

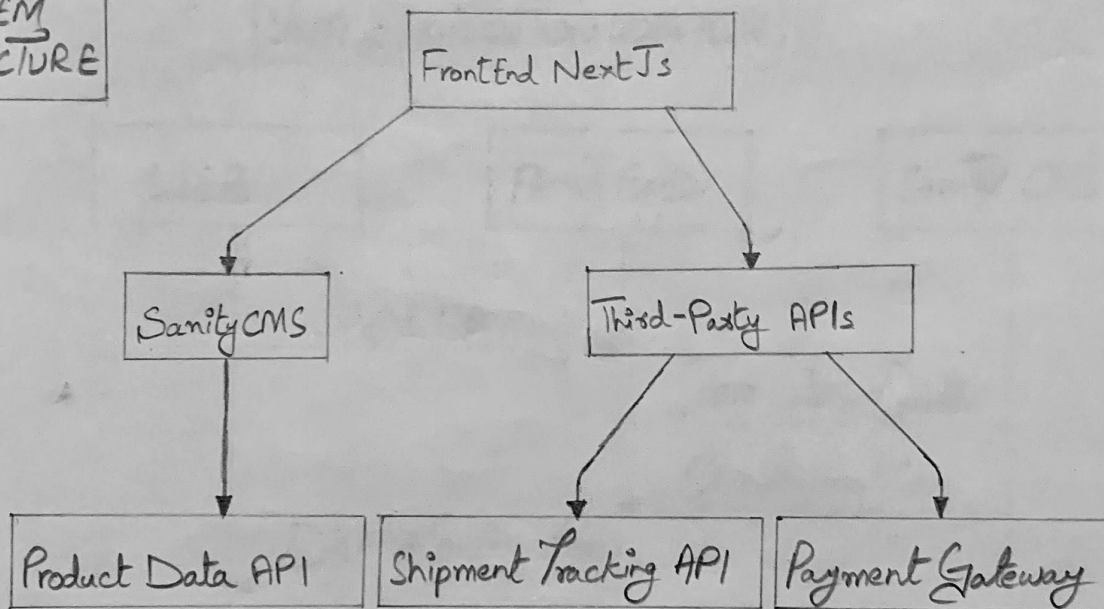
HACKTHON DAY:2

PLANNING THE TECHNICAL FOUNDATION

OVERVIEW:

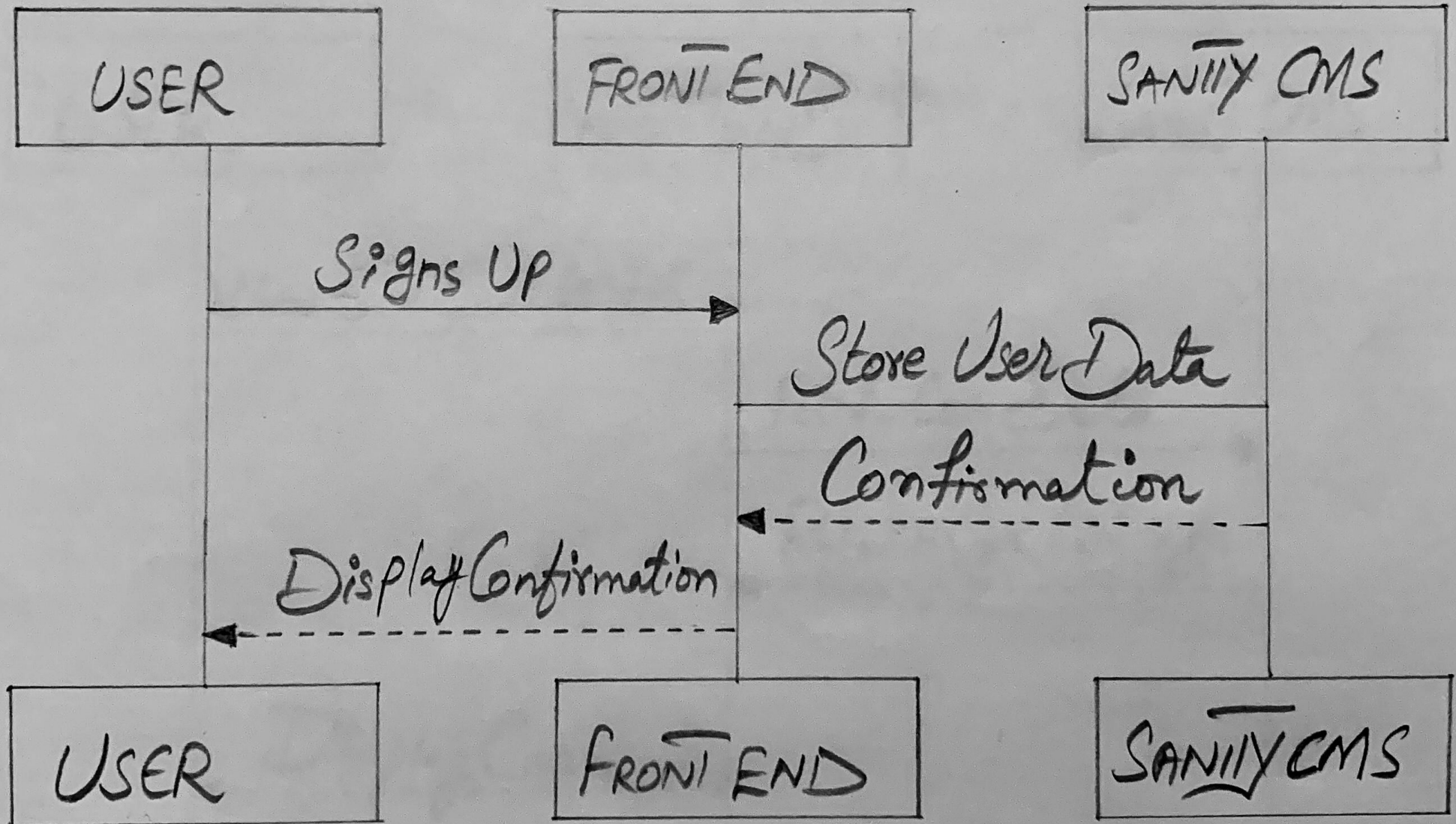
This document outlines the Technical Plans for developing an RENTAL E-commerce Marketplace to Make Businesses & Individuals Empower to Rent their Products On More efficient and in simple way.

HIGH-LEVEL SYSTEM ARCHITECTURE

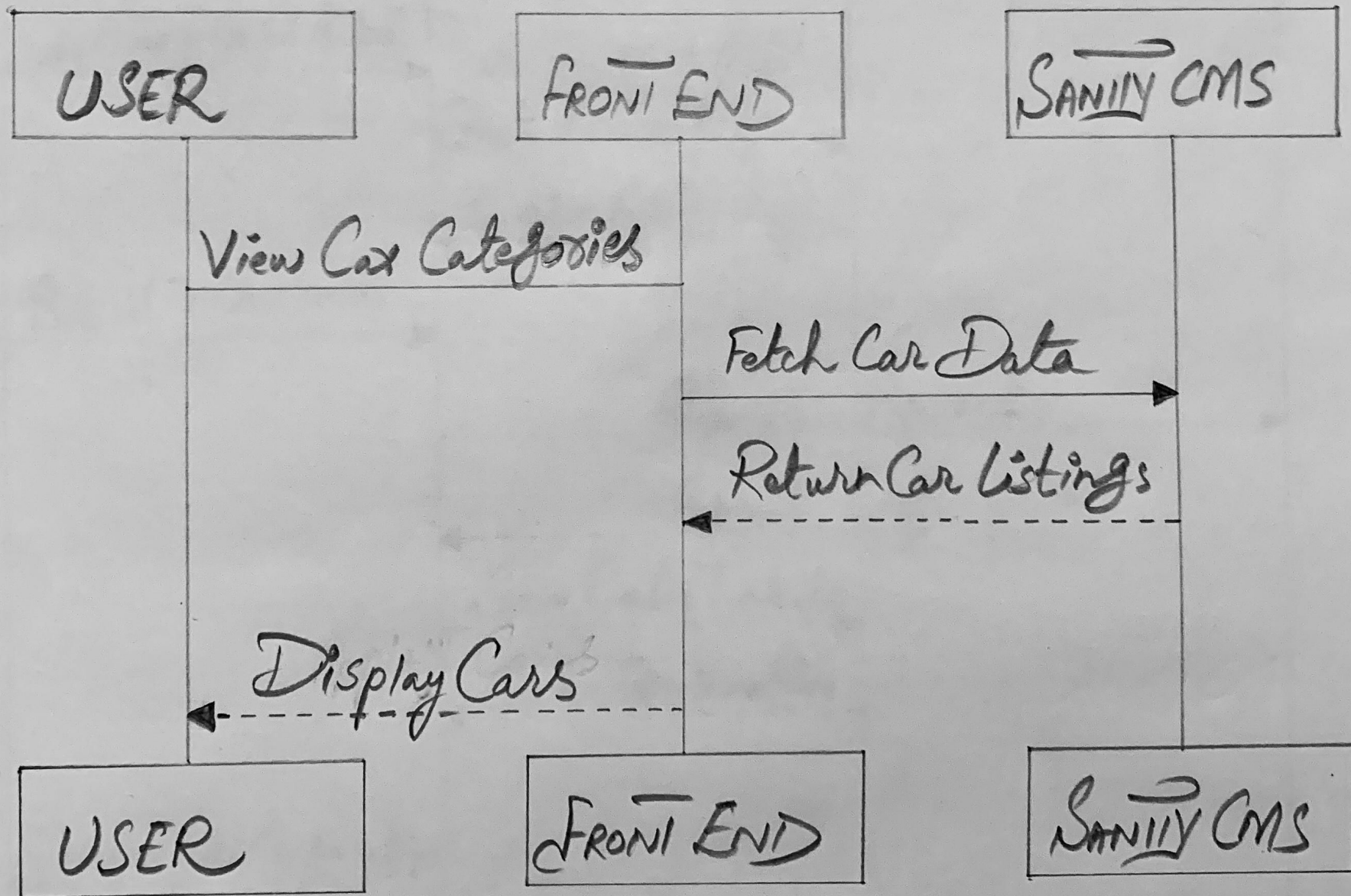


FRONTEND REQUIREMENTS	SANITY CMS AS BACKEND	THIRD PARTY APIs
User Friendly interface for Browsing Products.	Use Sanity CMS to manage Product data, customer details, and order records. Sanity acts as the database for marketplace.	Integrate APIs for shipment tracking, payment gateways, and other required backend services.
Responsive design for mobile & desktop Users.	Focus on designing Schemas in Sanity to align with the business goals from Day 1.	Ensure APIs provide the necessary data for frontend functionality.
Essential Pages:		
1 Home	4 Cart	
2 Product Listing	5 Checkout	
3 Product Details	6 Order Confirmation	

USER REGISTRATION WORKFLOW



CAR BROWSING WORKFLOW



Endpoint: /cars

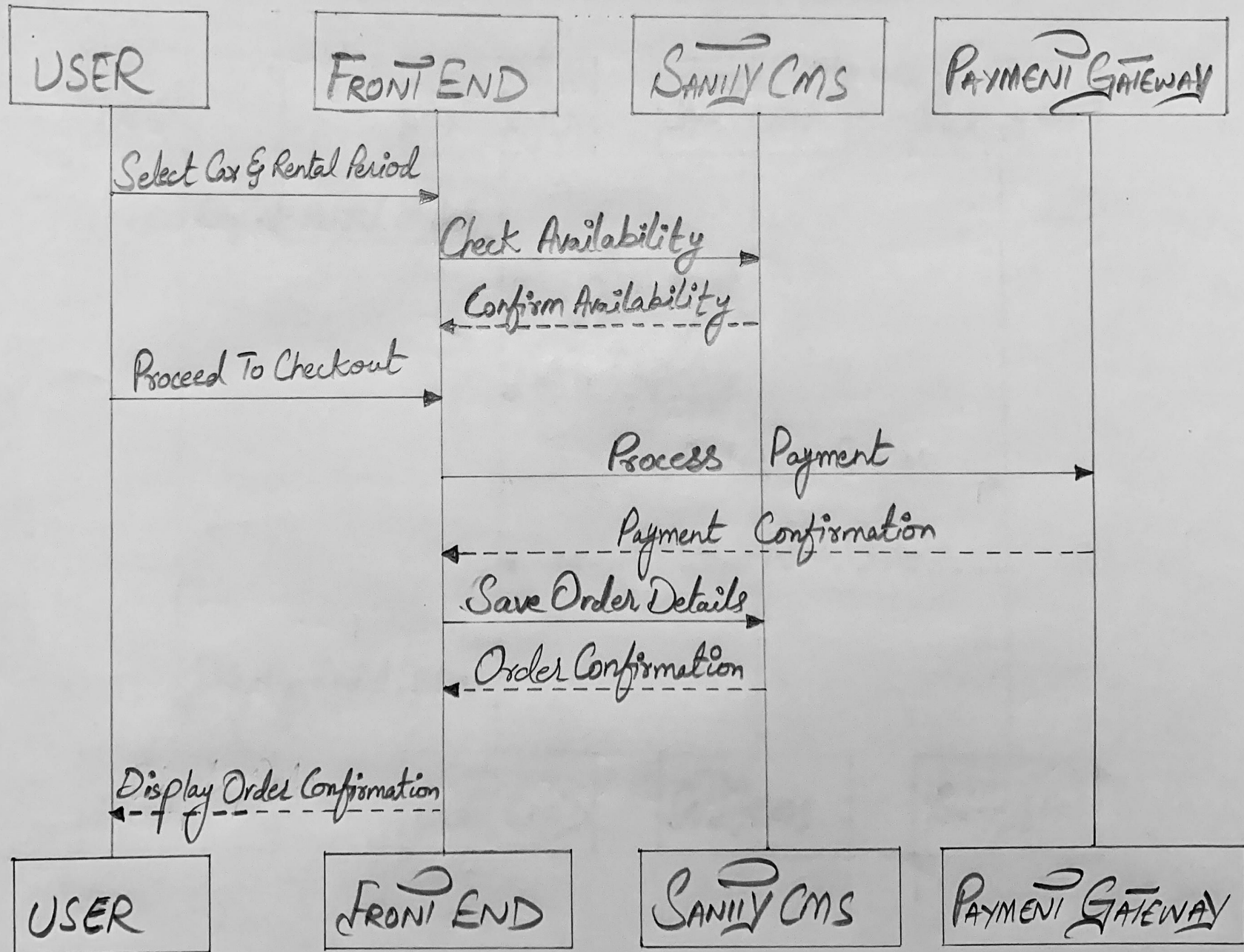
Method: GET

Description: Fetch all available cars

Response Example:

```
{
  "cars": [
    {
      "id": 1,
      "make": "Toyota",
      "model": "Camry",
      "year": 2023,
      "dailyRate": 50,
      "availability": true
    },
    // More cars...
  ]
}
```


RENTAL ORDER PLACEMENT WORKFLOW



Endpoint: /rentals

Method: POST

Description: Create a new rental order

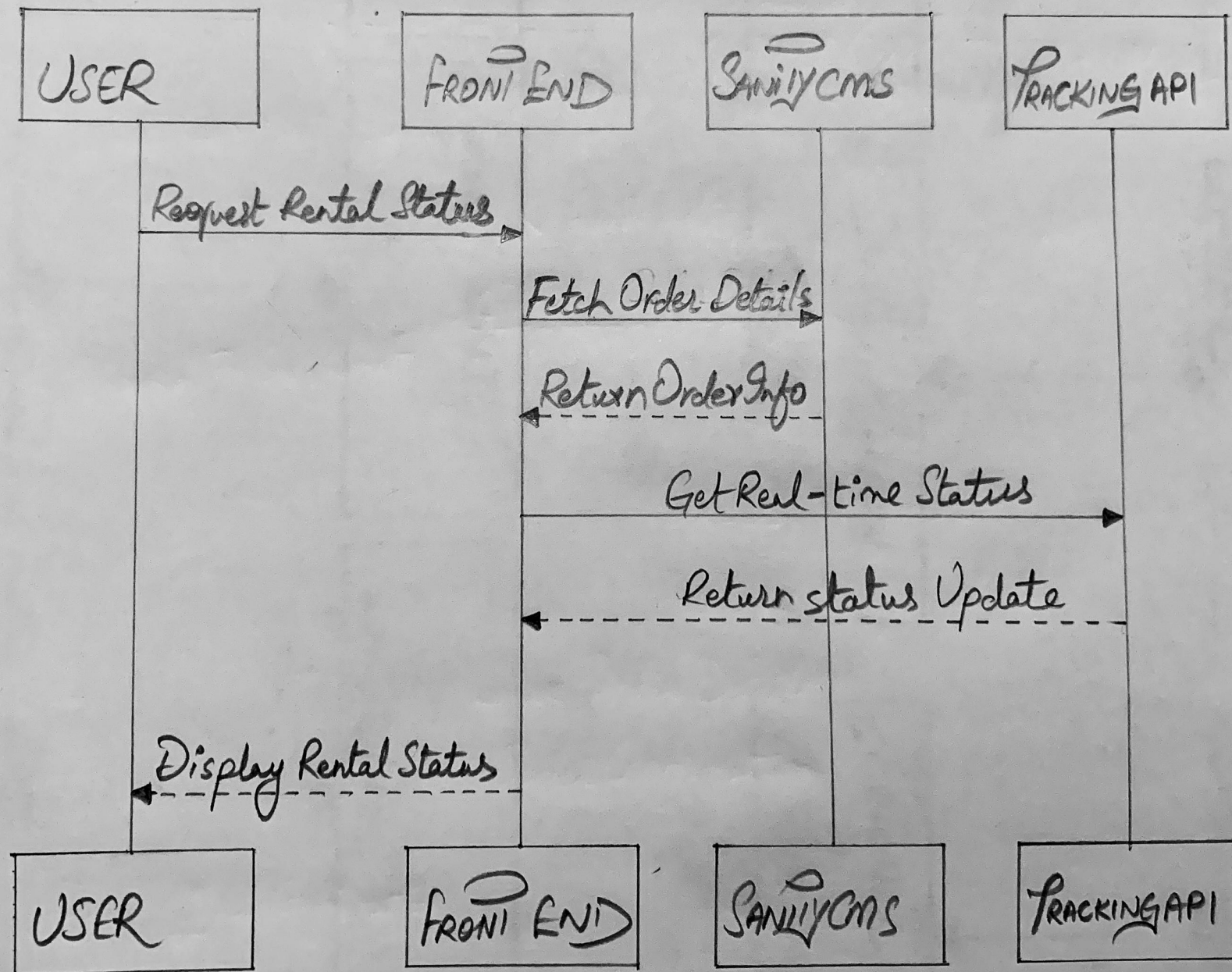
Payload:

```
{  
  "carId": 1,  
  "userId": 123,  
  "startDate": "2025-01-20",  
  "endDate": "2025-01-25",  
  "totalCost": 250  
}
```

Response Example:

```
{  
  "orderId": 789,  
  "status": "Confirmed",  
  "rentalDetails": {  
    "carId": 1,  
    "startDate": "2025-01-20",  
    "endDate": "2025-01-25",  
    "totalCost": 250  
  }  
}
```

RENTAL STATUS TRACKING WORKFLOW



Endpoint: /rental-status/{orderId}

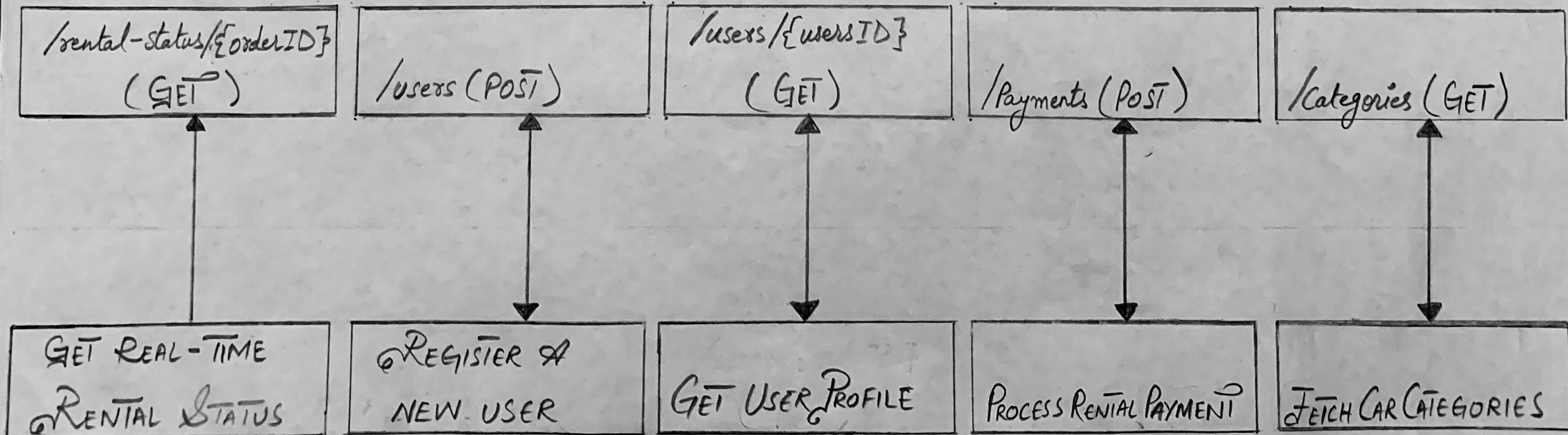
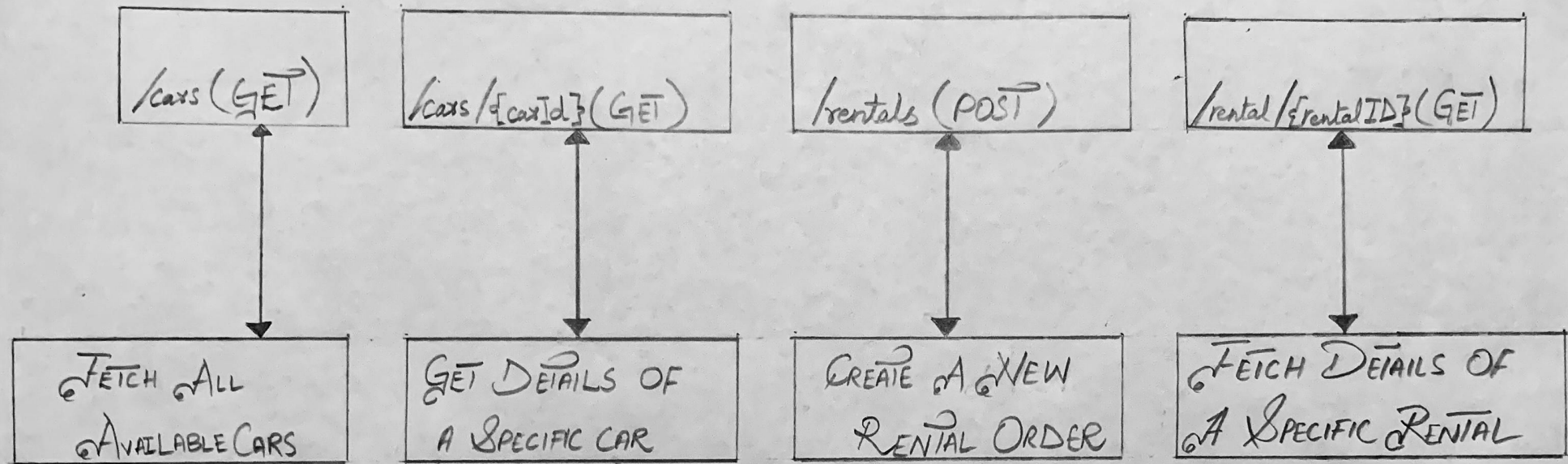
Method: GET

Description: Fetch real-time rental status

Response Example:

```
{  
  "orderId": 789,  
  "status": "Active",  
  "currentLocation": "Airport Parking Lot A",  
  "returnDate": "2025-01-25"  
}
```


API ENDPOINTS



`/cars (GET)` : Retrieves a list of all available cars for rent.

`/cars/{carId} (GET)` : Fetches detailed information about a specific car.

`/rentals (POST)` : Creates a new rental order when a user books a car.

`/rentals/{rentalId} (GET)` : Retrieves details of a specific rental order.

`/rental-status/{orderId} (GET)` : Provides real-time status updates for an active rental.

`/users (POST)` : Registers a new user in the system.

`/users/{userId} (GET)` : Fetches the profile information of a specific user.

`/payments (POST)` : Processes the payment for a rental order.

`/categories (GET)` : Retrieves the list of car categories (e.g., Economy, Luxury, SUV).

```
export default {
  name: 'car',
  type: 'document',
  fields: [
    { name: 'make', type: 'string', title: 'Make' },
    { name: 'model', type: 'string', title: 'Model' },
    { name: 'year', type: 'number', title: 'Year' },
    { name: 'dailyRate', type: 'number', title: 'Daily Rental Rate' },
    { name: 'availability', type: 'boolean', title: 'Is Available' },
    {
      name: 'image',
      type: 'image',
      title: 'Car Image',
      options: {
        hotspot: true
      }
    },
    {
      name: 'features',
      type: 'array',
      title: 'Features',
      of: [{ type: 'string' }]
    },
    {
      name: 'category',
      type: 'reference',
      to: [{ type: 'category' }],
      title: 'Category'
    }
  ]
}
```