Task Description

Your task is to create a Spring Boot application with the following requirements:

1. **CRUD Operations for Orders**:
    - Develop a backend that allows creating, reading, updating, and deleting orders.
    - Use a relational database of your choice (e.g., PostgreSQL, MySQL) to persist order data.
2. **Caching with Hazelcast**:
    - Integrate Hazelcast for caching.
    - When the user requests **all orders**, fetch them from the cache instead of querying the database.
    - Ensure that the cache is updated appropriately after any CRUD operation.
3. **Message Queue with RabbitMQ**:
    - Set up RabbitMQ for message handling.
    - When a user makes a **payment for an order**, send a message to a RabbitMQ queue.
    - Implement a RabbitMQ listener that consumes the payment messages and stores the data in a transactions table in the database.
4. **Transaction Table**:
    - Create a transactions table to store the payment information received from the RabbitMQ listener.
5. **Documentation and Instructions**:
    - Document how to run the application, including setting up Hazelcast and RabbitMQ.
    - Provide example API requests for each CRUD operation, retrieving orders from the cache, and triggering payment messages.
6. **Bonus Points**:
    - Use Swagger for API documentation.
    - Add proper exception handling and validation.
    - Write basic unit tests for critical components.
    - Write a Dockerfile.

Deliverables:

- A GitHub repository containing the code and README file with instructions.
- The database schema (SQL script or migration files).
- Brief documentation describing your solution.