Book store application

Prundus Vlad-George

Gr. 258

**Description:**

Book store application is a web application that offers the possibility to buy an online book with a click of a button. The system allows new users to create an account, to buy a book using a provided payment system, to read information about it and to chat with other users in an online lecture club. The store is secured and can only provide books to those logged in. You can also see the number of users that bought the products before you and you can also access your orders in your personal account.

**Level 1 - System context diagram:**

Diagram

Description automatically generated

**Level 2 - Container diagram:**

**Diagram, schematic

Description automatically generated**

**Level 3 - Component diagram:**

**User microservice component diagram:**

**Diagram

Description automatically generated**

**Book microservice component diagram:**

Diagram

Description automatically generated

**Book lecture chat microservice component diagram:**

Diagram

Description automatically generated

**Level 4 - Code:**

**User microservice UML diagram:**

**Graphical user interface

Description automatically generated**

**Book microservice UML diagram:**

**Graphical user interface

Description automatically generated with medium confidence**

**Book lecture chat microservice UML diagram:**

Graphical user interface

Description automatically generated

**Zuul UML diagram:**

**Graphical user interface

Description automatically generated**

**Eureka UML diagram:**

Graphical user interface, application

Description automatically generated

**Technologies used in the application:**

For communication with the microservices, I used Eureka (Spring Cloud Netflix Eureka) which holds detailed information about each one of the microservices. Eureka also helps the load-balancing and failover processes. For the API Gateway, I used Zuul (Zuul API Gateway) which is an application server that handles routing in a microservices architecture in Spring Boot applications. It handles all the requests and performs the dynamic routing of microservice applications. For the book lecture chat, I used Kafka and websockets as messaging service that publishes and subscribes to messages. For deploying the application, I used Docker images to build Docker containers for each one of the application’s microservices.

**SOA Patterns Used**:

The application is using multiple patterns: Microservices Pattern (the application is divided into multiple services; each service is taking care of a certain part of the application functionality), Integration Pattern (by integrating into the application the payments service), Facade Pattern (used to wrap third- party integrations to improve software design for payment service)