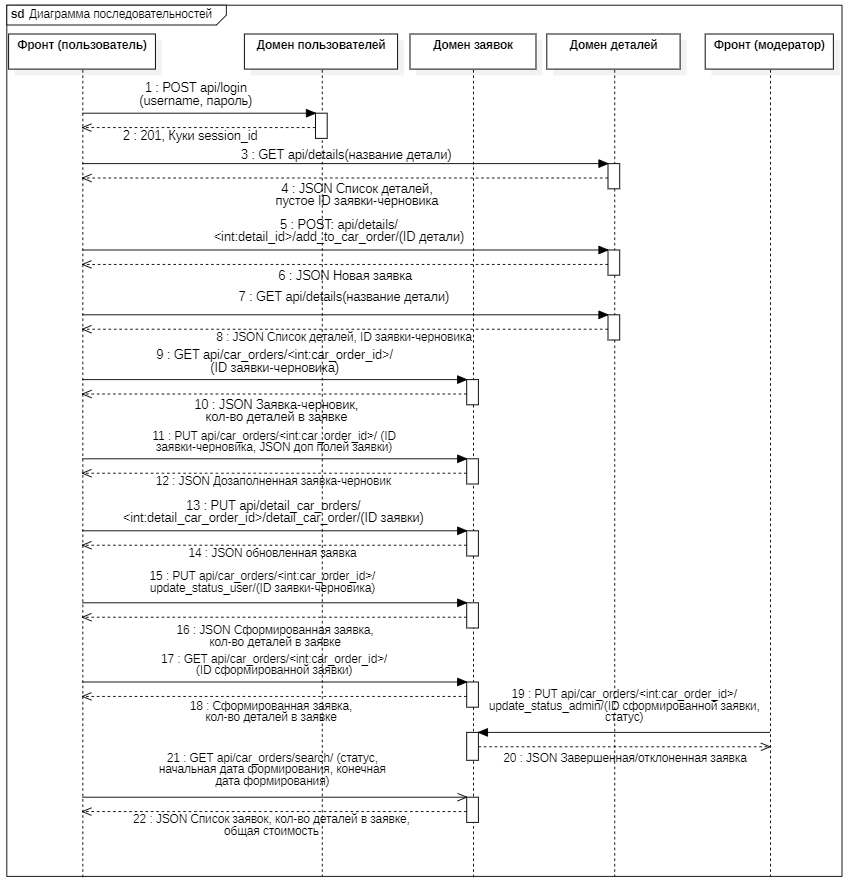


Рисунок 8 – Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация заказчика. Для этого он отправляет через графический интерфейс запрос, передавая в нем логин и пароль. Если аккаунт с указанными данными существует в базе, на клиент возвращается информация о пользователе и устанавливаются куки с полем session\_id (идентификатор текущей сессии) в ответном запросе. Если же такого аккаунта не существует, или пароль введен неверно, заказчик получит ошибку. В таком случае ему надо либо пройти регистрацию, либо ввести пароль верно. На этом же этапе происходит проверка: является пользователь заказчиком или членом совета директоров. Затем графический интерфейс заказчика запрашивает у веб-сервиса список деталей, которые возвращаются в JSON формате. Заказчик выбирает деталь, которую хочет добавить в заявку, и, нажимая на кнопку «В корзину» в графическом интерфейсе, отправляет запрос на добавление детали в свою черновую заявку. Этот процесс может продолжаться несколько раз.

Когда заказчик определится с выбором, он нажимает на кнопку «Сформировать» в графическом интерфейсе. После этого приложение запрашивает номер черновой заявки и затем отправляет запрос на формирование этой заявки. В этот момент основной веб-сервис выполняет асинхронный запрос к другому сервису, чтобы он рассмотрел заявку и вернул статус с решением в основной сервис. Заказчик может отслеживать статус сформированных заявок на соответствующей странице в графическом интерфейсе.

Процесс рассмотрения заявки происходит также через графический интерфейс. Совет директоров может просматривать список всех заявок и, нажимая на соответствующие кнопки, отправлять запросы на подтверждение или отклонение заявки в основной веб-сервис. В эти запросы также можно включить фильтры по имени создателя заявки, одному из статусов заявки и диапазону дат, в которых эти заявки были сформированы. Также совету директоров может управлять заявками через графический интерфейс. Им доступны такие функции, как создание и редактирование заявок, просмотр списка заявок в виде таблицы и удаление их. Для каждой из этих функций присутствует свой метод, отправляемый на основной веб-сервис.

4 ОПИСАНИЕ ИНТЕРФЕЙСА

Главное меню приложения включает разделы, доступность которых зависит от роли пользователя. Рисунок 9 иллюстрирует разделы доступные гостю.



Рисунок 9 – Меню приложения для гостя

На рисунке 10 изображено меню, которое видит заказчик.

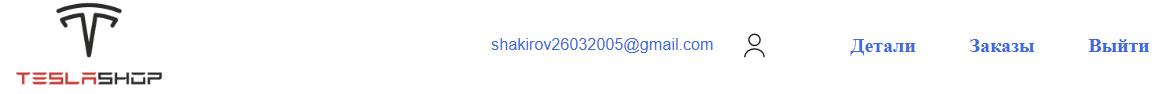


Рисунок 10 – Меню приложения для заказчика

Разделы, доступные совету директоров, отображает рисунок 11.



Рисунок 11 – Меню приложения для совета директоров

На странице с формой регистрации (рис. 12) отображается форма, при помощи которой гость может создать новый аккаунт. После успешной регистрации открывается главная страница приложения.

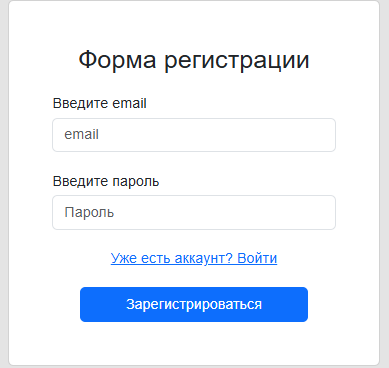


Рисунок 12 – Форма регистрации

На странице с формой входа (рис. 13) отображается форма, через которую пользователь может войти в свой аккаунт. При успешном вводе данных аккаунта на клиент приходят куки с идентификатором текущей сессии.

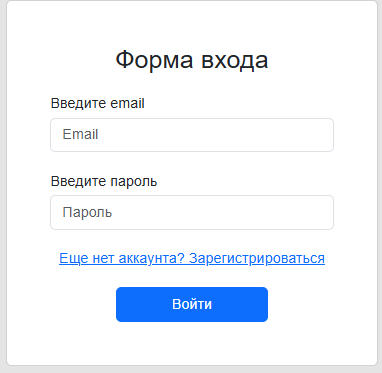


Рисунок 13 – Форма входа

После входа в аккаунт пользователь попадает на главную страницу (рис. 14), на которой располагается информация о магазине комплектующих для электрокаров.

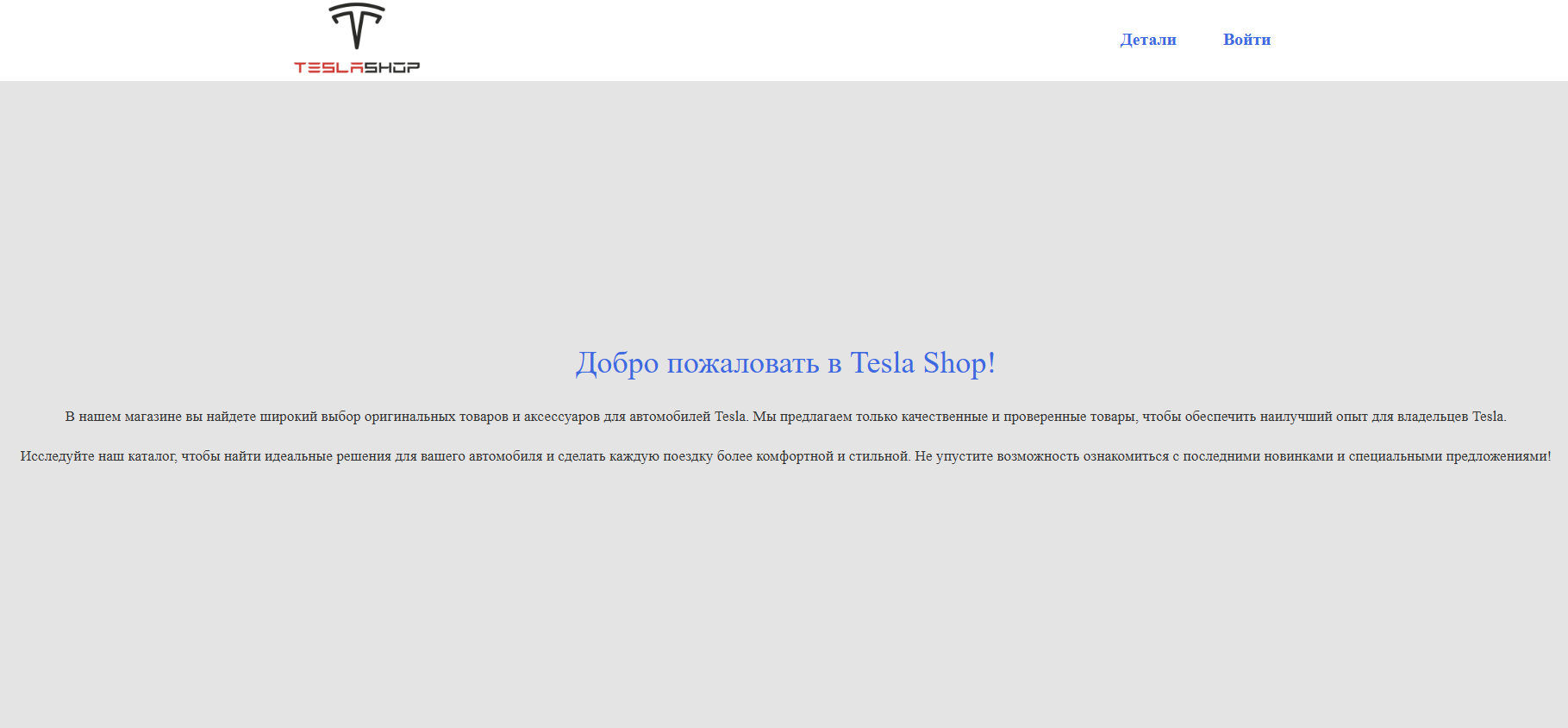


Рисунок 14 – Главная страница

Страница списка услуг магазина комплектующих для электрокаров (рис. 15) содержит список услуг в виде карточек. При нажатии на название товара на карточке, открывается страница с подробным описанием выбранной услуги.

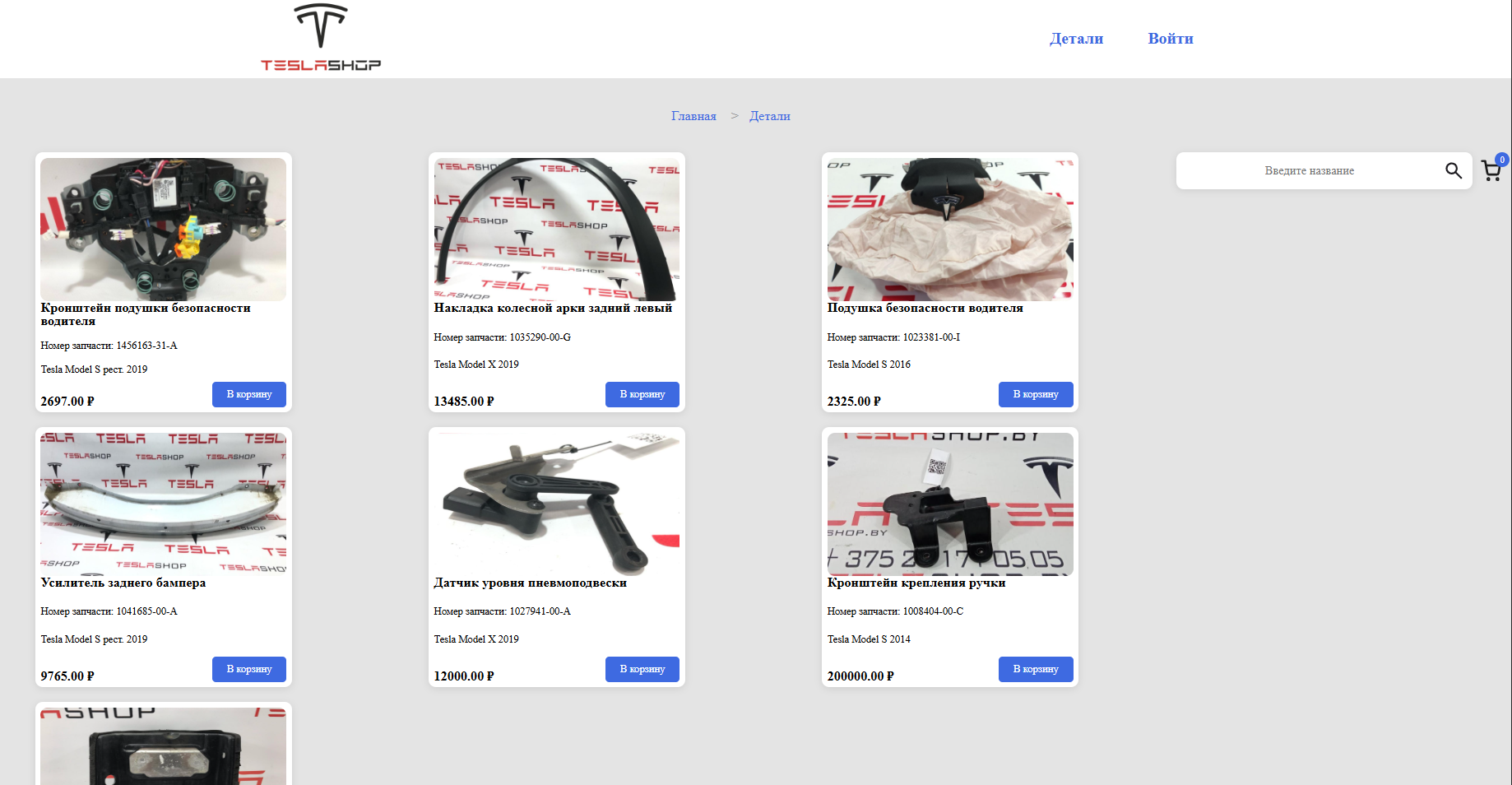


Рисунок 15 – Страница списка услуг магазина комплектующих для электрокаров для гостя

На странице услуги представлена информация о наименовании и стоимости услуги, а также её характеристики и подробное описание (рис. 16).

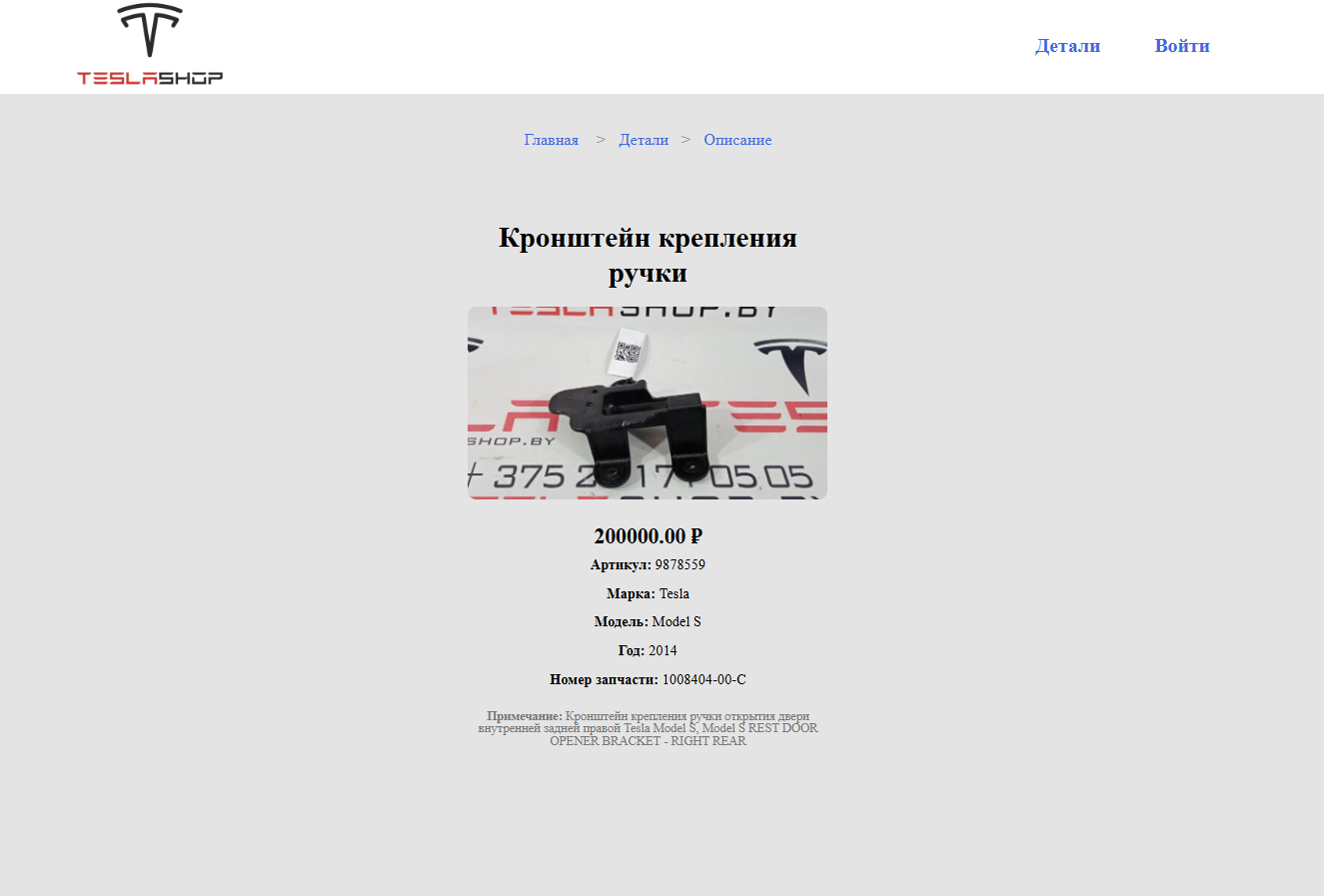


Рисунок 16 – Страница с описанием услуги магазина

В отличие от гостя заказчика на карточке каждой услуги кнопка «В корзину» активна (рис. 17), с помощью которой он может добавить услугу в черновик заказа. Просмотреть черновик заказа, при его наличии, заказчик может, нажав на соответствующую кнопку в правом верхнем углу страницы. Вверху страницы также располагается поисковая строка, позволяющая найти интересующую услугу по наименованию. Поиск не чувствителен к регистру.

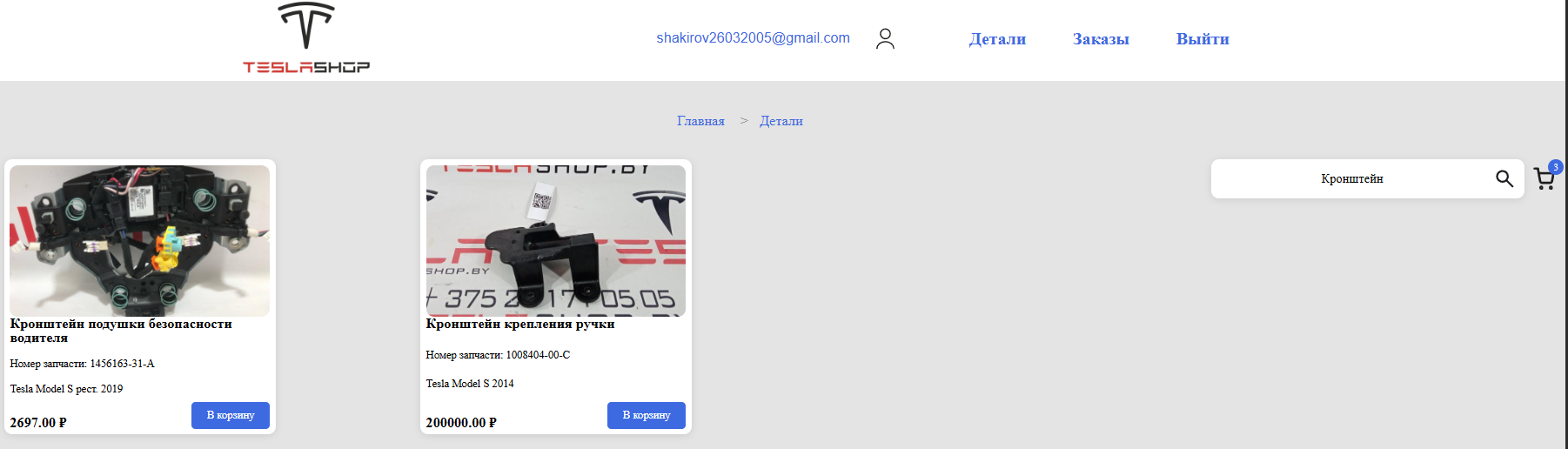


Рисунок 17 – Страница списка услуг магазина комплектующих для электрокаров для заказчика

На странице заказа (рис. 18) отображается текущий черновой заказ. Пользователь может удалить услугу из проекта нажатием «крестика» в правом верхнем углу карточки. Заказчику предоставляется возможность изменить количество определённой детали при помощи кнопок «-» и «+». По умолчанию установлено количество равное единице. Кнопки «Удалить заказ» и «Сформировать» позволяют выполнить одноименные операции над заказом. Редактирование оформленных заказов недоступно.

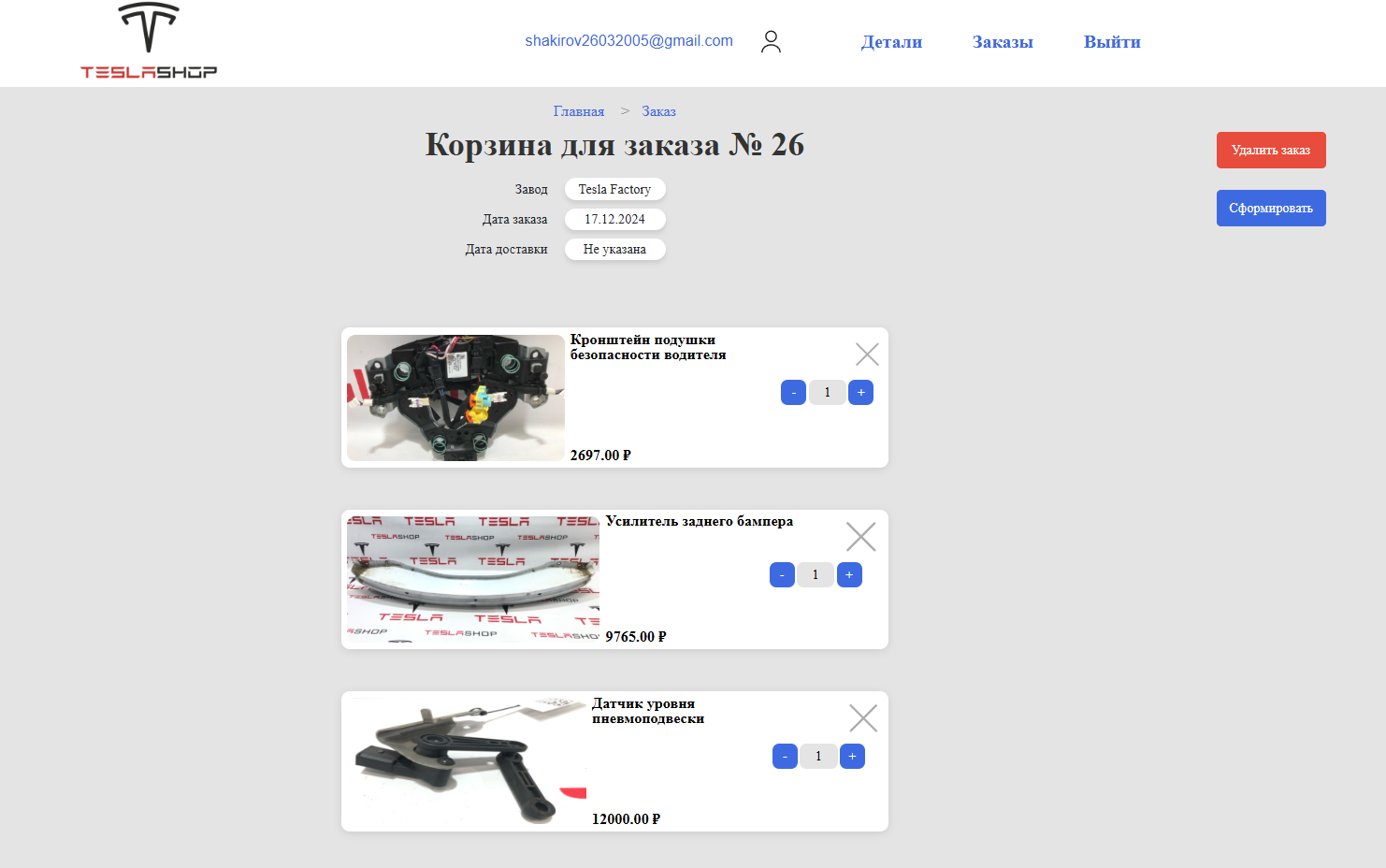


Рисунок 18 – Страница заказа

На странице списка заказов (рис. 19) заказчики могут просматривать созданные ими заказы в виде таблицы. На этой странице можно перейти к подробной информацию о заказе, нажав на его карточку. Также пользователю доступны сортировка заказов дате завершения и по статусу.

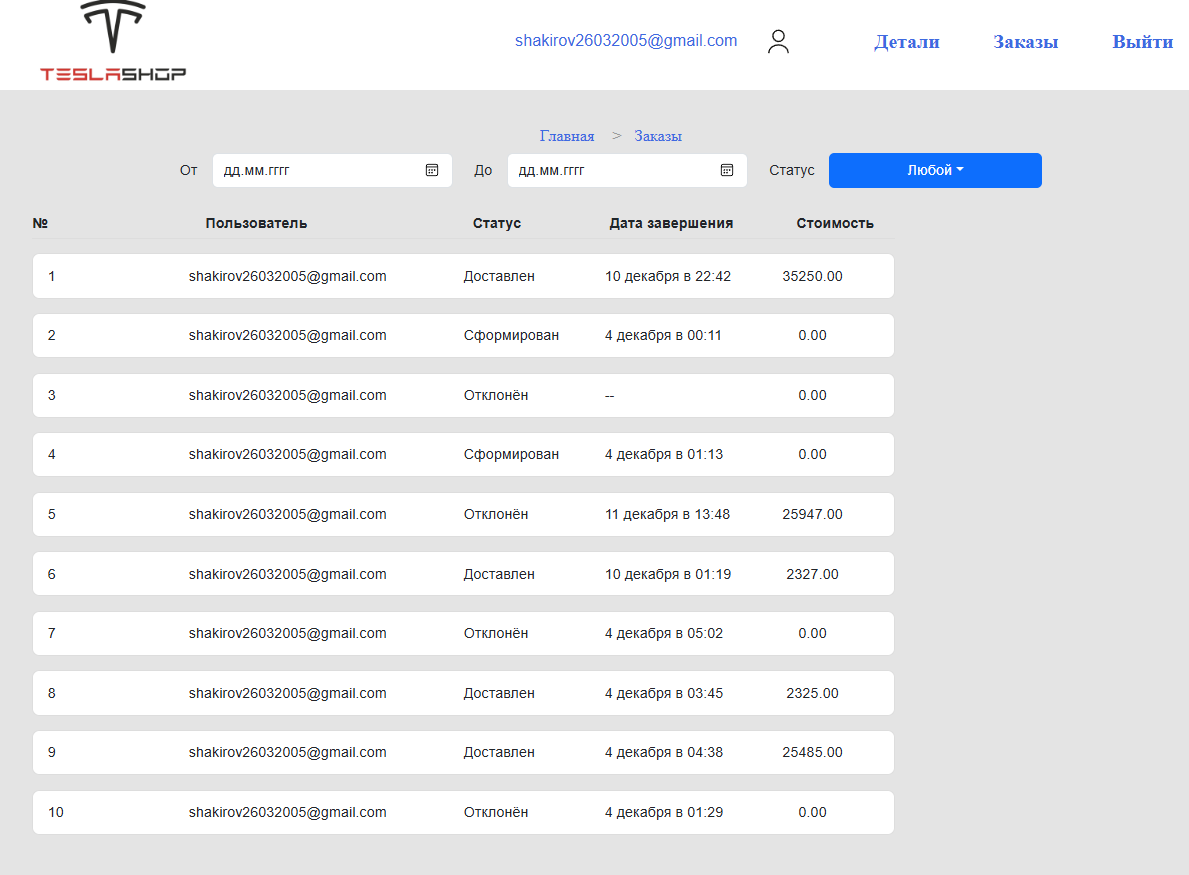


Рисунок 19 – Страница списка заказов для заказчика

Страница списка всех заказов (рис. 20) предоставляет расширенный функционал для совета директоров: он может завершать или отклонять заказы путём нажатия одноименных кнопок в интерфейсе. На странице доступна фильтрация по логину пользователя, статусу и датам формирования заявки. Заказчик может просматривать подробную информацию о заказе путём нажатия на него.

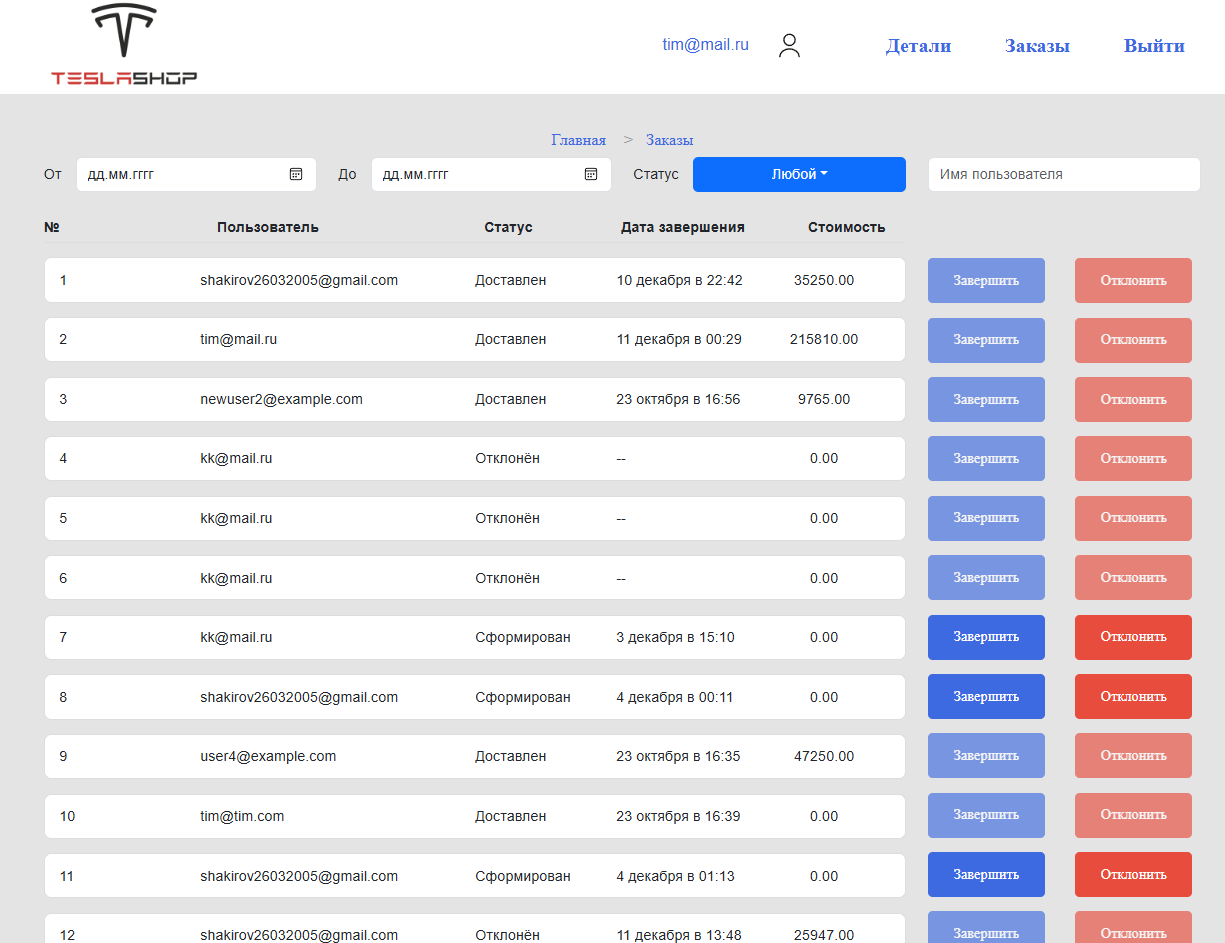


Рисунок 20 – Страница списка заказов для совета директоров

Как заказчик, так и совет директоров могут изменить свои данные (адрес электронной почты и пароль) на странице редактирования данных личного кабинета (рис. 21). Личный кабинет открывается при нажатии на иконку пользователя в меню.

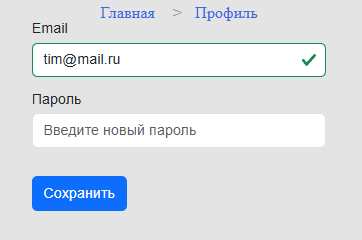


Рисунок 21 – Страница личного кабинета

Совет директоров может просматривать информацию о всех услугах магазина комплектующих для электрокаров (рис. 22). Помимо этого, он имеет возможность добавить новую услугу или удалить существующую, а также перейти к редактированию любой услуги из списка путем нажатия кнопок «Добавить», «Удалить», «Изменить» соответственно.

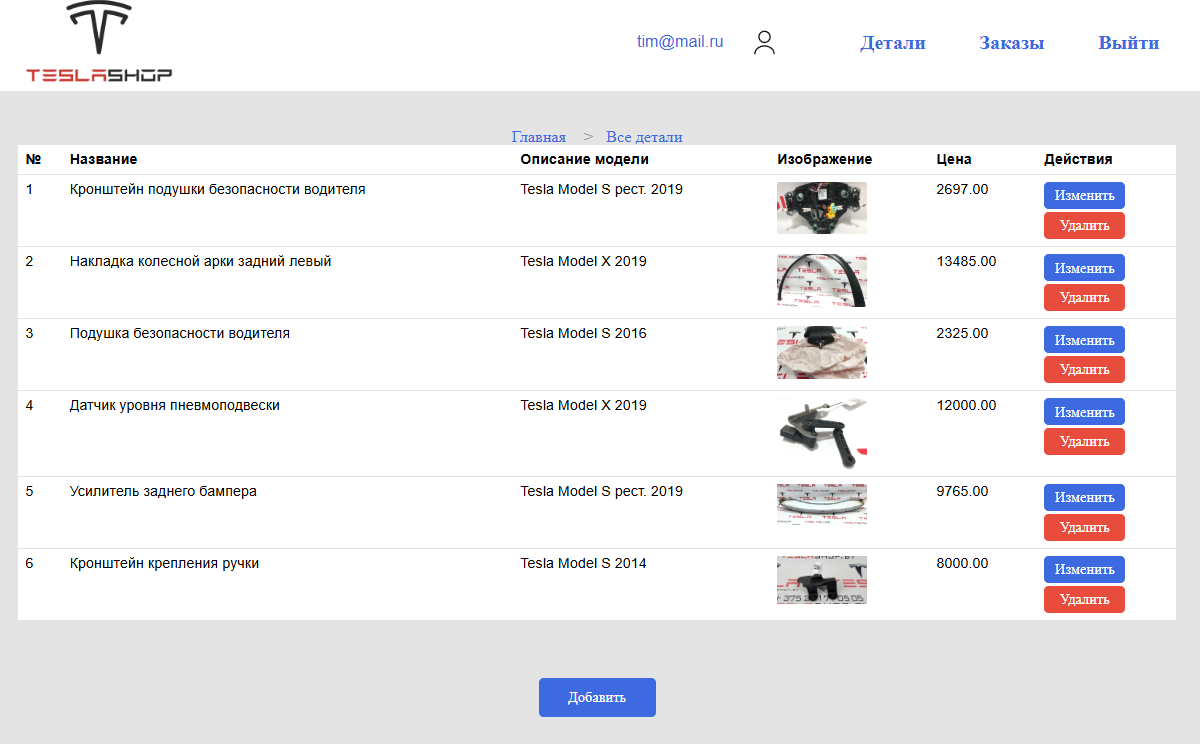


Рисунок 22 – Страница списка услуг для совета директоров

На странице редактирования (рис. 23) и создания (рис. 24) услуги магазина комплектующих для электрокаров совет директоров может изменить существующую или добавить новую услугу соответственно. Все поля, кроме картинки и примечания, являются обязательными для заполнения. Если совет директоров попытается сохранить изменения или новую услугу с незаполненным обязательным полем, то появится уведомление, информирующее о невозможности выполнения операции. Перейти к выбору файла изображения услуги можно, нажав на существующее фото. Затем появится диалоговое окно выбора файла. После завершения выбора требуется нажать кнопку «Выбрать». Для удобства набора описания услуги поле обладает свойством масштабируемости, оно может быть увеличено растягиванием за правый нижний угол. Кнопки «Сохранить» и «Добавить» позволяют сохранить внесенные изменения и сохранить новую услугу соответственно.

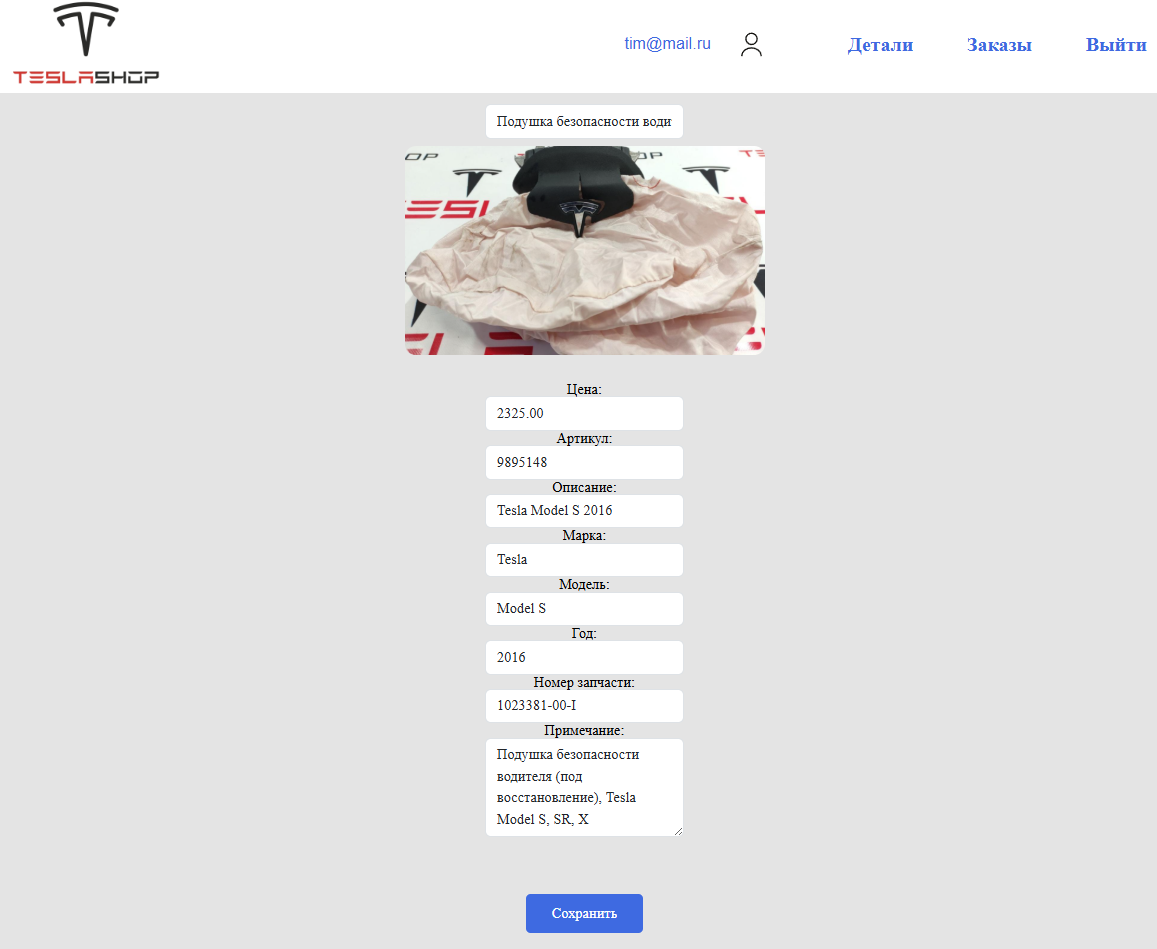


Рисунок 23 – Страница редактирования услуги

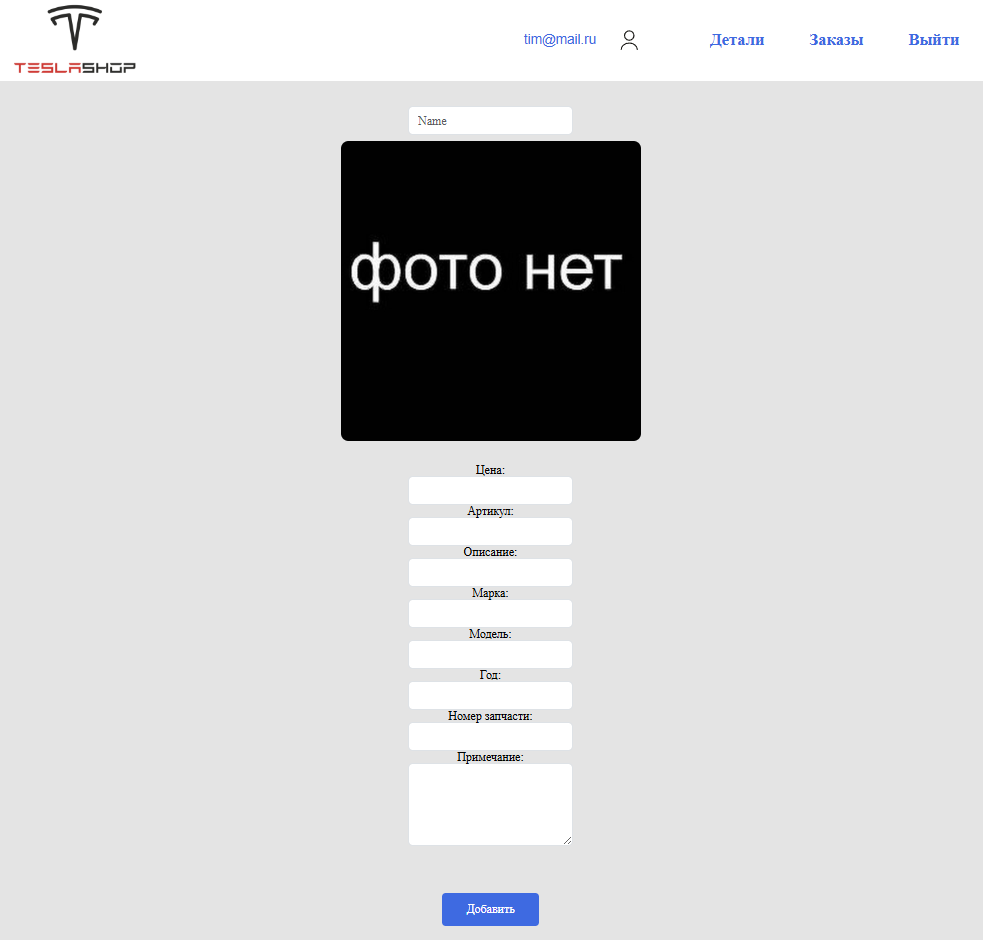


Рисунок 24 – Страница добавления услуги

В случае, если пользователь пытается получить доступ к страницам, для просмотра которых ему не хватает прав, он перенаправляется на страницу ошибки 403 (рис. 25).

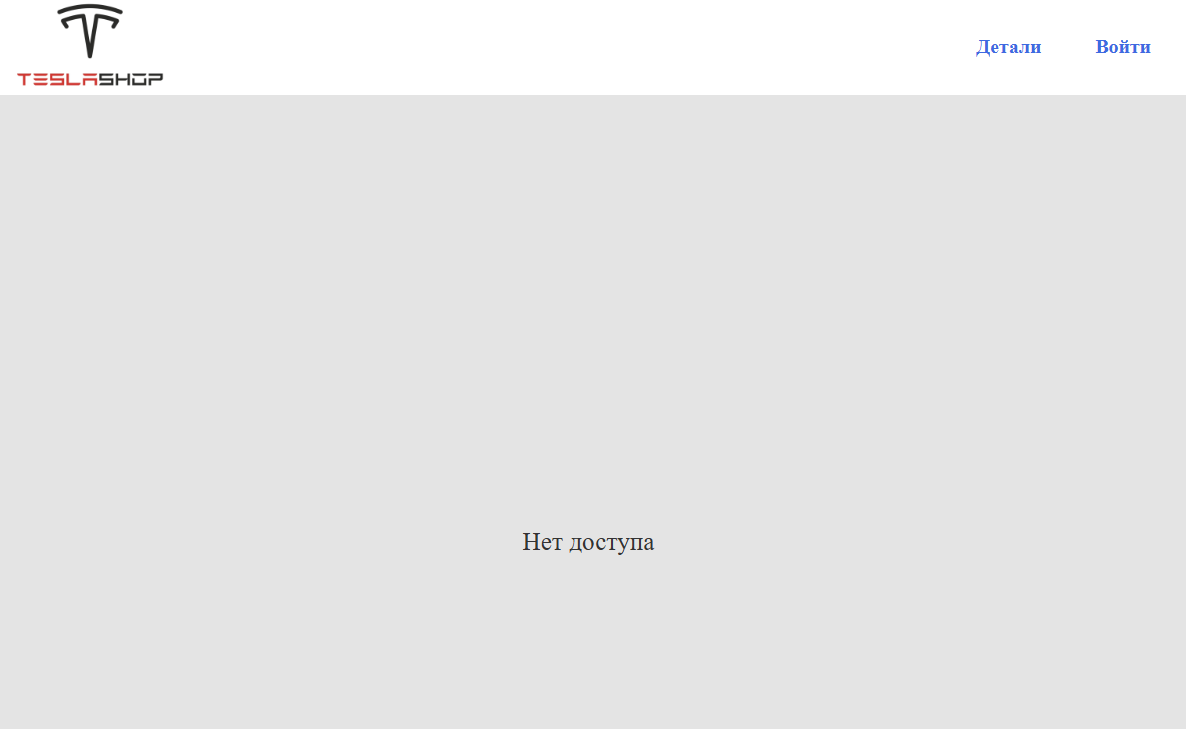


Рисунок 25 – Страница ошибки 403

В случае, если пользователь запрашивает несуществующую станицу, он перенаправляется на страницу ошибки 404 (рис. 26).

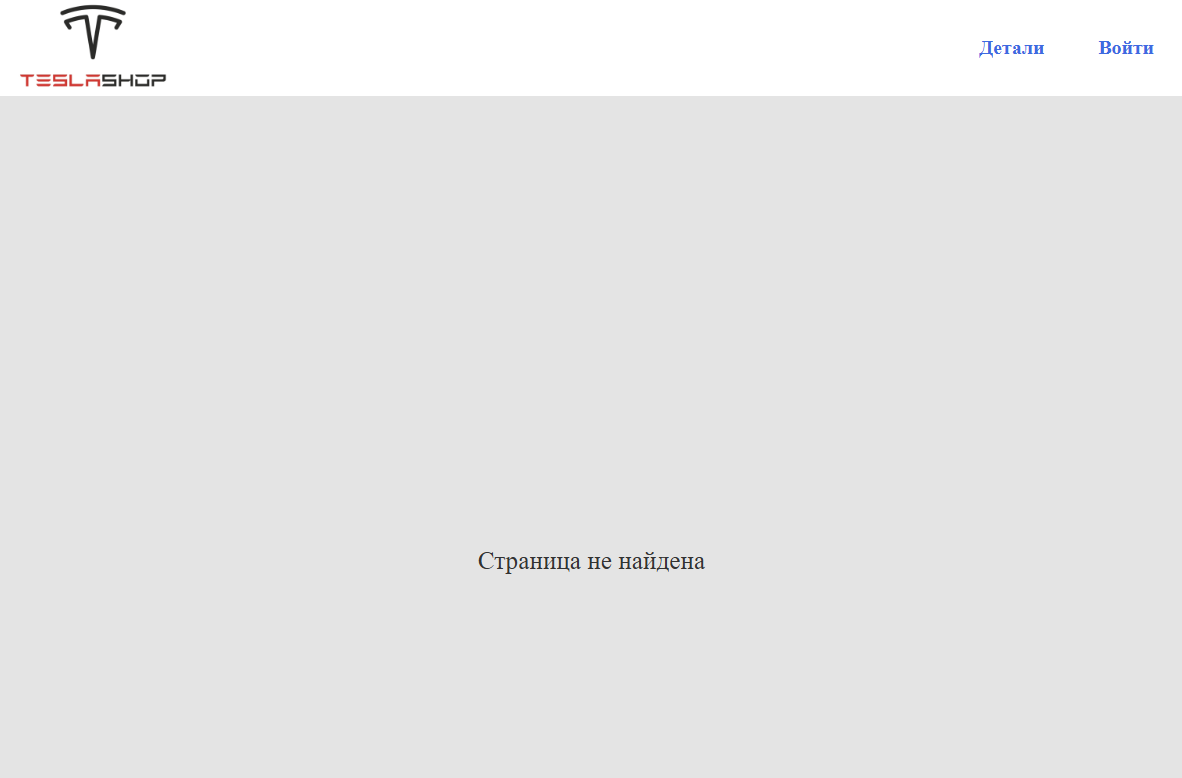


Рисунок 26 – Страница ошибки 404

Десктопное приложение, разработанное с использованием фреймворка Tauri, представлено на рисунке 27.

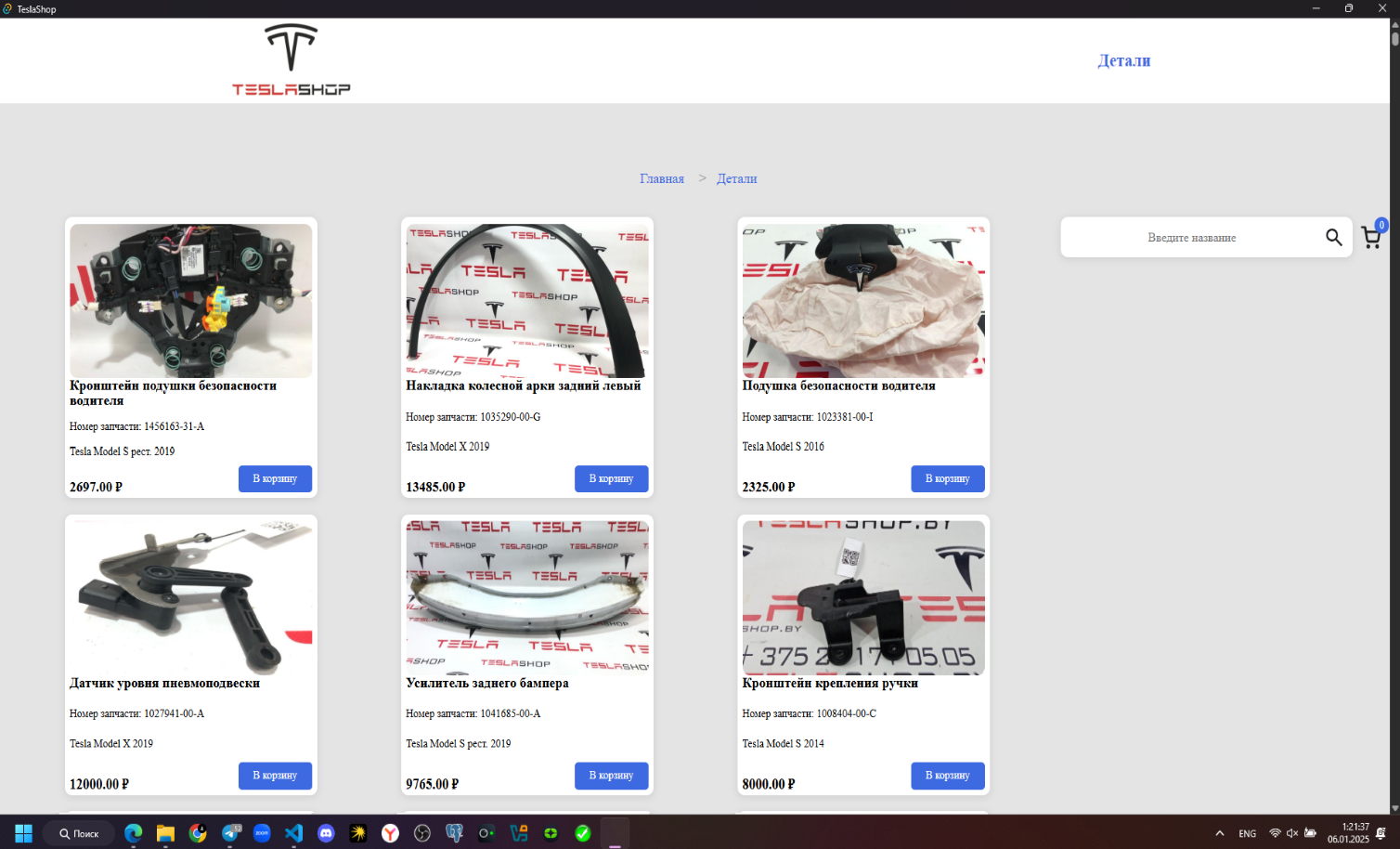


Рисунок 27 – Десктопное приложение на Tauri

Прогрессивное веб-приложение (PWA), разработанное для обеспечения удобного взаимодействия пользователей с сервисом, представлено на рисунке 28.

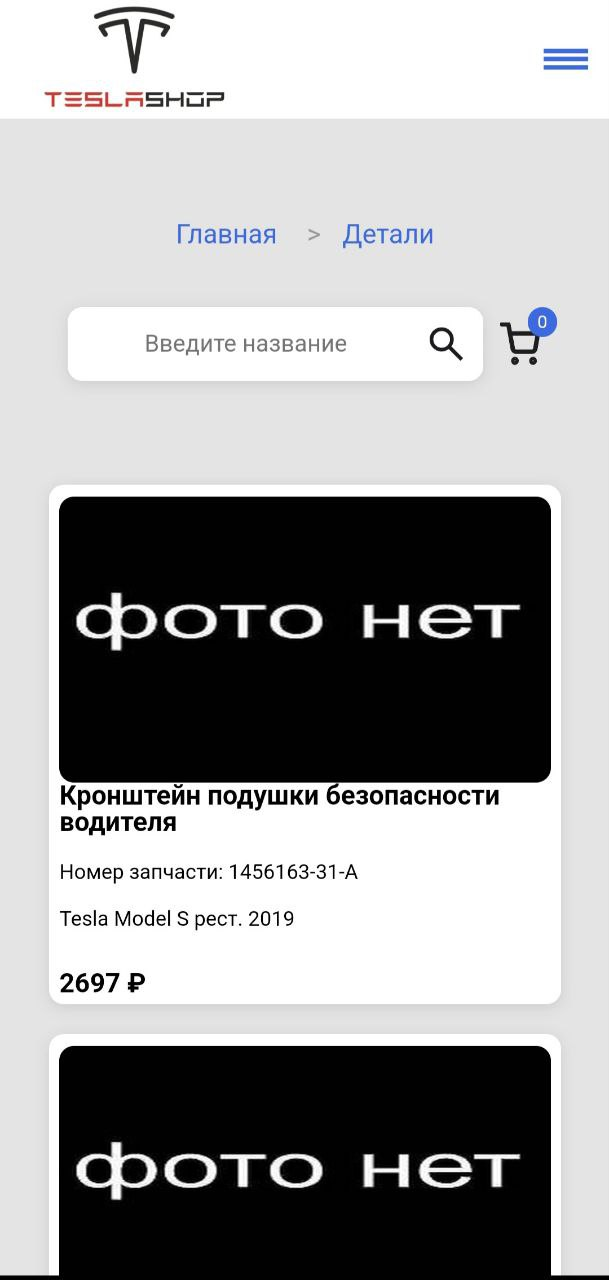


Рисунок 28 – Прогрессивное веб-приложение (PWA)

Также реализована пагинация при большом количестве данных (рисунок 29). Использован индекс по совпадению.

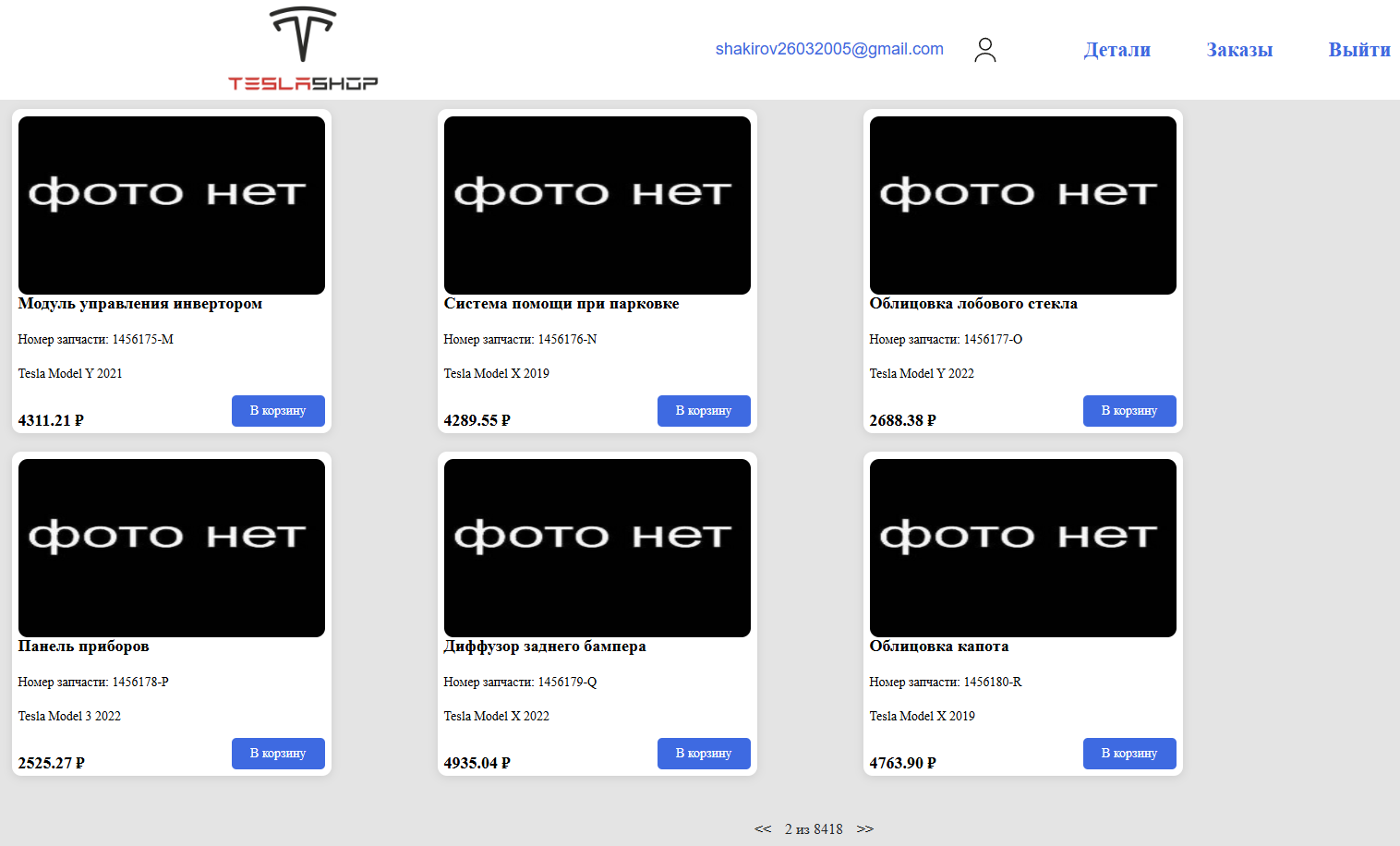


Рисунок 29 – Пагинация

ЗАКЛЮЧЕНИЕ

В ходе работы были достигнуты следующие результаты:

1. Разработан дизайн приложения в Figma на основе teslashop.ru. После ознакомления с разработкой бэкенда с использованием фреймворка Django создан MVP.
2. Спроектирована и создана база данных PostgreSQL, а затем подключена к бэкенду.
3. Создан веб-сервис со всей итоговой бизнес-логикой в бэкенде системы для использования его в SPA.
4. Добавлена авторизация с использованием Redis для хранения сессий, в веб-сервис внедрен Swagger.
5. Разработан базовый интерфейс приложения для гостя на React.
6. Внедрен менеджер состояний Redux Toolkit для хранения значений фильтров, добавлена адаптивность и PWA.
7. Завершена разработка интерфейса пользователя в React, для обращений к методам веб-сервиса использован Axios.
8. Реализован React интерфейс издателя, внедрен Real-time web.
9. Разработано десктопное приложение Tauri.
10. Приложение развернуто при помощи сервиса GitHub Pages и доступно по ссылке: [https://[t1mfsm.github.io/Tesla-Shop](https://t1mfsm.github.io/Tesla-Shop/)](https://victobes.github.io/book-office-web-app-frontend/)
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформлен git-репозиторий на сервисе GitHub, содержащий исходный код проекта: [https://[github.com/t1mfsm/Tesla-Shop](https://github.com/t1mfsm/Tesla-Shop)](https://github.com/victobes/book-office-web-app-backend).
13. Реализована пагинация с фильтрацией по индексу в React-приложении.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Exponential Adoption of Battery Electic Cars [Электронный ресурс]. URL: https://arxiv.org/abs/2306.16152 (дата обращения: 01.11.2024).
2. Разработка нового продукта для модельных потребительских товаров длительного пользования [Электронный ресурс]. URL: https://pubsonline.informs.org/doi/10.1287/mnsc.15.5.215 (дата обращения: 26.11.2024).
3. Гибридная структура: исследование экономики технологических изменений [Электронный ресурс]. URL: https://www.jstor.org/stable/1905380?origin=crossref (дата обращения: 26.11.2024).
4. Легковые автомобили, в разбивке по типу двигательной энергии [Электронный ресурс]. URL: https://ec.europa.eu/eurostat/databrowser/view/ROAD\_EQS\_CARPDA\_\_custom\_2967363/default/table?lang=en (дата обращения: 26.11.2024).
5. Будущее автомобилей за электричеством – но как скоро наступит это будущее? [Электронный ресурс]. URL: https://pv-magazine-usa.com/2020/05/19/the-future-of-cars-is-electric-but-how-soon-is-this-future/ (дата обращения: 26.11.2024).
6. Глобальный прогноз развития электромобилей на 2023 год [Электронный ресурс]. URL: https://www.iea.org/reports/global-ev-outlook-2023 (дата обращения: 26.11.2024).
7. Обновленная информация о целях правительства по поэтапному прекращению продаж новых легковых автомобилей с двигателями внутреннего сгорания [Электронный ресурс] URL: https://theicct.org/publication/update-on-government-targets-for-phasing-out-new-sales-of-internal-combustion-engine-passenger-cars (дата обращения: 26.11.2024).
8. Документация по DRF [Электронный ресурс] // Django-rest-framework. URL: https://www.django-rest-framework.org/topics/documenting-your-api/ (дата обращения: 20.10.2024).
9. Документация по Redis [Электронный ресурс] // Netlify. URL: https://master--redis-doc.netlify.app/docs/ (дата обращения: 10.10.2024).
10. Документация Minio [Электронный ресурс] // Min. URL: https://min.io/docs/minio/kubernetes/upstream/index.html (дата обращения: 09.09.2024).
11. Документация PostgreSQL [Электронный ресурс] // Postgresql. URL: https://www.postgresql.org/docs/ (дата обращения: 15.09.2024).
12. Документация по Python [Электронный ресурс] // Python. URL: https://docs.python.org/3/index.html/ (дата обращения: 20.10.2024).
13. Рецепты Django ORM [Электронный ресурс] // Django. URL: https://django.fun/docs/django-orm-cookbook/2.0/ (дата обращения: 10.10.2024).
14. Документация по Tauri [Электронный ресурс] // Tauri. URL: https://v2.tauri.app/develop/ (дата обращения: 20.11.2024).

ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.И. Канев  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Поставки комплектующих для производства электроавтомобилей**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

16

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Шакиров Тимур Маратович |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва - 2024

1. Введение

Система «Поставки комплектующих для производства электроавтомобилей» представляет собой веб-сервис, включающий веб-приложение и нативное приложение для управления хранимыми комплектующими и обработкой заявок на их доставку.

1. Назначение разработки

Данная система предназначена для хранения информации о комплектующих и управления их доступностью на складе. Сотрудники склада обращаются к сервису с целью добавления комплектующих в заявку. Также сотрудник склада может потом смотреть свои созданные заявки. Руководитель склада может отклонять либо завершать их сформированные заявки. Также руководитель может редактировать набор комплектующих, кроме того, он может просматривать отправки абсолютно всех сотрудников.

1. Стадии и этапы разработки
   1. Создание MVP и базового дизайна на основе teslashop.ru
   2. Реализовать хранение данных в БД (Postgres)
   3. Создать веб-сервис на бэкенде (DRF)
   4. Реализовать авторизацию и хранение сессий в Redis
   5. Разработать SPA на React для гостя
   6. Внедрить адаптивность, разработка Tauri приложения
   7. Интерфейс сотрудника на React с менеджером состояний Redux Toolkit
   8. Добавить в приложение React интерфейс руководителя
   9. Реализовать нативное приложение
   10. Развернуть веб-приложение в GitHub Pages
   11. Подготовка всей документации (РПЗ, ТЗ и набор диаграмм)
   12. Подготовка репозитория на GitHub
2. Требования к функциональным характеристикам
   1. Список HTTP методов
      1. GET Получение всех комплектующих
      2. GET Получение одной комплектующей
      3. POST Добавление комплектующей
      4. PUT Изменение комплектующей
      5. DELETE Удаление комплектующей
      6. POST Добавление комплектующей в заявку-черновик
      7. POST Обновление изображения комплектующей
      8. GET Получение всех заявок
      9. GET Получение одной заявки
      10. PUT Изменение дополнительных полей заявки
      11. PUT Сформировать заявку создателем
      12. PUT Завершить/отклонить заявку модератором
      13. DELETE Удаление заявки
      14. DELETE Удаление комплектующей из заявки
      15. PUT Изменение количества комплектующих в заявке
      16. POST Регистрация пользователя
      17. POST Аутентификация
      18. POST Деавторизация
      19. PUT Личный кабинет
   2. Меню
      1. Главная – перенаправляет на страницу 4.5
      2. Список комплектующих – перенаправляет на страницу 4.6 вызывается метод 4.1.1
      3. Список заявок – перенаправляет на страницу 4.9 вызывается метод 4.1.8
      4. Редактирование комплектующих – перенаправляет на страницу 4.11, вызывается метод 4.1.4, доступно только руководителям
      5. Зарегистрироваться – перенаправляет на страницу 4.3, вызывается метод 4.1.16, появляется только для гостей
      6. Личный кабинет – перенаправляет на страницу 4.10, вызывается метод 4.1.19, только аутентифицированным пользователям
      7. Войти – перенаправляет на страницу 4.4, вызывается метод 4.1.17, появляется только для гостей
      8. Выйти – перенаправляет на страницу 4.6, вызывается метод 4.1.18, доступно только аутентифицированным пользователям
   3. Регистрация
      1. Доступно только гостям
      2. Отображает форму регистрации
         1. Поле имени пользователя
         2. Поле пароля
         3. Поле имени
         4. Поле фамилии
      3. Действия
         1. Регистрация пользователя
         2. Вернуться к аутентификации – перенаправляет на страницу 4.4, вызывается метод 4.1.17.
   4. Аутентификация
      1. Доступно только гостям
      2. Отображает форму аутентификации
         1. Поле имени пользователя
         2. Поле пароля
      3. Действия
         1. Войти
         2. Регистрация – перенаправляет на страницу 4.3, вызывается метод 4.1.16.
   5. Личный кабинет
      1. Доступно аутентифицированному сотруднику
      2. Действия
         1. Изменить данные пользователя
   6. Главная
      1. Доступна всем
      2. Отображается статическая информация
         1. Назначение сервиса
         2. Контакты для связи
   7. Список комплектующих
      1. Доступна всем
      2. Отображаются элементы карточек с комплектующими
         1. Название комплектующей
         2. Краткое описание
      3. Действия
         1. Поиск – перенаправляет на страницу 4.7, с фильтрующем параметром, вызывается метод 4.1.1.
         2. Название комплектующей – перенаправляет на страницу 4.8, вызывается метод 4.1.2.
         3. Добавить – добавляет комплектующую в заявку-черновик, только аутентифицированные сотрудники, вызывается метод 4.1.6.
   8. Одна комплектующая
      1. Доступна всем
      2. Отображается подробная информация выбранной комплектующей.
   9. Одна заявка
      1. Доступно только аутентифицированным сотрудникам.
      2. Отображается текущая заявка-черновик сотрудника.
         1. Список выбранных комплектующих.
         2. Поле названия завода.
         3. Поле планируемой даты отправки.
      3. Действия, доступны только в случае, если статус «черновик»
         1. Сформировать – формирует текущую заявку-черновик, вызывается метод 4.1.11.
         2. Удалить – удаляет заявку-черновик, вызывается метод 4.1.13.
         3. Заполнить – вносит данные заявки
   10. Список отправок
       1. Доступно аутентифицированному сотруднику
       2. Отображается список заявок
          1. Только заявки, созданные данным сотрудником, если он не руководитель
          2. Все сохраненные заявки в противном случае
       3. Действия
          1. Фильтрация – фильтрует заявки по дате формирования или статусу
          2. Завершить – завершит заявку, доступно только руководителю, вызывается метод 4.1.12.
          3. Отклонить – отклоняет заявку, доступно только руководителю, вызывается метод 4.1.12.
          4. Посмотреть подробную информацию о заявке – перенаправляет на страницу 4.9, вызывается метод 4.1.9.
   11. Список комплектующих таблицей
       1. Доступно только руководителю
       2. Отображаются все существующие в системе комплектующие
       3. Действия
          1. Удалить – удаляет комплектующую, вызывается метод 4.1.5.
          2. Редактирование/создание – переход на страницу 4.12, вызывается метод 4.1.4.
   12. Редактирование/создание комплектующей
       1. Доступно только руководителю
       2. Отображается информация об изменяемой/добавляемой комплектующей
          1. Название
          2. Номер запчасти
          3. Цена
          4. Информация о модели
          5. Год
          6. Модель
          7. Артикул
          8. Бренд
          9. Описание
       3. Действия
          1. Убрать комплектующую – удаляет комплектующую из заявки, вызывается метод 4.1.14.
          2. Сохранить – добавляет новую комплектующую, вызывается метод 4.1.3.
          3. Изображение – добавляет/изменяет изображение комплектующей, вызывается метод 4.1.17.
   13. 404
       1. Доступно всем
       2. Отображается в случае отсутствия ресурса
   14. 403
       1. Доступно всем
       2. Отображается в случае запрета на использование ресурса
3. Требования к составу и параметрам технических средств
   1. Серверная часть
      1. Процессор минимум 2-ядерный с частотой от 2 ГГц
      2. Оперативная память от 4 Гб
      3. Свободное пространство на диске от 2 Гб
   2. Клиентская часть
      1. Процессор с частотой от 1 ГГц
      2. Оперативная память от 1 Гб
      3. Свободное пространство на диске от 1 Гб
4. Требования к информационной и программной совместимости
   1. Серверная часть
      1. ОС Linux (6.4.12) / Windows 10/11
      2. Redis (7.2)
      3. Minio (RELEASE 2022-10-15T19-57-03Z)
      4. PostgreSQL (16)
      5. Node JS
      6. Python 3.12
      7. Django 5.1
   2. Клиентская часть
      1. Браузер (Safari 16.5.2, Firefox 121.0, Chrome 119.0.6045, Yandex 24.6.3.729, Opera 105.0.4970.16)
      2. ОС: Windows/Linux/MacOS

ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ

Таблица 1 – HTTP методы разрабатываемого веб-сервиса

| **№** | **Метод** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| --- | --- | --- | --- | --- | --- |
| 4.1.1 | GET | api/details/ | Возвращает все детали. Доступно всем | name=string (опционально) | [  {  id: number;      name: string;      part\_number: string;      price: string;      model\_info: string;      year: string;      model: string;      article\_number: string;      brand: string;      note: string;      image: string;  }  draft\_carorder\_id; int,  detail\_count: int,  ] |
| 4.1.2 | GET | api/details/{pk}/ | Возвращает деталь по ID. Доступно всем | pk: int | {  id: number;      name: string;      part\_number: string;      price: string;      model\_info: string;      year: string;      model: string;      article\_number: string;      brand: string;      note: string;      image: string;  } |
| 4.1.3 | POST | /details/ | Создает новую деталь. Доступно модератору | {      name: string;      part\_number: string;      price: string;      model\_info: string;      year: string;      model: string;      article\_number: string;      brand: string;      note: string;      image: string;  } | {  id: number;      name: string;      part\_number: string;      price: string;      model\_info: string;      year: string;      model: string;      article\_number: string;      brand: string;      note: string;      image: string;  } |
| 4.1.4 | PUT | /details/{pk}/ | Обновляет данные детали по ID. Доступно модератору | pk: int,  {  id: number;      name: string;      part\_number: string;      price: string;      model\_info: string;      year: string;      model: string;      article\_number: string;      brand: string;      note: string;      image: string;  } | {  id: number;      name: string;      part\_number: string;      price: string;      model\_info: string;      year: string;      model: string;      article\_number: string;      brand: string;      note: string;      image: string;  } |
| 4.1.5 | DELETE | /details/{pk}/ | Удаляет деталь по ID. Доступно модератору | pk: int | [] |
| 4.1.6 | POST | /details/{pk}/draft/ | Добавляет деталь в заявку. Доступно авторизованным пользователям | pk: int | {} |
| 4.1.7 | POST | /details/{pk}/image/ | Добавляет/изменяет картинку детали, доступно модератору | pk: int, { image: file} | {  id: number;      name: string;      part\_number: string;      price: string;      model\_info: string;      year: string;      model: string;      article\_number: string;      brand: string;      note: string;      image: string;  } |
| 4.1.8 | GET | /car\_orders/ | Получает зявки, доступно любому аутентифицированному пользователю | pk: int (опционально) | [  {  ID: int,  order\_number: int  status: string,  creation\_date: datetime,  order\_date: datetime,  ship\_date: datetime,  factory: string,  total\_cost: int,  creator: string,  moderator: string,  }  ] |
| 4.1.9 | GET | /car\_orders/{pk} | Получает заявку, любому аутентифицированному пользователю | pk: int | {  ID: int,  order\_number: int  status: string,  creation\_date: datetime,  order\_date: datetime,  ship\_date: datetime,  factory: string,  total\_cost: int,  creator: string,  moderator: string,  details  } |
| 4.1.10 | PUT | /car\_orders/{pk} | Изменяет поля заявки, доступно аутентифицированному посетителю | pk: int,  {  factory: string,  order\_date: date,  } | {  ID: int,  order\_number: int  status: string,  creation\_date: datetime,  order\_date: datetime,  ship\_date: datetime,  factory: string,  total\_cost: int,  creator: string,  moderator: string,  details  } |
| 4.1.11 | PUT | /car\_orders/{pk}/form | Сохраняет заявку, доступно аутентифицированному посетителю | pk: int | {  ID: int,  order\_number: int  status: string,  creation\_date: datetime,  order\_date: datetime,  ship\_date: datetime,  factory: string,  total\_cost: int,  creator: string,  moderator: string,  details  } |
| 4.1.12 | PUT | /car\_orders/{pk}/complete | Модерирует заявку, доступно менеджеру | pk: int,  {  status: string  } | {  ID: int,  order\_number: int  status: string,  creation\_date: datetime,  order\_date: datetime,  ship\_date: datetime,  factory: string,  total\_cost: int,  creator: string,  moderator: string,  details  } |
| 4.1.13 | DELETE | /car\_orders/{pk} | Удаляет заявку, доступно любому аутентифицированному посетителю | pk: int | {  ID: int,  order\_number: int  status: string,  creation\_date: datetime,  order\_date: datetime,  ship\_date: datetime,  factory: string,  total\_cost: int,  creator: string,  moderator: string,  details  } |
| 4.1.14 | PUT | /car\_orders/{pk\_order}/dishes/ {pk\_detail} | Меняет поля М-М, доступно менеджеру | pk: int,  {  quantity: int  } | {  id: int,  quantity: int,  part: int  } |
| 4.1.15 | DELETE | /car\_orders/{pk\_order}/details/ {pk\_detail} | Удаляет деталь из заявки, доступно аутентифицированному посетителю | pk: int | {  id: int,  quantity: int,  part: int  } |
| 4.1.16 | POST | /user/auth/ | Регистрация нового пользователя. Доступно всем | {  email: string,  password: string,  first\_name: string,  last\_name: string,  username: string  } |  |
| 4.1.17 | POST | /login/ | Аутентификация, доступна всем | {  username: string, password: string  } | {  token: string  } |
| 4.1.18 | POST | /logout/ | Деавторизация, доступна любому аутентифицированному пользователю | {  token: string  } |  |
| 4.1.19 | PUT | /users/profile/ | Личный кабинет, доступно любому аутентифицированному пользователю | pk: int,  {  email: string,  password: string,  first\_name: string,  last\_name: string,  username: string  } | {  user\_id=int,  email: string,  password: string,  first\_name: string,  last\_name: string,  username: string  } |