

1. Introduction

The Product Catalog System is a backend application built using Spring Boot that provides a RESTful API for managing a catalog of products. This document provides an overview of the system architecture, technologies used, database schema, RESTful API endpoints, deployment instructions, and concluding remarks.

2. System Architecture

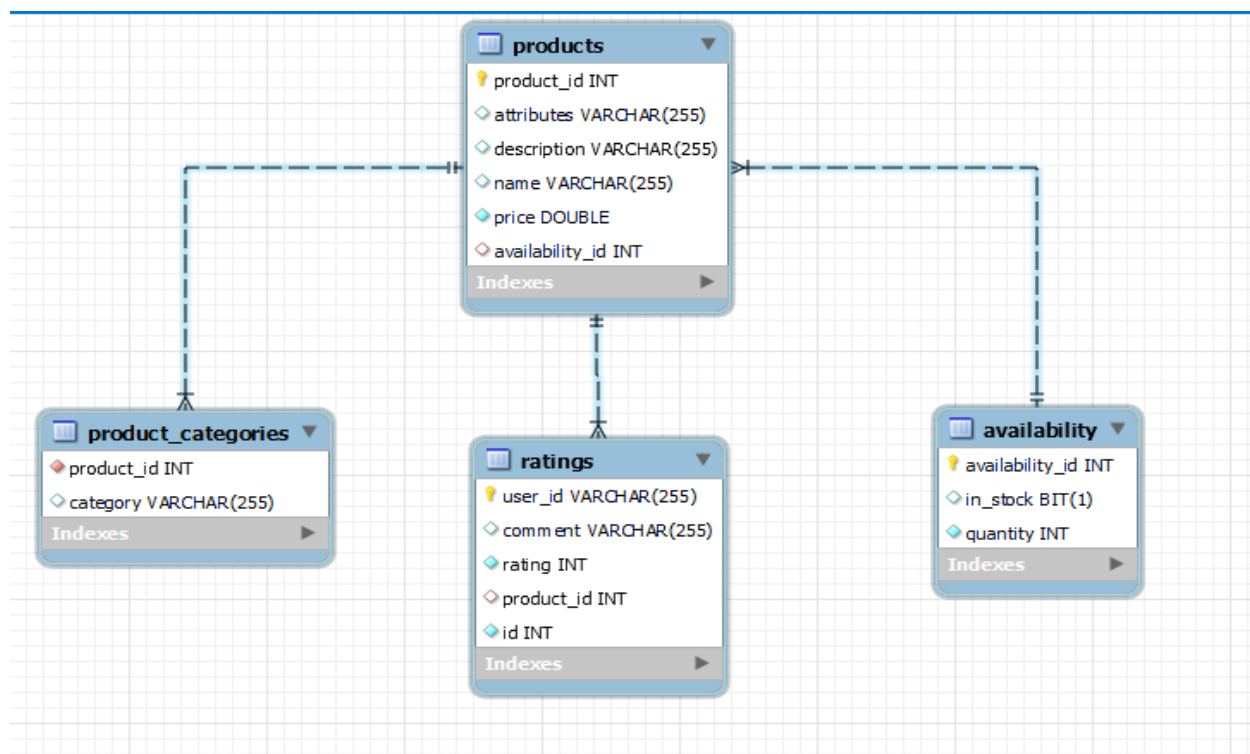
The system follows a simple architecture where Spring Boot serves as the backend, providing RESTful API endpoints for CRUD (Create, Read, Update, Delete and search) operations on product data. There is no frontend component included in this implementation.

3. Technologies Used

- Backend: Spring Boot, Spring Data JPA, Hibernate ORM
- Database: MySQL
- Postman : API development environment for testing and debugging API endpoints.

4. Database Schema

The database schema includes the following tables:



5. RESTful API Endpoints

The following RESTful API endpoints are available:

- **GET /http://localhost:9797/products:** Retrieve all products
- **GET /http://localhost:9797/products/2:** Retrieve a specific product by ID
- **POST /http://localhost:9797/products:** Create a new product
- **PUT / http://localhost:9797/products/2:** Update an existing product
- **DELETE / http://localhost:9797/products/2:** Delete a product by ID
- **GET/ http://localhost:9797/products/search?category=BOOK:** Search products by name, category, or attribute

6. Deployment

Prerequisites

- Java Development Kit (JDK)
- Apache Maven
- MySQL Database
- Eclipse

Steps

1. Clone Repository:

- Begin by cloning the project repository from your Git source control platform.

2. Configure Database Settings:

- Open the application.properties file in your project.
- Configure the database connection settings, including URL, username, password, and driver class name.

3. Build Project with Maven:

- Open a terminal or command prompt in the project directory.
- Execute the Maven command `mvn clean install` to build the project and generate the executable JAR file.
- Save the updated pom.xml file in your project directory.

4 .Run Application:

- Once the build is successful, navigate to the target directory using the terminal.
- Execute the command `java -jar target/product-catalogue-system.jar` to run the Spring Boot application.

5 .Verify that your Spring Boot application is running by accessing `http://localhost:9797` in your web browser. If your application uses a different port, replace 9797 with the correct port number.

7. Conclusion

The Product Catalog System backend provides a straightforward solution for managing product data through RESTful APIs.