

# Assignment 1

**Course Title: Distributed Systems and Applications**

**Course Code: DSA612S**

**Assessment: First Assignment**

**Released on: 30/08/2024.**

**Due date: 20/09/ 2024 at 23h59**

**Total Marks: 100**

## Question 1: Restful APIs

The problem centers on developing a Restful API for effectively managing Programme development and review workflow within the Programme Development Unit at Namibia University of Science and Technology. A programme is made up of multiple courses. A programme is characterized by essential attributes, including a programme code, NQF Level of Qualification, Faculty and Department name, Programme/Qualification Title, registration date and a list of the courses within the programme. Additionally, a course is characterized by specific details such as the course name, course code, and NQF (National Qualifications Framework) level. The programme runs for 5 years after registration before they are due for review.

The API should include the following functionalities:

- Add a new programme.
- Retrieve a list of all programme within the Programme Development Unit.
- Update an existing programme's information according to the programme code.
- Retrieve the details of a specific programme by their programme code.
- Delete a programme's record by their programme code.
- Retrieve all the programme that are due for review.
- Retrieve all the programmes that belong to the same faculty.

Note that the programme code should serve as a unique identifier for a programme.

Your task is to design and implement the client and service following the Restful API architecture using the **Ballerina language**.

Deliverables:

- **Service Implementation**
  - Add a new programme. **(10 marks)**

- Retrieve a list of all programme within the Programme Development Unit. **(5 marks)**
  - Update an existing programme's information according to the programme code. **(5 marks)**
  - Retrieve the details of a specific programme by their programme code. **(5 marks)**
  - Delete a programme's record by their programme code. **(5 marks)**
  - Retrieve all the programme that are due for review. **(5 marks)**
  - Retrieve all the programmes that belong to the same faculty. **(5 marks)**
- Client Implementation in a Ballerina that effectively interacts with the implemented API. **(10 marks)**

## Question 2: Remote invocation: Online Shopping System using gRPC

Your task is to design and implement an online shopping system using gRPC that allows two types of users—a customer and an admin—to interact with the system. The system should provide essential functionalities for managing products, adding them to a cart, placing an order, and processing it. The customer should be able to view available products, search for a product, add it to their cart, and place an order. On the other hand, an admin should be able to add a new product, update a product's details, remove a product, and list all orders.

**In short, we have the following operations:**

- **add\_product:** The admin creates a new product. The product should have the following fields: name, description, price, stock quantity, SKU (Stock Keeping Unit), and status (available or out of stock). This operation should return the unique code for the added product.
- **create\_users:** Multiple users (customers or admins) each with a specific profile are created and streamed to the server. The response is returned once the operation completes.
- **update\_product:** The admin alters the details of a given product.
- **remove\_product:** The admin removes a product from the inventory. The function should return the updated list of products after the removal.
- **list\_available\_products:** The customer gets a list of all the available products.
- **search\_product:** The customer searches for a product based on its SKU. If the product is available, the function should return the product's details; otherwise, notify the customer that the product is not available.

- **add\_to\_cart:** The customer adds a product to their cart by providing their user ID and the product's SKU.
- **place\_order:** The customer places an order for all products in their cart.

Your task is to define a protocol buffer contract with the remote functions and implement both the client and the server in the Ballerina Language.

### Server Implementation:

Implement the server logic using the Ballerina Language and gRPC. Your server should handle incoming requests from clients and perform appropriate actions based on the requested operation.

### Client Implementation:

The clients should be able to use the generated gRPC client code to connect to the server and perform operations as implemented in the service. Clients should be able to handle user input and display relevant information to the user.

*Please be aware that you have the freedom to include additional fields in your records if you believe they would enhance the performance and overall quality of your system.*

Deliverables:

We will follow the criteria below to assess this problem:

- Definition of the remote interface in Protocol Buffer. (15 marks)
- Implementation of the gRPC client in the Ballerina language. (10 marks)
- Implementation of the gRPC server and server-side logic in response to the remote invocations in the Ballerina Language. [25 marks]

## Submission Instructions

- This assignment is to be completed by groups of 5-7 students each.
- For each group, a repository should be created on Gitlab or github. The repository should have all group members set up as contributors.
- All assignments must be uploaded to a GitHub or GitLab 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 deadline **date is Friday, September 20, 2024, at 23h59**. 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 Submiss
- 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.
- In the case of plagiarism (groups copying from each other or submissions copied from the Internet), all submissions involved will be awarded the mark 0, and each student will receive a warning.