

Spring Microservices Project (Due 01/31/2025)

Fleet Management

Objective:

Create a military fleet-management application that enables an admiral to manage the commanders and pilots for the squadrons. Choose any real or fictional domain you like! But the admiral must possess the ability to view, add, remove and alter any of the squadrons, commanders, and pilots, including moving a commander or pilot from one squadron to another. The application should make it easy for admirals to do their jobs, so clear, consistent, REST-standard endpoints are critical. The application should also take into consideration possible edge cases, like squadrons having a maximum capacity, what happens if no commander is assigned to a squadron, what happens when squadrons with pilots are deleted, etc.

Functional Requirements:

- Must be a microservices-based REST API consisting of:
 - Spring Cloud Gateway
 - Eureka Service Discovery
 - At least 3 additional Spring Boot microservices
 - MySQL database
- All code should be available on a public GitHub repository
- Possesses all required CRUD functionality
- Handles edge cases, errors, and exceptions effectively

Non-Functional Requirements:

- Well-documented code
- Code upholds industry best practices (Solid/Dry/Secure)

Bonus Objectives (merely suggestions):

- Create an Angular frontend application that utilizes your API
- Achieve 90% code coverage for all microservices
- Add Swagger documentation for all API/microservice endpoints
- Use RabbitMQ for inter-service messaging/queueing
- Incorporate an external (web-based) API for some relevant data
- Have the system dynamically generate a training or patrol schedule
- Deploy the entire application to the AWS or Azure cloud
- Anything else fun you can think of (*after completing the requirements!*)