



**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Институт информационных технологий**

**Кафедра инструментального и прикладного программного обеспечения**

**Дисциплина «Шаблоны программных платформ языка Джава»**

**КУРСОВАЯ РАБОТА**

**Веб-приложение «Каршеринг»**

Студент: Сидоров С.Д.

Группа: ИКБО-20-21

Руководитель: старший преподаватель Сеницын А.В.

Москва 2023

## **Цель**

Разработка серверной части веб-приложения «Каршеринг»

## **Задачи**

1. Провести анализ предметной области и формирование основных требований к приложению
2. Обосновать выбор средств ведения разработки
3. Разработать приложение с использованием выбранной технологии и инструментария
4. Провести тестирование приложения
5. Оформить пояснительную записку по курсовой работе
6. Провести анализ текста на антиплагиат
7. Создать презентацию по выполненной курсовой работе

# Технологии разработки

Java

Spring

HTML5

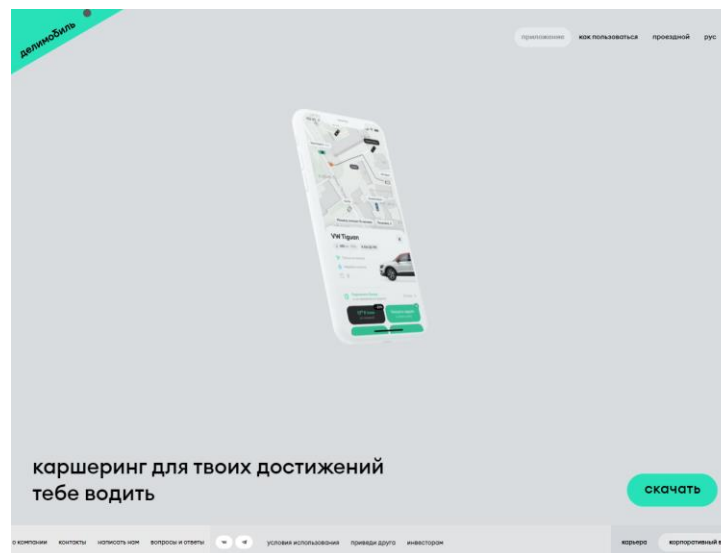
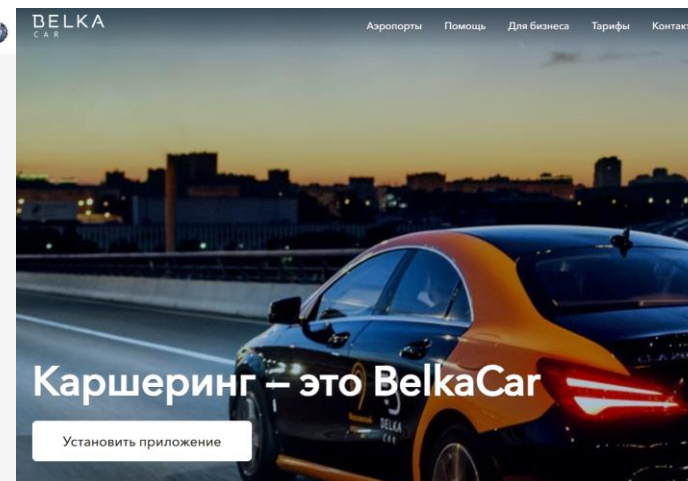
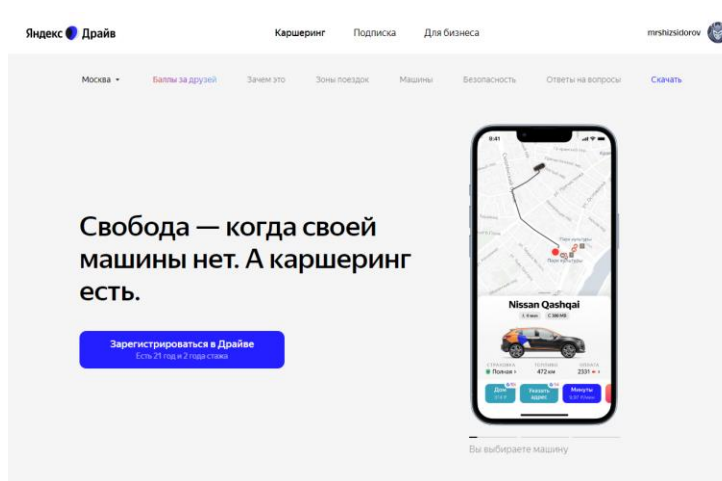
CSS3

PostgreSQL

HTTP

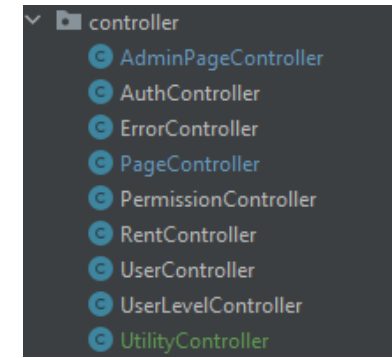
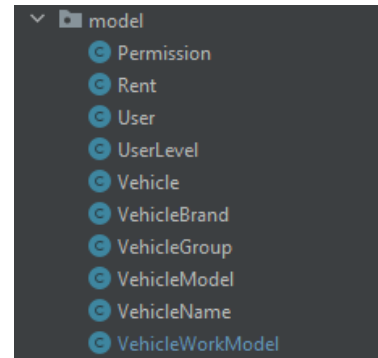
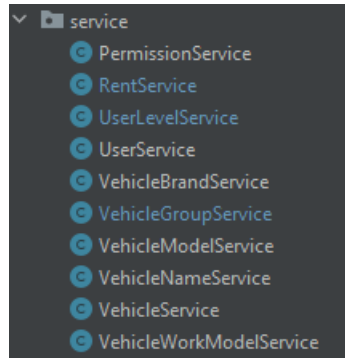
REST

# Анализ предметной области



# Создание приложения

Модели, контроллеры, представления



```
if (loginRequest.getPassword().equals(userDetails.getPassword())){
    LOGGER.info(msg: "User authenticated successfully: " + loginRequest);
    Map<String, String> claims = new HashMap<>();
    claims.put("username", loginRequest.getUsername());
    claims.put("password", loginRequest.getPassword());
    String authorities = userDetails.getAuthorities().stream()
        .map(Object::toString)
        .collect(Collectors.joining(" "));
    claims.put("authorities", authorities);
    claims.put("userId", "1");
    String jwt = jwtHelper.createJwtForClaims(loginRequest.getUsername());
    return new LoginResult(jwt);
}
```

```
@PostMapping("/start")
public ResponseEntity<ExecutionResult<Rent>> startRent(@RequestBody Rent rent){
    ExecutionResult<Rent> result = rentService.startRent(rent);
    if (result.getErrorMessage() != null) {
        return ResponseEntity.badRequest().body(result);
    }

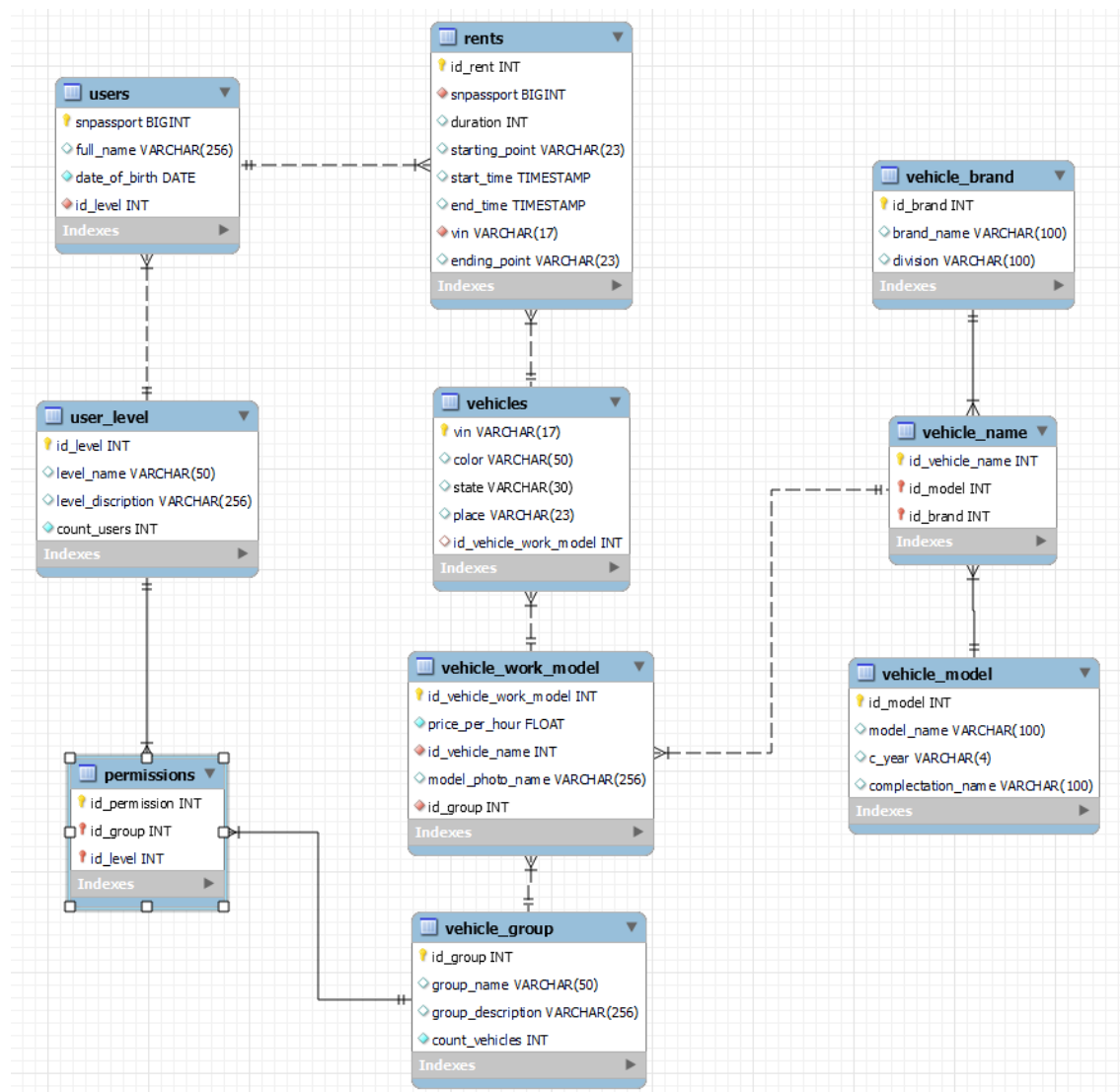
    return ResponseEntity.ok(result);
}

MShizikU
@PostMapping("/close/{id}")
public ResponseEntity<ExecutionResult<Rent>> startRent(@PathVariable Integer id, @RequestBody
    ExecutionResult<Rent> result = rentService.closeRent(id, rent);
    if (result.getErrorMessage() != null) {
        return ResponseEntity.badRequest().body(result);
    }

    return ResponseEntity.ok(result);
}
```

# База данных

## Структура базы данных



# Тестирование

Регистрация

Регистрация

Вход

Серия номер паспорта:

1718598304

ФИО пользователя:

Сидоров Станислав Дмитриевич

Дата рождения:

2004-06-03

Имя пользователя:

admin

Пароль:

...

Register

# Тестирование

Авторизация

**Вход**

Регистрация

**Username:**

admin

**Пароль:**

...

Login



# Тестирование

Основная страница



Basic carsharing

Поиск...

admin



Toyota Mark II

20.0

АРЕНДА

# Тестирование

Доступ к странице администрирования

Admin Panel

Star

Hello, there ▾

Rents

Users

Vehicles

Work models

Names

Permission

Table of brands

ID brand	Brand name	Division
1	Toyota	Russia
2	Mercedes	Russia
3	Honda	Russia

Table of models

ID model	Model name	Construction year
1	Mark II	2000
2	e200	2001
3	CR-V	2004

Table of names

ID name	Brand id	Model id
1	1	1
2	2	2
3	3	3

Add new brand

Brand name:

Add new model

Model name:

# Результаты

- Разработана база данных
- Разработана серверная часть приложения
- Разработана клиентская часть приложения
- Написано 1500 строк кода серверной части и 800 строк кода клиентской части приложения
- Обеспечена отправка запросов клиентской части на сервер, их обработка и отправка ответа с серверной части.
- Обеспечена связь сервера с базой данных

URL репозитория на GitHub:

<https://github.com/MShizikU/BasicCarSharing>



The screenshot shows the GitHub repository page for MShizikU/BasicCarSharing. The repository is in the 'initial' state, with a commit hash of dd1be7e made 2 minutes ago. There are 2 commits in total. The table below lists the files and folders in the repository, all of which were created in the initial commit.

File/Folder	Commit	Time
auth	initial	2 minutes ago
config	initial	2 minutes ago
main	initial	2 minutes ago
.gitignore	initial	2 minutes ago
docker-compose.yml	initial	2 minutes ago

**СПАСИБО ЗА ВНИМАНИЕ!**