

**МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Приложение для проката спортивного инвентаря

Курсовой проект

09.03.02 Информационные системы и технологии
Программная инженерия в информационных системах

Обучающийся _____ М.К. Пилеич, 3 курс, д/о

Обучающийся _____ Е.А. Селиванова, 3 курс, д/о

Обучающийся _____ А.Е. Шакайло, 3 курс, д/о

Руководитель _____ В.С. Тарасов, ст. преподаватель

Руководитель  И.В. Клейменов, ассистент

Воронеж 2022

Оглавление

Введение	4
1 Постановка задачи	5
1.1 Цель	5
1.2 Сфера использования	5
1.3 Требования	5
1.4 Задачи	5
2 Анализ предметной области	6
2.1 Глоссарий	6
2.2 Целевая аудитория	6
2.3 Анализ существующих решений	7
2.4 Пользователи системы	10
2.5 Анализ задачи	11
2.5.1 Функциональные требования	11
2.5.2 Варианты использования системы	12
2.5.3 Взаимодействие компонентов системы	18
2.5.4 Диаграмма IDEF0	19
2.5.5 Диаграмма состояний	20
2.5.6 Схема базы данных	21
2.5.7 Диаграмма классов	23
2.5.8 Диаграмма развёртывания	24
2.5.9 Архитектура приложения	24
3 Проектная часть	27
3.1 Главная страница (неавторизованный пользователь)	27
3.2 Страницы авторизации и регистрации	28
3.3 Главная страница (авторизованный пользователь)	29
3.4 Профиль пользователя	29
3.5 Мои аренды	30
3.6 Арендный чек	31
3.7 Главная страница (администратор)	31
3.8 Профили пользователей	32

4 Аналитика	33
Заключение	33
Список используемых источников.....	34

Введение

В настоящее время существует много различных сервисов, позволяющих покупать или брать в аренду необходимые вещи. С развитием информационных технологий этот процесс стал значительно проще и быстрее. Коммерческий успех различных организаций также напрямую зависит от качества технического оснащения.

В связи с популяризацией физической культуры и здорового образа жизни было решено создать веб-приложение для проката спортивного инвентаря.

В данном курсовом проекте разрабатывается веб-приложение, которое можно использовать для конкретной точки проката спортивного снаряжения. Приложение должно облегчить и ускорить процесс взятия инвентаря в аренду благодаря тому, что оформить и оплатить аренду можно будет не только непосредственно в точке проката, но и онлайн.

1 Постановка задачи

1.1 Цель

Разработать веб-приложение для автоматизации взятия спортивного инвентаря в аренду Пользователями. Также приложение должно помогать Администратору вести учёт спортивного инвентаря в наличии.

1.2 Сфера использования

Повседневная жизнь. Спорт. Хобби.

1.3 Требования

Система должна выполнять следующие функциональные требования:

- регистрация пользователей (с помощью логина и пароля, а также через Google почту)
- авторизация пользователей
- добавление, редактирование, удаление данных о спортивном инвентаре
- просмотр данных о доступном инвентаре
- поиск инвентаря по названию
- расчёт стоимости аренды инвентаря
- выдача QR-кода на оплату
- редактирование, удаление записей о пользователях

1.4 Задачи

Для достижения поставленной цели были выделены следующие задачи:

- провести анализ существующих решений с целью определения достоинств и недостатков систем, реализующих схожие требования
- спроектировать приложение с учётом данных, полученных в результате выполнения анализа
- реализовать приложение, соответствующее требованиям, представленным выше
- описать результат разработки

Для реализации веб-приложения были выделены следующие подзадачи:

- разработать front-end часть приложения, развернуть её на удалённом сервере
- разработать back-end часть приложения, развернуть её на удалённом сервере
- создать связь между front-end и back-end с помощью REST API
- разработать базу данных, развернуть её на удалённом сервере

2 Анализ предметной области

2.1 Глоссарий

Аренда (заявка на аренду) – заполненная пользователем информация, необходимая и достаточная для аренды спортивного инвентаря.

Администратор сайта – специалист, осуществляющий информационную поддержку сайта, управление контентом.

Карточка товара (инвентаря) – визуальное представление товарной единицы, которую можно взять в аренду.

Контент – совокупность информационного наполнения веб-приложения.

Неавторизованный пользователь (гость) – неавторизованный в веб-приложении человек, которому доступны ограниченные функциональные возможности системы.

Пользователь – человек, который зарегистрирован в системе и имеет доступ к личному кабинету и основному функционалу системы.

Система – разрабатываемое веб-приложение.

2.2 Целевая аудитория

Целевая аудитория данного приложения будет зависеть от спортивного инвентаря, который будет предоставляться в аренду в пункте проката. К примеру, если пункт проката будет предоставлять в аренду инвентарь для

проведения соревнований по какому-либо виду спорта, то приложение будет полезно для организаторов спортивных мероприятий или праздников.

Если же пункт проката будет предоставлять в аренду индивидуальное спортивное снаряжение, то категории людей, которые может заинтересовать приложение, становятся ещё более разнообразными: это могут быть родители, чей ребёнок хочет попробовать научиться кататься на коньках, это могут быть молодые люди, желающие покататься на велосипедах за городом, или человек, который хочет выработать привычку добираться до работы на велосипеде. Это могут быть спортсмены-профессионалы и спортсмены-любители. Это могут быть люди, которые хотят начать вести здоровый образ жизни «с понедельника» и находятся в поиске вида спорта, который придётся им по душе.

Отдельной категорией пользователей данного веб-приложения являются владельцы пунктов проката спортивного инвентаря.

Жизнь большинства людей в определённой мере связана со спортом, поэтому приложение окажется полезным и подойдёт практически любому человеку.

2.3 Анализ существующих решений

1. RentStation

<https://rentstation.ru/>

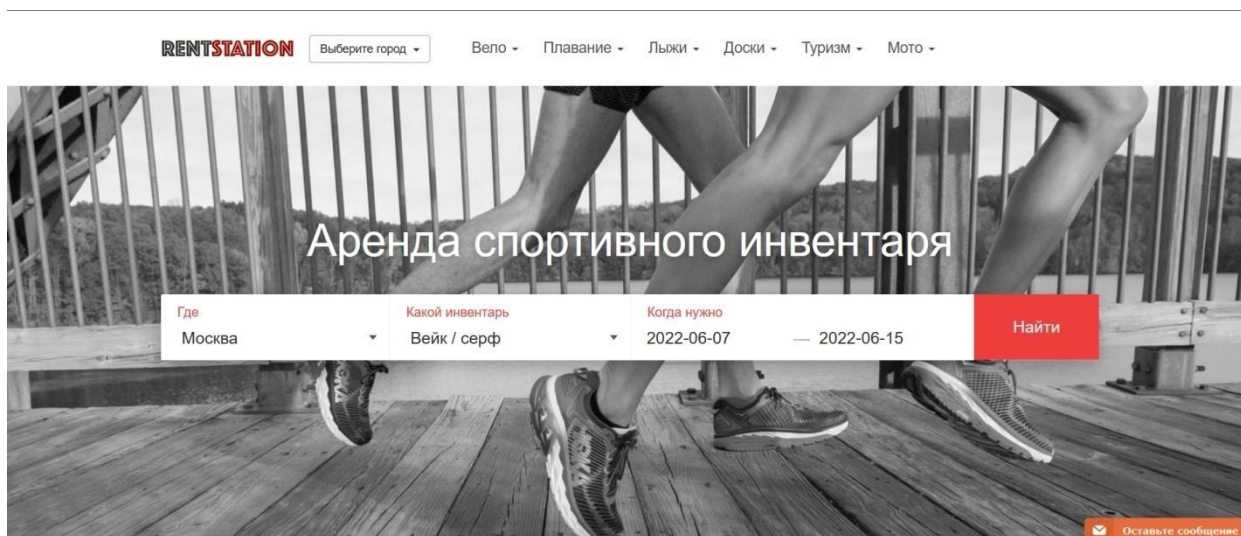


Рисунок 1 - Начальная страница RentStation

Достоинства:

- большой выбор категорий спортивного инвентаря для аренды
- можно авторизоваться с помощью учётной записи на Facebook
- можно забронировать товар за собой на 30 минут без оформления заявки на аренду

Недостатки:

- нет почасовой аренды, минимальный срок аренды – 1 неделя
- нельзя оплатить аренду с помощью сайта

2. EasyPro

<https://easysoftware.pro/>

Дата и время создания	№ карточки	Клиент	Телефон	Состояние	Вид аренды	Начало аренды	Окончание аренды	Сумма со скидкой	Задолженность	Точка выдачи	Точка возврата
22.04.2014 12:34	454578	Степанько Федор Иванович	7 888 555 45 14	В аренде	Посуточная	22.04.2014 12:34	26.04.2014 12:34:00	8 400	0	Арбат	Чистые пруды
22.04.2014 12:33	55789555	Саманович Елена	+7 895 221 11 11	В аренде	Посуточная	22.04.2014 12:33	24.04.2014 12:33:00	1 800	0	Чистые пруды	Чистые пруды
22.04.2014 12:38	132132123	Бендер Остап Никифорович	2585522	В аренде	Почасовая	22.04.2014 12:38	22.04.2014 17:38:00	1 200	1 200	Чистые пруды	Чистые пруды
22.04.2014 12:30	123456789	Иванова Мила	7 985 354 74 14	В аренде	Почасовая	22.04.2014 12:30	22.04.2014 17:30:00	712,5	0	Арбат	Арбат
12.04.2014 14:11	МММ123	Мухомов Михаил Михайлович	1322 456 789797	В аренде	Посуточная	12.04.2014 14:11	14.04.2014 14:11:00	4 950	2 950	Чистые пруды	Чистые пруды

Рисунок 2 - Страница учёта проката EasyPro

Достоинства:

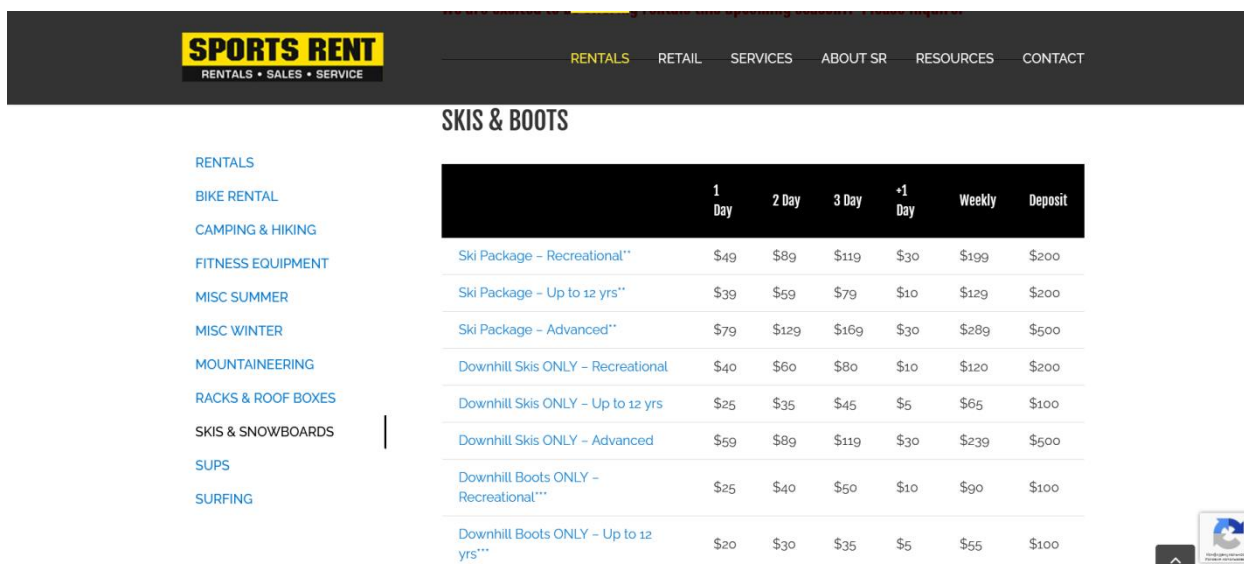
- большое количество функций для учёта спортивного инвентаря со стороны владельца проката
- возможность создания различных отчётов и выгрузки их в Excel, PDF, Word
- возможность просмотреть статистику по конкретному клиенту (количество прокатов, общая сумма всех его аренд, примечания)
- возможность выдать инвентарь в аренду по его штрих-коду
- возможность настроить оповещения для каких-либо событий
- возможность оповещения клиентов по sms, email

Недостатки:

- сервис предназначен только для учёта проката владельцем, пользовательская часть отсутствует

3. Sports Rent

<https://sportsrentbc.com/>



SPORTS RENT RENTALS • SALES • SERVICE						
RENTALS RETAIL SERVICES ABOUT SR RESOURCES CONTACT						
SKIS & BOOTS						
	1 Day	2 Day	3 Day	4 Day	Weekly	Deposit
Ski Package – Recreational**	\$49	\$89	\$119	\$30	\$199	\$200
Ski Package – Up to 12 yrs**	\$39	\$59	\$79	\$10	\$129	\$200
Ski Package – Advanced**	\$79	\$129	\$169	\$30	\$289	\$500
Downhill Skis ONLY – Recreational	\$40	\$60	\$80	\$10	\$120	\$200
Downhill Skis ONLY – Up to 12 yrs	\$25	\$35	\$45	\$5	\$65	\$100
Downhill Skis ONLY – Advanced	\$59	\$89	\$119	\$30	\$239	\$500
Downhill Boots ONLY – Recreational**	\$25	\$40	\$50	\$10	\$90	\$100
Downhill Boots ONLY – Up to 12 yrs***	\$20	\$30	\$35	\$5	\$55	\$100

Рисунок 3 - Страница с информацией об аренде Sports Rent

Достоинства:

- лаконичный и интуитивно понятный интерфейс сайта, стильный дизайн
- предоставляются услуги не только по аренде спортивного инвентаря, а также по его продаже/перепродаже и обслуживанию
- подробные таблицы с ценообразованием для каждой категории спортивного инвентаря
- в каждой категории спортивного снаряжения также находится сопутствующий инвентарь, который может потребоваться в данном виде спорта

Недостатки:

- нельзя оформить заявку на аренду через сайт, забронировать инвентарь можно только позвонив в точку проката

После изучения аналогов было принято решение разработать приложение, которое должно иметь как пользовательский интерфейс, так и интерфейс администратора для ведения учёта спортивного инвентаря. Также в приложении должна быть реализована оплата аренды по QR-коду.

2.4 Пользователи системы

В системе существует три роли:

- гость
- пользователь
- администратор

Пользователь обладает следующими параметрами:

- логин
- пароль
- электронная почта
- ФИО

Использование системы пользователем включает в себя работу с личным кабинетом, просмотр каталога с инвентарём и поиск по нему, добавление/удаление инвентаря из заявки на аренду, оплата аренды, просмотр оплаченных аренд.

Гость – это неавторизованный в системе пользователь, поэтому он не обладает никакими параметрами. Из функциональных возможностей системы ему доступен только просмотр каталога инвентаря и поиск по нему. Также гость может пройти авторизацию или регистрацию для того, чтобы открыть доступ к функциональным возможностям пользователя.

Администратор обладает такими же параметрами, как и пользователь. Использование системы администратором включает в себя работу с записями о спортивном инвентаре: их добавление, удаление, редактирование и работу с записями о клиентах: их редактирование и удаление.

2.5 Анализ задачи

2.5.1 Функциональные требования

В разрабатываемом приложении неавторизованный пользователь (гость) должен обладать следующими функциональными возможностями:

- просмотр главной страницы с доступным для аренды спортивным инвентарём
- поиск инвентаря по названию
- сортировка инвентаря по сезону
- сортировка инвентаря по возрасту
- возможность регистрации
- возможность авторизации

Функциональные возможности пользователя:

- просмотр главной страницы с доступным для аренды спортивным инвентарём
- поиск инвентаря по названию
- сортировка инвентаря по сезону
- сортировка инвентаря по возрасту
- возможность сформировать заявку на аренду инвентаря
- просмотр страницы со всеми арендами (активная аренда, прошедшие аренды)
- просмотр собственного профиля
- возможность изменения данных в профиле
- получение QR-кода для оплаты аренды инвентаря
- выход из профиля

Функциональные возможности администратора:

- просмотр главной страницы с доступным для аренды спортивным инвентарём
- поиск инвентаря по названию
- сортировка инвентаря по сезону

- сортировка инвентаря по возрасту
- добавление инвентаря в базу данных
- удаление инвентаря из базы данных
- подтверждение возврата инвентаря пользователем/**/
- редактирование описания существующих записей о спортивном инвентаре
- удаление пользователей

2.5.2 Варианты использования системы

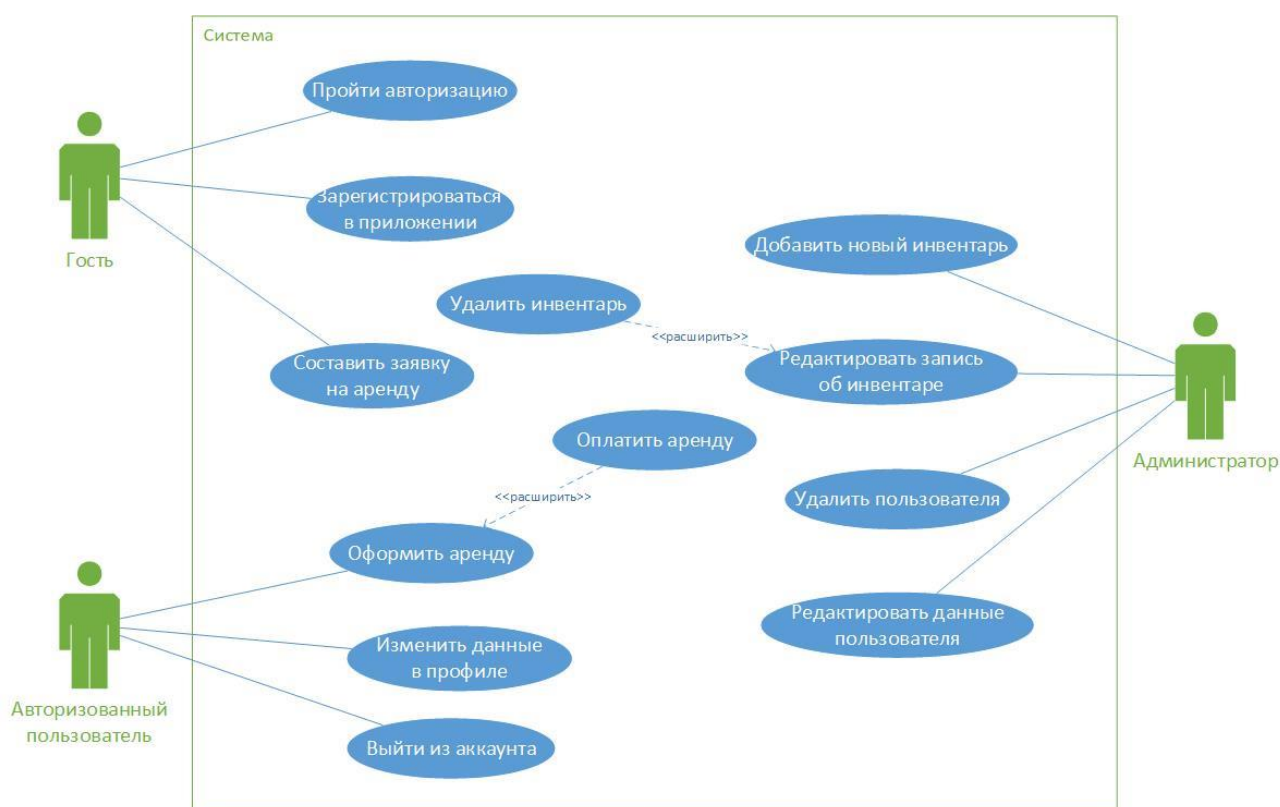


Рисунок 4 - Диаграмма прецедентов

Вариант использования: Зарегистрироваться в приложении.

Актёры: Гость.

Краткое описание: Неавторизованный пользователь (Гость) проходит регистрацию в приложении.

Цель: Зарегистрироваться в приложении и получить доступ к функциональности авторизованного пользователя.

Предусловия: Пользователь не авторизован.

Типичный ход выполнения сценария:

1. Неавторизованный пользователь заходит в приложение.
2. Неавторизованный пользователь в шапке сайта выбирает раздел «Профиль» или «Мои аренды» или нажимает кнопку «Арендовать» на карточке инвентаря.
3. На экране пользователя появляется форма для регистрации/авторизации на выбор.
4. Пользователь выбирает форму для регистрации.
5. Пользователь вводит логин, электронную почту и пароль.
(Исключение №1)
6. Система добавляет данные пользователь в БД.
7. Пользователь переходит на страницу, которую он выбрал в пункте 2, и получает доступ к функциональности авторизованного пользователя.

Исключения сценария выполнения варианта использования:

1. Пользователь с данным логином или электронной почтой уже существует. Система выводит соответствующее предупреждение на экран пользователя.

Вариант использования: Пройти авторизацию.

Актёры: Гость.

Краткое описание: Неавторизованный пользователь (Гость) проходит авторизацию в приложении.

Цель: Авторизоваться в приложении и получить доступ к функциональности авторизованного пользователя.

Предусловия: Пользователь не авторизован.

Типичный ход выполнения сценария:

1. Неавторизованный пользователь заходит в приложение.

2. Неавторизованный пользователь в шапке сайта выбирает раздел «Профиль» или «Мои аренды» или нажимает кнопку «Арендовать» на карточке инвентаря.
3. На экране пользователя появляется форма для регистрации/авторизации на выбор.
4. Пользователь выбирает форму для авторизации.
5. Пользователь вводит свой логин и пароль и нажимает кнопку «Войти».
(Исключение №1)
6. Пользователь переходит на страницу, которую он выбрал в пункте 2, и получает доступ к функциональности авторизованного пользователя.

Исключения сценария выполнения варианта использования:

1. Пользователь с введенным логином и паролем не существует.
Система выводит соответствующее предупреждение на экран пользователя.

Вариант использования: Составить заявку на аренду.

Актёры: Гость.

Краткое описание: Гость составляет заявку на аренду спортивного инвентаря.

Цель: Составить заявку на аренду спортивного инвентаря.

Типичный ход выполнения сценария:

1. В каталоге гость выбирает инвентарь, который он хочет взять в аренду.
2. Гость в шапке сайта выбирает раздел «Мои аренды» и переходит на страницу, где отображается его последняя оформленная заявка на аренду.
3. Гостю предлагается авторизоваться/зарегистрироваться для оплаты аренды инвентаря.

Вариант использования: Оформить аренду.

Актёры: Авторизованный пользователь.

Краткое описание: Пользователь оформляет аренду спортивного инвентаря.

Цель: Оформить и оплатить аренду спортивного инвентаря.

Предусловия: Пользователь авторизован в приложении.

Типичный ход выполнения сценария:

1. В каталоге пользователь выбирает инвентарь, который он хочет взять в аренду.
2. Пользователь в шапке сайта выбирает раздел «Мои аренды» и переходит на страницу, где отображается его последняя неоплаченная аренда и QR-код для оплаты.
3. Пользователь вводит время начала и окончания аренды инвентаря.
4. Пользователь оплачивает аренду.
5. Система автоматически корректирует количество инвентаря в наличии.

Вариант использования: Изменить данные в профиле.

Актёры: Авторизованный пользователь.

Краткое описание: Пользователь изменяет свои данные в приложении.

Цель: Изменить данные в приложении.

Предусловия: Пользователь авторизован в приложении.

Типичный ход выполнения сценария:

1. Пользователь выбирает в шапке сайта раздел «Мой профиль».
2. Пользователь выбирает данные для редактирования.
3. Пользователь редактирует выбранные данные.
(Исключение №1)
4. Новые данные сохраняются в БД.

Исключения сценария выполнения варианта использования:

1. Новый логин или электронная почта уже есть в БД.
Система выводит соответствующее предупреждение на экран пользователя.

Вариант использования: Добавить новый инвентарь.

Актёры: Администратор.

Краткое описание: Администратор добавляет новый инвентарь.

Цель: Добавить запись о новом спортивном инвентаре.

Предусловия: Пользователь авторизован в приложении в качестве администратора.

Типичный ход выполнения сценария:

1. Администратор в шапке сайта выбирает раздел «Добавить инвентарь» и переходит на соответствующую страницу.
2. Администратор вводит наименование инвентаря, сезон, возраст, количество, цену за час и нажимает кнопку «Добавить».
(Исключение №1)
3. В БД добавляется запись о новом спортивном инвентаре.

Исключения сценария выполнения варианта использования:

1. Инвентарь с таким названием и характеристиками уже существует в БД.
Система выводит соответствующее предупреждение.

Вариант использования: Редактировать запись об инвентаре.

Актёры: Администратор.

Краткое описание: Администратор редактирует запись выбранного инвентаря.

Цель: Редактировать запись о выбранном спортивном инвентаре.

Предусловия: Пользователь авторизован в приложении в качестве администратора.

Типичный ход выполнения сценария:

1. В разделе «Каталог» администратор выбирает карточку инвентаря, информацию о котором нужно отредактировать, и переходит на страницу для редактирования.
2. Администратор вносит необходимые изменения.

2а. При необходимости, выбранный спортивный инвентарь можно удалить.

3. Администратор сохраняет изменения.

Вариант использования: Удалить пользователя.

Актёры: Администратор.

Краткое описание: Администратор удаляет пользователя из системы.

Цель: Удалить информацию о пользователе из системы.

Предусловия: Пользователь авторизован в приложении в качестве администратора.

Типичный ход выполнения сценария:

1. Администратор в шапке сайта выбирает раздел «Профили» и переходит на соответствующую страницу.
2. В списке пользовательских профилей администратор выбирает профиль для удаления и удаляет его.

2.5.3 Взаимодействие компонентов системы

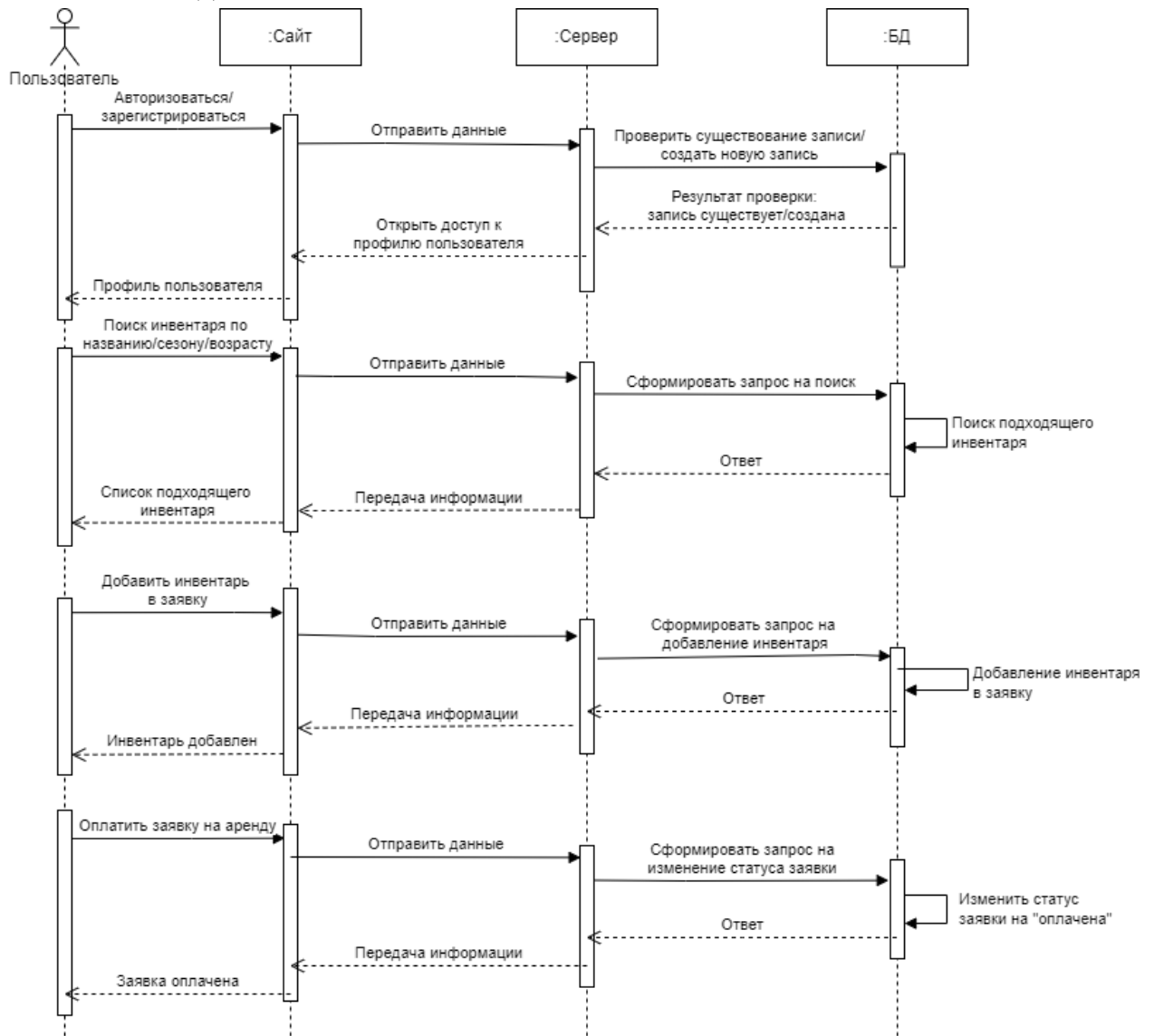


Рисунок 5 - Диаграмма последовательностей

На рисунке изображена диаграмма последовательности для пользователя системы. На ней показано упорядоченное во времени взаимодействие объектов при авторизации/регистрации пользователя в системе, поиске инвентаря по названию, сезону или возрасту, добавлении инвентаря в заявку на аренду, оплате аренды.

2.5.4 Диаграмма IDEF0

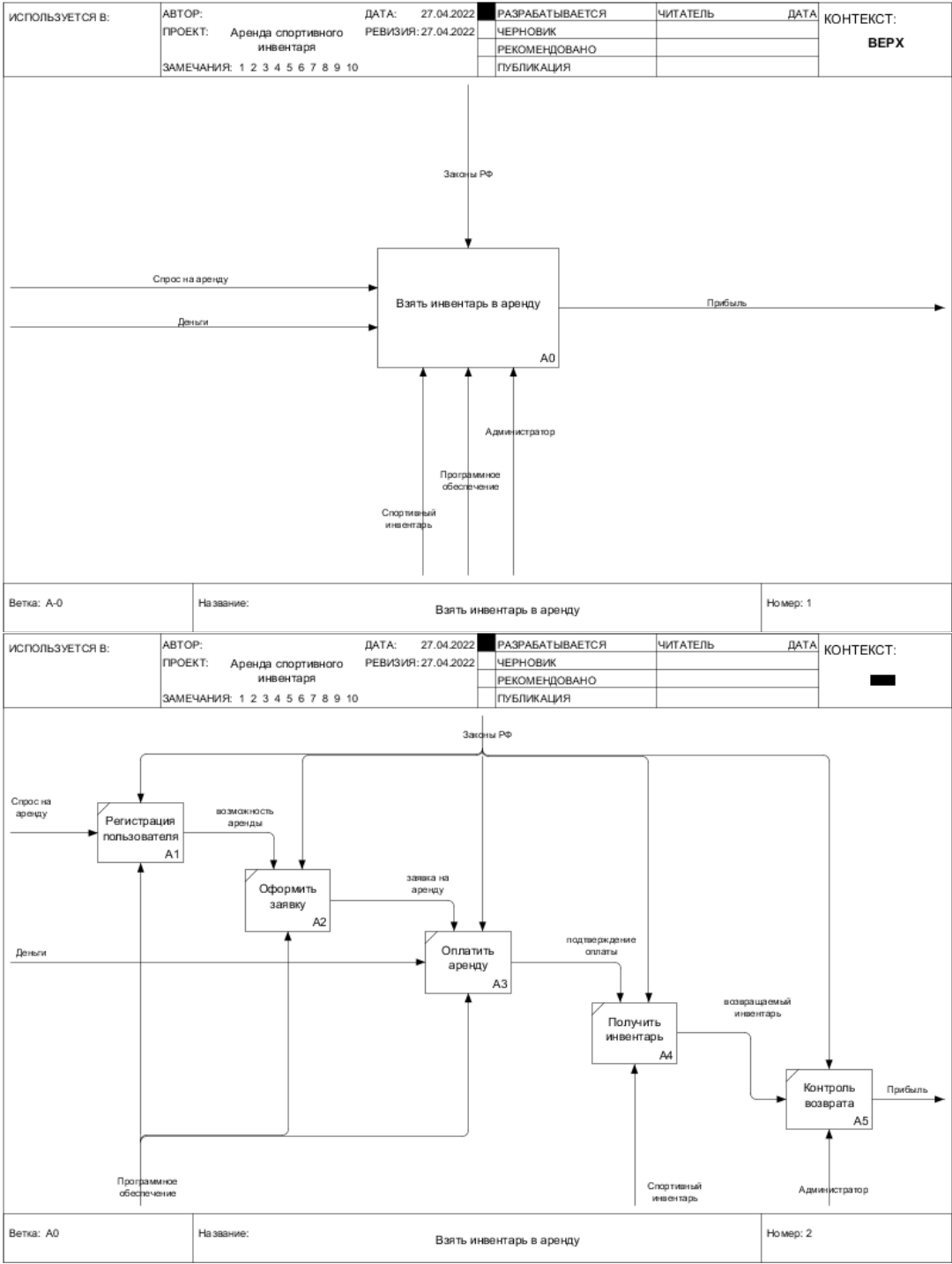


Рисунок 6 - Диаграмма IDEF0

Диаграмма на рисунке 6 описывает процесс взятия спортивного инвентаря в аренду, его оплату и возврат.

2.5.5 Диаграмма состояний

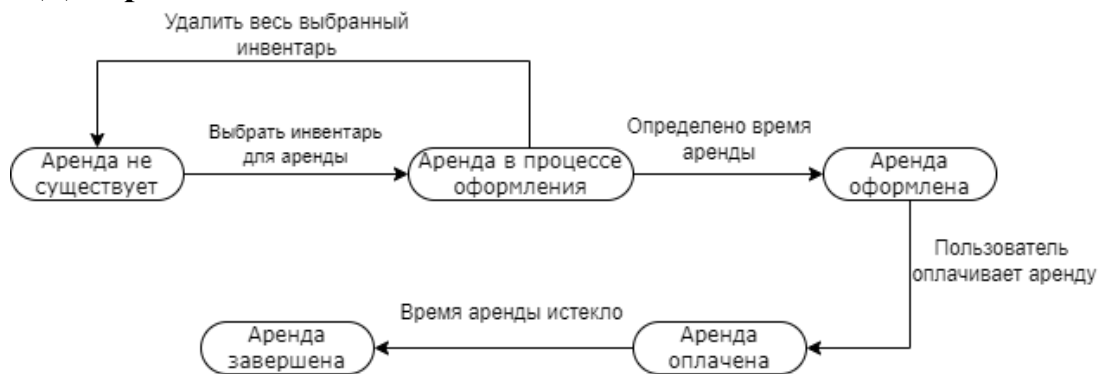


Рисунок 7 - Диаграмма состояний для аренды

На рисунке изображена диаграмма состояний аренды спортивного инвентаря. Для оформления аренды пользователю необходимо выбрать спортивный инвентарь, который он хочет взять в прокат и указать его количество. После этого в личном кабинете в сформированной заявке указать время аренды. В результате будет сформирован QR-код для оплаты аренды. По истечении времени аренда будет завершена.

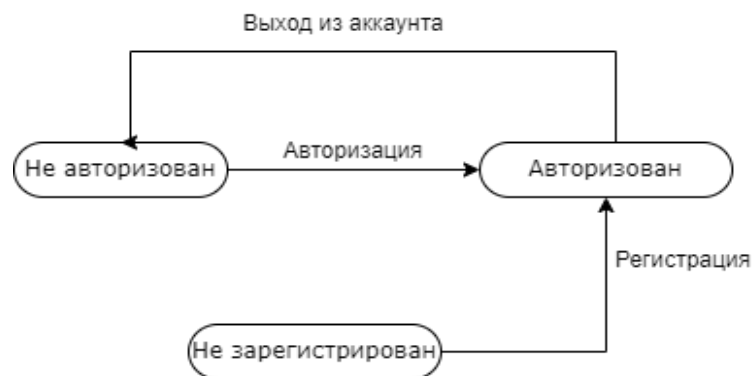


Рисунок 8 - Диаграмма состояний пользователя

На рисунке выше изображены возможные состояния пользователя. Изначально пользователь находится либо в неавторизованном, либо в незарегистрированном состоянии. После авторизации пользователь переходит в состояние авторизованного пользователя, после выхода из аккаунта – вернётся в неавторизованное состояние. Во время процесса регистрации пользователю необходимо заполнить поля, значения которых система проверит на корректность и существование в пользователя в системе. Если данные корректны, пользователь будет зарегистрирован в

системе и перейдёт в авторизованное состояние, в противном случае будет выведено сообщение об ошибке.

2.5.6 Схема базы данных

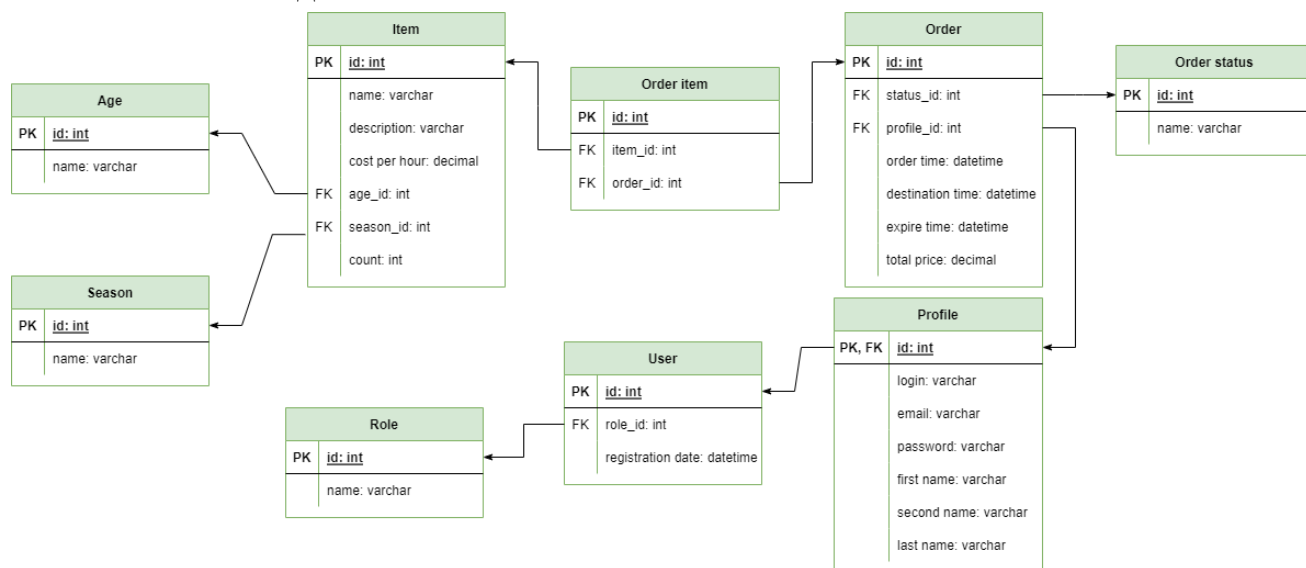


Рисунок 9 - Схема базы данных

На рисунке 9 представлена схема базы данных. В схеме данных существуют таблицы для хранения пользователей и их профилей, инвентаря, аренд, роли пользователя, сезона/возрастной категории инвентаря, статуса аренды, а также связующая таблица для хранения единицы инвентаря в заявке на аренду. А именно:

Profile – таблица, содержащая информацию о профилях пользователей: логин, электронная почта, пароль, ФИО. Поле *id* является одновременно первичным и внешним ключом, который ссылается на запись в таблице User.

User – таблица, содержащая информацию о пользователях: их роль и дату регистрации. Поле *id* является ключевым. Поле *role_id* является внешним ключом, который ссылается на запись в таблице Role.

Role – таблица, содержащая список ролей, которые существуют в приложении (авторизованный пользователь/администратор). Поле *id* является ключевым. Поле *name* содержит название роли.

Order – таблица, содержащая информацию об арендах инвентаря: их статус, профиль, к которому они относятся, время начала и завершения

аренды, стоимость аренды. Поле *id* является ключевым. Поле *status_id* является внешним ключом и ссылается на запись в таблице Order status. Поле *profile_id* является внешним ключом и ссылается на запись в таблице Profile.

Order status – таблица, содержащая список возможных состояний аренды инвентаря (завершена/не завершена). Поле *id* является ключевым.

Item – таблица, содержащая информацию о спортивном инвентаре: его название, краткое описание, цену за час аренды, количество инвентаря в наличии, сезон и возрастную категорию. Поле *id* является ключевым. Поле *age_id* является внешним ключом и ссылается на запись в таблице Age. Поле *season_id* является внешним ключом и ссылается на запись в таблице Season.

Order item – таблица, разрешающая связь многие-ко-многим между сущностями Order и Item. Поле *id* является ключевым. Поле *item_id* является внешним ключом и ссылается на запись в таблице Item. Поле *order_id* является внешним ключом и ссылается на запись в таблице Order.

Age – таблица, содержащая список возрастных категорий спортивного инвентаря. Поле *id* является ключевым. Поле *name* содержит название категории.

Season – таблица, содержащая список сезонов, в которые используется тот или иной спортивный инвентарь. Поле *id* является ключевым. Поле *name* содержит название сезона.

2.5.7 Диаграмма классов

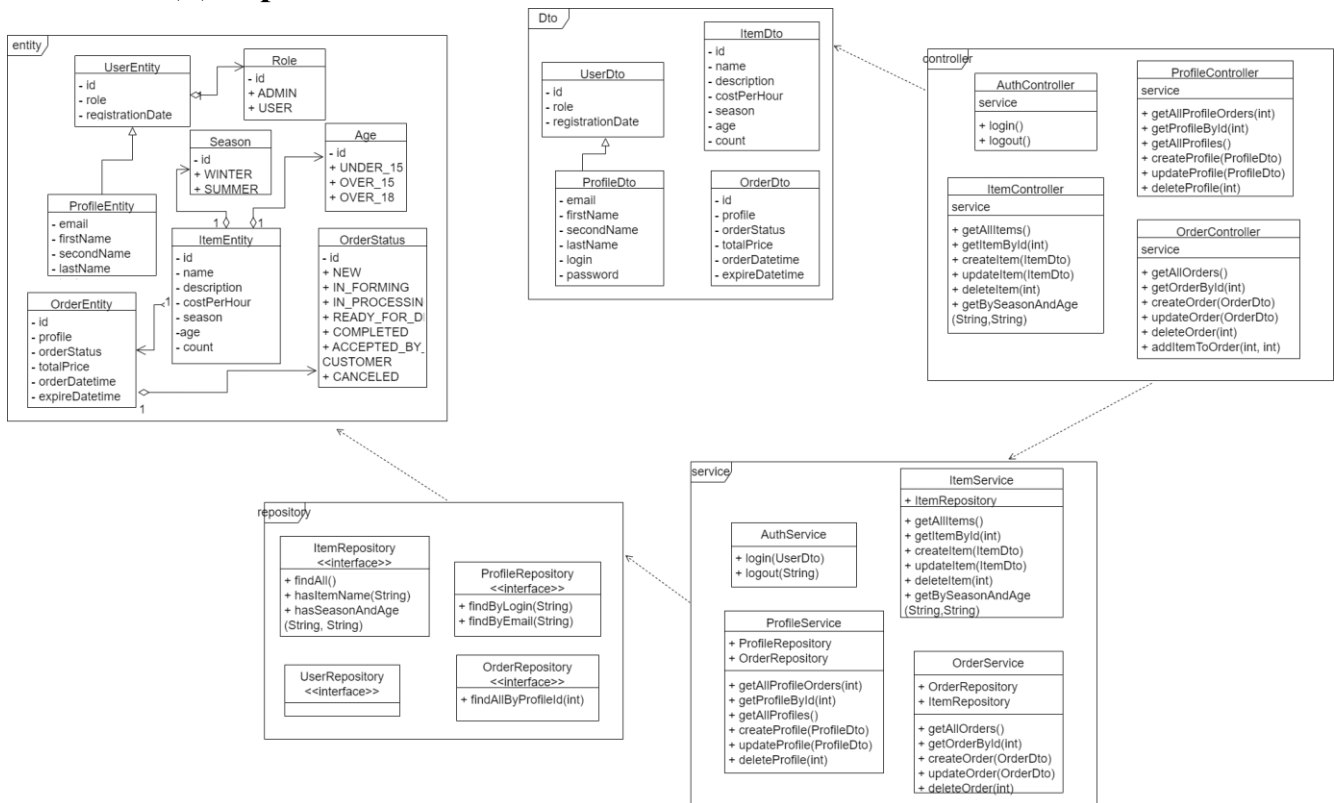


Рисунок 10 - Диаграмма классов

На рисунке 10 представлена диаграмма классов, соответствующая архитектуре приложения.

В пакете *entity* размещены классы, которые описывают модели сущностей базы данных. В пакете *repository* находятся интерфейсы, наследуемые от *JpaRepository* – интерфейса фреймворка Spring Data, предоставляющего стандартные методы для работы с БД. В пакете *controller* описаны классы для взаимодействия между front-end, back-end и БД по получению, проверке, отправлению данных. В пакете *service* находятся классы, которые предоставляют основную бизнес-логику. Сервисы используют Dto, repository или другие сервисы, чтобы предоставить конечные данные для пользовательского интерфейса. В пакете *Dto* описаны объекты, которые используются для передачи данных между front-end и back-end.

2.5.8 Диаграмма развёртывания

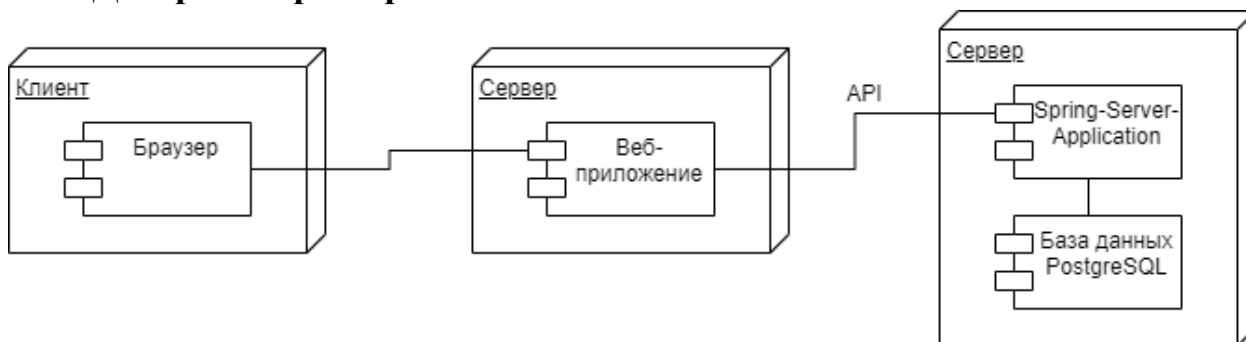


Рисунок 11 - Диаграмма развёртывания

На рисунке 11 изображена диаграмма развертывания, показывающая топологию системы и распределение компонентов по ее узлам - аппаратные компоненты, а также соединения – маршруты передачи информации. Для разрабатываемого веб-приложения узлами являются персональный компьютер и сервер, в качестве узла среды выполнения выступает web-браузер. На сервере развернуты back-end, front-end и база данных. Взаимодействие между back-end и front-end частями осуществляется через API. Клиент взаимодействует с сервером через front-end.

2.5.9 Архитектура приложения

В качестве средств реализации приложения были выбраны следующие технологии:

CSS3 + HTML5 были выбраны потому, что это самый современный стандарт вёрстки и разметки. Он поддерживается большинством браузеров и предоставляет множество новых свойств, упрощающих разработку.

Технология Vue.js 3 выбрана потому, что это современный фреймворк, в основе которого лежит компонентный подход с принципами реактивности. Это упрощает написание кода, сокращает количество повторений одного и того же кода для похожих элементов вёрстки. Кроме того, Vue.js сильно стандартизирован и гибок одновременно. Это позволяет писать компоненты интерфейса на одних и тех же принципах, не жертвуя функциональностью. Vue.js предоставляет множество дополнительных модулей, работающих в

одной экосистеме. Также эта технология упрощает работу с данными (принцип реактивности), которого нет в React или Angular.

Язык TypeScript был выбран потому, что единственным стандартом веб-программирования является EcmaScript, в его JavaScript реализации присутствует множество минусов, которые вытекают из слабой типизированности JavaScript. TypeScript код - это компилируемый в JavaScript код, который нивелирует недостатки слабой типизации JavaScript и берёт множество проверок безопасности кода на себя во время трансляции в JavaScript код.

Библиотека Axios была выбрана потому, что она очень простая и хорошо подходит для лёгких запросов.

Библиотека Pinia — аналог Axios, предназначенная для работы с Vue.js. Её преимуществом перед Axios является разделение ответственности при работе с данными. С помощью этой библиотеки можно вынести работу с данными из компонента там, где это необходимо.

Spring Boot — это фреймворк на основе Java с открытым исходным кодом. Spring Boot является расширением Spring Framework и используется для создания приложений на основе микросервисов. Благодаря автоконфигурации Spring Boot избавляет от написания лишнего кода и помогает избежать ненужной настройки, он настраивает приложение на основе ранее добавленных зависимостей. Также приложение не нужно устанавливать в среде развёртывания, так как встроенный сервер является частью приложения, что сокращает время перезапуска приложения. Сам же Spring Framework обладает такими функциями, как внедрение зависимостей и обработка транзакций.

Spring Data JPA является частью Spring Data. Данный модуль — это удобный механизм для взаимодействия с сущностями базы данных, организации их в репозитории, извлечение данных, изменение, в некоторых

случаях для этого будет достаточно объявить интерфейс и метод в нем, без имплементации.

Maven — это инструмент для управления и сборки проектов. Проекты редко пишутся без использования сторонних библиотек (зависимостей). Эти сторонние библиотеки зачастую тоже используют библиотеки разных версий. Maven позволяет управлять такими сложными зависимостями. Он разрешает подобные конфликты версий и в случае необходимости позволяет легко переходить на новые версии библиотек.

В качестве СУБД была выбрана PostgreSQL в силу открытого доступа и высокой производительности. PostgreSQL даёт возможность использовать различные языки для серверного программирования: встроенного PL/pgSQL (удобного тесной интеграцией с SQL), C для критичных по производительности задач, Perl, Python, Tcl, а также JavaScript, Java и другие. Также PostgreSQL поддерживает все актуальные ОС — более 30 *nix, а также Windows, включая 2000, XP, 2003 и более современные, что обеспечивает доступность установки практически на любом оборудовании.

3 Проектная часть

3.1 Главная страница (неавторизованный пользователь)

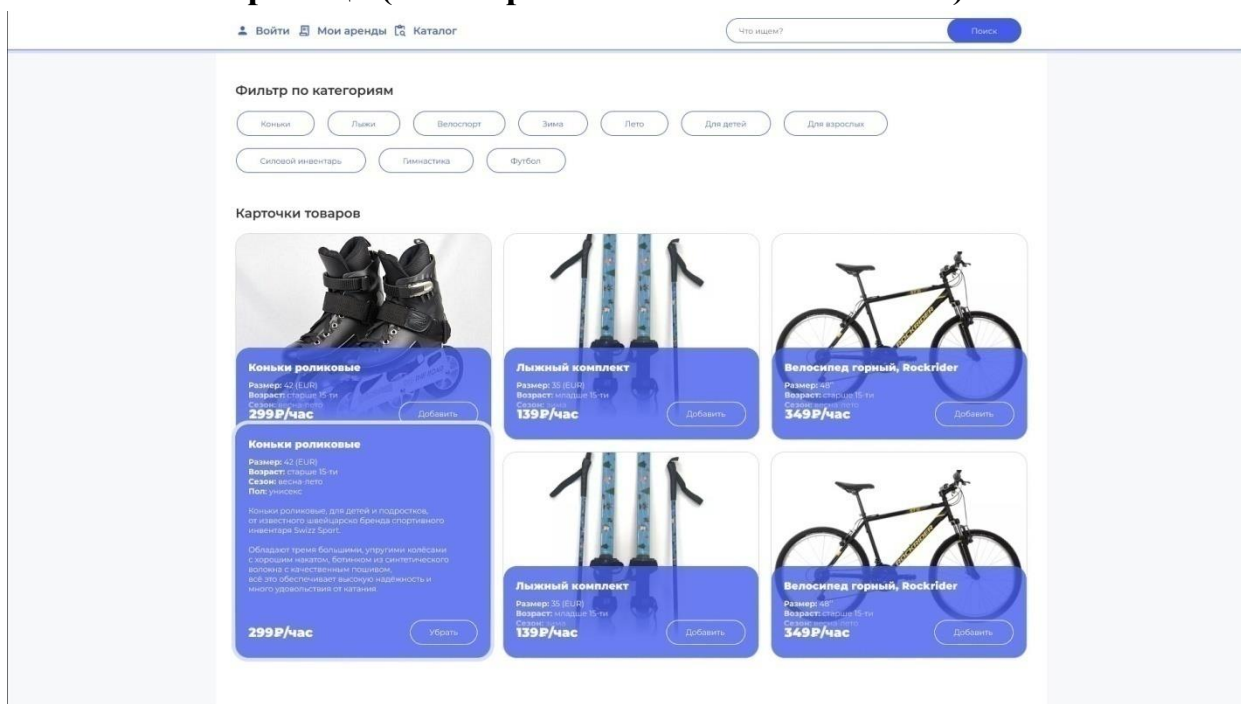


Рисунок 12 - Главная страница для гостя

На главной странице расположены карточки инвентаря, доступного для взятия в аренду, поисковая строка и фильтр по категориям спортивного снаряжения. При наведении курсора на карточку товара выдвигается плашка с его кратким описанием. Инвентарь можно добавить или удалить из заявки на аренду.

Главная страница даёт возможность перейти на страницу авторизации и регистрации, а также на страницу просмотра всех аренд пользователя.

3.2 Страницы авторизации и регистрации

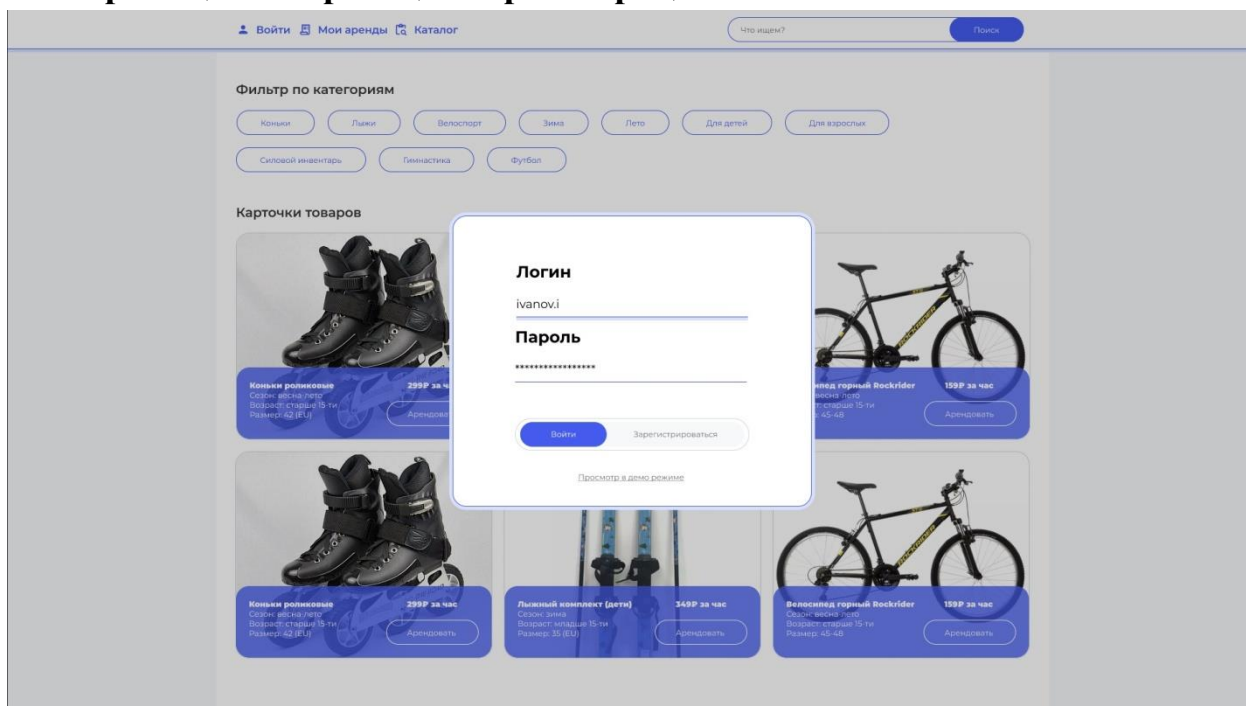


Рисунок 13 - Страница авторизации

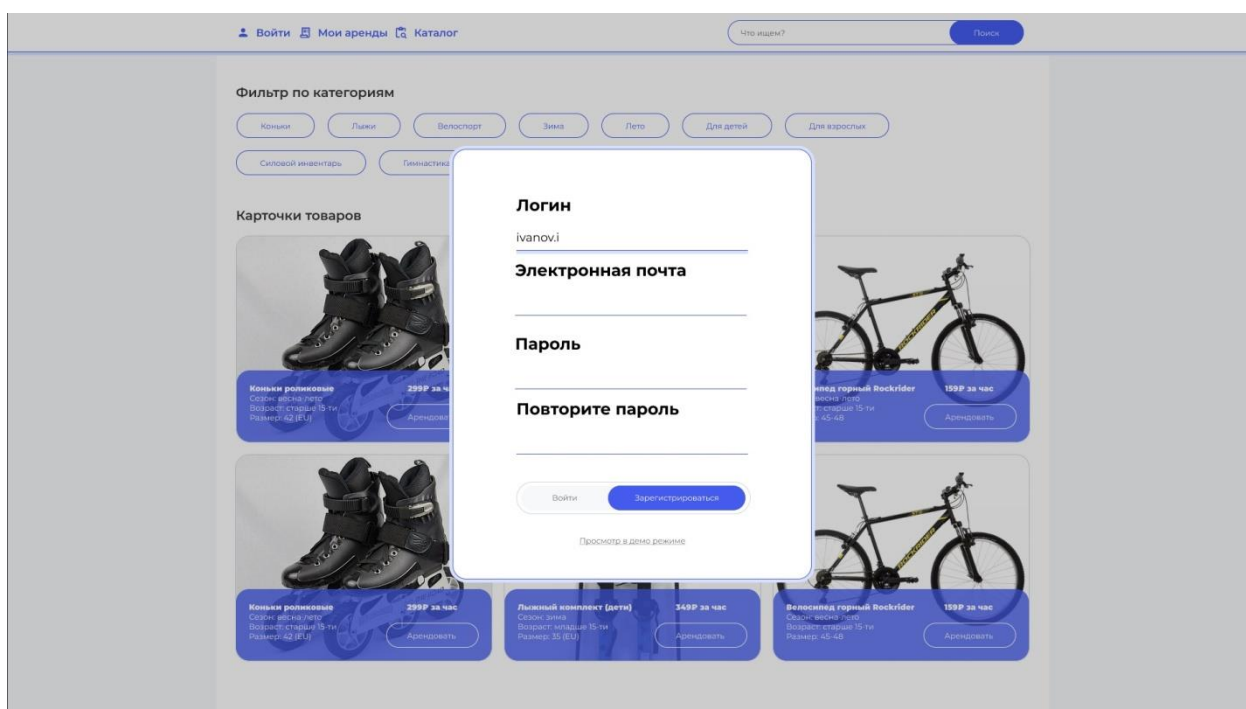


Рисунок 14 - Страница регистрации

Предоставляют пользователю возможность пройти авторизацию или регистрацию для получения функциональных возможностей пользователя.

3.3 Главная страница (авторизованный пользователь)

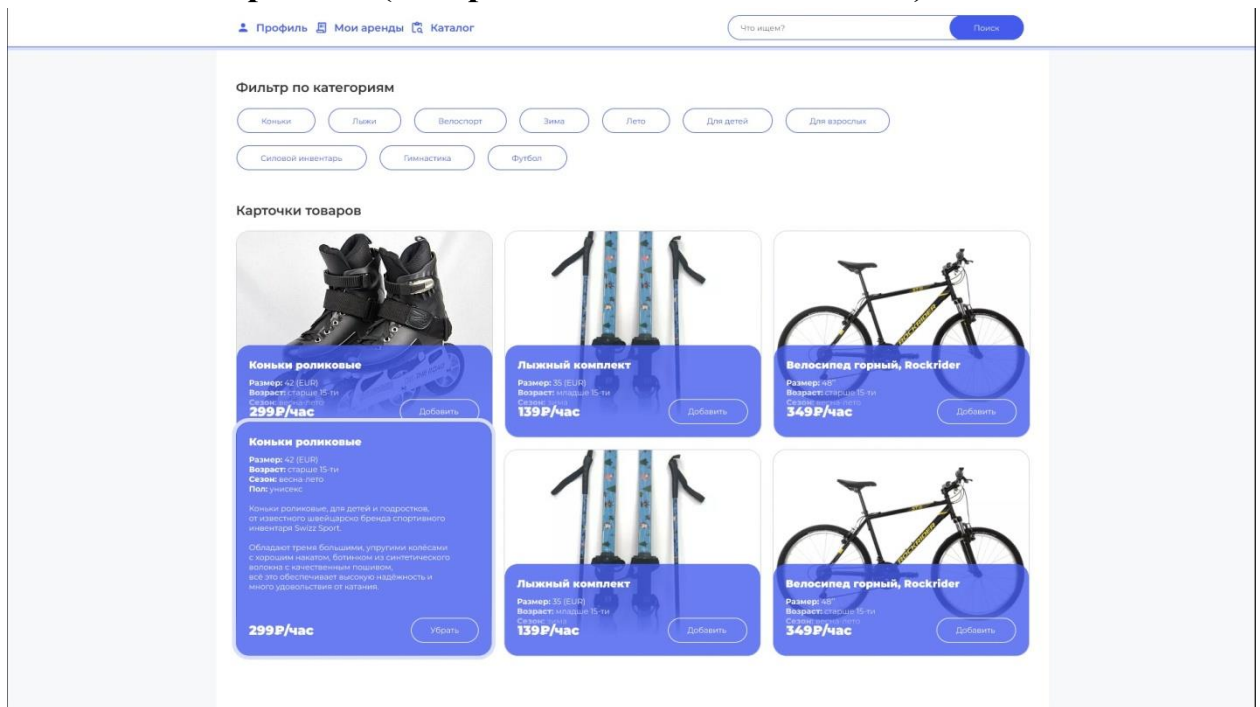


Рисунок 15 - Главная страница для пользователя

Аналогична главной странице неавторизованного пользователя, только вместо возможности регистрации/авторизации добавлена возможность перейти в профиль пользователя.

3.4 Профиль пользователя

На странице профиля пользователь может изменить свои данные, выйти из профиля или удалить свой профиль.

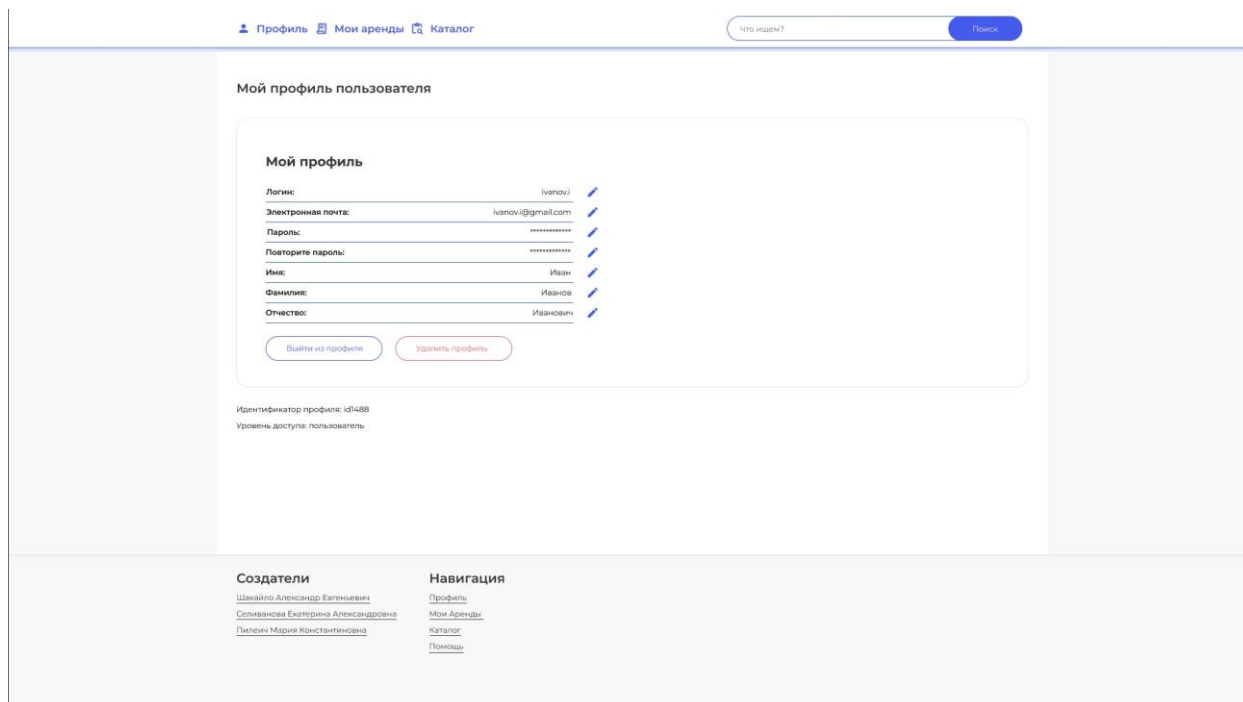


Рисунок 16 - Страница профиля пользователя

3.5 Мои аренды

На данной странице пользователь может просмотреть свои предыдущие аренды спортивного инвентаря, а также оплатить последнюю сформированную заявку на аренду, если она есть. Для оплаты можно либо отсканировать QR-код, либо нажать по нему для перехода на следующую страницу для оплаты.

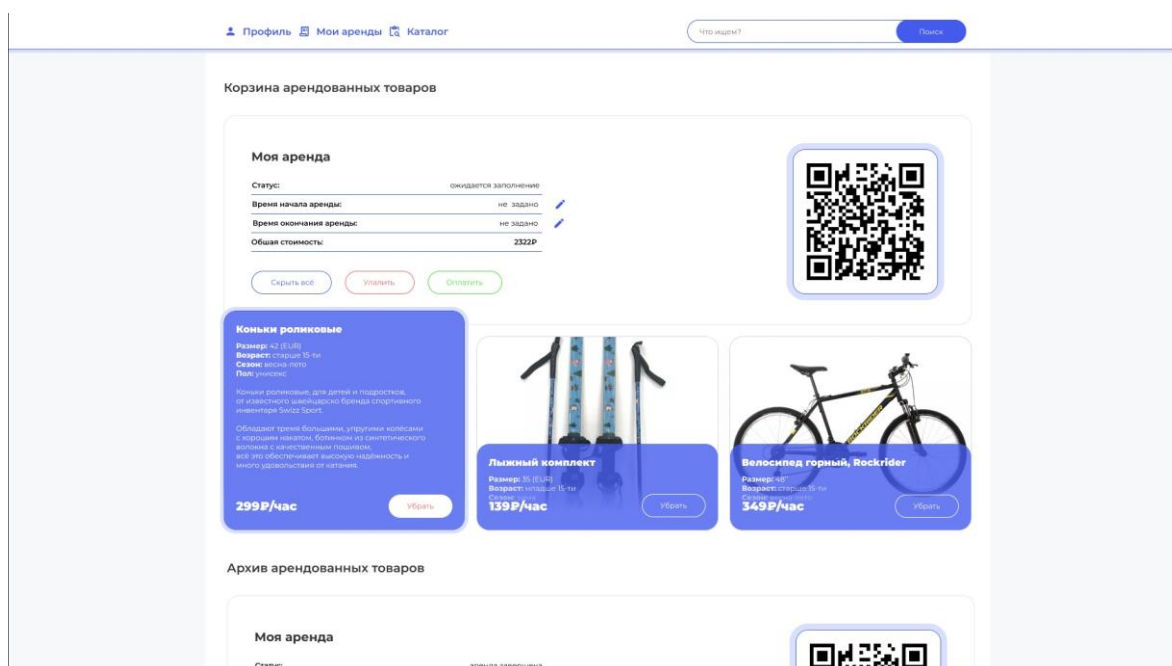


Рисунок 17 - Страница аренд пользователя

3.6 Арендный чек

На данной странице можно посмотреть статус аренды, наименования выбранного инвентаря, общую стоимость аренды и оплатить аренду.

Арендный чек №99481488

Статус: _____ к оплате

Время начала аренды: 22.02.2022

Время окончания аренды: 25.02.2022

Общая стоимость: 19368P

Время формирования чека: 22.02.2022

Список товаров

Наименование: _____ Коньки роликовые

Сезон: _____ Весна-лето

Возраст: _____ старше 15-ти

Размер: _____ 42 (EU)

Количество: _____ x1 шт

Стоимость за арендный период: 7776P

Наименование: _____ Лыжный комплект дети

Сезон: _____ Зима

Возраст: _____ младше 15-ти

Размер: _____ 35 (EU)

Количество: _____ x1 шт

Стоимость за арендный период: 8376P

Наименование: _____ Горный велосипед Rock'ider

Сезон: _____ Весна-лето

Возраст: _____ старше 15-ти

Размер: _____ 45-48

Количество: _____ x1 шт

Стоимость за арендный период: 3816P

[Оплатить](#)

3.7 Главная страница (администратор)

Для администратора на главной странице добавлена возможность добавления, удаления и редактирования карточек инвентаря. Для добавления новой записи о спортивном инвентаре нужно нажать на пустую карточку товара «Добавить». Откроется форма (нижняя левая карточка), в которой нужно указать необходимые характеристики инвентаря, после чего нажать кнопку «Сохранить».

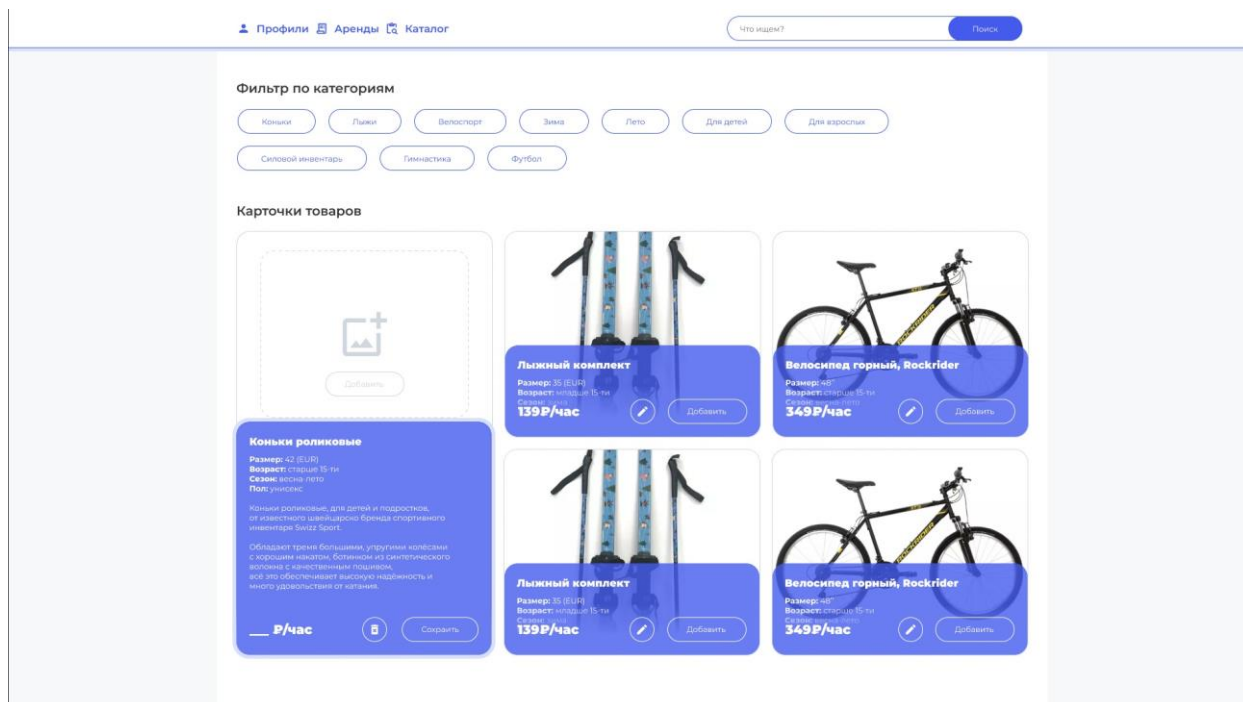


Рисунок 18 - Главная страница для администратора

3.8 Профили пользователей

Страница просмотра пользовательских профилей доступна только администратору. На данной странице можно редактировать информацию в профилях пользователей или удалять их.

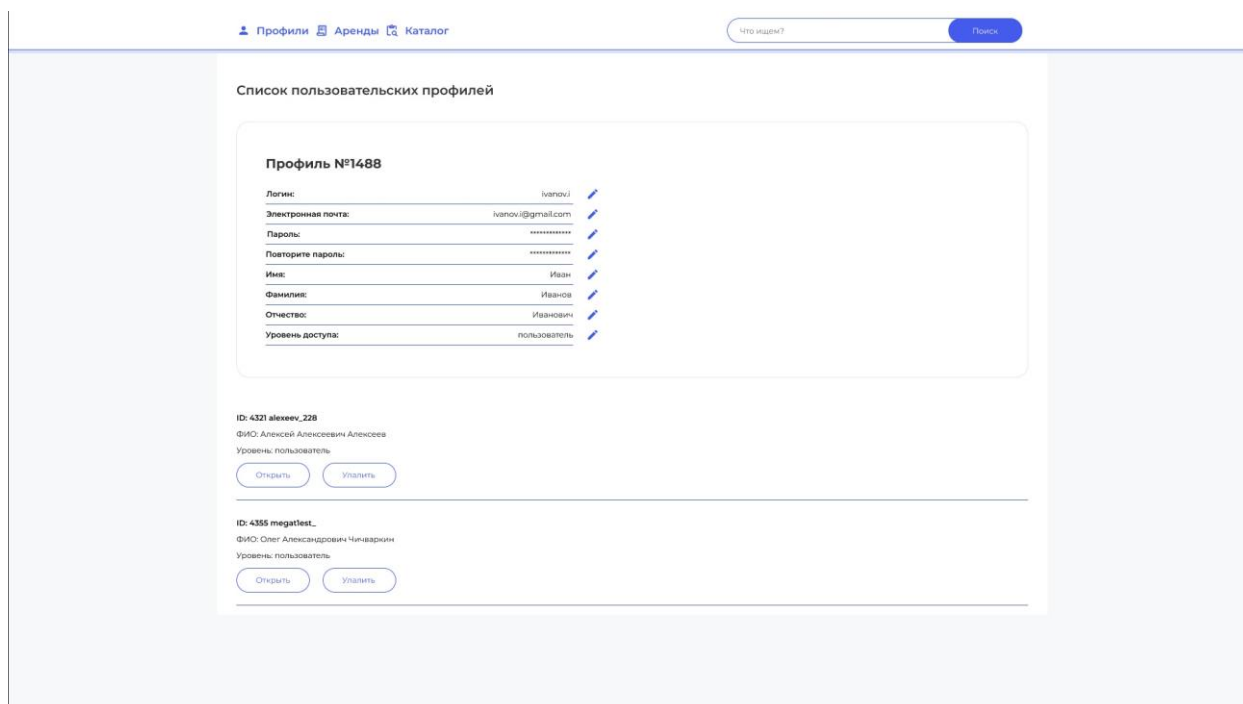


Рисунок 19 - Профили пользователей

4 Аналитика

В веб-приложении были определены следующие сценарии воронок конверсии:

1. Посетил главную страницу – Авторизовался – Нажал кнопку «Добавить» на карточке товара – Перешёл на страницу «Мои аренды» – Нажал кнопку «Удалить».

2. Посетил главную страницу – Нажал кнопку «Добавить» на карточке товара – Перешёл на страницу «Мои аренды» – Ввёл время аренды в заявке на аренду – Нажал кнопку «Оплатить».

3. Посетил главную страницу – Ввёл название инвентаря в поисковой строке – Нажал кнопку «Поиск».

Заключение

В заключение стоит отметить, что данное приложение полностью реализовано с учетом вышеописанной архитектуры, удовлетворяет схемам и диаграммам, а также выполняет все цели и поставленные задачи.

В результате было реализовано веб-приложение для взятия в аренду спортивного инвентаря. Были выполнены следующие задачи:

- проведён анализ предметной области, определены достоинства и недостатки существующих решений
- сформулированы требования, спроектирована архитектура приложения с учётом результатов анализа
- реализовано веб-приложение, соответствующее сформулированным ранее требованиям
- описаны результаты разработки, проведено тестирование

Также выполнены все подзадачи, входившие в реализацию веб-приложения.

Разработанное приложение отвечает всем заявленным требованиям.

Список используемых источников

1. Понимание различий Spring Framework и Spring Boot: сайт. – URL: <https://topjava.ru/blog/spring-framework-vs-spring-boot-differences> (дата обращения: 15.03.2022). – Текст: электронный.
2. Как проводить unit тестирование: сайт. – URL: <https://javarush.ru/groups/posts/2500-vse-o-unit-testing-metodiki-ponjatija-praktika> (дата обращения: 28.05.2022). – Текст: электронный.
3. Обзор преимуществ баз данных PostgreSQL: сайт. – URL https://market.cnews.ru/news/top/2022-04-25_obzor_baz_dannyh_postgresql/ (дата обращения – 12.05.2022). – Текст: электронный.