Project Documentation

This application is an online ticket store designed to sell tickets for various sports events and provide detailed information about them. It targets users interested in purchasing tickets for sports events or simply exploring event details online.

The application displays a list of sports events, allowing users to view event details and purchase tickets. While browsing the event list or viewing event details does not require login, purchasing tickets requires the user to be logged in. After a ticket purchase, the application simulates sending an invoice email to the user. Currently, this email is represented as a console message displaying the invoice content.

The backend of the application is structured using six microservices, each serving a specific purpose. These include an authentication service (authservice), a load balancer implemented with Nginx, a service for retrieving sports event details (sports-service), a service for generating tickets (tickets-service), a service for generating invoices (invoices-service), and a service for simulating email notifications (email-service). The load balancer efficiently distributes incoming requests among the authservice, sports-service, and tickets-service. Communication between tickets-service and invoices-service is facilitated through a Kafka queue, while invoices-service interacts with email-service via a RabbitMQ queue.

A diagram below provides an overview of the backend architecture and its services.

O imagine care conține text, diagramă, Plan, Paralel

Conținutul generat de inteligența artificială poate fi incorect.

The frontend is a separate microservice, that exposes pages for login, viewing the sports events, viewing the details to a sport event and buying a ticket.

The application uses Docker for creating the images and deploying the containers.

UML Diagram of frontend package:

O imagine care conține text, captură de ecran

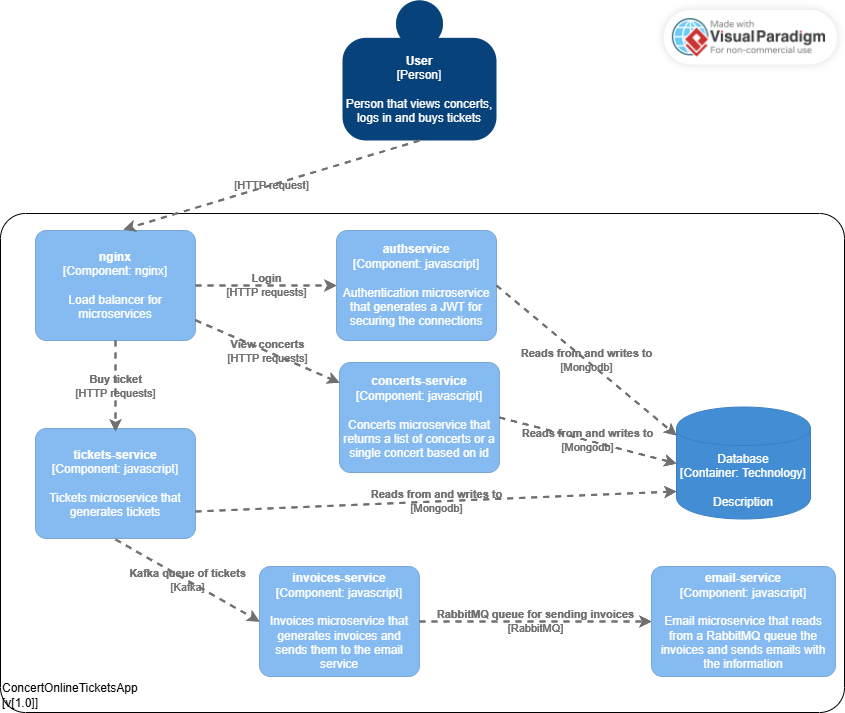
Conținutul generat de inteligența artificială poate fi incorect.

O imagine care conține text, captură de ecran, Font, diagramă

Conținutul generat de inteligența artificială poate fi incorect.c4 Context Diagram:

c4 Container Diagram:

O imagine care conține text, captură de ecran, diagramă

Conținutul generat de inteligența artificială poate fi incorect.

UML Backend diagram: