

## COEN 6731 Winter 2024 Assignment Two

This assignment has the purpose of practicing (1) RPC communication for resources and services; (2) data operations on MongoDB, a NoSQL database; (3) implementation of aggregation pipelines for data processing.

A public data source is available on Kaggle. It is compiled from the National Center of Education Statistics Annual Digest, USA. The data contains the statistics of average undergraduate tuition and fees and room and board rates charged for full-time students in degree-granting postsecondary institutions.

<https://www.kaggle.com/datasets/kfoster150/avg-cost-of-undergrad-college-by-state/versions/10?resource=download>

**Avg Cost of Undergrad College by State**

Data Card Code (2) Discussion (0)

33 New Notebook Download (22 kB)

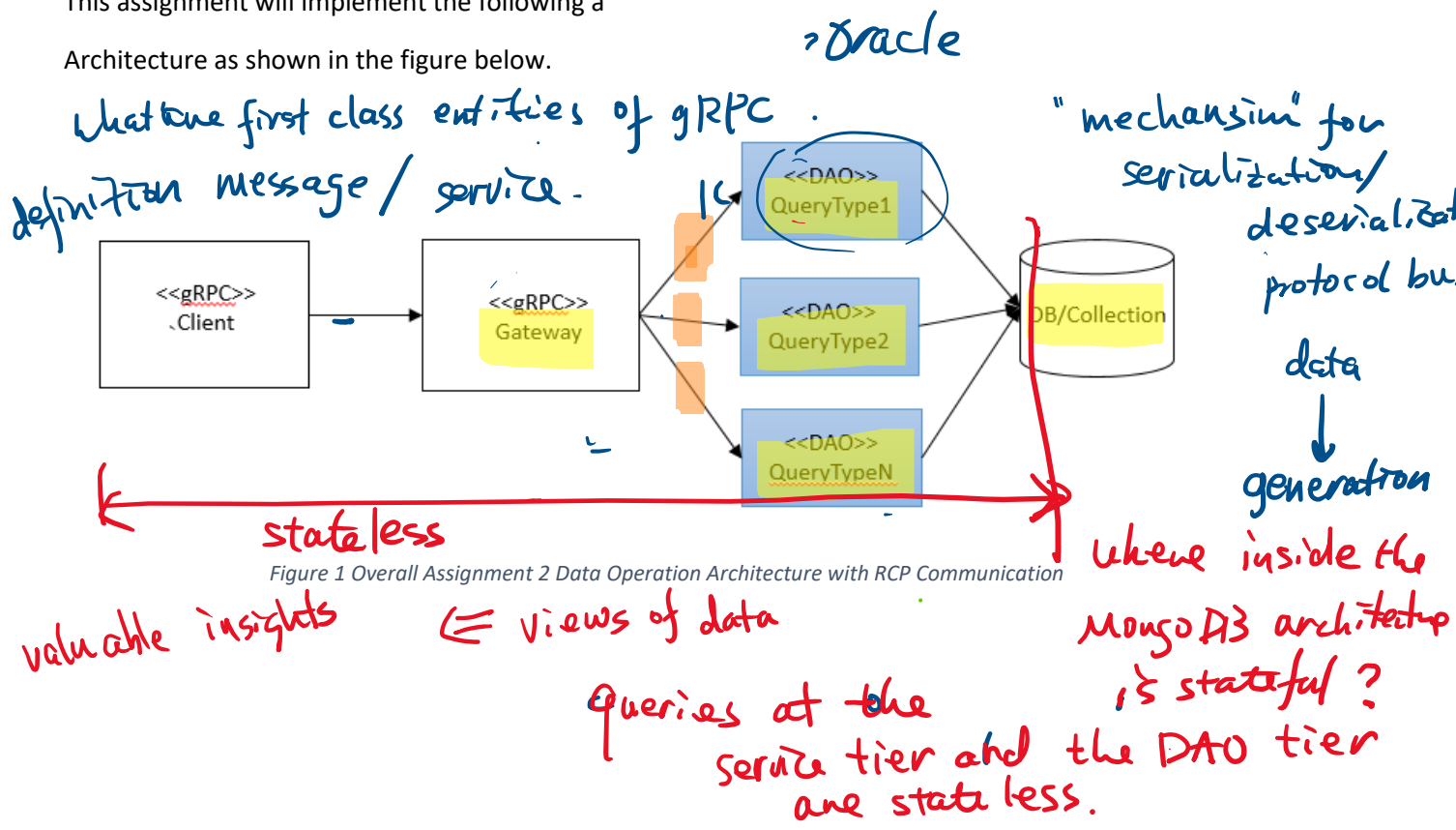
State	Type	Length	Expense	# Value
Alabama	Public Out-of-State	4-year	Fees/Tuition	4848
Alabama	Public In-State	2-year	Fees/Tuition	8873
Alabama	Public In-State	4-year	Fees/Tuition	8473
Alabama	Public In-State	4-year	Room/Board	8473
Alaska	Public In-State	2-year	Fees/Tuition	3972
Alaska	Public In-State	4-year	Fees/Tuition	6317
Alaska	Public In-State	4-year	Room/Board	9898
Arizona	Public In-State	2-year	Fees/Tuition	1842

Summary

- 1 file
- 6 columns
- String
- Integer

This assignment will implement the following a

Architecture as shown in the figure below.



## Task 1. MongoDB Data Storage and Operation

Task 1.1 is to create a MongoDB **collection** named **EduCostStat** to store the data to the MongoDB instance running on MongoDB online cluster. No local database is accepted. The dataset can be downloaded as excel sheets or a .csv file. Please write a program to create a collection in the MongoDB by reading the data samples from the file. (10 marks)

*files (JSON)*  
*Importing .csv to MongoDB collection*

Task 1.2 is to develop data access objects in Java that represents different queries to the data. Please be aware that **duplicated queries should not be inserted as a new document in a collection.**

- 1) Query the cost given specific year, state, type, length, expense; and save the query as a document in a collection named **EduCostStatQueryOne**. (10 marks)
- 2) Query the **top 5 most expensive states** (with overall expense) given a year, type, length; and save the query as a document in a collection named **EduCostStatQueryTwo**. (10 marks)
- 3) Query the **top 5 most economic states** (with overall expense) given a year, type, length; and save the query as a document in a collection named **EduCostStatQueryThree**. (10 marks)
- 4) Query the **top 5 states of the highest growth rate of** overall expense given a range of past years, one year, three years and five years (using the latest year as the base), type and length; and save the query as a document in a collection named **EduCostStatQueryFour**. (10 marks)
- 5) **Aggregate region's average** overall expense for a given year, type and length; and save the query as a document in a collection named **EduCostStatQueryFive**. The region's map can be found here: <https://education.nationalgeographic.org/resource/united-states-regions/> (you can search the key words "Five US Region Map" to find our text based mapping) (10 marks)

## Task 2. Data Communication Interface Definition and Service Implementation

Task 2.1 Define a **ProtocolBuff definition file** to represent the request, response and service for each query in Task 1. (10 marks) You may use the conversion tool from JSON to ProtocolBuff.

*gRPC messages / services*

*type*

<https://www.site24x7.com/tools/json-to-protobuf.html>

Task 2.2 Develop the Java program for each service defined in Task 2.1 as gRPC services. Each service invokes the corresponding data access object classes developed in Task 1. (50 marks)

Task 2.3 Develop the gRPC client and server (or gateway) code to communicate as RPC calls for the five queries defined in Task 1. (20 marks)

## Submission Specification

*communication characteristics*

- Redraw the Figure 1 to reflect the context of your distributed system architecture. Provide the concrete names of each component in Figure 1 as your own program solution. For example, replace the name "gateway" with the actual component's name that implements service.
- A report in PDF that documents the concrete architecture diagram with explanation of each component's function in the diagram and presents the solution of tasks 1 and 2 with necessary screen shots.
- The project code with the pom.xml file.
- Make one archive file of the project code and the report with .zip or .gz or .tar. NO .rar is accepted for grading.
- The deadline of submission is March 31 23:59