

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Мобильное приложение для проката инвентаря «Sportique»

Курсовой проект  
09.03.04 Программная инженерия  
Информационные системы и сетевые технологии

Допущено к защите в ГЭК \_\_\_\_\_.2023

Зав. Кафедрой\_\_\_\_\_С.Д. Махортов, д. ф.-м. н., профессор

Обучающийся\_\_\_\_\_Г.В. Прядченко, 3 курс, д/о

Обучающаяся\_\_\_\_\_М.В. Киселева, 3 курс, д/о

Обучающийся\_\_\_\_\_Н.Е. Мошкин, 3 курс, д/о

Руководитель\_\_\_\_\_В.С. Тарасов, ст. преподаватель

Воронеж 2023

## Содержание

Содержание.....	2
Введение.....	4
1 Постановка задачи.....	7
1.1 Требования к разрабатываемой системе.....	7
1.1.1 Функциональные требования .....	7
1.1.2 Требования к приложению и программному обеспечению .....	8
2 Анализ предметной области .....	9
2.1 Терминология (гlossарий) предметной области .....	9
2.2 Анализ рынка проката спортивного инвентаря .....	10
2.3 Обзор аналогов .....	11
2.3.1 Московский велопрокат Велобайк.....	11
2.3.2 Whoosh .....	14
2.3.3 Urent.....	16
2.3.4 Зеленый город .....	18
3 Реализация.....	22
3.1 Средства реализации.....	22
3.2 Входные - выходные данные (IDEF0).....	23
3.3 Диаграмма вариантов использования .....	25
3.3.1 Диаграмма пользователя .....	25
3.3.2 Диаграмма сотрудника .....	26
3.4 Диаграмма состояний .....	27
3.5 Диаграмма классов сущностей .....	28
3.6 Диаграмма объектов .....	29
3.7 Диаграмма развертываний .....	30

3.8 Диаграмма прецедентов .....	31
3.8.1 Диаграмма создания заказа.....	31
3.8.2 Диаграмма получения и оплаты заказа .....	32
3.9 Реализация серверной части приложения .....	33
3.9.1 Слой доступа к данным.....	33
3.9.2 Слой контроллеров .....	34
3.9.3 Слой моделей.....	35
3.9.4 Слой бизнес-логики .....	36
3.9.5 Механика работы приложения .....	37
3.10 Реализация клиентской части приложения .....	38
3.10.1 Интерфейс для пользователя .....	39
3.10.2 Интерфейс для работника .....	46
3.11 Диаграмма классов.....	51
3.11.1 Диаграмма классов модели.....	52
3.11.2 Диаграмма классов представления .....	53
3.11.3 Диаграмма классов моделей представления .....	53
3.12 Удаленная конфигурация приложения .....	54
3.13 Анализ .....	55
3.13.1 Анализ удержания пользователей.....	55
3.13.2 А/В тестирование .....	55
Заключение .....	57
Список используемых источников.....	58

## **Введение**

С появлением мобильных приложений многие области нашей жизни стали более доступными и удобными. Одной из таких областей стал спорт и отдых. В настоящее время многие люди предпочитают заниматься спортом и проводить свободное время на природе. Использование спортивного инвентаря, такого как велосипеды, скейтборды, ролики, лыжи и др., становится все более востребованным при активном отдыхе. Однако, далеко ни у каждого есть эти предметы, чтобы полноценно заняться любимым делом. Разработка мобильного приложения для проката спортивного инвентаря может решить эту проблему.

В данной курсовой работе будет рассмотрена тема создания приложения для проката спортивного инвентаря. Будет проведен анализ существующих приложений и определены особенности проката спортивного инвентаря. Также мы рассмотрим основные требования к приложению, а также принципы его работы. Будет рассмотрена архитектура приложения, включая его структуру и функциональность.

Цель работы - создание полноценного мобильного приложения, которое позволит пользователям организовывать прокат спортивного инвентаря со своего мобильного устройства, а также удобно и быстро выбирать необходимое оборудование из имеющегося ассортимента. Создание такого приложения позволит повысить комфорт и удобство для любителей активного отдыха и спортивных мероприятий.

Актуальность темы данной курсовой работы обусловлена спросом на бронирование услуг и товаров онлайн, что подтверждают исследования [5]. Часть результатов представлена ниже.

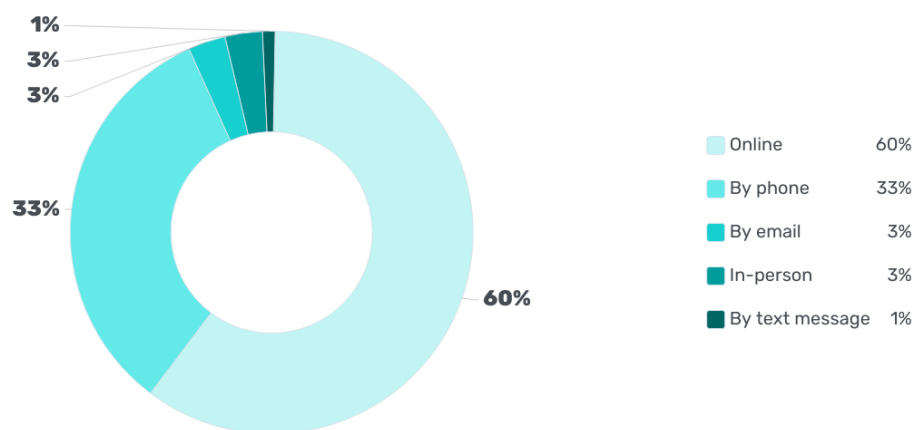


Рисунок 1 - График предпочтений в способе аренды [5]

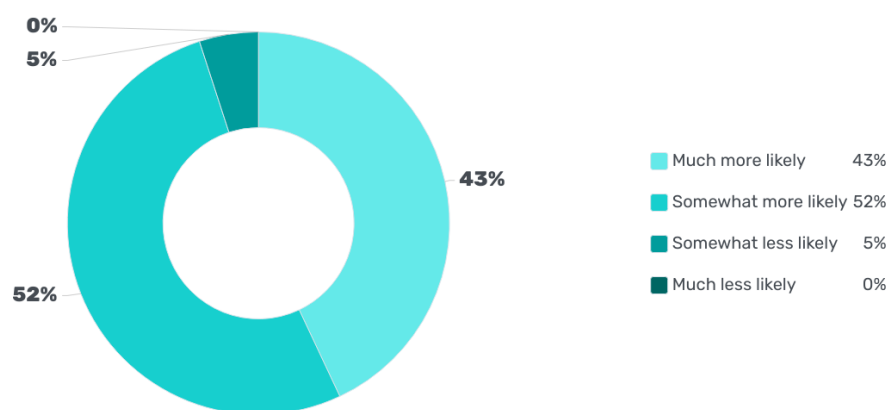


Рисунок 2 - Как часто люди выбирают онлайн бронирование по сравнению с другими [5]

Также можно сказать про прирост людей, которые для занятия спортом обращаются в организации, а не делают это дома [6], и прирост количества спортивных сооружений [7].

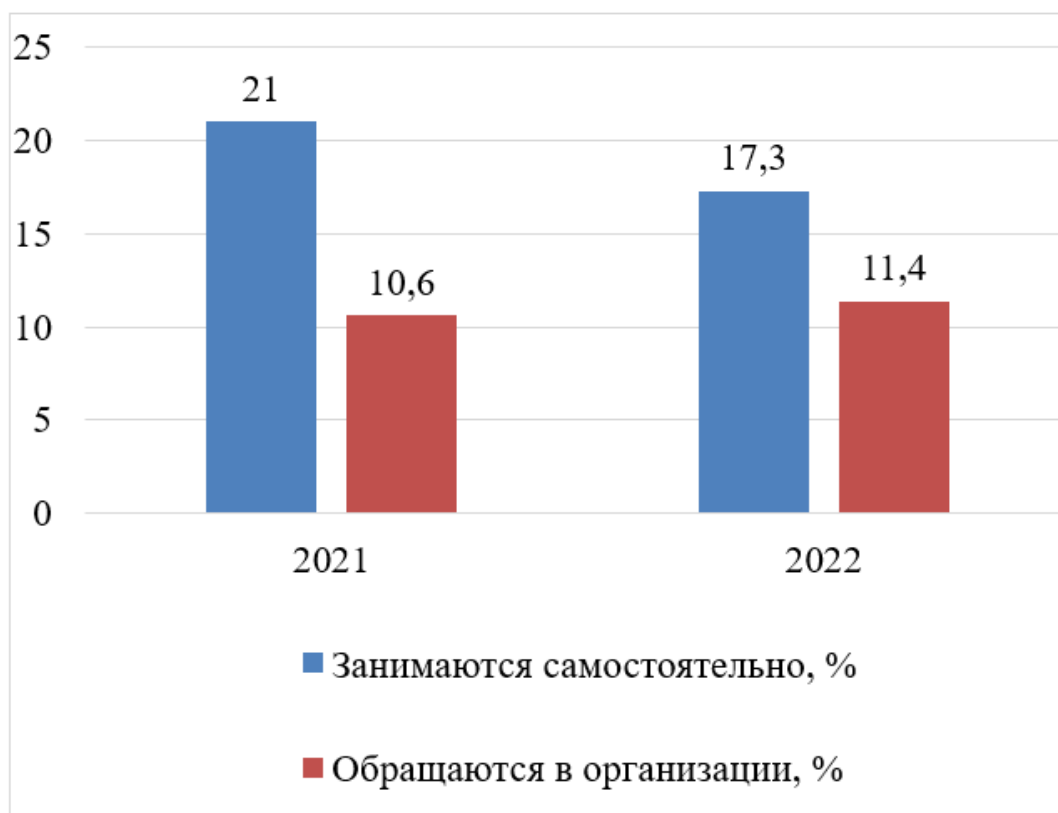


Рисунок 3 - График предпочтений в отношении занятия спортом: самостоятельно или в организации

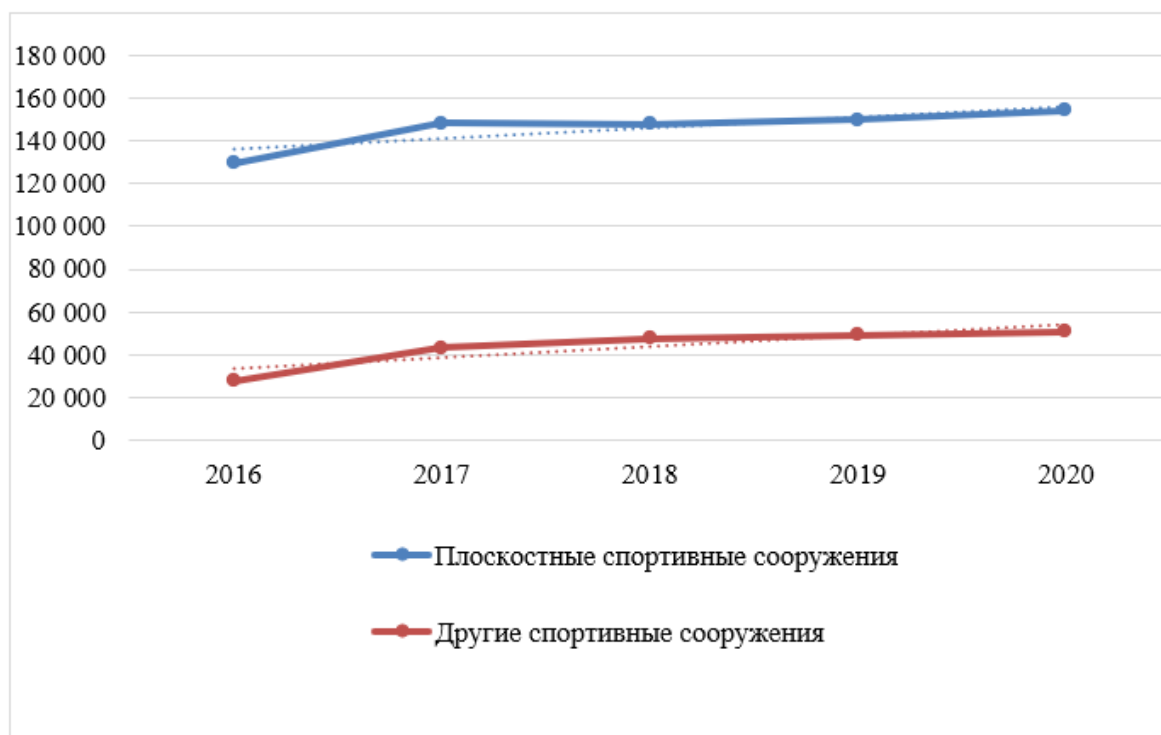


Рисунок 4 - График количества спортивных сооружений

## **1 Постановка задачи**

Целью данного курсового проекта является создание мобильного приложения для проката спортивного инвентаря «Sportique». Данное мобильное приложение разрабатывается с целью:

- Введения в систему проката возможности безналичной оплаты, в частности, таких возможностей, как оплата по QR-коду и формирование электронных чеков;
- Реализации возможности для клиента выбора инвентаря и бронирования его через мобильное приложение, а также для получения инвентаря по QR-коду;
- Автоматического подсчета времени и стоимости сессии;
- Реализации механизма примерки.

### **1.1 Требования к разрабатываемой системе**

#### **1.1.1 Функциональные требования**

Разрабатываемый проект должен обеспечивать следующие возможности:

- Просматривать каталог товаров и выбирать необходимый;
- Бронировать несколько товаров за раз;
- Бронирования товаров на определенную дату;
- Оплаты по qr коду;
- Отмены заказа;
- Изменения и отмены заказа со стороны работника;
- Зарегистрироваться и авторизоваться в приложении;

- Просмотра истории заказов;
- Просмотра всех последних активных заказов со стороны работника;
- Узнавать информацию о заказе;
- Подтвердить заказ со стороны работника;
- Поиска заказов для работника по фильтру.

### **1.1.2 Требования к приложению и программному обеспечению**

К разрабатываемому приложению выдвигаются следующие требования:

- Приложение должно корректно работать на устройствах, работающих на операционной системе Android 8.0 и новее;
- Реализовывать основные задачи, стоящие перед данным проектом;
- Созданное приложение должно иметь архитектуру, соответствующую шаблону Клиент-Серверного приложения, а также иметь разделение на Back-end и клиентскую часть, взаимодействие между которыми должно происходить с помощью REST API.



## **2 Анализ предметной области**

### **2.1 Терминология (гlossарий) предметной области**

- Сервер, серверная часть – компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач;
- Клиент, клиентская сторона – в данном проекте, мобильное устройство с установленным на него приложением, предоставляет возможности пользователю взаимодействовать со всей системой;
- Front-end – клиентская часть приложения. Отвечает за получение информации с программно-аппаратной части и отображение ее на устройстве пользователя. В нашем проекте, это само android приложение;
- Back-end – программно-аппаратная часть приложения. Отвечает за функционирование внутренней (серверной) части приложения;
- Пользователь, клиент – человек, пользующийся функционалом приложения;
- Заказ – бронирование товара на определенный день;
- Корзина – виртуальное хранилище для выбранных товаров;
- Сессия проката – время, в которое клиент/клиенты пользуются инвентарем;
- Работник – человек, который выдает заказ и выполняет все организационные действия (изменяет заказ, отменяет, выдает qr для оплаты);
- Статус заказа – состояние, в котором находится заказ;

- Активный заказ – статус заказа, когда человек получил свой инвентарь и пользуется им, но до окончания сеанса;
- Инвентарь – спортивное оборудование, предназначенное для выдачи в аренду;
- Сформированный заказ – заказ, с забронированными, но еще не полученными товарами.

## **2.2 Анализ рынка проката спортивного инвентаря**

Рынок проката спортивного инвентаря является конкурентным и разнообразным. Он привлекает многих любителей спорта, которые не хотят покупать дорогостоящее оборудование, но хотят заниматься спортом. Рынок также дает возможность получать доход людям, которые предоставляют услуги по прокату.

Основным конкурентным преимуществом на рынке проката спортивного инвентаря является широкий выбор спортивных товаров и гибкие условия аренды. Например, многие компании предлагают аренду на час, день, неделю или больший период времени. Кроме того, рынок проката спортивного инвентаря использует маркетинговые стратегии, чтобы привлечь клиентов, такие как скидки для постоянных клиентов.

Однако, на рынок проката спортивного инвентаря влияют и некоторые ограничения. Например, сезонность спроса – летом, когда больше людей занимаются спортом на открытом воздухе, спрос на прокат спортивного инвентаря выше, а зимой спрос снижается. Кроме того, сложность обслуживания некоторых видов спортивного инвентаря, могут повышать затраты на его содержание.

В целом, рынок проката спортивного инвентаря перспективен и продолжает расти за счет повышения интереса к спорту. Однако, как и в любом бизнесе, конкуренция не дремлет, и успешная деятельность на рынке

проката требует дальнейшего развития, совершенствования услуг и усиления маркетинговых стратегий.

### **2.3 Обзор аналогов**

Перед тем как начинать разработку нового продукта, сперва нужно проанализировать похожие приложения. Ведь зачастую придумать что-то кардинально новое не представляется возможным, так как всегда найдется аналог предоставляющий похожий функционал. Поэтому нужно подробно изучить такие продукты, выделить их преимущества и недостатки, чтобы при разработке своего приложения иметь представление как должен выглядеть готовый продукт удобный для пользователя.

Большинство сервисов для проката спортивного инвентаря представляют собой CRM-системы служащие для организации и учета бизнес-процессов, что далеко от нашей темы. Ниже мы представим примеры «шеринговых» онлайн сервисов для проката электросамокатов и велосипедов. Ключевое отличие таких сервисов от нашего решения – это то, что взять и поставить на стоянку самокат или велосипед можно только на специальных зонах, отмеченных на карте в приложении. Для этого не требуется дополнительный персонал, клиент делает все сам в приложении.

#### **2.3.1 Московский велопрокат Велобайк**

«Велобайк» — сеть городского общественного велопроката в Москве. Московский велопрокат работает в круглосуточном режиме обычно в период с 25 апреля по 31 октября. Сервис служит для коротких поездок по городу.

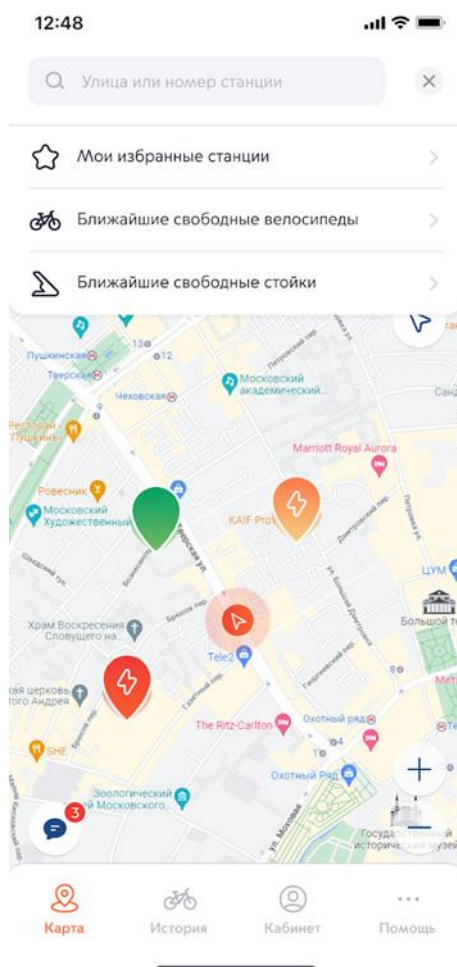


Рисунок 5 - Пример интерфейса Велобайк

Возможности приложения Велобайк:

- Найти ближайшую станцию велопроката и проверить наличие свободных велосипедов и парковочных мест;
- Зарегистрироваться в системе и купить удобный доступ;
- Всегда знать точное время и стоимость своей поездки;
- Отслеживать историю поездок.

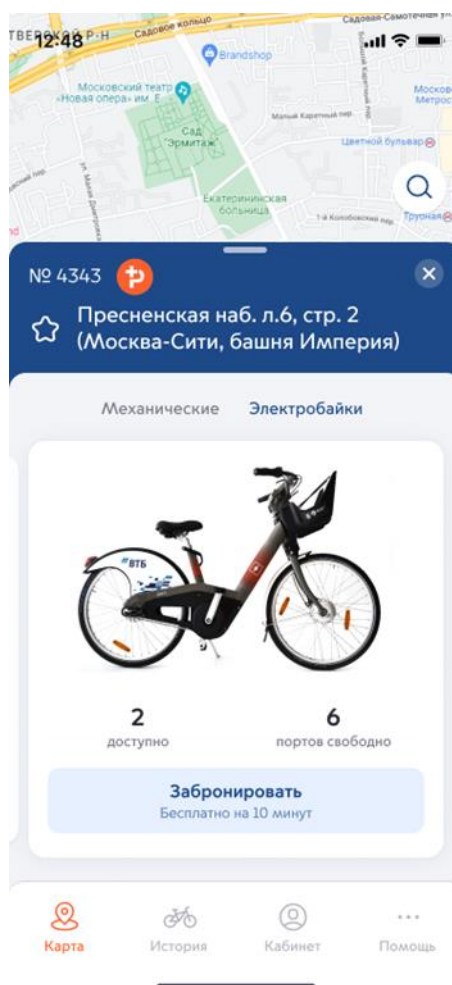


Рисунок 6 - Выбор велосипеда в Велобайк

Преимущества приложения Велобайк:

- Удобная оплата, существует возможность бесплатной поездки в течении 30 минут между двумя станциями;
- Возможность выбрать механический или электрический велосипед;
- Отзывчивая служба поддержки, по отзывам в интернете.

Недостатки приложения Велобайк:

- В ассортименте представлены только велосипеды;
- Необходимость в приобретении одного из тарифов, вместо поминутной оплаты у других сервисов;

— Нехватка вело парковок и велостоек на них.

### 2.3.2 Whoosh

Одним из лидеров на рынке проката электросамокатов является Whoosh. Whoosh – «кикшеринг», имеющий самую большую зону покрытия среди московских сервисов. На данный момент самокаты Whoosh можно найти не только в столице, а еще в Санкт-Петербурге, Сочи и других 14 городах России.

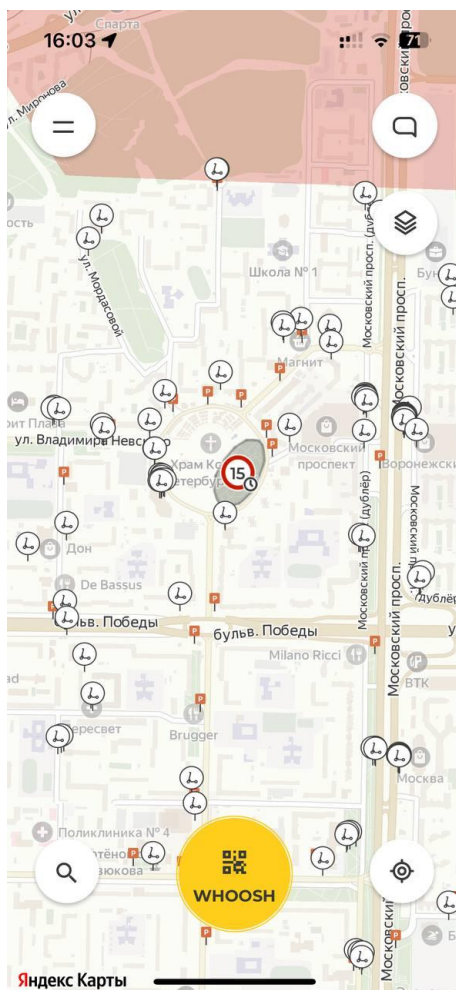


Рисунок 7 - Пример интерфейса приложения Whoosh

Возможности приложения Whoosh:

- На карте в приложении можно найти ближайший самокат;
- Чтобы разблокировать самокат нужно через приложение отсканировать QR-код на самокате;

- Можно следить за ходом поездки: количество минут, скорость, зоны аренды и другая важная информация;
- Завершить поездку можно оставив самокат на парковке, отмеченной на карте знаком Р.

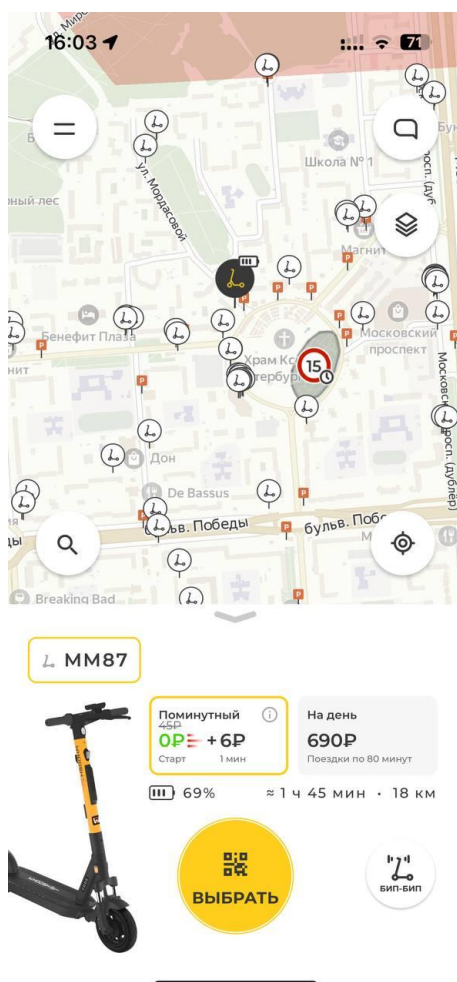


Рисунок 8 - Выбор самоката в приложении Whoosh

Преимущества приложения Whoosh:

- Представлен во многих городах России;
- Большое количество парковок на карте;
- Возможность построения маршрута внутри приложения;
- Возможность выбора режима езды в приложении.

Недостатки приложения Whoosh:

- В сервисе представлены только самокаты;
- Высокая стоимость аренды по сравнению с другими сервисами;
- Большой взнос за начало поездки по сравнению с другими сервисами.

### 2.3.3 Urent

Вторым по популярности сервисом по прокату является Urent. Urent — это сервис 2 в 1, который позволяет пользователю с помощью мобильного приложения арендовать электросамокат или велосипед. Urent представлен в 100 городах России и имеет самый большой парк электросамокатов.

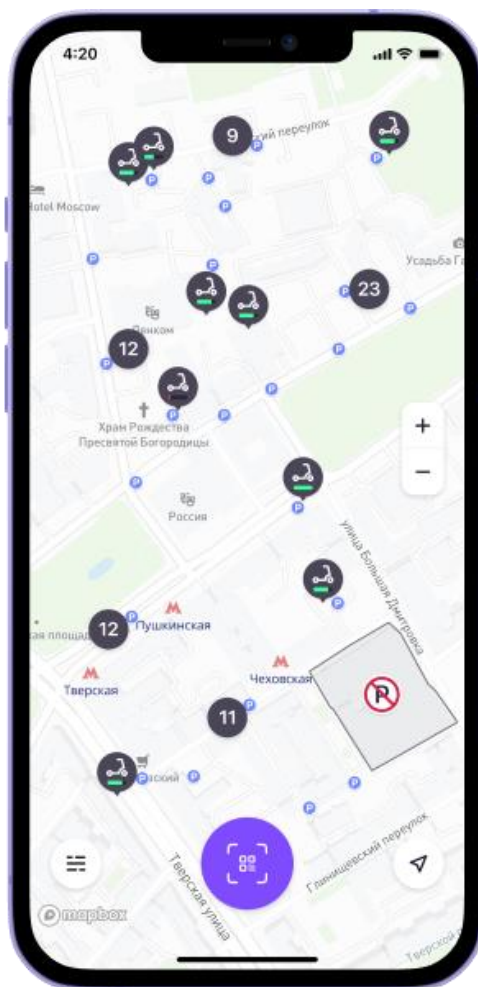


Рисунок 9 - Пример интерфейса приложения Urent



## Возможности приложения Urent:

- Ближайший самокат можно найти на карте в приложении;
- Чтобы начать поездку нужно отсканировать QR-код на руле, выбрать тариф и нажать на кнопку "Старт";
- На карте отмечено сотни парковок, где можно удобно застегнуть самокат, чтобы он никому не мешал, и завершить аренду;
- Можно отслеживать свою историю поездок.

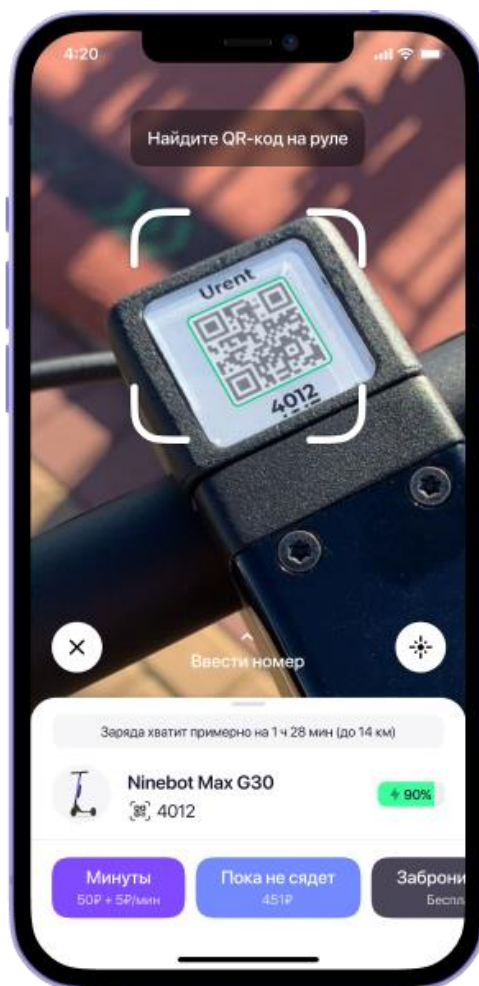


Рисунок 10 - Выбор самоката в приложении Urent

## Преимущества приложения Urent:

- Возможность аренды и самокатов и велосипедов;

- Мультиаренда до 5 самокатов и/или велосипедов с одного аккаунта;
- Доступен во многих городах России.

Недостатки приложения Urent:

- Большое количество неисправных самокатов, по отзывам пользователей в интернете;
- Плохая работа службы поддержки и отсутствие горячей линии, по отзывам пользователей в интернете.

#### **2.3.4 Зеленый город**

Зеленый город – приложение для аренды велосипедов в городе Краснодар. Это приложение так же является «шеринговым» сервисом. В черте города можно найти зоны для парковки, в которых можно взять велосипед, а затем после завершения поездки оставить.

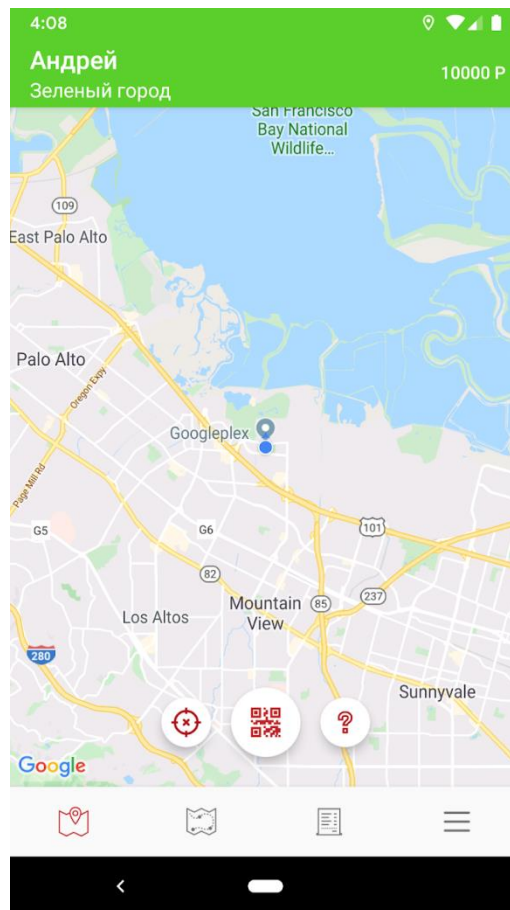


Рисунок 11 - Пример интерфейса приложения Зеленый город

Возможности приложения Зеленый город:

- На карте вы сможете найти свободные байки;
- Можете посмотреть рейтинг пользователей;
- Можете посмотреть историю поездок;
- Нужно отсканировать QR-код для начала поездки.

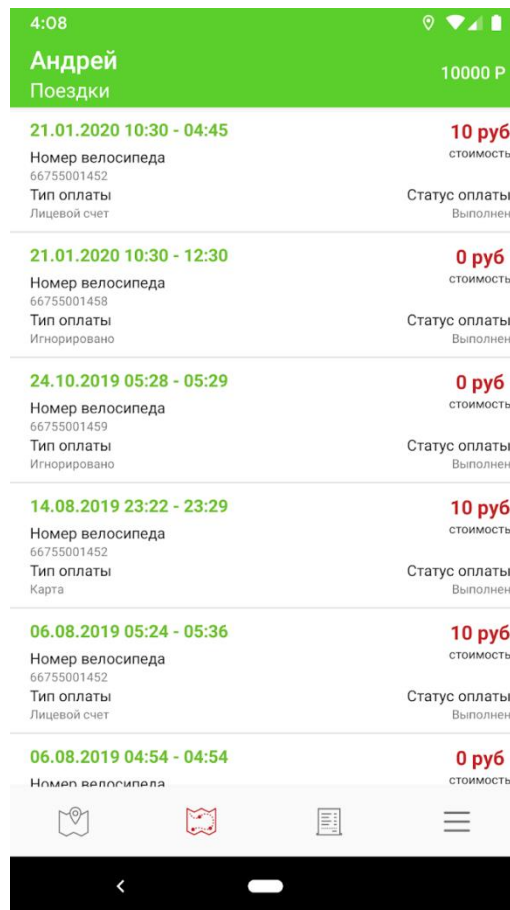


Рисунок 12 - История поездок в приложении Зеленый город

Преимущества приложения Зеленый город:

- Довольно простой интерфейс;
- Низкая цена аренды по сравнению с другими сервисами;
- Удобные велосипеды для длительных прогулок.

Недостатки приложения Зеленый город:

- Возможность аренды только велосипедов;
- Множество багов в приложении, пользователи жалуются на частые ошибки в работе приложения;
- На велосипедах отсутствует возможность переключения скоростей, что сильно ограничивает максимальную скорость передвижения;

- Безопасное передвижение доступно только на тротуарах и велослорожках;
- Малый штат сотрудников технической поддержки, из-за чего часто можно не получить поддержку.

Подведем итоги нашего обзора аналогов. Представленные здесь приложения ограничены по ассортименту предлагаемого в аренду инвентаря, это велосипеды или электросамокаты. Причиной тому специфика «шеринговых сервисов», так как не каждый инвентарь таким образом можно сдавать в аренду. В нашем же приложении широкий выбор спортивного инвентаря для аренды. Изучая аналоги, мы подчеркнули для себя некоторые возможности, которые затем добавили в свое приложение. Такие как оплата по QR-коду и некоторые детали интерфейса.

### 3 Реализация

#### 3.1 Средства реализации

В качестве средств реализации мобильного приложения были выбраны:

Для сервера:

- Язык программирования Java 17 версия;
- Фреймворк Spring Boot;
- СУБД PostgreSQL;
- Инструмент для создания документации API Swagger.

Для клиента:

- Flutter SDK версии 2.19.6.

Данные технологии были выбраны исходя возможностей, которые они дают для решения задач. В качестве преимуществ выбранных технологий можно отметить:

Для Java и фреймворка SpringBoot:

- Готовые решения для реализации RestFul архитектуры;
- Удобные инструменты для работы с PostgreSQL БД;
- Готовые встроенные серверы (Tomcat), обеспечивающие ускоренное и более продуктивное развертывание приложений.

Для PostgreSQL БД:

- Предоставляет большой бесплатный функционал;
- Надежная и высокопроизводительная.

Для Flutter:

- Мультиплатформенность;
- Понятная и полная документация;
- Возможность быстро проектировать мобильные приложения.

Для развертывания приложения использовались следующие технологии:

- Клиент Certbot для создания и получения SSL сертификата;
- Docker для автоматизации развертывания приложения;
- Nginx – прокси-сервер с поддержкой SSL.

### 3.2 Входные - выходные данные (IDEF0)

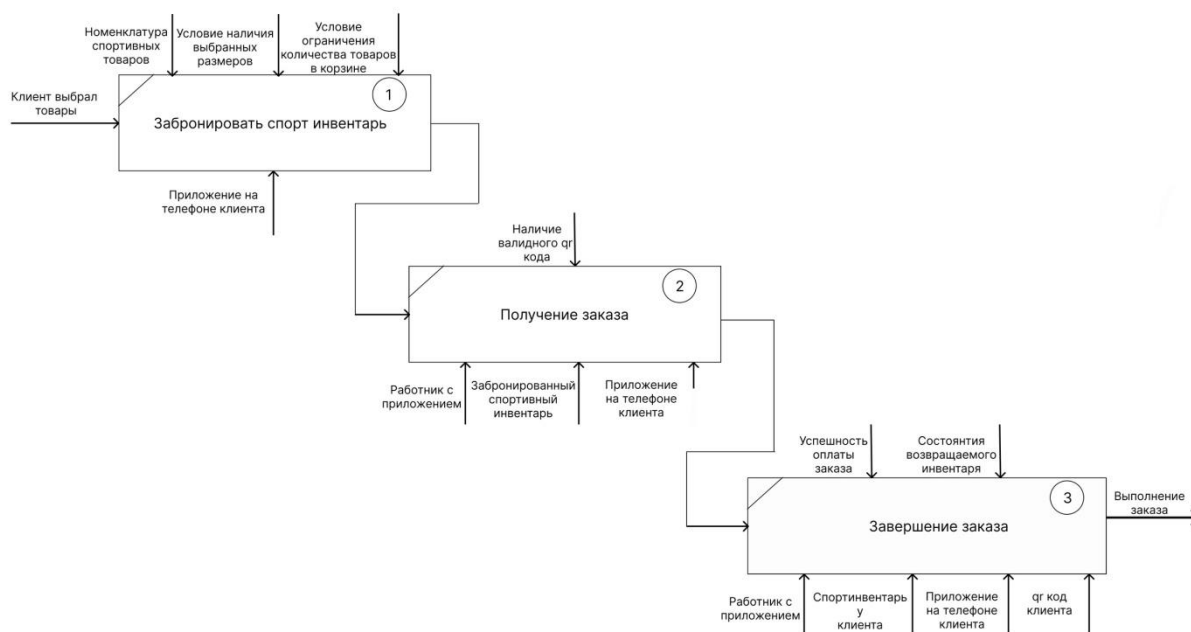


Рисунок 13 - Диаграмма IDEF0

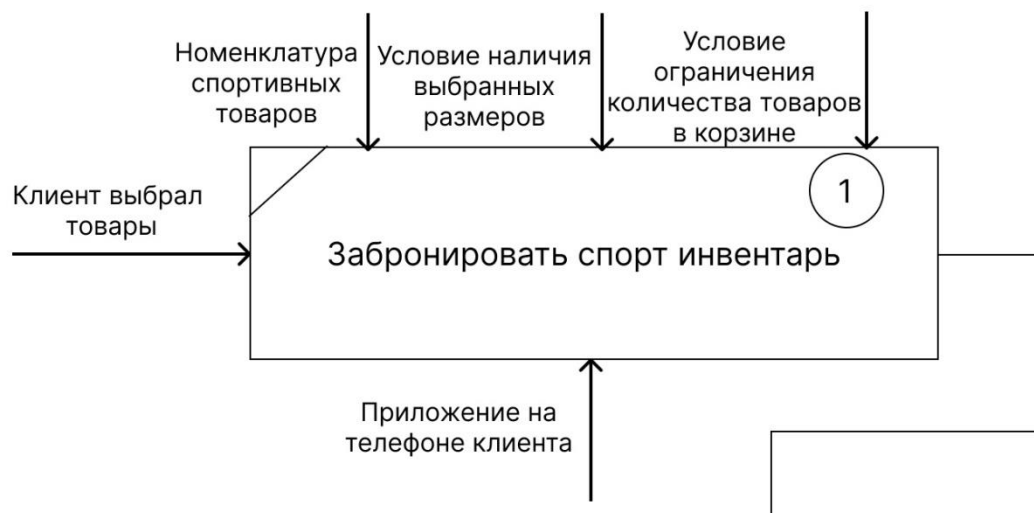


Рисунок 14 - Бронирование

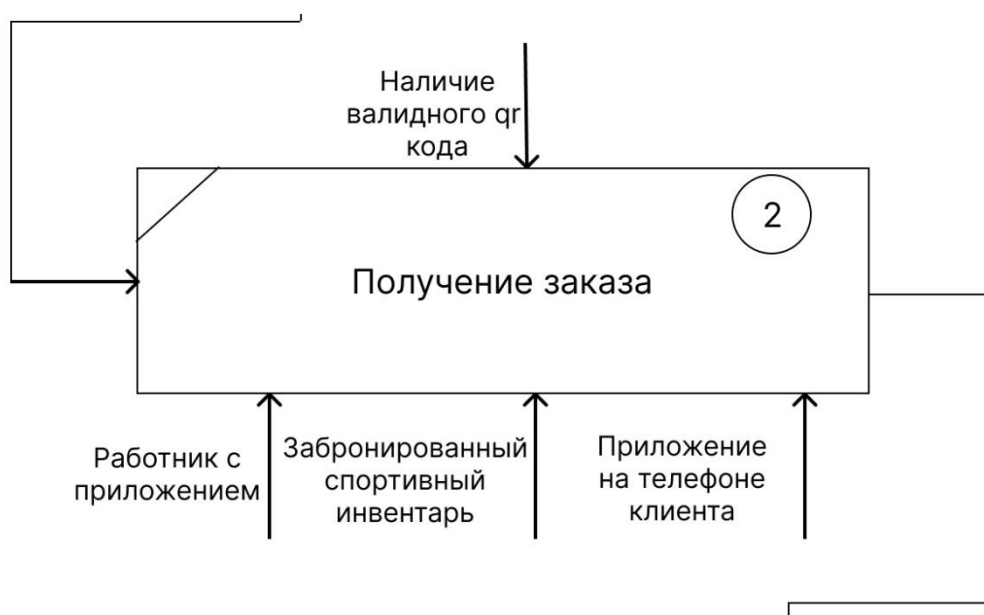


Рисунок 15 - Получение заказа



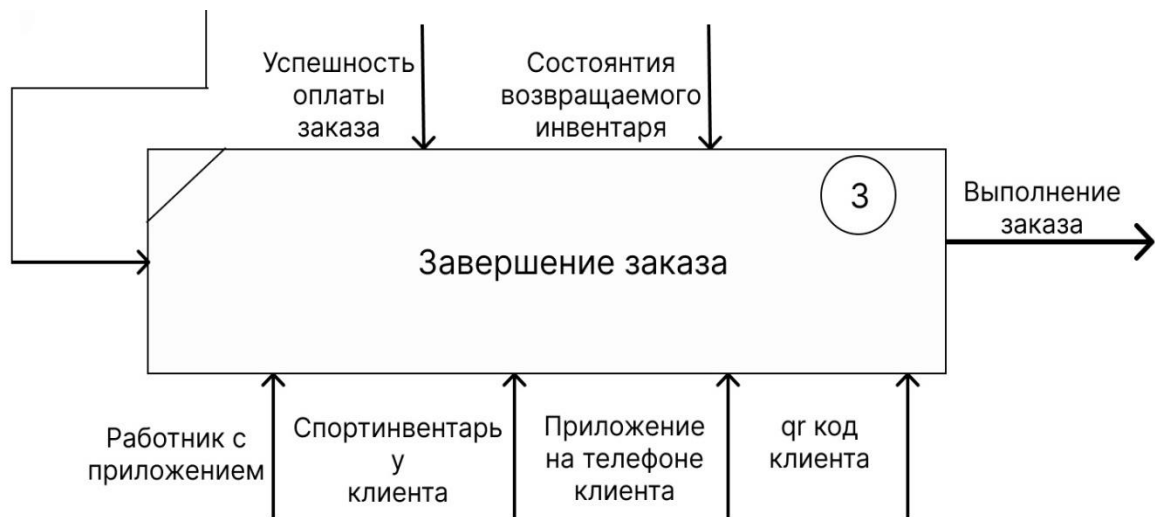


Рисунок 16 - Завершение заказа

### 3.3 Диаграмма вариантов использования

#### 3.3.1 Диаграмма пользователя

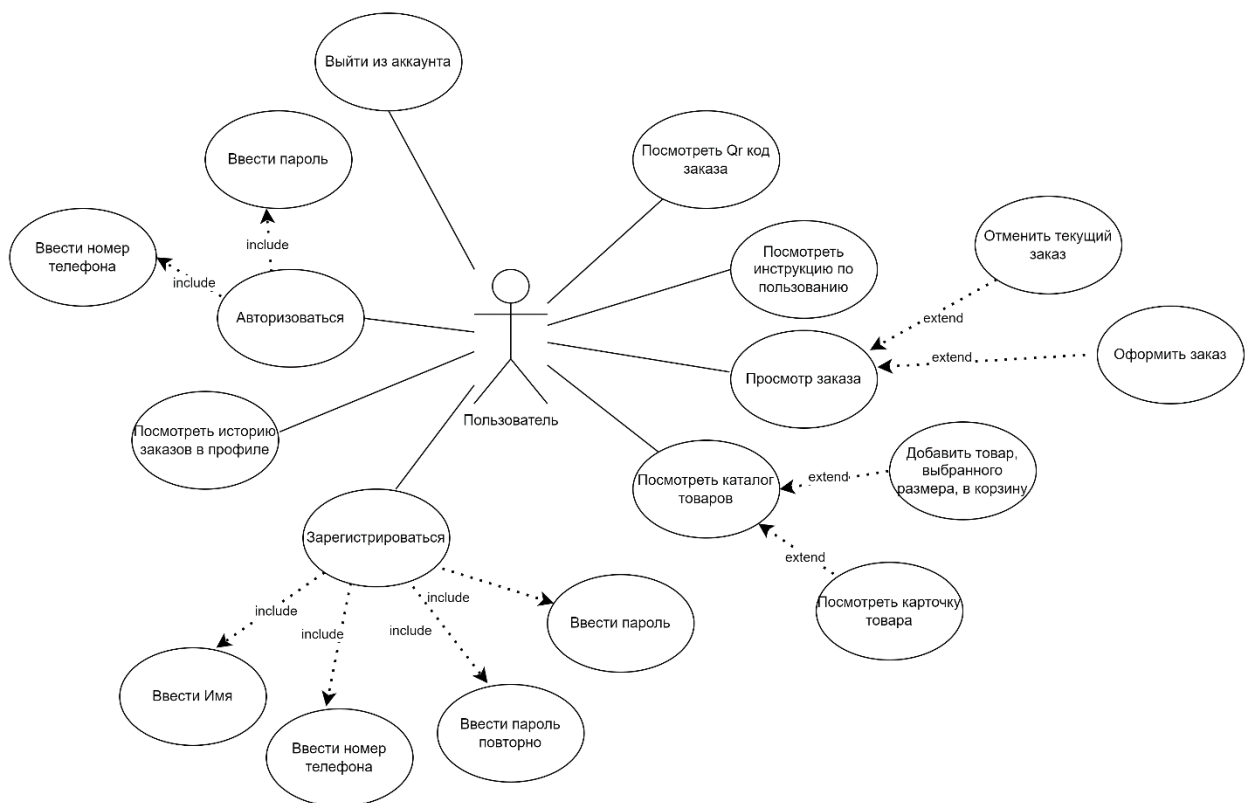


Рисунок 17 - Диаграмма вариантов использования для Пользователя

### 3.3.2 Диаграмма сотрудника

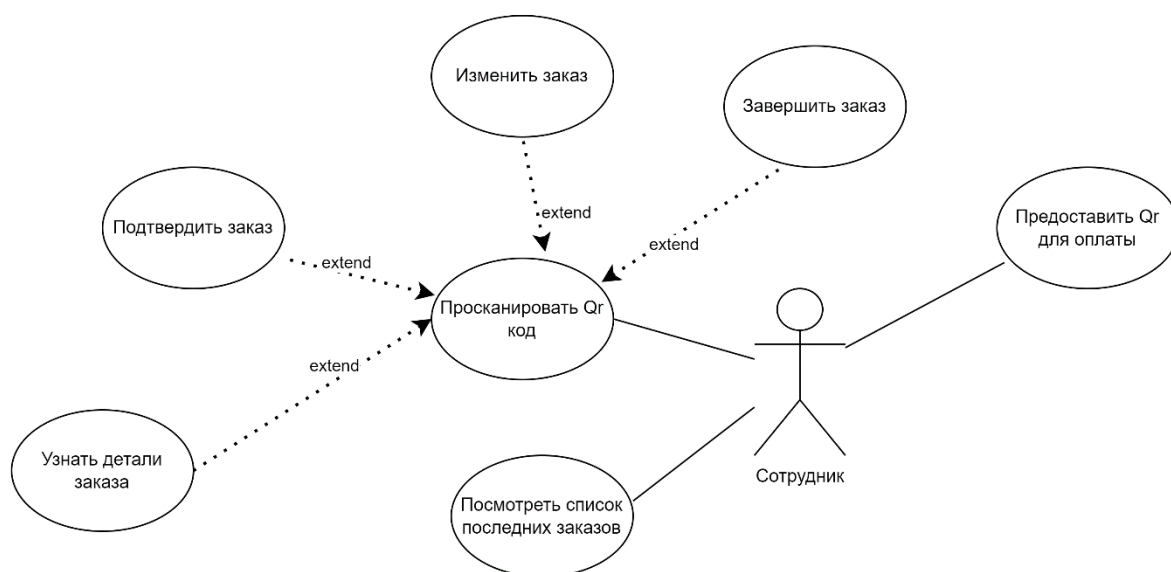


Рисунок 18 - Диаграмма вариантов использования для Сотрудника

### 3.4 Диаграмма состояний

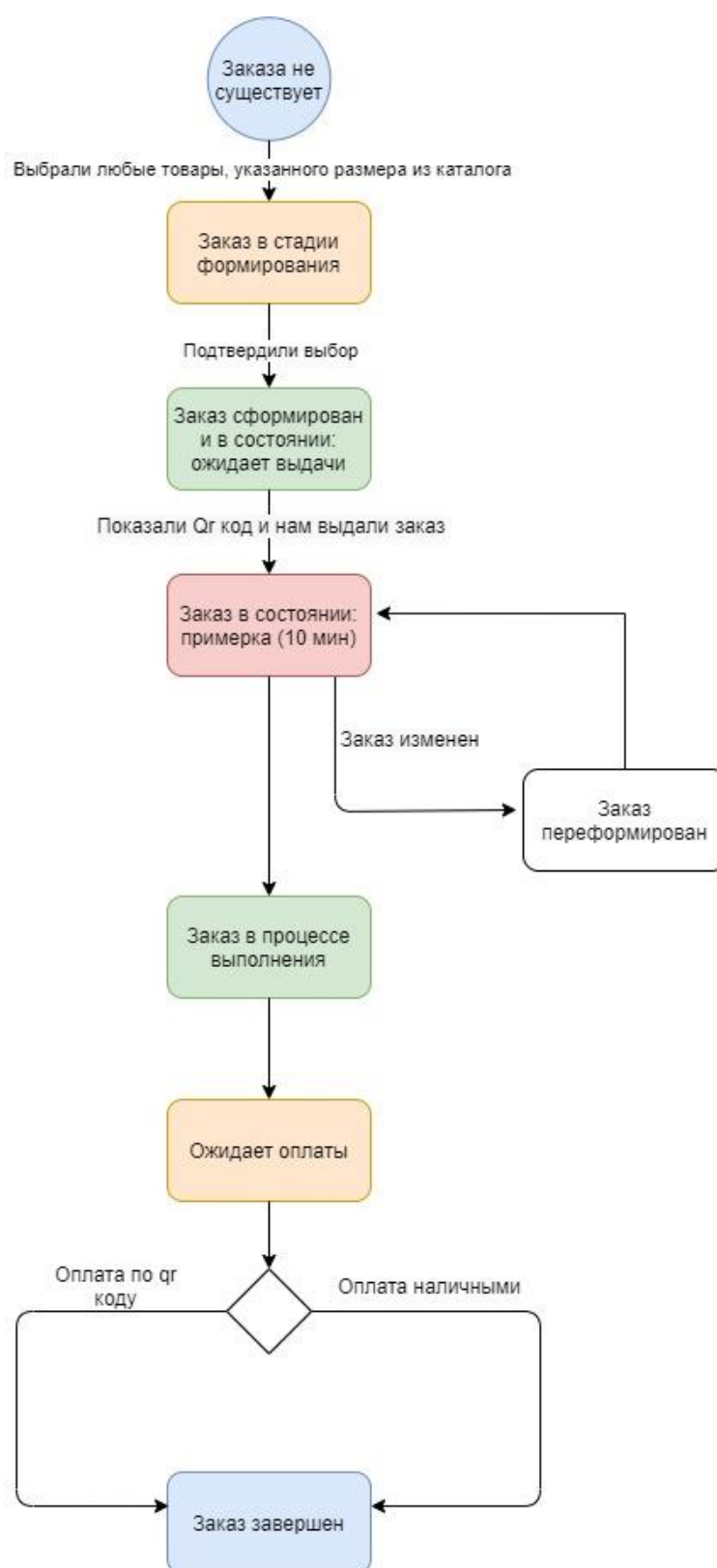


Рисунок 19 - Диаграмма состояний

### 3.5 Диаграмма классов сущностей

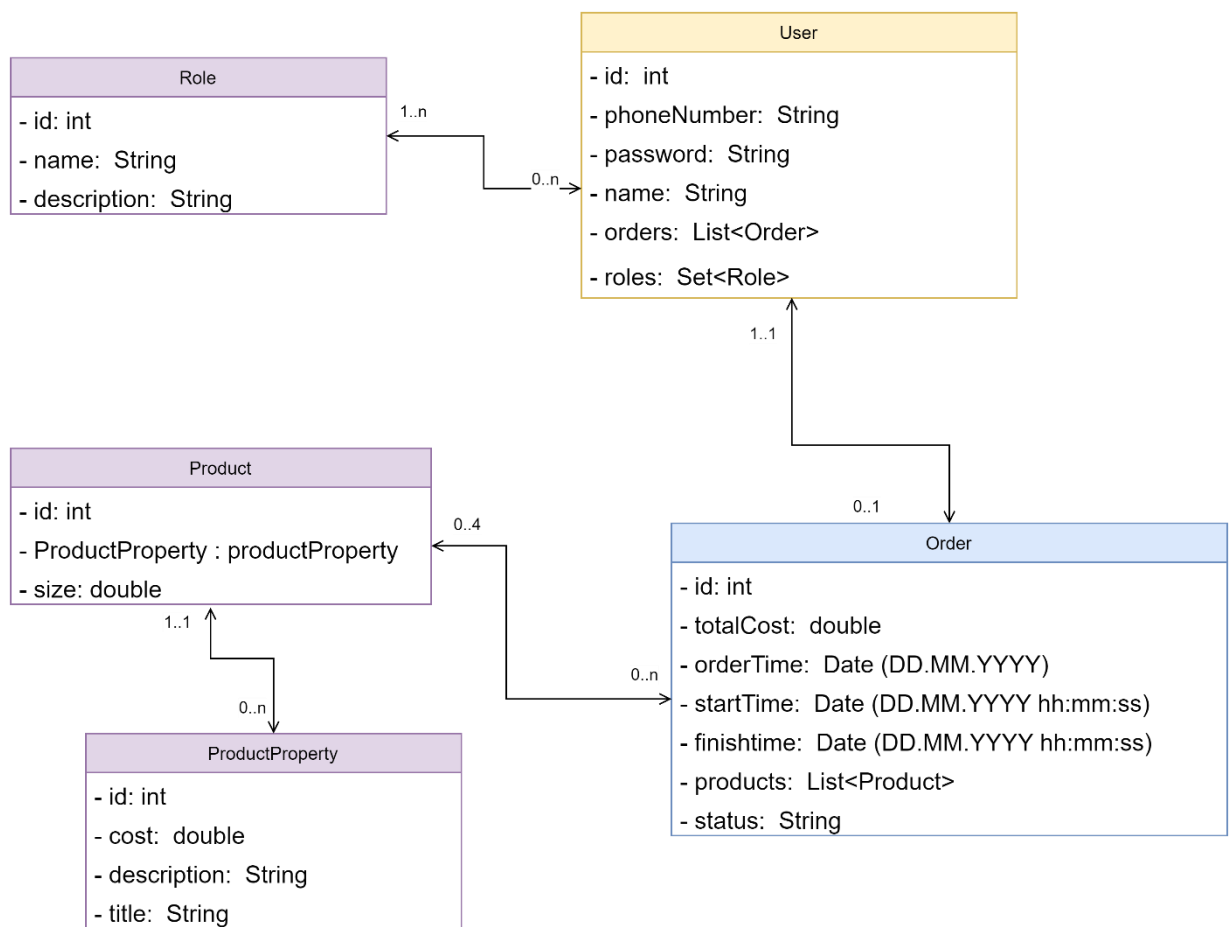


Рисунок 20 - Диаграмма классов сущностей

### 3.6 Диаграмма объектов

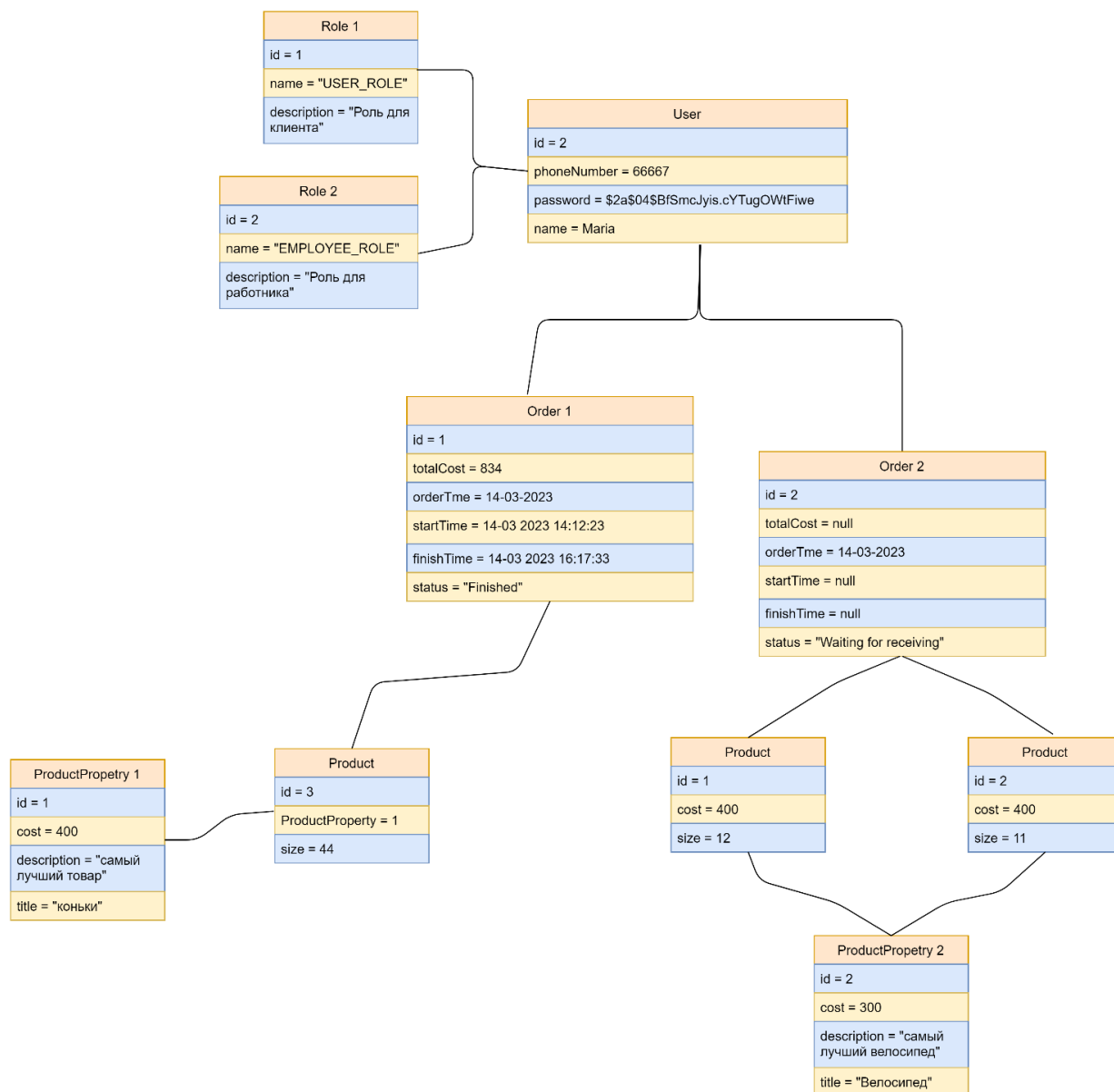


Рисунок 21 - Диаграмма объектов

### 3.7 Диаграмма развертываний

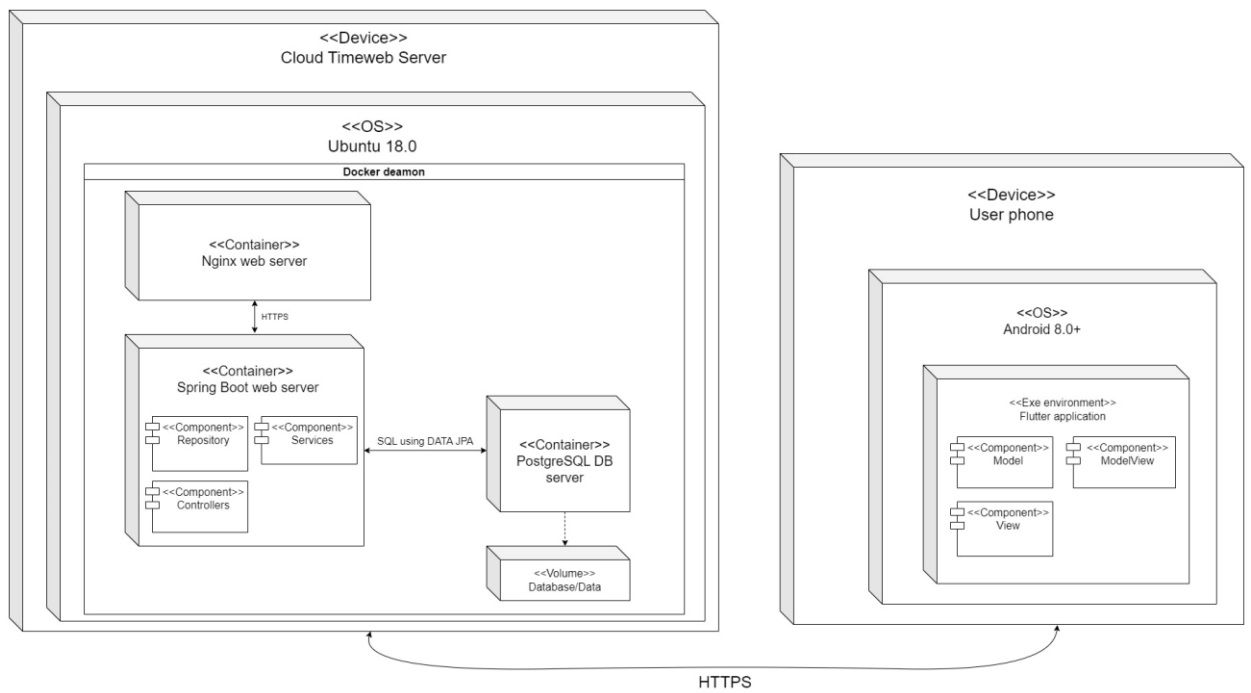


Рисунок 22 - Диаграмма развертывания

## 3.8 Диаграмма прецедентов

### 3.8.1 Диаграмма создания заказа

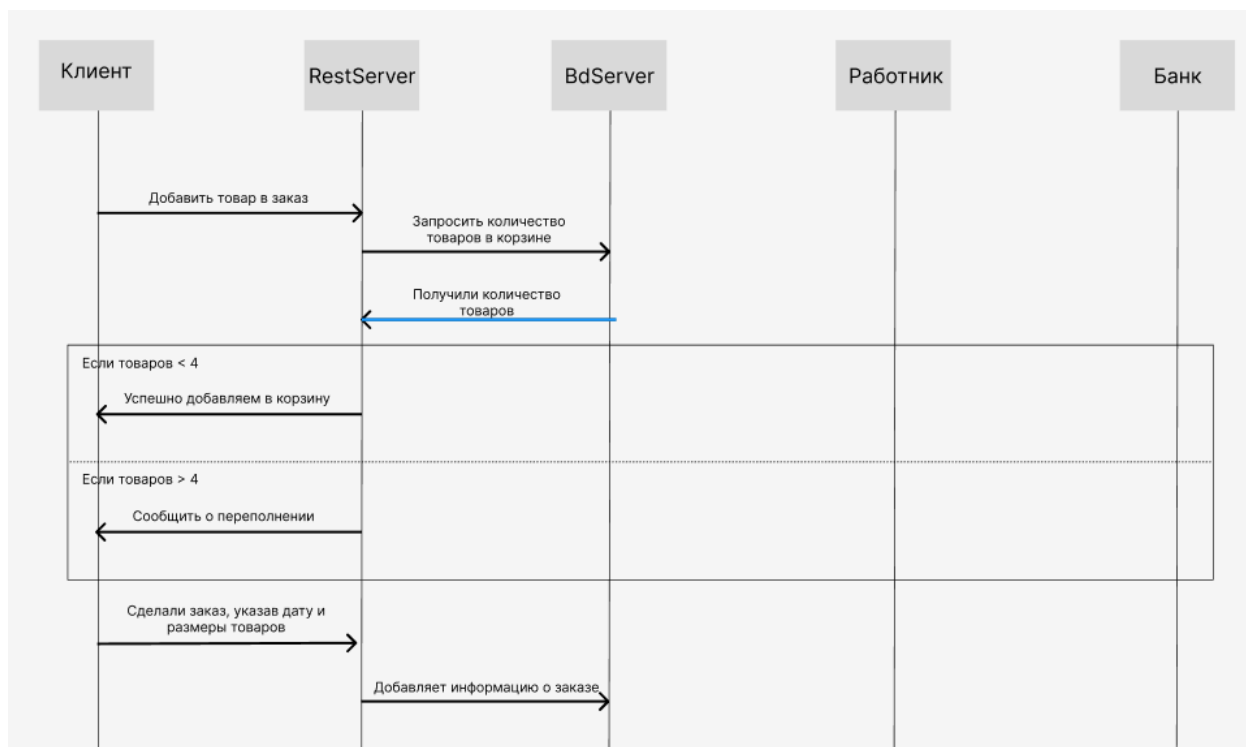


Рисунок 23 - Диаграмма создания заказа

### 3.8.2 Диаграмма получения и оплаты заказа

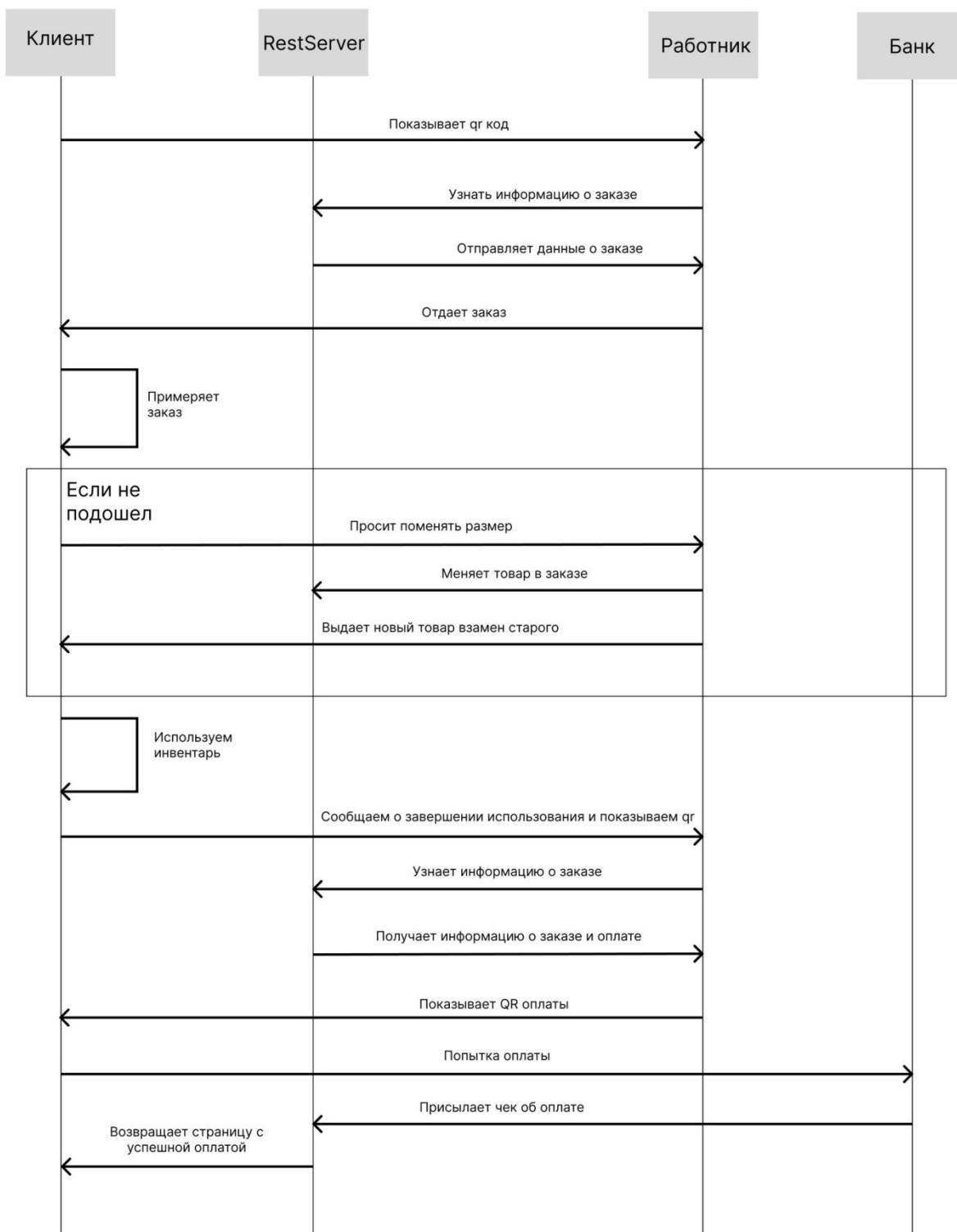


Рисунок 24 - Диаграмма совершения заказа



### 3.9 Реализация серверной части приложения

Для исполнения серверной части используем Spring Boot фреймворк для разработки веб-приложений, который облегчает создание приложений на основе архитектуры MVC (Model-View-Controller). MVC — это популярный шаблон проектирования, который разделяет приложение на три основных компонента: модель (Model), представление (View) и контроллер (Controller). Модель представляет собой данные приложения и логику их обработки. Представление отвечает за отображение данных пользователю. Контроллер обрабатывает запросы от пользователя, связывает модель и представление, и определяет логику обработки запросов.

К плюсам MVC архитектуры можно отнести:

- Разделение ответственности между моделью, представлением и контроллером;
- Улучшенная читаемость и поддерживаемость кода;
- Легкая замена или модификация компонентов без изменения других частей приложения;
- Улучшенная масштабируемость и возможность повторного использования компонентов.

Таким образом, Spring Boot и MVC архитектура предоставляют удобный и эффективный способ разработки веб-приложений, позволяя разделить логику, данные и представление, и обеспечивая удобство использования и поддержки кода.

#### 3.9.1 Слой доступа к данным

Слой доступа к данным отвечает за взаимодействие с базой данных или другими источниками данных. В приложении используется Spring Data JPA, что позволяет нам упростить доступ к данным и уменьшить объем написания повторяющегося кода. Для каждой сущности в приложении был создан

соответствующий репозиторий, наследующийся от интерфейса JpaRepository. JpaRepository предоставляет широкий набор методов для основных операций доступа к данным, включая сохранение, поиск по id и т.д.

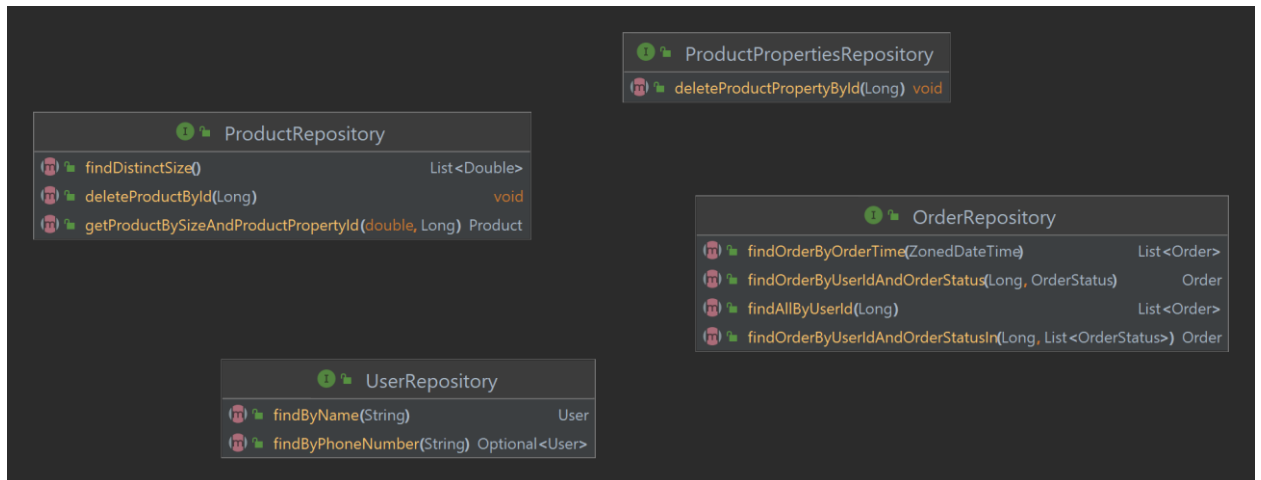


Рисунок 26 - Реализация интерфейсов

### 3.9.2 Слой контроллеров

Слой контроллеров отвечает за обработку входящих запросов от клиентов на определенный маппинг и возвращает ответ в виде ResponceEntityRestAPI. Для каждой из сущностей были написаны контроллеры, методы которых отвечают за необходимые приложению действия с этими сущностями (EmployeeController, UserController, OrderController). На рисунке 27 можно увидеть реализацию контроллеров.

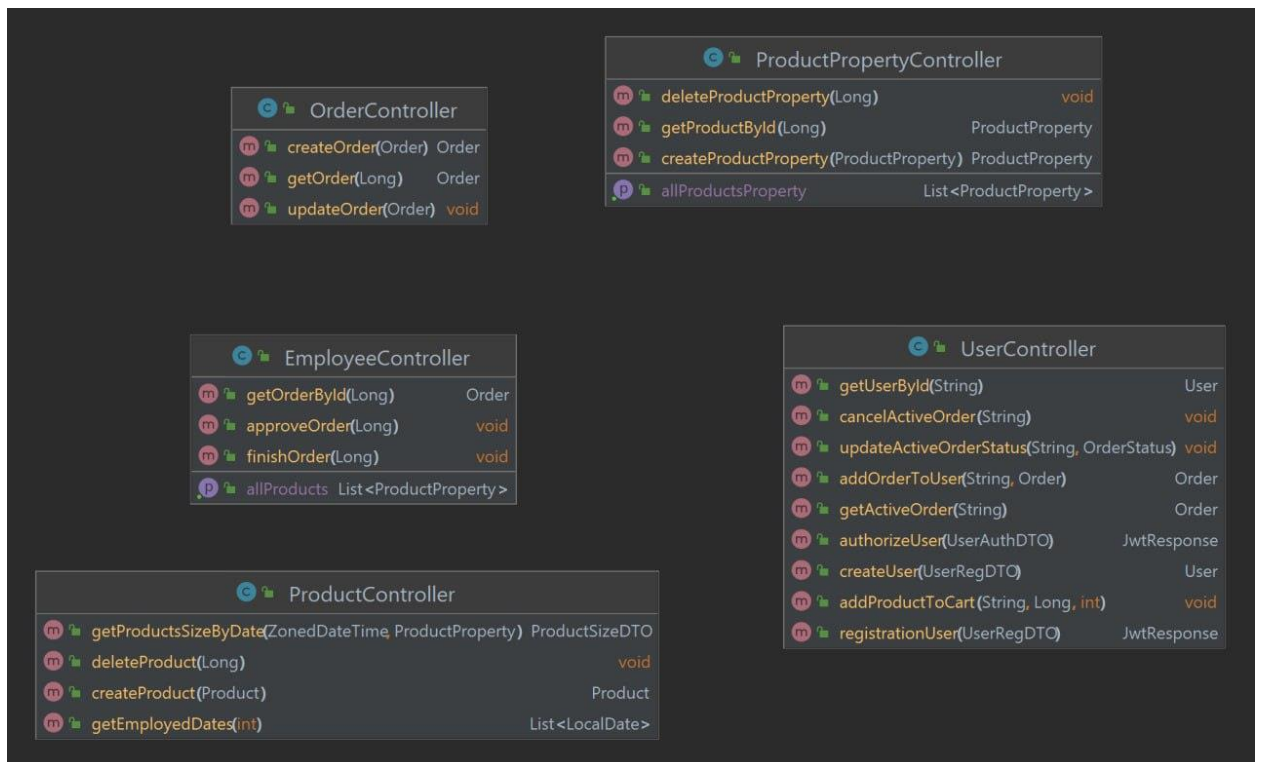


Рисунок 27 - Реализация контроллеров

### 3.9.3 Слой моделей

Слой моделей является ключевой частью трехслойной архитектуры веб-приложения на основе фреймворка Spring Boot и паттерна MVC. В этом слое определены модели данных, которые представляют бизнес-объекты или сущности, с которыми работает приложение. Слой моделей отвечает за работу с данными. На рисунке ниже можно увидеть реализацию моделей.

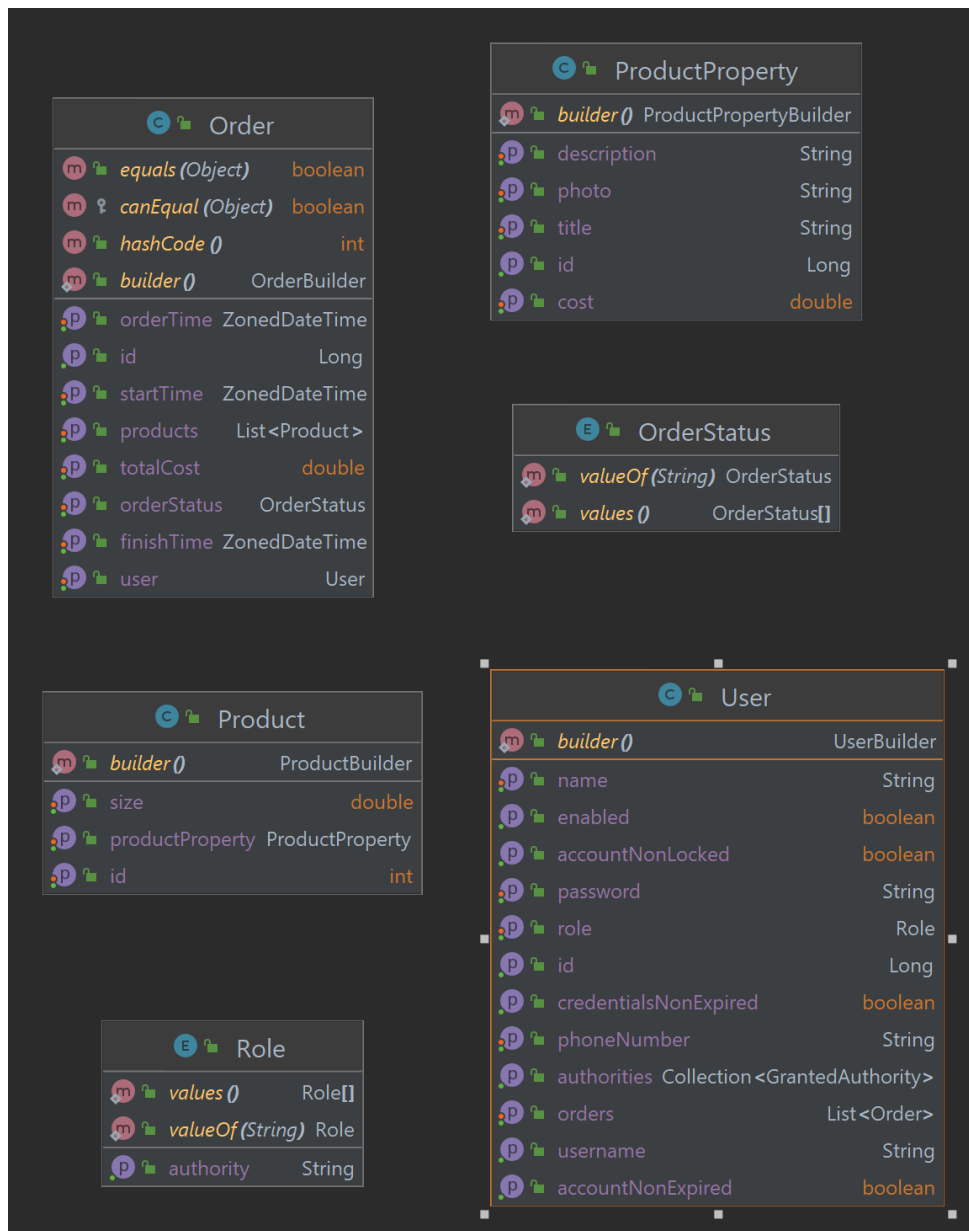


Рисунок 28 - Реализация моделей

### 3.9.4 Слой бизнес-логики

Вся бизнес-логика реализована в service слое. Слой сервисов абстрагирует бизнес-логику от слоя контроллеров и доступа к данным в базе данных, поэтому чаще всего методы просто передают аргумент, полученный из контроллера, вызвавшего его, в соответствующий метод в репозитории, затем возвращают ответ в контроллер, или собирают объект из полученных аргументов и также передают его в репозиторий, а полученный ответ затем в

возвращают контроллеру для дальнейшего использования данных. На рисунке 29 можно увидеть схему сервисов.

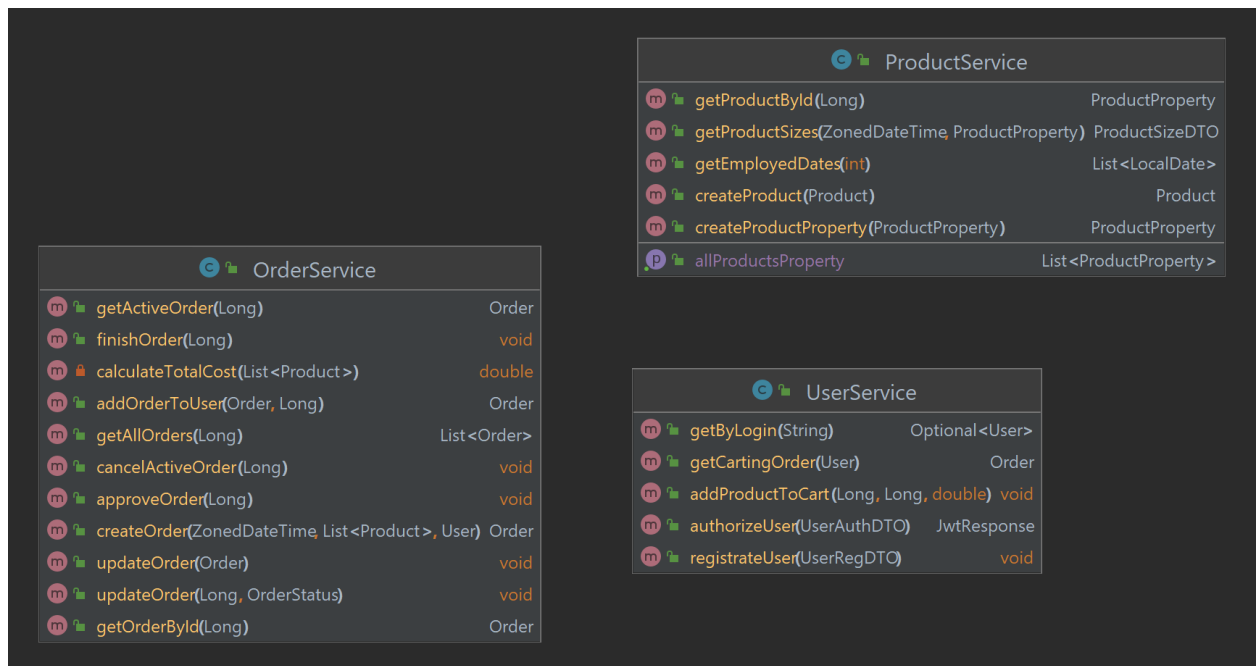


Рисунок 29 - Схема сервисов

### 3.9.5 Механика работы приложения

Для работы нашего приложения необходимы сервисы для большинства из моделей и отвечающие за работу и взаимодействие функций. Это сервисы:

- OrderService;
- UserService;
- ProductService.

OrderService:

Имеет методы для получения заказа по его id, изменения заказа, изменения статуса заказа из активного в один из предложенных по id пользователя, удаления активного заказа по тому же параметру, создания и получения всех заказов пользователя.

UserService:

Имеет методы для получения пользователя по логину (номеру телефона), добавления продуктов в корзину, несколько методов для получения заказа с определенным статусом, авторизации и регистрации пользователя по его телефону, паролю и имени.

ProductService:

Имеет методы для получения продукта по его id, получения всех описаний продуктов, получения всех актуальных размеров товара на определенную дату, как и наоборот, по размеру и id получить список дней, в которые он доступен. Также создания описания продукта и его самого.

### **3.10 Реализация клиентской части приложения**

Для исполнения клиентской части используем Flutter. Приложение построено на MVVM архитектуре. MVVM — это популярный шаблон проектирования, который разделяет приложение на три основных компонента: Модель (Model), Представление (View) и Модель Представления (ModelView). Модель представляет собой данные приложения и логику их обработки. Представление отвечает за отображение данных пользователю. Модель представления — содержит в себе Модель, преобразованную к Представлению, а также команды, которыми может пользоваться Представление, чтобы влиять на Модель.

К плюсам MVVM архитектуры можно отнести:

- Возможность параллельной разработки пользовательских интерфейсов;
- Абстрагирование представления и бизнес-логики;
- Разделяет ответственность различных слоев приложения.

Таким образом, Flutter и MVVM архитектура предоставляют удобный и эффективный способ разработки мобильных приложения, позволяя разделить логику, данные и представление, и обеспечивая удобство использования и поддержки кода.

### 3.10.1 Интерфейс для пользователя

Сразу после запуска приложения пользователю будет показан загрузочный экран.



Рисунок 30 - Загрузочный экран.

При первом заходе необходимо просмотреть инструкцию из трех страниц с короткими справками по использованию приложения.







Рисунок 32 - Экран каталога

Если заказ уже сформирован, то на экране есть qr код заказа и список заказанных товаров с указанием количества и размера.

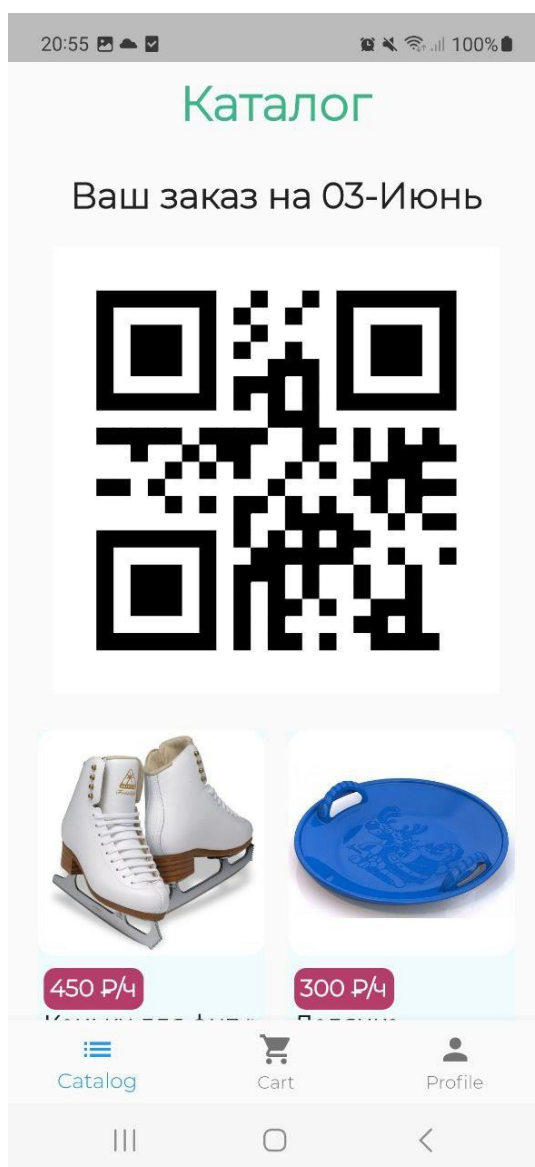


Рисунок 33 - Главный экран приложения при активном заказе

На экране карточки продукта есть изображение товара, его описание, название и цена за час использования.

Есть кнопка “Выберите удобную дату” для выбора даты.

Есть кнопка “Добавить” для добавления товара в корзину. Есть кнопка «Выбранный размер», по нажатию на которую всплывает окошко выбора размера. На этой карточке отображаются все возможные размеры на эту дату.

В правом верхнем углу экрана находится иконка перехода на предыдущий экран.

При попытке неавторизованного пользователя добавить товар в корзину, появляется уведомление о предложении войти или зарегистрироваться.



Рисунок 34 - Экран карточки товара

В центральной части экрана входа в учетную запись находятся текстовые поля для ввода телефона и пароля учетной записи. Есть кнопка “Войти”, при нажатии на которую, при правильно введенных данных, осуществляется переход на страницу пользователя. Есть кнопка “Зарегистрироваться” для незарегистрированных пользователей.

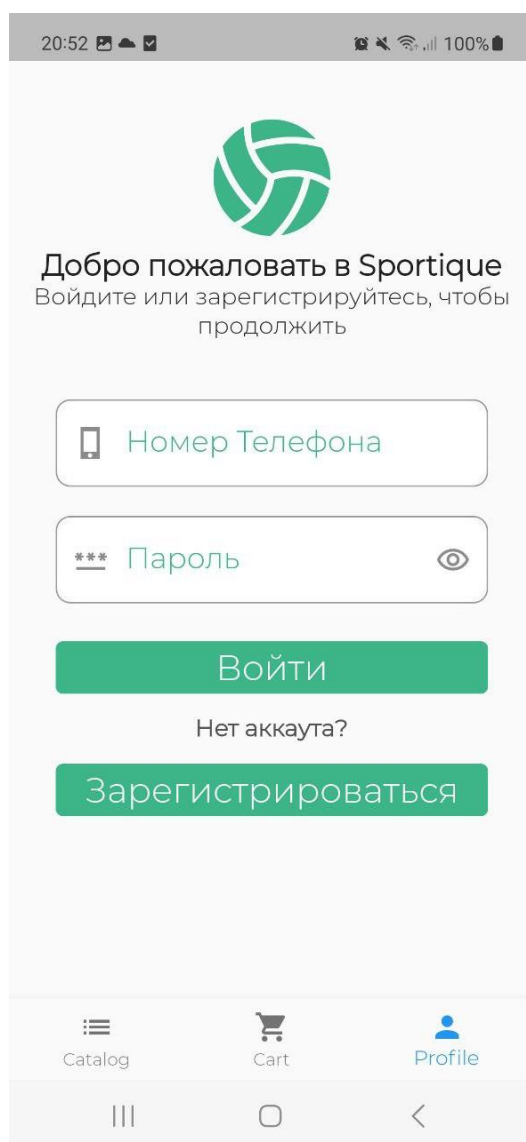


Рисунок 35 - Экран входа в учетную запись

На экране регистрации в центральной части находятся текстовые поля для ввода телефона, имени пользователя учетной записи. Есть кнопка “Зарегистрироваться”.

В центральной части экрана страницы пользователя находятся карточки с историями заказов с информацией о дате заказа, времени и стоимости. Есть кнопка “Отменить” для сформированного заказа, для неактуальных заказов же доступен только просмотр. Вверху есть кнопка для выхода из аккаунта. Также вверху экрана есть информация об имени пользователя и его телефоне.

Вверху экрана корзины находятся выбранные товары (до 4х включительно), у которых можно указать количество и размер. Также

находится информация об общей стоимости заказа. Внизу можно дату, на которую пользователь хочет забронировать. Есть кнопка “Заказать” для формирования заказа.

При сформированном заказе в верхней части экрана находится qr код для получения заказа. Под ним находится список заказанных товаров с указанием количества и размера. В нижней части экрана находится кнопка “Отменить” для отмены заказа.

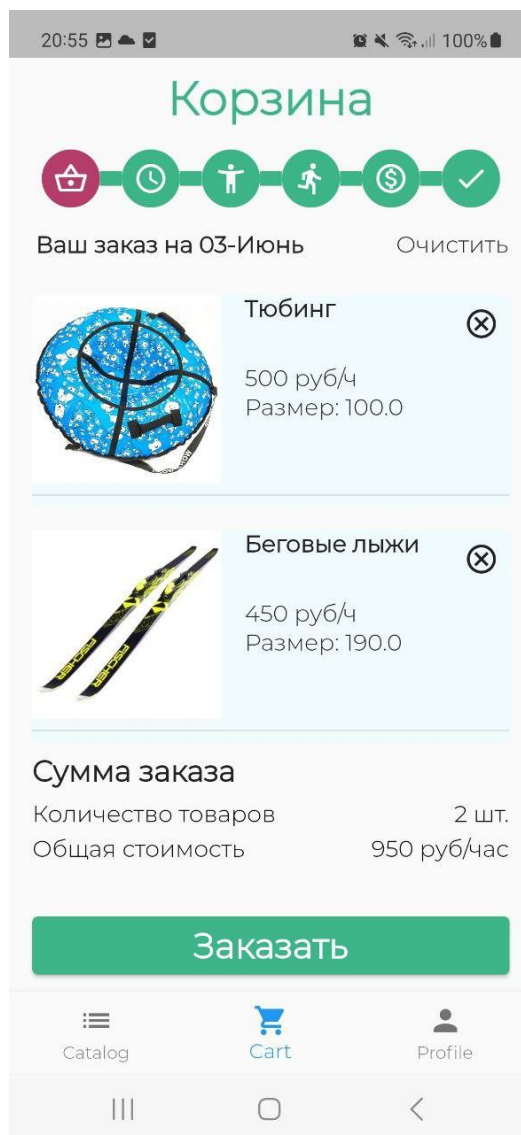


Рисунок 36 - Экран корзины с выбранными товарами

### 3.10.2 Интерфейс для работника

Экран последних заказов состоит из списка заказов, в каждом из которых есть имя пользователя, его номер телефона и изображения товаров. Возможен поиск по номеру телефона или имени.

На домашнем экране в верхней части экрана находится кнопка отсканировать.

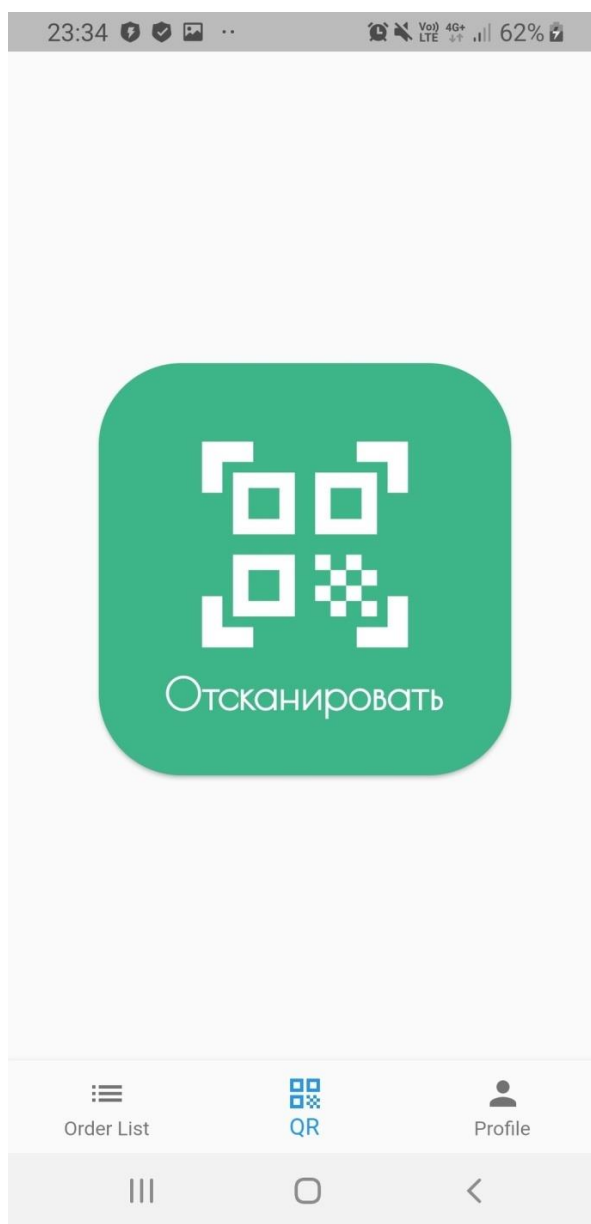


Рисунок 37 - Домашний экран

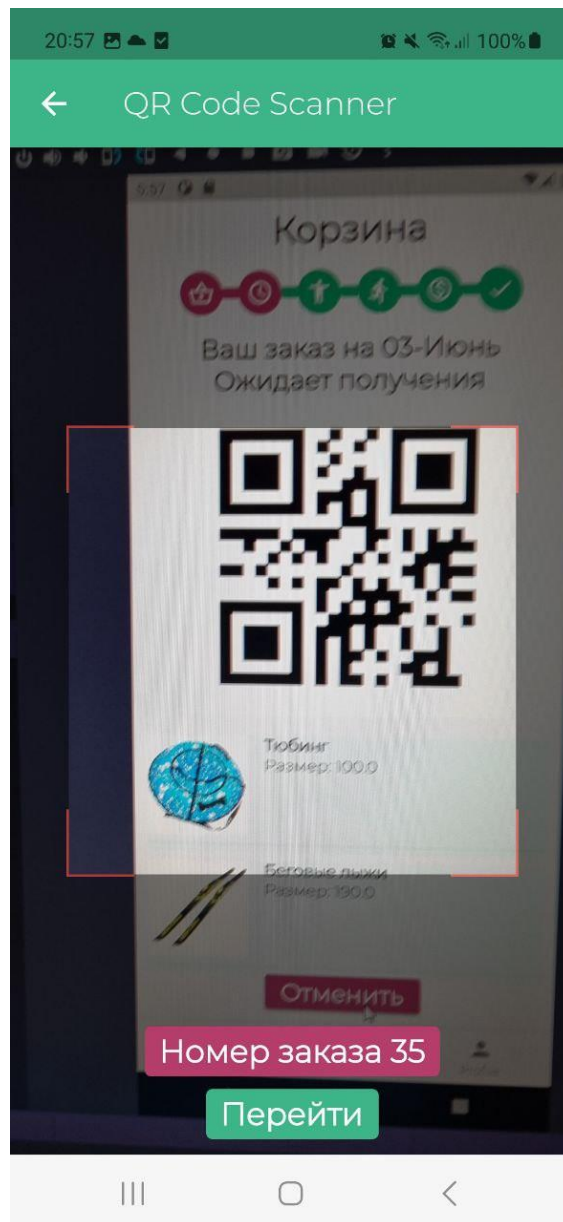


Рисунок 38 - Процесс сканирования.

На экране профиля работника расположена информация о работнике, и кнопка «Выйти из аккаунта».

На экране заказа, ожидающего подтверждения, вверху находится список товаров. Внизу кнопка “Подтвердить”. Переход на данную страницу возможен только через прочтение qr кода.

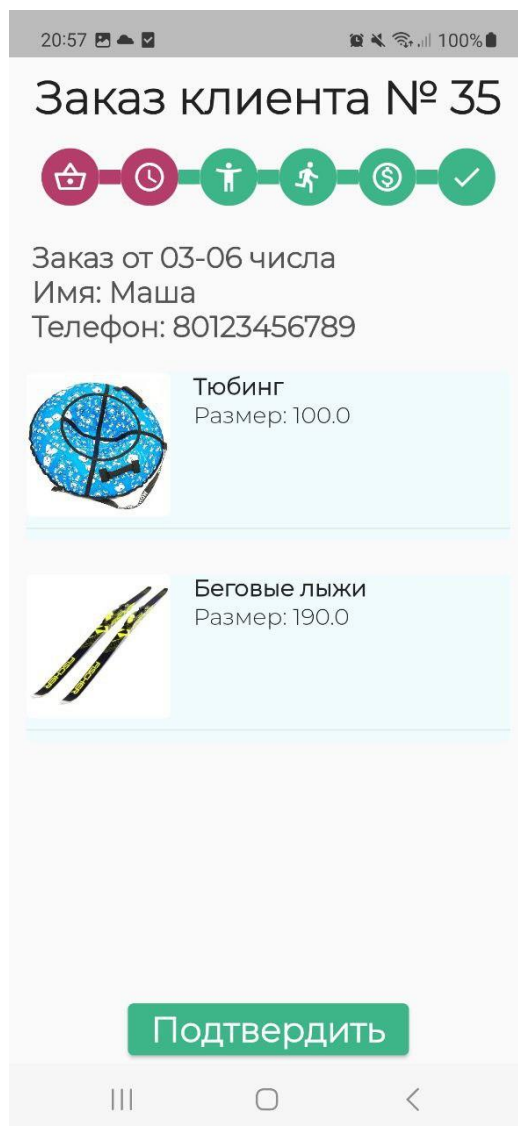


Рисунок 39 - Экран заказа, ожидающего подтверждения

На экране заказа в статусе “примерка” находится список товаров с указанием их размеров. У каждой карточки товара есть кнопка «Удалить» и можно изменить размер. Вверху указывается время, оставшееся для примерки. Переход на этот экран возможен из экрана последних заказов или через QR код клиента.



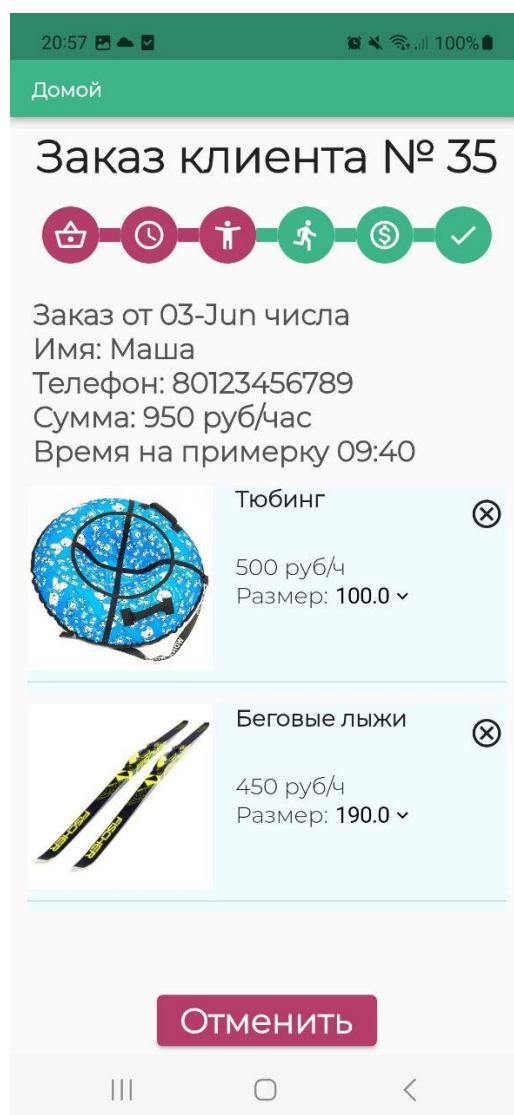


Рисунок 40 - Экран заказа в статусе “примерка”

На экране активного заказа вверху находится список товаров с информацией о цене за час. Ниже находится информация времени заказа и его стоимости. Внизу кнопка “Завершить”. Переход на данный экран возможен при прочтении qr кода или через список последних заказов. Переход на данный экран осуществляется, если время примерки вышло.

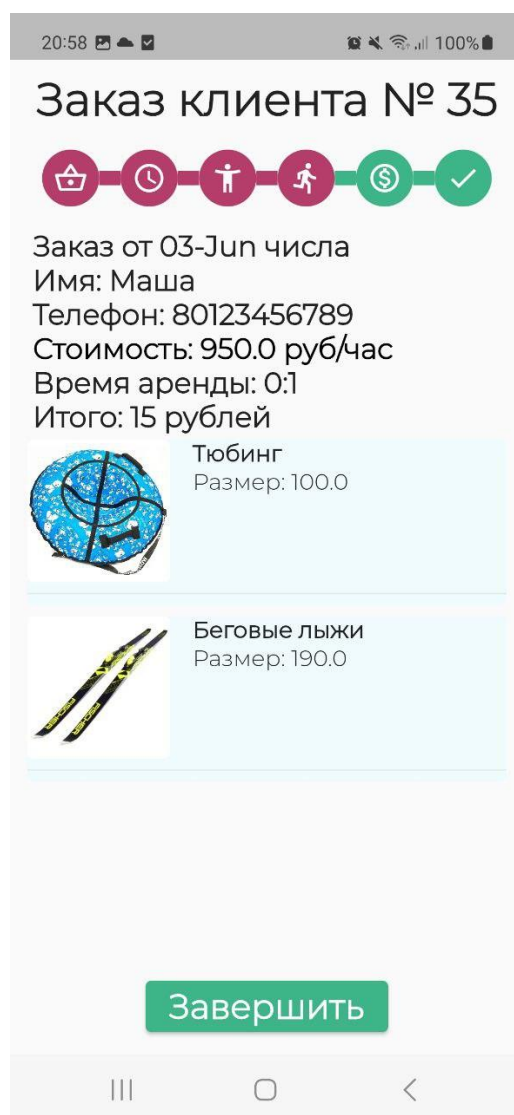


Рисунок 41 - Экран активного заказа

В верхней части экрана оплаты находится qr код для оплаты. Ниже находится информация о заказе. Переход на данный экран осуществляется только после нажатия кнопки “Завершить” на экране активного заказа.

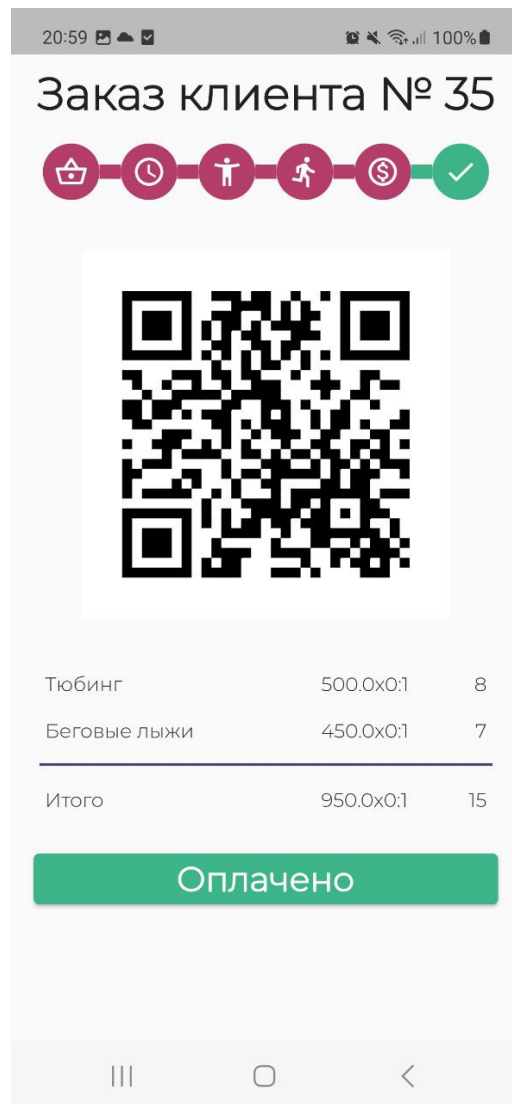


Рисунок 42 - Экран оплаты

### 3.11 Диаграмма классов

Код разделен на 3 части:

- Модель
- Представление
- Модель представления

Рассмотрим классы приложения, исходя из этого разделения.

### 3.11.1 Диаграмма классов модели

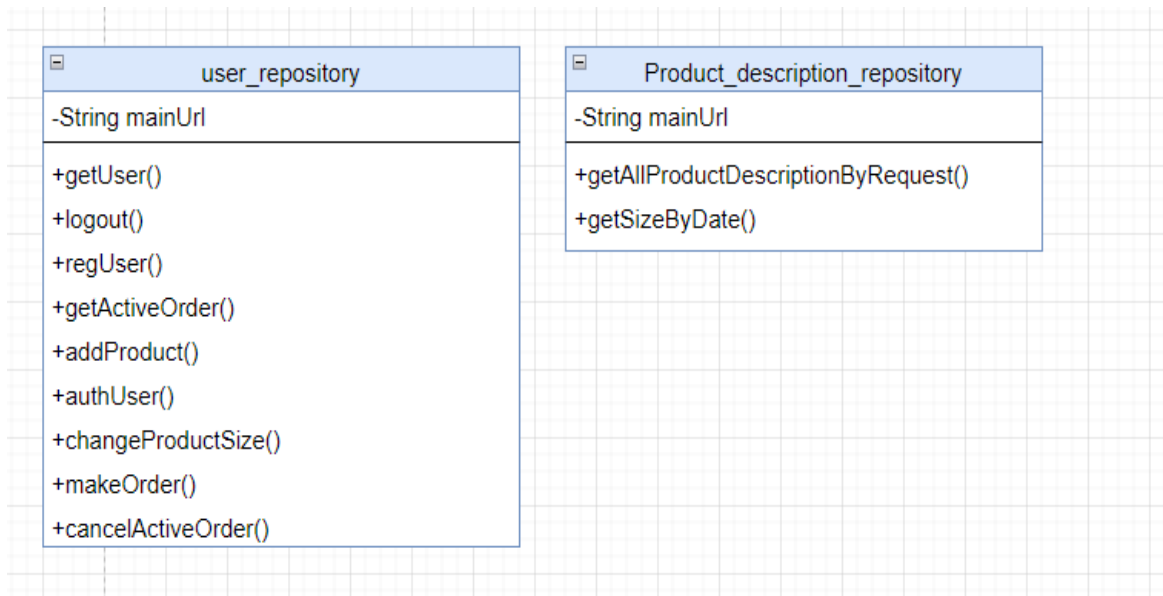


Рисунок 43 - Диаграмма классов репозиториев

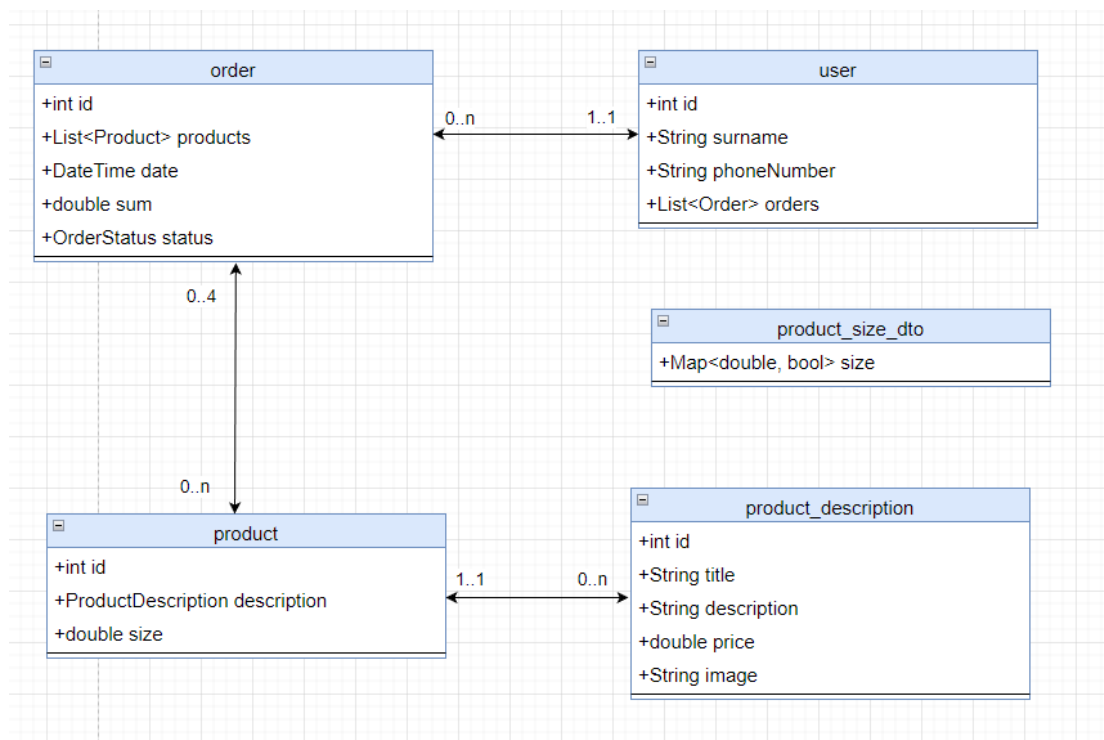


Рисунок 44 - Диаграмма классов слоя данных

### 3.11.2 Диаграмма классов представления

Классы представления в Flutter– это классы виджеты. Они разделены на 2 вида:

- Классы страницы;
- Классы элементы, которые представляют наименьшую единицу представления. Например, это может быть карточка товара или карточка заказа.

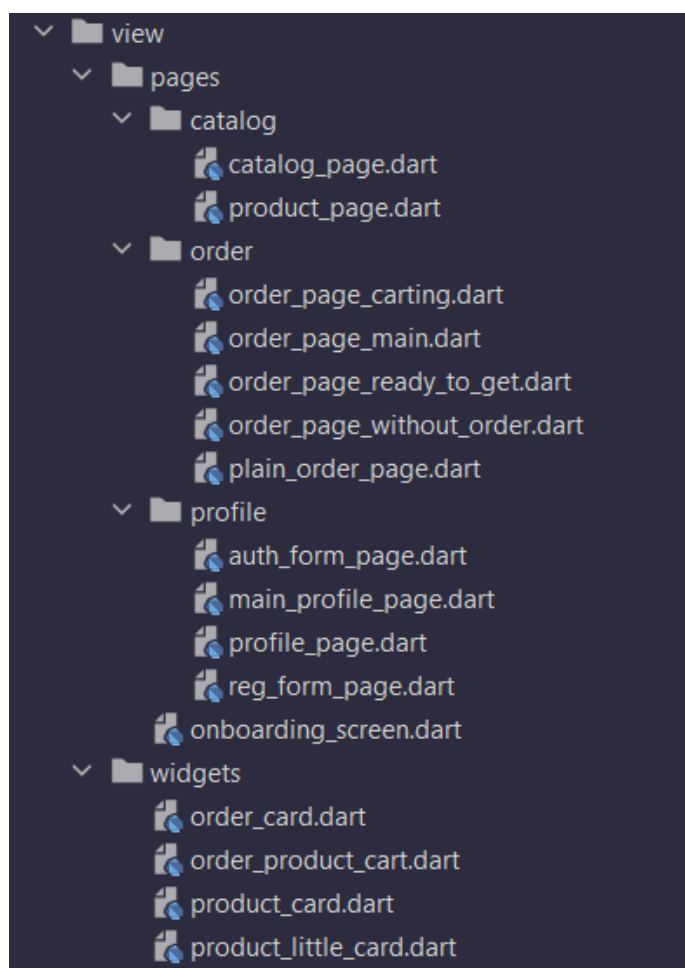


Рисунок 45 - Вид классов представления в файловой структуре

### 3.11.3 Диаграмма классов моделей представления

Модель представления обращается к нашей модели, для получения данных, и уведомляет подписанные классы представления для обновлений данных на них.

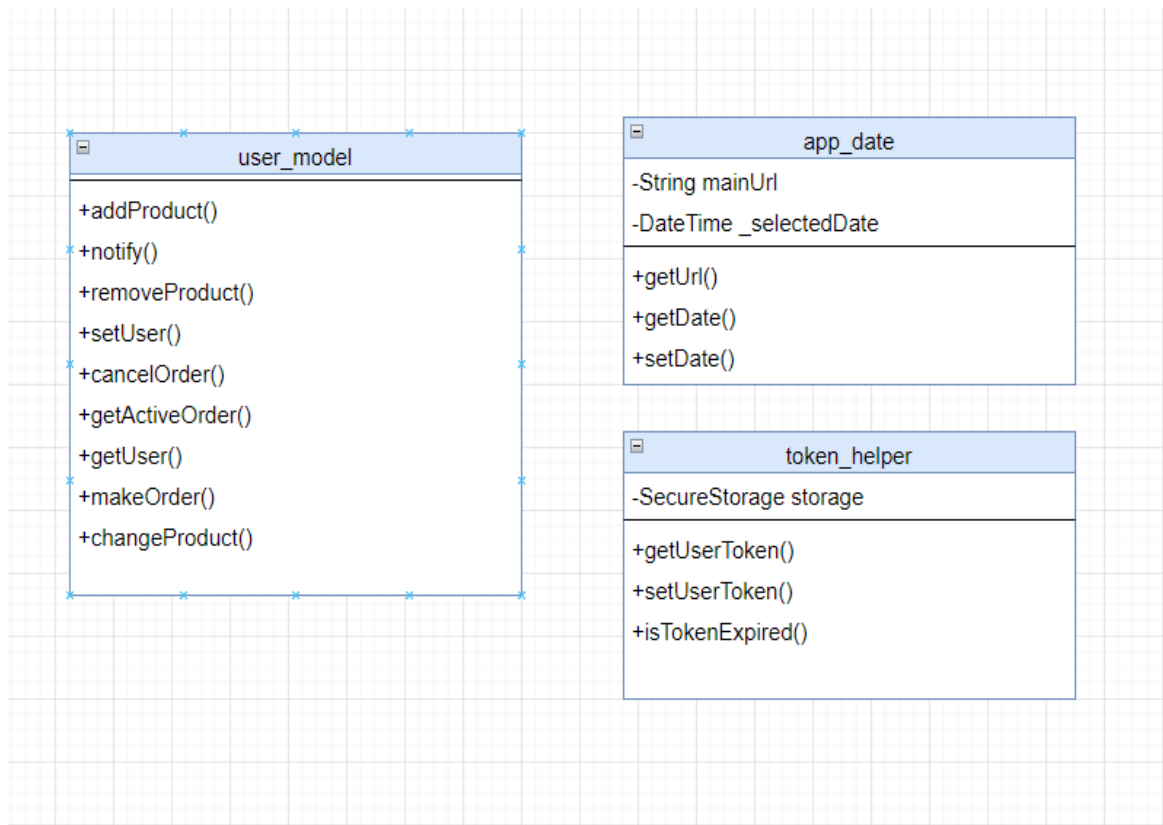


Рисунок 46 - Диаграмма классов моделей представления

### 3.12 Удаленная конфигурация приложения

Для изменения внешнего вида приложения и его поведения без распространения обновления приложения был использован Firebase Remote Config. С помощью этого инструмента мы можем менять цвет заголовка и имеем возможность включать и отключать возможность изменения размера товара в заказе пользователем. Конфигурацию Remote Config вы можете видеть на рисунке 47.

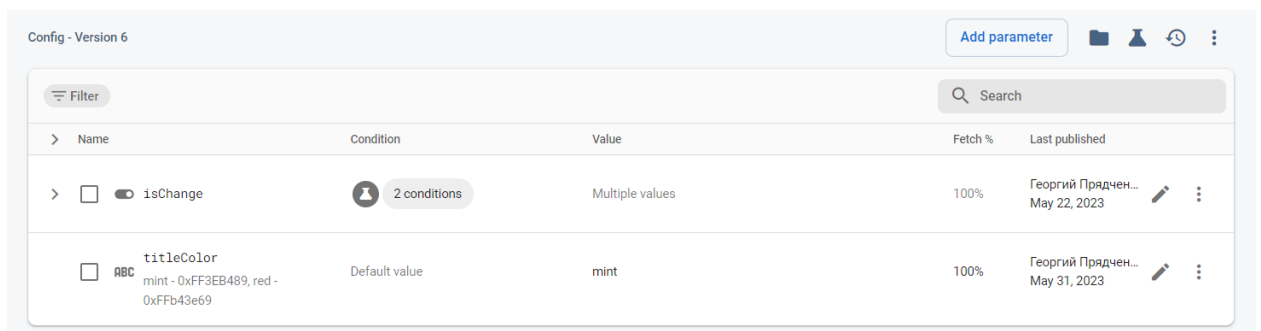


Рисунок 47 - Конфигурация Remote Config

### 3.13 Анализ

В ходе тестирования приложения был проведен ряд аналитических экспериментов.

#### 3.13.1 Анализ удержания пользователей

Используя инструмент Firebase analytics, был проведен анализ количества пользователей, которые открыв приложение, прошли весь пользовательский путь и сделали заказ. График воронки конверсии по количеству уникальных пользователей вы можете видеть на рисунке.

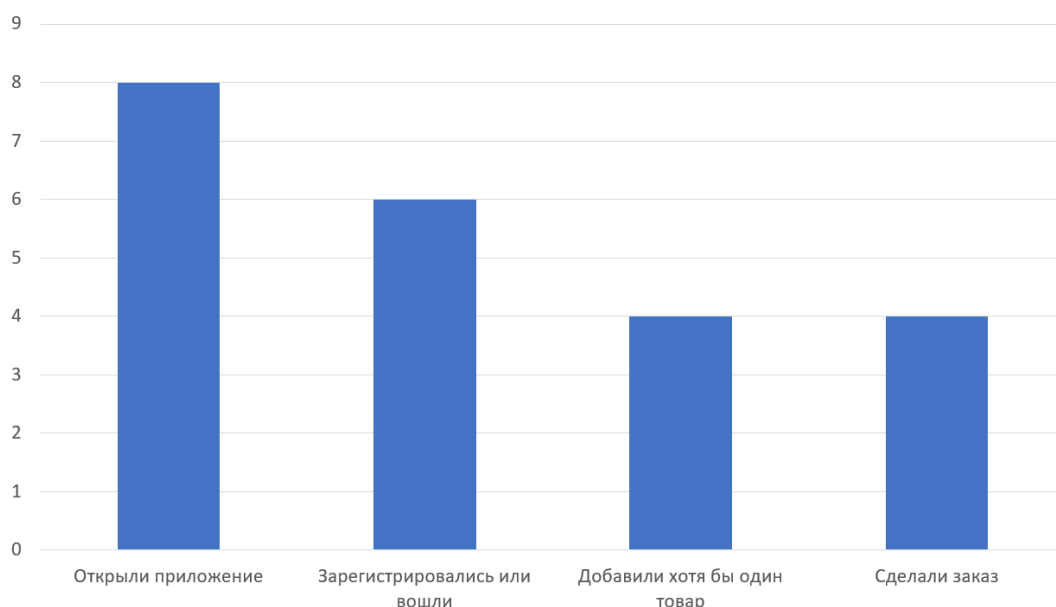


Рисунок 48 - График воронки конверсии по количеству уникальных пользователей

#### 3.13.2 А/В тестирование

Также было проведено А/В тестирование. Объектом нашего тестирования является экран совершения заказа. Его варианты вы можете видеть на рисунке.

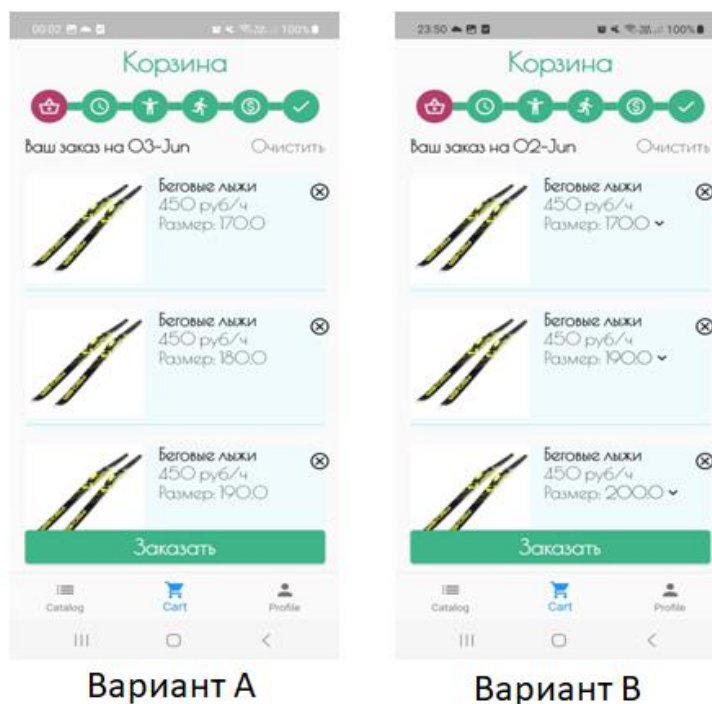


Рисунок 49 - Объект А/В тестирования

Была выдвинута гипотеза, что клиент заранее знает размер товара, который он хочет, и дату заказа. Исходя из этого в варианте А отсутствует функционал изменения размеры и даты. В качестве метрики было выбрано условие совершения заказа. В ходе эксперимента получили следующий график на рисунке 50.

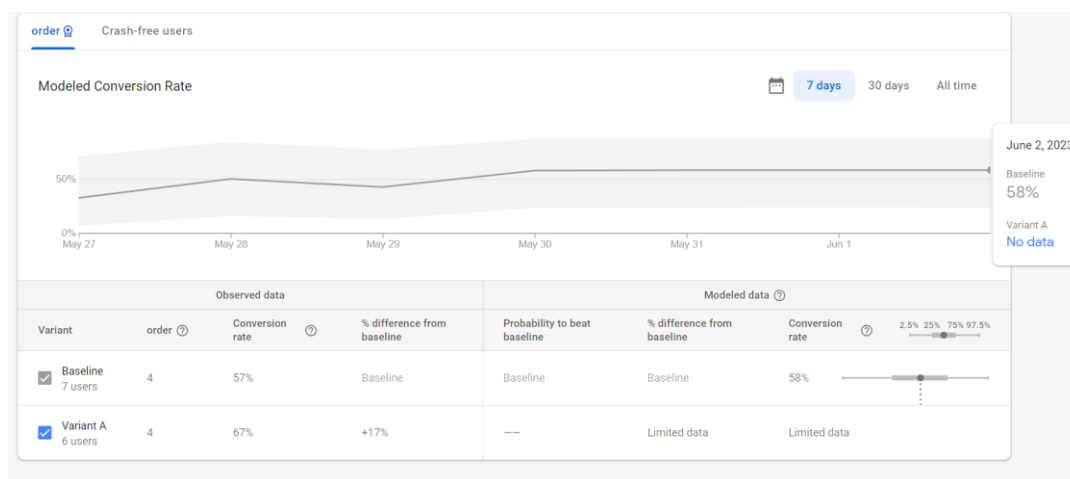


Рисунок 50 - Результаты А/В тестирования

В ходе эксперимента было выяснено, что клиенты часто совершают заказ при варианте А.



## **Заключение**

В ходе выполнения данного курсового проекта были выполнены все поставленные задачи.

Все поставленные цели были выполнены:

- Введена в систему проката возможность безналичной оплаты, в частности, таких возможностей, как оплата по QR-коду и формирование электронных чеков;
- Реализованы возможности для клиента выбора инвентаря и бронирования его через мобильное приложение, а также получение инвентаря по QR-коду;
- Автоматический подсчет времени и стоимости сессии;
- Реализация механизма примерки.

## Список используемых источников

1. Документация Flutter [Электронный ресурс]. – Режим доступа: URL: <https://docs.flutter.dev/> – Заглавие с экрана. – (Дата обращения 01.05.2023).
2. Документация SpringBoot [Электронный ресурс]. – Режим доступа: URL: <https://docs.spring.io/spring-boot/docs/current/reference/html/> – Заглавие с экрана. – (Дата обращения 22.04.2023).
3. Документация Swager [Электронный ресурс]. – Режим доступа: URL: <https://swagger.io/docs/> - (Дата обращения 10.05.2023).
4. Вигерс К. Разработка требований к программному обеспечению/ Карл Вигерс, Джой Бити. — М.: Изд-во Русская редакция, 2014. — 736 с.
5. The Prosand Cons of Online Booking Systems [Электронный ресурс]. – Режим доступа: URL: <https://miyn.app/the-pros-and-cons-of-online-booking-system/> – Заглавие с экрана. – (Дата обращения 21.05.2023).
6. Платное обслуживание населения России [Электронный ресурс]. – Режим доступа: URL: <https://rosstat.gov.ru/folder/210/document/13235> – Заглавие с экрана. – (Дата обращения 21.05.2023).
7. Особенности специфики формирования доступности услуг массового спорта [Электронный ресурс]. – Режим доступа: URL: <https://apni.ru/article/1095-osobennosti-spetsifiki-formirovaniya-dostupn> – Заглавие с экрана. – (Дата обращения 21.05.2023).