

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

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Приложение для аренды автомобилей «FarSharing»

Курсовой проект

09.03.02 Информационные системы и технологии
Программная инженерия в информационных системах

Зав. кафедрой _____ д. ф.-м. н., профессор С.Д. Махортов

Руководитель _____ *ст. преподаватель* В.С. Тарасов

_____ И.В. Клейменов

Обучающиеся:

_____ В.И. Валеев

_____ Е.Р. Ивашечкин

_____ П.С. Мишин

_____ К.С. Бондаренко

Воронеж 2022

Содержание

Содержание.....	2
Введение.....	4
1. Постановка задачи.....	5
1.1 Функциональные требования	5
1.2 Технические требования	6
1.3 Требования к интерфейсу.....	6
2. Анализ предметной области	7
2.1 Анализ существующих решений.....	7
2.1.1 Яндекс.Драйв.....	7
2.1.2 Делимобиль	7
2.2 Анализ задачи.....	8
2.2.1 Варианты использования приложения	8
2.2.2 Взаимодействие компонентов системы.....	10
2.2.3 Взаимодействие компонентов системы.....	12
2.2.4 Развертывание приложения	14
2.2.5 Модель базы данных	15
2.2.6 Входные-выходные данные (IDEF0)	16
2.2.7 Воронки конверсии	17
3. Реализация.....	19
3.1 Frontend	19
3.2 Backend.....	19
4. Интерфейс	20
4.1 Интерфейс неавторизованного пользователя	20
4.2 Интерфейс авторизованного пользователя	24
4.3 Интерфейс пользователя, авторизованного под ролью «Администратор»	26
5. Тестирование	27
5.1 Модульное тестирование серверной части приложения	27
5.1.1 CarService	28

5.1.2 ClientService	28
5.1.3 ContractService	28
5.1.4 UserService	29
5.2 Яндекс метрики	30
6. Заключение	31
Список используемых источников	33

Введение

Прокат автомобилей востребован в современном обществе. Рынок предъявляет высокие требования к уровню доступности, качеству и удобству использования сервиса. Немаловажную роль играют уникальные свойства продукта, включая, визуализацию, маркетинговые особенности и т.д.

Одним из видов проката автомобилей является каршеринг. Каршеринг (от англ. car sharing) — это автоматизированный почасовой или поминутный прокат автомобиля. Услуга предназначена для коротких поездок по городу. Доступные для аренды автомобили стоят в разных частях города, об их точном расположении можно узнать из мобильных приложений для смартфонов. Через приложение частично осуществляется управление автомобилем – бронирование, запуск двигателя, открытие дверей.

В настоящее время на рынке представлено более 20 компаний, предоставляющих услуги каршеринга. Для предоставления данных услуг используется более 10 мобильных приложений, основными из которых являются Яндекс.Драйв, Делимобиль, BelkaCar, YouDrive, Bumerang, Rentmee, Cars7, URАмобиль и Vi-Bi.car. Основными параметрами, которыми должны обладать данные приложения являются: функциональность, удобство использования, информационная безопасность, производительность и надёжность, правовая оценка, информация о приложении.

При разработке проекта используется каскадная модель, которая включает в себя следующие этапы:

1. Определение требований.
2. Проектирование.
3. Конструирование (также «реализация» либо «кодирование»).
4. Воплощение.
5. Тестирование и отладка (также «верификация»).
6. Инсталляция.
7. Поддержка.

1. Постановка задачи

Целью курсового проекта является создание мобильного приложения, выполняющего функции по аренде автомобилей.

1.1 Функциональные требования

В системе предусмотрены три уровня доступа:

1. Неавторизованный пользователь;
2. Авторизованный пользователь;
3. Администратор.

Неавторизованный пользователь обладает следующими возможностями:

- Регистрация;
- Подтверждение почты;
- Авторизация.

Авторизованный пользователь обладает следующими возможностями:

- Просмотр доступных автомобилей;
- Отправка заявки на бронь;
- Оплата брони;
- Отмена брони;
- Просмотр личных данных;
- Связь с технической поддержкой;
- Изменение личных данных;
- Выход из системы;
- Удаление аккаунта.

Администратор обладает следующими возможностями:

- Обработка брони;
- Добавление автомобиля;
- Выход из системы;
- Просмотр данных пользователя;
- Блокировка пользователя;
- Разблокировка пользователя;

1.2 Технические требования

Приложение должно обеспечивать:

- Авторизацию его пользователей посредством email и пароля;
- Автоматическую корректировку данных о арендованных автомобилях.

1.3 Требования к интерфейсу

Интерфейс должен обладать:

- Раздельным интерфейсом для пользователя и администратора;
- Отображать списки данных, перечисленные в функциональных требованиях.

2. Анализ предметной области

2.1 Анализ существующих решений

2.1.1 Яндекс.Драйв

Яндекс.Драйв является самой развитой и известной сетью на данный момент.

Достоинства:

1. Наличие личного кабинета;
2. С помощью фильтров легко найти интересующий автомобиль;
3. Наличие раздела «Отзывы»;
4. Встроенный навигатор;

Недостатки:

1. Довольно сложный интерфейс;
2. Время ожидания ответа службы поддержки;
3. Непрозрачное ценообразование.

2.1.2 Делимобиль

Делимобиль является одним из ключевых игроков на российском рынке.

Достоинства:

1. Наличие личного кабинета;
2. Дают время на осмотр автомобиля;
3. Огромный автопарк;
4. Наличие скидочной системы;
5. Наличие раздела «Отзывы».

Недостатки:

1. Поминутная аренда автомобиля;
2. Не компетентная поддержка;
3. Затянутая регистрация.

2.2 Анализ задачи

2.2.1 Варианты использования приложения



Рисунок 1. Диаграмма прецедентов.

Пользователь имеет возможность взаимодействовать с приложением без авторизации. При таком сценарии ему будут доступны следующие действия:

1. Регистрация:
 - Подтверждение почты.
2. Авторизация;

Также пользователь имеет возможность авторизоваться под ролью «Администратор» или ролью «Авторизованный пользователь». Если пользователь авторизуется под ролью «Администратор», то ему будут доступны следующие действия:

1. Обработка брони:
2. Добавить автомобиль,
3. Просмотр данных пользователей:
 - Заблокировать пользователя
 - Разблокировать пользователя
4. Выход из системы.

Если же пользователь авторизуется под ролью «Авторизованный пользователь», то ему будут доступны следующие действия:

1. Просмотр личных данных;
2. Удаление аккаунта;
3. Связь с технической поддержкой;
4. Изменение личных данных;
5. Выход из системы;
6. Просмотр доступных автомобилей;
7. Отправка заявки на бронь:
 - Оплата брони;
 - Отмена брони.

2.2.2 Взаимодействие компонентов системы

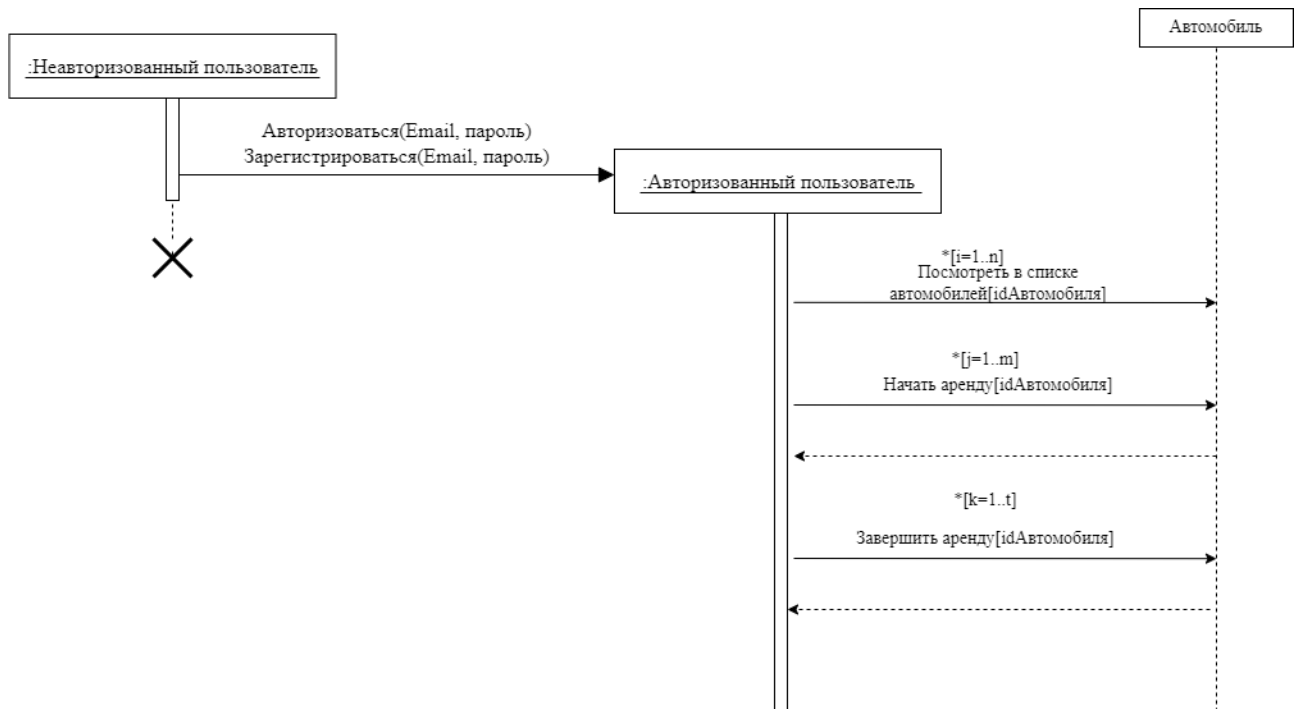


Рисунок 2. Диаграмма последовательностей для авторизованного пользователя.

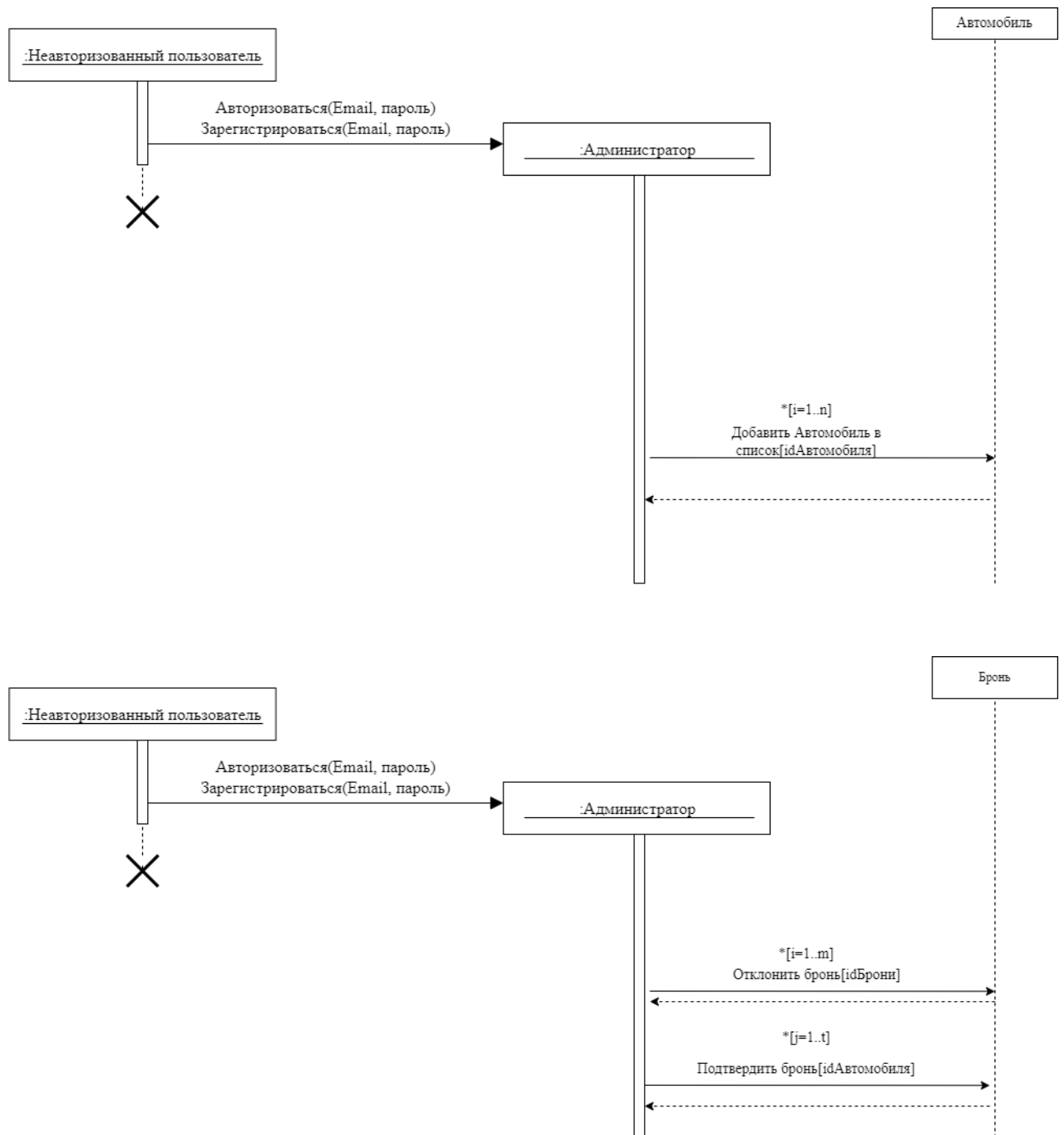


Рисунок 3. Диаграмма последовательностей для Авторизованного пользователя под ролью “Администратор”.

2.2.3 Взаимодействие компонентов системы

Возможные варианты взаимодействия компонентов системы, включающей неавторизованного пользователя, авторизованного пользователя и администратора, отражено на рисунках 4 – 6.

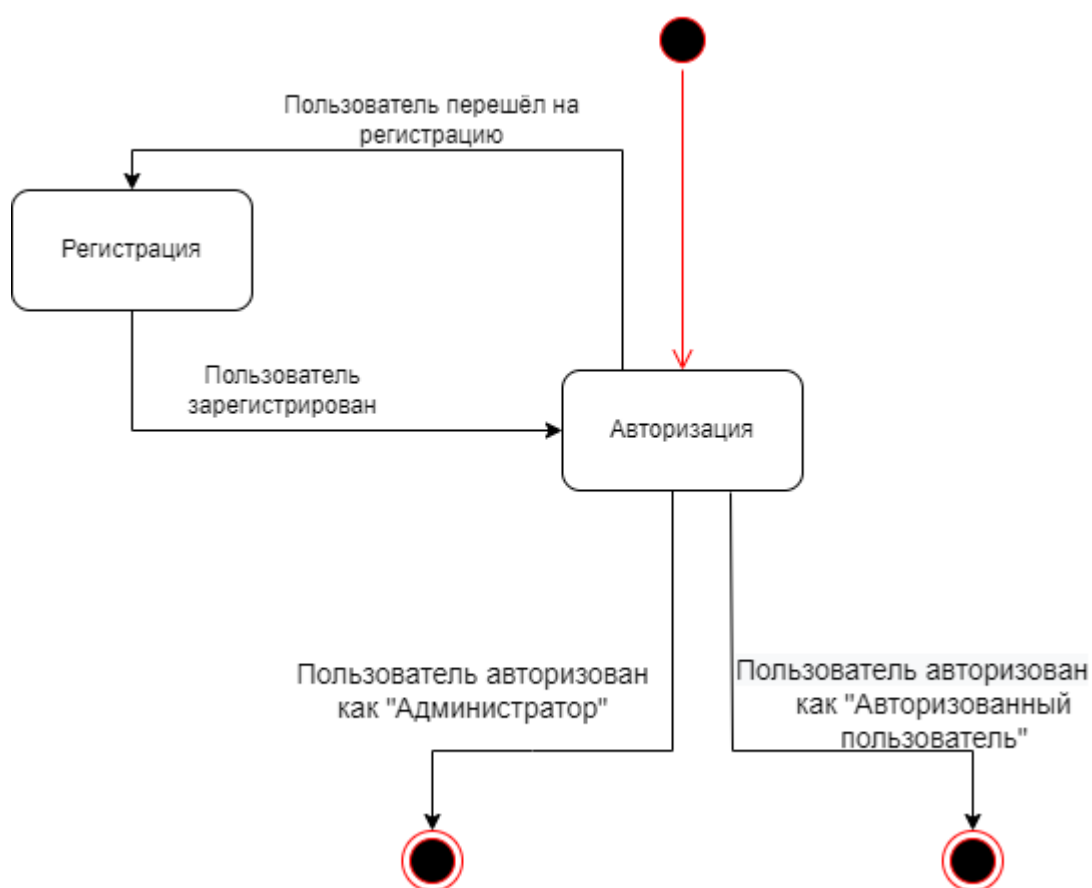


Рисунок 4. Диаграмма состояния неавторизованного пользователя.



Рисунок 5. Диаграмма состояния авторизованного пользователя.

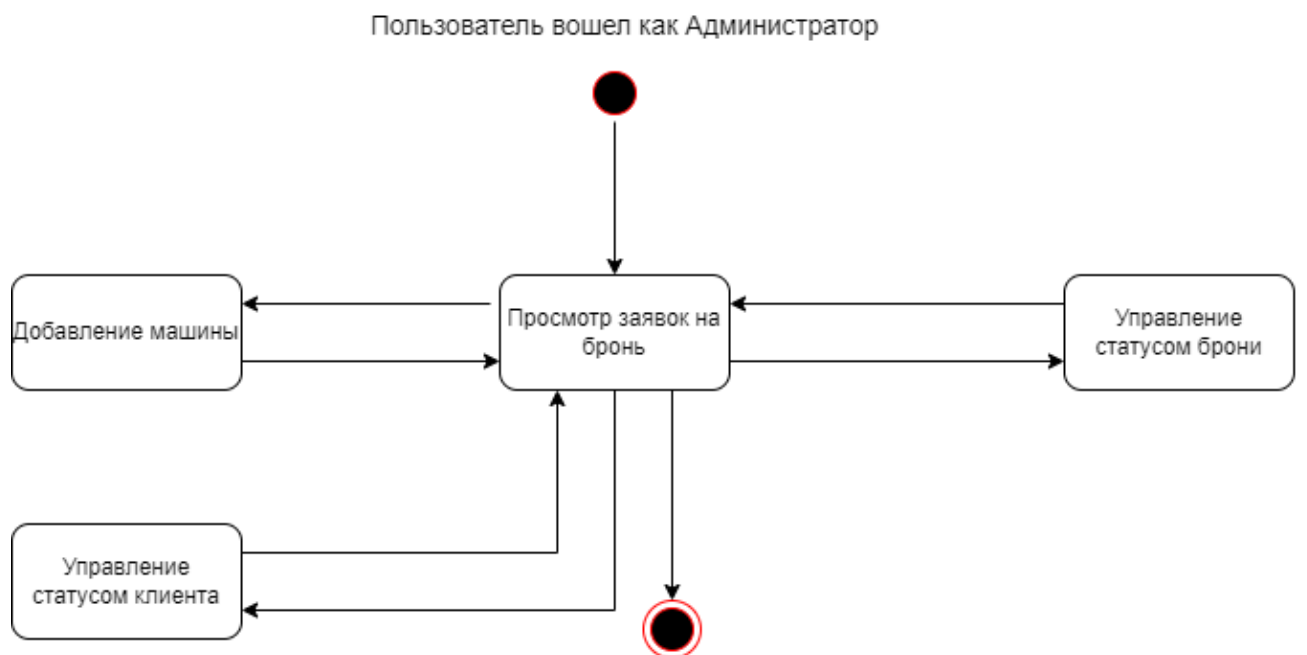


Рисунок 6. Диаграмма состояния администратора.

2.2.4 Развертывание приложения

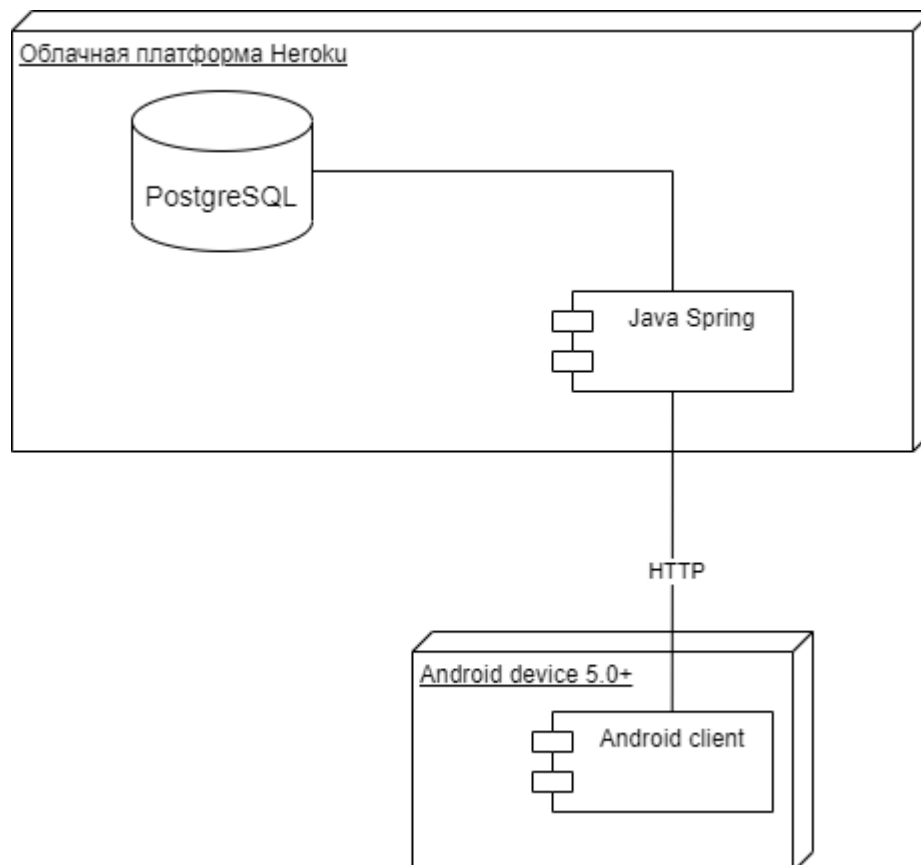


Рисунок 7. Диаграмма развертывания

На Рисунке 7 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты (узлы) существуют, какие программные компоненты (артефакты) работают на каждом узле, и как различные части этого комплекса соединяются друг с другом.

2.2.5 Модель базы данных

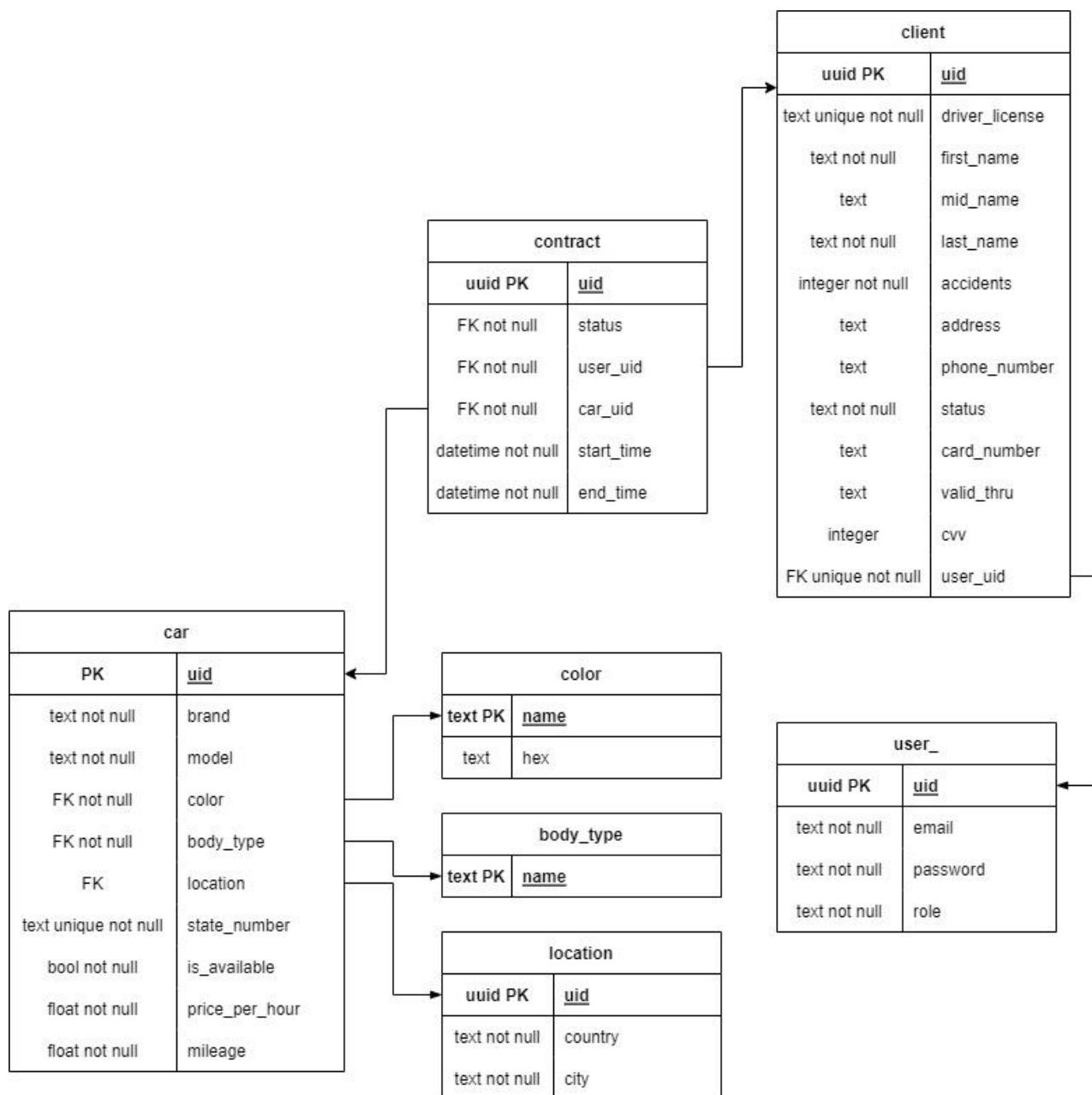


Рисунок 8. Схема базы данных.

На Рисунке 8 представлена схема базы данных для разрабатываемого приложения.

2.2.6 Входные-выходные данные (IDEF0)

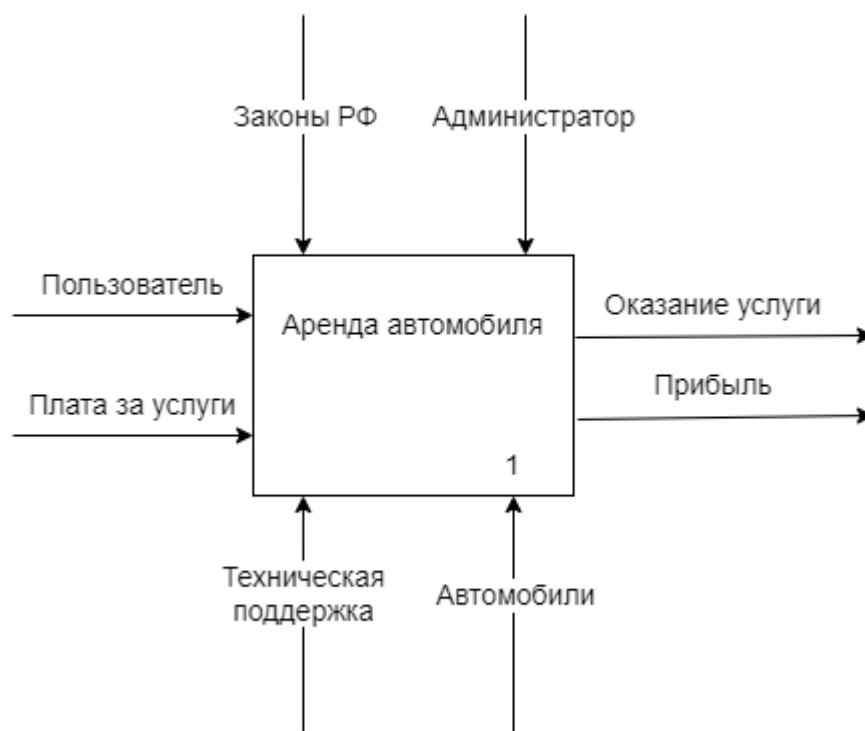


Рисунок 9. Контекстная диаграмма.

2.2.7 Воронки конверсии

Составная цель: Бронирование автомобиля

Посетить главную страницу – Перейти на выбранный автомобиль –
Запросить бронь – Оплатить бронь.



Рисунок 10. Воронка конверсии бронирования автомобиля.

Составная цель: Удаление аккаунта

Посетить главную страницу – Перейти в меню - Перейти в личный кабинет
– Удалить аккаунт.



Рисунок 11. Воронка конверсии удаления аккаунта.

Составная цель: Изменить личные данные

Посетить главную страницу - Перейти в меню – Перейти в личный кабинет
– Изменить личные данные.



Рисунок 12. Воронка конверсии завершения аренды автомобиля.

Составная цель: Обратиться в поддержку

Посетить главную страницу – Перейти в меню – Перейти в личный кабинет
– Написать сообщение в поддержку.

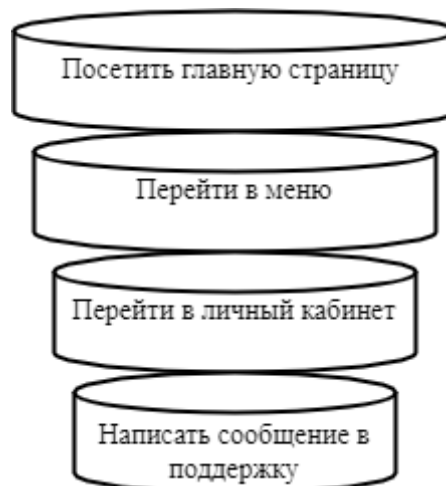


Рисунок 13. Воронка конверсии обращения в поддержку.

3. Реализация

3.1 Frontend

При создании клиентской части использовались набор инструментов Android SDK и REST-клиент Retrofit.

3.2 Backend

При создании серверной части использовался фреймворк Java Spring Boot. Для работы программы была выбрана Hibernate framework.

Документирование бэкенда приложения ведётся с помощью ПО с открытым исходным кодом Swagger. Этот инструмент позволяет разработчикам вести и предоставлять документацию REST веб-сервисов с возможностью тестирования запросов к методам API.

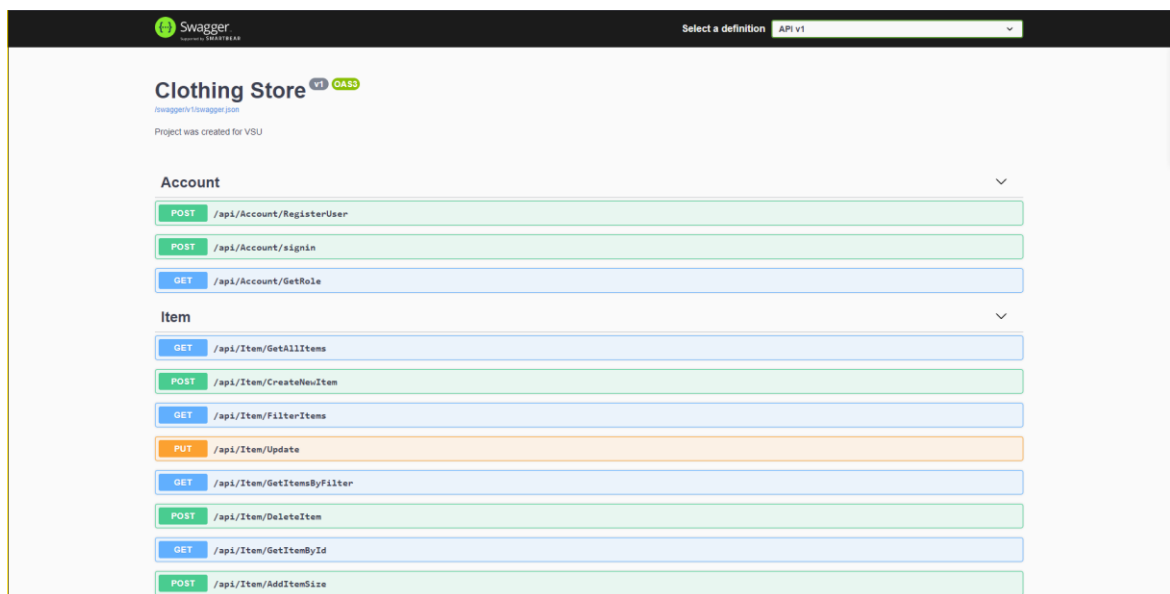


Рисунок 14. Пример документирования с использованием Swagger.

4. Интерфейс

4.1 Интерфейс неавторизованного пользователя

При открытии приложения пользователь попадает на главную страницу приложения (Рисунок 15). На ней пользователь имеет возможность перейти на страницу авторизации или регистрации.

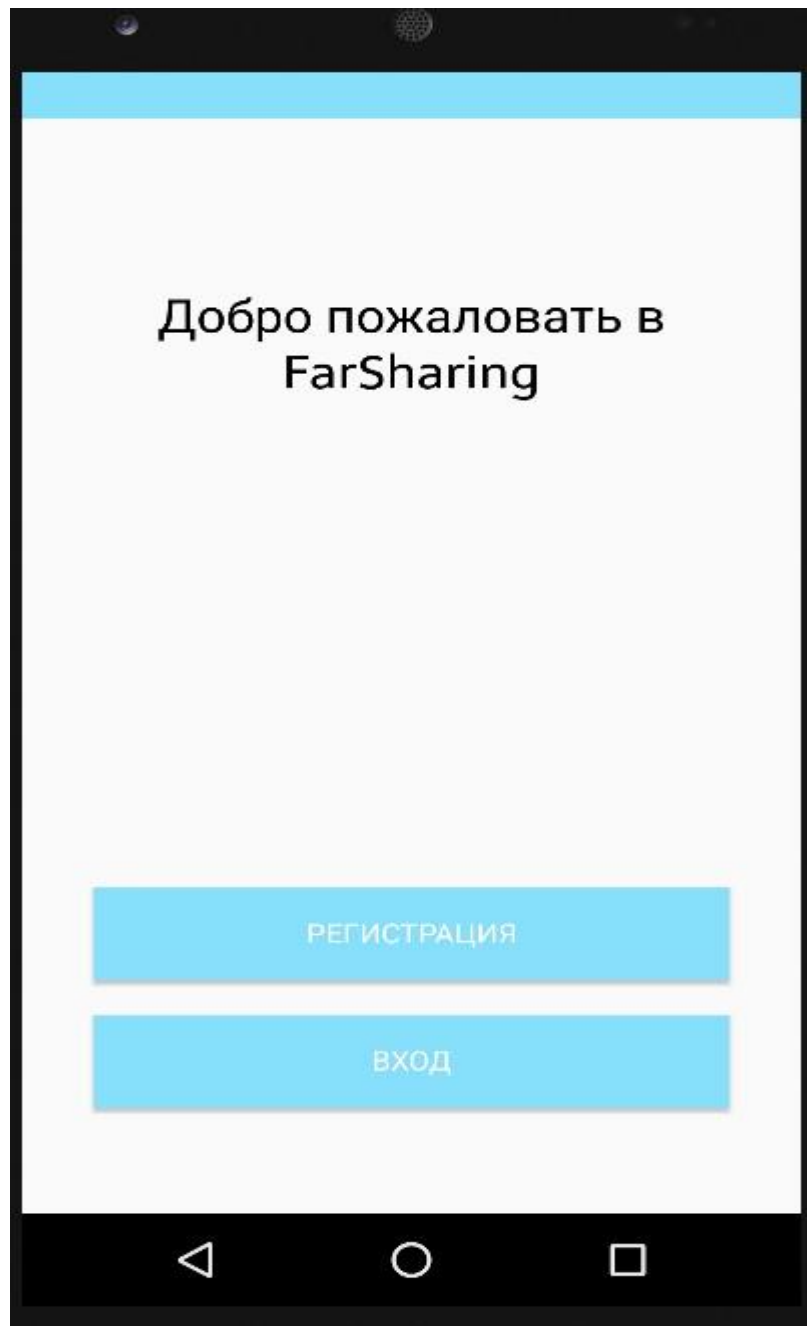
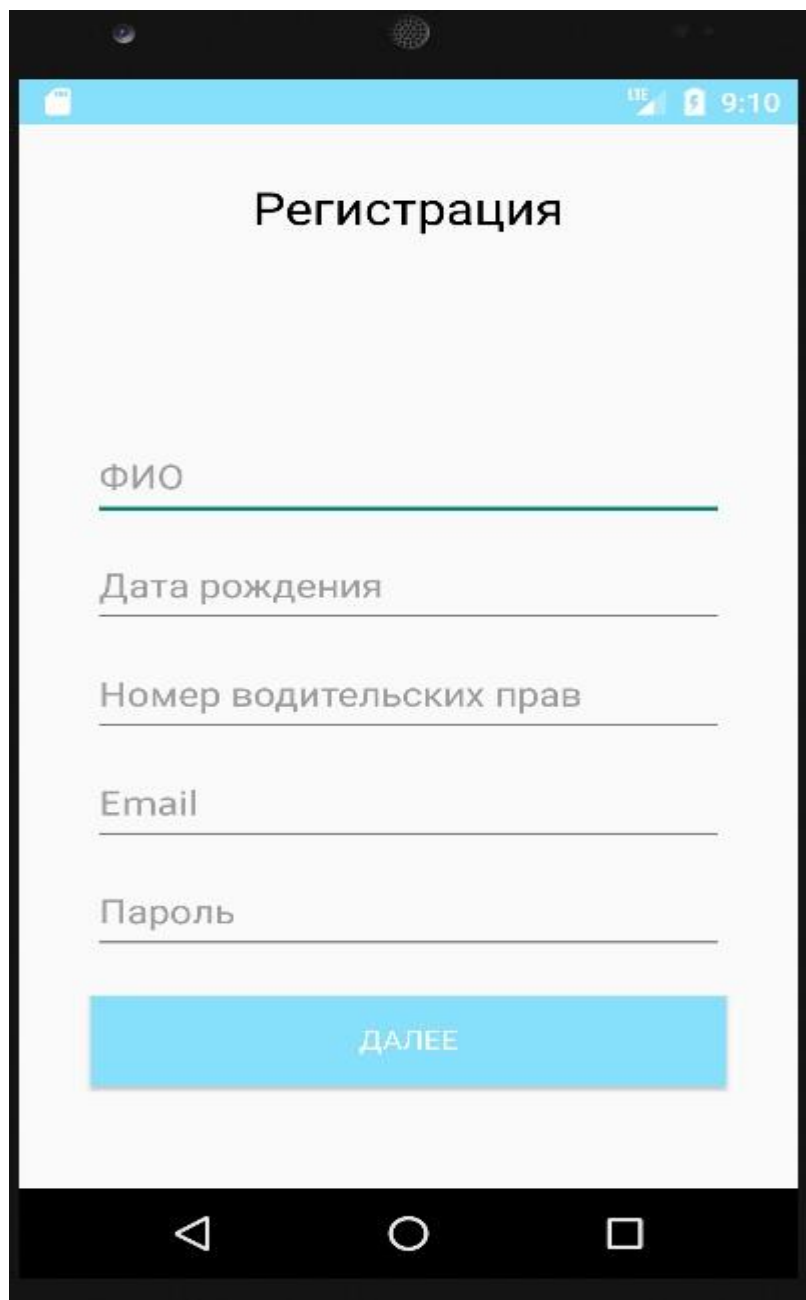


Рисунок 15. Стартовый экран.

Если по какой-то причине пользователь еще не зарегистрирован, то он имеет возможность нажать на кнопку «Регистрация» для перехода на страницу регистрации (Рисунок 16) и дальнейшей регистрации. Для того, чтобы зарегистрироваться пользователю необходимо заполнить поля «ФИО», «Дата рождения», «Номер водительских прав», «Email», «Пароль» и нажать кнопку «Далее».



Регистрация

ФИО

Дата рождения

Номер водительских прав

Email

Пароль

ДАЛЕЕ

Рисунок 16. Экран регистрации.

При нажатии кнопки «Далее» пользователь перейдёт на экран Подтверждения почты (Рисунок 17).



Рисунок 17. Экран подтверждения почты.

После подтверждения электронной почты пользователь перейдёт на экран авторизации (Рисунок 18).

The image shows a mobile application interface for login. At the top, there is a status bar with a light blue background, displaying a battery icon, signal strength, and the time 9:11. Below the status bar, the word "Вход" (Login) is centered in a large, black, sans-serif font. Underneath, there are two input fields: the first is labeled "Email" and the second is labeled "Пароль" (Password). Both labels are in a grey, sans-serif font. Below the input fields, there is a large, rectangular blue button with the word "ДАЛЕЕ" (Next) in white, sans-serif font. The entire content is set against a light grey background. The bottom of the screen shows a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

Рисунок 18. Экран авторизации.

4.2 Интерфейс авторизованного пользователя

Если пользователь авторизовался, то он попадает на главную страницу (Рисунок 19).

Автомобили		
Mersedes AMG	A111AA	Свободен
Renault Logan	B222BB	Свободен
Land Rover Range Rover	X777XX	Свободен
Mercedes GL 500	K555KK	Свободен

Рисунок 19. Главный экран пользователя.

На этой странице авторизованный пользователь может перейти на экран с его личными данными, нажав кнопку «Меню». После нажатия на эту кнопку пользователь переходит на страницу личного кабинета (Рисунок 19). На этом экране клиент может просмотреть свои личные данные и при необходимости их отредактировать, нажав на кнопку «Редактировать».

Также на главной странице авторизованный пользователь может, ознакомиться со списком доступных автомобилей. Выбрав автомобиль, перейдём на следующий экран, на котором сможем ознакомиться с более точными параметрами автомобиля и арендовать его (Рисунок 20).

Выбранный Вами авто:

Бренд	Mercedes		
Модель	GL 500		
Тип кузова	Внедорожник		
Цвет	Черный		
Адрес парковки	г. Воронеж		
Цена в час, р.	800	Аренда на, ч.	5
Итого, р. 4000			

ПОДТВЕРДИТЬ

Рисунок 20. Экран – характеристик выбранного автомобиля.

4.3 Интерфейс пользователя, авторизованного под ролью «Администратор»

Если пользователь авторизовался под ролью «Администратор», то он попадает на главную страницу (Рисунок 21).

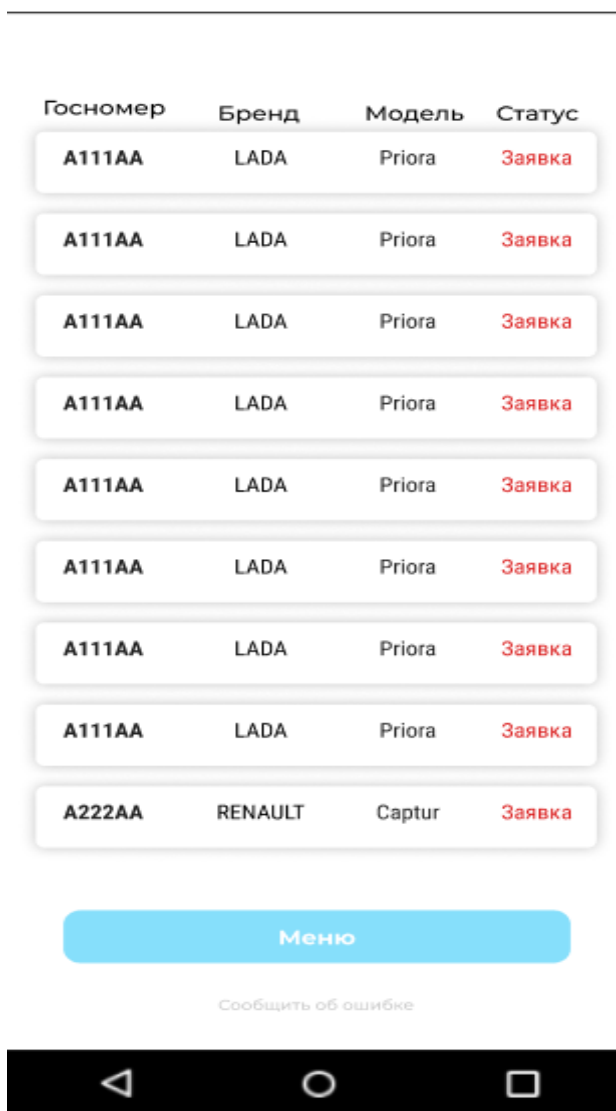


Рисунок 21. Главный экран администратора.

Администратор имеет возможность добавлять автомобили. Для этого ему необходимо перейти на страницу добавления автомобиля, нажав кнопку «Добавить автомобиль» в шапке страницы.

5. Тестирование

В целях проверки работоспособности системы в целом и подсистем приложения были проведены тесты по принципу «белого ящика» следующего вида:

- Модульное тестирование серверной части приложения;
- Анализ яндекс метрики.

5.1 Модульное тестирование серверной части приложения

Для данного тестирования, было выбрано четыре класса с основной бизнес – логикой:

- CarService;
- ClientService;
- ContractService;
- UserService.

Для каждого из этих классов было реализовано UNIT тестирование.

5.1.1 CarService

На Рисунке 22 представлено тестирование основных методов, реализованных в классе CarService.

CarServiceTest			
№	Действие	Количество тестов	Количество пройденных
1	Добавление машины	8	8
2	Получение данных о машине	3	3
3	Обновление данных машины	6	6

Рисунок 22. Тестирование CarService.

5.1.2 ClientService

На Рисунке 23 представлено тестирование основных методов, реализованных в классе ClientService.

ClientService			
№	Действие	Количество тестов	Количество пройденных
1	Регистрация клиента	5	5
2	Получение данных о клиенте	3	3
3	Удаление клиента	3	3
4	Обновление данных клиента	5	5
5	Смена статуса клиента	3	3
6	Получение всех клиентов	1	1

Рисунок 23. Тестирование ClientService.

5.1.3 ContractService

На Рисунке 24 представлено тестирование основных методов, реализованных в классе ContractService.

ContractService			
№	Действие	Количество тестов	Количество пройденных
1	Предоставление клиенту информации о машине	8	8
2	Получение платежных данных клиента	3	3
3	Оплата брони	5	5
4	Отмена брони	2	2
5	Добавление контракта	5	5
6	Получение информации о запросе на бронь клиента	4	4
7	Одобрение/отклонение брони	4	4

Рисунок 24. Тестирование ContractService.

5.1.4 UserService

На Рисунке 25 представлено тестирование основных методов, реализованных в классе UserService.

№	Действие	Количество тестов	Количество пройденных
1	Установка код активации	4	4
2	Активация аккаунт	7	7
3	Получение данных пользователя	3	3
4	Удаление пользователя	1	1
5	Обновление данных учетной записи пользователя	5	5
6	Авторизация	8	8

Рисунок 25. Тестирование UserService.

5.2 Яндекс метрики

На Рисунке 26 представлены яндекс метрики:

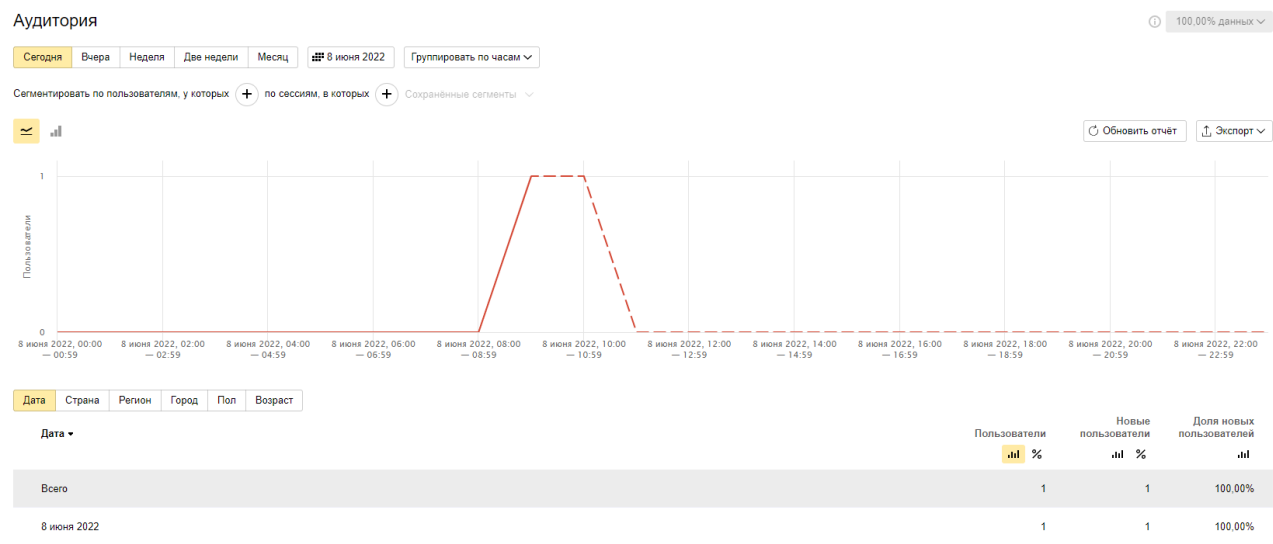


Рисунок 26. Яндекс метрики.

6. Заключение

В результате проделанной работы было реализовано приложение Farsharing, которое позволяет пользователю арендовать автомобиль. Мобильное приложение Farsharing обладает интерактивным интерфейсом, который необходим для взаимодействия пользователя с приложением и позволяет видеть и работать с блоками для удобства работы с сервисом.

Были выполнены следующие задачи:

1. Создано три уровня доступа:

- Неавторизованный пользователь;
- Авторизованный пользователь;
- Администратор.

2. Добавлены следующие возможности для авторизованного пользователя:

- Просмотр доступных автомобилей;
- Отправка заявки на бронь;
- Оплата брони;
- Отмена брони;
- Просмотр личных данных;
- Изменение личных данных;
- Просмотр информации о конкретном автомобиле.

3. Добавлены следующие возможности для администратора:

- Обработка брони;
- Добавление автомобиля;
- Просмотр данных пользователя;
- Блокировка пользователя;
- Разблокировка пользователя.

Приложение отвечает заявленным заказчиком техническим требованиям, таким как:

1. Авторизация пользователя посредством email и пароль;
2. Корректировка данных о арендованных автомобилях.

Так же приложение отвечает заявленным заказчиком требованиям к интерфейсу, таким как:

1. Обладает отдельным интерфейсом для пользователя и администратора;
2. Отображает списки данных, перечисленные в функциональных требованиях.

Список используемых источников

1. Spring Boot / [сайт]. – URL: https://www.tutorialspoint.com/spring_boot/spring_boot_rest_template.htm
2. Яндекс Драйв / [сайт]. – URL: <https://yandex.ru/drive>
3. Делимобиль / [сайт]. – URL: <https://delimobil.ru>
4. Swagger / [сайт]. – URL: <https://swagger.io>
5. Android sdk / [сайт]. – URL: <https://developer.android.com/studio>
6. Hibernate / [сайт]. – URL: <https://hibernate.org>
7. Unit тестирование / [сайт]. – URL: <https://habr.com/ru/post/526406/>
8. Retrofit / [сайт]. – URL: <http://developer.alexanderklimov.ru/android/library/retrofit.php>
9. Yandex.Metrica / [сайт]. – URL: <https://artvk.blogspot.com/2016/02/android-android-studio.html>
10. Однорукий интерфейс / [сайт]. – URL: <https://habr.com/ru/post/524374/>
11. Отправка писем по smtp джава / [сайт]. – URL: <https://javarush.ru/groups/posts/1226-kak-otpraviti-pismo-iz-java-prilozheniya-s-primerom>