

Application Design

Austrian Tourism Blogger / ReisePortal



September 23, 2021

FH Technikum

Chahed Rajoub (if19b166)

Stefan Tirea (if19b207)

Table of Contents

[Introduction: 2](#_Toc90231580)

[GitHub repository: 2](#_Toc90231581)

[Implementations: 2](#_Toc90231582)

[Use Case Diagram: 3](#_Toc90231583)

[UML Diagram: 4](#_Toc90231584)

[Database Design: 5](#_Toc90231585)

[Backend: 5](#_Toc90231586)

[Frontend: 5](#_Toc90231587)

[API Contract/Specifications: 6](#_Toc90231588)

# Introduction:

* This travel portal is used to stimulate Austrian tourism blogger; authors can publish their sensational blog posts on given Austrian sights. These articles are designed to get the reader interested in having a vacation in the Alpine republic.
* 1 cent is to be paid out per article view. The commission is paid to the authors monthly.
* Data is collected statistically. readers can see where blogs are particularly popular right now.
* It is implemented in the form of a microservice architecture
* UML diagram is developed based on this use case.

## GitHub repository:

<https://github.com/Shahed-RAJOUB/ReiseProtal>

# Implementations:

* Java
* Maven
* Java Persistence API
* Spring Boot
* Spring Cloud Config
* Eureka
* Spring Cloud Load Balancer
* Spring Cloud Gateway
* Kafka
* PostgreSQL
* Zipkin server
* Docker

# Use Case Diagram:

Diagram

Description automatically generated

# UML Diagram:

Diagram

Description automatically generated

# Database Design:

Diagram

Description automatically generated

* Three different tables: Author, Location, Blogs
* Each Blog has two foreign keys one for an Author and other for location to simplify the design
* Docker container for common Postgres Database is used for this design and services are connected to the same container

# Backend:

Notes on the backend:

* The payment of the commission to the authors is only to be carried out fictitiously. It is sufficient to issue a corresponding line of text on the standard console.
* All technologies presented should be used sensibly.
* Only the functionality that is required should be implemented.

# Frontend:

Notes on the front end:

* There should be a page for an author to enter a blog post. Each article is assigned to a specific point of interest.
* For reading articles there should be an overview page with the headlines and given details page on which a specific article is presented.
* There should be a page on which you can view the total statistics as well as monthly statistics.
* The validation of the data entered, and the styling of the pages can be completely dispensed with.

# API Contract/Specifications:

Postman collection documentation: <https://documenter.getpostman.com/view/12261877/UVR4Pqdm>