

# Senior Backend Engineer - Case Study

## Senior Backend Engineer – Case Study

We are absolutely thrilled that you are interested in joining Rohlik Group! By now, we've had the opportunity to discuss your experience and personality in detail. The next step is to understand how you approach problem-solving, system design, and code quality in practice.

### Task

Write a RESTful application that maintains a database of products and orders, ensuring accuracy, consistency, and scalability while demonstrating ownership, creativity, and efficiency in your solution.

### Requirements

#### Order Operations

- Create an order – Orders can contain one or multiple products in specific quantities.
- Cancel an order – Canceling an order should release reserved stock.
- Pay for an order – Marks the order as paid.

#### Product Operations

- Create a product – Add a new product to the catalog.
- Delete a product – Remove a product (if no active orders depend on it).
- Update a product – Modify product details such as name, price, or stock quantity.
- List products - List all products

#### Business Rules & Constraints

- Every product must have a name, quantity in stock, and price per unit.
- Orders decrease stock availability immediately upon creation.
- If an order exceeds stock availability, return an error response with missing product details.
- Orders reserve stock for 30 minutes. If unpaid within that period, the order becomes invalid, and stock is released.
- Canceled orders immediately release the reserved stock.

#### Build a Front-End for the Product Operations using AI-Powered Code Generative Tooling

- Leverage AI-powered code generative tooling (Cursor.io, Builder.io) to build front end for

the product operations declared above

- Create a user-friendly admin interface
- Pay attention to input validation and error handling

## **Key Considerations & Areas to Explore**

### **Delivery Ownership & Scope Management**

- How would you improve delivery speed?
- Would you propose changes to the scope to accelerate delivery while still meeting business needs?
- What trade-offs would you make to balance fast execution vs. long-term maintainability?

### **Alternative Technologies & AI for Efficiency**

- How would you leverage AI-powered tools (e.g. Copilot, ChatGPT, code generation) to increase development speed and quality?
- Would you consider asynchronous processing (e.g. Kafka, event-driven architecture) to improve performance?
- How would you implement automated testing (unit, integration, end-to-end) to ensure robustness?

### **Creativity & Problem-Solving**

- How would you design a system that can scale efficiently under high load?
- What solution would you use to handle multiple concurrent orders touching the same product?
- How would you ensure high availability and resilience?

### **Effective Product & Engineering Partnership**

- How would you work with a Product Manager to understand business objectives and refine technical solutions accordingly?
- What scope modifications could be proposed to accelerate delivery while maintaining robustness?
- How do you balance business needs with technical constraints?

## **Technical Expectations**

- Implement using Java 17+ and Spring Boot 2+.
- Use an in-memory database (H2) for simplicity, but consider scalability concerns.
- Document API endpoints (e.g., Swagger/OpenAPI).
- Implement proper validation, exception handling, and concurrency control.
- Ensure proper test coverage (unit and integration tests).

## **Presentation & Format**

Please share your work day ahead - source code and presentation. We are flexible with how you present your solution:

- Live discussion – Walk us through your code and decision-making.
- Technical documentation – Provide an overview of your architecture, trade-offs, and improvements.
- Whiteboarding session – Explain your thought process and optimizations.

**Assessment Criteria**

- Code Quality & Readability – Is the code clean, structured, and maintainable?
- Performance & Scalability – Can the system handle increasing load efficiently?
- Technical Creativity – Are alternative technologies or AI-powered solutions used effectively?
- Delivery Ownership – Can scope modifications improve delivery speed?
- Business & Stakeholder Alignment – Does the solution balance technical execution with business goals?
- Testing & Robustness – Is the solution well-tested and resilient?