UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

CarRental

Aplicație Web pentru Închirieri Auto

Documentație Tehnică

Jalbă Simion, Ilco Doina-Cezara

20 ianuarie 2026

Cuprins

1 Descriere Generală a Proiectului

1.1 Ce este CarRental?

CarRental este o aplicație web full-stack pentru gestionarea închirierilor de mașini. Platforma permite utilizatorilor să vizualizeze mașinile disponibile, să efectueze rezervări și să își gestioneze contul, în timp ce administratorii pot gestiona flota de vehicule și rezervările clienților.

1.2 Obiective Propuse

La începutul proiectului, ne-am propus să implementăm următoarele funcționalități:

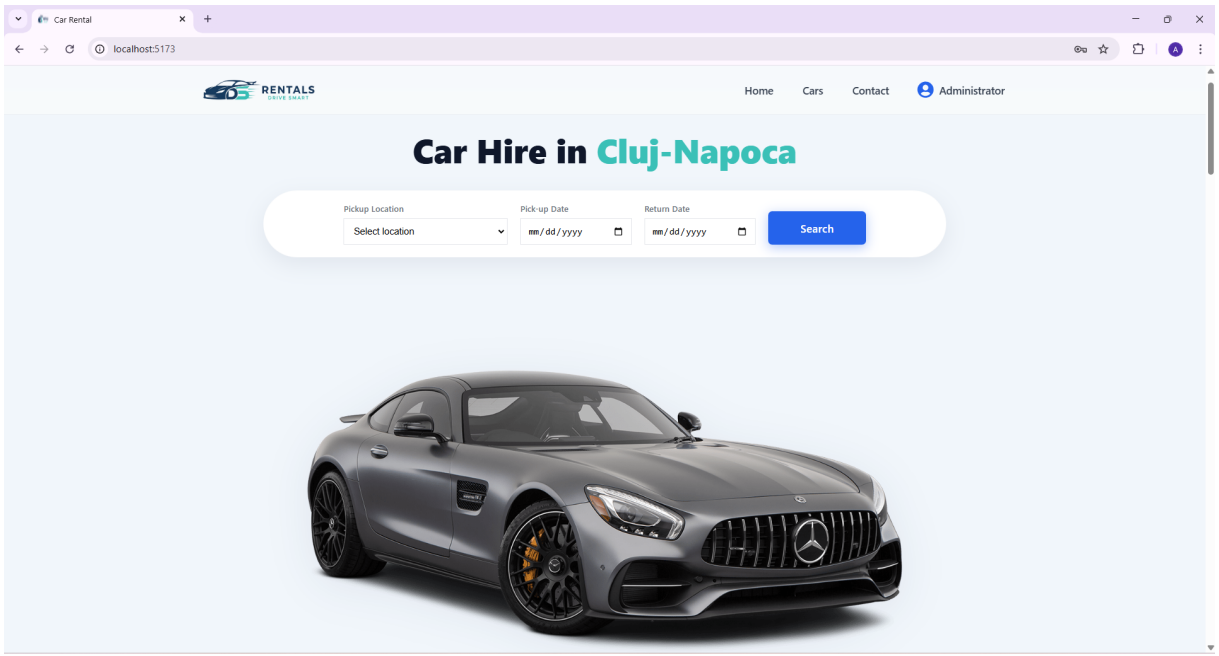
- **Pagina Principală:** Calendar pentru selectarea datelor, imagine de fundal atractivă, listă cu toate mașinile disponibile, secțiuni FAQ și Contact.
- **Pagina Booking Form:** Formular complet pentru rezervare cu pop-up de confirmare și notificare prin email.
- **Pagina Available Cars:** Listă de mașini cu carduri individuale pentru fiecare vehicul
- **Pagina Admin:** Funcționalitate pentru adăugarea și gestionarea mașinilor
- **Pagina User:** Istoric rezervări și date personale
- **Elemente comune:** Logo, FAQ, Contact pe toate paginile
- **Detalii Mașină:** An fabricație, tip combustibil, cutie de viteze, număr locuri

1.3 Ce s-a Realizat

Toate obiectivele propuse au fost implementate cu succes, mai puțin confirmarea prin email, plus funcționalități suplimentare:

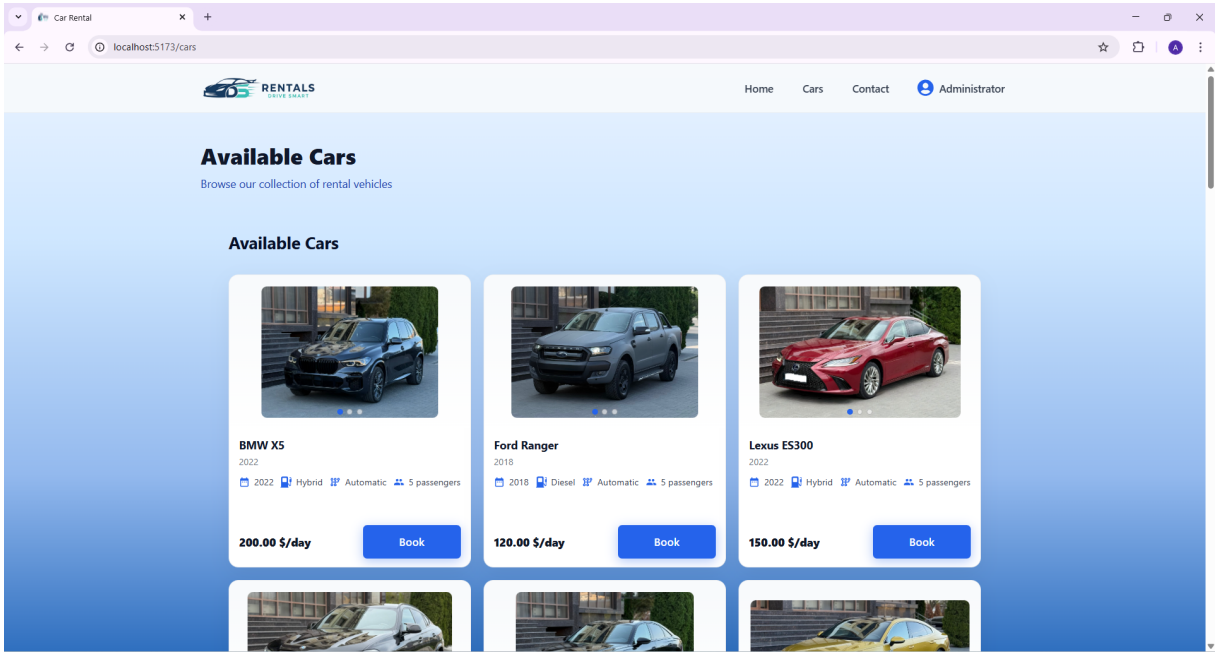
1.3.1 Funcționalități Implementate - Frontend

1. **Pagina Home (Principală)**
   - Hero section cu imagine de fundal
   - Formular de căutare cu selectare locație (Aeroportul Avram Iancu, Autogara Beta)
   - Lista completă a mașinilor disponibile



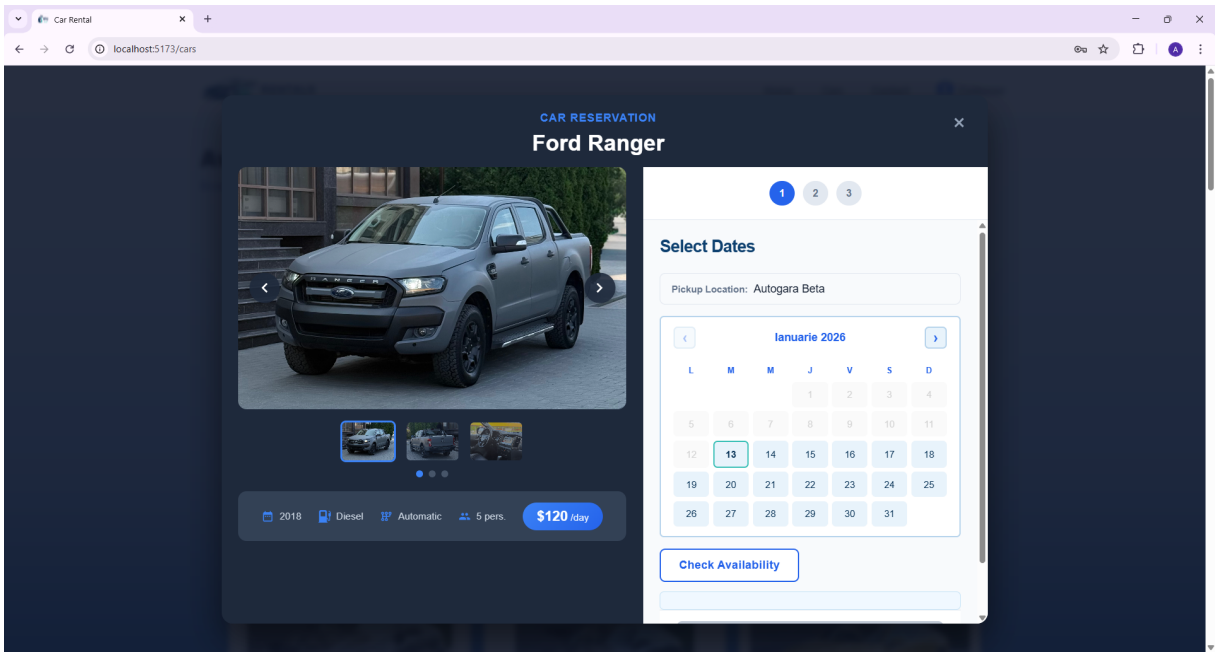
2. **Pagina Available Cars**

   - Afișare mașini filtrate după criterii de căutare
   - Card-uri pentru fiecare mașină cu galerie de imagini



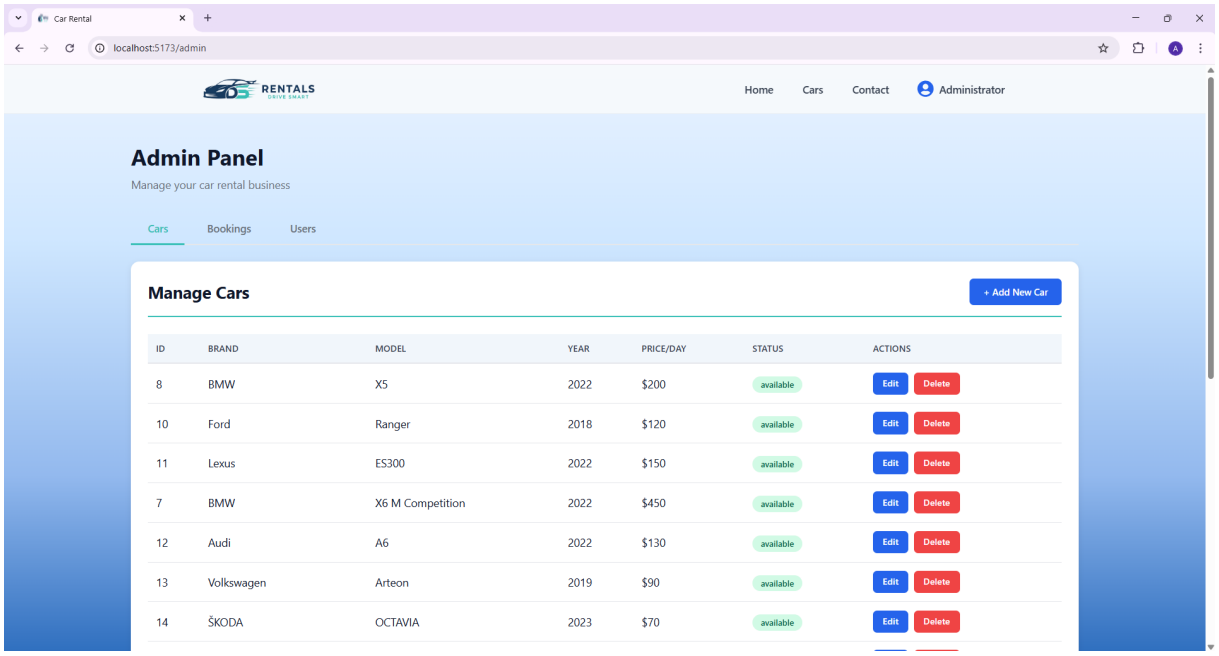
3. **Sistem de Booking**

   - Formular în pași multipli (Locație/Dată → Sumar → Date Client)
   - Verificare disponibilitate în timp real
   - Upload permis de conducere (față și verso)



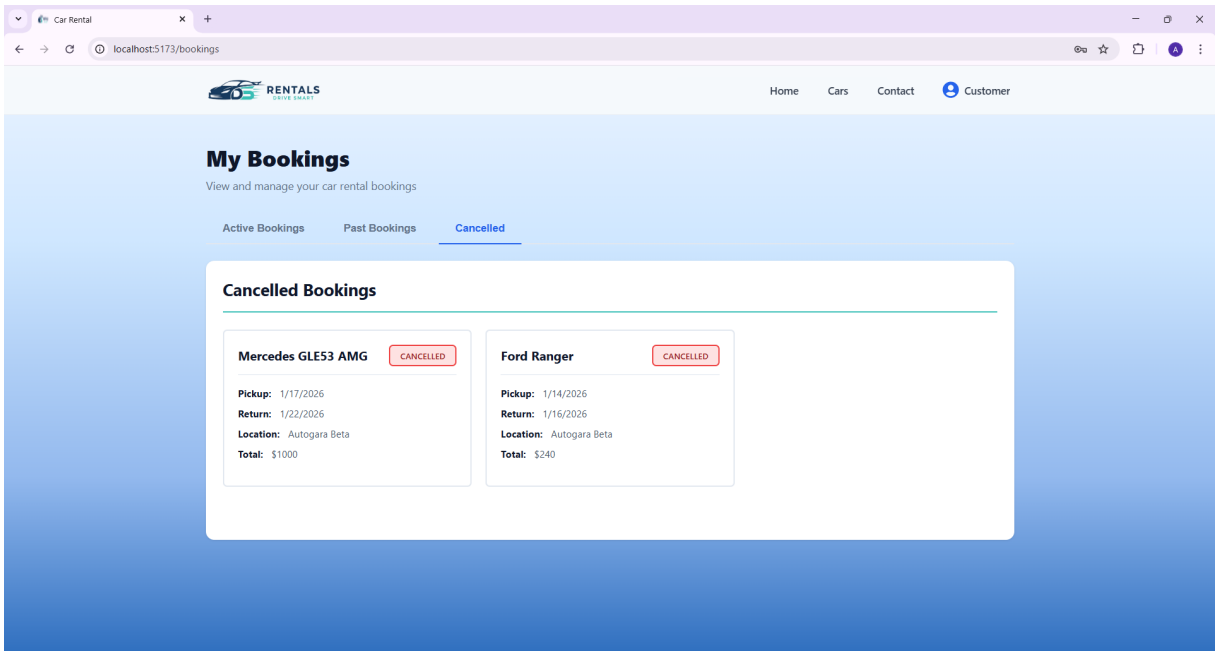
4. **Pagina Admin**

- CRUD complet pentru mașini (Create, Read, Update, Delete)
- Gestionare utilizatori
- Aprobare/Respingere rezervări cu workflow de status
- Badge-uri pentru rezervări în așteptare



5. **Pagina My Bookings (User)**

- Tab-uri: Active, Past, Cancelled
- Vizualizare istoric complet
- Posibilitate de anulare rezervări



6. **Autentificare și Autorizare**
   - Pagină Login
   - Pagină Register
   - Pagină Profile cu editare date personale
   - Protecție rute (ProtectedRoute)
   - Diferențiere rol customer/admin
7. **Navbar comun**
   - Logo
   - Navigare adaptată rolului utilizatorului
   - Link-uri către Contact

1.3.2 Funcționalități Implementate - Backend

1. **API REST complet**
   - AuthController - autentificare și înregistrare
   - BookingController - gestionare rezervări
   - CarController - gestionare mașini
   - AdminController - operațiuni administrative
   - UserController - gestionare utilizatori
2. **Securitate**
   - JWT (JSON Web Token) pentru autentificare
   - Spring Security pentru autorizare
   - Rate limiting pentru protecție împotriva atacurilor

- CORS configurat pentru frontend

3. **Modele de Date**
   - User - utilizatori cu roluri (customer/admin)
   - Car - mașini cu toate specificațiile
   - Booking - rezervări cu status și detalii client
   - CarImage - imagini multiple pentru mașini
4. **Servicii Business Logic**
   - CarService - logică pentru mașini
   - UserService - logică pentru utilizatori
   - BookingCleanupService - anulare automată rezervări pending după 24h
5. **Upload Fișiere**
   - Imagini mașini
   - Permise de conducere

2 Tehnologii Utilizate

2.1 Model de Date

2.1.1 Entitatea Car

```
Car {
    Long id
    String brand          // Marca (ex: BMW, Audi)
    String model          // Model (ex: X5, A4)
    Integer year          // An fabricatie
    Double pricePerDay    // Pret pe zi
    String fuelType       // Tip combustibil
    String gearbox        // Cutie viteze
    Integer passengers    // Numar locuri
    String status         // Stare (available/rented)
    String location       // Locatie
    String description    // Descriere
    List<CarImage> images // Imagini
}
```

2.1.2 Entitatea Booking

```
Booking {
    Long id
    User user             // Utilizator
    Car car               // Masina rezervata
    String pickupLocation // Locatie preluare
```

```
    LocalDate pickupDate  // Data preluare
    LocalDate returnDate  // Data returnare
    BigDecimal totalPrice // Pret total
    String status         // Status (pending/active/completed/
       cancelled)
    String customerName   // Nume client
    String customerEmail  // Email client
    String customerPhone  // Telefon client
    String idnp           // IDNP
    String driverLicenseFrontUrl  // Permis fata
    String driverLicenseBackUrl   // Permis verso
    LocalDateTime createdAt       // Data creare
}
```

2.1.3 Entitatea User

```
User {
    Long id
    String name           // Nume complet
    String email          // Email (unic)
    String passwordHash   // Parola (hash)
    String phone          // Telefon
    String role           // Rol (customer/admin)
}
```

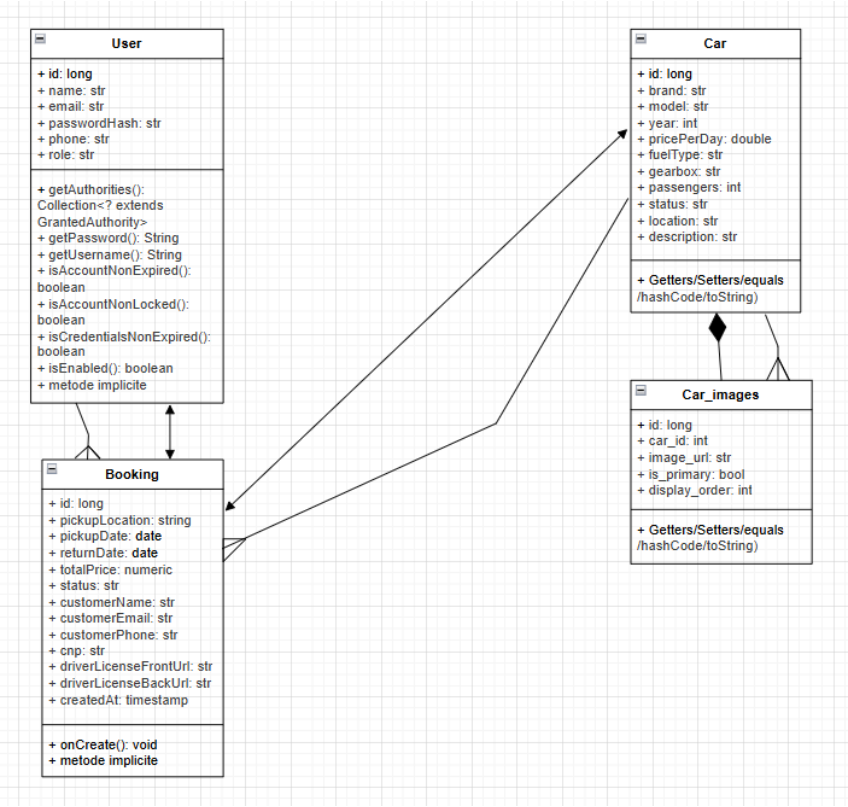


Figura 1: Diagrama de clase

3 Împărțirea pe Task-uri

3.1 Simion

1. **Backend**
   - Implementare schelet backend Spring Boot
   - Configurare securitate cu JWT
   - API-uri REST pentru toate entitățile
   - Cleanup service pentru rezervări pending
2. **Autentificare și Autorizare**
   - Backend pentru autentificare (login/register)
   - Frontend pentru autentificare
   - Navigare diferențiată customer/admin
   - Protected routes
3. **Pagina Admin**
   - CRUD mașini (adăugare, editare, ștergere)
   - Gestionare utilizatori
   - Workflow aprobare rezervări cu status
   - Badge-uri pentru rezervări pending
4. **Sistem Booking**
   - Implementare workflow pending booking
   - Blocare disponibilitate pentru rezervări pending
   - Auto-anulare rezervări pending după 24h
5. **Funcționalități Search**
   - Pre-selectare locație mașină
6. **UI/UX Improvements**
   - Traducere interfață din română în engleză
   - Aliniere labels și inputs în formularul de booking
   - Fix-uri CSS pentru butoane edit/delete/cancel
   - Clean code și refactorizare

3.2 Doina

1. **Aspect și Design**
   - Modificare aspect general al aplicației
   - Dezvoltare pagină home
   - Implementare secțiune de contact
   - Implementare bară de navigare
   - Design și integrare logo
   - Adăugare și gestionare imagini
   - Schimbări de stil și design
2. **Sistem de Rezervare**
   - Implementare formular de rezervare - frontend (aspect, layout, validări)
   - Implementare formular de rezervare - backend (procesare date, integrare bază de date)
3. **Funcționalități Search**
   - Backend pentru butonul de căutare
   - Locații corecte în dropdown-ul de căutare
4. **UI/UX Improvements**
   - Fix-uri CSS pentru butoane edit/delete/cancel

4 Fluxuri Principale ale Aplicației

4.1 Flux Rezervare Mașină

1. Utilizatorul accesează pagina principală
2. Selectează locația, data de preluare și returnare
3. Apasă butonul "Search" - este redirecționat la Available Cars
4. Vizualizează mașinile disponibile în perioada selectată
5. Apasă "Book Now" pe mașina dorită
6. Se deschide modal-ul de rezervare:
   - Pasul 1: Confirmare locație și date
   - Pasul 2: Sumar rezervare cu preț total
   - Pasul 3: Completare date personale și upload permis
7. Rezervarea este creată cu status "pending"
8. Administratorul aprobă/respinge rezervarea
9. Dacă nu este aprobată în 24h, se anulează automat

4.2 Flux Administrare

1. Administratorul se autentifică
2. Accesează pagina Admin
3. Poate gestiona:
    - **Cars:** Adaugă, editează, șterge mașini cu imagini
    - **Users:** Vizualizează și gestionează utilizatori
    - **Bookings:** Aprobă, respinge sau anulează rezervări

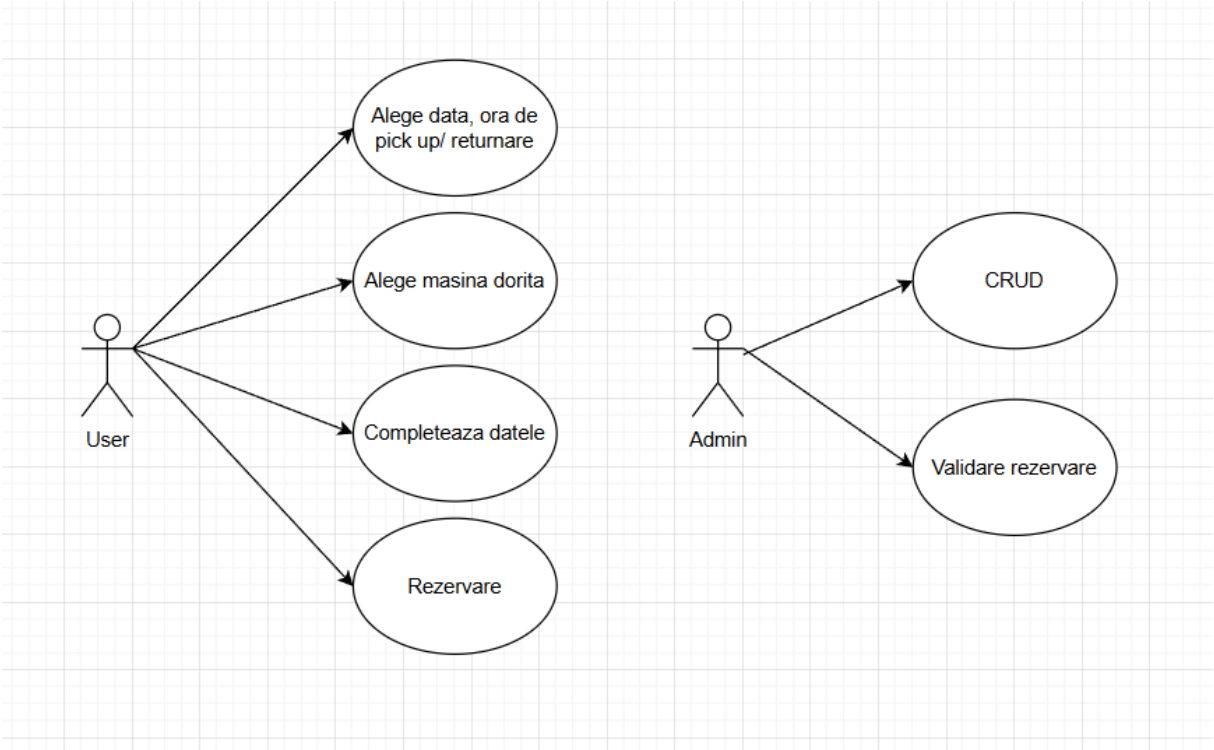


Figura 2: Diagrama de clase

5 Endpoint-uri API

5.1 Autentificare

Metodă	Endpoint	Descriere
POST	/api/auth/login	Autentificare utilizator
POST	/api/auth/register	Înregistrare utilizator nou

5.2 Mașini

Metodă	Endpoint	Descriere
GET	/api/cars	Lista tuturor mașinilor
GET	/api/cars/{id}	Detalii mașină
GET	/api/cars/available	Mașini disponibile

5.3 Rezervări

Metodă	Endpoint	Descriere
POST	/api/bookings	Creare rezervare
GET	/api/bookings/my	Rezervările utilizatorului
GET	/api/bookings/check-availability	Verificare disponibilitate
POST	/api/bookings/upload-license	Upload permis conducere
DELETE	/api/bookings/{id}	Anulare rezervare

5.4 Administrare

Metodă	Endpoint	Descriere
GET	/api/admin/cars	Lista mașini (admin)
POST	/api/admin/cars	Adaugă mașină
PUT	/api/admin/cars/{id}	Editează mașină
DELETE	/api/admin/cars/{id}	Șterge mașină
GET	/api/admin/users	Lista utilizatori
GET	/api/admin/bookings	Lista rezervări
PUT	/api/admin/bookings/{id}/approve	Aprobă rezervare
PUT	/api/admin/bookings/{id}/status	Schimbă status

6 Instrucțiuni de Rulare

6.1 Cerințe Preliminare

- Java 21 sau versiune ulterioară
- Node.js 18+ și npm
- PostgreSQL instalat și configurat
- Maven (sau se poate folosi wrapper-ul inclus)

6.2 Configurare Bază de Date

1. Creați o bază de date PostgreSQL pentru proiect
2. Configurați conexiunea în `application.properties`
3. Rulați scriptul `db_data.sql` pentru date inițiale

6.3 Pornire Backend

```
cd CarRental-backend
./mvnw spring-boot:run
```

Backend-ul va porni pe `http://localhost:8080`

6.4 Pornire Frontend

```
cd CarRental-frontend
npm install
npm run dev
```

Frontend-ul va porni pe `http://localhost:5173`

7 Concluzii

Proiectul **CarRental** a fost finalizat cu succes, îndeplinind obiective propuse inițial și adăugând funcționalități suplimentare pentru o experiență completă de utilizare.

7.1 Puncte Forte

- Arhitectură modernă cu React 19 și Spring Boot 3.5
- Securitate robustă cu JWT și Spring Security
- Interfață utilizator intuitivă și responsive
- Workflow complet pentru rezervări cu aprobare
- Sistem automat de cleanup pentru rezervări abandonate

7.2 Posibile Îmbunătățiri Viitoare

- Notificări prin email pentru confirmare rezervări
- Sistem de plată online integrat
- Aplicație mobilă nativă
- Dashboard cu statistici pentru administratori
- Sistem de review-uri și rating pentru mașini