

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»  
Факультет среднего профессионального образования

Направление подготовки (специальность) 09.02.07 Информационные системы и программирование

**О Т Ч Е Т**

**об учебной практике по профессиональному модулю  
ПМ.11 «Разработка, администрирование и защита баз данных»**

Тема задания: Разработка и реализация базы данных по предметной области

Обучающийся Кузьмина Е. А. Группа У2331  
(Фамилия И.О.) (номер группы)

Руководитель практики: Ефимова Т.Н., преподаватель факультета СПО  
Говоров А.И., преподаватель факультета СПО

Ответственный за практику от университета: Королев В.В.зам. директора  
факультета СПО

Практика пройдена с оценкой \_\_\_\_\_

Подписи членов комиссии \_\_\_\_\_ (\_\_\_\_\_) (подпись)

\_\_\_\_\_ (А.И.Говоров)  
(подпись)

\_\_\_\_\_ (Т.Н.Ефимова)  
(подпись)

Дата \_\_\_\_\_

Санкт-Петербург  
2020

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**  
**Факультет среднего профессионального образования**

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**  
**на учебную практику**  
**по ПМ.11 «Разработка, администрирование и защита баз данных»**

Студент Кузьмина Е. А. (Фамилия И. О.) Группа № У2331

Руководитель Ефимова Т.Н., преподаватель факультета СПО  
Говоров А.И., преподаватель факультета СПО

Тема задания: Проектирование и реализация базы данных.  
\_\_\_\_\_  
\_\_\_\_\_

Сроки прохождения практики: 02.02.2020 -02.07.2020

Место прохождения практики: Факультет СПО

**1. Виды работ и требования к их выполнению:**

Учебная практика проводится распределенно (понедельно в течение семестра) на базе факультета СПО в лаборатории разработки баз данных. В ходе прохождения практики выполняются следующие виды работ:

- I. Вводный инструктаж по технике безопасности и общим целям, и задачам практики.
- II. Анализ поставленной задачи
- III. Выполнение индивидуального задания: проектирование БД, разработка прототипа веб-приложения.
- IV. Формирование отчета по учебной практике.

**2. Виды отчетных материалов и требования к их оформлению:**

По результатам прохождения практики составляется отчет, в котором представляются индивидуальное задание, модель базы данных, перечень использованных технологий, программных средств, использованных паттернов (шаблонов) проектирования программ, программный код, описание результатов работы программы. Оформление отчета должно соответствовать Рекомендациям по оформлению технических документов факультета СПО Университета ИТМО.

### 3. ПЛАН-ГРАФИК

№ эта па	Наименование этапа	Срок завершения этапа	Виды работ	Форма отчетности
1	2	3	4	5
1.	Вводный инструктаж	02.02.2020 – 09.02.2020	Ознакомление с инструкцией по технике безопасности. Ознакомление с целями и задачами производственной практики	Журнал по технике безопасности
2.	Постановка задачи	09.02.2020 – 09.03.2020	Анализ индивидуального задания. Обследование предметной области согласно индивидуальной теме учебной практики.	Отчет по практике: индивидуальное задание Дневник практики
3.	Моделирование базы данных и реализация	10.03.2020 – 31.03.2020	Описание предметной области. Создание диаграммы классов Создание таблиц Заполнение таблиц данными (команды)	Отчет по практике: индивидуальное задание Дневник практики
4.	Реализация модели данных средствами Django ORM	01.04.2020 – 07.04.2020	Создание модели Django в соответствии с моделью данных и настройка связи между таблицами	Отчет по практике: индивидуальное задание Дневник практики
5.		82.04.2020 – 21.04.2020	Реализация элементов инфраструктуры Django, в соответствии с архитектурным паттерном Model-View-Template или сокращенно MVT. Реализация интерфейсов к системе средствами Django Templates или сторонними средствами.	Отчет по практике: индивидуальное задание Дневник практики
6.	Подготовка отчетных материалов	17.06.2020 – 23.06.2020	Формирование отчета о практике	Отчет по практике: индивидуальное задание Дневник практики
7.	Защита результатов практики	24.06.2020 – 02.07.2020	Защита результатов практики в форме устного собеседования и представления результатов практики	

Задание утверждено председателем выпускающей комиссии факультета СПО  
 Председатель выпускающей комиссии факультета СПО \_\_\_\_\_ Королев В.В.  
 « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г

Дата выдачи задания: \_\_\_\_\_

Руководитель от факультета \_\_\_\_\_  
 (подпись руководителя)

Задание принял к  
 исполнению \_\_\_\_\_  
 (подпись студента)

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**  
**Факультет среднего профессионального образования**

**ДНЕВНИК ПРАКТИКИ**

за период с 02.02.2020 по 02.07.2020

Студент	<u>Кузьмина Е. А.</u> (Фамилия, И.О.)
Факультет	<u>СПО</u>
Группа	<u>Y2331</u>
Направление (специальность)	<u>09.02.07 Информационные системы и программирование</u>
Место прохождения практики	<u>Факультет СПО</u>
Руководитель практики	<u>Ефимова Т.Н., факультет СПО, преподаватель Говоров А.И., факультет СПО, преподаватель</u>
Ответственный за проведение практики от университета	<u>Королев В.В., зам. директора факультета</u>

Индивидуальное задание  
выполнено полностью

\_\_\_\_\_  
(подпись ответственного  
за проведение практики от  
университета)

\_\_\_\_\_  
(дата)

Санкт-Петербург

2020

Период	Краткое содержание работы	Отметка о выполнении
02.02.2020 – 09.02.2020	Вводный инструктаж. Ознакомление с инструкцией по технике безопасности. Ознакомление с целями и задачами практики.	
09.02.2020 – 09.03.2020	Анализ индивидуального задания. Обследование предметной области согласно индивидуальной теме учебной практики.	
10.03.2020 – 31.03.2020	Описание предметной области. Создание диаграммы классов. Создание таблиц.	
01.04.2020 – 07.04.2020	Создание модели Django в соответствии с моделью данных и настройка связи между таблицами	
08.04.2020 – 21.04.2020	Реализация элементов инфраструктуры Django, в соответствии с архитектурным паттерном Model-View-Template или сокращенно MVT. Реализация интерфейсов к системе средствами Django Templates или сторонними средствами.	
17.06.2020 – 23.06.2020	Подготовка отчетных материалов по результатам практики.	
24.06.2020 – 02.07.2020	Защита результатов практики.	

---

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**  
**Факультет среднего профессионального образования**

**О Т З Ы В**

**руководителя учебной практики**  
**по профессиональному модулю**  
**ПМ.11 «Разработка, администрирование и защита баз данных»**

Обучающийся      Кузьмина Е. А.

(Фамилия, И.О.)

Факультет      СПО

Группа      У2331

Направление (специальность)      09.02.07 Информационные системы и программирование

Место прохождения практики      Факультет СПО

Тема индивидуального задания

**Оценка достигнутых результатов**

№ п/п	Планируемые результаты обучения (показатели)	Оценка			
		5	4	3	2
1.	Корректность определения структуры базы данных				
2.	Качество реализации компонентов описания модели данных средствами Django ORM				
3.	Качество реализации контроллеров Django				
4.	Качество реализации интерфейсов к системе средствами Django Templates или сторонними средствами.				
Итоговая оценка					

**Отмеченные достоинства:**

.....

**Отмеченные недостатки:**

.....

**Заключение:** Считаю, все задачи, поставленные на практику, выполнены и по результатам практики студент(ка) ..... заслуживает оценки «.....».

Руководитель практики \_\_\_\_\_ .....

(подпись)

(ФИО)

« \_\_\_\_ » \_\_\_\_\_ 2020г.

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**АТТЕСТАЦИОННЫЙ ЛИСТ**  
**Характеристика профессиональной деятельности студента во время**  
**прохождения учебной практики**

Студент Кузьмина Е. А.

Группа Y2331

Специальность 09.02.07 Информационные системы и программирование

Место проведения практики Факультет СПО

Сроки прохождения практики 20.01.2020 – 02.07.2020

Наименование профессиональных модулей (видов деятельности)

ПМ.11 «Разработка, администрирование и защита баз данных»

Виды выполняемых работ:

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Отметка о выполнении <sup>1</sup>
<b>ПМ.11 Разработка, администрирование и защита баз данных</b>		
ПК 11.1. Осуществлять сбор, обработку и анализ информации для проектирования баз данных	— осуществление корректного сбора, обработки и анализа информации для проектирования баз данных	
	— обоснование выявления объектов проектируемой БД и установки отношений между ними на основе анализа предметной области;	
ПК 11.2. Проектировать базу данных на основе анализа предметной области.	— использование CASE-средств автоматизированного проектирования при моделировании базы данных при построении концептуальной, даталогической и физической моделей БД;	
	— соответствие проекта структурной и манипуляционной частей БД заданным критериям функциональности.	

<sup>1</sup> Указывается «+» или «-». Считается, что программа практики выполнена, если студентом выполнено не менее 70% перечисленных видов работ.



Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Отметка о выполнении <sup>1</sup>
ПК 11.3. Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.	– использование CASE-средств автоматизированного проектирования при моделировании базы данных при построении концептуальной, даталогической и физической моделей БД;	
	– обоснование выбранных методов защиты объектов базы данных в соответствии с требованиями задачи.	
ПК 11.4. Реализовывать базу данных в конкретной системе управления базами данных.	– соответствие реализации структурной части БД средствами СУБД даталогической и физической моделям данных;	
	– соответствие реализации манипуляционной части БД средствами СУБД заданным критериям;	
	– соответствие реализации приложения БД заданным критериям функциональности;	
ПК 11.5. Администрировать базы данных.	– обоснованность выбора архитектуры клиент-серверного взаимодействия в соответствии с технологией разработки базы данных;	
	– соответствие заданным требованиям обеспечения целостности данных и контроля доступа к данным;	
	– соответствие заданным требованиям управления привилегиями пользователей базы данных программными средствами;	
	– соответствие конфигурирования сетевых устройств требованиям обеспечения доступа к данным.	
ПК 11.6. Защищать информацию в базе данных с использованием технологии защиты информации.	– соответствие конфигурирования сетевых устройств требованиям защиты данных при передаче данных по сети	
	– соответствие заданным требованиям программных средств защиты информации в базе данных средствами СУБД	
	– соответствие заданным требованиям управления привилегиями пользователей базы данных программными средствами;	

Руководитель практики от факультета СПО: \_\_\_\_\_

Дата: \_\_\_\_\_

# СОДЕРЖАНИЕ

Введение .....	11
1 Индивидуальное задание .....	12
1.1 Описание предметной области .....	12
1.2 Функциональные требования к системе .....	12
2 Проектирование базы данных .....	14
3 Технологии и программные средства .....	15
4 Программная реализация .....	17
Заключение .....	32
Список источников .....	33
ПРИЛОЖЕНИЕ А .....	34
ПРИЛОЖЕНИЕ Б .....	41

## ВВЕДЕНИЕ

Целью учебной практики по профессиональному модулю ПМ.11 «Разработка, администрирование и защита баз данных» является углубление знаний и практических умений и получение начального практического опыта по основным видам деятельности «Разработка, администрирование и защита баз данных» и овладение соответствующими общими и профессиональными компетенциями: ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9, ОК 10, ОК 11, ПК 11.1., ПК 11.2., ПК 11.3, ПК 11.4., ПК 11.5., ПК 11.6. (см. рабочая программа и фонд оценочных средств по производственной практике).

Учебная практика проводится на базе факультета СПО Университета ИТМО.

Результатом практики является разработка прототипа веб-приложения по заданной предметной области, использующего реляционную базу данных.

Задачи по разработке системы:

- разработать модель базы данных по заданной предметной области;
- определить будущие интерфейсы веб-приложения
- разработать программно – аппаратную часть веб – приложения;
- разработать пользовательский интерфейс и функционал веб – приложения.

# 1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

## 1.1 Описание предметной области

«Парфюмерный базар». Постоянно работающий парфюмерный базар характеризуется участием в ней оптовых фирм поставщиков и оптовых фирм покупателей, при этом все сделки заключаются через специальных маклеров, имеющих доступ ко всем товарам.

Управление базаром должно иметь сведения:

- о маклерах: имя, фамилия маклера, адрес, дата рождения;
- о товаре: название товара, вид, срок годности, пол, производитель;
- о производителе: название фирмы производителя, страна производства, юридический адрес;
- о фирмах: название фирмы, страна, юридический адрес, тип фирмы (продавец, покупатель);
- о заключенных сделках: дата сделки, маклер, оптовая фирма покупатель, оптовая фирма продавец;
- о заказе сделки: сделка, товар, количество проданных единиц, цена единицы товара от продавца, цена единицы товара от маклера.

## 1.2 Функциональные требования к системе

Требования:

- вход в веб – приложение по заданному логину и паролю (в случае неправильного логина или пароля – соответствующее сообщение);
- меню с возможностью выбора интерфейса;
- интерфейс маклера – просмотр информации обо всех маклерах;
- интерфейс фирмы – просмотр информации обо всех фирмах;
- интерфейс производителя – просмотр информации обо всех производителях;
- интерфейс заказа сделки – просмотр информации обо всех заказах;

- интерфейс сделки – CRUD (создание, просмотр, обновление удаление);
- интерфейс товара – CRUD (создание, просмотр, обновление удаление).

## 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Используемая методология – EER (расширенная модель “сущность – связь”). Используемая нотация – Crow’s Foot.

Сущности, выделенные при анализе предметной области:

- маклер;
- фирма;
- сделка;
- заказ сделки;
- товар;
- производитель.

Инфологическая модель базы данных находится в приложении А.

Логическая модель базы данных соответствует первой нормальной форме.

Логическая модель базы данных в виде описания таблиц представлена в приложении А.

### 3 ТЕХНОЛОГИИ И ПРОГРАММНЫЕ СРЕДСТВА

Использованные технологии:

- СУБД SQLite;
- интегрированная среда разработки PyCharm;
- фреймворк Django;
- библиотека Django Rest Framework;
- программная платформа Node.js;
- JavaScript – фреймворк Vue.js;
- программное обеспечение Docker;
- инструмент Docker Compose.

Программное средство, использованное при проектировании базы данных – MySQL Workbench.

Программные средства реализации системы:

- СУБД SQLite (версия – 3.32.3);
- интегрированная среда разработки PyCharm (версия – 2020.1.2);
- фреймворк Django (версия – 3.0.7);
- библиотека Django Rest Framework (версия – 3.11.0);
- программная платформа Node.js (версия – 8.10.0);
- JavaScript – фреймворк Vue.js (версия – 2.5.2);
- библиотека компонентов пользовательского интерфейса Material Design Muse UI (версия – 3.0.2);
- пакет модулей JavaScript Webpack (версия 3.6.0);
- JavaScript-транскомпилятор Babel (версия – 7.10.3);
- библиотека Material Design Icons (версия – 3.0.1);
- библиотека django-cors-headers (версия – 3.3.0);
- библиотека django-rest-swagger (версия – 2.2.0);
- библиотека django-summernote (версия – 0.8.11.6);
- библиотека django-templated-mail (версия – 1.1.1);

- библиотека `django-rest-framework-jsonapi` (версия – 3.1.0);
- библиотека `django-rest-framework-simplejwt` (версия – 4.4.0);
- библиотека `Djoser` (версия – 2.0.3);
- программное обеспечение `Docker` (версия – 19.03.11);
- инструмент `Docker Compose` (версия – 3).

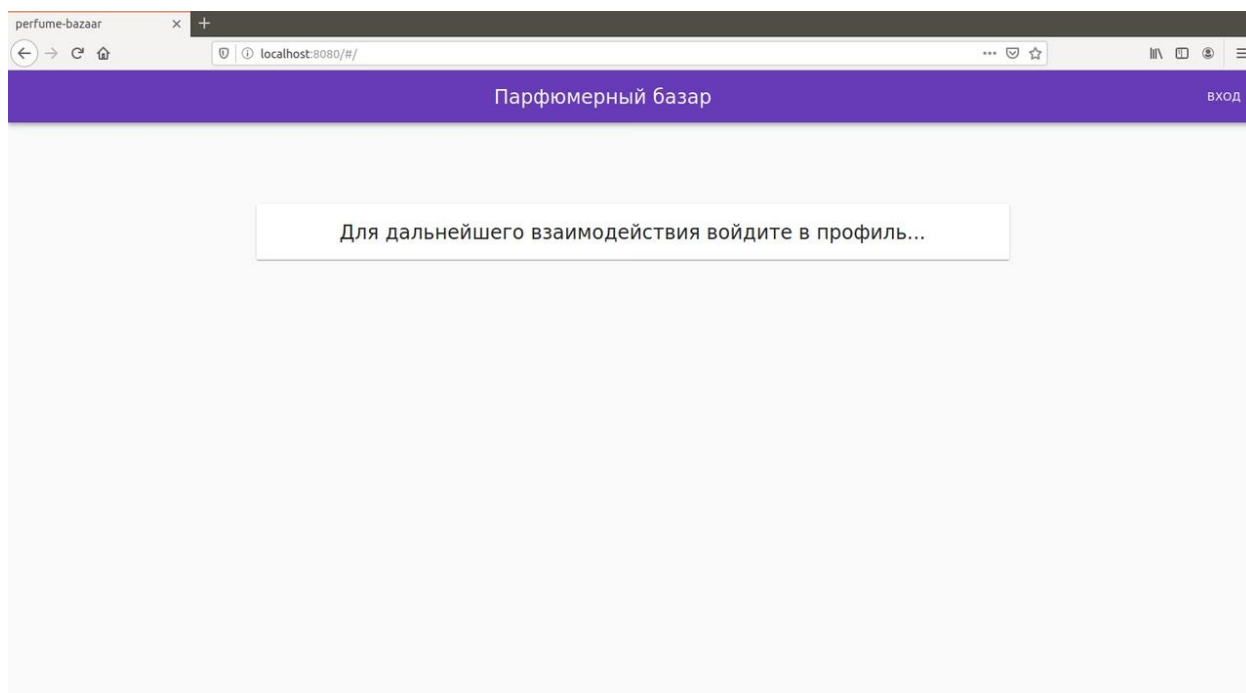
В данном веб – приложении паттерну проектирования MVT соответствуют:

- URL dispatcher – `urls` в приложении Django;
- Model – `models` в приложении Django (SQLite база данных);
- View – `views` в приложении Django с использованием DRF;
- Template – `Vue.js` приложение.



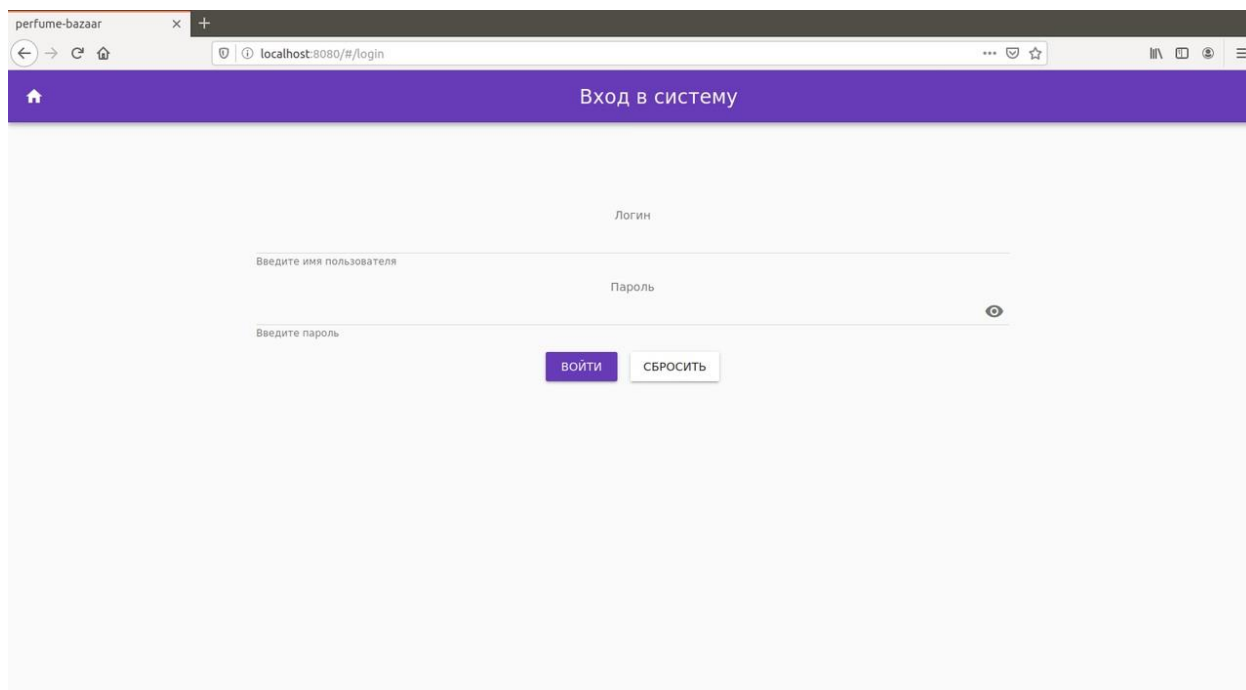
## 4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Главное окно веб – приложения до входа в систему (рисунок 1).



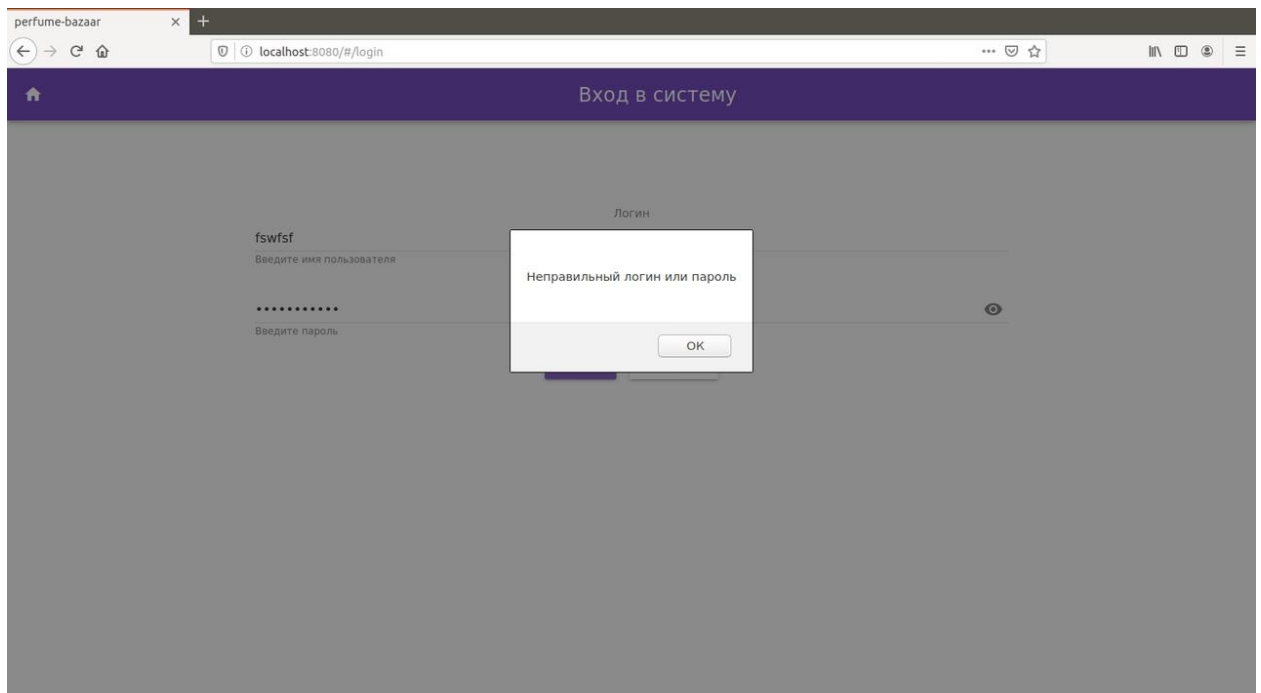
*Рисунок 1 – Главное окно до входа*

Окно входа в систему (рисунок 2).



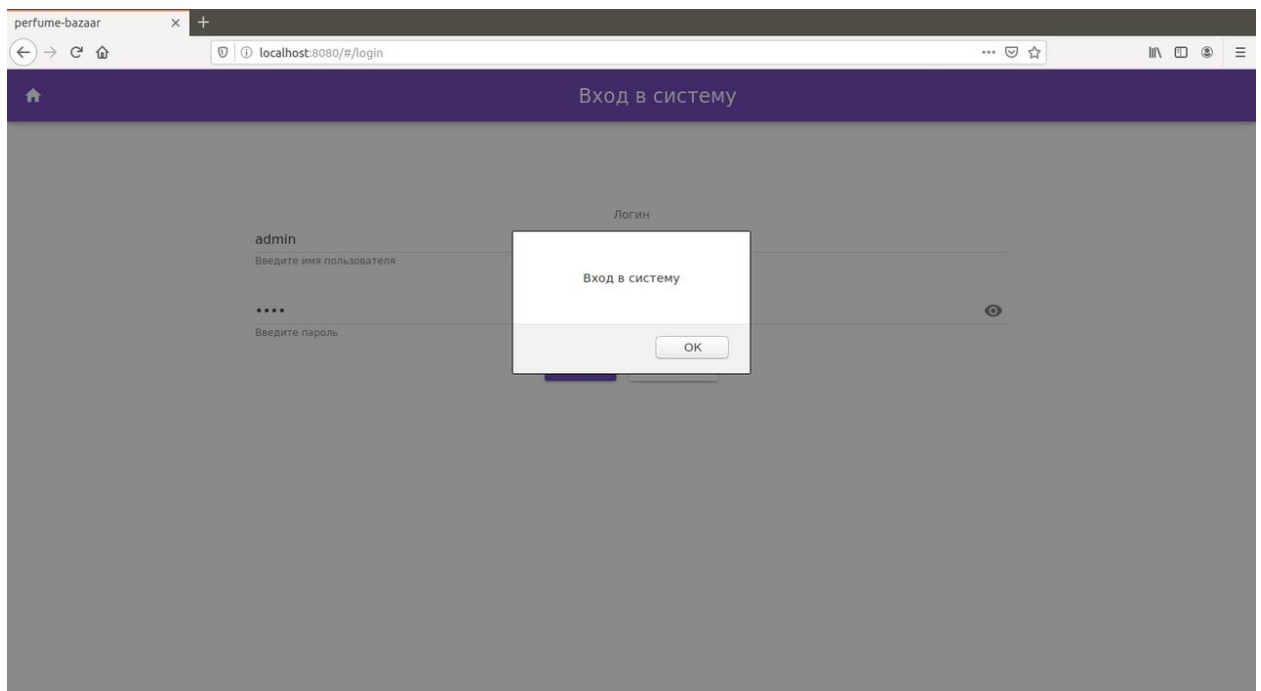
*Рисунок 2 – Окно входа*

Неудачная попытка входа в систему (рисунок 3).



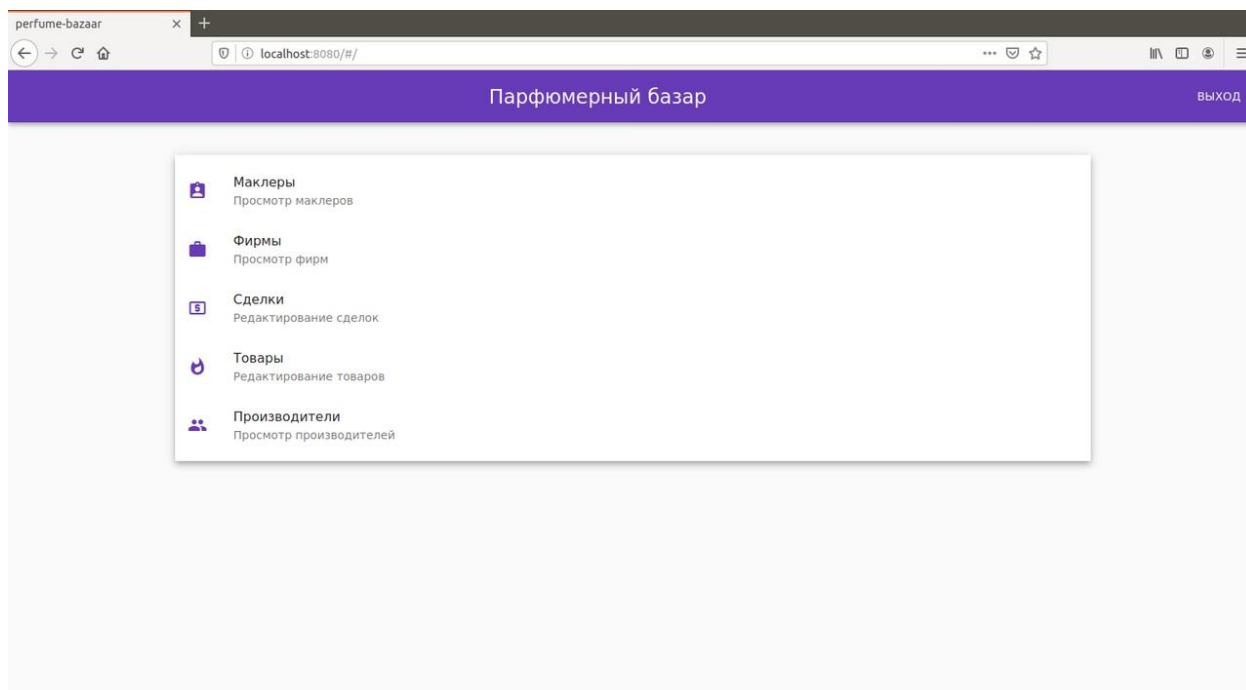
*Рисунок 3 – Неудачная попытка входа*

Удачная попытка входа в систему (рисунок 4).



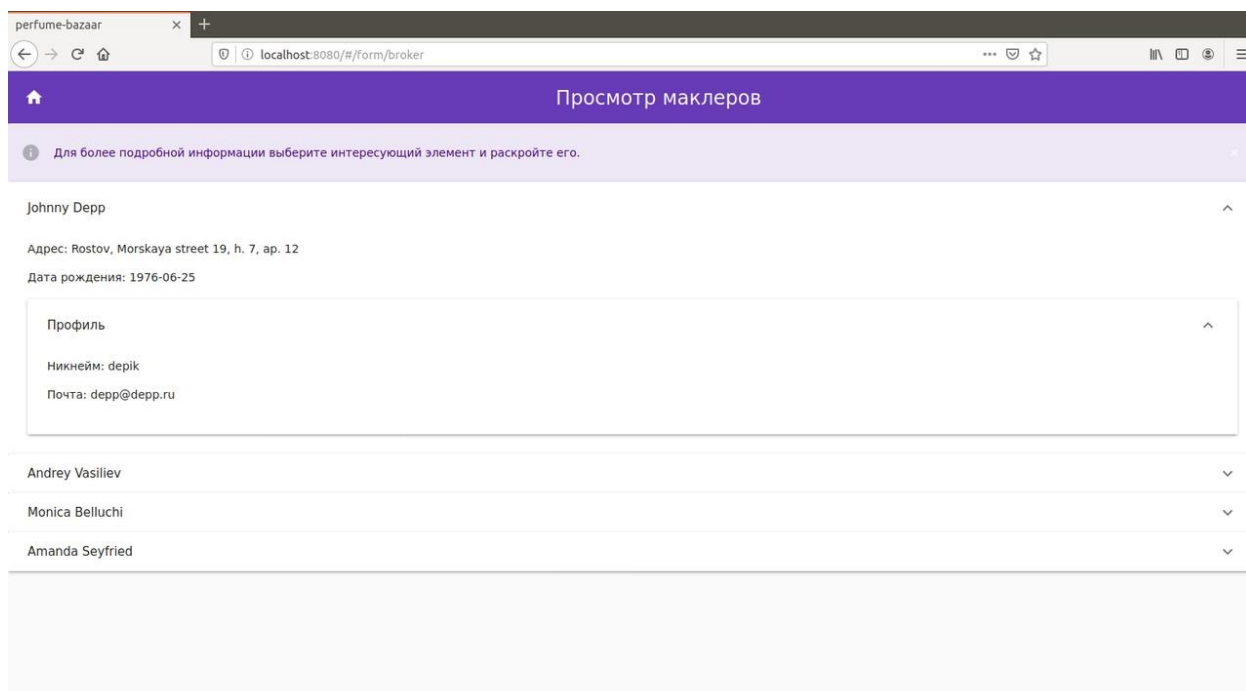
*Рисунок 4 – Удачная попытка входа*

Главное окно приложения после входа в систему (рисунок 5).



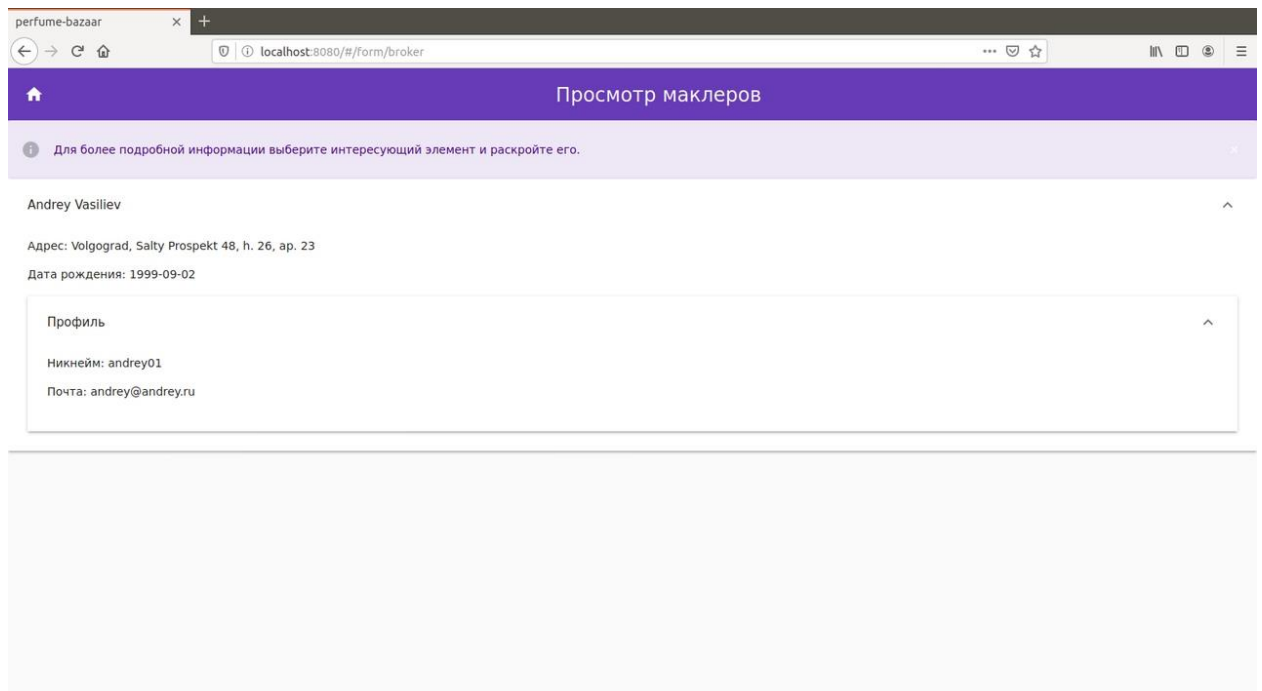
*Рисунок 5 – Главное окно после входа*

Интерфейс просмотра доступных маклеров, вход от админа (главного маклера), рисунок 6.



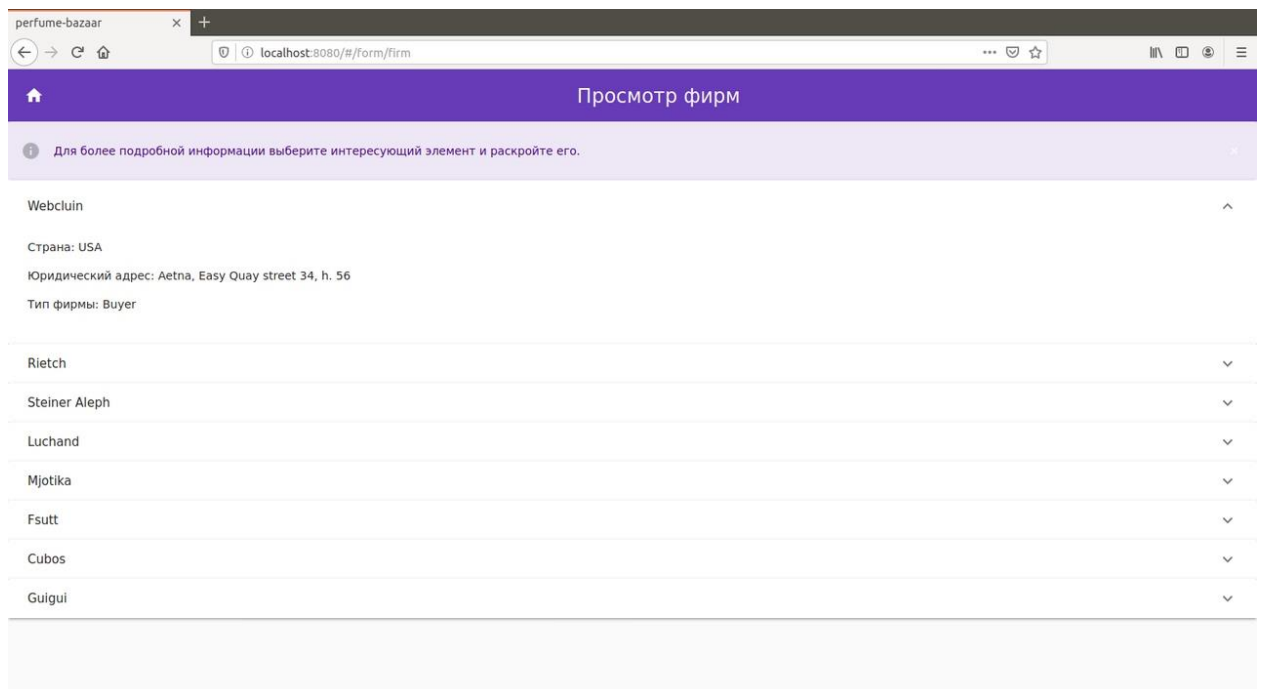
*Рисунок 6 – Просмотр маклеров от админа*

Интерфейс просмотра текущего маклера, вход от простого пользователя (простого маклера), рисунок 7.



*Рисунок 7 – Просмотр маклера от простого пользователя*

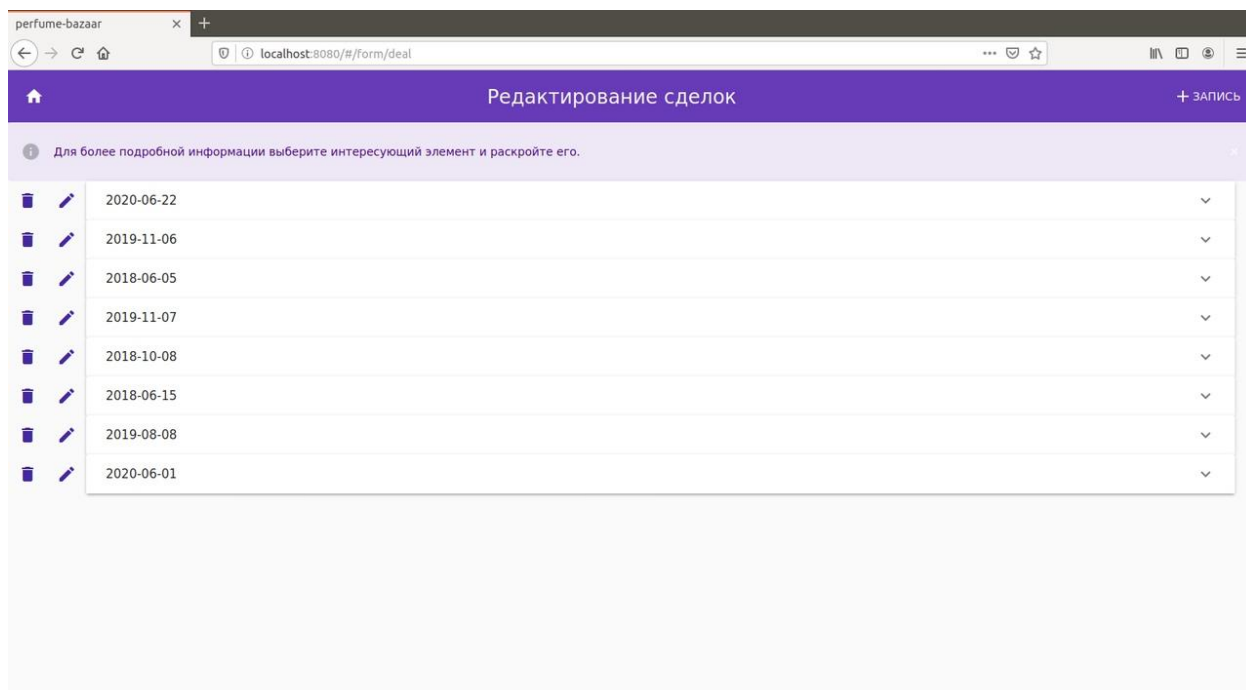
Интерфейс просмотра фирм (рисунок 8).



*Рисунок 8 – Просмотр фирм*

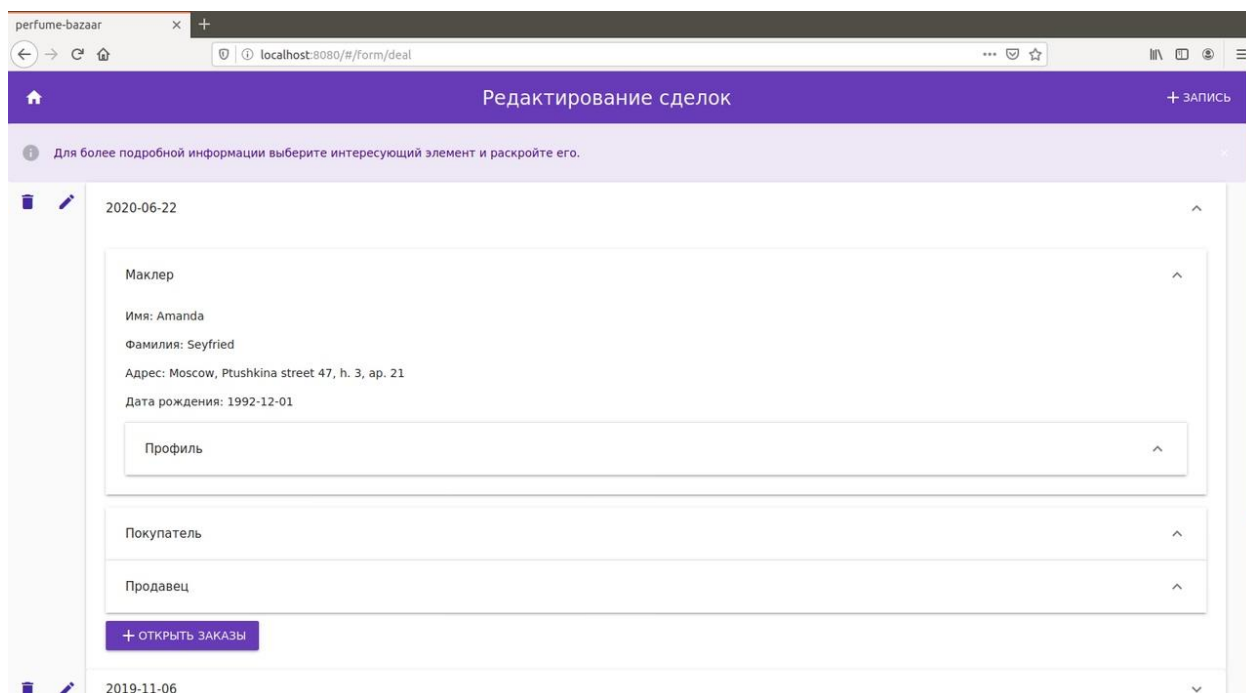
CRUD интерфейс сделок при входе от админа представлен ниже.

Просмотр сделок (рисунок 9).



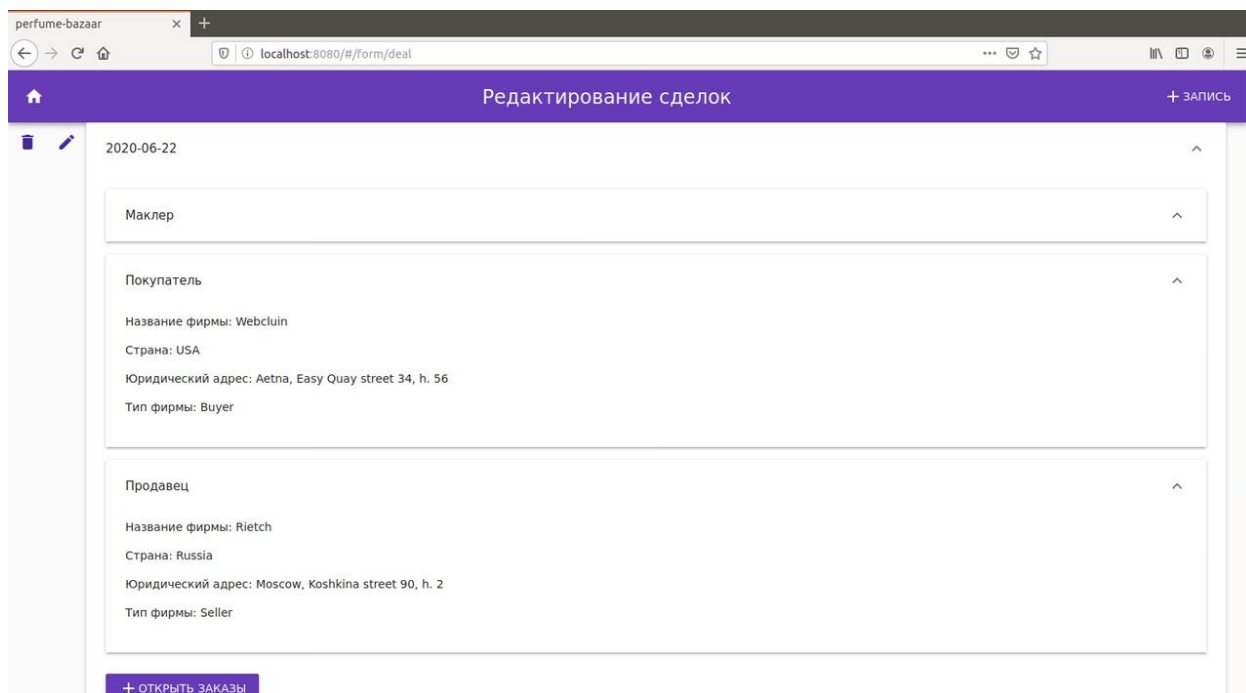
*Рисунок 9 – Просмотр сделок от админа*

Раскрытая сделка, информация о маклере (рисунок 10).



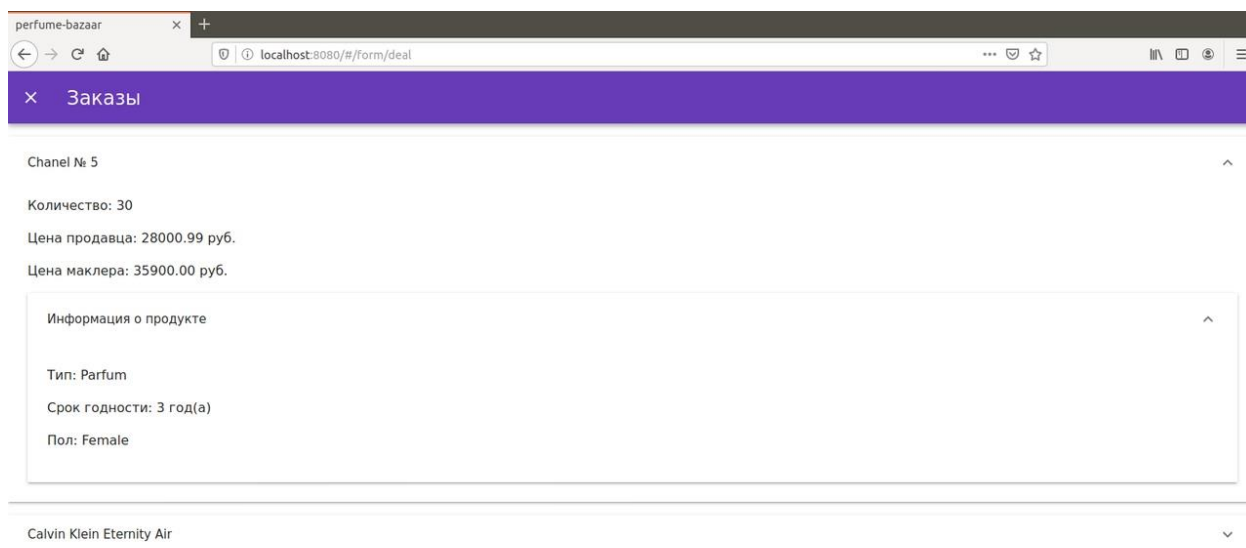
*Рисунок 10 – Раскрытая сделка, маклер*

Раскрытая сделка, продавец и покупатель, закрытая подсказка (рисунок 11).



*Рисунок 11 – Раскрытая сделка, покупатель и продавец*

Просмотр заказов конкретной сделки (рисунок 12).



*Рисунок 12 – Заказы*

Редактирование сделки (рисунок 13).

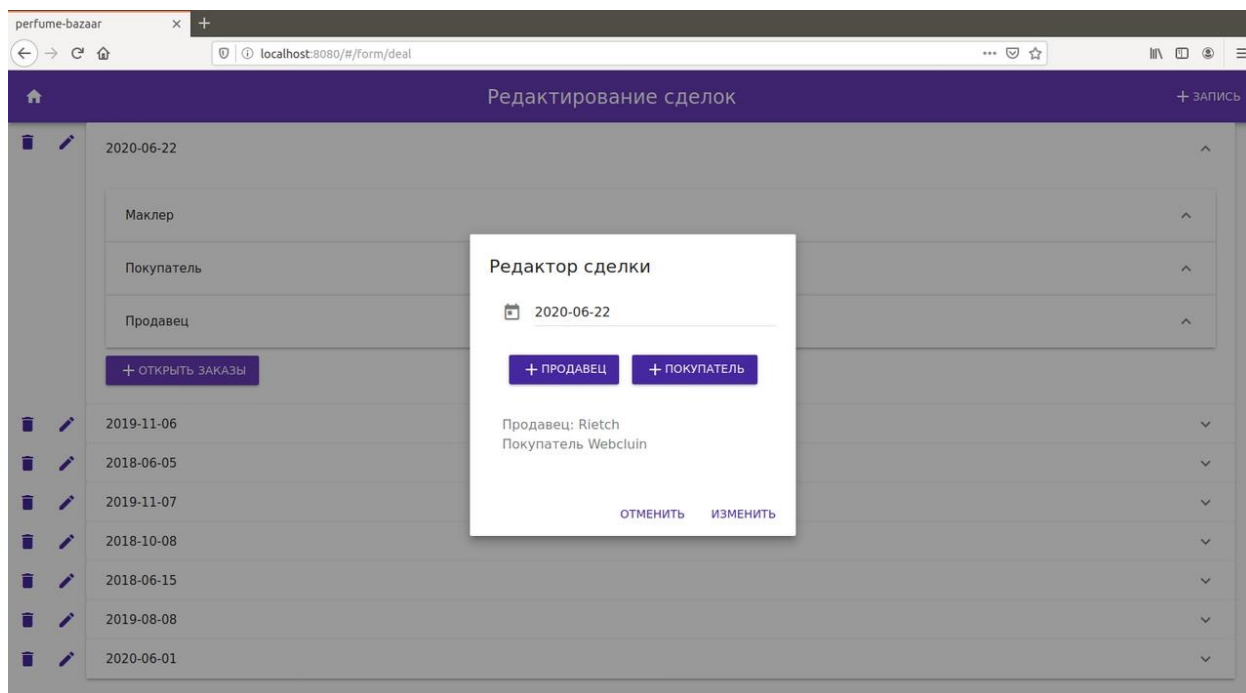


Рисунок 13 – Редактирование сделки

Выбор продавца (рисунок 14).

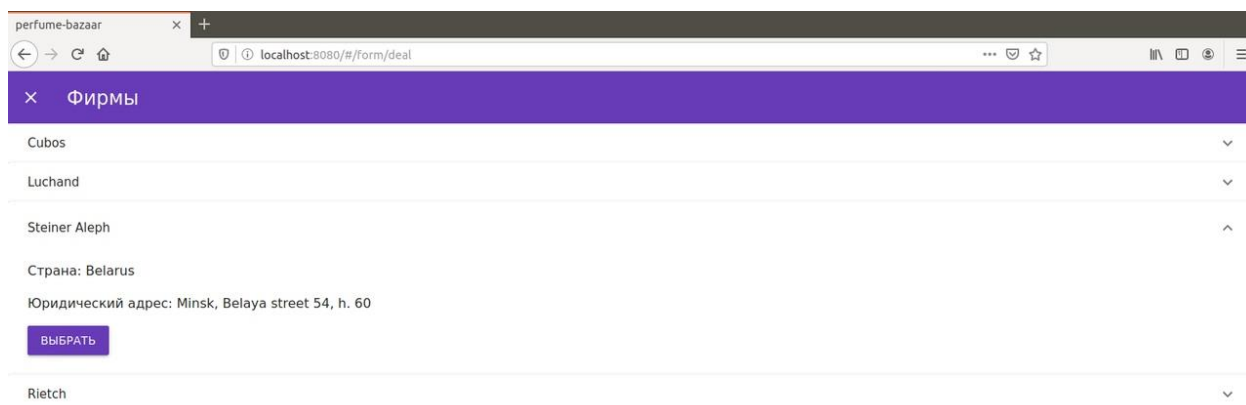
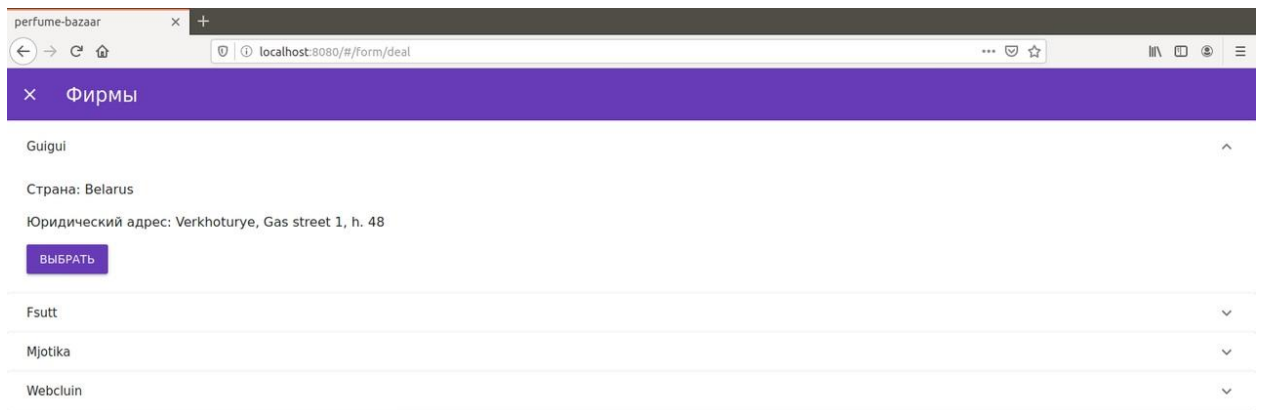


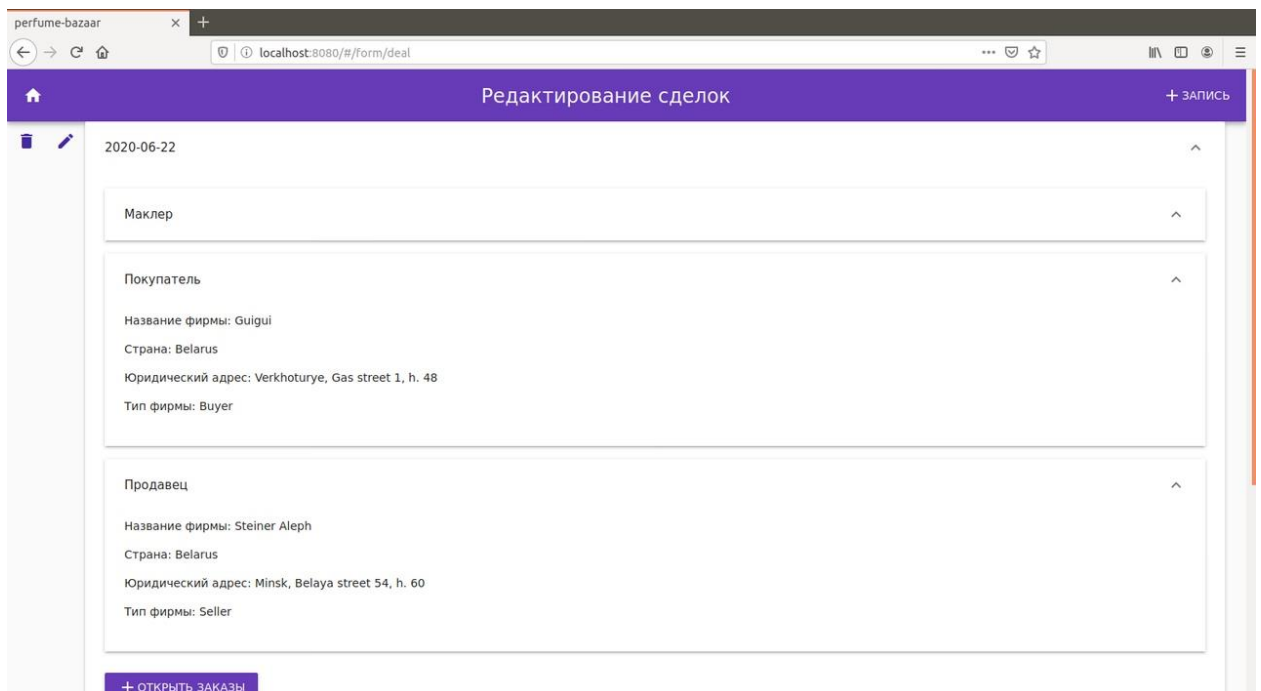
Рисунок 14 – Выбор продавца

Выбор покупателя (рисунок 15).



*Рисунок 15 – Выбор покупателя*

После изменения сделки (рисунок 16).



*Рисунок 16 – После изменения сделки*

После удаления сделки (рисунок 17).



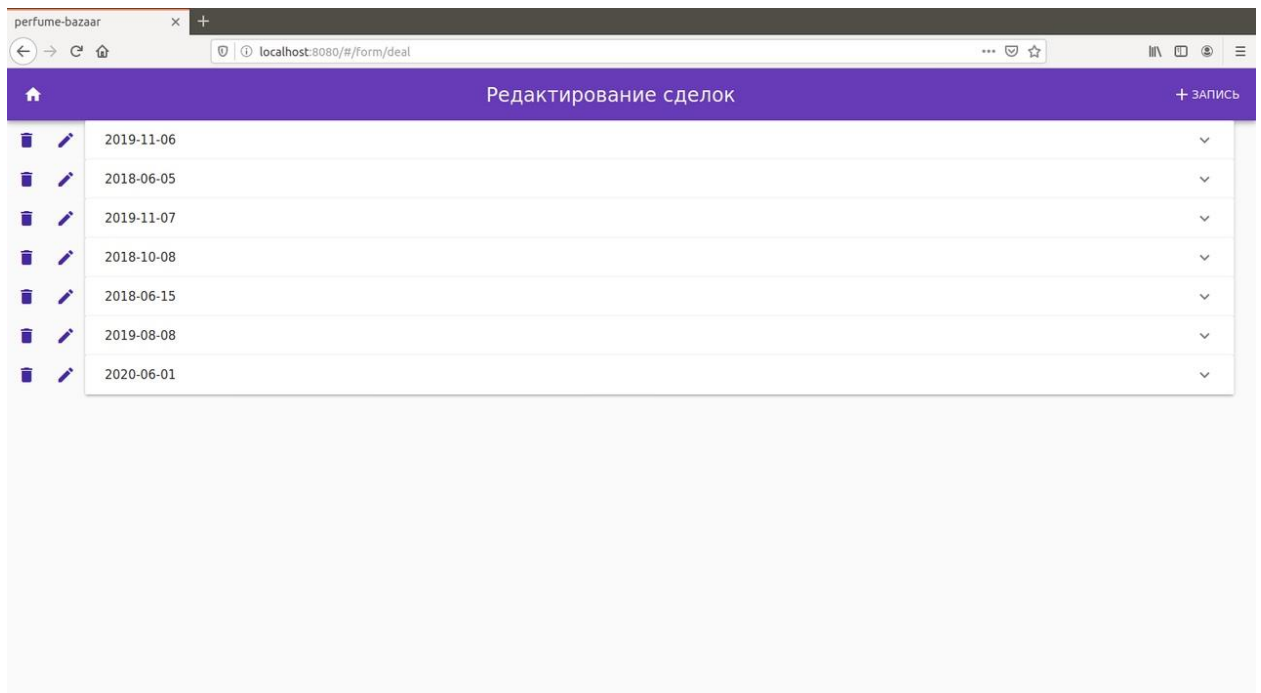


Рисунок 17 – Удаление сделки

Создание новой сделки (рисунок 18).

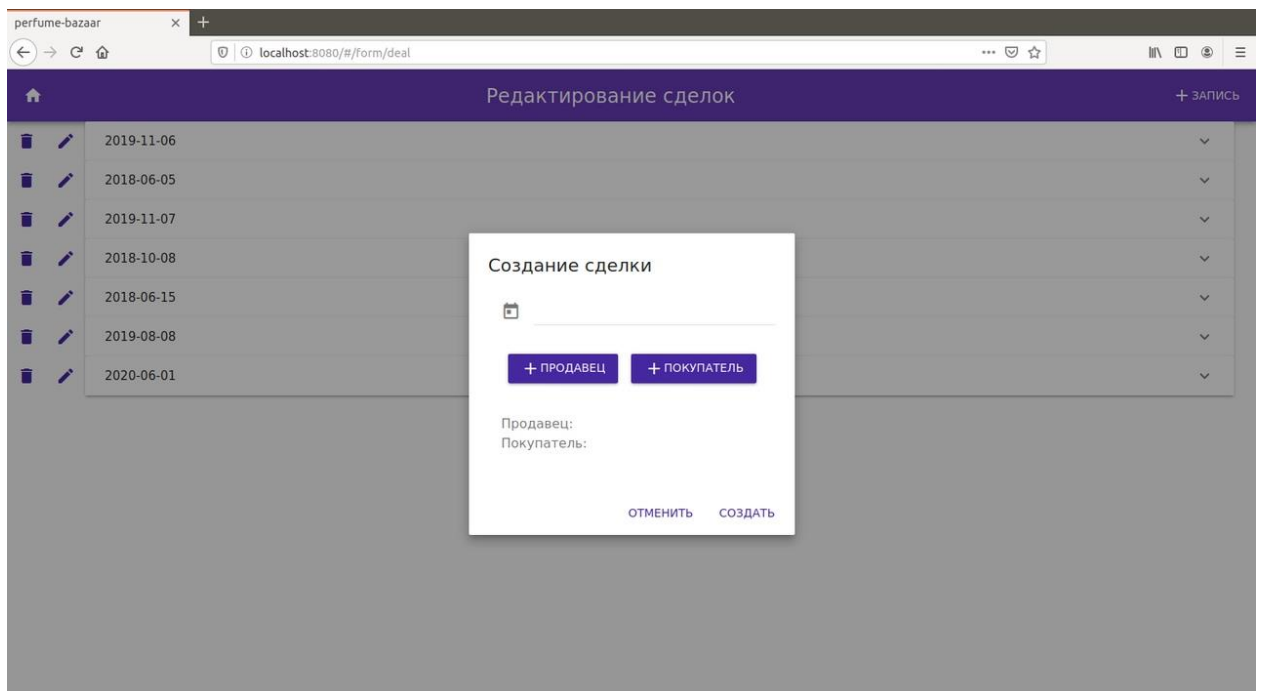
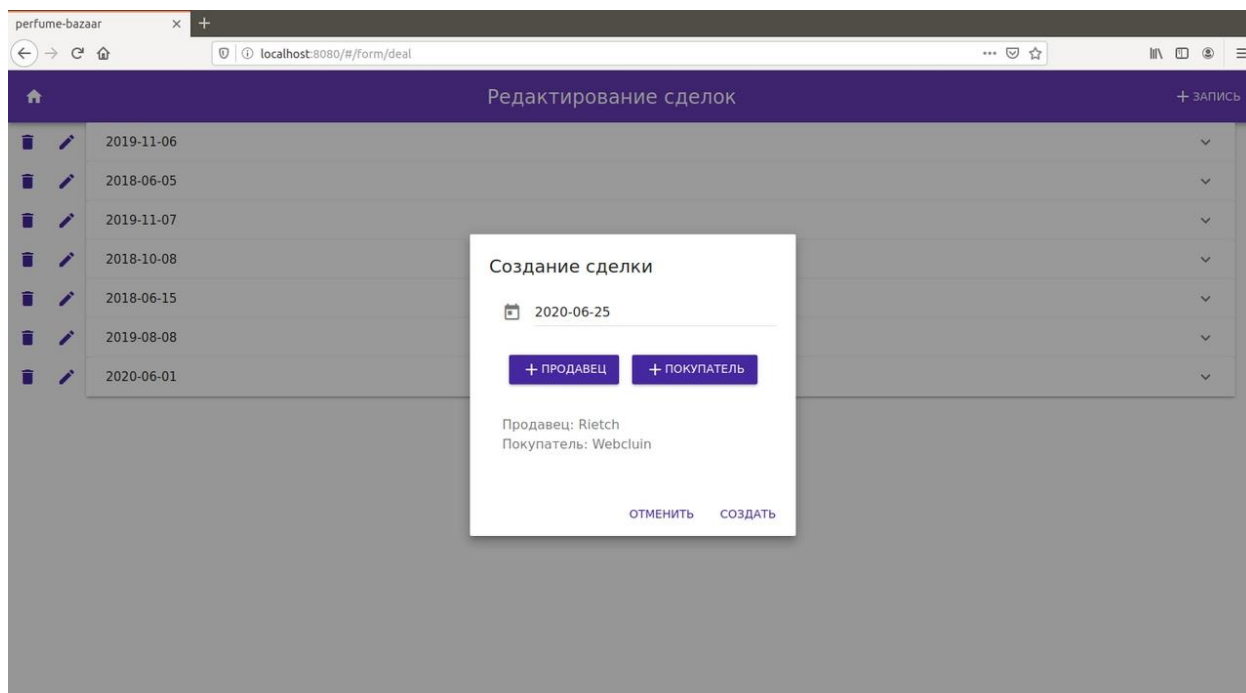


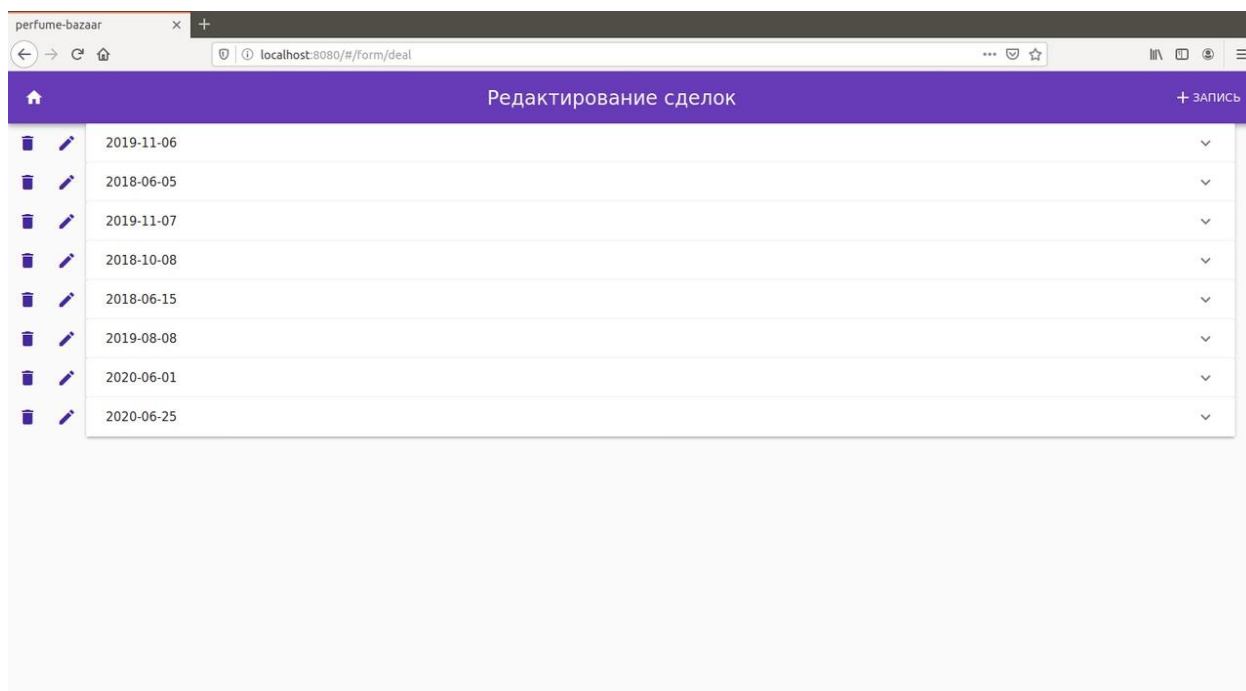
Рисунок 18 – Создание сделки

Заполнение полей для создания сделки (рисунок 19).



*Рисунок 19 – Заполнение полей для создания сделки*

После создания сделки (рисунок 20).



*Рисунок 20 – После создания сделки*

Просмотр сделок при входе от простого пользователя (рисунок 21).

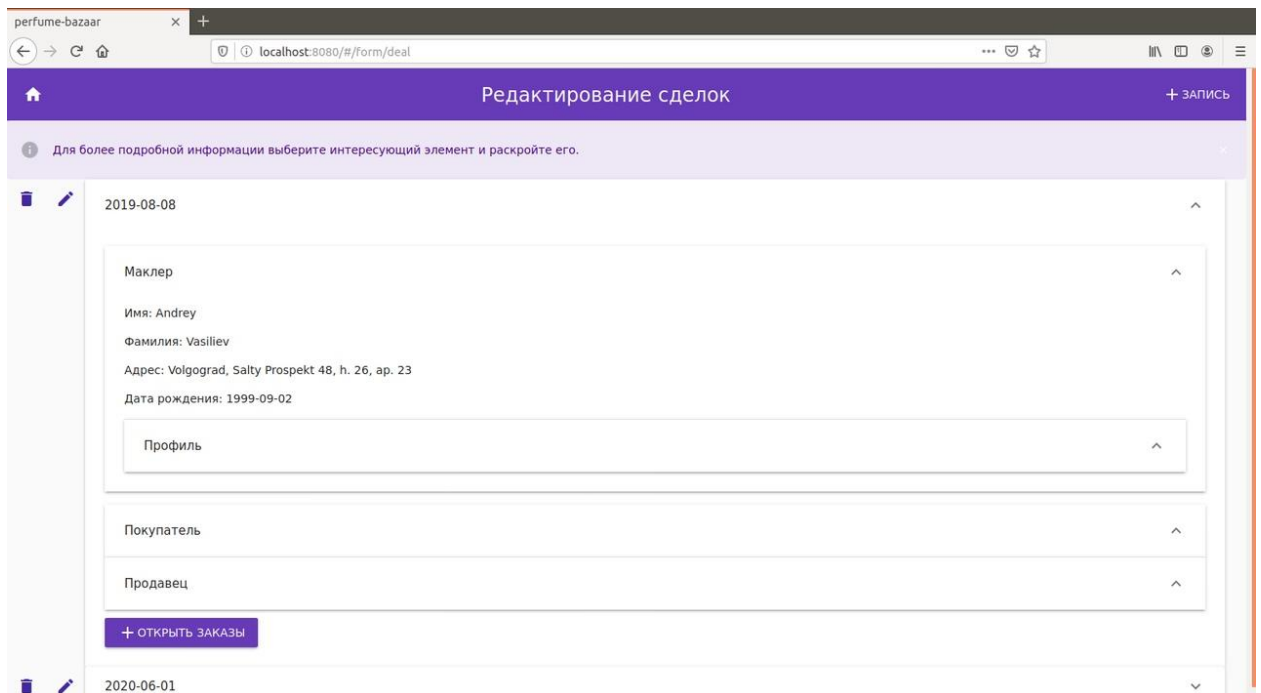


Рисунок 21 – Просмотр сделок от простого пользователя

CRUD интерфейс товаров представлен ниже.

Просмотр товаров (рисунок 22).

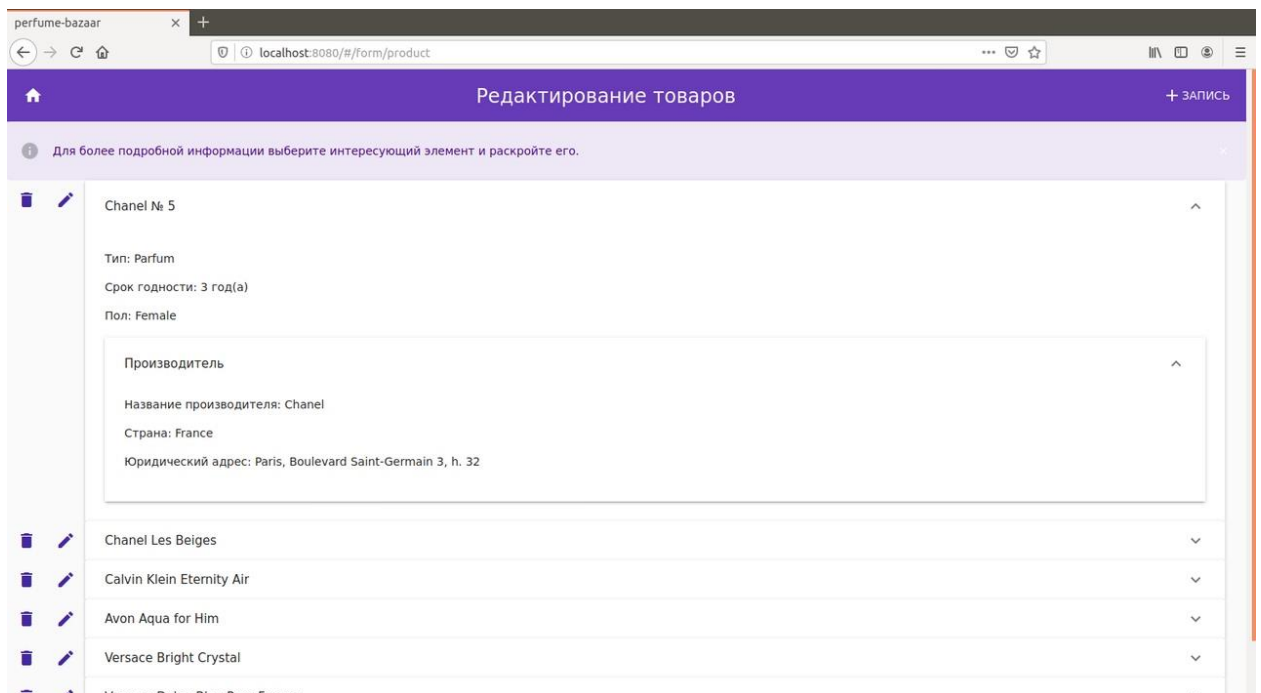


Рисунок 22 – Просмотр товаров

Редактирование товара (рисунок 23).

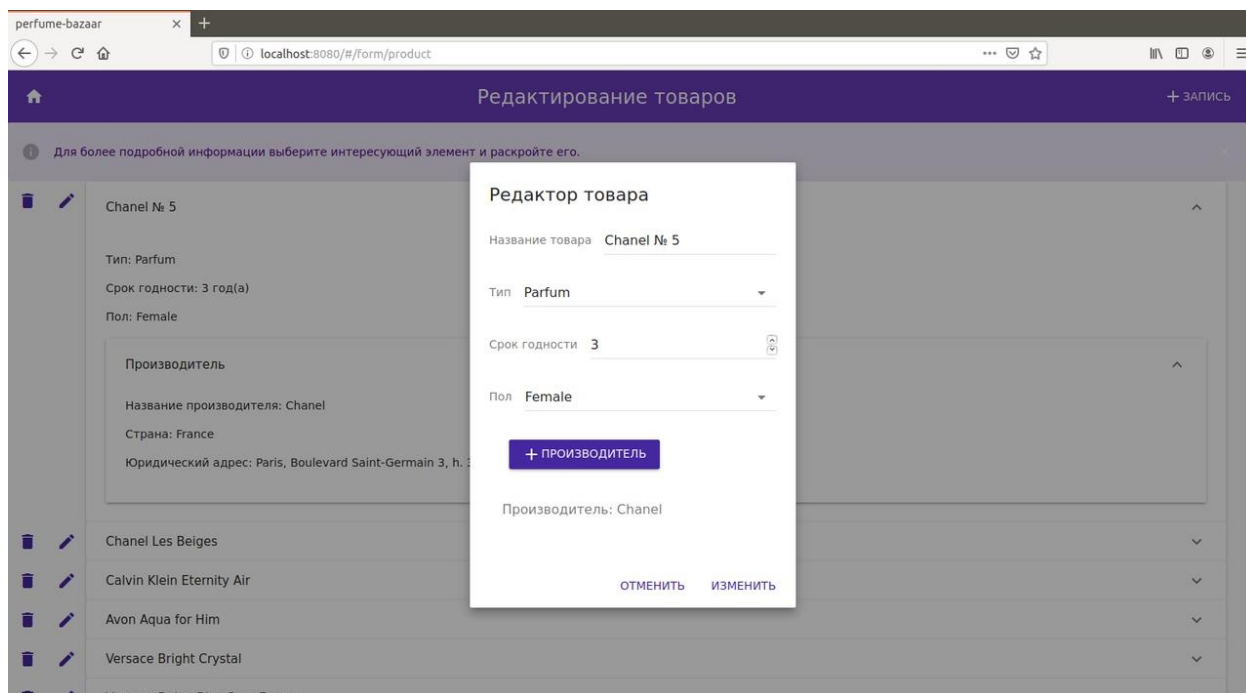


Рисунок 23 – Редактирование товара

Выбор производителя (рисунок 24).

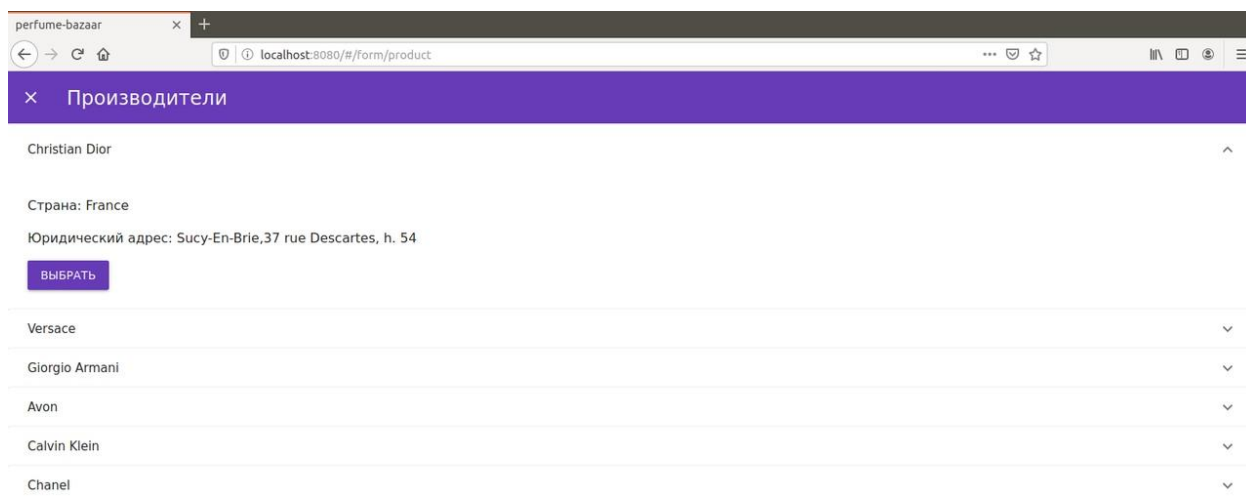


Рисунок 24 – Выбор производителя

После изменения товара (рисунок 25).

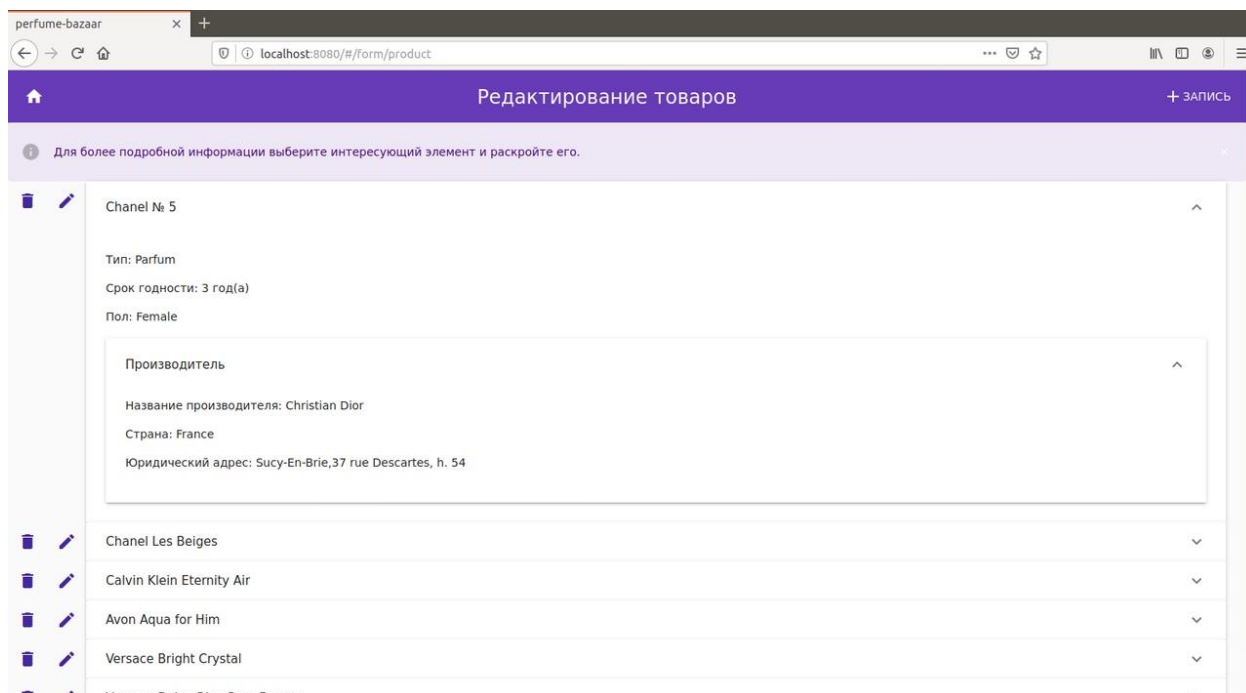


Рисунок 25 – После изменения товара

После удаления товара (рисунок 26).

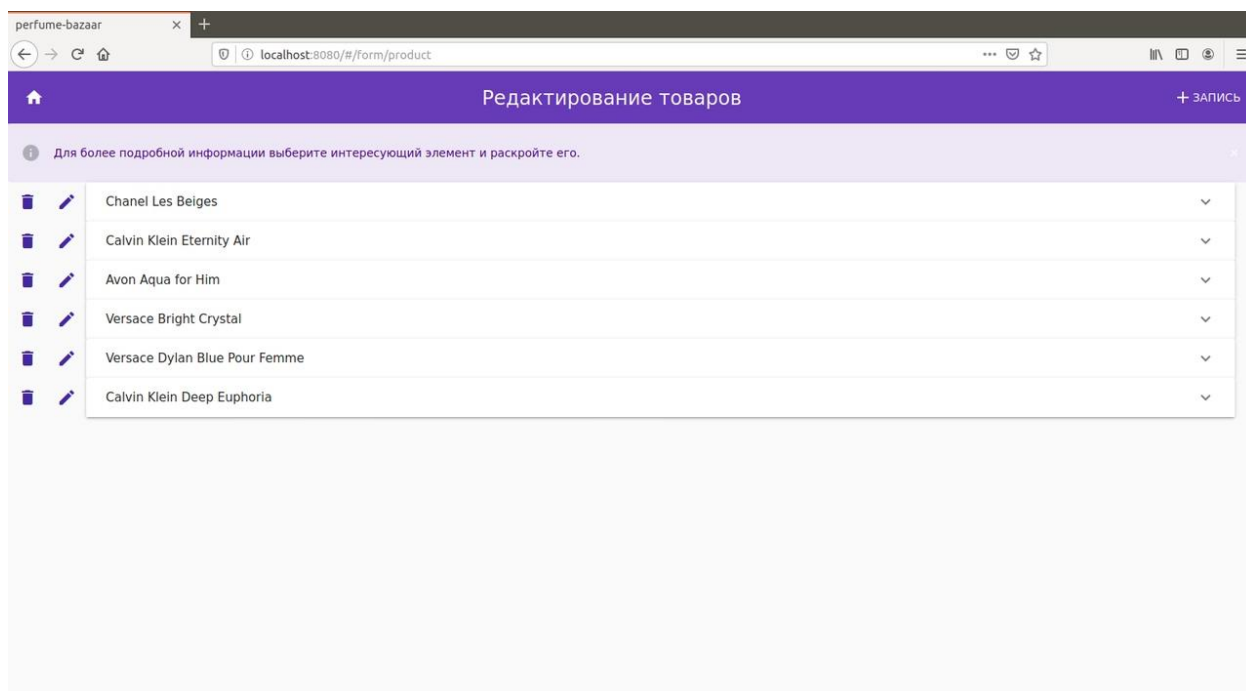
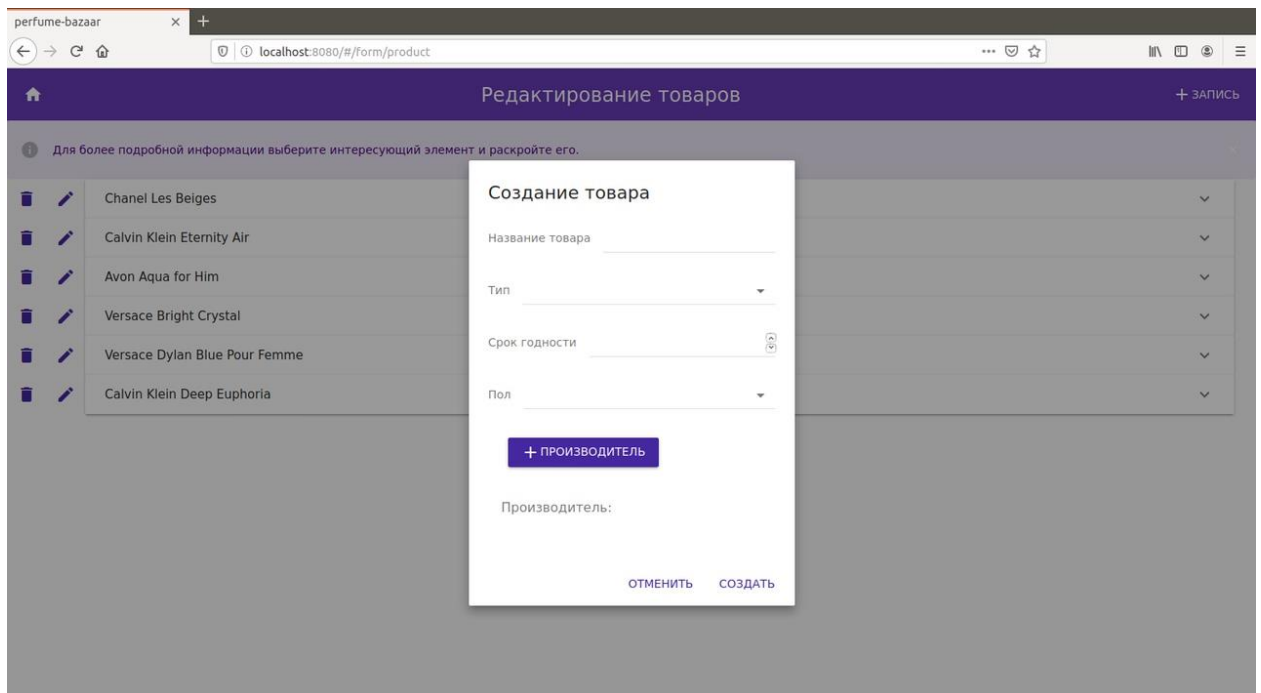


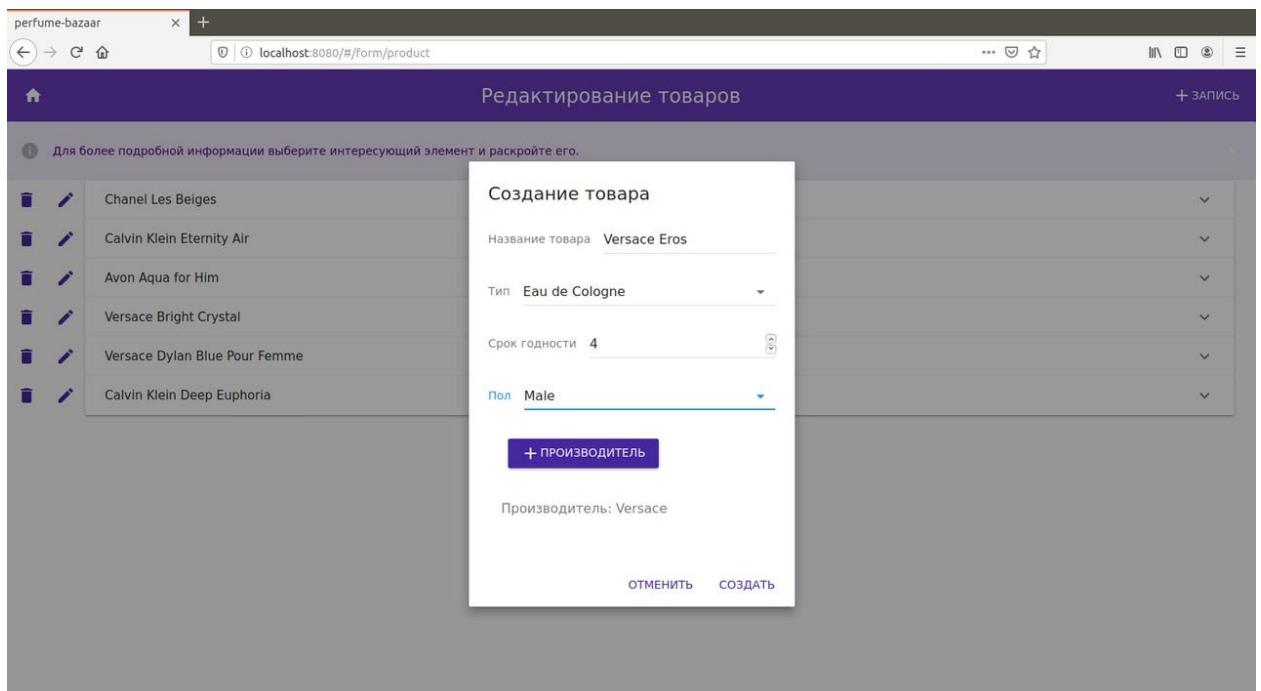
Рисунок 26 – После удаления товара

Создание нового товара (рисунок 27).



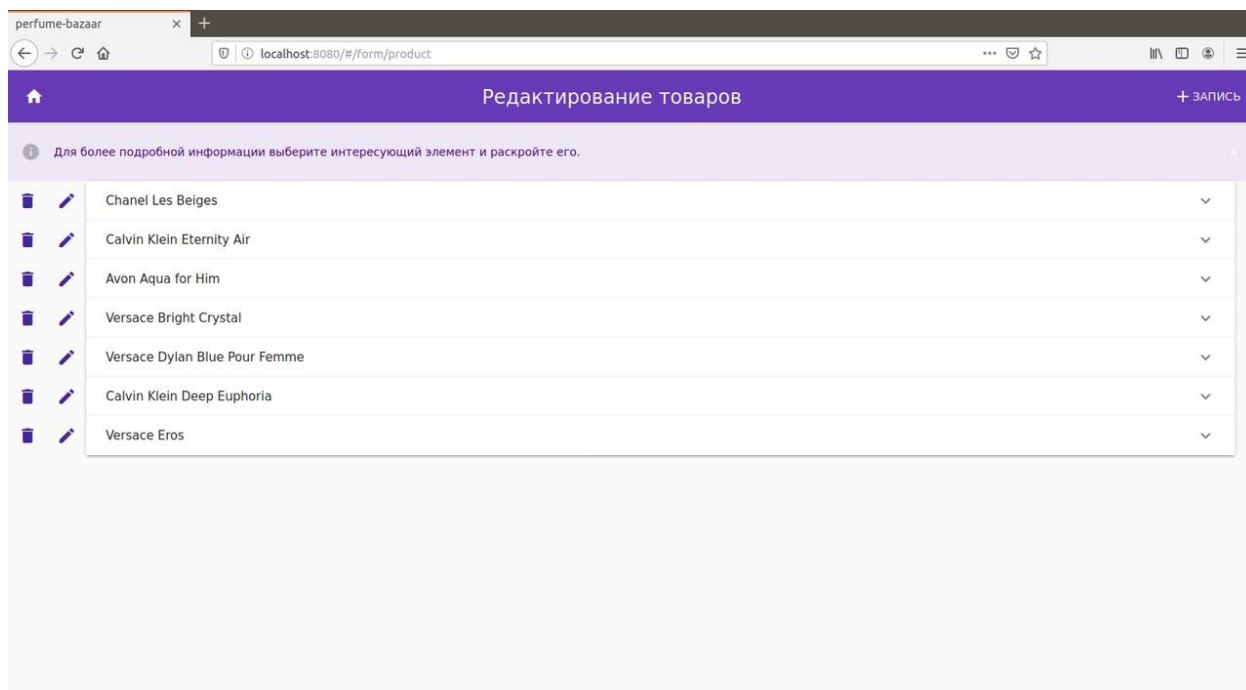
*Рисунок 27 – Создание нового товара*

Заполнение полей для создания нового товара (рисунок 28).



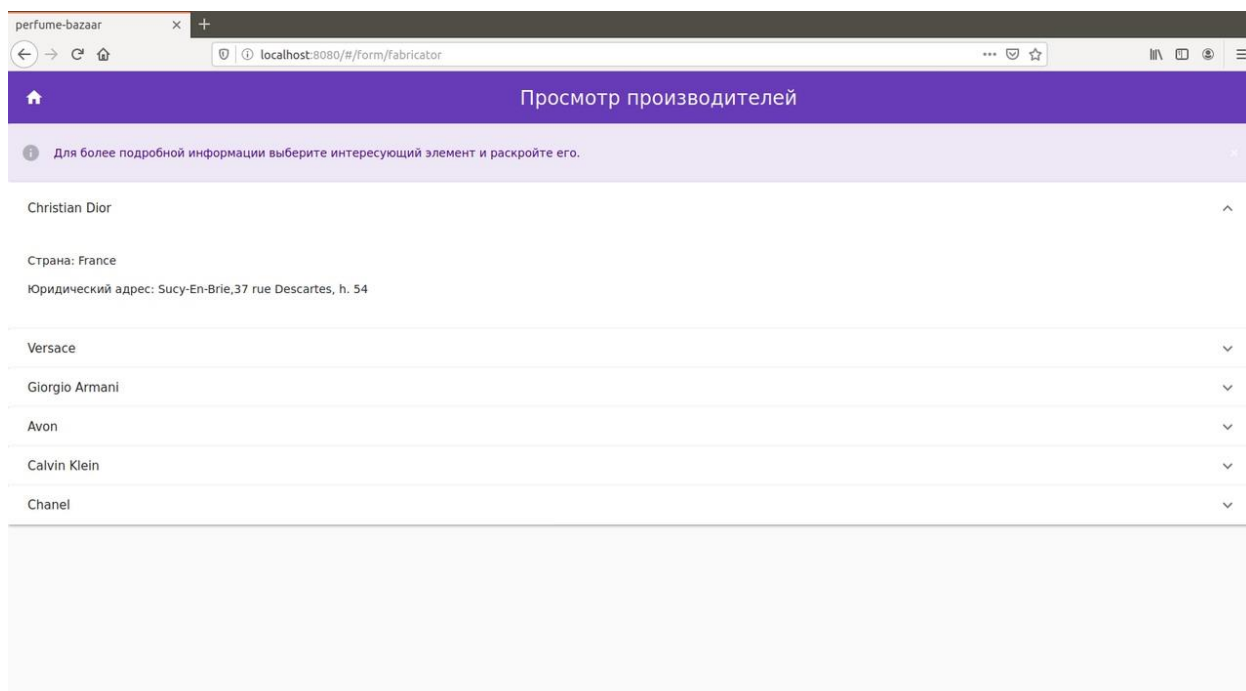
*Рисунок 28 – Заполнение полей для создания нового товара*

После создания нового товара (рисунок 29).



*Рисунок 29 – После создания товара*

Интерфейс просмотра производителей (рисунок 30).



*Рисунок 30 – Просмотр производителей*

Программный код находится в приложении Б.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения работы были выполнены задачи по разработке системы, а именно:

- была разработана модель базы данных по предметной области “Парфюмерный базар” в MySQL Workbench;
- были определены интерфейсы приложения;
- была разработана программно-аппаратная часть веб – приложения;
- был разработан пользовательский интерфейс и функционал веб – приложения.

В результате был разработан прототип веб – приложения для предметной области – “Парфюмерный базар” с использованием СУБД SQLite. Соответственно, были углублены знания и практические умения и получен начальный практический опыт по основным видам деятельности «Разработка, администрирование и защита баз данных» и освоены соответствующие общие и профессиональные компетенции: ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9, ОК 10, ОК 11, ПК 11.1., ПК 11.2., ПК 11.3, ПК 11.4., ПК 11.5., ПК 11.6.



## СПИСОК ИСТОЧНИКОВ

1. Введение в Django [Электронный ресурс] // Metanit.com [сайт] // URL: <https://metanit.com/python/django/1.1.php> (дата обращения 26.06.2020).
2. Работа в MySQL Workbench [Электронный ресурс] // life-prog.ru [сайт] // URL: [https://life-prog.ru/2\\_43979\\_sozdanie-EER-diagrammi.html](https://life-prog.ru/2_43979_sozdanie-EER-diagrammi.html) (дата обращения 26.06.2020).
3. Muse-UI [Электронный ресурс] // Muse-UI.org [сайт] // URL: <https://muse-ui.org/#/en-US> (дата обращения 26.06.2020).

## ПРИЛОЖЕНИЕ А

Схема инфологической модели базы данных (рисунок 31).

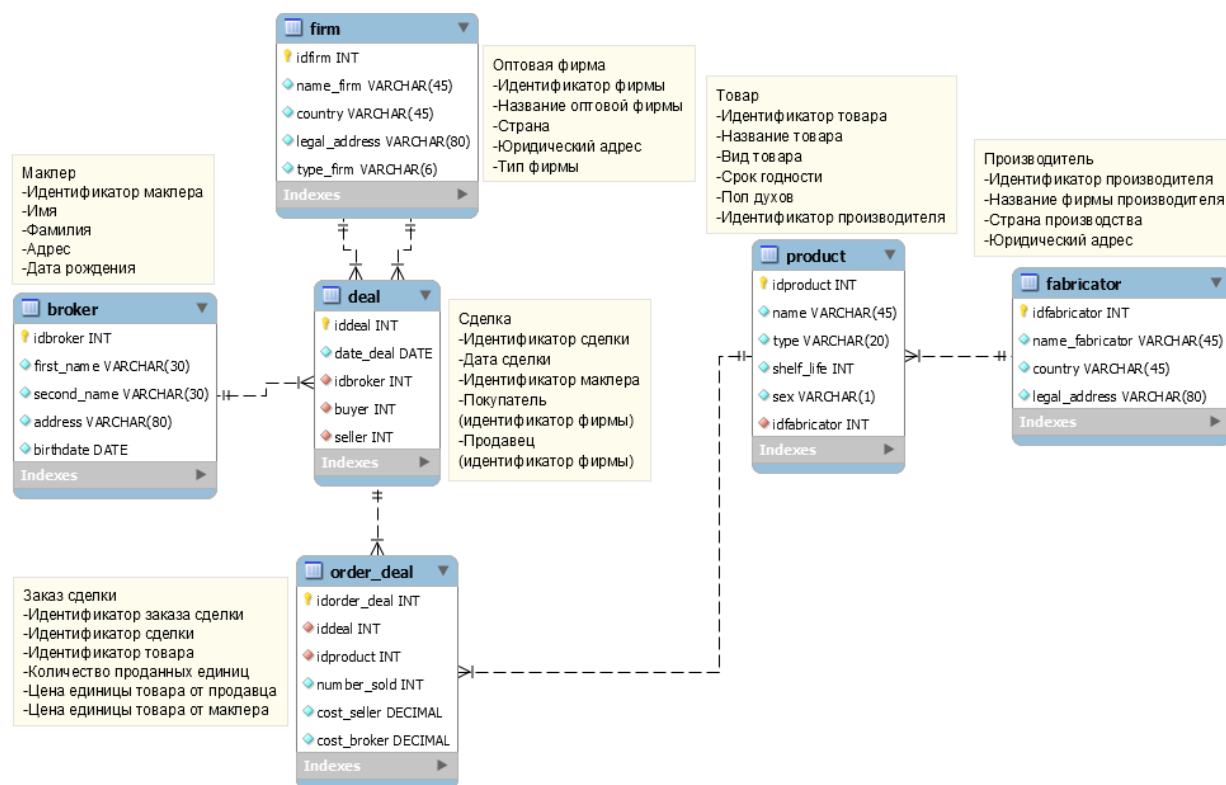


Рисунок 31 – Схема инфологической модели

Логическая модель базы данных в виде описания таблиц представлена ниже.

Таблица 1 – broker (маклер)

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
idbroker	Int	+	+	-	Уникален	Идентификатор	1
first_name	Varchar (30)	+	-	-	Буквы	Имя	Angelina
second_name	Varchar (30)	+	-	-	Буквы	Фамилия	Jolie
address	Varchar (80)	+	-	-	Буквы, цифры	Адрес	Moscow, Griboyedov street 1, h. 10, ap. 23
birthdate	Date	+	-	-	Маска: YYYY:MM:DD	Дата рождения	1992-10-12

Таблица 2 – firm (фирма)

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
idfirm	Int	+	+	-	Уникален	Идентификатор	1
name_firm	Varchar (45)	+	-	-	Буквы, цифры	Название оптовой фирмы	Mjonto
country	Varchar (45)	+	-	-	Буквы	Страна	Russia
legal_address	Varchar (80)	+	-	-	Буквы, цифры	Юридический адрес	Sochi, Morskaya street 15, h. 45
type_firm	Varchar (6)	+	-	-	Буквы	Тип фирмы (продавец, покупатель)	Buyer

Таблица 4 – deal (сделка)

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
iddeal	Int	+	+	-	Уникален	Идентификатор	1
date_deal	Date	+	-	-	Маска: YYYY:MM:DD	Дата сделки	2020-01-19
idbroker	Int	+	-	+	Каскадируется по значению первичного ключа таблицы broker	Идентификатор маклера	1
buyer	Int	+	-	+	Каскадируется по значению первичного ключа таблицы firm	Покупатель (идентификатор фирмы)	2
seller	Int	+	-	+	Каскадируется по значению первичного ключа таблицы firm	Продавец (идентификатор фирмы)	3

Таблица 4 – order\_deal (заказ сделки)

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
idorder_deal	Int	+	+	-	Уникален	Идентификатор	1
ideal	Int	+	-	+	Каскадируется по значению первичного ключа таблицы deal	Идентификатор сделки	2
idproduct	Int	+	-	+	Каскадируется по значению первичного ключа таблицы product	Идентификатор товара	3
number_sold	Int	+	-	-	Только цифры, не отрицательные	Количество проданных единиц товара	120
cost_seller	Decimal	+	-	-	Только цифры, не отрицательные	Цена единицы товара от продавца	12000,99
cost_buyer	Decimal	+	-	-	Только цифры, не отрицательные	Цена единицы товара от маклера	16000,99

Таблица 4 – product (товар)

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
idproduct	Int	+	+	-	Уникален	Идентификатор	1
name	Varchar (45)	+	-	-	Буквы, цифры, спецсимволы	Название товара	Chanel № 5
type	Varchar (20)	+	-	-	Буквы	Тип товара	Parfum
shelf_life	Int	+	-	-	Только цифры, не отрицательные	Срок годности	3
sex	Varchar (1)	+	-	-	Буквы	Пол товара	W
idfabricator	Int	+	-	+	Каскадируется по значению первичного ключа таблицы fabricator	Идентификатор производителя	4

Таблица 4 – fabricator (производитель)

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
idfabricator	Int	+	+	-	Уникален	Идентификатор	1
name_fabricator	Varchar(45)	+	-	-	Буквы, цифры	Название производителя	Calvin Klein
country	Varchar(45)	+	-	-	Буквы	Страна производства	USA
legal_address	Varchar(80)	+	-	-	Буквы, цифры	Юридический адрес	New York, Bank Street 13, h. 10



## ПРИЛОЖЕНИЕ Б

Ниже представлен листинг документированного кода программы.

*Листинг Б.1 – perfume/settings*

```
"""
Django settings for perfume project.

Generated by 'django-admin startproject' using Django 3.0.7.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See
# https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '63) (034_eybmbs7)-!o40_23fd04zu%fp(@^70w$a&t6cl7q5'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'django_summernote',

    'rest_framework',
    'rest_framework.authtoken',
```

```

    'djoser',
    'perfume_app'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'perfume.urls'

TEMPLATES = [
    {
        'BACKEND':
'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')]
    },
    {
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'perfume.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-
password-validators

```

```

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarity
Validator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator'
,
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator'
,
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidato
r',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'

REST_FRAMEWORK = {
    'DEFAULT_PERMISSION_CLASSES': (
        'rest_framework.permissions.IsAdminUser',
        'rest_framework.permissions.AllowAny',
    ),
    'PAGE_SIZE': 10,
    'DEFAULT_AUTHENTICATION_CLASSES': (
        #
'rest_framework_jwt.authentication.JSONWebTokenAuthentication',
        'rest_framework.authentication.TokenAuthentication',
    ),
}

```

```

        'rest_framework.authentication.BasicAuthentication',
        'rest_framework.authentication.SessionAuthentication',
    ),
    'EXCEPTION_HANDLER':
'rest_framework_json_api.exceptions.exception_handler',
    'DEFAULT_PAGINATION_CLASS':

'rest_framework_json_api.pagination.PageNumberPagination',
    'DEFAULT_PARSER_CLASSES': (
        'rest_framework_json_api.parsers.JSONParser',
        'rest_framework.parsers.FormParser',
        'rest_framework.parsers.MultiPartParser'
    ),
    'DEFAULT_RENDERER_CLASSES': (
        'rest_framework_json_api.renderers.JSONRenderer',
        'rest_framework.renderers.BrowsableAPIRenderer',
    ),
    'DEFAULT_METADATA_CLASS':
'rest_framework_json_api.metadata.JSONAPIMetadata',
}

```

```
CORS_ORIGIN_ALLOW_ALL = True
```

### *Листинг Б.2 – perfume/urls*

```
"""perfume URL Configuration
```

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/3.0/topics/http/urls/>

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to urlpatterns: `path('', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to urlpatterns: `path('', Home.as_view(), name='home')`

Including another URLconf

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to urlpatterns: `path('blog/', include('blog.urls'))`

```
"""
```

```
from django.contrib import admin
```

```
from django.urls import path, include
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/v1/perfume/', include("perfume_app.urls")),
    path('auth/', include('djoser.urls')),

```

```

    path('auth/', include('djoser.urls.authtoken')),
    path('auth/', include('djoser.urls.jwt')),
]

```

*Листинг Б.3 – perfume\_app/admin*

```

from django.contrib import admin
from perfume_app.models import *

```

```

class Admin(admin.ModelAdmin):
    pass

```

```

admin.site.register(Broker)
admin.site.register(Firm)
admin.site.register(Deal)
admin.site.register(OrderDeal)
admin.site.register(Product)
admin.site.register(Fabricator)

```

*Листинг Б.4 – perfume\_app/models*

```

from django.db import models
from django.contrib.auth.models import User

```

```

# from djoser.urls.base import User

```

```

class Broker(models.Model):
    first_name = models.CharField(max_length=30)
    second_name = models.CharField(max_length=30)
    address = models.CharField(max_length=80)
    birthdate = models.DateField()
    broker = models.ForeignKey(User, blank=True, null=True,
on_delete=models.PROTECT)

```

```

class Firm(models.Model):
    TYPE_FIRM = [('S', 'Seller'), ('B', 'Buyer')]
    name_firm = models.CharField(max_length=45)
    country = models.CharField(max_length=45)
    legal_address = models.CharField(max_length=80)
    type_firm = models.CharField(max_length=1, choices=TYPE_FIRM)

```

```

class Deal(models.Model):
    date_deal = models.DateField()
    broker = models.ForeignKey(Broker, on_delete=models.CASCADE)
    buyer = models.ForeignKey(Firm, related_name='buyer',
on_delete=models.CASCADE)

```

```

        seller = models.ForeignKey(Firm, related_name='seller',
on_delete=models.CASCADE)

class Fabricator(models.Model):
    name_fabricator = models.CharField(max_length=45)
    country = models.CharField(max_length=45)
    legal_address = models.CharField(max_length=80)

class Product(models.Model):
    TYPE = [
        ('PA', 'Parfum'),
        ('EP', 'Eau de Parfum'),
        ('ET', 'Eau de Toilette'),
        ('EC', 'Eau de Cologne'),
        ('EF', 'Eau Fraiche')
    ]
    SEX = [('M', 'Male'), ('F', 'Female'), ('U', 'Unisex')]
    name = models.CharField(max_length=45)
    type = models.CharField(max_length=2, choices=TYPE)
    shelf_life = models.IntegerField()
    sex = models.CharField(max_length=1, choices=SEX)
    fabricator = models.ForeignKey(Fabricator,
on_delete=models.CASCADE)

class OrderDeal(models.Model):
    deal = models.ForeignKey(Deal, on_delete=models.CASCADE)
    product = models.ForeignKey(Product,
on_delete=models.CASCADE)
    number_sold = models.IntegerField()
    cost_seller = models.DecimalField(max_digits=12,
decimal_places=2)
    cost_broker = models.DecimalField(max_digits=12,
decimal_places=2)

```

#### *Листинг Б.5 – perfume\_app/serializers*

```

from rest_framework import serializers
from perfume_app.models import *

class UserSerializers(serializers.ModelSerializer):
    """Пользователь для get запроса"""
    class Meta:
        model = User
        fields = ("username", "email")

class BrokerSerializers(serializers.ModelSerializer):
    """Маклер для get запроса"""

```

```

broker = UserSerializers()

class Meta:
    model = Broker
    fields = ("id", "first_name", "second_name", "address",
"birthdate", "broker")

class FirmSerializers(serializers.ModelSerializer):
    """Фирма для get запроса"""
    class Meta:
        model = Firm
        fields = ("id", "name_firm", "country", "legal_address",
"type_firm")

class DealSerializers(serializers.ModelSerializer):
    """Сделка для get запроса"""
    broker = BrokerSerializers()
    buyer = FirmSerializers()
    seller = FirmSerializers()

    class Meta:
        model = Deal
        fields = ("id", "date_deal", "broker", "buyer", "seller")

class DealCRUDSerializers(serializers.ModelSerializer):
    """Сделка для put и post запросов"""
    class Meta:
        model = Deal
        fields = ("date_deal", "buyer", "seller")

class FabricatorSerializers(serializers.ModelSerializer):
    """Производитель для get запроса"""
    class Meta:
        model = Fabricator
        fields = ("id", "name_fabricator", "country",
"legal_address")

class ProductSerializers(serializers.ModelSerializer):
    """Товар для get запроса"""
    fabricator = FabricatorSerializers()

    class Meta:
        model = Product
        fields = ("id", "name", "type", "shelf_life", "sex",
"fabricator")

```

```

class ProductCRUDSerializers(serializers.ModelSerializer):
    """Товар для put и post запросов"""
    class Meta:
        model = Product
        fields = ("name", "type", "shelf_life", "sex",
"fabricator")

class OrderDealSerializers(serializers.ModelSerializer):
    """Заказ сделки для get запроса"""
    product = ProductSerializers()

    class Meta:
        model = OrderDeal
        fields = ("id", "deal", "product", "number_sold",
"cost_seller", "cost_broker")

```

*Листинг Б.6 – perfume\_app/views*

```

# from django.shortcuts import render
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import permissions
from rest_framework.generics import get_object_or_404
from django.db.models import Q
from perfume_app.models import *
from perfume_app.serializers import *

class BrokerView(APIView):
    """Просмотр маклеров"""
    permission_classes = [permissions.IsAuthenticated, ]

    def get(self, request):
        if request.user.is_superuser:
            brokers = Broker.objects.all()
            serializer = BrokerSerializers(brokers, many=True)
            return Response({"data": serializer.data})
        else:
            brokers = Broker.objects.filter(broker=request.user)
            serializer = BrokerSerializers(brokers, many=True)
            return Response({"data": serializer.data})

class FirmView(APIView):
    """Просмотр фирм"""
    permission_classes = [permissions.IsAuthenticated, ]

    def get(self, request):
        firms = Firm.objects.all()
        serializer = FirmSerializers(firms, many=True)
        return Response({"data": serializer.data})

```



```

class DealCRUDView(APIView):
    """CRUD сделок"""
    permission_classes = [permissions.IsAuthenticated, ]

    def get(self, request):
        if request.user.is_superuser:
            deals = Deal.objects.all()
            serializer = DealSerializers(deals, many=True)
            return Response({"data": serializer.data})
        else:
            broker = Broker.objects.filter(broker=request.user)
            deal = Deal.objects.filter(broker__in=broker)
            serializer = DealSerializers(deal, many=True)
            return Response({"data": serializer.data})

    def post(self, request):
        deal = DealCRUDSerializers(data=request.data)
        if deal.is_valid():
            broker_found = Broker.objects.get(broker=request.user)
            broker = get_object_or_404(Broker, id=broker_found.id)
            seller = get_object_or_404(Firm, id=self.request.data.get('seller'))
            buyer = get_object_or_404(Firm, id=self.request.data.get('buyer'))
            deal.save(broker=broker, seller=seller, buyer=buyer)
            return Response(status=201)
        else:
            return Response(status=400)

    def put(self, request, pk):
        saved_deal = get_object_or_404(Deal.objects.all(), pk=pk)
        serializer = DealCRUDSerializers(instance=saved_deal, data=request.data, partial=True)
        if serializer.is_valid(raise_exception=True):
            serializer.save()
            return Response(status=201)

    def delete(self, request, pk):
        deal = get_object_or_404(Deal.objects.all(), pk=pk)
        deal.delete()
        return Response(status=201)

class FabricatorView(APIView):
    """Просмотр производителей"""
    permission_classes = [permissions.IsAuthenticated, ]

    def get(self, request):

```

```

        fabricators = Fabricator.objects.all()
        serializer = FabricatorSerializers(fabricators,
many=True)
        return Response({"data": serializer.data})

class ProductCRUDView(APIView):
    """CRUD товаров"""
    permission_classes = [permissions.IsAuthenticated, ]

    def get(self, request):
        fabricator = request.GET.get("fabricator")
        if fabricator == None:
            products = Product.objects.all()
            serializer = ProductSerializers(products, many=True)
            return Response({"data": serializer.data})
        else:
            product =
Product.objects.filter(fabricator=fabricator)
            serializer = ProductSerializers(product, many=True)
            return Response({"data": serializer.data})

    def post(self, request):
        product = ProductCRUDSerializers(data=request.data)
        if product.is_valid():
            fabricator = get_object_or_404(Fabricator,
id=self.request.data.get("fabricator"))
            product.save(fabricator=fabricator)
            return Response(status=201)
        else:
            return Response(status=400)

    def put(self, request, pk):
        saved_product = get_object_or_404(Product.objects.all(),
pk=pk)
        serializer =
ProductCRUDSerializers(instance=saved_product, data=request.data,
partial=True)
        if serializer.is_valid(raise_exception=True):
            serializer.save()
            return Response(status=201)

    def delete(self, request, pk):
        product = get_object_or_404(Product.objects.all(), pk=pk)
        product.delete()
        return Response(status=201)

class OrderDealView(APIView):
    """Просмотр заказов сделки"""
    permission_classes = [permissions.IsAuthenticated, ]

```

```

def get(self, request):
    deal = request.GET.get("deal", -1)
    if deal == -1:
        order_deals = OrderDeal.objects.all()
        serializer = OrderDealSerializers(order_deals,
many=True)
        return Response({"data": serializer.data})
    else:
        order_deals = OrderDeal.objects.filter(deal=deal)
        serializer = OrderDealSerializers(order_deals,
many=True)
        return Response({"data": serializer.data})

```

*Листинг Б.7 – perfume\_app/urls*

```

from django.urls import path
from perfume_app.views import *

urlpatterns = [
    path('broker/', BrokerView.as_view()),
    path('firm/', FirmView.as_view()),
    path('deal/', DealCRUDView.as_view()), # post и get запросы
    path('deal/<int:pk>', DealCRUDView.as_view()), # put и delete
запросы
    path('fabricator/', FabricatorView.as_view()),
    path('product/', ProductCRUDView.as_view()), # post и get
запросы
    path('product/<int:pk>', ProductCRUDView.as_view()), # put и
delete запросы
    path('order_deal/', OrderDealView.as_view()),
]

```

*Листинг Б.8 – perfume/perfume-vue/src/App*

```

<template>
<!-- <div id="app">-->
    <router-view/>
<!-- </div>-->
</template>

<script>
export default {
  name: 'App'
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;

```

```

    color: #2c3e50;
    margin-top: 60px;
}
</style>

```

*Листинг Б.9 – perfume-vue/src/router/index*

```

import Vue from 'vue'
import Router from 'vue-router'
import Home from '@components/Home'
import Login from '@components/Login'
import Form from '@components/Form'

Vue.use(Router)

export default new Router({
  routes: [
    {
      path: '/',
      name: 'home',
      component: Home
    },
    {
      path: '/login',
      name: 'login',
      component: Login
    },
    {
      path: '/form/:model',
      name: 'form',
      component: Form
    }
  ]
})

```

*Листинг Б.10 – perfume-vue/src/components/Home*

```

<template>
  <div>
    <mu-appbar style="width: 100%;text-align: center"
color="deepPurple500">
      Парфюмерный базар
      <mu-button flat slot="right" v-if="!auth"
@click="goLogin">Вход</mu-button>
      <mu-button flat slot="right" v-else
@click="logout">Выход</mu-button>
    </mu-appbar>
    <mu-container>
      <mu-row>
        <Menu v-if="auth"></Menu>
      </mu-row>
      <mu-row>

```

```

        <mu-card v-if="!auth" style="width: 100%;text-align:
center;margin: 100px">
            <mu-card-title title="Для дальнейшего взаимодействия
войдите в профиль..."></mu-card-title>
        </mu-card>
    </mu-row>
</mu-container>
</div>
</template>

```

```

<script>
    import Menu from '@components/Menu';

    export default {
        name: "Home",
        components: {
            Menu
        },
        computed: {
            auth() {
                if (localStorage.getItem("auth_token")) {
                    return true
                }
            }
        },
        methods: {
            goLogin() {
                this.$router.push({name: "login"})
            },
            logout() {
                localStorage.removeItem("auth_token");
                window.location = '/'
            },
        }
    }
</script>

```

```
<style scoped>
```

```
</style>
```

*Листинг Б.11 – perfume-vue/src/components/Login*

```

<template>
    <div>
        <mu-appbar style="width: 100%;text-align: center"
color="deepPurple500">
            Вход в систему
            <mu-button flat slot="left" @click="mainWindow">
                <mu-icon value="home"></mu-icon>
            </mu-button>
        </mu-appbar>
    </div>
</template>

```

```

    <mu-container style="text-align: center">
      <mu-form ref="form" :model="validateForm" style="padding:
100px">
        <mu-form-item label="Логин" help-text="Введите имя
пользователя" prop="username" :rules="usernameRules">
          <mu-text-field v-model="validateForm.username"
prop="username"></mu-text-field>
        </mu-form-item>
        <mu-form-item label="Пароль" help-text="Введите пароль"
prop="password" :rules="passwordRules">
          <mu-text-field v-model="validateForm.password"
prop="password"
                        :action-icon="visibility" ?
'visibility_off' : 'visibility'
                        :action-click="() => (this.visibility =
!visibility)"
                        :type="visibility ? 'text' :
'password'"></mu-text-field>
        </mu-form-item>
        <mu-form-item style="align-items: center">
          <mu-button color="deepPurple500"
@click="setLogin">Войти</mu-button>
          <mu-button @click="clear">Сбросить</mu-button>
        </mu-form-item>
      </mu-form>
    </mu-container>
  </div>
</template>

<script>
  export default {
    name: "Login",
    data() {
      return {
        validateForm: {
          username: '',
          password: '',
        },
        visibility: false,
        usernameRules: [
          {validate: (val) => !!val, message: 'Имя пользователя
должно быть заполнено'},
        ],
        passwordRules: [
          {validate: (val) => !!val, message: 'Пароль должен быть
заполнен'},
        ],
      }
    },
    methods: {
      setLogin() {
        this.$refs.form.validate().then((result) => {

```

```

        if (result === false) {
            alert('Какие-то поля заполнены неверно.');
```

```

        } else {
            $.ajax({
                url: "http://127.0.0.1:8000/auth/token/login/",
                type: "POST",
                data: {
                    username: this.validateForm.username,
                    password: this.validateForm.password,
                },
                success: (response) => {
                    alert("Вход в систему");
                    localStorage.setItem("auth_token",
response.data.attributes.auth_token);
                    this.$router.push({name: "home"})
                },
                error: (response) => {
                    if (response.status === 400) {
                        alert("Неправильный логин или пароль")
                    }
                }
            })
        }
    })
},
clear() {
    this.$refs.form.clear();
    this.validateForm = {
        username: '',
        password: '',
    };
},
mainWindow() {
    this.$router.push({name: "home"})
},
}
}
</script>

<style scoped>
    .mu-demo-form {
        width: 100%;
        max-width: 460px;
    }
</style>

```

*Листинг Б.12 – perfume-vue/src/components/Menu*

```

<template>
    <mu-paper style="width: 100%;margin-top: 40px" :z-depth="5">
        <mu-list textline="two-line">

```

```

        <mu-list-item v-for="map in choices" :key="map[0]" button
:ripple="false" @click="openItem(map[3])">
            <mu-list-item-action>
                <mu-icon v-model="map[2]" color="deepPurple500"></mu-
icon>
            </mu-list-item-action>
            <mu-list-item-content>
                <mu-list-item-title>{{map[0]}}</mu-list-item-title>
                <mu-list-item-sub-title>{{map[1]}}</mu-list-item-sub-
title>
            </mu-list-item-content>
        </mu-list-item>
    </mu-list>
</mu-paper>
</template>

<script>
    export default {
        name: "Menu",
        data() {
            return {
                choices: '',
            },
        },
        created() {
            this.loadMenu()
        },
        methods: {
            loadMenu() {
                this.choices = [
                    ['Маклеры', 'Просмотр маклеров', 'assignment_ind',
'broker'],
                    ['Фирмы', 'Просмотр фирм', 'work', 'firm'],
                    ['Сделки', 'Редактирование сделок', 'local_atm',
'deal'],
                    ['Товары', 'Редактирование товаров', 'whatshot',
'product'],
                    ['Производители', 'Просмотр производителей', 'group',
'fabricator'],
                ];
            },
            openItem(model) {
                this.$router.push({name: 'form', params: {model: model}})
            }
        }
    }
</script>

<style scoped>

</style>

```



```

<template>
  <div>
    <mu-appbar style="width: 100%;text-align: center"
color="deepPurple500">
      {{name}}
      <mu-button flat slot="left" @click="mainWindow">
        <mu-icon value="home"></mu-icon>
      </mu-button>
      <mu-button v-if="this.$route.params.model === 'product' ||
this.$route.params.model === 'deal'" flat
        slot="right" @click="post">
        <mu-icon value="add"></mu-icon>
        запись
      </mu-button>
    </mu-appbar>
    <mu-alert color="deepPurple50" delete v-if="isOpen"
@delete="closeAlert">
      <mu-icon left value="info"></mu-icon>
      <div v-bind:style="{color:'Indigo'}">Для более подробной
информации выберите интересующий элемент и
      раскройте его.
    </div>
    </mu-alert>
    <mu-row>
      <Broker v-if="this.$route.params.model === 'broker'" v-
bind:items="items"
        @giveName="giveName"></Broker>
      <Firm v-if="this.$route.params.model === 'firm'" v-
bind:items="items"
        @giveName="giveName"></Firm>
      <Deal v-if="this.$route.params.model === 'deal'" v-
bind:items="items"
        @giveName="giveName" @getPanels="getPanels"></Deal>
      <Product v-if="this.$route.params.model === 'product'" v-
bind:items="items"
        @giveName="giveName"
@getPanels="getPanels"></Product>
      <Fabricator v-if="this.$route.params.model ===
'fabricator'" v-bind:items="items"
        @giveName="giveName"></Fabricator>
    </mu-row>
    <mu-row v-if="this.$route.params.model === 'product'">
      <CreateProduct v-bind:openAlert="open" @reload="getPanels"
@closePost="closePost"
        ref="openProduct"></CreateProduct>
    </mu-row>
    <mu-row v-if="this.$route.params.model === 'deal'">
      <CreateDeal v-bind:openAlert="open" @reload="getPanels"
@closePost="closePost" ref="openDeal"></CreateDeal>
    </mu-row>
  </div>

```

```

    </div>
</template>

<script>
import Broker from "@/components/Broker";
import Firm from "@/components/Firm";
import Deal from "@/components/Deal";
import Product from "./Product";
import Fabricator from "./Fabricator";
import EditProduct from "./CRUD/EditProduct";
import EditDeal from "./CRUD/EditDeal";
import CreateDeal from "./CRUD/CreateDeal";
import CreateProduct from "./CRUD/CreateProduct";

export default {
  name: "Form",
  components: {
    CreateProduct,
    CreateDeal,
    EditDeal,
    EditProduct,
    Fabricator,
    Firm,
    Broker,
    Deal,
    Product,
  },
  props: {
    model: '',
  },
  data() {
    return {
      items: '',
      name: '',
      isOpen: true,
      open: false
    }
  },
  created() {
    $.ajaxSetup({
      headers: { 'Authorization': 'Token ' +
localStorage.getItem('auth_token') },
    });
    this.getPanels();
  },
  methods: {
    giveName(name) {
      this.name = name;
    },
    closeAlert() {
      this.isOpen = false
    },
  },

```

```

    mainWindow() {
      this.$router.push({name: "home"})
    },
    getPanels() {
      $.ajax({
        url: "http://127.0.0.1:8000/api/v1/perfume/" +
this.$route.params.model + '/',
        type: "GET",
        data: {},
        success: (response) => {
          this.items = response.data.data
          // console.log(response)
        },
        error: (response) => {
          alert("Проблемы с подключением к серверу. Повторите
попытку позже.")
        }
      })
    },
    post() {
      if (this.$route.params.model === 'deal')
this.$refs.openDeal.openAlertDialog();
      else this.$refs.openProduct.openAlertDialog();
      this.open = true;
    },
    closePost() {
      this.open = false;
    },
  }
}
</script>

<style scoped>

</style>

```

*Листинг Б.14 – perfume-vue/src/components/Broker*

```

<template>
  <div style="width: 100%">
    <mu-row v-for="item in items" :key="item.id">
      <mu-expansion-panel style="width: 100%;text-align: left;">
        <div slot="header">{{item.first_name}}
{{item.second_name}}</div>
        Адрес: {{item.address}}
        <p>Дата рождения: {{item.birthdate}}</p>
        <mu-expansion-panel v-if="item.broker" style="width:
100%;">
          <div slot="header">Профиль</div>
          Никнейм: {{item.broker.username}}
          <p>Почта: {{item.broker.email}}</p>

```

```

        </mu-expansion-panel>
    </mu-expansion-panel>
</mu-row>
</div>
</template>

<script>
    export default {
        name: "Broker",
        props: {
            items: ''
        },
        created() {
            this.$emit('giveName', 'Просмотр маклеров');
        }
    }
</script>

<style scoped>

</style>

```

*Листинг Б.15 – perfume-vue/src/components/Fabricator*

```

<template>
    <div style="width: 100%">
        <mu-row v-for="item in items" :key="item.id">
            <mu-expansion-panel style="width: 100%;text-align: left;">
                <div slot="header">{{item.name_fabricator}}</div>
                <p>Страна: {{item.country}}</p>
                <p>Юридический адрес: {{item.legal_address}}</p>
            </mu-expansion-panel>
        </mu-row>
    </div>
</template>

<script>
    export default {
        name: "Fabricator",
        props: {
            items: ''
        },
        created() {
            this.$emit('giveName', 'Просмотр производителей');
        },
    }
</script>

<style scoped>

</style>

```

*Листинг Б.16 – perfume-vue/src/components/Firm*

```
<template>
  <div style="width: 100%">
    <mu-row v-for="item in items" v-bind:key="item.id">
      <mu-expansion-panel style="width: 100%;text-align: left;">
        <div slot="header">{{item.name_firm}}</div>
        Страна: {{item.country}}
        <p>Юридический адрес: {{item.legal_address}}</p>
        <p v-text="'Тип фирмы: '+typeFirm(item.type_firm) "></p>
      </mu-expansion-panel>
    </mu-row>
  </div>
</template>

<script>
  export default {
    name: "Firm",
    props: {
      items: ''
    },
    created() {
      this.$emit('giveName', 'Просмотр фирм');
    },
    methods: {
      typeFirm(firm) {
        return firm === 'S' ? 'Seller' : 'Buyer'
      },
    }
  }
</script>

<style scoped>

</style>
```

*Листинг Б.17 – perfume-vue/src/components/Order*

```
<template>
  <div style="width: 100%">
    <mu-button color="deepPurple500" style="margin-top: 10px"
    @click="openFullscreenDialog">
      <mu-icon value="add"></mu-icon>
      Открыть заказы
    </mu-button>
    <mu-dialog width="360" transition="slide-bottom" fullscreen
    :open.sync="openFullscreen">
      <mu-appbar color="deepPurple500" title="Заказы">
        <mu-button slot="left" icon
        @click="closeFullscreenDialog">
          <mu-icon value="close"></mu-icon>
        </mu-button>
      </mu-appbar>
    </mu-dialog>
  </div>
</template>
```

```

    </mu-appbar>
    <mu-expansion-panel v-if="dealItem" v-for="item in
dealItem" :key="dealItem.id">
      <div slot="header">{{item.product.name}}</div>
      Количество: {{item.number_sold}}
      <p>Цена продавца: {{item.cost_seller}} руб.</p>
      <p>Цена маклера: {{item.cost_broker}} руб.</p>
      <mu-expansion-panel>
        <div slot="header">Информация о продукте</div>
        <p v-text="'Тип: '+typePerfume(item.product.type)"></p>
        <p>Срок годности: {{item.product.shelf_life}}
год(a)</p>
        <p v-text="'Пол: '+typeSex(item.product.sex)"></p>
      </mu-expansion-panel>
    </mu-expansion-panel>
    <mu-alert color="deepPurpleA100" v-if="dealItem.length ===
0" style="margin-top: 20px;">
      <mu-icon left value="info"></mu-icon>
      Пока что заказов для данной сделки нет.
    </mu-alert>
  </mu-dialog>
</div>
</template>

<script>
export default {
  name: "Order",
  props: {
    dealId: '',
  },
  data() {
    return {
      openFullscreen: false,
      dealItem: ''
    };
  },
  methods: {
    openFullscreenDialog() {
      this.openFullscreen = true;
      $.ajax({
        url:
"http://127.0.0.1:8000/api/v1/perfume/order_deal/",
        type: "GET",
        data: {
          "deal": this.dealId
        },
        success: (response) => {
          this.dealItem = response.data.data;
        },
        error: (response) => {
          alert("Произошла ошибка. Повторите попытку
позже...");

```

```

        this.openFullscreen = false;
    }
    });
},
closeFullscreenDialog() {
    this.openFullscreen = false;
},
typePerfume(type) {
    switch (type) {
        case 'PA':
            return 'Parfum';
        case 'EP':
            return 'Eau de Parfum';
        case 'ET':
            return 'Eau de Toilette';
        case 'EC':
            return 'Eau de Cologne';
        case 'EF':
            return 'Eau Fraiche';
    }
},
typeSex(sex) {
    if (sex === 'F') return 'Female';
    else if (sex === 'M') return 'Male';
    else return 'Unisex'
}
},
}
</script>

```

<style scoped>

</style>

*Листинг Б.18 – perfume-vue/src/components/Deal*

```

<template>
  <div style="width: 100%">
    <mu-row v-for="item in items" :key="item.id">
      <Remove v-bind:id="item.id" v-bind:model="'deal'"
@itemDelete="reload"></Remove>
      <EditDeal v-bind:item="item" @reload="reload"></EditDeal>
      <span style="width: 92%">
        <mu-expansion-panel style="width: 100%;text-align:
left;">
          <div slot="header">{{item.date_deal}}</div>
          <mu-expansion-panel style="width: 100%;margin-
top: 10px">
            <div slot="header">Маклер</div>
            Имя: {{item.broker.first_name}}
            <p>Фамилия: {{item.broker.second_name}}</p>
            <p>Адрес: {{item.broker.address}}</p>
          </mu-expansion-panel>
        </mu-expansion-panel>
      </span>
    </mu-row>
  </div>
</template>

```

```

        <p>Дата рождения:
{{item.broker.birthdate}}</p>
        <mu-expansion-panel v-if="item.broker.broker"
style="width: 100%;">
            <div slot="header">Профиль</div>
            Никнейм: {{item.broker.broker.username}}
            <p>Почта:
{{item.broker.broker.email}}</p>
        </mu-expansion-panel>
    </mu-expansion-panel>
    <mu-expansion-panel
        v-for="firms in massive={buyer: {info:
item.buyer, name: 'Покупатель'}, seller: {info: item.seller, name:
'Продавец'}}"
        :key="firms.info.id + ' firm'">
        <div slot="header">{{firms.name}}</div>
        Название фирмы: {{firms.info.name_firm}}
        <p>Страна: {{firms.info.country}}</p>
        <p>Юридический адрес:
{{firms.info.legal_address}}</p>
        <p v-text="'Тип фирмы:
'+typeFirm(firms.info.type_firm)'"></p>
    </mu-expansion-panel>
    <Order v-bind:dealId="item.id"></Order>
</mu-expansion-panel>
</span>
</mu-row>
</div>
</template>

<script>
import Order from "./Order";
import Remove from "./CRUD/Remove";
import EditDeal from "./CRUD/EditDeal";

export default {
    name: "Deal",
    components: {EditDeal, Remove, Order},
    props: {
        items: '',
    },
    created() {
        this.$emit('giveName', 'Редактирование сделок');
    },
    methods: {
        typeFirm(firm) {
            return firm === 'S' ? 'Seller' : 'Buyer'
        },
        reload() {
            this.$emit('getPanels',);
        }
    }
}

```





```

    },
    methods: {
      typePerfume(type) {
        switch (type) {
          case 'PA':
            return 'Parfum';
          case 'EP':
            return 'Eau de Parfum';
          case 'ET':
            return 'Eau de Toilette';
          case 'EC':
            return 'Eau de Cologne';
          case 'EF':
            return 'Eau Fraiche';
        }
      },
      typeSex(sex) {
        if (sex === 'F') return 'Female';
        else if (sex === 'M') return 'Male';
        else return 'Unisex'
      },
      reload() {
        this.$emit('getPanels',);
      },
    }
  }
</script>

```

<style scoped>

</style>

*Листинг Б.20 – perfume-vue/src/components/CRUD/CreateDeal*

```

<template>
  <div>
    <mu-dialog title="Создание сделки" width="400" max-
width="80%" :esc-press-close="false"
      :overlay-close="false" :open.sync="openAlert">
      <mu-form :model="form" :label-position="labelPosition">
        <mu-date-input icon="today" v-model="form.date" label-
float full-width value-format="YYYY-MM-DD"
          :date-time-format="endDateFormat"
color="deepPurple800" display-color="deepPurple800">
        </mu-date-input>
        <!-- {{form.date}}-->
        <mu-form-item>
          <mu-button color="deepPurple800" @click="seller">
            <mu-icon value="add"></mu-icon>
            Продавец
          </mu-button>
          <mu-button color="deepPurple800" @click="buyer">

```

```

        <mu-icon value="add"></mu-icon>
        Покупатель
    </mu-button>
    <Picker      v-bind:openFullscreen="openFullscreen"      v-
bind:items="firm"

@closeFullscreenDialog="closeFullscreenDialog"></Picker>
    </mu-form-item>
    <mu-form-item>
        <mu-text-field      v-text="'Продавец:
'+names.sellerName"></mu-text-field>
        <mu-text-field      v-text="'Покупатель:
'+names.buyerName"></mu-text-field>
    </mu-form-item>
    </mu-form>
    <mu-button      slot="actions"      flat      color="deepPurple800"
@click="closeAlertDialog">Отменить</mu-button>
    <mu-button      slot="actions"      flat      color="deepPurple800"
@click="closeSaveAlertDialog">Создать</mu-button>
    </mu-dialog>
</div>
</template>

<script>
    const dayAbbreviation = ['Вс', 'Пн', 'Вт', 'Ср', 'Чт', 'Пт',
'Сб'];
    const dayList = ['Вс', 'Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб'];
    const monthList = ['Янв', 'Февр', 'Март', 'Апр', 'Май', 'Июнь',
'Июль', 'Авг', 'Сент',
    'Окт', 'Нояб', 'Дек'];
    const monthLongList = ['Январь', 'Февраль', 'Март', 'Апрель',
'Май', 'Июнь',
    'Июль', 'Август', 'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь'];

    const enDateFormat = {
        formatDisplay(date) {
            return ` ${dayList[date.getDay()]},
${monthList[date.getMonth()]} ${date.getDate()}`;
        },
        formatMonth(date) {
            return ` ${monthLongList[date.getMonth()]}
${date.getFullYear()}`;
        },
        getWeekDayArray(firstDayOfWeek) {
            let beforeArray = [];
            let afterArray = [];
            for (let i = 0; i < dayAbbreviation.length; i++) {
                if (i < firstDayOfWeek) {
                    afterArray.push(dayAbbreviation[i]);
                } else {
                    beforeArray.push(dayAbbreviation[i]);
                }
            }
        }
    }

```

```

    }
    return beforeArray.concat(afterArray);
  },
  getMonthList() {
    return monthList;
  }
};
import Picker from "../Picker";

export default {
  name: "CreateDeal",
  components: {
    Picker
  },
  props: {
    item: '',
    openAlert: false,
  },
  data() {
    return {
      labelPosition: 'left',
      form: {
        date: '',
        sellerId: '',
        buyerId: ''
      },
      names: {
        sellerName: '',
        buyerName: '',
      },
      items: '',
      firm: [],
      openFullscreen: false,
      endDateFormat
    };
  },
  methods: {
    openAlertDialog() {
      $.ajax({
        url: "http://127.0.0.1:8000/api/v1/perfume/firm/",
        type: "GET",
        data: {},
        success: (response) => {
          this.items = response.data.data;
          // console.log(response)
        },
        error: (response) => {
          alert("Проблемы с подключением к серверу. Повторите попытку позже.")
        }
      })
    }
  }
},

```

```

closeAlertDialog() {
    this.clear();
    this.$emit('closePost',);
},
closeSaveAlertDialog() {
    if (!this.checkValidation()) {
        alert('Какие-то поля не заполнены.');
```

попытку позже.");

```

    } else {
        $.ajax({
            url: "http://127.0.0.1:8000/api/v1/perfume/deal/",
            type: "POST",
            data: {
                date_deal: this.form.date,
                seller: this.form.sellerId,
                buyer: this.form.buyerId
            },
            success: (response) => {
                this.$emit('reload',);
                this.clear();
                this.$emit('closePost',);
            },
            error: (response) => {
                alert("Проблемы с созданием элемента. Повторите
                // this.clear();
                // this.$emit('closePost',);
            }
        })
    }
},
seller() {
    var items2 = [];
    for (var element of this.items) {
        if (element['type_firm'] === 'S')
            items2.splice(0, 0, element)
    }
    this.firm = items2;
    this.openFullscreen = true;
},
buyer() {
    var items2 = [];
    for (var element of this.items) {
        if (element['type_firm'] === 'B')
            items2.splice(0, 0, element)
    }
    this.firm = items2;
    this.openFullscreen = true;
},
closeFullscreenDialog(openFullscreen, id, type, name) {
    this.openFullscreen = openFullscreen;
    if (name !== undefined) {
        if (type === 'S') {

```

```

        this.form.sellerId = id;
        this.names.sellerName = name
      } else {
        this.form.buyerId = id;
        this.names.buyerName = name
      }
    }
  },
  checkValidation() {
    if (this.form.buyerId    &&    this.form.sellerId    &&
this.form.date)
      return true;
    else return false
  },
  clear() {
    this.form = {
      date: '',
      sellerId: '',
      buyerId: ''
    };
    this.names = {
      sellerName: '',
      buyerName: '',
    };
  }
},
}
</script>

<style scoped>

</style>

```

*Листинг Б.21 – perfume-vue/src/components/CRUD/CreateProduct*

```

<template>
  <div>
    <mu-dialog    title="Создание    товара"    width="400"    max-
width="80%" :esc-press-close="false"
      :overlay-close="false" :open.sync="openAlert">
      <mu-form    :model="form"    :label-position="labelPosition"
ref="form">
        <mu-form-item    prop="name"    label="Название    товара"
:rules="lineRule">
          <mu-text-field                                v-model="form.name"
color="deepPurple800"></mu-text-field>
        </mu-form-item>
        <mu-form-item prop="type" label="Тип" :rules="lineRule">
          <mu-select v-model="form.type">
            <mu-option    v-for="(option,index)    in    options"
:key="index" :label="option"
              :value="option"></mu-option>

```

```

        </mu-select>
      </mu-form-item>
      <mu-form-item prop="shelf_life" label="Срок годности"
:rules="numberRule">
        <mu-text-field v-model.number="form.shelf_life"
color="deepPurple800" type="number" min="1"
max="30"></mu-text-field>
      </mu-form-item>
      <mu-form-item prop="sex" label="Пол" :rules="lineRule">
        <mu-select v-model="form.sex">
          <mu-option v-for="(option,index) in sex" :key="index"
:label="option"
:value="option"></mu-option>
        </mu-select>
      </mu-form-item>
      <mu-form-item>
        <mu-button color="deepPurple800" @click="fabric">
          <mu-icon value="add"></mu-icon>
          Производитель
        </mu-button>
        <PickerFabric v-bind:openFullscreen="openFullscreen" v-
bind:items="items"

@closeFullscreenDialog="closeFullscreenDialog"></PickerFabric>
      </mu-form-item>
      <mu-form-item>
        <mu-text-field v-text="'Производитель:
'+nameFabric"></mu-text-field>
      </mu-form-item>
    </mu-form>
    <mu-button slot="actions" flat color="deepPurple800"
@click="closeAlertDialog">Отменить</mu-button>
    <mu-button slot="actions" flat color="deepPurple800"
@click="closeSaveAlertDialog">Создать</mu-button>
  </mu-dialog>
</div>
</template>

<script>
  import PickerFabric from "./PickerFabric";

  export default {
    name: "CreateProduct",
    components: {PickerFabric},
    props: {
      item: '',
      openAlert: false,
    },
    data() {
      return {
        options: [

```

```

        'Parfum', 'Eau de Parfum', 'Eau de Toilette', 'Eau de
Cologne',
        'Eau Fraiche',
    ],
    sex: ['Male', 'Female', 'Unisex'],
    labelPosition: 'left',
    form: {
        name: '',
        type: '',
        shelf_life: '',
        sex: '',
        fabricantId: '',
    },
    nameFabric: '',
    items: '',
    openFullscreen: false,
    lineRule: [
        {validate: (val) => !!val, message: 'Поле должно быть
заполнено'}],
    ],
    numberRule: [
        {validate: (val) => !!val, message: 'Поле должно быть
заполнено'}],
        {validate: (val) => Number.isInteger(val), message:
'Поле должно быть целочисленным'}],
    ],
    littleToBig: [['PA', 'EP', 'ET', 'EC', 'EF'],
        ['Parfum', 'Eau de Parfum', 'Eau de Toilette', 'Eau de
Cologne', 'Eau Fraiche']
    ]
    }
    },
    methods: {
        openAlertDialog() {
            $.ajax({
                url:
"http://127.0.0.1:8000/api/v1/perfume/fabricator/",
                type: "GET",
                data: {},
                success: (response) => {
                    this.items = response.data.data;
                    // console.log(response)
                },
                error: (response) => {
                    alert("Проблемы с подключением к серверу. Повторите
попытку позже.");
                    // this.clear();
                    // this.$emit('closePost', );
                }
            })
        },
        closeAlertDialog() {

```



```

        this.clear();
        this.$emit('closePost',);
    },
    closeSaveAlertDialog() {
        this.$refs.form.validate().then((result) => {
            if (result === false || this.nameFabric === '') {
                alert('Какие-то поля заполнены неверно или не
заполнены.');
```

заполнены.');

```

            } else {
                $.ajax({
                    url:
"http://127.0.0.1:8000/api/v1/perfume/product/",
                    type: "POST",
                    data: {
                        name: this.form.name,
                        type: this.typeRevertPerfume(this.form.type),
                        shelf_life: this.form.shelf_life,
                        sex: this.typeRevertSex(this.form.sex),
                        fabricator: this.form.fabricatorId,
                    },
                    success: (response) => {
                        this.$emit('reload',);
                        this.clear();
                        this.$emit('closePost',);
                    },
                    error: (response) => {
                        alert("Проблемы с созданием элемента. Повторите
попытку позже.");
                        // this.clear();
                        // this.$emit('closePost', );
                    }
                })
            }
        });
    },
    fabric() {
        this.openFullscreen = true;
    },
    closeFullscreenDialog(openFullscreen, id, name) {
        this.openFullscreen = openFullscreen;
        if (name !== undefined) {
            this.form.fabricatorId = id;
            this.nameFabric = name;
        }
    },
    typePerfume(type) {
        switch (type) {
            case 'PA':
                return 'Parfum';
            case 'EP':
                return 'Eau de Parfum';
        }
    }
}

```

```

        case 'ET':
            return 'Eau de Toilette';
        case 'EC':
            return 'Eau de Cologne';
        case 'EF':
            return 'Eau Fraiche';
    }
},
typeSex(sex) {
    if (sex === 'F') return 'Female';
    else if (sex === 'M') return 'Male';
    else return 'Unisex'
},
typeRevertPerfume(type) {
    switch (type) {
        case 'Parfum':
            return 'PA';
        case 'Eau de Parfum':
            return 'EP';
        case 'Eau de Toilette':
            return 'ET';
        case 'Eau de Cologne':
            return 'EC';
        case 'Eau Fraiche':
            return 'EF';
    }
},
typeRevertSex(sex) {
    if (sex === 'Female') return 'F';
    else if (sex === 'Male') return 'M';
    else return 'U'
},
clear() {
    this.$refs.form.clear();
    this.form = {
        name: '',
        type: '',
        shelf_life: '',
        sex: '',
        fabricantId: '',
    };
    this.nameFabric = '';
}
}
}
</script>

<style scoped>

</style>

```

*Листинг Б.22 – perfume-vue/src/components/CRUD/EditDeal*

```

<template>
  <div>
    <mu-button icon color="deepPurple800"
@click="openAlertDialog">
      <mu-icon value="edit"></mu-icon>
    </mu-button>
    <mu-dialog title="Редактор сделки" width="400" max-
width="80%" :esc-press-close="false"
:overlay-close="false" :open.sync="openAlert">
      <mu-form :model="form" :label-position="labelPosition">
        <mu-date-input icon="today" v-model="form.date" label-
float full-width value-format="YYYY-MM-DD"
:date-time-format="enDateFormat"
color="deepPurple800" display-color="deepPurple800">
        </mu-date-input>
        <mu-form-item>
          <mu-button color="deepPurple800" @click="seller">
            <mu-icon value="add"></mu-icon>
            Продавец
          </mu-button>
          <mu-button color="deepPurple800" @click="buyer">
            <mu-icon value="add"></mu-icon>
            Покупатель
          </mu-button>
          <Picker v-bind:openFullscreen="openFullscreen" v-
bind:items="firm"
@closeFullscreenDialog="closeFullscreenDialog"></Picker>
        </mu-form-item>
        <mu-form-item>
          <mu-text-field v-text="'Продавец:
'+names.sellerName"></mu-text-field>
          <mu-text-field v-text="'Покупатель
'+names.buyerName"></mu-text-field>
        </mu-form-item>
      </mu-form>
      <mu-button slot="actions" flat color="deepPurple800"
@click="closeAlertDialog">Отменить</mu-button>
      <mu-button slot="actions" flat color="deepPurple800"
@click="closeSaveAlertDialog">Изменить</mu-button>
    </mu-dialog>
  </div>
</template>

<script>
  const dayAbbreviation = ['Вс', 'Пн', 'Вт', 'Ср', 'Чт', 'Пт',
'Сб'];
  const dayList = ['Вс', 'Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб'];
  const monthList = ['Янв', 'Февр', 'Март', 'Апр', 'Май', 'Июнь',
'Июль', 'Авг', 'Сент',
'Окт', 'Нояб', 'Дек'];

```

```

    const monthLongList = ['Январь', 'Февраль', 'Март', 'Апрель',
    'Май', 'Июнь',
    'Июль', 'Август', 'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь'];

    const enDateFormat = {
      formatDisplay(date) {
        return ` ${dayList[date.getDay()]},
    ${monthList[date.getMonth()]} ${date.getDate()}`;
      },
      formatMonth(date) {
        return ` ${monthLongList[date.getMonth()]}
    ${date.getFullYear()}`;
      },
      getWeekDayArray(firstDayOfWeek) {
        let beforeArray = [];
        let afterArray = [];
        for (let i = 0; i < dayAbbreviation.length; i++) {
          if (i < firstDayOfWeek) {
            afterArray.push(dayAbbreviation[i]);
          } else {
            beforeArray.push(dayAbbreviation[i]);
          }
        }
        return beforeArray.concat(afterArray);
      },
      getMonthList() {
        return monthList;
      }
    };
import Picker from "../Picker";

export default {
  name: "EditDeal",
  components: {
    Picker
  },
  props: {
    item: '',
  },
  data() {
    return {
      openAlert: false,
      labelPosition: 'left',
      form: {
        date: this.item.date_deal,
        sellerId: this.item.seller.id,
        buyerId: this.item.buyer.id,
      },
      names: {
        sellerName: this.item.seller.name_firm,
        buyerName: this.item.buyer.name_firm,
      },
    },
  },

```

```

        items: '',
        firm: [],
        openFullscreen: false,
        endDateFormat
    };
},
methods: {
    openAlertDialog() {
        this.openAlert = true;
        $.ajax({
            url: "http://127.0.0.1:8000/api/v1/perfume/firm/",
            type: "GET",
            data: {},
            success: (response) => {
                this.items = response.data.data
                // console.log(response)
            },
            error: (response) => {
                alert("Проблемы с подключением к серверу. Повторите
попытку позже.")
            }
        })
    },
    closeAlertDialog() {
        this.form.date = this.item.date_deal;
        this.form.sellerId = this.item.seller.id;
        this.form.buyerId = this.item.buyer.id;
        this.openAlert = false;
    },
    closeSaveAlertDialog() {
        $.ajax({
            url: "http://127.0.0.1:8000/api/v1/perfume/deal/" +
this.item.id,
            type: "PUT",
            data: {
                date_deal: this.form.date,
                seller: this.form.sellerId,
                buyer: this.form.buyerId
            },
            success: (response) => {
                // console.log(response);
                this.$emit('reload',);
                this.openAlert = false;
            },
            error: (response) => {
                alert("Проблемы с изменением элемента. Повторите
попытку позже.");
                this.openAlert = false;
            }
        })
    },
    seller() {

```

```

        var items2 = [];
        for (var element of this.items) {
            if (element['type_firm'] === 'S')
                items2.splice(0, 0, element)
        }
        this.firm = items2;
        this.openFullscreen = true;
        // console.log(this.firm)
    },
    buyer() {
        var items2 = [];
        for (var element of this.items) {
            if (element['type_firm'] === 'B')
                items2.splice(0, 0, element)
        }
        this.firm = items2;
        this.openFullscreen = true;
        // console.log(this.firm)
    },
    closeFullscreenDialog(openFullscreen, id, type, name) {
        this.openFullscreen = openFullscreen;
        if (name !== undefined) {
            if (type === 'S') {
                this.form.sellerId = id;
                this.names.sellerName = name
            } else {
                this.form.buyerId = id;
                this.names.buyerName = name
            }
        }
    }
},
}
</script>

```

<style scoped>

</style>

### *Листинг Б.23 – perfume-vue/src/components/CRUD/EditProduct*

```

<template>
  <div>
    <mu-button icon color="deepPurple800"
    @click="openAlertDialog">
      <mu-icon value="edit"></mu-icon>
    </mu-button>
    <mu-dialog title="Редактор товара" width="400" max-
    width="80%" :esc-press-close="false"
      :overlay-close="false" :open.sync="openAlert">
      <mu-form :model="form" :label-position="labelPosition"
      ref="form">

```

```

        <mu-form-item prop="name" label="Название товара"
:rules="lineRule">
            <mu-text-field v-model="form.name"
color="deepPurple800"></mu-text-field>
        </mu-form-item>
        <mu-form-item prop="select" label="Тип">
            <mu-select v-model="form.type">
                <mu-option v-for="(option,index) in options"
:key="index" :label="option"
:value="option"></mu-option>
            </mu-select>
        </mu-form-item>
        <mu-form-item prop="shelf_life" label="Срок годности"
:rules="numberRule">
            <mu-text-field v-model.number="form.shelf_life"
color="deepPurple800" type="number" min="1"
max="30"></mu-text-field>
        </mu-form-item>
        <mu-form-item prop="select" label="Пол">
            <mu-select v-model="form.sex">
                <mu-option v-for="(option,index) in sex" :key="index"
:label="option"
:value="option"></mu-option>
            </mu-select>
        </mu-form-item>
        <mu-form-item>
            <mu-button color="deepPurple800" @click="fabric">
                <mu-icon value="add"></mu-icon>
                Производитель
            </mu-button>
            <PickerFabric v-bind:openFullscreen="openFullscreen" v-
bind:items="items"

@closeFullscreenDialog="closeFullscreenDialog"></PickerFabric>
        </mu-form-item>
        <mu-form-item>
            <mu-text-field v-text="'Производитель:
'+nameFabric"></mu-text-field>
        </mu-form-item>
    </mu-form>
    <mu-button slot="actions" flat color="deepPurple800"
@click="closeAlertDialog">Отменить</mu-button>
    <mu-button slot="actions" flat color="deepPurple800"
@click="closeSaveAlertDialog">Изменить</mu-button>
</mu-dialog>
</div>
</template>

<script>
    import PickerFabric from "../PickerFabric";

    export default {

```

```

name: "EditProduct",
components: {PickerFabric},
props: {
  item: '',
},
data() {
  return {
    options: [
      'Parfum', 'Eau de Parfum', 'Eau de Toilette', 'Eau de
Cologne',
      'Eau Fraiche',
    ],
    sex: ['Male', 'Female', 'Unisex'],
    openAlert: false,
    labelPosition: 'left',
    form: {
      name: this.item.name,
      type: this.typePerfume(this.item.type),
      shelf_life: this.item.shelf_life,
      sex: this.typeSex(this.item.sex),
      fabricantId: this.item.fabricator.id,
    },
    nameFabric: this.item.fabricator.name_fabricator,
    items: '',
    openFullscreen: false,
    lineRule: [
      {validate: (val) => !!val, message: 'Поле должно быть
заполнено'}],
    numberRule: [
      {validate: (val) => !!val, message: 'Поле должно быть
заполнено'},
      {validate: (val) => Number.isInteger(val), message:
'Поле должно быть целочисленным'}],
    littleToBig: [['PA', 'EP', 'ET', 'EC', 'EF'],
      ['Parfum', 'Eau de Parfum', 'Eau de Toilette', 'Eau de
Cologne', 'Eau Fraiche']]
  ]
},
methods: {
  openAlertDialog() {
    this.openAlert = true;
    $.ajax({
      url:
"http://127.0.0.1:8000/api/v1/perfume/fabricator/",
      type: "GET",
      data: {},
      success: (response) => {
        this.items = response.data.data;
        // console.log(response)
      }
    })
  }
}

```



```

        },
        error: (response) => {
            alert("Проблемы с подключением к серверу. Повторите
попытку позже.")
        }
    })
},
closeAlertDialog() {
    this.form.name = this.item.name;
    this.form.type = this.typePerfume(this.item.type);
    this.form.shelf_life = this.item.shelf_life;
    this.form.sex = this.typeSex(this.item.sex);
    this.form.fabricatorId = this.item.fabricator.id;
    this.openAlert = false;
},
closeSaveAlertDialog() {
    this.$refs.form.validate().then((result) => {
        if (result === false) {
            alert('Какие-то поля заполнены неверно.');
```

```

        this.nameFabric = name;
    }
},
typePerfume(type) {
    switch (type) {
        case 'PA':
            return 'Parfum';
        case 'EP':
            return 'Eau de Parfum';
        case 'ET':
            return 'Eau de Toilette';
        case 'EC':
            return 'Eau de Cologne';
        case 'EF':
            return 'Eau Fraiche';
    }
},
typeSex(sex) {
    if (sex === 'F') return 'Female';
    else if (sex === 'M') return 'Male';
    else return 'Unisex'
},
typeRevertPerfume(type) {
    switch (type) {
        case 'Parfum':
            return 'PA';
        case 'Eau de Parfum':
            return 'EP';
        case 'Eau de Toilette':
            return 'ET';
        case 'Eau de Cologne':
            return 'EC';
        case 'Eau Fraiche':
            return 'EF';
    }
},
typeRevertSex(sex) {
    if (sex === 'Female') return 'F';
    else if (sex === 'Male') return 'M';
    else return 'U'
},
    }
}
</script>

<style scoped>

</style>

```

*Листинг Б.24 – perfume-vue/src/components/CRUD/Picker*

```
<template>
```

```

<div>
  <mu-dialog width="360" transition="slide-bottom" fullscreen
:open.sync="openFullscreen">
    <mu-appbar color="deepPurple500" title="Фирмы">
      <mu-button slot="left" icon
@click="closeFullscreenDialog">
        <mu-icon value="close"></mu-icon>
      </mu-button>
    </mu-appbar>
    <mu-row v-for="firm in this.items" :key="firm.id">
      <mu-expansion-panel style="width: 100%;text-align:
left;">
        <div slot="header">{{firm.name_firm}}</div>
        Страна: {{firm.country}}
        <p>Юридический адрес: {{firm.legal_address}}</p>
        <mu-button color="deepPurple500"

@click="closeFullscreenDialog(firm.id,firm.type_firm,firm.name_f
irm)">Выбрать
      </mu-button>
    </mu-expansion-panel>
  </mu-row>
</mu-dialog>
</div>
</template>

<script>
  export default {
    name: "Picker",
    props: {
      items: '',
      openFullscreen: false
    },
    methods: {
      closeFullscreenDialog(id, type, name) {
        this.$emit('closeFullscreenDialog', false, id, type,
name);
      }
    }
  }
</script>

<style scoped>

</style>

```

*Листинг Б.25 – perfume-vue/src/components/CRUD/PickerFabric*

```

<template>
  <div>
    <mu-dialog width="360" transition="slide-bottom" fullscreen
:open.sync="openFullscreen">

```

```

        <mu-appbar color="deepPurple500" title="Производители">
            <mu-button slot="left" icon
@click="closeFullscreenDialog">
                <mu-icon value="close"></mu-icon>
            </mu-button>
        </mu-appbar>
        <mu-row v-for="fabric in items" :key="fabric.id">
            <mu-expansion-panel style="width: 100%;text-align:
left;">
                <div slot="header">{{fabric.name_fabricator}}</div>
                <p>Страна: {{fabric.country}}</p>
                <p>Юридический адрес: {{fabric.legal_address}}</p>
                <mu-button color="deepPurple500"

@click="closeFullscreenDialog(fabric.id,fabric.name_fabricator)"
>Выбрать
            </mu-button>
        </mu-expansion-panel>
    </mu-row>
</mu-dialog>
</div>
</template>

<script>
    export default {
        name: "PickerFabric",
        props: {
            items: '',
            openFullscreen: false
        },
        methods: {
            closeFullscreenDialog(id, name) {
                this.$emit('closeFullscreenDialog', false, id, name);
            }
        }
    }
</script>

<style scoped>

</style>

```

*Листинг Б.26 – perfume-vue/src/components/CRUD/Remove*

```

<template>
    <div>
        <mu-button icon color="deepPurple800" @click="signalDelete"
delete v-if="remove">
            <mu-icon value="delete"></mu-icon>
        </mu-button>
    </div>
</template>

```

```

<script>
  export default {
    name: "Remove",
    props: {
      model: '',
      id: ''
    },
    data() {
      return {
        remove: true
      }
    },
    methods: {
      signalDelete() {
        $.ajax({
          url: "http://127.0.0.1:8000/api/v1/perfume/" +
this.$route.params.model + '/' + this.id,
          type: "DELETE",
          data: {},
          success: (response) => {
            this.remove = false;
            this.$emit('itemDelete',);
          },
          error: (response) => {
            alert("Не удалось произвести удаление. Повторите
попытку позже.")
          }
        })
      }
    }
  }
}
</script>

<style scoped>

</style>

```