



UNAMIBIA
UNIVERSITY
OF SCIENCE
AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS
DEPARTMENT OF SOFTWARE ENGINEERING

Assignment 2

Course Title: Distributed Systems and Applications
Course Code: DSA612S
Assessment: Second Assignment

Released on: 08/10/2023.
Due date: 25/10/ 2023 at midnight

Total Marks: 100

Question 1 - GraphQL

Description

The Faculty of Computing and Informatics (FCI) at NUST has initiated the development of a Performance Management System. This system is designed to track the performance of staff members consistently and measurably within the faculty. The process of tracking staff performance involves three key users: the Head of Department (HoD), the employee, and the supervisor.

To assess the performance of staff members, each department within the Faculty establishes annual objectives. Each of these objectives contributes a certain percentage to the overall departmental goals. Additionally, every employee within a department has specific Key Performance Indicators (KPIs) that are directly related to these departmental objectives.

Furthermore, supervisors are responsible for evaluating and grading the KPIs of their respective employees. These KPIs may be measured using various units, depending on the specific metric being assessed. These units can include percentages, time, counts (numbers), currency, scores (e.g., on a scale of 1 to 10), ratios, units of measurement (e.g., kilograms, litres), distances, and others.

Moreover, the system is designed to calculate an employee's score (ranging from 1 to 5) for each KPI once the appropriate unit is inputted. Finally, the system is capable of computing the total scores of employees, presented as a percentage of the department's objectives.

All the information to be used must be input into the system and stored in a database (e.g., MySQL or MongoDB). Furthermore, every operation within the system necessitates the successful authentication of the user with the appropriate authorization. The authorization model is defined as follows:

HoD

- Create department objectives.
- Delete department objectives.
- View Employees Total Scores.
- Assign the Employee to a supervisor.

Supervisor

- Approve Employee's KPIs.
- Delete Employee's KPIs.
- Update Employee's KPIs.
- View Employee Scores. (Only employees assigned to him/her).
- Grade the employee's KPIs

Employee

Create their KPIs

Grade their Supervisor

View Their Scores

Your task is implementing the client and service in the Ballerina language using GraphQL.

Additional Information

Note: Key Performance Indicators(KPIs) are specific, measurable metrics or targets used to evaluate the performance of individual employees within an organization. For example, a lecturer's KPIs can be Peer Recognition, Professional Development, Student Progress and Success, Innovative teaching methods, Research Output etc.

Important things to note:

1. You need to store user information, like first name, last name, job title, position, and role, along with performance records, including KPIs and objectives, in a data store, such as MongoDB or SQL databases.
2. Use GraphQL for communication between client and server.
3. Implement the client and server in the Ballerina Language.
4. Deploy the service and client in Docker containers.

Evaluation criteria

- Setup of the MongoDB/SQL instance. (10 marks).
- Docker container configuration. (5 marks).
- Implementation of the client in the Ballerina language. (10 marks).
- Implement the services in the Ballerina language. (25 marks).

Question 2 - Microservices

Description

In this project, we're working on a part of a bigger healthcare system. Our part of the system includes a health admin service and various specialist services like dermatology, gastroenterology, gynaecology, and more. We're only focusing on situations where a patient needs to see a specialist.

Here's how it works:

1. When a patient needs medical help from a specialist, they send a request to the health admin service. In this request, they specify which specialist they want to see, when they're available, and their personal information like first name, last name, and phone number.
2. Once the health admin service gets the request, it talks to the different specialist services to find an appointment for the patient. Sometimes, the specialist services might need to talk to each other to figure out the best schedule.
3. When all the appointments are set, the health admin service sends all the details back to the patient in response.

Your job is to create and set up this healthcare sub-system.

Important things to note:

1. You are required to adopt a microservice architectural style to design the distributed system. (i.e admin service and 3 other specialist services of your choice).
2. Use a Kafka instance to support the communication between a client and your service(s).
3. Use data stores like (MongoDB or SQL) to store patient information and appointments.
4. Finally, each service will be deployed in a Docker container.

Evaluation criteria

- Setup of the Kafka cluster, including topic management. (15 marks)
- Setup of the MongoDB/SQL instance. (5 marks)
- Docker container configuration. (5 marks)
- Implementation of the client and services in the Ballerina language. (25 marks)

Submission Instructions

- This assignment is to be completed by groups of four students each.
- For each group, a repository should be created on Github. The repository should have all group members set up as contributors.
- All assignments must be uploaded to a GitHub repository. Students who haven't **pushed** any codes to the repository will not be given the opportunity **to present and defend the assignment**. More particularly, if a student's username does not appear in the commit log of the group repository, that student will be assumed not to have contributed to the project and thus be awarded the mark 0.
- The assignment will be group work, but individual marks will be allocated based on each student's contribution to the assignment.
- Marks for the assignment will only be allocated to students who have presented the assignment.
- It's the responsibility of all group members to make sure that they are available for the assignment presentation. An assignment cannot be presented more than once.
- The submission **date is Wednesday, October 25, 2023, at midnight**. Please note that commits after that deadline will not be accepted. Therefore, a submission will be assessed based on the clone of the repository at the deadline.
- Any group that fails to submit on time will be awarded the mark 0. Late Submission will not be considered.
- There should be no assumption about the execution environment of your code. It could be run using a specific framework or on the command line.