

# Week 4 Exercise: Use Case-Driven API Design & Implementation

Objective: Learn HTTP methods, status codes, and API design best practices.  
Brainstorm a use case diagram to define required APIs for a ride-hailing system.

---

## Lab Overview

### Key Topics

1. **Use Case Diagramming:** Identify actors, use cases, and required APIs.
2. **API Specification:** Define RESTful endpoints, HTTP methods, and status codes.
3. **Implementation:** Build a new feature (e.g., user authentication) based on the design.

### Deliverables

1. Use case diagram and API specifications.
  2. Node.js/Express implementation of the new feature.
  3. Answers to design-focused questions.
- 

## Lab Procedures

### Lab Procedures

#### Part 1: Use Case Diagram & API Specification

##### Task 1: Brainstorm Actors and Use Cases

1. **Actors (Example):** Customer, Driver, Admin.
2. **Use Cases (Example):**
  - **Customer:** Register, Login, View Profile.
  - **Driver:** Update Availability, View Earnings.
  - **Admin:** Block User, View System Analytics.

##### Task 2: Design the Use Case Diagram

1. Use draw.io or Lucidchart to create a diagram.

### Task 3: Define API Specifications

1. Translate use cases into RESTful endpoints (Example)

Use Case	Endpoint	Method	Status Codes
Customer Registration	/users	POST	201 Created, 400 Bad Request
Customer Login	/auth/login	POST	200 OK, 401 Unauthorized
Update Driver Status	/drivers/{id}/status	PATCH	200 OK, 404 Not Found
Block User (Admin)	/admin/users/{id}	DELETE	204 No Content, 403 Forbidden

## Part 2: Implement the RESTful APIs

### Task 1: Develop the API to the index.js

1. Continue from the last week exercise, develop the API designed in previous part

### Task 2: The data model

1. Based on the API, draft the data model required for your application
- 

## Submission Requirements

1. Use Case Diagram:
  - Image file or diagram link
2. GitHub Repository with:
  - Complete code.
3. Postman Collection:
  - Export and include the collection file.