МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет *компьютерных наук*

*Кафедра технологий обработки и защиты информации*

*Приложение для аренды спортивного снаряжения «ProCat»*

*Курсовой проект*

*09.03.02 Информационные системы и технологии*

*Обработка информации и машинное обучение*

Допущен к защите

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_П.Л. Некрасов*, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Е. Яньшина*, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Ю. Сахаров*, 3 курс, д/о*

Воронеж 2021

Оглавление

[Оглавление 2](#_Toc74950340)

[Введение 3](#_Toc74950341)

[1. Постановка задачи 4](#_Toc74950342)

[2. Анализ предметной области 5](#_Toc74950343)

[2.1 Глоссарий 5](#_Toc74950344)

[2.2 Анализ существующих решений 5](#_Toc74950345)

[2.3 Анализ задачи 7](#_Toc74950346)

[2.3.1 Варианты использования приложения 7](#_Toc74950347)

[2.3.2 Взаимодействие компонентов системы 9](#_Toc74950348)

[2.3.3 Варианты состояния системы 10](#_Toc74950349)

[2.3.4 Варианты действия в системе 12](#_Toc74950350)

[2.3.5 Развертывание приложения 14](#_Toc74950351)

[2.3.6 Контекстная диаграмма 15](#_Toc74950352)

[3. Анализ средств реализации 16](#_Toc74950353)

[4. Реализация 17](#_Toc74950354)

[4.1 Сущности 17](#_Toc74950355)

[4.2 Сценарии воронок конверсии 20](#_Toc74950356)

[4.3 Графический интерфейс 22](#_Toc74950357)

[4.3.1 Начальный экран с выбором пункта проката 22](#_Toc74950358)

[4.3.2 Экран с выбором снаряжения 23](#_Toc74950359)

[4.3.3 Экран проверки номера телефона 24](#_Toc74950360)

[4.3.4 Экран регистрации 25](#_Toc74950361)

[4.3.5 Экран авторизации 26](#_Toc74950362)

[4.3.6 Экран профиля 27](#_Toc74950363)

[4.3.7 Экран сканирования QR-кода 29](#_Toc74950364)

[4.3.8 Экран с информацией о сканируемом предмете 30](#_Toc74950365)

[4.3.9 Экран оплаты 33](#_Toc74950366)

[4.3.10 Экран арендованных предметов пользователем 35](#_Toc74950367)

[5. Тестирование 37](#_Toc74950368)

[5.1 Дымовое тестирование 37](#_Toc74950369)

[5.2 UI тесты 38](#_Toc74950370)

[5.3 Юзабилити тестирование 46](#_Toc74950371)

[5.4 UNIT тесты 47](#_Toc74950372)

[Заключение 49](#_Toc74950373)

# Введение

В современном интернете есть различные сервисы, которые позволяют покупать или арендовать различные вещи. Но любому человеку нужен отдых, и самое очевидное и первостепенное, о чем думает каждый — это хобби. По статистике 42% людей всего мира занимаются спортом. В настоящее время существуют приложения и сайты, посвященные поиску и аренде спортивного снаряжения в каком-либо конкретном городе, но у каждого из них различные функциональные возможности, разный интерфейс. Сейчас набирают обороты именно мобильные приложения из-за их удобства и внешнего вида, именно поэтому было решено заняться именно разработкой мобильного приложения для Android.

Желаемое приложение должен облегчить пользователям поиск и аренду спортивного снаряжения. Тем самым необходимости в просмотре большого количества сайтов и установки приложений с арендой снаряжения не будет. Также приложение будет ориентироваться владельцев проката тоже, чтобы упростить приемку и выдачу спортивного товара.

Желаемое приложение должно предоставлять основную необходимую функциональность:

* Регистрацию, авторизацию, просмотр личного кабинета и выход из него
* Поиск необходимого спортивного снаряжения пользователем
* Оформление заявки, аренда и оплата

Данный курсовой проект посвящен разработке именно такого, простого в освоении, но в то же время выполняющего самые необходимые функции приложения, способного уменьшить временные затраты каждого человека на поиск и аренду необходимого снаряжения.

# 1. Постановка задачи

Цель курсовой работы: реализовать приложение, которое отвечает следующим требованиям:

- регистрация Гостей и авторизация неавторизированных Пользователей

- просмотр списка доступного снаряжения для аренды Гостем

- формирование заявки на аренду снаряжения Пользователем

- просмотр списка заявок Пользователем

- возможность формирования двух и более заявок на аренду одним

Пользователем

- считывание информации о товаре с помощью QR-кода Пользователем

- оплата аренды спортивного снаряжения Пользователем

- начало и завершение аренды Пользователем

Для достижения данной цели были выделены следующие задачи:

- Разработка Android части приложения, находящиеся на телефоне пользователя;

- Разработка Back-end части приложения, развернутый на удаленном сервере;

- Создание связи между Android и Back-end частями приложения;

- Разработка базы данных, расположенной на удаленном сервере.

# 2. Анализ предметной области

## 2.1 Глоссарий

***Заявка*** — заполненная пользователем информация, необходимая и достаточная для аренды спортивного снаряжения

## 2.2 Анализ существующих решений

1. <https://rentstation.ru/>

Достоинства:

* Аренда нескольких видов снаряжения
* Оплата онлайн

Недостатки:

* Несовременный дизайн
* Нет сканирования QR-кода для аренды
* Необходимость залога

1. https://rentmania.ru

Достоинства:

* Аренда нескольких видов снаряжения

Недостатки:

* Несовременный дизайн
* Нет сканирования QR-кода для аренды
* Необходимость залога
* Оплата онлайн отсутствует

1. <https://whoosh.bike>

Достоинства:

* Современный дизайн
* Оплата онлайн
* Сканирование QR-кода для аренды

Недостатки:

* Нет возможности аренды нескольких видов снаряжения
* Необходимость залога

1. https://velobike.ru

Достоинства:

* Оплата онлайн
* Сканирование QR-кода для аренды
* Нет необходимости залога

Недостатки:

* Нет возможности аренды нескольких видов снаряжения
* Несовременный дизайн

## 2.3 Анализ задачи

### 2.3.1 Варианты использования приложения

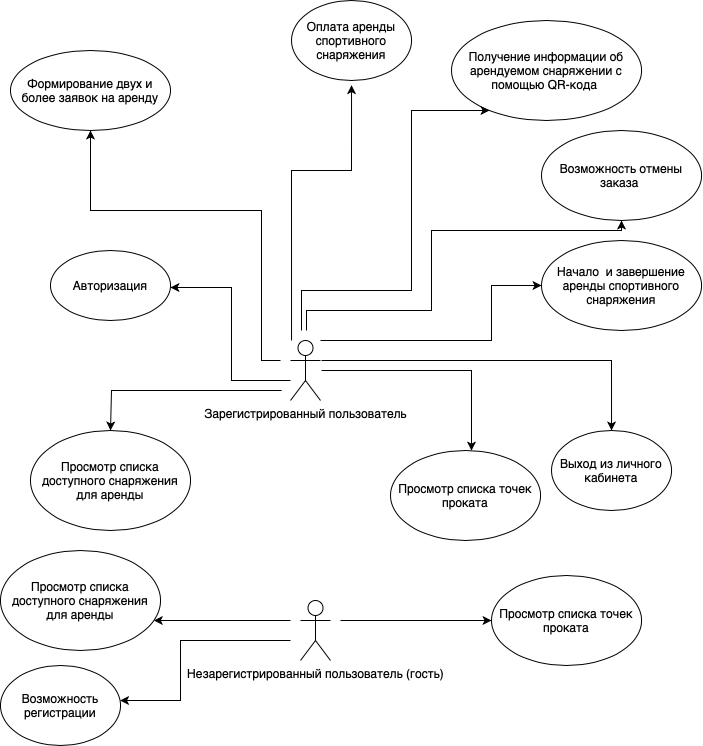


Рисунок 1. Диаграмма прецедентов.

При взаимодействии с сайтом у пользователя есть определенный

список возможностей, который более наглядно изображен на рисунке 1:

* Функционал гостя
* Просмотр главного экрана с выбором точек проката.
* Просмотр доступного для аренды снаряжения.
* Возможность регистрации
* Возможность авторизации
* Функционал авторизированного пользователя
* Просмотр главного экрана с выбором точек проката.
* Просмотр доступного для аренды снаряжения.
* Возможность авторизации
* Получение информации об арендуемом снаряжении при помощи QR-кода
* Формирование двух и более заявок на аренду
* Просмотр списка оформленных заявок
* Начало и завершение аренды спортивного снаряжения
* Оплата аренды спортивного снаряжения и получение чека
* Выход из личного кабинета

### 2.3.2 Взаимодействие компонентов системы

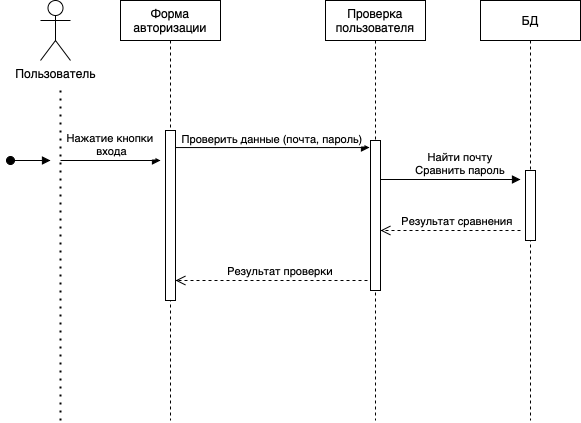


Рисунок 2. Диаграмма последовательностей.

На рисунке 2 показана диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов при авторизации пользователя.  
 Для авторизации пользователь обращается к форме авторизации, которая передаёт введённые данные на проверку в модуль проверки пользователя. Тот в свою очередь проверяет существование данного пользователя в базе данных и совпадение введённого пароля с паролем, хранящимся в базе данных. Модуль проверки пользователя посылает статус проверки на форму авторизации, которая выводит пользователю результат авторизации.

### 2.3.3 Варианты состояния системы

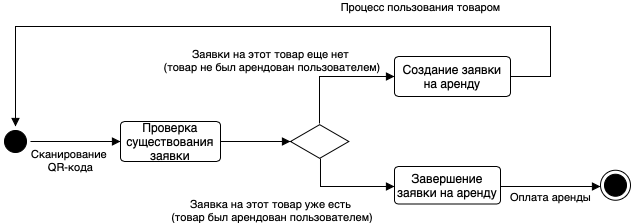


Рисунок 3. Диаграмма состояний.

Диаграмма состояний, изображенная на Рисунке 3, отражает возможные состояния заявки. Посредством сканирования QR-кода производится проверка существования заявки. Если заявки на данный товар еще не было, то создается новая заявка, и далее идет процесс пользования товаром, после окончания которого будет производиться повторная проверка существования заявки. Если заявка на данный товар уже была заведена, то происходит завершение заявки на аренду и оплата аренды.

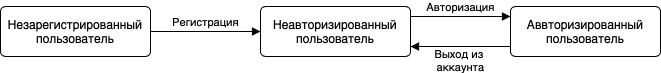


Рисунок 4. Диаграмма состояний.

Диаграмма состояний, изображенная на Рисунке 4, отражает возможные состояния пользователя. Изначально любой пользователь, не прошедший регистрацию, находится в состоянии незарегистрированного пользователя. После прохождения регистрации пользователь переходит в состояние неавторизованного пользователя. После авторизации пользователь переходит в состояние авторизованного пользователя. Если пользователь выходит из аккаунта он возвращается к состоянию неавторизованного пользователя и может авторизоваться снова. Если пользователь захочет пройти процесс регистрации, заполнив необходимые поля, система проверит входящие данные на корректность и в случае ошибки выдаст предупреждение. Если данные корректны, пользователь перейдет в личный кабинет.  
 При желании пользователя авторизоваться, заполнив необходимые поля, система так же проверит корректность введенных данных, в случае, если пользователя с такими данными нет, система предложит зарегистрироваться.

### 2.3.4 Варианты действия в системе

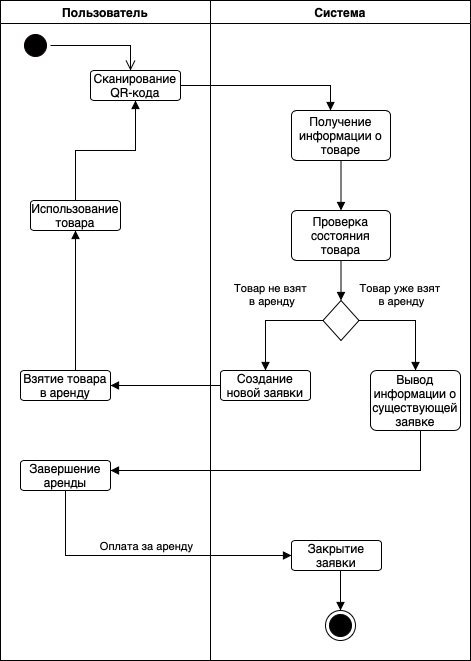


Рисунок 5. Диаграмма активности.

Диаграмма активности, изображенная на Рисунке 5, отражает действия авторизованного пользователя при создании заявки. После того, как авторизованный пользователь отсканирует QR-код, указанный на товаре, система получит данные о нем. Когда система проверит состояние товара, возможно два варианта развития событий:

1. В случае, если товар не был еще взят в аренду, то система создаст новую заявку на аренду снаряжения, после чего пользователь сможет взять в аренду товар. Попользовавшись товаром определенное время, пользователь должен отсканировать QR-код заново, чтобы впоследствии завершить аренду.
2. В случае, если товар уже был взят в аренду, система выведет информацию о существующей заявке, после чего пользователь сможет завершить аренду.

Далее пользователь оплачивает арендную плату, и заявка закрывается системой.

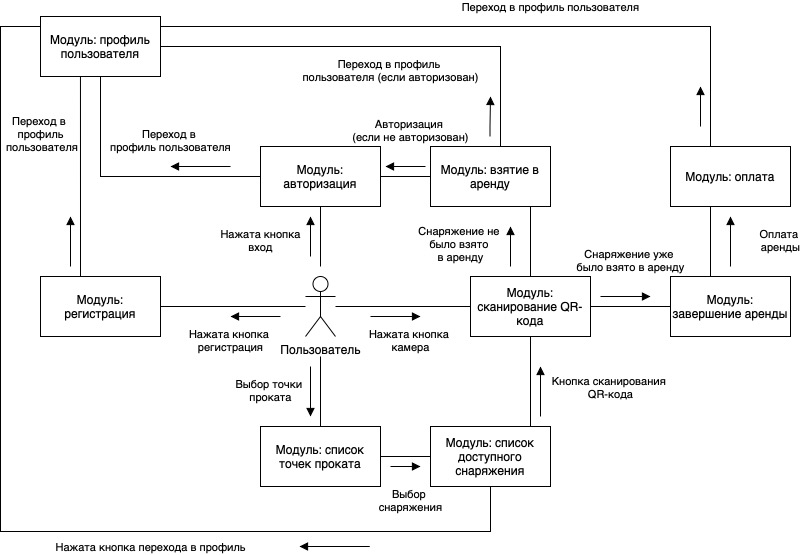


Рисунок 6. Диаграмма взаимодействий.

На Рисунке 6 представлена диаграмма взаимодействий. Она отражает возможные действия пользователя и системы.

Если пользователь захочет взять в аренду спортивное снаряжение, система проверит статус авторизации клиента, если он не авторизирован, то ему будет предоставлена возможность авторизоваться. Если пользователь авторизован, то он может арендовать товар. Информация о списке текущих аренд появляется в профиле пользователя.

Если пользователь захочет пройти процесс регистрации, заполнив необходимые поля, система проверит входящие данные на корректность и в случае ошибки выдаст предупреждение. Если данные корректны, пользователь перейдет в личный кабинет.

При желании пользователя авторизоваться, заполнив поля, система так же проверит корректность введенных данных.

### 2.3.5 Развертывание приложения

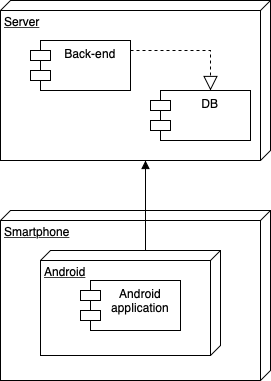


Рисунок 7. Диаграмма развертывания.

На Рисунке 7 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты работают на каждом узле и как различные части этого комплекса соединяются друг с другом. телефон и сервер, а в качестве узла среды выполнения выступает приложение. Развернуто Andriod-приложение, а на серверной части back-end и база данных.

### 2.3.6 Контекстная диаграмма

Рассмотрим основной бизнес – процесс на примере контекстной диаграммы, представленной на Рисунке 8. Данная диаграмма представляет собой общее видение процесса работы Системы.

Работу сервиса регулируют Законы РФ. Обеспечивает работу Системы наличие Спортивного снаряжения в точках проката. На вход в Систему поступают Пользователь, желающий взять в прокат спортивное снаряжение, и Плата за услуги. На выходе Система выдает Оказание услуги по аренде спортивного снаряжения и Прибыль.

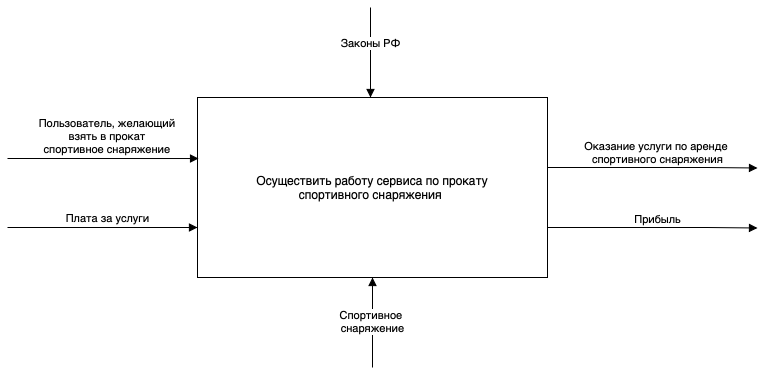


Рисунок 8. Контекстная диаграмма.

# 3. Анализ средств реализации

В качестве средств реализации Android приложения были выбраны следующие технологии:

* Android SDK v30 – набор инструментов для разработки мобильных приложений;
* Kotlin - современный язык программирования, обеспечивающий лаконичный, читаемый, хорошо поддерживаемый код;
* RxJava 3 - инструмент для асинхронной работы;
* Dagger - удобный инструмент для инъекции зависимостей;
* Retrofit - безопасный HTTP-клиент для Android и Java.

В качестве СУБД была выбрана PostgreSQL. Она является хорошо масштабируемой, в равной степени легко может быть использована для работы, как с малыми, так и с большими объемами данных. А за счет упрощения некоторых используемых в ней стандартов система имеет высокую производительность.

В качестве языка разработки в back-end использовался язык Java 8, а также:

* Maven – инструмент для сборки и управления проектом
* Spring Boot – фреймворк, позволяющий упростить приложение на основе Spring
* Spring Data JPA - дополнительный удобный механизм для взаимодействия с сущностями базы данных, организации их в репозитории, извлечение данных

# 4. Реализация

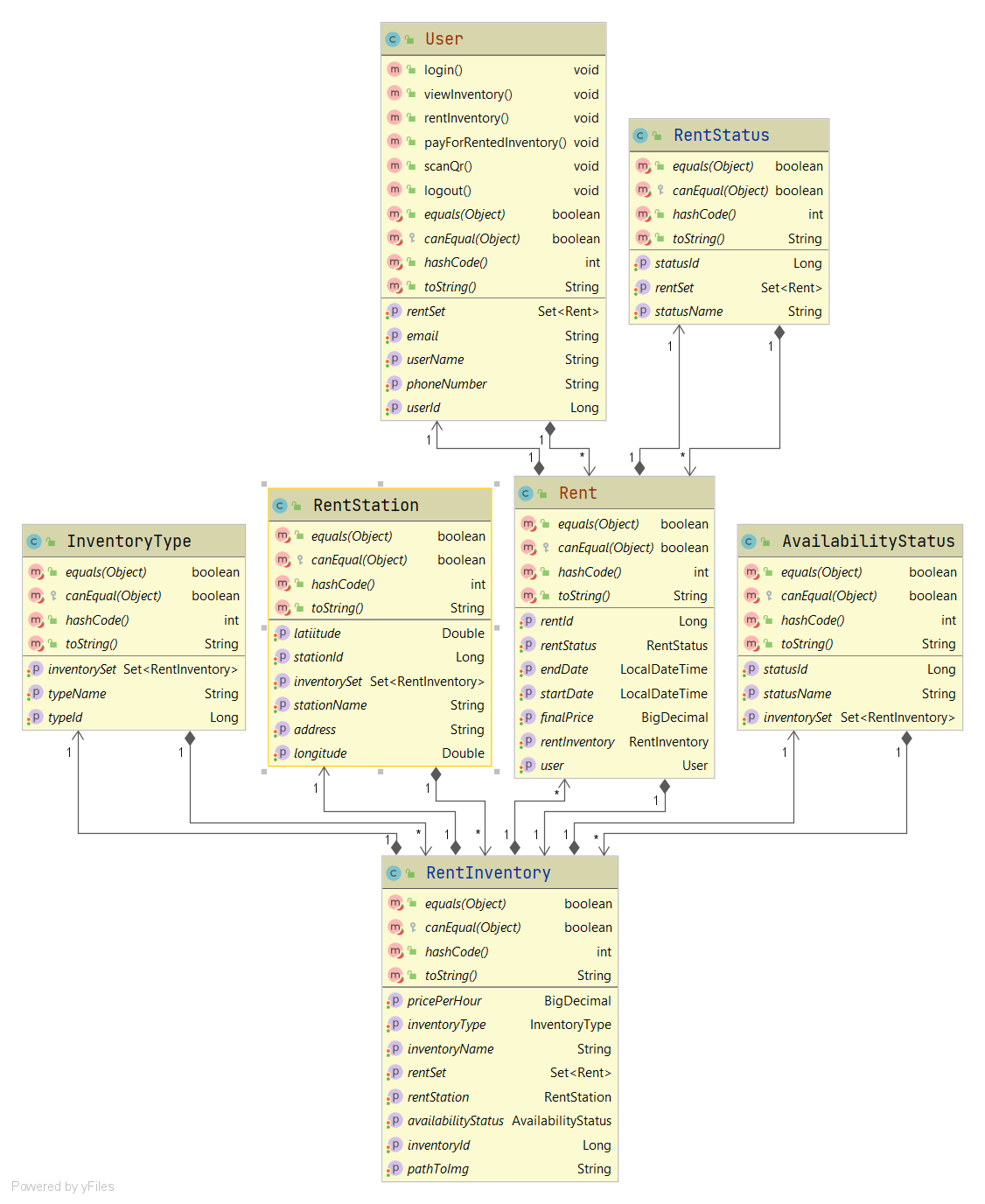
4.1 Сущности

Рисунок 9. Диаграмма классов.

На Рисунке 9 изображена диаграмма классов, отражающая их отношения.

1. Класс «User» – представляет собой отражение сущности пользователь. Класс имеет следующие свойства:

* «userId» – уникальный идентификатор,
* «userName» – имя пользователя,
* «phoneNumber» – номер телефона,
* «email» – электронная почта.

1. Класс «RentStation» – представляет собой отражение сущности точка проката. Класс имеет следующие свойства:

* «stationId» – уникальный идентификатор,
* «stationName» – название точки проката,
* «address» – адрес,
* «latitude» – широта,
* «longitude» – долгота.

1. Класс «Rent» – представляет собой отражение сущности аренда. Класс имеет следующие свойства:

* «rentId» – уникальный идентификатор,
* «user» – пользователь,
* «inventory» – инвентарь,
* «startDate» – время начала аренды,
* «endDate» – время окончания аренды,
* «finalPrice» – цена за аренду,
* «rentStatus» – статус аренды.

1. Класс «RentInventory» – представляет собой отражение сущности аренда инвентаря. Класс имеет следующие свойства:

* «inventoryId» – уникальный идентификатор,
* «inventoryType» – тип инвентаря,
* «inventoryName» – наименование инвентаря,
* «pathToImg» – путь до картинки инвентаря,
* «pricePerHour» – цена за час аренды,
* «rentStation» –точка проката,
* «availabilityStatus» – статус доступности инвентаря.

1. Класс «RentStatus» – представляет собой отражение сущности статус аренды. Класс имеет следующие свойства:

* «statusId» – уникальный идентификатор,
* «statusName» – имя статуса аренды инвентаря.

1. Класс «InventoryType» – представляет собой отражение сущности тип инвентаря. Класс имеет следующие свойства:

* «typeId» – уникальный идентификатор,
* «typeName» – имя типа инвентаря.

1. Класс «AvailabilityStatus» – представляет собой отражение сущности доступность снаряжения. Класс имеет следующие свойства:

* «statusId» – уникальный идентификатор,
* «statusName» – наименование доступности снаряжения.

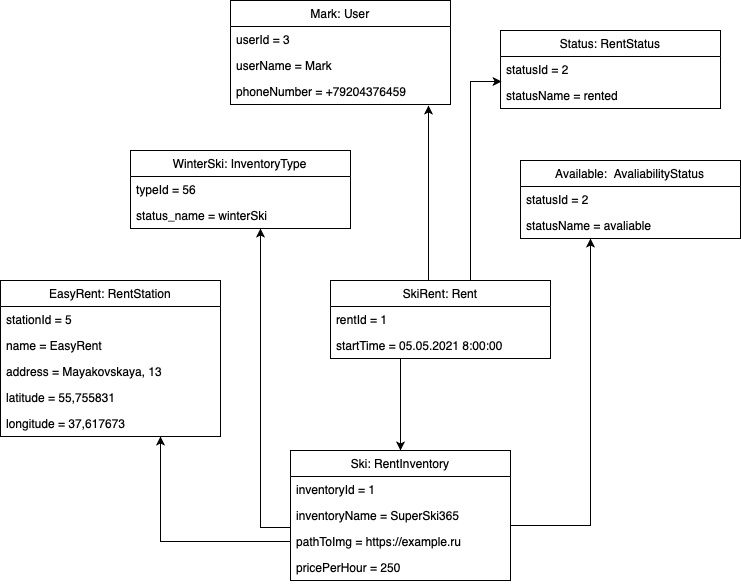


Рисунок 10. Диаграмма объектов.

На Рисунке 10 изображена диаграмма объектов, которая отражает множество экземпляров классов и отношений между ними в некоторый момент времени. На ней изображён экземпляр класса “Пользователь”, экземпляр класса “Аренда”, экземпляр класса “Статус аренды”, экземпляр класса “Точка проката”, экземпляр класса “Инвентарь”, экземпляр класса “Тип инвентаря” и экземпляр класса “Доступность снаряжения”.

## 

## 4.2 Сценарии воронок конверсии

1. Посетил начальную страницу – Перешел на страницу просмотра списка доступного оборудования - Отсканировал QR-код – Оформил заявку

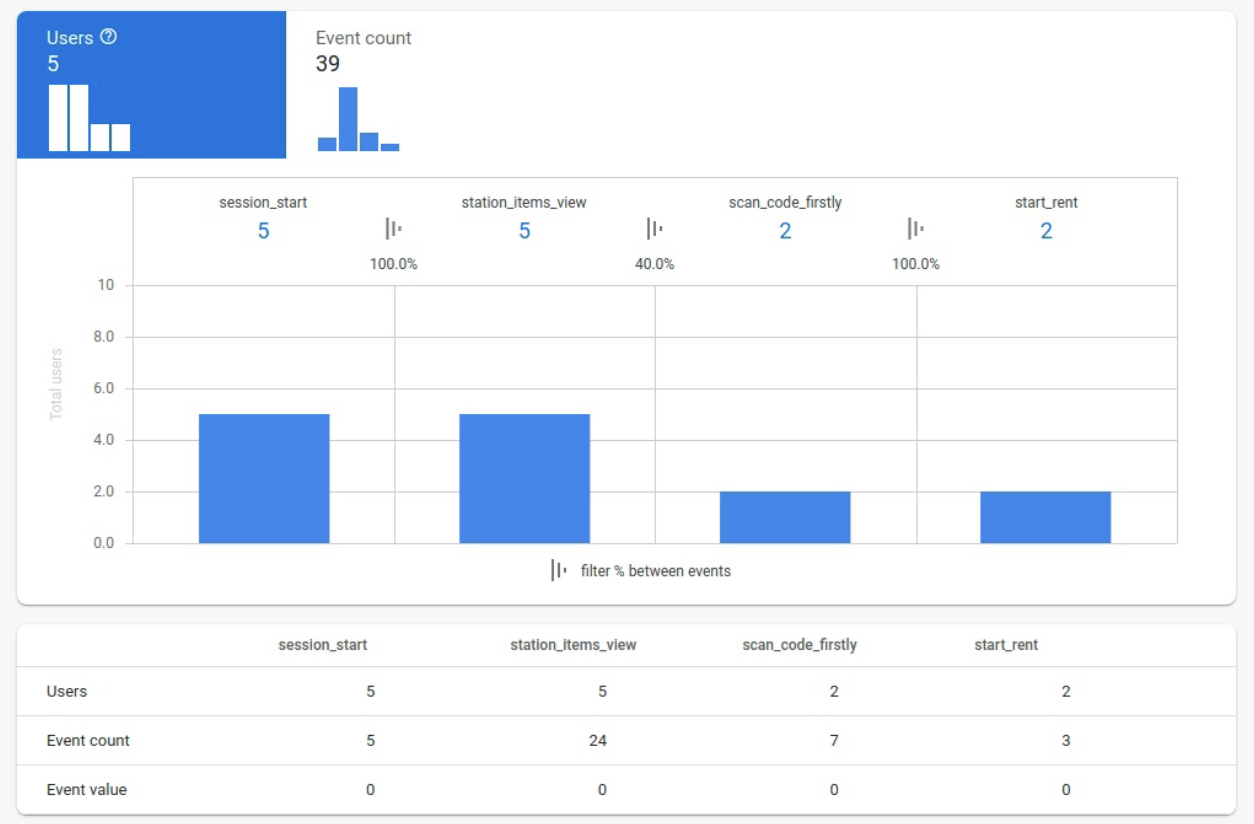


Рисунок 11. Метрики воронки оформления заявки.

1. Посетил начальную страницу – Перешел на страницу просмотра списка доступного оборудования - Отсканировал QR-код – Оформил заявку – Оплатил

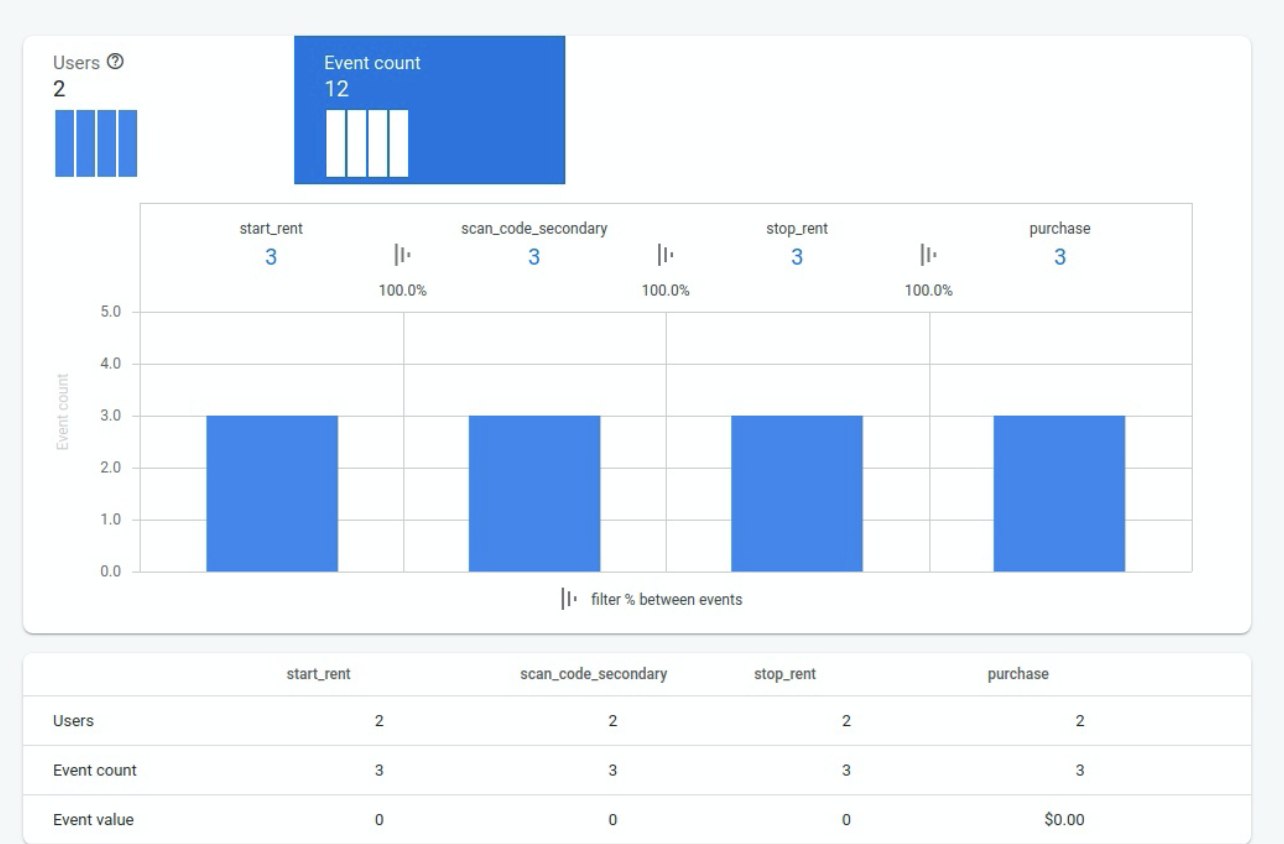


Рисунок 12. Метрики воронки оплаты.

## 4.3 Графический интерфейс

### 4.3.1 Начальный экран с выбором пункта проката

Предоставляет пользователю возможность выбрать точку проката или отсканировать QR-код.

Начальный экран изображен на рисунке 13.

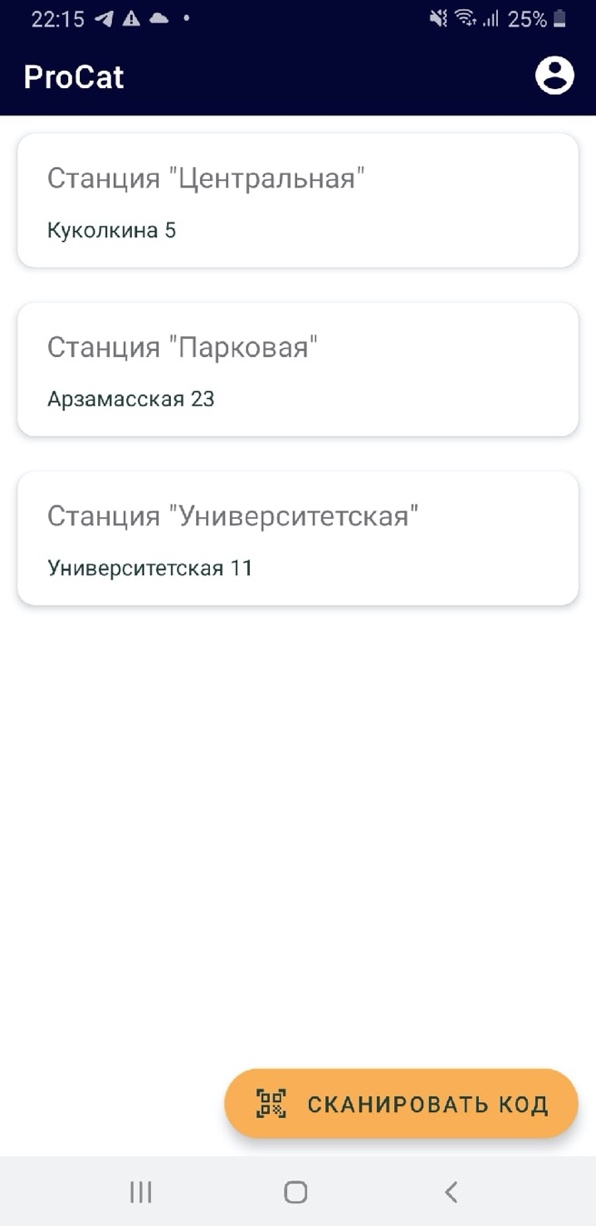


Рисунок 13 – Начальный экран

### 4.3.2 Экран с выбором снаряжения

Предоставляет пользователю возможность просмотреть доступное снаряжение и выбрать нужный предмет путем сканирования QR-кода.

Экран с выбором снаряжения изображен на рисунке 14.

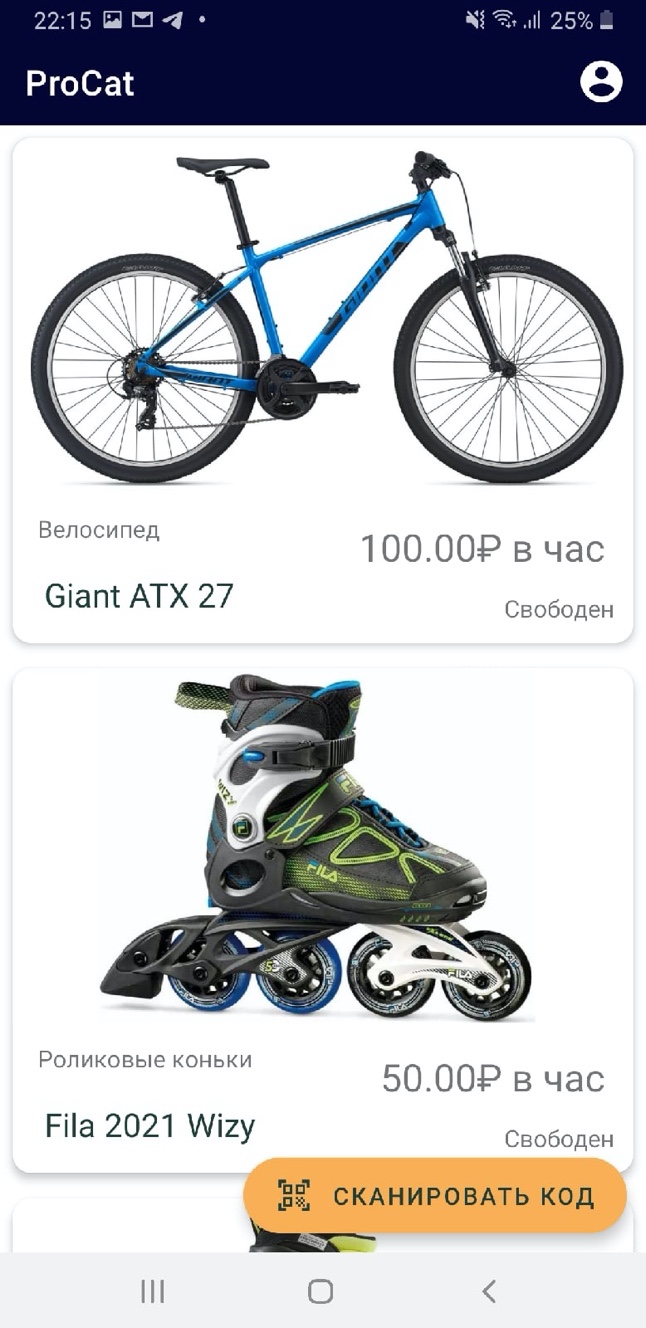


Рисунок 14 – Экран с выбором снаряжения

### 4.3.3 Экран проверки номера телефона

Проверка на то, был ли ранее зарегистрирован пользователь с введенным номером телефона. Для проверки необходимо ввести номер телефона и проверочный код.

Экран с проверкой номера телефона изображен на рисунке 15.

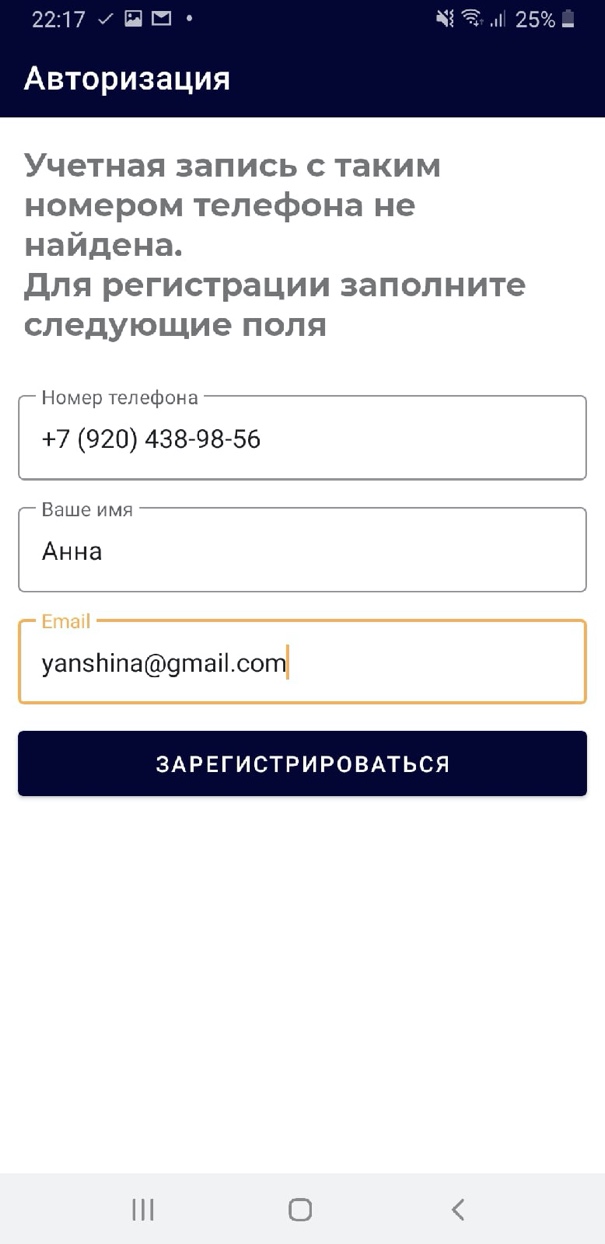


Рисунок 15 – Экран с проверкой номера телефона

### 4.3.4 Экран регистрации

Предоставляет пользователю возможность регистрации в случае, если пользователь с введенным номером телефона ранее не был зарегистрирован. Для регистрации необходимо ввести номер телефона, имя, email.

Экран регистрации изображен на рисунке 16.

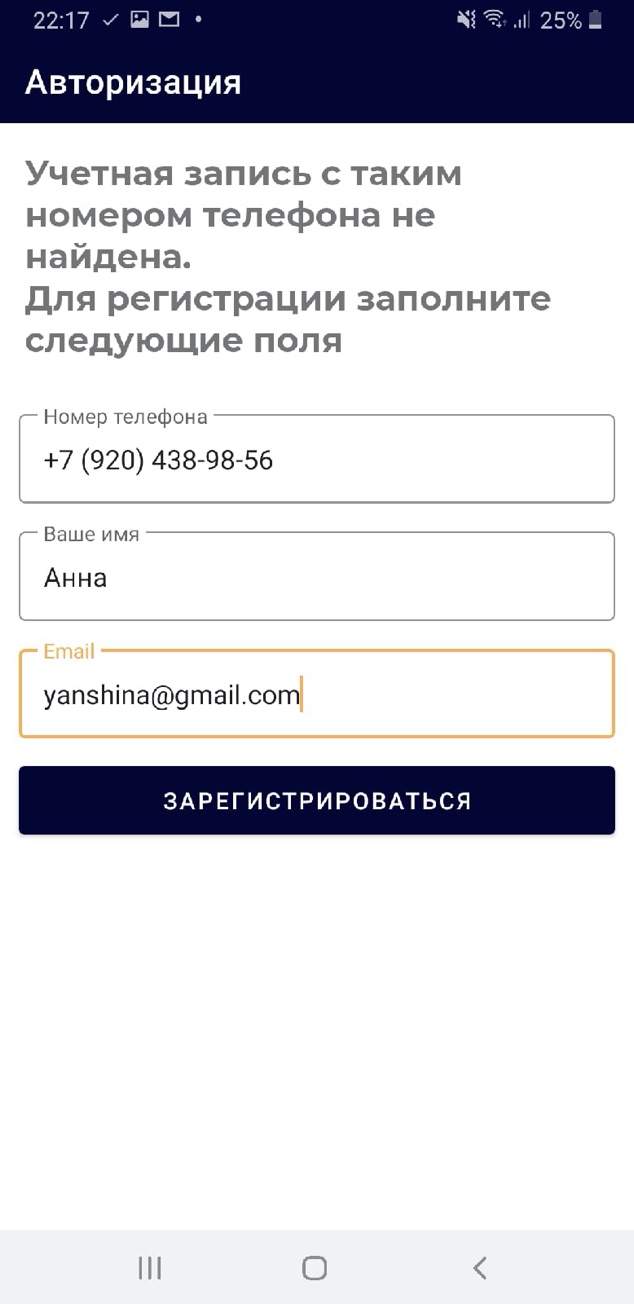


Рисунок 16 – Экран регистрации

### 4.3.5 Экран авторизации

Предоставляет пользователю возможность авторизации в случае, если пользователь с введенным номером телефона ранее был зарегистрирован.

Экран авторизации изображен на рисунке 17.

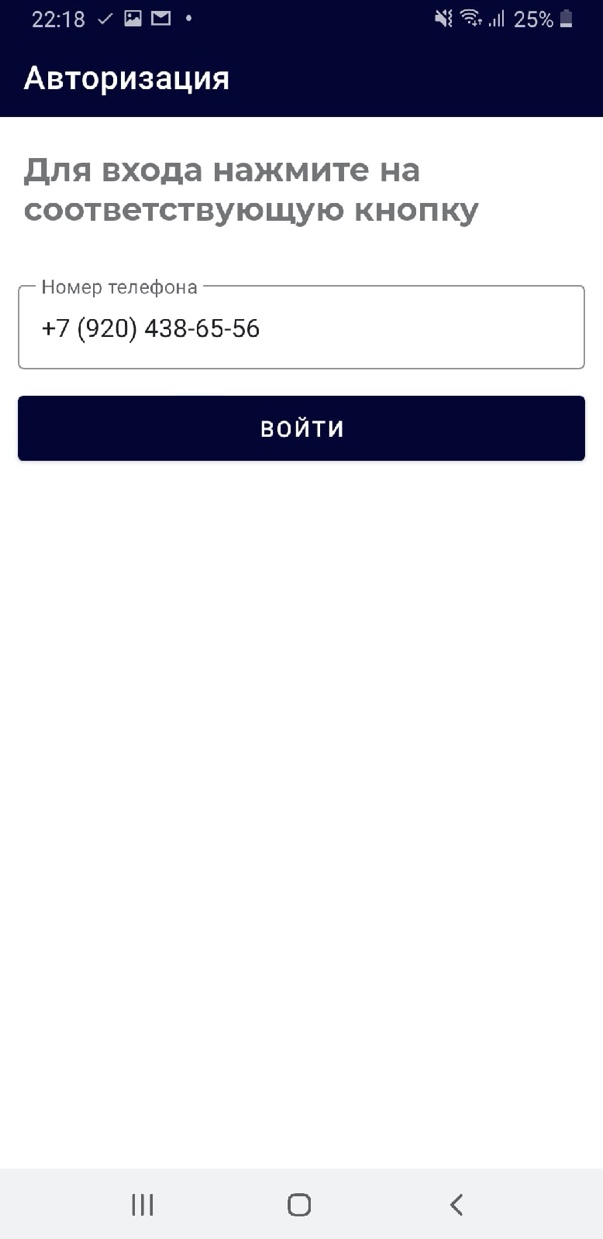


Рисунок 17 – Экран авторизации

### 4.3.6 Экран профиля

Предоставляет следующие возможности авторизованному пользователю:

- посмотреть личную информацию

- посмотреть список аренд

- сканировать QR-код предмета

- выйти из личного кабинета

- вернуться на начальную страницу

В случае неавторизированного пользователя будут предоставлен возможности входа, регистрации, сканирования QR-кода .

Экраны профиля изображены на рисунке 18 и рисунке 19.

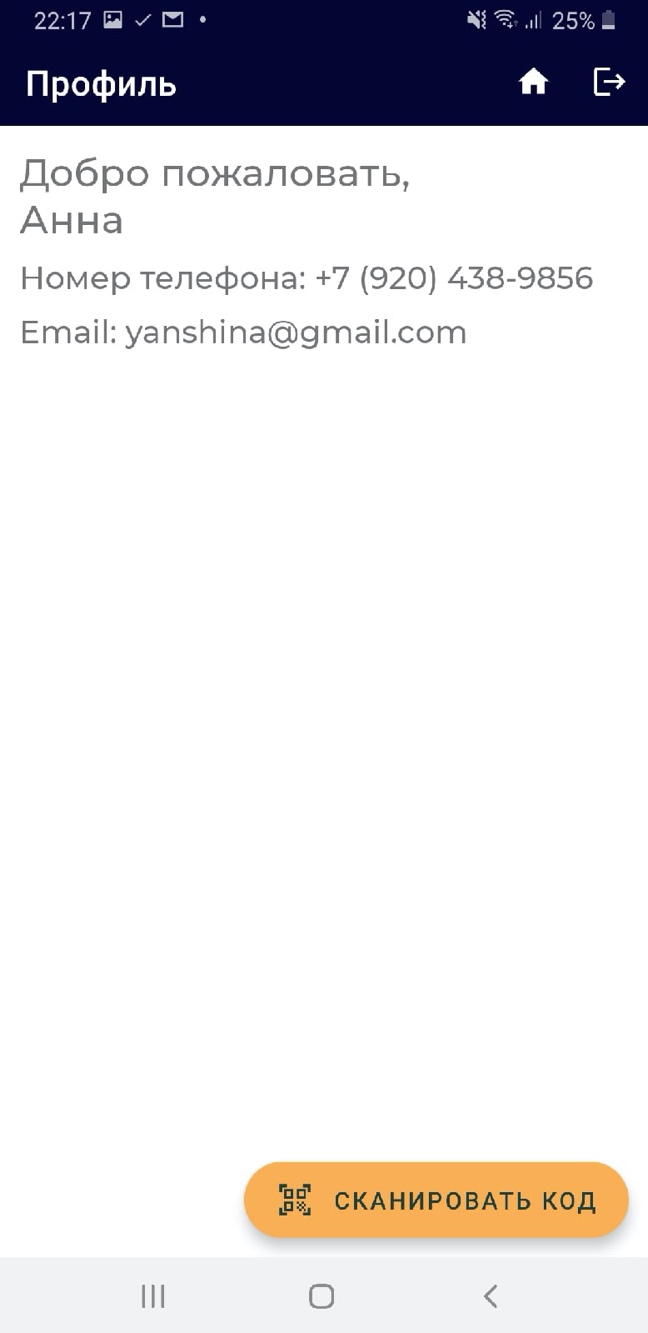


Рисунок 18 – Экран профиля авторизированного пользователя

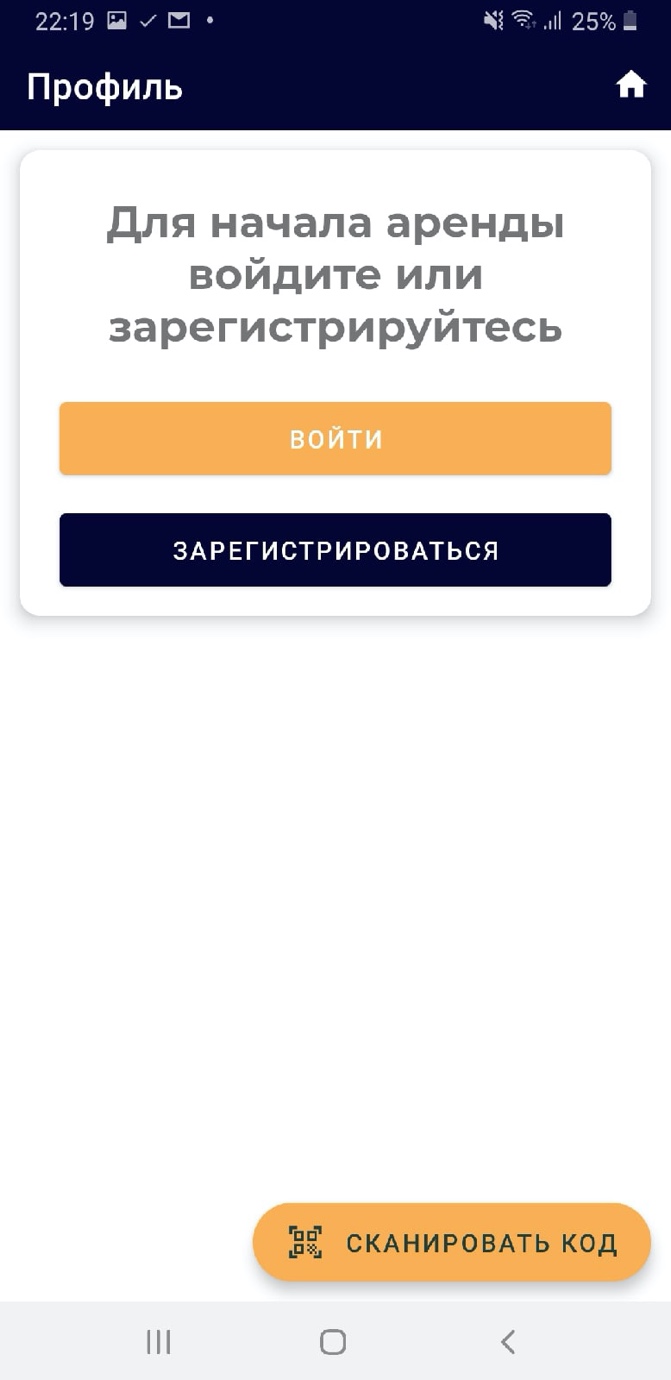


Рисунок 19 – Экран профиля неавторизированного пользователя

### 4.3.7 Экран сканирования QR-кода

Предоставляет пользователю возможность сканирования QR-кода спортивного снаряжения для просмотра информации о предмете.

Экран сканирования QR-кода изображен на рисунке 20.

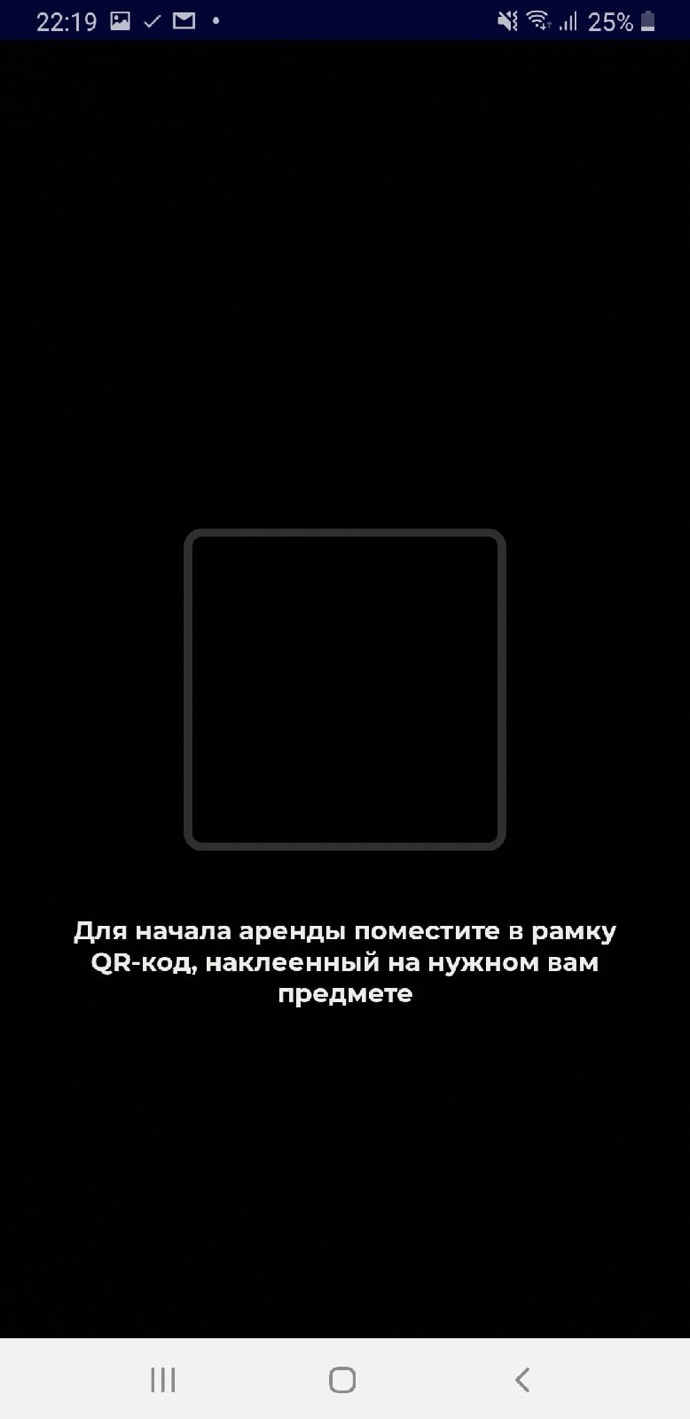


Рисунок 20 – Экран сканирования QR-кода

### 4.3.8 Экран с информацией о сканируемом предмете

После сканирования QR-кода выводится информация о предмете проката. Он может быть уже арендован другим пользователем, тогда появится сообщения об этом, может быть свободен, тогда будет отображаться кнопка «Начать аренду», а может быть ранее арендован пользователем, тогда появится кнопка «Завершить аренду».

Экраны с информацией о сканируемом предмете изображены на рисунке 21, рисунке 22 и рисунке 23.

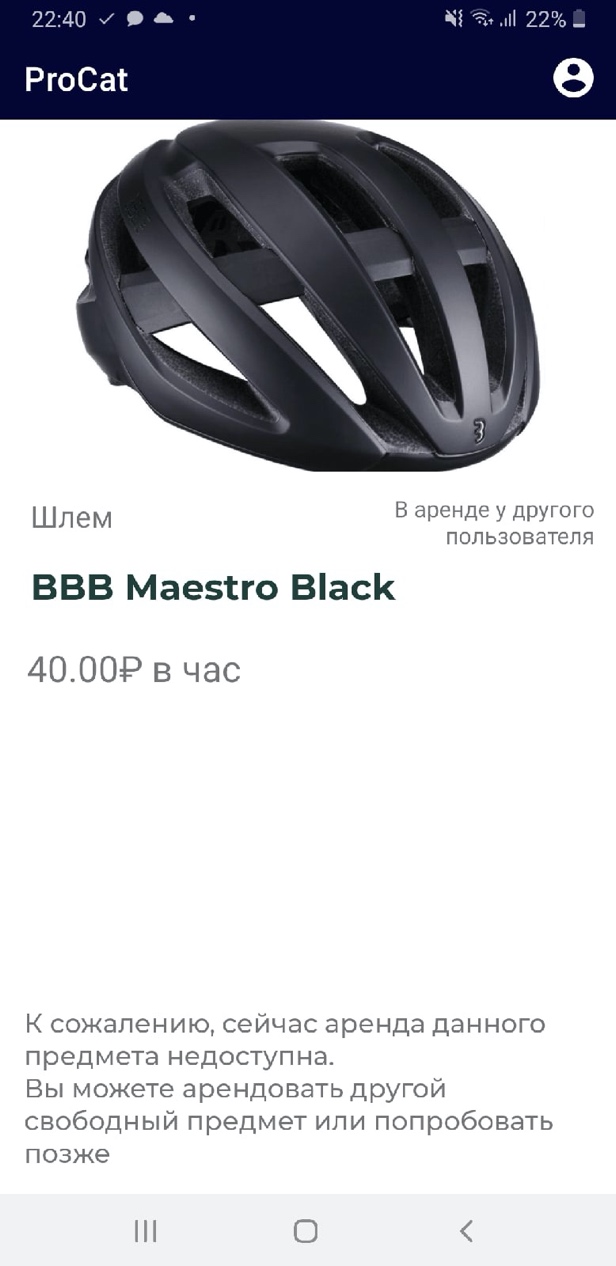


Рисунок 21 – Экран с информацией о сканируемом предмете (аренда недоступна)

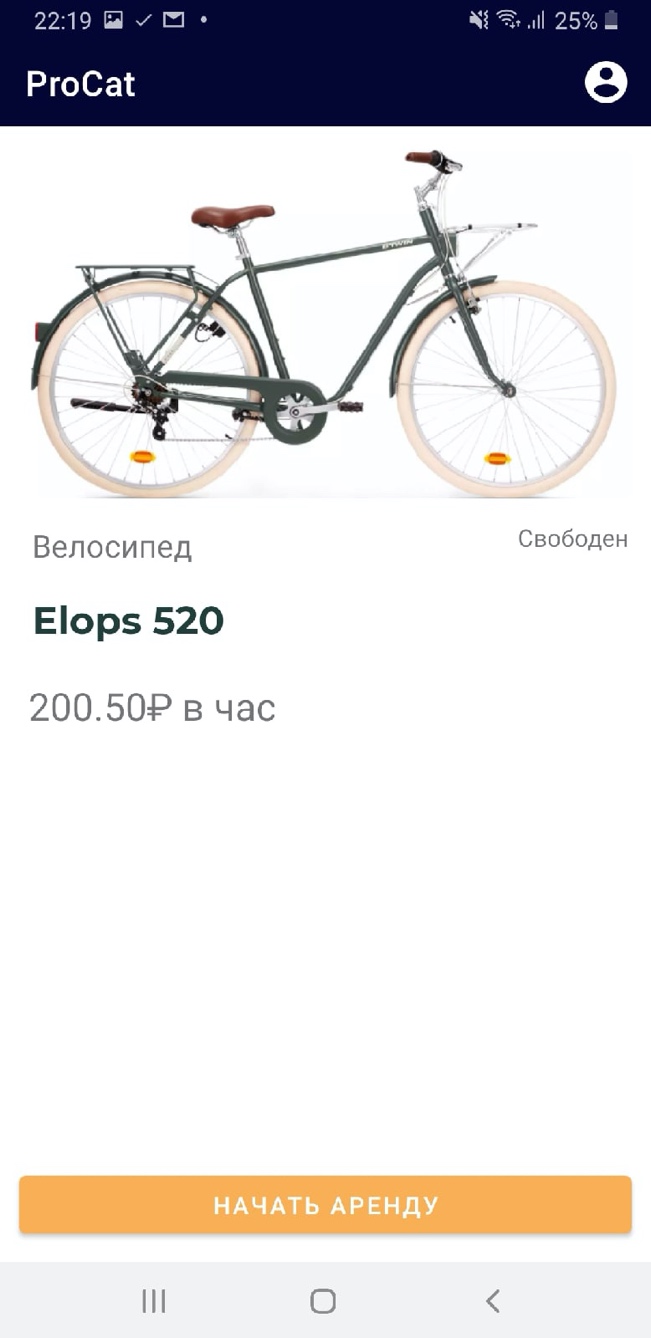


Рисунок 22 – Экран с информацией о сканируемом предмете (аренда доступна)

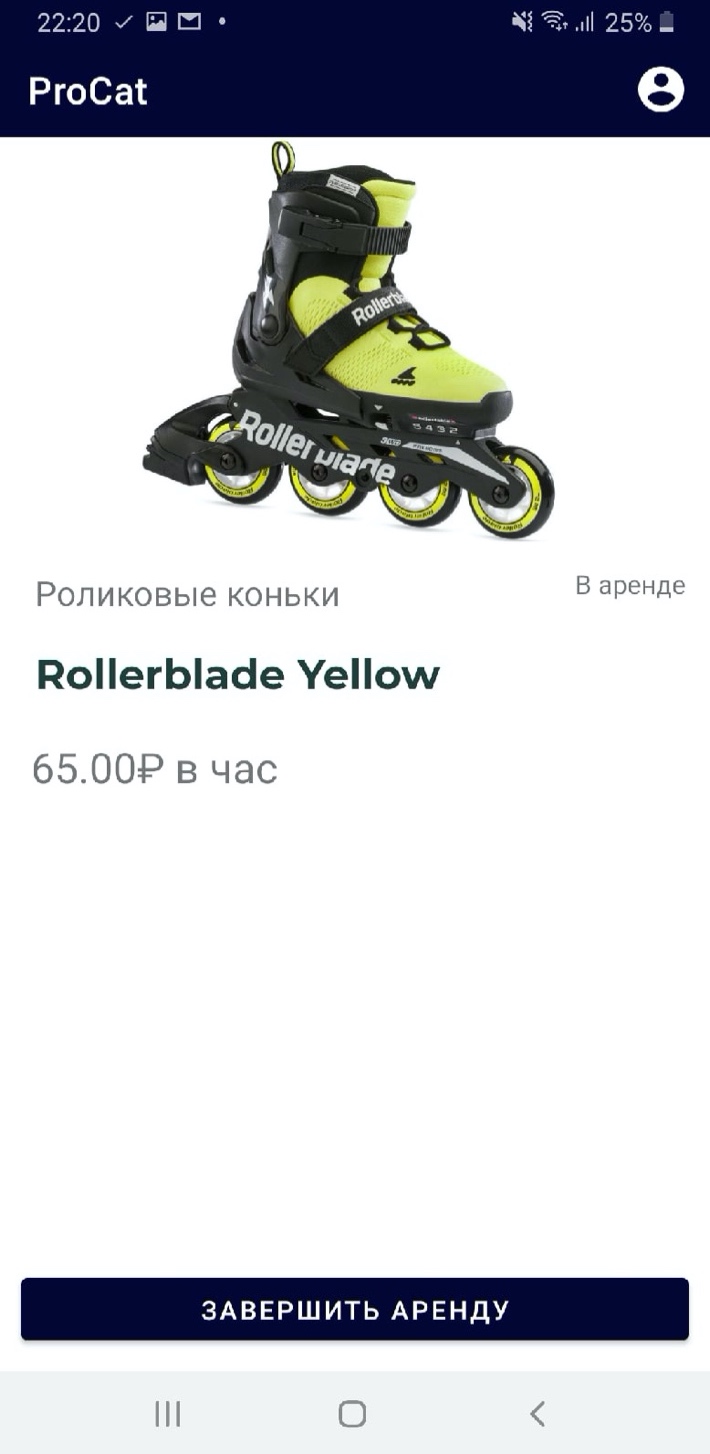


Рисунок 23 – Экран с информацией о сканируемом предмете (завершение аренды)

### 4.3.9 Экран оплаты

После завершения аренды требуется ее оплата. Экран с оплатой позволяет произвести ее путем ввода номера банковской карты. После оплаты выводится чек с подробной информацией и появляется возможность вернуться в личный кабинет.

Экраны оплаты изображены на рисунке 24 и рисунке 25.

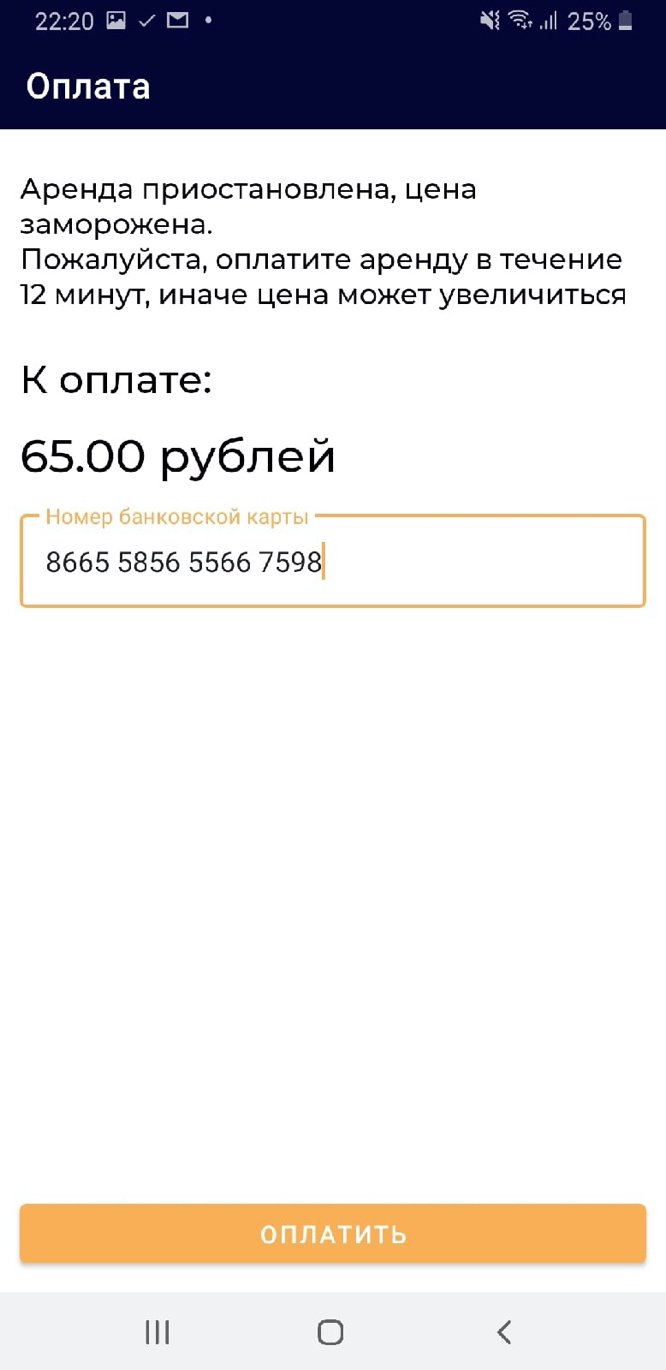


Рисунок 24 – Экран оплаты (ввод номера карты)

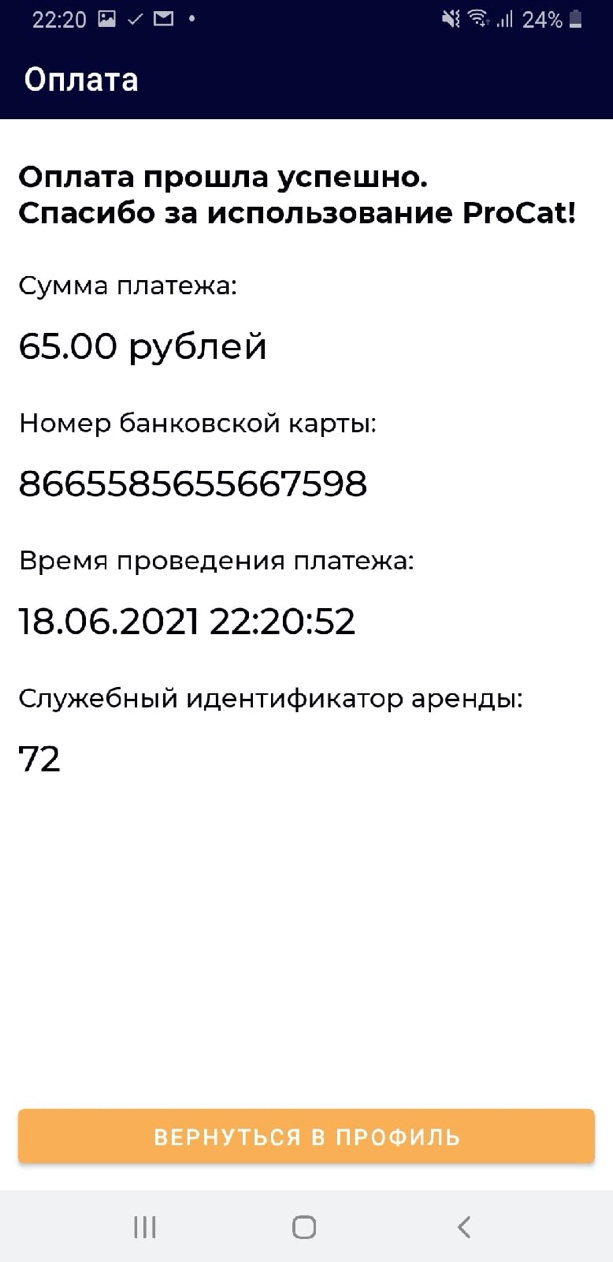


Рисунок 25– Экран оплаты (подробная информация)

### 4.3.10 Экран арендованных предметов пользователем

После взятия предметов в аренду они появляются в личном кабинете пользователя.

Экран арендованных предметов пользователем изображен на рисунке 26.

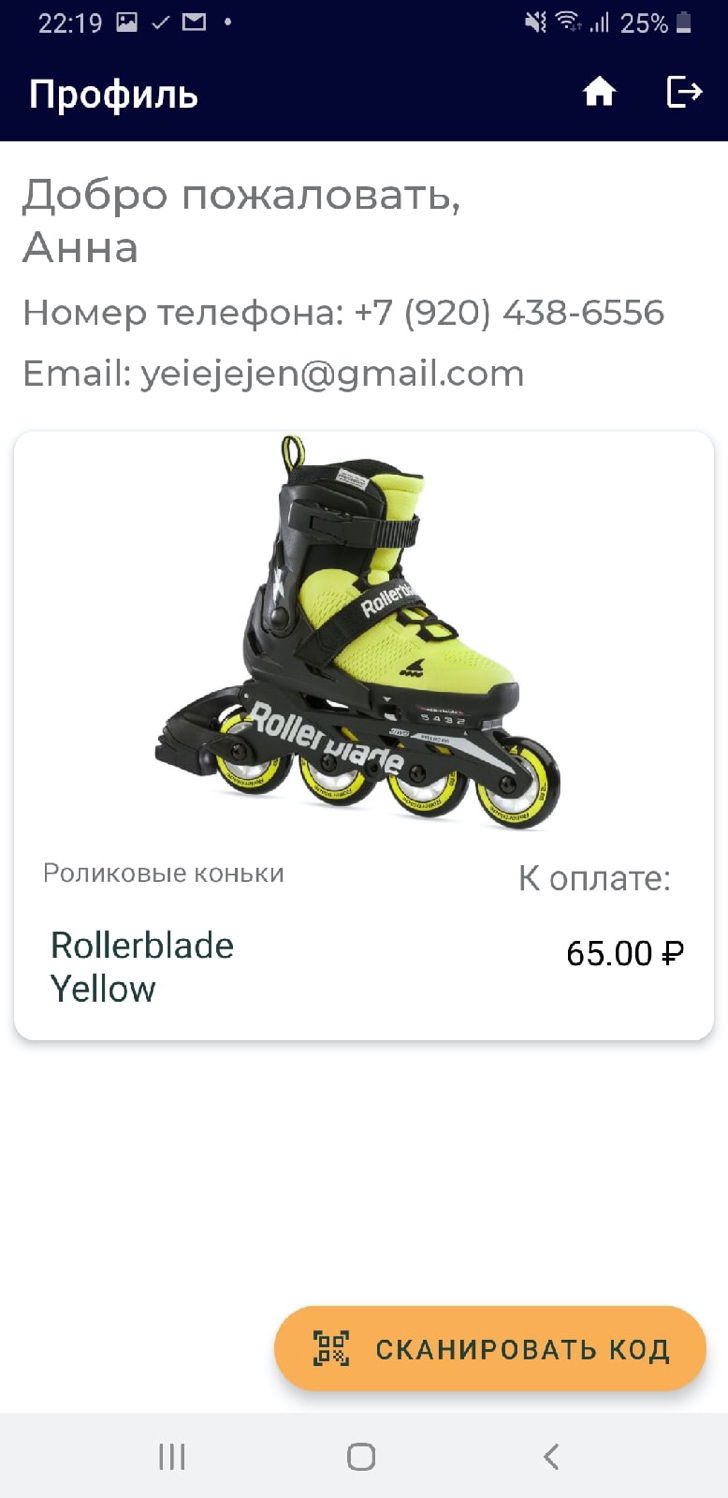


Рисунок 26– Экран арендованных предметов (подробная информация)

# 5. Тестирование

После реализации всех задач, был проведен запланированный набор тестов. Он включает 4 вида тестирования:

* + дымовое тестирование,
  + UI тесты,
  + юзабилити тесты
  + UNIT тесты

## 5.1 Дымовое тестирование

Для данного тестирования необходимо было проверить работоспособность приложения на следующих основных сценариях:

* регистрация,
* авторизация,
* сканирование QR-кода,
* взятие в аренду,
* просмотр списка аренд
* завершение аренды,
* оплата.

Дымовое тестирование проводилось ручным способом на смартфоне Android версии 10   
 с включенным WI-FI для связи с back-end частью сайта. Результаты, полученные в ходе тестирования представлены в таблице 1.

Таблица 1. Результаты дымового тестирования.

|  |  |
| --- | --- |
| **Сценарий** | **Результат** |
| Регистрация | Пройден |
| Авторизация | Пройден |
| Сканирование QR-кода | Пройден |
| Взятие в аренду | Пройден |
| Просмотр списка аренд | Пройден |
| Завершение аренды | Пройден |
| Оплата | Пройден |

По итогу дымового тестирования было установлено, что приложение проходит все основные утвержденные сценарии.

## 5.2 UI тесты

В результате UI тестирования было выполнено 20 тестов, охватывающие основные возможности приложения.

Таблица 2. Результаты UI тестирования.

|  |  |  |
| --- | --- | --- |
| Шаги теста | Ожидаемый результат | Статус |
| 1. Нажимается кнопки станции на начальном экране | Открылся экран со списком снаряжения на выбранной станции | Пройден |
| 1. Нажимается изображение снаряжения на экране выбора спортивного снаряжения | Показывается предупреждение о необходимости сканирования QR-кода на данном предмете | Пройден |
| 1. Нажимается кнопка «Сканировать код» на экране выбора спортивного снаряжения | Открылся экран сканирования QR-кода | Пройден |
| 1. Сканирование QR-кода | Открылся экран с информацией об отсканированном предмете | Пройден |
| 1. Нажимается кнопка «Начать аренду» в случае сканирования свободного предмета | Открылся экран профиля, высветилось сообщение об успешной начале аренды | Пройден |
| 1. Нажимается кнопка «Завершить аренду» в случае сканирования ранее взятого предмета.  2. В поле вводится корректный номер банковской карты и нажимается кнопка «Оплатить»  3. Нажимается кнопка «Вернуться в профиль» | 1. Открылся экран оплаты (ввод номера карты)  2. Открывается экран с подробной информацией об оплаченном заказе  3. Открывается экран профиля | Пройден |
| 1. Нажимается кнопка «Завершить аренду» в случае сканирования ранее взятого предмета.  2. В поле вводится некорректный номер банковской карты(пусто или меньшее кол-во символов) и нажимается кнопка «Оплатить» | 1. Открылся экран оплаты (ввод номера карты)  2. Появилось сообщение о том, что необходимо проверить правильность номера банковской карты | Пройден |
| 1. Сканирование QR-кода  2. Появилось сообщение о том, что предмет уже находится в аренде у другого пользователя | 1. Открылся экран с информацией об отсканированном предмете  2. Невозможность взять в аренду или завершить аренду для выбранного предмета | Пройден |
| 1. Нажимается иконка личного кабинета (авторизированный пользователь) | Открывается страница личного кабинета | Пройден |
| 1. Нажимается иконка личного кабинета (неавторизированный пользователь) | Открывается страница личного кабинета неавторизированного пользователя с выбором входа или регистрации | Пройден |
| 1. Нажимается кнопка «Войти»  2. В поле вводятся некорректные данные и нажимается кнопка «Получить код подтверждения» | 1. Открылась страница с авторизацией  2. Появилось сообщение о том, что необходимо проверить правильность номер телефона | Пройден |
| 1. Нажимается кнопка «Войти»  2. В поле телефона вводятся корректные данные и нажимается кнопка «Получить код подтверждения»  3. В поле ввода кода вводится некорректный код и нажимается кнопка «Продолжить» | 1. Открылась страница с авторизацией  2. Появилось дополнительное поле для ввода кода подтверждения  3. Появилось сообщение о том, что введен неверный код | Пройден |
| 1. Нажимается кнопка «Войти»  2. В поле телефона вводятся корректные данные(ранее незарегистрированный телефон) и нажимается кнопка «Получить код подтверждения»  3. В поле ввода кода вводится корректный код и нажимается кнопка «Продолжить» | 1. Открылась страница с авторизацией  2. Появилось дополнительное поле для ввода кода подтверждения  3. Открылся экран регистрации | Пройден |
| 1. В открытом экране регистрации вводится корректное имя и email и нажимается кнопка «Зарегистрироваться» | Открылся экран профиля | Пройден |
| 1. В открытом экране регистрации вводится некорректное или пустое имя и нажимается кнопка «Зарегистрироваться» | 1. Появилось сообщение о том, что имя не может быть меньше трех символов | Пройден |
| 1. В открытом экране регистрации вводится корректное имя,email остается пустым или вводится некорректно и нажимается кнопка «Зарегистрироваться» | 1. Появилось сообщение о необходимости проверить правильность введенного email. | Пройден |
| 1. Нажимается кнопка «Выхода из личного кабинета» | Открылся экран профиля неавторизированного пользователя | Пройден |
| 1. Нажимается кнопка «Зарегистрироваться»  2. В поле телефона вводятся корректные данные (ранее незарегистрированный телефон) и нажимается кнопка «Получить код подтверждения»  3. В поле ввода кода вводится корректный код и нажимается кнопка «Продолжить» | 1. Открылась страница с регистрацией  2. Появилось дополнительное поле для ввода кода подтверждения  3. Открылся экран регистрации | Пройден |
| 1. Нажимается кнопка «Войти»  2. В поле телефона вводятся корректные данные(ранее зарегистрированный телефон) и нажимается кнопка «Получить код подтверждения»  3. В поле ввода кода вводится корректный код и нажимается кнопка «Продолжить»  4. Нажимается кнопка «Войти» | 1. Открылась страница с авторизацией  2. Появилось дополнительное поле для ввода кода подтверждения  3. Открылся экран авторизации  4. Открылся экран профиля | Пройден |
| 1. Авторизированный пользователь берет в аренду предмет. Нажимается кнопка личного кабинета | В личном кабинете отображен предмет, взятый в аренду | Пройден |
| 1.Авторизированный пользователь берет в аренду несколько предметов. Нажимается кнопка личного кабинета | В личном кабинете отображены предметы взятые в аренду | Пройден |
| 1.Неавторизированный пользователь нажимает на кнопку «Сканировать код».  2. Сканируется QR-код свободного предмета | 1. Открывается экран сканирования QR-кода  2. Происходит переход на страницу авторизации с сообщением о необходимости авторизоваться для взятия предмета в аренду | Пройден |
| 1.Неавторизированный пользователь нажимает на кнопку «Сканировать код».  2. Сканируется QR-код занятого предмета | 1. Открывается экран сканирования QR-кода  2. Появляется сообщение о том, что предмет уже в аренде у другого пользователя | Пройден |
| 1.Открыта страница личного кабинета. Нажимается кнопка возвращения на начальную страницу | Происходит возвращение на начальную страницу | Пройден |

## 5.3 Юзабилити тестирование

Для проведения юзабилити тестирования было случайно отобрано 3 человека, не пользовавшиеся заранее приложение. Для данного тестирования необходимо проверить следующие основные сценарии взаимодействия пользователя с приложением:

* регистрация,
* авторизация,
* сканирование QR-кода,
* взятие в аренду,
* просмотр списка аренд
* завершение аренды,
* оплата.

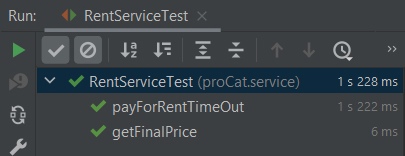
Результаты тестирования отображены в таблице 3.

Таблица 3. Результаты юзабилити тестирования.

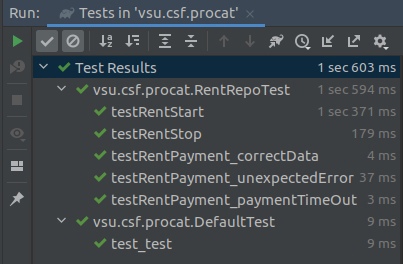
|  |  |  |  |
| --- | --- | --- | --- |
| Сценарий | Пользователь 1 | Пользователь 2 | Пользователь 3 |
| Регистрация | Пройден | Пройден | Пройден |
| Авторизация | Пройден | Пройден | Пройден |
| Сканирование QR-кода | Пройден | Пройден | Пройден |
| Взятие в аренду | Пройден | Пройден | Пройден |
| Просмотр списка аренд | Пройден | Пройден | Пройден |
| Завершение аренды | Пройден | Пройден | Пройден |
| Оплата | Пройден | Пройден | Пройден |

## 5.4 UNIT тесты

1) метод сервиса, отвечающий за оплату (back-end)



2) тестирование работы репозитория, который получает и отправляет данных об оплате (android)



# Заключение

В результате работы было разработано приложение по аренде спортивного снаряжения. Были выполнены следующие задачи:

- была разработана Android часть приложения, находящаяся на телефоне пользователя;

- был разработан Back-end части приложения, развернутый на удаленном сервере приложения;

- была создана связь между Android и Back-end частями приложения;

- была разработана база данных, расположенная на удаленном сервере;

- были проанализированы метрики и сделаны выводы о том, что предложенная бизнес-модель работает.

Приложение отвечает всем заявленным требованиям.