Online Bookstore Microservices Project

Project Progress Report

Prepared For: Prof. Pedram Habibi
Prepared By: Group 9
Group Members:

Helly Rajeshbhai Patel
Jiyad Mohammed
Arif Shaikh
Nicholas Nwanua Ilechie
Nirajbhai Ranchhodbhai Limbasiya

Date Submitted: June 12th, 2025

https://github.com/SOA915-NAA-GROUP9/online-bookstore-microservices

Introduction

This report outlines the initial design and setup of our Online Bookstore Microservices Project, developed as part of the Service-Oriented Architecture (SOA) course. The objective is to build a scalable and modular e-commerce application that allows users to register, browse available books, place orders, and receive notifications.

To achieve this, the system adopts a microservices architecture comprising four core services: User Service, Product Service, Order Service, and Notification Service. Each service is independently developed with well-defined responsibilities and exposed via RESTful APIs. These services are containerized using Docker, orchestrated with Kubernetes, and supported by a continuous integration/continuous deployment (CI/CD) pipeline and monitoring tools to ensure reliability and scalability.

This report covers the project's use case and scope, the design and responsibilities of each microservice, API contracts, the initial system architecture, and our GitHub repository setup, including documentation and branching strategy. As a team we have worked collaboratively to establish a solid foundation for iterative development in the weeks ahead.

https://github.com/SOA915-NAA-GROUP9/online-bookstore-microservices

Use Case and Overall Project Scope

We propose developing an **online bookstore** as our microservices-based application. This platform will enable users to browse a catalog of books, register and manage their profiles, place orders, and receive notifications about their transactions. The project will leverage a microservices architecture, where each service is independent, modular, and communicates via RESTful APIs. Our scope includes:

- Designing and implementing at least four core microservices.
- Containerizing the services using Docker and orchestrating them locally with Kubernetes (e.g., Minikube or Kind).
- Setting up testing, monitoring, and a CI/CD pipeline to ensure reliability and automation.
- Documenting the architecture, APIs, and setup instructions in a Git repository.

This use case allows us to explore real-world scenarios like user management, inventory handling, order processing, and event-driven notifications, aligning with the SOA principles outlined in the project document.

Core Microservices and API Contracts

We have identified the following four core microservices for the online bookstore, each with specific responsibilities and RESTful API contracts:

1. User Service

- **Responsibilities**: Manages user registration, authentication, and profile information.
- **API Contracts**:
    - POST /users/register – Registers a new user (e.g., accepts JSON with username, password, email).
    - POST /users/login – Authenticates a user and returns a token (e.g., JWT).
    - GET /users/{id} – Retrieves details of a specific user by ID.
    - PUT /users/{id} – Updates user profile information (e.g., name, email).

2. Product Service

- **Responsibilities**: Manages the book catalog, including adding, updating, and retrieving book details.
- **API Contracts**:
    - GET /books – Returns a list of all books (supports query parameters like genre or author).
    - GET /books/{id} – Retrieves details of a specific book by ID.
    - POST /books – Adds a new book to the catalog (admin-only endpoint).
    - PUT /books/{id} – Updates details of an existing book (admin-only).

3. Order Service

- **Responsibilities**: Handles order creation, status updates, and order history management.
- **API Contracts**:
    - POST /orders – Creates a new order (e.g., accepts user ID and book IDs).
    - GET /orders/{id} – Retrieves details of a specific order by ID.
    - PUT /orders/{id}/status – Updates the status of an order (e.g., "shipped", "delivered").

4. Notification Service

- **Responsibilities**: Sends notifications to users (e.g., order confirmations, promotions) via email or SMS.
- **API Contracts**:
  - POST /notifications/email – Sends an email notification (e.g., accepts recipient email and message).
  - POST /notifications/sms – Sends an SMS notification (e.g., accepts phone number and message).

These services align with the suggested microservices in the project document and provide a modular structure for our application.

Initial Architecture Diagram and Service Responsibilities

Here’s an initial architecture diagram illustrating how the microservices interact:

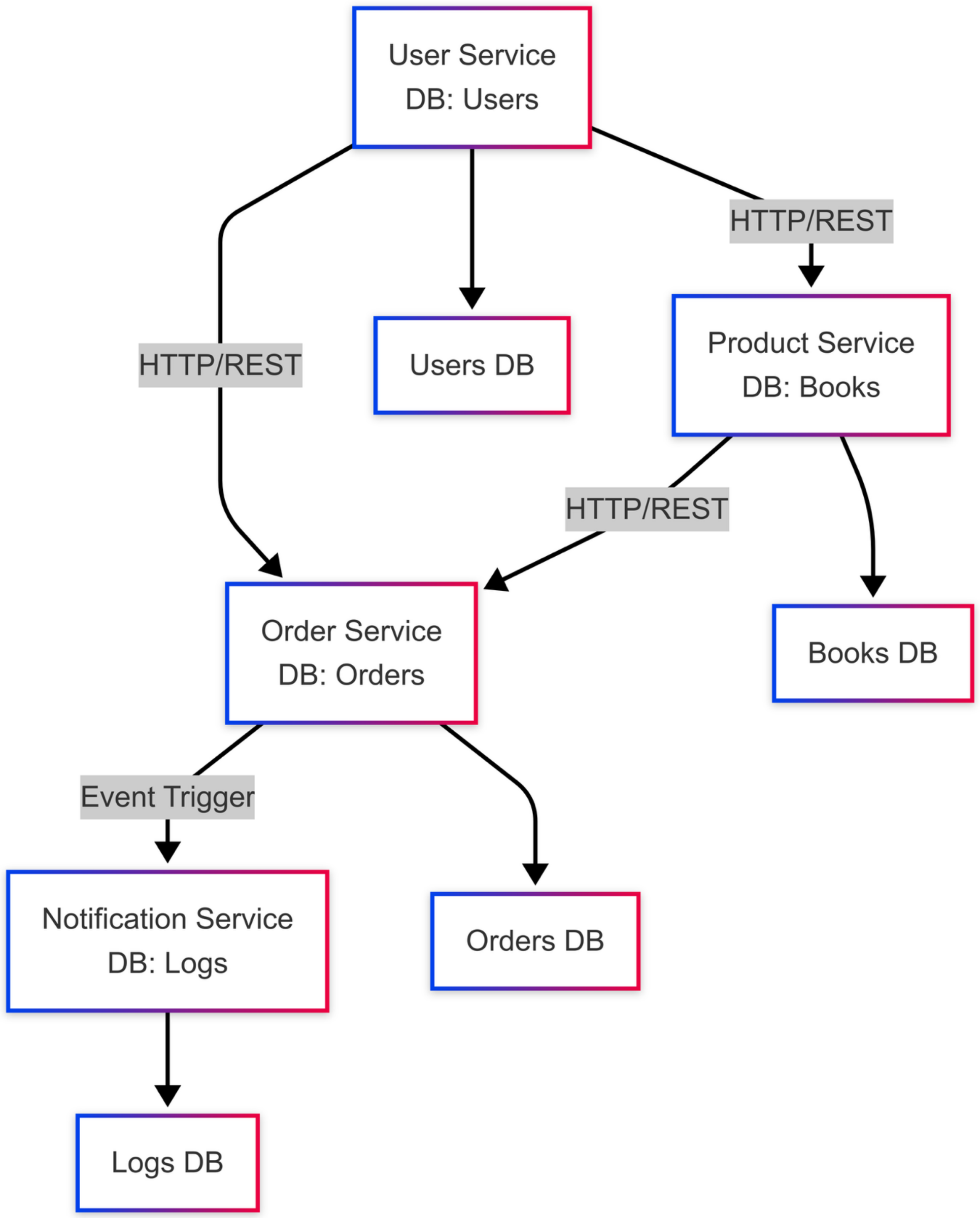

Figure 1: Architecture diagram

https://github.com/SOA915-NAA-GROUP9/online-bookstore-microservices

- **Service Interactions**:
  - The **Order Service** interacts with the **User Service** to verify user details and the **Product Service** to check book availability before processing an order.
  - The **Notification Service** is triggered by events from the **Order Service** (e.g., sending an order confirmation email).
- **Decentralized Data**: Each service has its own database to adhere to microservices principles (e.g., Users DB, Books DB, Orders DB, Notification Logs DB).
- **Service Responsibilities**:
  - **User Service**: User authentication and profile management.
  - **Product Service**: Book inventory management.
  - **Order Service**: Order processing and tracking.
  - **Notification Service**: Event-driven notifications.

Git Repository and Project Documentation

- **Git Repository**: We have set up a Git repository on GitHub, accessible via [https://github.com/SOA915-NAA-GROUP9/online-bookstore-microservices]. The repository includes:
    - A README.md file with an overview of the project, setup instructions, and prerequisites (e.g., Docker, Kubernetes).
    - Initial folder structure for each microservice (e.g., /user-service, /product-service).
- **Documentation**: Initial documentation includes:
    - The architecture diagram (above).
    - Service responsibilities and API contracts (as listed).
    - Setup instructions for running the project locally (to be expanded as we progress).

The repository will be updated as we implement the services, Dockerfiles, and Kubernetes manifests.

https://github.com/SOA915-NAA-GROUP9/online-bookstore-microservices

Service List, Architecture Draft, and Initial Project Plan

Service List:

- User Service
- Product Service
- Order Service
- Notification Service

Architecture Draft: See the diagram above in figure 1, which outlines service interactions and data separation.

Initial Project Plan:

Phase 1: Microservices Design and Dockerization (Weeks 6-7)

- Implement User and Product Services with RESTful APIs.
- Dockerize both services and test locally with Docker Compose.
- Assign Helly and Jiyad to User Service, Nicholas and Niraj to Product Service.

Phase 2: Kubernetes Deployment (Weeks 9-10)

- Add Order and Notification Services.
- Write Kubernetes YAML files and deploy all services to a local cluster (e.g., Minikube).
- Helly and Jiyad handle Order Service, Nicholas and Niraj handle Notification Service.

Phase 3: Testing, CI/CD, and Monitoring (Week 11)

- Write unit and integration tests for all services.
- Set up a CI/CD pipeline (e.g., GitHub Actions) and monitoring (e.g., Prometheus).
- Team collaborates on testing and pipeline setup.

Phase 4: Final Presentation (Week 12)

- Prepare a demo and presentation showcasing the architecture, deployment, and functionality.
- All members contribute to the final presentation.

Team Roles

With a group of four :

1. Helly Rajeshbhai Patel,
2. Jiyad Mohammed Arif Shaikh,
3. Nicholas Nwanua Ilechie,
4. Nirajbhai Ranchhodbhai Limbasiya

We plan to:

- Pair up initially: Helly and Jiyad on User and Order Services; Nicholas and Niraj on Product and Notification Services.
- We also intend to rotate responsibilities as needed to ensure everyone gains experience with all aspects (e.g., Docker, Kubernetes, CI/CD).

Next Steps

For Thursday's presentation, we will:

- Present the online bookstore use case and scope.
- Introduce the four microservices with their API contracts.
- Display the architecture diagram and explain service responsibilities.
- Share the Git repository link and initial documentation.
- Outline our project plan and team

We look forward to feedback from Prof. Pedram and our peers during the lab session!

https://github.com/SOA915-NAA-GROUP9/online-bookstore-microservices