

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**  
**по дисциплине «Введение в нереляционные базы данных»**  
**Тема: Сервис аренды инструментов для ремонта**

Студент гр. 1384

Корякин А.И.

Студент гр. 1384

Овчинников М.Ю.

Студентка гр. 1384

Шиняева А.А.

Преподаватель

Заславский М.М.

Санкт-Петербург

2024

**ЗАДАНИЕ**  
**НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**

Студент Корякин А.С.

Студент Овчинников М.Ю.

Студентка Шиняева А.А.

Группа 1384

Тема задания: Сервис аренды инструментов для ремонта

Исходные данные:

Необходимо создать веб-приложение для аренды ремонтных инструментов с использованием СУБД MongoDB.

Содержание пояснительной записи:

«Содержание»

«Введение»

«Сценарии использования»

«Модель данных»

«Разработанное приложение»

«Выводы»

«Приложения»

«Список использованных источников»

Предполагаемый объем пояснительной записи:

Не менее 20 страниц.

Дата выдачи задания: 05.09.2024

Дата сдачи реферата: 23.12.2024

Дата защиты реферата: 23.12.2024

Студент

---

Корякин А.И.

Студент

---

Овчинников М.Ю.

Студентка

---

Шиняева А.А.

Преподаватель

---

Заславский М.М.

## **АННОТАЦИЯ**

В рамках ИДЗ было разработано веб-приложение для аренды ремонтных инструментов. Приложение предоставляет функции поиска инструментов с возможностью фильтрации, просмотра подробной информации и отзывов о выбранных инструментах, а также создания заказов. Дополнительно реализована административная панель, которая позволяет добавлять новые инструменты, просматривать список заказов и пользовательских отзывов с возможностью их фильтрации.

Приложение создано с использованием следующих технологий: Vue.js, Python, MongoDB и Docker.

Исходный код доступен по ссылке: <https://github.com/moevm/nosql2h24-tools>

## **SUMMARY**

As part of the individual assignment, a web application for renting repair tools was developed. The application provides functionality for searching tools with filtering options, viewing detailed information and reviews about selected tools, and creating orders. Additionally, an admin panel was implemented, allowing the addition of new tools, viewing the list of orders and user reviews with filtering options.

The application was developed using the following tools: Vue.js, Python, MongoDB, and Docker.

The source code is available at the following link:  
<https://github.com/moevm/nosql2h24-tools>

## СОДЕРЖАНИЕ

1.	Введение	6
1.1	Актуальность проблемы	6
1.2	Постановка задачи	6
1.3	Предлагаемое решение	7
1.4	Качественные требования к решению	7
2.	Сценарии использования	8
2.1	Макет UI	8
2.2	Сценарии использования для импорта данных	20
2.3	Сценарии использования для предоставления данных	23
2.4	Сценарии использования для анализа данных	27
2.5	Сценарии использования для экспорта данных	27
2.6	Выводы	28
3.	Модель данных	29
3.1	Нереляционная модель данных	29
3.2	Реляционная модель данных	39
3.3	Сравнение моделей	44
4.	Разработанное приложение	45
4.1	Краткое описание приложения	45
4.2	Использованные технологии	46
4.3	Схема экранов приложения	47
5.	Выводы	48
5.1	Достигнутые результаты	48
5.2	Недостатки и пути для улучшения	48
5.3	Будущее развитие решения	50
6.	Приложения	51
6.1	Документация по сборке и развертыванию приложения	51
6.2	Инструкция для пользователя	51
7.	Литература	55

# **1. ВВЕДЕНИЕ**

## **1.1 Актуальность проблемы**

В современных условиях строительные и ремонтные работы часто требуют использования дорогостоящего специализированного оборудования, которое экономически нецелесообразно приобретать для разового использования. Для таких случаев аренда инструментов становится оптимальным решением, позволяя минимизировать затраты и сократить время на выполнение задач. Традиционные методы, такие как телефонные звонки или посещение пунктов проката, требуют значительных временных затрат и не всегда позволяют получить полную информацию об инструментах, их доступности или стоимости. Разработка веб-приложения для аренды инструментов может значительно улучшить этот процесс, предоставляя пользователям возможность быстро находить необходимое оборудование, изучать его характеристики, отзывы других арендаторов и оформлять заказы в удобное время. Кроме того, приложение позволяет автоматизировать управление инструментами и заказами для сотрудников, делая процесс аренды более удобным и эффективным. Дополнительно приложение предоставляет возможность эффективно собирать и анализировать статистику об арендах, популярности инструментов и пользовательской активности, что способствует дальнейшему улучшению сервиса.

## **1.2 Постановка задачи**

Задача проекта заключается в разработке веб-приложения, которое объединяет удобный пользовательский интерфейс для арендаторов и гибкие инструменты управления для работников. Приложение должно предоставлять следующие функции:

- Поиск инструментов с возможностью фильтрации по категориям, типам, цене и другим параметрам
- Просмотр карточки инструмента с описанием, фотографиями, характеристиками и отзывами

- Удобное оформление заказа с отслеживанием статуса
- Административную панель для управления списком инструментов, просмотра списка заказов и отзывов с возможностью фильтрации по различным параметрам.

### **1.3 Предполагаемое решение**

Приложение будет создано с использованием Vue.js для клиентской части, Python для серверной части, MongoDB для хранения данных и Docker для контейнеризации и развертывания.

### **1.4 Качественные требования к решению**

Приложение должно предоставлять интуитивный интерфейс для пользователей и работников, обеспечивать удобные механизмы поиска и фильтрации, быть высоко производительным, надежным и легко масштабируемым, а также должно быть предусмотрено быстрое развертывания приложения на различных платформах для упрощения внедрения и поддержки.

## 2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

### 2.1 Макет UI

Макет пользовательского интерфейса представлен ниже (рис. 1 - 15).

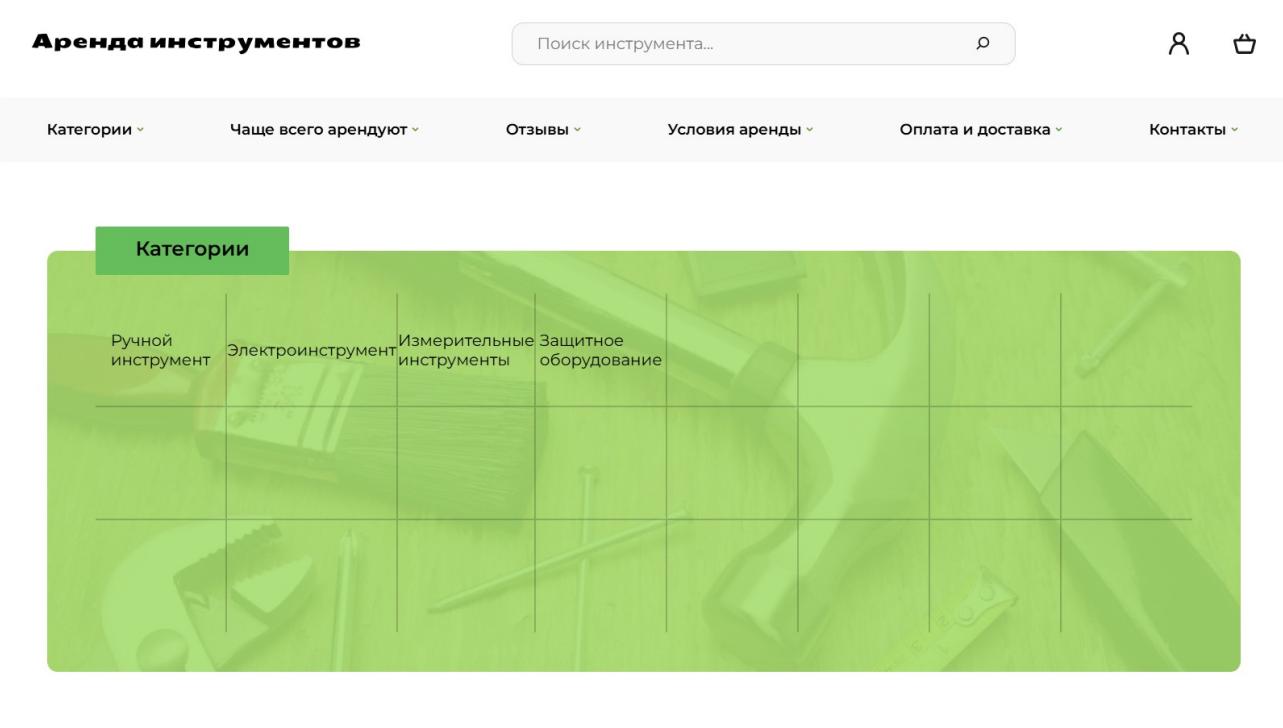


Рисунок 1 - Макет главной страницы

## Инструменты категории: «Электроинструмент»

**Тип** Скрыть

- Дрель
- Болгарка
- Перфоратор

**Цена**

0 - 100000

**Применить фильтры**

### Дрель ударная Bosch GSB 13 RE

Компактная ударная дрель от Bosch для работы с бетоном, деревом и металлом. Эргономичная ручка для комфортной работы.

★★★★★

300Р

**Просмотр**

### Дрель DeWalt DWD024

Мощная дрель от DeWalt с высокой производительностью. Подходит для интенсивного использования на стройке.

★★★★★

350Р

**Просмотр**

### Дрель-шуруповерт Makita DF330DWE

Аккумуляторная дрель-шуруповерт от Makita. Идеальный инструмент для сборки мебели и мелких работ.

★★★★★

400Р

**Просмотр**

### Угловая шлифмашина Bosch GWS 7-125

Компактная болгарка для точной резки и шлифовки металла и камня. Эргономичный дизайн обеспечивает комфортную работу.

★★★★★

400Р

**Просмотр**

### Болгарка Makita GA5030

Удобная и мощная углашлифовальная машина, предназначенная для различных строительных и ремонтных работ.

★★★★★

350Р

**Просмотр**

### Угловая шлифмашина DeWalt DWE4051

Мощная болгарка с компактными размерами, идеальная для профессиональных работ с металлом и бетоном.

★★★★★

500Р

**Просмотр**

Рисунок 2 - Макет страницы инструментов  
одной категории

## Аренда инструментов

Поиск инструмента...



Мой профиль

Мои аренды

Редактировать профиль

Изменить пароль

Выйти



**Петров Иван**  
client\_1@example.com

Контакты: +79161234567

Рисунок 3 - Макет профиля

## Мои аренды

Фильтр по названию инструментов

Минимальная цена заказа

Максимальная цена заказа

Минимальная дата заказа

Максимальная дата заказа

**Применить фильтры**

ID	Цена	Сроки аренды	Инструменты	Способ получения	Способ оплаты	Создано
6762e490 5644d684 6bbf246f	1620	12 декабря 2024 г. - 30 декабря 2024 г.	Молоток-гвоздодер Kraftool	Доставка	Наличными	12 декабря 2024 г.
			Молоток слесарный Matrix			
6762f8074 7f9d9502 5a19268	275	18 декабря 2024 г. - 25 декабря 2024 г.	Молоток столярный Stanley	Доставка	Наличными	18 декабря 2024 г.
			Ключ рожковый 10x12 мм Sparta			

Рисунок 4 - Макет страницы с арендами

[Мой профиль](#)
[Мои аренды](#)
[Редактировать профиль](#)
[Изменить пароль](#)
[Выйти](#)

### Редактировать профиль

Загрузите новое изображение профиля

Файл не выбран

Имя

Фамилия

Телефон

Электронная почта

**Обновить данные**

Рисунок 5 - Макет страницы редактирования профиля

[Мой профиль](#)

[Мои аренды](#)

[Редактировать профиль](#)

[Изменить пароль](#)

[Выйти](#)

### Изменить пароль

Текущий пароль

Текущий пароль

Подтвердите пароль

**Обновить пароль**

Рисунок 6 - Макет страницы изменения пароля профиля

Аренда инструментов

×
👤
📄

**Результаты поиска по запросу «молоток»**

**Категория**

Ручной инструмент

Электроинструмент

Измерительные инструменты

Защитное оборудование

**Тип**

Молоток

Отвертка

Рожковый ключ

Дрель

Болгарка

Перфоратор

Рулетка

Уровень

Очки защитные

Каска

Перчатки рабочие

Скрыть



**Молоток-гвоздодор Kraftool**

Многофункциональный молоток с гвоздодором. Подходит для легких строительных работ и ремонта.

★★★ ⚡ 40P

**Просмотр**

**Молоток слесарный Matrix**

Тяжелый молоток для выполнения слесарных работ. Удобная двухкомпонентная ручка снижает вибрацию при работе.

★★★★ ⚡ 50P

**Просмотр**

**Молоток столярный Stanley**

Легкий молоток, идеально подходящий для столярных и плотницких работ. Деревянная ручка обеспечивает комфортный захват.

★★★★ ⚡ 30P

**Просмотр**

<
1
>

Цена

-

**Применить фильтры**

Рисунок 7 - Макет страницы с результатами поиска

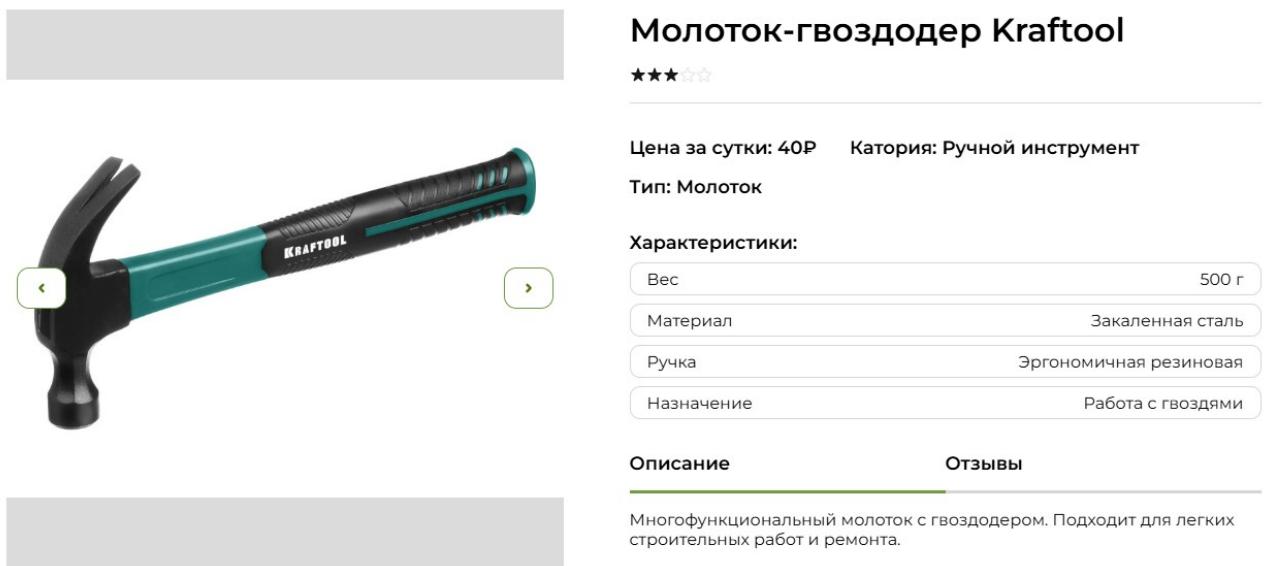


Рисунок 8 - Макет страницы инструмента

**Описание** **Отзывы**

**ПетровИван** 5★ 2024-12-19T15:22:28.835000

Очень удобный молоток-гвоздодер! Ручка не скользит, а вес идеально подходит для работы. Однако металл на гвоздодере немного мягче, чем ожидалось, и чуть согнулся после работы с твердыми гвоздями.

**СмирноваОльга** 2★ 2024-12-18T15:22:28.835000

Молоток оказался неудобным в использовании. Ручка хоть и прорезиненная, но скользит в руках при длительной работе. Гвоздодер согнулся после первой же попытки вытащить плотный гвоздь. За такую цену ожидал большего качества.

**Добавить в корзину**

Рисунок 9 - Макет списка отзывов инструмента

## Оформление заказа

### Контактные данные

Ваши контактные данные

Шаг 1 из 4

Имя

Иван

Фамилия

Петров

Электронная почта

client\_1@example.com

### Способ получения

Пожалуйста, выберите способ получения арендованных инструментов

Шаг 2 из 4

Самовывоз

Доставка

### Оплата

Пожалуйста, выберите способ оплаты

Шаг 3 из 4

Наличными (при получении)

Банковской картой

### Подтверждение

Пожалуйста, поставьте галочку

Шаг 4 из 4

Я согласен с условиями аренды

**Оформить заказ**

### Итоги заказа



Молоток-гвоздодер Kraftool

Тип: Молоток

Удалить 40 ₽

Начальная дата:

Конечная дата:

дд.мм.гггг

**Итоговая стоимость: 40 ₽**

Рисунок 10 - Макет страницы оформления заказа

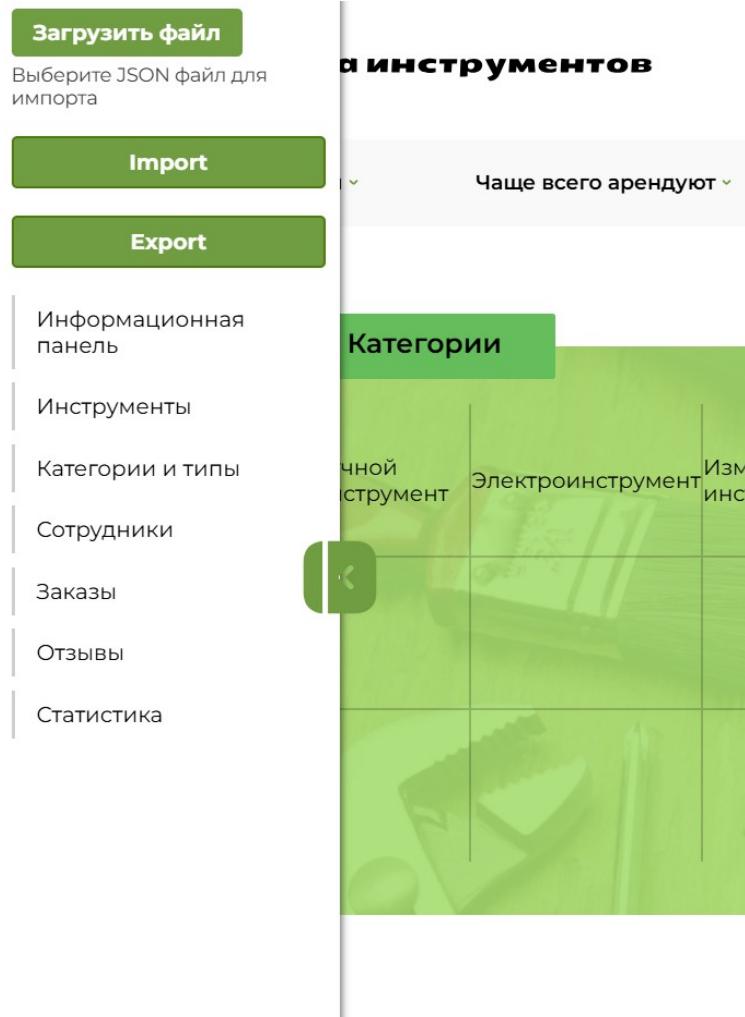


Рисунок 11 - Макет боковой панели администратора

Инструмент	Цена	Описание
Молоток-гвоздодер Kraftool	40P	Многофункциональный молоток с гвоздодером. Подходит для легких строительных работ и ремонта.
Молоток слесарный Matrix	50P	Тяжелый молоток для выполнения слесарных работ. Удобная двухкомпонентная ручка снижает вибрацию при работе.
Молоток столярный Stanley	30P	Легкий молоток, идеально подходящий для столярных и плотницких работ. Деревянная ручка обеспечивает комфортный захват.
Отвертка крестовая PH2 Vira	20P	Качественная крестовая отвертка для сборки мебели и ремонта. Удобная ручка обеспечивает надежный хват.

Рисунок 12 - Макет страницы инструментов  
для администратора

## Категории и типы

+ Добавить новую категорию

+ Добавить новый тип

Название	Типы
Ручной инструмент	Молоток Отвертка Рожковый ключ
Электроинструмент	Дрель Болгарка Перфоратор
Измерительные инструменты	Рулетка Уровень
Защитное оборудование	Очки защитные Каска Перчатки рабочие

Рисунок 13 - Макет страницы категорий и типов  
для администратора

## Сотрудники

Фильтр по имени

Имя

Фильтр по фамилии

Фамилия

Фильтр по телефону

Телефон

Фильтр по электронной почте

Электронная почта

Фильтр по должности

Должность

Применить фильтры

Имя	Фамилия	Телефон	Электронная почта	Должность	Дата присоединения
Иван	Иванов	+71234567890	worker@example.com	Главный работник	28 ноября 2024 г.

< 1 >

Рисунок 14 - Макет страницы сотрудников  
для администратора

## Заказы

Фильтр по названию инструментов      Минимальная цена заказа      Максимальная цена заказа      Минимальная дата заказа

Название инструмента

0

0

Минимальная дата заказа

дд.мм.гггг

Максимальная дата заказа

дд.мм.гггг

Фильтр по имени заказчика

Имя заказчика

Фильтр по фамилии заказчика

Фамилия заказчика

**Применить фильтры**

ID	Заказчик	Цена	Сроки аренды	Инструменты	Способ получения	Способ оплаты	Создано
	Петров Иван	1620	12 декабря 2024 г. - 30 декабря 2024 г.	Молоток-гвоздодер Kraftool Молоток слесарный Matrix	Доставка	Наличными	12 декабря 2024 г.
	Смирнова Ольга	7105	12 декабря 2024 г. - 30 января 2025 г.	Молоток-гвоздодер Kraftool Молоток слесарный Matrix Молоток столярный Stanley Ключ рожковый 10x12 мм Sparta	Доставка	Наличными	12 декабря 2024 г.

Рисунок 15 - Макет страницы заказов

для администратора

## Отзывы

Фильтр по названию инструмента      Фильтр по имени отзыва  
Название инструмента      Имя отзыва

Фильтр по фамилии отзыва      Фильтр по оценке  
Фамилия отзыва      0

Минимальная дата      Максимальная дата  
дд.мм.гггг      дд.мм.гггг

**Применить фильтры**

Имя	Фамилия	Оценка	Инструмент	Текст	Дата
Ольга	Смирнова	★★★★★ ★	Угловая шлифмашина Bosch GWS 7-125	Шлифмашина Bosch показала себя идеально! Рекомендую.	2 января 2025 г.
Алексей	Васильев	★★★★★ ☆	Перфоратор Makita HR2470	Перфоратор мощный, но немного тяжеловат для долгой работы.	2 января 2025 г.
Иван	Петров	★★★★★ ☆	Молоток столярный Stanley	Отличный столярный молоток, но немного легковат для крупных задач.	26 декабря 2024 г.
Ольга	Смирнова	★★★★★ ★	Ключ рожковый 10x12 мм Sparta	Ключ удобный, антикоррозийное покрытие работает отлично. Рекомендую!	26 декабря 2024 г.

Рисунок 16 - Макет страницы отзывов

для администратора

Регистрация      Вход      X

---

Электронная почта  
Электронная почта

Пароль  
Пароль

**Войти**

Рисунок 17 - Макет авторизации

## Регистрация

Вход

X

Имя

Имя

Фамилия

Фамилия

Электронная почта

Электронная почта

Пароль

Пароль

Повторите пароль

Повторите пароль



Соглашаюсь с политикой обработки данных и  
условиями предоставления товаров.

Зарегистрироваться

Рисунок 18 - Макет регистрации

## Статистика по заказам

Фильтрация по заказам :

Фильтр по названию инструментов      Минимальная цена заказа      Максимальная цена заказа      Минимальная дата заказа

Название инструмента      0      0      дд.мм.гггг

Максимальная дата заказа      Фильтр по имени заказчика      Фильтр по фамилии заказчика

дд.мм.гггг      Имя заказчика      Фамилия заказчика

Выбрать данные по Оси X

Выбрать данные по Оси Y

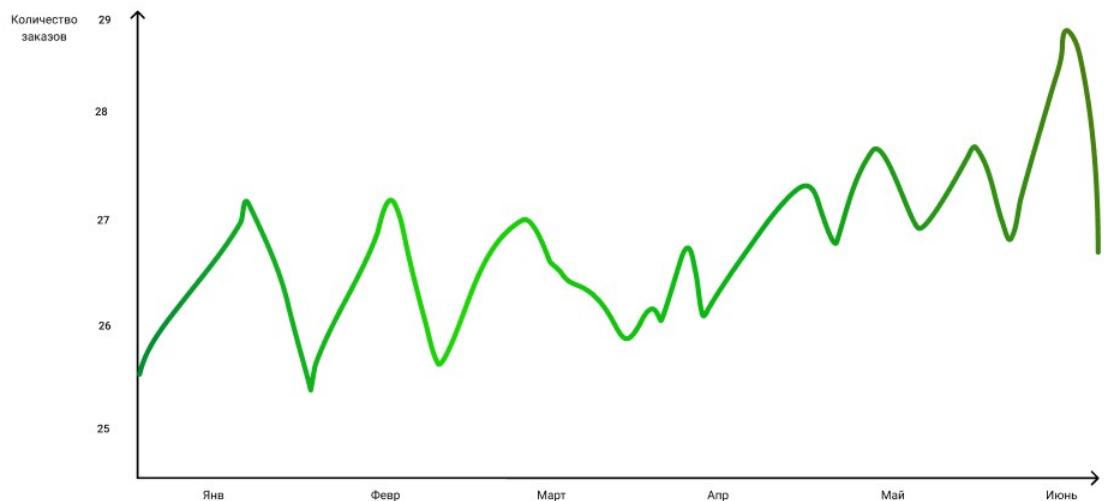


Рисунок 19 - Макет страницы со статистикой

### Выберите данные по оси Y

Количество заказов

Итоговая стоимость заказов

Количество инструментов в заказах

Продолжительность аренды в заказах

Выбрать

### Выберите данные по оси X

Статус заказа

Стоимость заказа

Дата

Дата начала

00.00.0000

Дата конца

00.00.0000

Интервал:

Месяц

Выбрать

Рисунок 20 - Макет настройки данных по осям  
для графика статистики

## **2.2 Сценарии использования для импорта данных**

### **2.2.1 Сценарий использования - «Массовый импорт данных»**

Действующее лицо: Администратор

Основной сценарий:

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на кнопку «Загрузить файл» и выбирает файл с расширением «json»
4. Пользователь нажимает на кнопку «Import»
5. Пользователь видит сообщение об успешном импорте

Альтернативный сценарий:

1. Пользователь выполняет 4 первых шага основного сценария
2. Пользователь видит сообщение об ошибке импорта из-за неподходящего файла с данными

### **2.2.2 Сценарий использования - «Добавление нового пользователя»**

Действующее лицо: Клиент

Основной сценарий:

1. Пользователь открывает сайт и попадает на главную страницу
2. Пользователь нажимает на значок профиля в шапке сайта
3. Появляется модальное окно с возможностью входа в систему или регистрации. Пользователь выбирает регистрацию
4. Пользователь вводит свои данные в поля: «Имя», «Фамилия», «Электронная почта», «Пароль», «Подтверждение пароля» и ставит галочку на согласие с обработкой персональных данных
5. Пользователь нажимает на кнопку «Зарегистрироваться»
6. Пользователь видит сообщение об успешной регистрации

Альтернативный сценарий №1:

1. Пользователь выполняет 2 первых шага основного профиля
2. Пользователь оказывается уже авторизованным в системе, так как

оказывается на странице своего профиля

3. Пользователь нажимает на «Выйти» в боковом меню
4. Пользователь выполняет шаги 3-6 основного сценария

Альтернативный сценарий №2:

1. Пользователь выполняет 5 первых шагов основного сценария
2. Пользователь видит сообщение о неудачной попытке регистрации из-за некорректных данных

### **2.2.3 Сценарий использования - «Добавление новой категории»**

Действующее лицо: Администратор

Основной сценарий:

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Категории и типы»
4. Пользователь попадает на страницу с категориями и типами и нажимает на кнопку «Добавить новую категорию»
5. Пользователь вводит данные в поле «Название категории»
6. Пользователь видит сообщение об успешном добавлении

Альтернативный сценарий:

1. Пользователь выполняет 5 первых шагов основного сценария
2. Пользователь видит сообщение о неудачной попытке добавления из-за некорректных данных

### **2.2.4 Сценарий использования - «Добавление нового типа»**

Действующее лицо: Администратор

Основной сценарий:

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Категории и типы»
4. Пользователь попадает на страницу с категориями и типами и нажимает

на кнопку «Добавить новый тип»

5. Пользователь вводит данные в поле «Название тип» и выбирает из списка категорию к которой будет принадлежать тип

6. Пользователь видит сообщение об успешном добавлении

Альтернативный сценарий:

1. Пользователь выполняет 5 первых шагов основного сценария
2. Пользователь видит сообщение о неудачной попытке добавления из-за некорректных данных

## **2.2.5 Сценарий использования - «Добавление нового инструмента»**

Действующее лицо: Администратор

Основной сценарий:

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Инструменты»
4. Пользователь попадает на страницу с инструментами и нажимает на кнопку «Добавить новый инструмент»
5. Пользователь вводит данные инструмента в поля: «Название инструмента», «Цена за аренду на 1 день», «Цена инструмента», «Изображения инструмента», «Характеристика инструмента», «Значение характеристики инструмента», «Категория», «Тип», «Описание инструмента»

6. Пользователь видит сообщение об успешном добавлении

Альтернативный сценарий:

1. Пользователь выполняет 5 первых шагов основного сценария
2. Пользователь видит сообщение о неудачной попытке добавления из-за некорректных данных

## **2.2.6 Сценарий использования - «Добавление нового заказа»**

Действующее лицо: Клиент

Основной сценарий:

1. Пользователь выполняет вход в аккаунт
2. Пользователь вводит название нужного инструмента в поисковую строку
3. Пользователь нажимает на кнопку «Арендовать» в карточке найденного инструмента
4. Пользователь попадает на страницу инструмента и нажимает на кнопку «Добавить в корзину»
5. Пользователь может повторить шаги 2-4 и добавить больше инструментов в корзину
6. Пользователь нажимает на значок корзины в шапке сайта и попадает на страницу оформления заказа
7. Пользователь вводит нужные данные в поля: «Начальная дата», «Конечная дата», «Способ получения», «Оплата» и ставит галочку с подтверждением
8. Пользователь видит сообщение об успешной

Альтернативный сценарий:

1. Пользователь выполняет 7 первых шагов основного сценария
2. Пользователь видит сообщение о неудачной попытке оформления заказа из-за некорректных данных

## **2.3 Сценарии использования для представления данных**

### **2.3.1 Сценарий использования - «Просмотр инструментов»**

Действующее лицо: Клиент

Основной сценарий:

1. Пользователь вводит в поисковую форму название инструмента и нажимает на значок поиска
2. Пользователь попадает на страницу с инструментами и видит их карточки
3. Пользователь может отфильтровать инструменты по типу, категории и

ценовому диапазону

Альтернативный сценарий:

1. Пользователь в главной странице нажимает на название любой категории в таблице
2. Пользователь оказывается в странице с инструментами, принадлежащими выбранной категории
3. Пользователь может отфильтровать инструменты по типу и ценовому диапазону

Действующее лицо: Администратор

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Инструменты» и попадает на страницу с инструментами

### **2.3.2 Сценарий использования - «Просмотр категорий»**

Действующее лицо: Клиент

1. Пользователь попадает на главную страницу и видит таблицу с категориями

Действующее лицо: Администратор

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Категории и типы»
4. Пользователь попадает на другую страницу и видит таблицу с категориями

### **2.3.3 Сценарий использования - «Просмотр типов»**

Действующее лицо: Клиент

1. Пользователь попадает на главную страницу и видит таблицу с категориями
2. Пользователь нажимает на название любой категории

3. Пользователь попадает на страницу с инструментами, принадлежащими выбранной категории и видит слева в фильтрах типы данной категории  
Действующее лицо: Администратор
5. Пользователь выполняет вход в аккаунт с правами администратора
6. Пользователь открывает боковую администраторскую панель
7. Пользователь нажимает на «Категории и типы»
8. Пользователь попадает на другую страницу и видит таблицу с типами

#### **2.3.4 Сценарий использования - «Просмотр сотрудников»**

Действующее лицо: Администратор

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Сотрудники»
4. Пользователь попадает на другую страницу и видит таблицу с категориями
5. Пользователь может отфильтровать сотрудников по имени, фамилии, телефону, электронной почте и должности

#### **2.3.5 Сценарий использования - «Просмотр данных профиля»**

Действующее лицо: Пользователь

1. Пользователь выполняет вход в свой аккаунт
2. Пользователь нажимает на значок профиля в шапке сайта
3. Пользователь попадает на страницу профиля и видит свои данные

#### **2.3.6 Сценарий использования - «Просмотр заказов»**

Действующее лицо: Клиент

1. Пользователь выполняет вход в свой аккаунт
2. Пользователь нажимает на значок профиля в шапке сайте
3. Пользователь попадает на страницу своего профиля и нажимает на «Мои аренды» в боковом меню

4. Пользователь видит таблицу своих заказов
5. Пользователь может отфильтровать свои заказы по названиям инструментов, ценовому диапазону и по диапазону дат  
Действующее лицо: Администратор
  1. Пользователь выполняет вход в аккаунт с правами администратора
  2. Пользователь открывает боковую администраторскую панель
  3. Пользователь нажимает на «Заказы»
  4. Пользователь попадает на страницу со всеми заказами и видит их в виде таблицы
  5. Пользователь может отфильтровать заказы по названиям инструментов, ценовому диапазону, диапазону дат, фамилии заказчика и по его имени

### **2.3.7 Сценарий использования - «Просмотр отзывов»**

Действующее лицо: Клиент

1. Пользователь выполняет сценарий «Просмотр инструментов» и находит инструмент, отзывы которых он хочет посмотреть
2. Пользователь нажимает на кнопку «Арендовать» и попадает на страницу инструмента
3. Пользователь нажимает на «Отзывы» и видит отзывы в виде списка

Действующее лицо: Администратор

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Отзывы»
4. Пользователь попадает на страницу со всеми отзывами и видит их в виде таблицы
5. Пользователь может отфильтровать отзывы по фамилии, имени оставившего отзыв, названию инструмента, оценке отзыва и датам

## **2.4 Сценарии использования для анализа данных**

### **2.4.1 Сценарий использования - «Просмотр статистики»**

Действующее лицо: Администратор

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на «Статистика»
4. Пользователь попадает на страницу со статистикой заказов и видит его в виде графика
5. Пользователь может отфильтровать заказы по названиям инструментов, ценовому диапазону, диапазону дат, фамилии заказчика и по его имени. Также пользователь может выбрать данные по осям для графика. По оси X можно установить статус заказа, стоимость заказа и диапазон дат. По оси Y можно установить количество заказов, итоговую стоимость заказов, количество инструментов в заказах и продолжительность аренды в заказах.

## **2.5 Сценарии использования для экспорта данных**

### **2.5.1 Сценарий использования - «Массовый импорт данных»**

Действующее лицо: Администратор

1. Пользователь выполняет вход в аккаунт с правами администратора
2. Пользователь открывает боковую администраторскую панель
3. Пользователь нажимает на кнопку «Export»
4. У пользователя скачивается файл в формате json

## **2.6 Выводы**

В системе будут преобладать операции чтения данных. Клиенты будут выполнять поиск инструментов, их фильтрацию, будут смотреть на детали определенных инструментов, отзывы на них. Также клиенты могут смотреть данные своего профиля, свои ранее сделанные заказы и их детали. Из операций записи они могут зарегистрироваться и оформить заказ.

Администраторы из операций чтения могут также делать все что могут делать клиенты, кроме оформления заказов. Возможен просмотр статистики по заказам в виде графика, фильтрация заказов для которых будет делаться статистика и настройка данных по осям. Также они могут смотреть в виде отдельных таблиц отзывы, заказы, сотрудников, инструменты, категории и типы. Также делать по отдельным полям их фильтрацию. Из операций записи администраторы могут добавлять новые инструменты, категории, типы.

### 3. МОДЕЛЬ ДАННЫХ

#### 3.1 Нереляционная модель данных

##### 3.1.1 Графическое представление

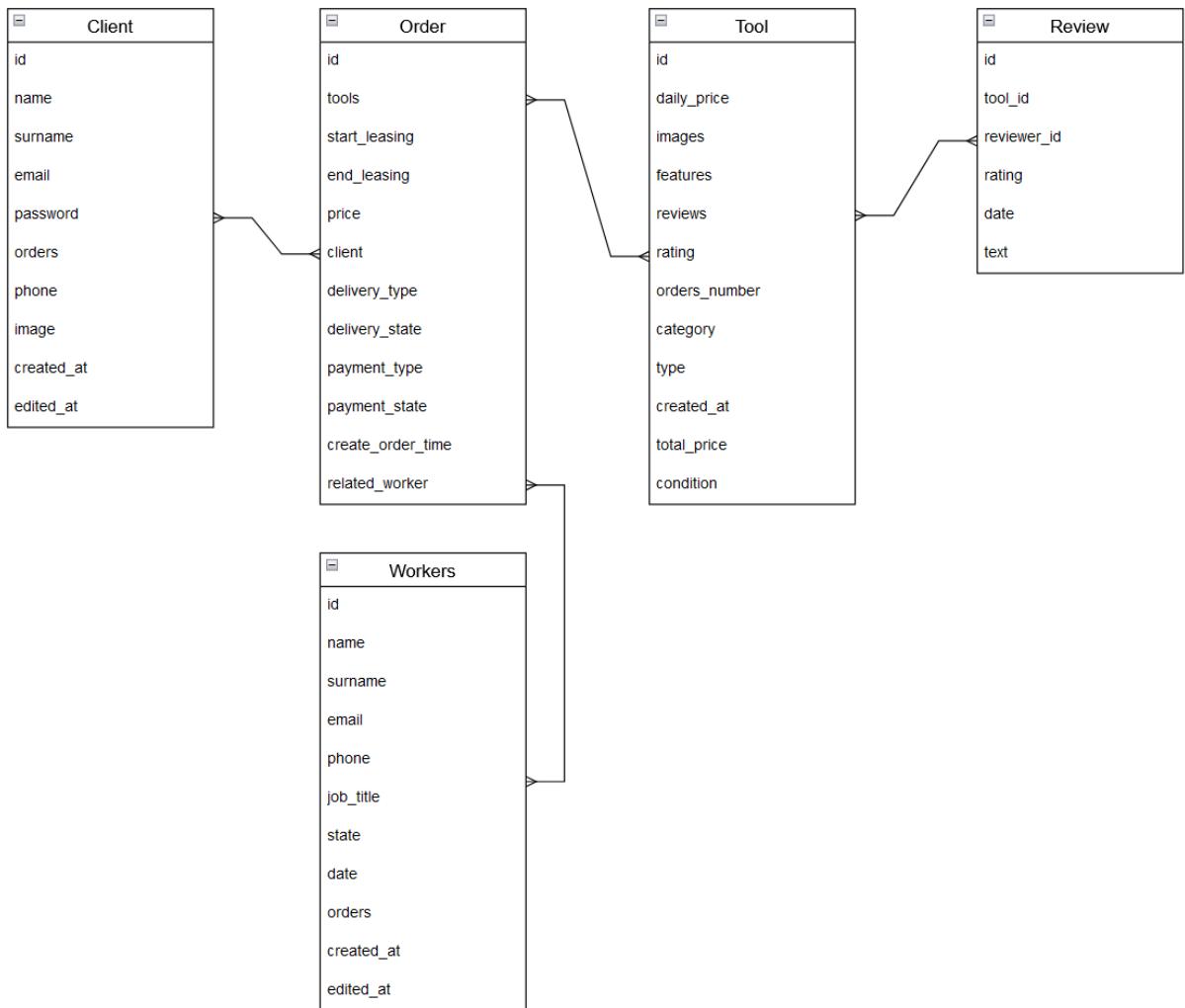


Рисунок 22 - Нереляционная модель данных

##### 3.1.2 Описание назначений коллекций, типов данных и сущностей

1. client - коллекция для хранения данных клиентов.

- \_id - уникальный идентификатор клиента;
- name - имя клиента;
- surname - фамилия клиента;
- email - почта клиента;
- password - пароль клиента;

- orders - заказы клиента;
  - phone - контактный телефон клиента;
  - image - изображение клиента в виде URI ссылки;
  - createdAt - дата создания аккаунта;
  - editedAt - дата последнего изменения аккаунта;
1. order - коллекция для хранения данных о заказах.
    - \_id - уникальный идентификатор заказа;
    - tools - список инструментов в заказе;
      - toolName - наименование инструмента;
      - toolNumber - идентификатор инструмента;
    - startLeasing - дата начала аренды;
    - endLeasing - дата окончания аренды;
    - price - цена заказа;
    - client - клиента, сделавший заказ;
      - name - имя
      - surname - фамилия
      - phone - номер телефона
    - deliveryType - тип доставки (SELF\_PICKUP, DOOR\_DELIVERY);
    - deliveryState - статус доставки (IN\_WAREHOUSE, ON\_WAY, DELIVERED, PICKED\_UP);
    - paymentType - тип оплаты (CASH, CARD, SBP);
    - paymentState - статус оплаты (NOT\_PAID, PAID, IN\_PROCESS);
    - createOrderTime - время создания заказа;
    - relatedWorker - работник, отвечающий за заказ;
      - name - имя
      - surname - фамилия
      - phone - номер телефона

2. worker - коллекция для хранения данных сотрудников.

- \_id - уникальный идентификатор работника;
- name - имя работника;

- surname - фамилия работника;
- email - почта работника;
- phone - контактный телефон работника;
- image - изображение клиента в виде URI ссылки;
- jobTitle - должность;
- state - текущий статус работника;
- date - дата начала работы;
- orders - заказы работника;
  - orderNumber - номер заказа
  - tools - инструменты в заказе
    - toolName - название инструмента
    - toolNumber - идентификатор инструмента;
- createdAt - дата создания аккаунта;
- editedAt - дата последнего изменения аккаунта;

### 3. tool - коллекция для хранения данных о инструментах.

- \_id - уникальный идентификатор инструмента;
- name - название инструмента;
- dailyprice - цена аренды инструмента за день;
- images - список изображений инструмента в бинарном виде;
- features - ассоциативный массив характеристик инструмента;
- reviews - список отзывов о инструменте;
  - reviewer - клиент, оставивший отзыв
    - name - имя
    - surname - фамилия
    - phone - номер телефона
  - rating - оставленная оценка
  - text - содержание отзыва
- rating - оценка инструмента;
- ordersNumber - количество заказов на инструмент;

- category - категория инструмента;
  - type - тип инструмента;
  - totalPrice - цена инструмента;
  - description - описание инструмента;
  - createdAt - дата добавления инструмента;
  - condition - состояние инструмента  
(LESS\_THAN\_HUNDREDUSES,  
LESS\_THAN\_THOUSANDUSES, OLD);
4. review - коллекция для хранения отзывов о инструментах.
- \_id - уникальный идентификатор отзыва;
  - tool - инструмент, на который оставлен отзыв;
    - toolName - название инструмента
    - toolId - идентификатор инструмента
  - reviewer - пользователь, оставивший отзыв;
    - name - имя
    - surname - фамилия
    - phone - номер телефона
  - rating - оценка инструмента;
  - date - дата отзыва;
  - text - текст отзыва;

### **3.1.3 Оценка удельного объема информации, хранимой в модели**

Коллекция client. Пусть на одного клиента приходится 5 заказов (в том числе неактивных). Тогда для хранения x клиентов понадобится:

$$x * (12+15+15+30+12+60+12+500000+8+8) = 500172x$$

- \_id - ObjectId V = 12 байт
- name - String V = 15 байт
- surname - String V = 15 байт
- email - String V = 30 байт

- password - String V = 12 байт
- orders - Array V = 12\*5 = 60 байт
- phone - String V = 12 байт
- image - Binary Data (предположим, что средний размер изображения составляет 500 Кб) V = 500000 байт
- createdAt - Date V = 8 байт
- editedAt - Date V = 8 байт

Коллекция order. Допустим, что у каждого клиента 2 заказа, в каждом заказе 3 инструмента. Тогда:

$$2x(12+60+8+8+8+36+12+12+12+12) = 360x$$

- \_id - ObjectId V = 12 байт
- tools - Array V = 3\*(12 + 8) = 60 байт
- startLeasing - Date V = 8 байт
- endLeasing - Date V = 8 байт
- price - Double V = 8 байт
- client - Object V = (12+12+12) = 36 байт
- deliveryType - String V = 12 байт
- deliveryState - String V = 12 байт
- paymentType - String V = 12 байт
- paymentState - String V = 12 байт

Коллекция worker. Пусть на каждого работника приходится 10 клиентов. Тогда:

$$(x/10+1) * (12+15+15+15+15+15+15+8+240+8+8) = 37x + 370$$

- \_id - ObjectId V = 12 байт
- name - String V = 15 байт
- surname - String V = 15 байт
- email - String V = 15 байт
- phone - String V = 15 байт

- jobTitle - String V = 15 байт
- state - String V = 15 байт
- date - Date V = 8 байт
- orders - Array V =  $5 * (8 + 2 * (12 + 8)) = 240$  байт
- createdAt - Date V = 8 байт
- editedAt - Date V = 8 байт

Коллекция tool. Допустим, что у каждого инструмента есть 3 характеристики, 5 картинок и 50 отзывов, а всего существует 100 инструментов. Тогда:

$$100 * (12 + 8 + 250000 + 36 + 3400 + 8 + 12 + 12 + 8 + 8 + 12) = 25351600$$

- \_id - ObjectID V = 12 байт
- dailyPrice - Double V = 8 байт
- images - Array (предположим, что средний размер изображения составляет 50 Кб) V =  $5 * 50000 = 250000$  байт
- features - Array V =  $12 * 3 = 36$  байт
- reviews - Array V =  $50 * ((12 + 12 + 12) + 8 + 24) = 3400$  байт
- rating - Double V = 8 байт
- category - String V = 12 байт
- type - String V = 12 байт
- createdAt - Date V = 8 байт
- totalPrice - Double V = 8 байт
- condition - String V = 12 байт

Коллекция review:

$$100 * (12 + 20 + 36 + 8 + 100) = 17600$$

- \_id - ObjectID V = 12 байт
- tool - Object V =  $12 + 8 = 20$  байт
- reviewer - Object V =  $12 + 12 + 12 = 36$  байт
- rating - Double V = 8 байт

- date - Date V = 8 байт
- text - String V = 100 байт

Общая оценка объема данных для x клиентов:

$$V(x) = 500172x + 360x + 37x + 370 + 25351600 + 7400 + 17600 = 500569x + 25351600$$

Для грубой оценки:

$$V(x) = 500569x$$

Избыточность данных

В БД у клиентов может не быть картинки (пусть в среднем у 1 из 2 клиентов будет картинка, тогда средний размер будет равен половине от изначального).

Тогда получаем:

$$V(x) = 250201x + 360x + 35x + 250090600 = 250296x + 25351600$$

Отношение между фактическим и "чистым" объемом (отбросив свободный член) равно

$$500569x/250296x = 1.9$$

Направление роста модели при увеличении количества объектов каждой сущности

Выразим объем БД через количество клиентов x:  $V(x) = 500596x + 250090600$ .

Виден линейный рост зависимости объема базы данных от количества клиентов.

### **3.1.4 Запросы к модели, с помощью которых реализуются сценарии использования**

- регистрация нового клиента

```
db.client.insert({
    _id: ObjectId(),
    name: "Иван",
    surname: "Иванов",
    email: "ivan@example.com",
    password: "password",
    orders: [],
    phone: "+79123456789",
    image: "someBinaryData"
})
```

- поиск аккаунта для входа в систему

```
db.client.find({
    email: "ivan@example.com",
    password: "password"
})
```

- поиск предстоящих заказов для определенного клиента

```
db.order.find({
    "client.name": "Иван",
    "client.surname": "Иванов",
    endLeasing: { $gt: new Date() }
})
```

- просмотр всех инструментов

```
db.tool.find({})
```

- создание нового заказа

```
db.order.insert({
    _id: ObjectId(),
    tools: [
        { toolName: "Дрель", toolNumber:
```

```

ObjectId("60725a34e8b57c4b3c2d9f12") } ,
{
    toolName: "Перфоратор",
    toolNumber:
ObjectId("60725a34e8b57c4b3c2d9f13")
],
startLeasing: new Date(),
endLeasing: new Date(new Date().getTime() + 3 * 24 * 60 * 60 * 1000),
price: 100,
client: {
    name: "Иван",
    surname: "Иванов",
    phone: "1234567890"
},
deliveryType: "courier",
deliveryState: "pending",
paymentType: "credit card",
paymentState: "unpaid",
createOrderTime: new Date(),
relatedWorker: {
    name: "Анна",
    surname: "Сидорова",
    phone: "0987654321"
}
)

```

- Вычисление и обновление рейтингов инструмента

```

db.tool.find().forEach(function(tool) {
    var totalRating = 0;
    var reviewCount = 0;
    db.review.find({
        toolId:
tool._id }).forEach(function(review) {

```

```

        totalRating += review.rating;
        reviewCount++;
    }) ;

    if (reviewCount > 0) {
        var newRating = totalRating / reviewCount;
        db.tool.update(
            { _id: tool._id },
            { $set: { rating: newRating } }
        );
    }
}
);

```

- ПОИСК ВСЕХ ИНСТРУМЕНТОВ, КОТОРЫЕ ЕЩЕ НЕ ВЕРНУЛИ

```

db.order.find({
    endLeasing: { $gt: new Date() }
}).forEach(function(order) {
    order.tools.forEach(function(tool) {
        printjson(tool);
    });
});

```

- ПОИСК САМЫХ ПОПУЛЯРНЫХ ИНСТРУМЕНТОВ

```

db.tool.aggregate([
    { $sort: { ordersNumber: -1 } },
    { $limit: 10 } // 10 самых популярных инструментов
]).forEach(function(tool) {
    printjson(tool);
});

```

## 3.2. Реляционная модель данных

### 3.2.1 Графическое представление

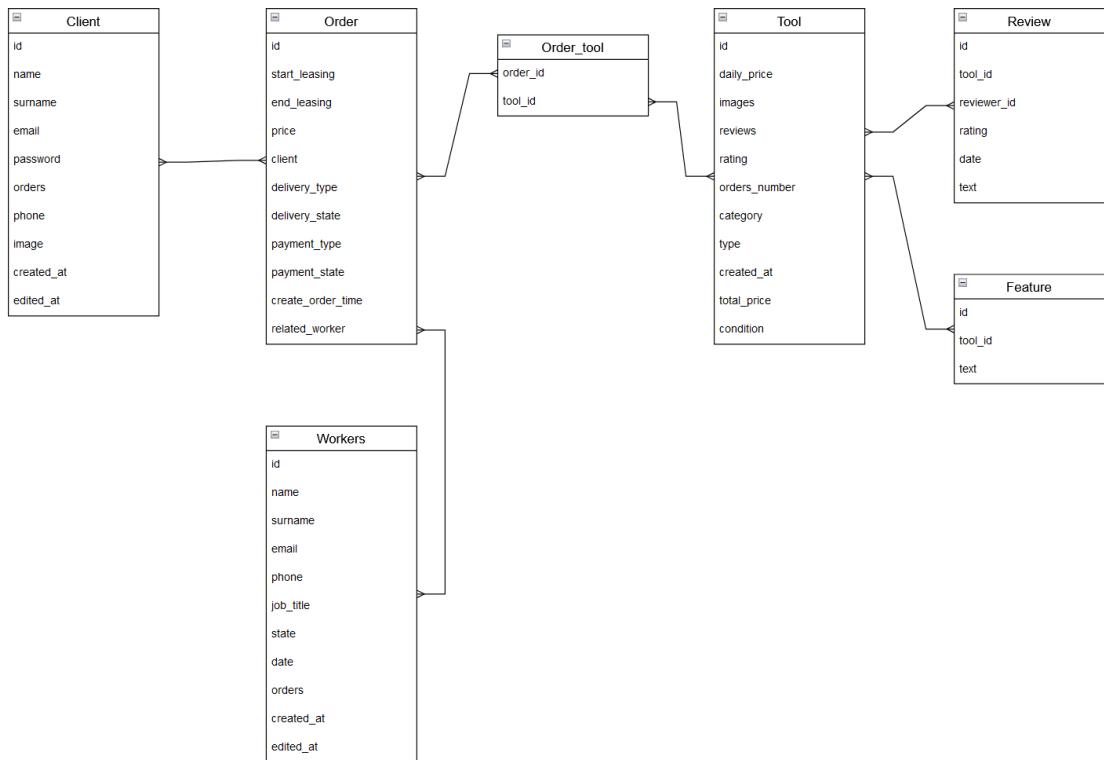


Рисунок 21 - Реляционная модель данных

### 3.2.2 Описание назначений коллекций, типов данных и сущностей

Разница от нереляционной версии:

- наличие отдельной таблицы feature для хранения характеристик инструментов
- наличие таблицы order\_tool для реализации связи между таблицами order и tool

Поля и типы данных полей идентичны.

### 3.2.3 Оценка удельного объема информации, хранимой в модели

Таблица client

Пусть x - количество клиентов, тогда:

$$V(x) = x * (8+15+15+30+12+12+500000+8+8) = 500108x$$

- \_id - INT V = 8 байт
- name - VARCHAR V = 15 байт
- surname - VARCHAR V = 15 байт
- email - VARCHAR V = 30 байт
- password - VARCHAR V = 12 байт
- contact - VARCHAR V = 12 байт
- image - VARCHAR V = 500000 байт
- createdAt - Date V = 8 байт
- editedAt - Date V = 8 байт

Таблица order

Пусть у каждого клиента будет 2 заказа. Тогда

$$V(x) = 2x(8+8+\dots) = 184x$$

- \_id - INT V = 8 байт
- startLeasing - Date V = 8 байт
- endLeasing - Date V = 8 байт
- price - Double V = 8 байт
- clientId - INT V = 8 байт
- deliveryType - VARCHAR V = 12 байт
- deliveryState - VARCHAR V = 12 байт
- paymentType - VARCHAR V = 12 байт
- paymentState - VARCHAR V = 12 байт
- workerId - INT = V = 8 байт

Таблица order\_tool

Пусть в каждом заказе будет 3 инструмента. Так как у каждого клиента по 2 заказа, то

$$V(x) = 2316*x = 96x$$

- orderId - INT = 8 байт
- toolId - INT = 8 байт

### Таблица tool

Пусть у каждого инструмента есть 3 характеристики, 5 картинок и 50 отзывов, а всего существует 100 инструментов. Тогда:

$$V(x) = 100 * (8 + 8 + 250000 + 8 + 12 + 12 + 8 + 8 + 12) = 25007600$$

- \_id - INT V = 8 байт
- dailyPrice - Double V = 8 байт
- images - Array (предположим, что средний размер изображения составляет 50 Кб) V = 5 \* 50000 = 250000 байт
- rating - Double V = 8 байт
- category - VARCHAR V = 12 байт
- type - VARCHAR V = 12 байт
- createdAt - Date V = 8 байт
- totalPrice - Double V = 8 байт
- condition - String V = 12 байт

### Таблица worker

Пусть на каждого работника приходится 10 клиентов. Тогда:

$$(x/10+1) * (12+15+15+15+15+15+15+15+8+240+8+8) = 10x + 100$$

- \_id - INT V = 8 байт
- name - VARCHAR V = 12 байт
- surname - VARCHAR V = 12 байт
- email - VARCHAR V = 12 байт
- phone - VARCHAR V = 12 байт
- jobTitle - VARCHAR V = 12 байт
- state - VARCHAR V = 12 байт

- date - Date V = 8 байт
- createdAt - Date V = 8 байт
- editedAt - Date V = 8 байт

Таблица feature

$$V(x) = 100 * 3(8+8+12) = 8400$$

- \_id - INT V = 8 байт
- toolId - INT V = 8 байт
- text - VARCHAR V = 12 байт

Таблица review

$$V(x) = 100 * 50(8+8+8+12+8+8) = 260000$$

- \_id - INT V = 8 байт
- toolId - INT V = 8 байт
- reviewerId - INT V = 8 байт
- text - VARCHAR V = 12 байт
- rating - INT V = 8 байт
- date - Date V = 8 байт

Тогда получаем общий объём:

$$V(x) = 10x + 96x + 184x + 500108x + 8400 + 260000 + 25004800 + 90 = \\ 500398x + 25273290$$

Избыточность данных

В БД у клиентов может не быть картинки (пусть в среднем у 1 из 2 клиентов будет картинка, тогда средний размер будет равен половине от изначального). Тогда получаем:

$$V(x) = 250199x + 289x + 250090600 = 250388x + 250090250$$

Отношение между фактическим и "чистым" объемом (отбросив свободный член) равно

$$500398x/250388x = 1.9$$

Направление роста модели при увеличении количества объектов каждой сущности

Объём имеет линейный рост в зависимости от количества клиентов.

### **3.2.4 Запросы к модели, с помощью которых реализуются сценарии использования**

- регистрация нового клиента

```
INSERT INTO client (_id, name, surname, email, password,
phone, image)
VALUES (NULL, 'Иван', 'Иванов', 'ivan@example.com',
'password', '+79123456789', 'someBinaryData');
```

- поиск аккаунта для входа в систему

```
SELECT * FROM client
WHERE email = 'ivan@example.com' AND password = 'password';
```

- поиск предстоящих заказов для определенного клиента

```
SELECT * FROM orders
WHERE clientId = (SELECT _id FROM client WHERE name = 'Иван'
AND surname = 'Иванов')
AND endLeasing > CURRENT_DATE;
```

- просмотр всех инструментов

```
SELECT * FROM tool;
```

- создание нового заказа

```
INSERT INTO orders (_id, startLeasing, endLeasing, price,
clientId, deliveryType, deliveryState, paymentType,
paymentState, createOrderTime, relatedWorkerId)
VALUES (NULL, CURRENT_DATE, DATE_ADD(CURRENT_DATE, INTERVAL 3
DAY), 100, (SELECT _id FROM client WHERE name = 'Иван' AND
surname = 'Иванов'), 'courier', 'pending', 'credit card',
'unpaid', CURRENT_DATE, (SELECT _id FROM worker WHERE name =
```

```
'Анна' AND surname = 'Сидорова'));  
  
INSERT INTO order_tool (orderId, toolId)  
VALUES (LAST_INSERT_ID(), GET_TOOL_ID('Дрель'));  
  
INSERT INTO order_tool (orderId, toolId)  
VALUES (LAST_INSERT_ID(), GET_TOOL_ID('Перфоратор'));
```

### 3.3. Сравнение моделей

NoSQL требует больше памяти, по сравнению с SQL, так в нем дублируются некоторые данные. SQL выигрывает по памяти, так как вместо того, чтобы хранить сами объекты целиком (как NoSQL), он хранит только id определенного элемента из другой таблицы.

По удобству запросов выигрывает NoSQL, так как ввиду дублирования данных в некоторых сущностях, нам не приходится JOIN-ить с другими коллекциями. Для некоторых запросов SQL приходится JOIN-ить несколько таблиц, что может сказать в скорости доступа.

Вывод:

NoSQL требует больше памяти, но удобен и быстр в запросах, в то время как SQL требует меньше памяти и не совсем удобен в запросах, из-за того что приходится JOIN-ить таблицы. Для данной задачи NoSQL подходит лучше, чем SQL.

## **4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ**

### **4.1 Краткое описание приложения**

Приложение использует клиент-серверную архитектуру, разделяя функциональность между фронтеном и бэкендом.

Frontend отвечает за отображение пользовательского интерфейса и взаимодействие с сервером через API. Он реализован с использованием фреймворка Vue.js, который обеспечивает создание динамического и отзывчивого интерфейса. Пользовательский интерфейс включает такие элементы как карточки, таблицы, формы ввода данных, кнопки, позволяющие пользователям просматривать, добавлять, редактировать или удалять записи. Для упрощения работы с маршрутами и переходами между страницами используется Vue Router, а взаимодействие с сервером осуществляется с помощью Axios. Кроме того, реализована валидация пользовательского ввода, чтобы предотвратить ошибки перед отправкой на сервер.

Backend реализован с использованием языка программирования Python и основан на принципах чистой архитектуры, что обеспечивает четкое разделение ответственности и модульность кода. Основным фреймворком для разработки серверной части является FastAPI, который поддерживает асинхронность и позволяет быстро и удобно создавать REST API с автоматической генерацией документации. Для взаимодействия с базой данных MongoDB используется асинхронный драйвер Motor, обеспечивающий эффективное выполнение операций. Модели данных описаны с использованием Pydantic, что позволяет одновременно задавать структуру данных, применяемую в бизнес логике, и формировать схемы для работы с базой данных. Pydantic также обеспечивает встроенную поддержку валидации данных, что минимизирует риск ошибок на этапе их обработки. Таким образом, каждая модель может выступать связующим звеном между входными данными API, бизнес-логикой и хранилищем данных, гарантируя целостность и правильность структуры. Бэкенд обрабатывает запросы от клиента, взаимодействует с базой данных и возвращает ответы, соответствующие

запросам клиента. Благодаря асинхронной природе FastAPI и Motor достигается высокая производительность.

Для удобства развертывания и управления приложением используется Docker Compose, который позволяет создавать изолированные контейнеры для всех компонентов приложения. Конфигурация Docker Compose описывает связи между сервисами и автоматизирует запуск всей системы, что позволяет избежать сложностей при настройке и развертывании на новых серверах.

## 4.2 Использованные технологии

- Frontend:
  - Vue.js
  - Vue Router
  - Axios
- Backend:
  - Python
  - FastAPI
  - Pydantic
  - Motor
- БД: MongoDB
- Контейнеризация: Docker, Docker Compose

## 4.3 Схема экранов приложения

Схема экранов приложения представлена на рис. 23

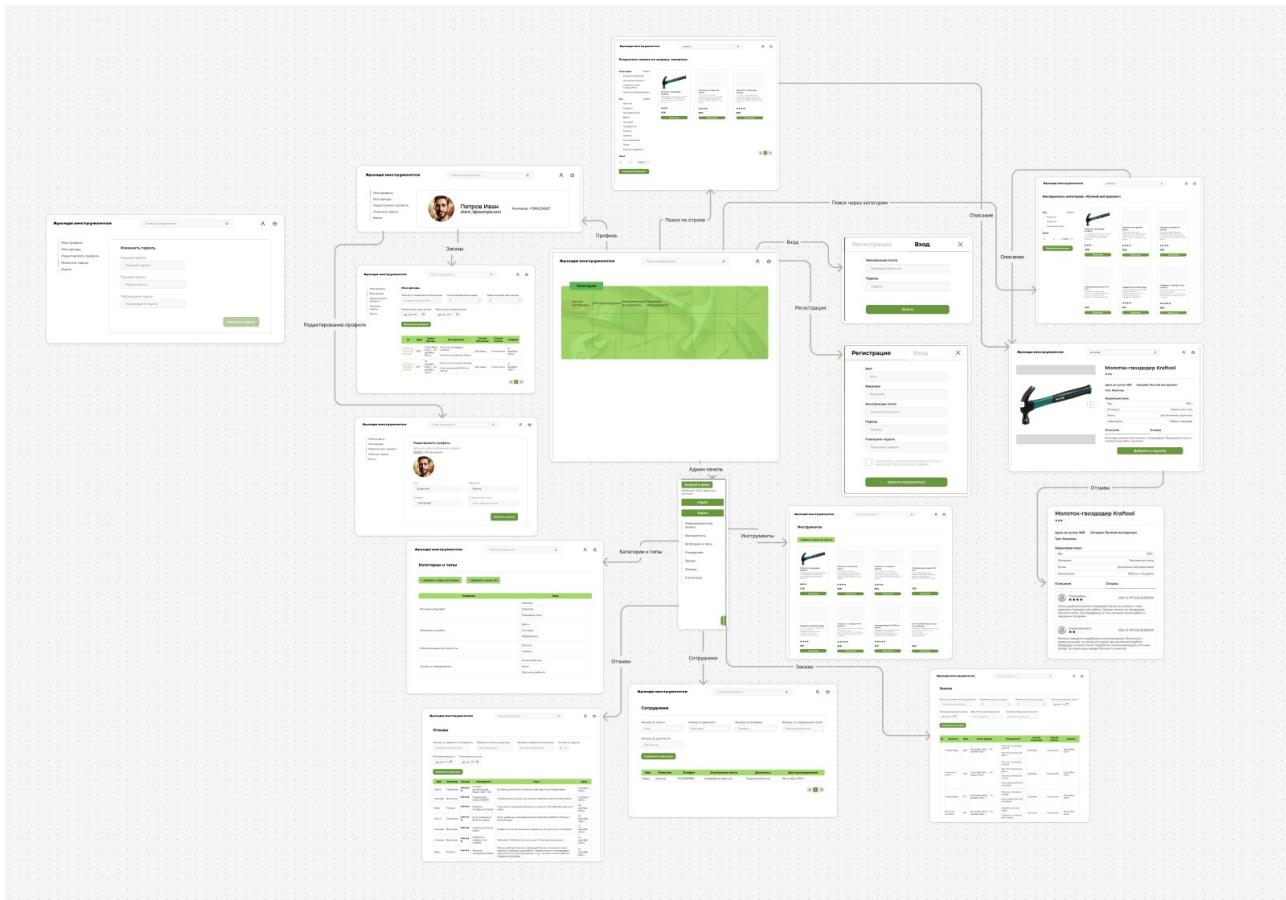


Рисунок 23 - схема экранов приложения

## **5. ВЫВОДЫ**

### **5.1 Достигнутые результаты**

Результатом является приложение, представляющее собой платформу для аренды приложения, ориентированную на взаимодействие между клиентами и сотрудниками. Приложение предоставляет пользователям удобный инструмент для управления процессом аренды.

Клиенты могут искать инструменты по запросу, фильтровать их, просматривать полную информацию по каждому инструменту, выбирать их и оформлять заказы. После завершения аренды клиенты имеют возможность оставлять отзыв, что помогает улучшать качество предоставляемых услуг.

Сотрудники платформы обладают доступом к функционалу управления инструментами. Они могут добавлять новые инструменты в систему, эффективно просматривать заказы и отзывы с использованием фильтрации. Система отзывов позволяет сотрудникам оценивать удовлетворенность клиентов и выявлять области для улучшения.

Приложение также поддерживает функции массового импорта и экспорта данных, что упрощает управление информацией, позволяет переносить данные между системами и создавать резервные копии для безопасного хранения.

### **5.2 Недостатки и пути для улучшения полученного решения**

На данный момент приложение для аренды инструментов имеет следующие недостатки, устранение которых позволит улучшить пользовательский опыт и функциональность:

- Отсутствие статистики:

Пользователи, включая сотрудников, не могут анализировать популярность инструментов, частоту их аренды или другие важные данные. Реализация панели аналитики позволит получить ценную информацию и улучшить управление ресурсами

- Неоптимизированный пользовательский интерфейс:

Текущий интерфейс приложения может быть улучшен для повышения

интуитивности и удобства использования. Доработка дизайна, улучшение навигации и адаптация под различные устройства сделают работу с приложением комфортнее

- Отсутствие нативных приложений:

Приложение доступно только через веб-интерфейс, что может быть неудобно для пользователей, предпочитающих мобильные платформы. Разработка нативных приложений для iOS и Android позволит расширить аудиторию.

- Ограниченный функционал управления инструментами

В текущей версии приложения отсутствует возможность редактировать данные добавленных инструментов. Добавление этой функции значительно упростить управлением каталогом сотрудниками.

- Отсутствие уведомлений

Пользователи не получают уведомления о статусе заказов, обновлениях или важных событиях. Реализация системы уведомлений (например, через email или push-уведомлений) повысит информированность клиентов и сотрудников.

- Нет интеграции с внешними сервисами оплаты

Процесс аренды не может быть осуществлен без внешнего сервиса оплаты, необходима интеграция с популярными платежными системами.

- Улучшение поисковой системы инструментов

В текущей версии приложения используется поиск по индексу \$text в MongoDB. Этот механизм позволяет находить документы, содержащие заданные слова или их производные, благодаря применению стемминга и разбиению текста на токены. Для улучшения точности и скорости поиска можно использовать специализированные поисковые системы.

### **5.3 Будущее развитие решения**

Основываясь на выявленных недостатках, дальнейшее развитие приложения может включать реализацию панели аналитики, улучшение пользовательского интерфейса, разработку нативных мобильных приложений и расширение функционала управления инструментами. Также стоит уделить внимание интеграции системы уведомлений и внешних сервисов оплаты, а для повышения точности и скорости поиска рассмотреть внедрение специализированных поисковых систем. Эти улучшения позволят значительно расширить функциональность, обеспечив пользователям более удобный и эффективный инструмент для работы.

## **6. ПРИЛОЖЕНИЯ**

### **6.1 Документация по сборке и развертыванию приложения**

1. Склонировать репозиторий с проектом по ссылке:  
<https://github.com/moevm/nosql2h24-tools>
2. Перейти в корневую директорию проекта
3. Собрать контейнеры приложения командой `docker-compose build -no-cache`
4. Запустить контейнеры приложения командой `docker-compose up`
5. Открыть приложение в браузере по адресу: <http://localhost:5173/>

### **6.2 Инструкция для пользователя**

- Регистрация

Для создания аккаунта нажмите на главной странице на иконку пользователя, после выберите раздел «Регистрация». Введите ваше имя, фамилию, электронную почту, пароль и подтверждение пароля, после чего нажмите на кнопку «Зарегистрироваться». В случае, если данные будут некорректны (например, подтверждение пароля не совпадает с реальным паролем), то вы увидите всплывающие окна, которые пояснят в чем ошибка.

- Вход

Для входа нажмите на главной странице на иконку пользователя, после выберите раздел «Вход». Введите вашу электронную почту и пароль, указанные при регистрации, после чего нажмите на кнопку «Войти» . В случае, если данные будут некорректны, то вы увидите всплывающие окна, которые пояснят в чем ошибка.

- Выход

Для выхода из профиля нажмите на главной странице на иконку пользователя, после нажмите на раздел «Выйти»

- Редактирование профиля

Для редактирования профиля нажмите на главной странице на иконку пользователя, после выберите раздел «Редактировать профиль». Введите ваши

новые данные, затем нажмите на кнопку «Обновить профиль». В случае, если новые данные будут некорректны, то вы увидите всплывающую ошибку.

- Изменение пароля

Для редактирования пароля нажмите на главной странице на иконку пользователя, после выберите раздел «Изменить пароль». Введите ваш текущий пароль, новый пароль и подтверждение нового пароля, затем нажмите на кнопку «Изменить пароль». В случае, если новые данные будут некорректны, то вы увидите всплывающую ошибку.

- Просмотр ваших аренд

Для просмотра аренд нажмите на главной странице на иконку пользователя, после выберите раздел «Мои аренды». Вам выдаст таблицу, где представлены ваши аренды. В данном разделе можно применить фильтры, благодаря чему можно просмотреть аренды соответствующие определенным критериям.

- Просмотр инструментов по категориям

На главной странице нажмите на любую категорию, в результате будут показаны инструменты, которые относятся к данной категории, после чего их можно будет фильтровать по определенным параметрам.

- Поиск инструментов

Для поиска инструмента введите поисковой запрос в строке на главной странице.

- Фильтрация инструментов

После поиска полученные результаты можно отфильтровать с помощью панели, расположенной слева. После выбора фильтров нажмите на кнопку «Применить фильтры».

- Просмотр инструментов

После поиска и/или фильтрации инструментов, чтобы просмотреть детали определенного инструмента нажмите на кнопку «Просмотр» в карточке инструмента, после чего вам откроется полное описание инструмента, а также отзывы.

Далее описаны функции, которые доступны только сотрудникам

- Просмотр существующих категорий и типов

Для просмотра существующих категорий и типов откройте админ-панель, после выберите раздел «Категории и типы».

- Добавление категориев

Для добавления категорий откройте админ-панель, после выберите раздел «Категории и типы», затем нажмите на кнопку «Добавить новую категорию», после чего введите название категории и подтвердите выбор

- Добавление типов

Для добавления типов откройте админ-панель, после выберите раздел «Категории и типы», затем нажмите на кнопку «Добавить новый тип», после чего введите название типа, и выберите категорию, к которой относится данный тип, затем подтвердите выбор.

- Просмотр существующих инструментов

Для просмотра существующих инструментов откройте админ-панель, после выберите раздел «Инструменты».

- Добавление инструментов

Для добавления инструментов откройте админ-панель, после выберите раздел «Инструменты», затем нажмите на кнопку «Добавить новый инструмент», введите все параметры и подтвердите выбор.

- Просмотр сотрудников

Для просмотра существующих сотрудников откройте админ-панель, после выберите раздел «Сотрудники». Сотрудников можно отфильтровать с помощью представленных фильтров.

- Просмотр заказов

Для просмотра существующих заказов откройте админ-панель, после выберите раздел «Заказы». Заказы можно отфильтровать с помощью представленных фильтров.

- Просмотр отзывов

Для просмотра существующих отзывов откройте админ-панель, после выберите раздел «Отзывы». Отзывы можно отфильтровать с помощью представленных фильтров.

- Массовый экспорт

Для массового экспорта нажмите на кнопку «Export», после чего будет скачан JSON, содержащий все записи БД.

- Массовый импорт

Для массового импорта нажмите на кнопку «Загрузить файл», выберите соответствующий JSON, после чего нажмите на кнопку «Import».

## **7. ЛИТЕРАТУРА**

1. GitHub URL: <https://github.com/moevm/nosql2h24-tools> (дата обращения: 23.12.2024).
2. MongoDB Documentation URL: <https://www.mongodb.com/docs/> (дата обращения: 23.12.2024).
3. Vue.js Documentation URL: <https://vuejs.org/guide/introduction.html> (дата обращения: 23.12.2024).
4. Docker Documentation URL: <https://docs.docker.com/manuals/> (дата обращения: 23.12.2024).
5. Python Documentation URL: <https://docs.python.org> (дата обращения: 23.12.2024).
6. Motor Documentation URL: <https://motor.readthedocs.io/en/stable/> (дата обращения: 23.12.2024).