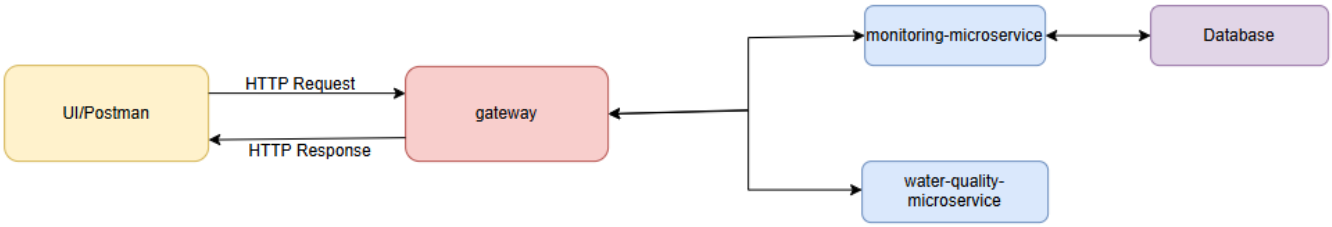
Module Title	Advanced Programming
Module Code	KF7014
Academic Year / Semester	2024-25 / Semester 2
Module Leader	Dr Maria Salama

ASSESSMENT – Development of microservices-based application	
Individual Component	
Student Name	Prathamesh Belnekar
Student ID	W24044966
Programme	MSc Advanced Computer Science

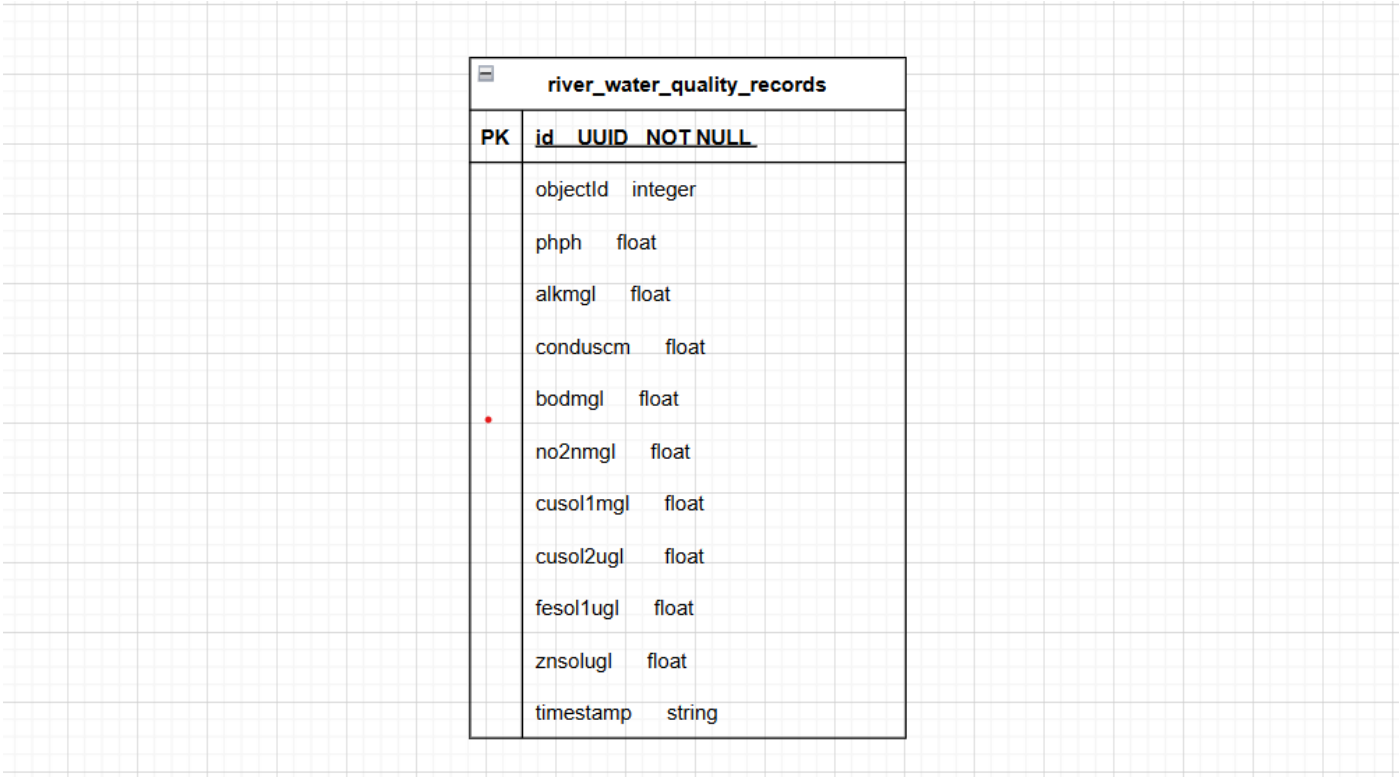
Word Count: 466

Table of Contents

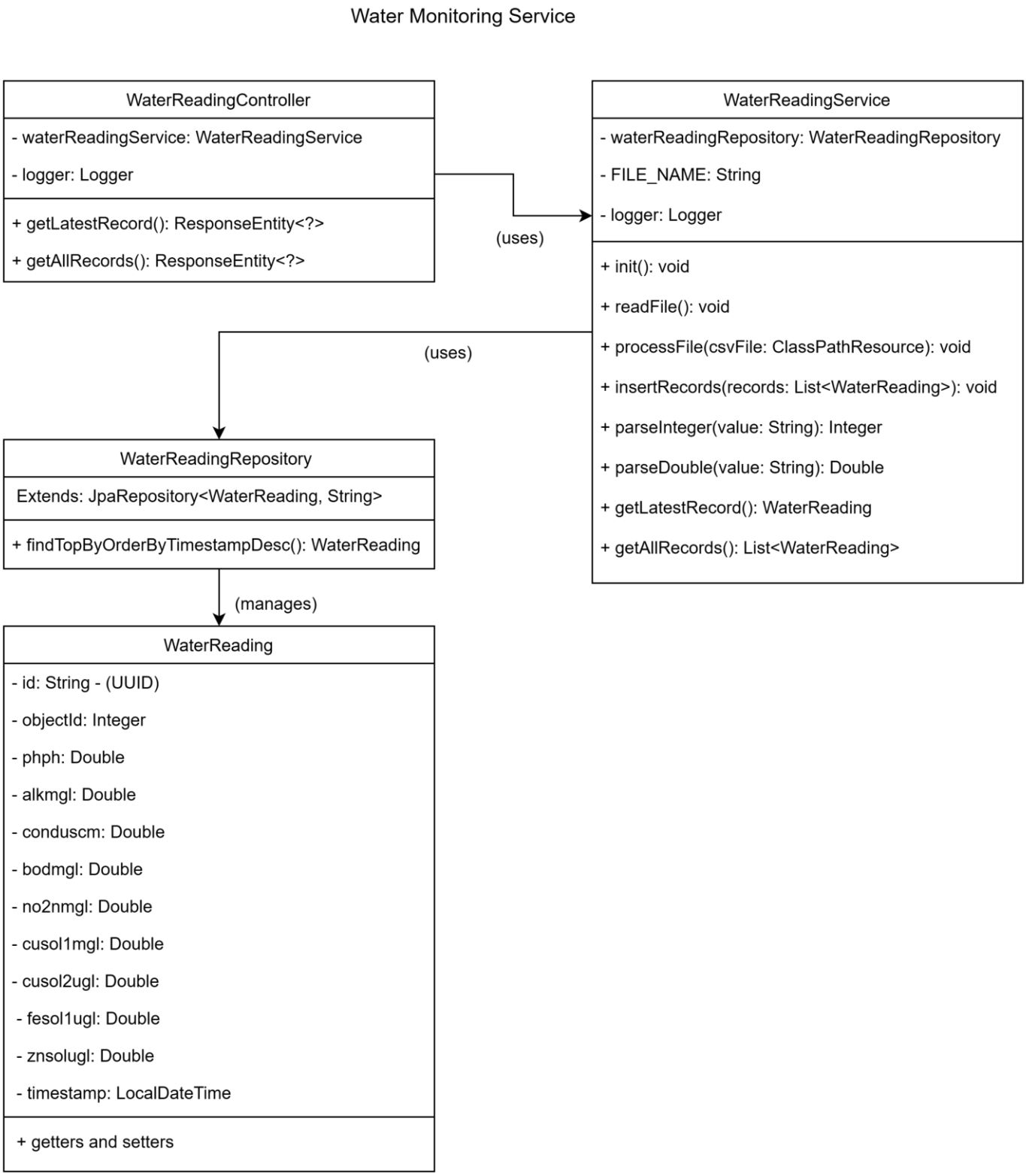
1. Architecture diagram



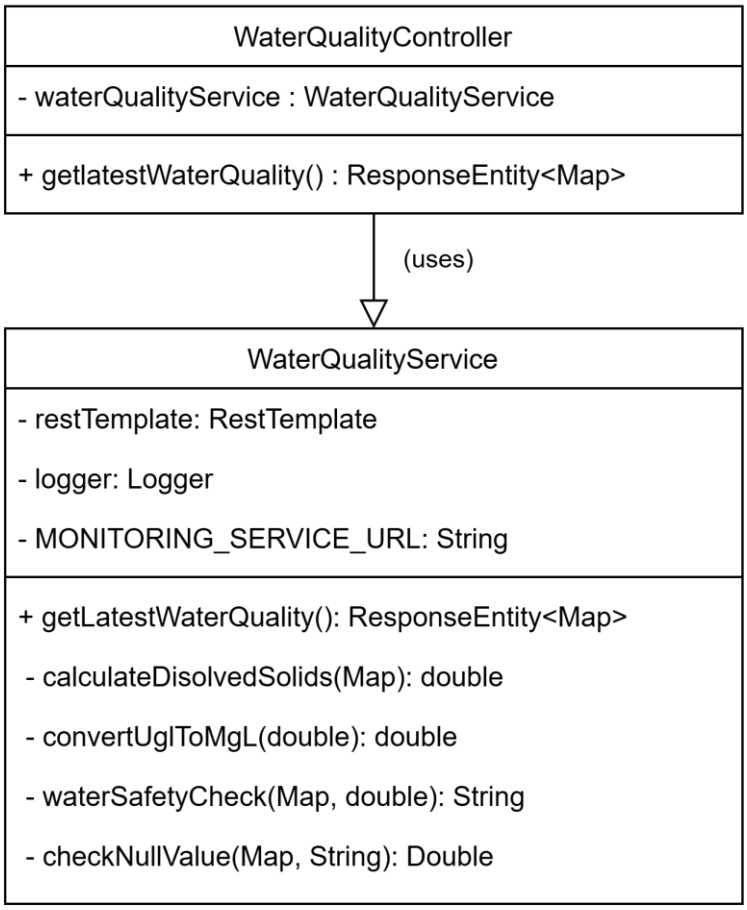
2. Database Schema



3. Class diagram

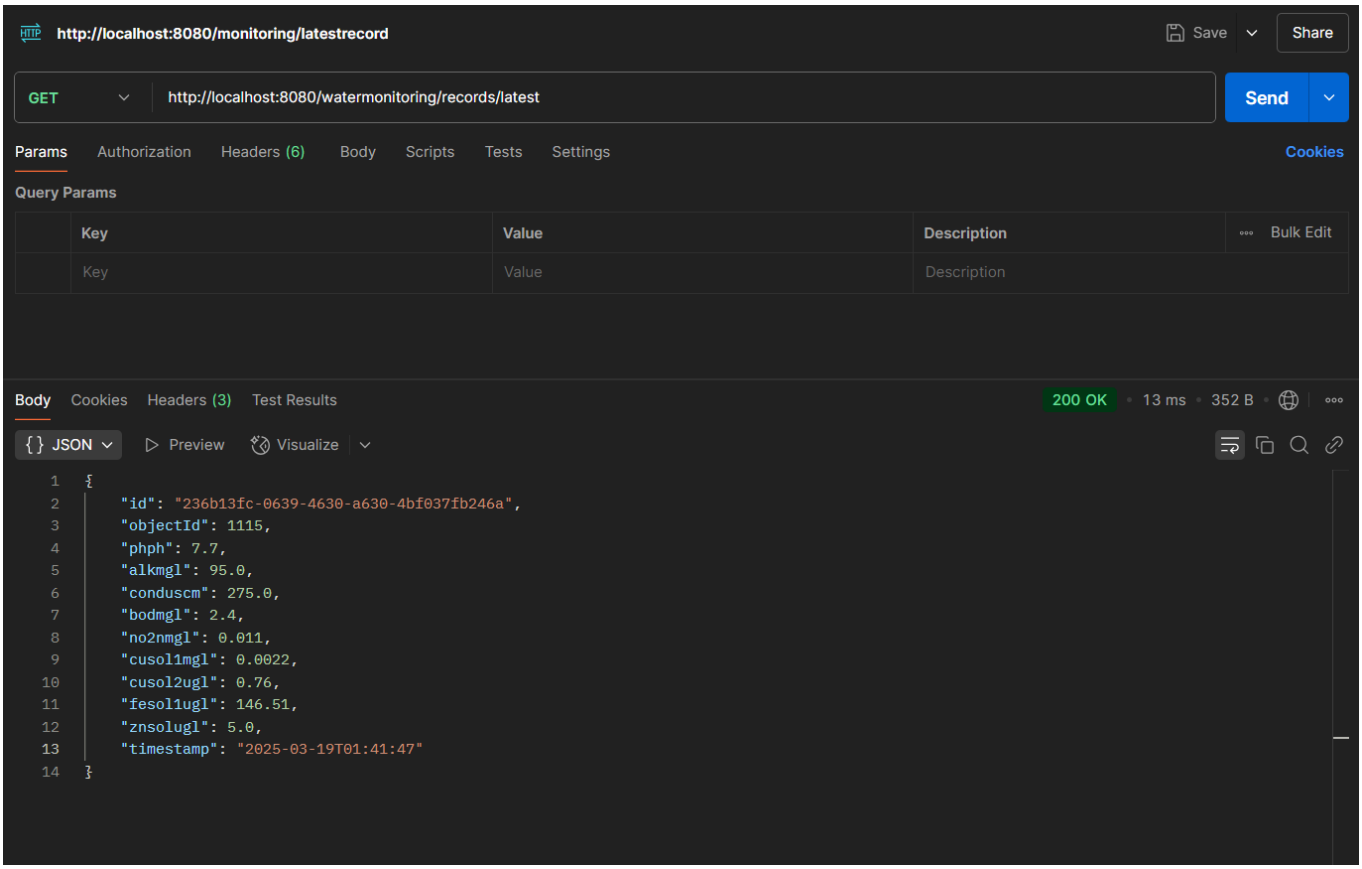


Water Quality Service

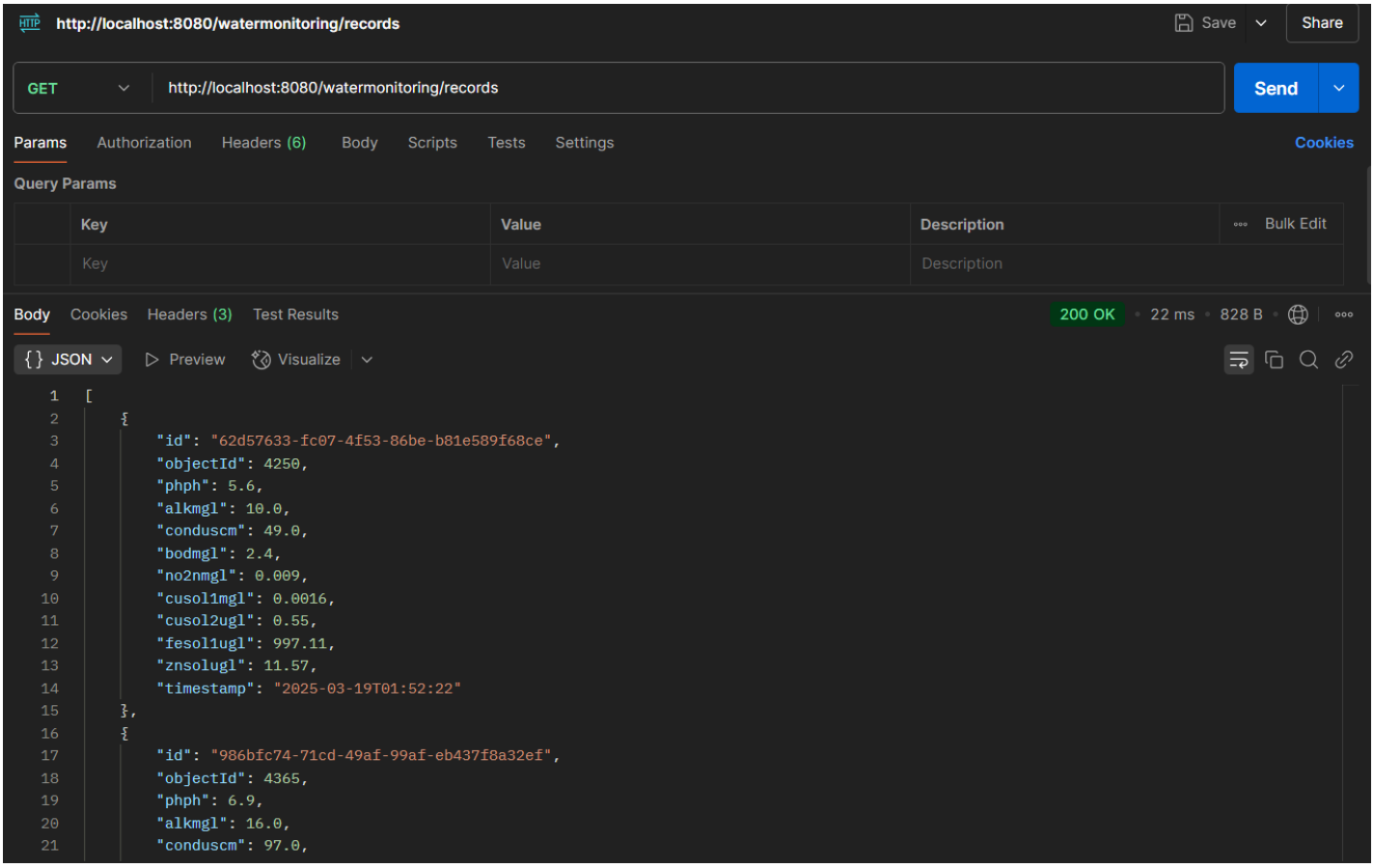


4. APIs documentation

- **Water Monitoring Microservice -**
  - **URI - GET** - http://localhost:8080/watermonitoring/records/latest
    - **Description** - Fetches the latest water record based on the timestamp.
    - **Request Params -** No parameters in request URL.
    - **Response Status Code -**
      - 200 OK HTTP – Successfully retrieved the latest water quality record.
      - 204 No Content - No record found.
      - 500 Internal Server Error - Internal server error.
    - **Response –**

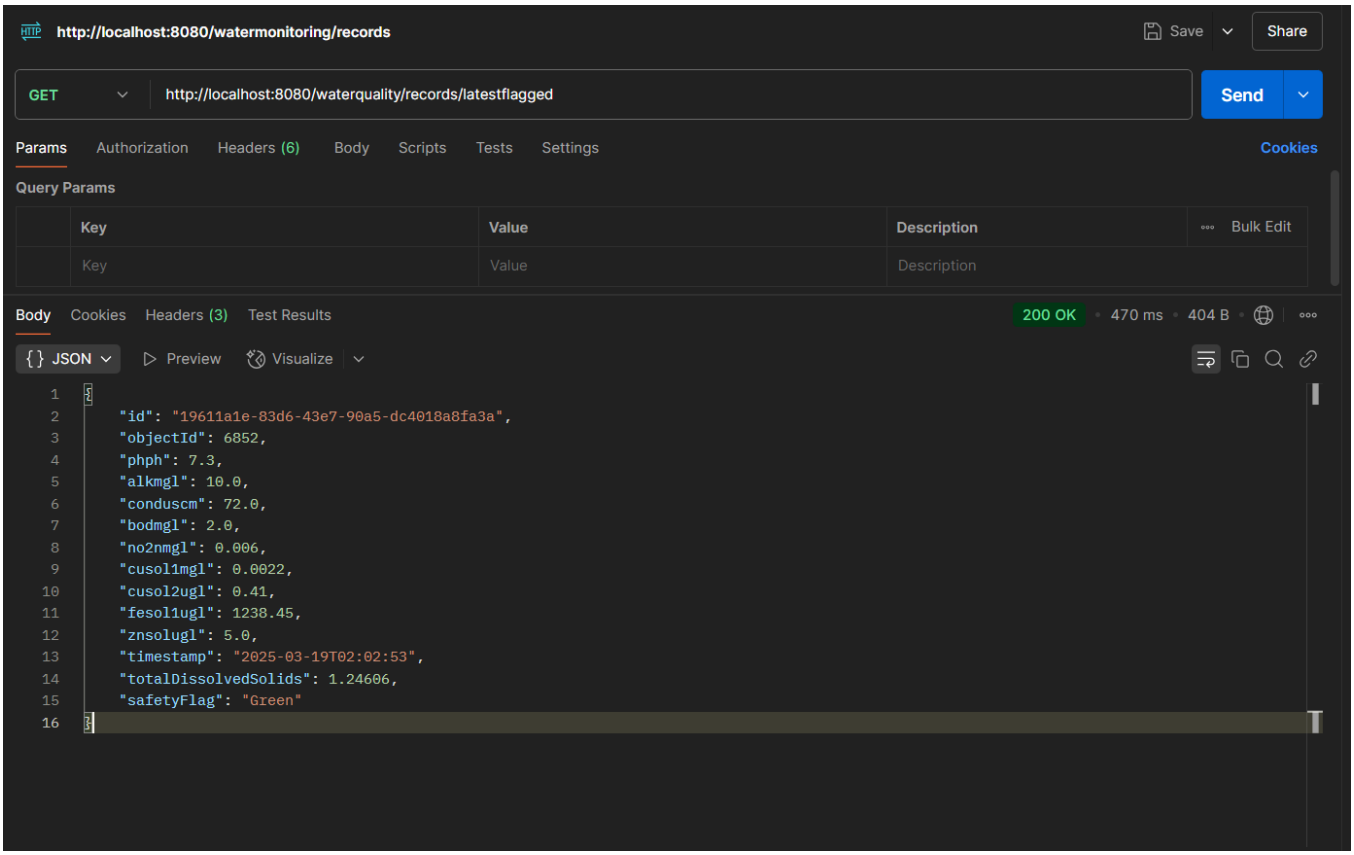


- **URI - GET** - http://localhost:8080/watermonitoring/records
  - **Description** - Fetches all water from the database.
  - **Request Params** – No parameters in the request URL.
  - **Response Codes:**
    - 200 OK HTTP – Successfully retrieved list of all water quality records.
    - 204 No Content - No records found.
    - 500 Internal Server Error - Internal server error.
  - **Response –**

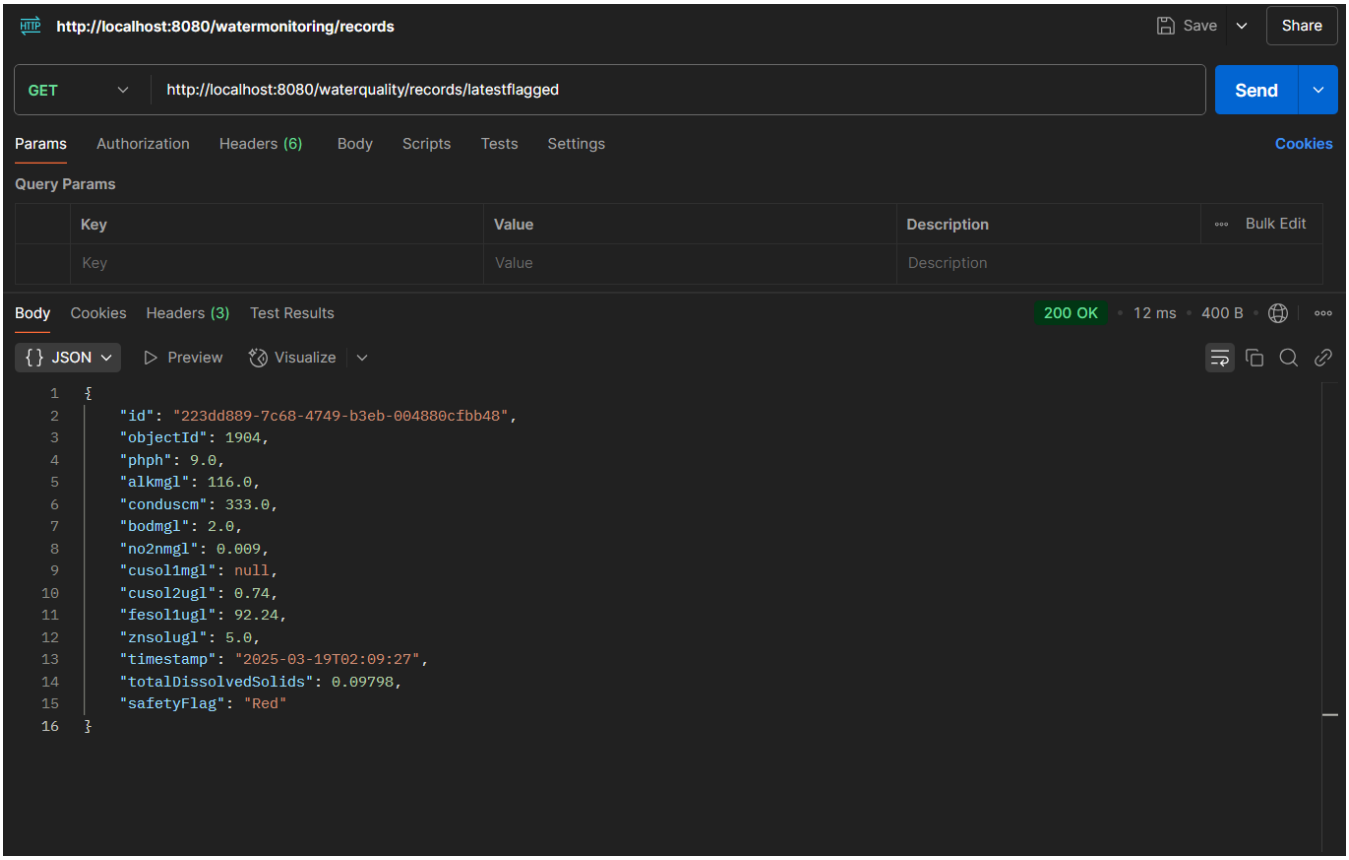


- **Water Quality Microservice –**
  - **URI - GET** - http://localhost:8080/waterquality/records/latestflagged
    - **Description** - This microservice interacts with the water monitoring service to retrieve the latest water quality data, calculates the total dissolved solids and other water quality parameters and sets the safety flag “Green” or “Red” based on WHO standard.
    - **Request Params** – No parameters in request URL.
    - **Response Codes -**
      - 200 OK HTTP – Successfully retrieved the water quality record with safety flag.
      - 204 No Content - No records found.
      - 500 Internal Server Error - Internal server error.
    - **Response** –

(Water safety flag (Green) as all parameters meet the WHO safety standards)



(Water safety flag (Red) as all parameters does not meet the WHO safety standards)



- **Gateway –**
  - **Description -** Manages incoming requests and forwards them to the appropriate microservices if water quality service fails the request is routed to the fallback route
  - /watermonitoring/** - Routes to **Monitoring Service**
  - /waterquality/** - Routes to **Water Quality Service**
  - /fallback – If **Water Quality Service** is not available (Fallback route).

5. Critical evaluation

This Report is for my project which includes Water Monitoring Microservice, Water Quality Service and A Gateway. Unit Testing Is also been done on each microservice to test the service layer the controller and the Repository to ensure the proper functionality of each service.

The Water Monitoring Microservice is responsible for collecting and storing real-time water quality data from a CSV file which acts as an IOT sensor, while saving the data an UUID and a timestamp is generated for each record. While reading the CSV file each record is saved in the database with and interval of 30 seconds and an REST API endpoint is exposed which fetches the latest water quality record from the database based on the timestamp, Additionally I have also provided an endpoint to retrieve all the water quality data in the database.

The Water Quality Service consumes the endpoint provided by the monitoring service and retrieves the Water Records such as Cusol1, Cusol2, Fesol1, and Znsol which are total dissolved solids and the values such as ph, alkalinity, conductivity, nitrite which are compared against the WHO parameters for determining the safety of the water and other saved data such as timestamp and UUID the Water Quality Microservice also provides an endpoint which is used to retrieve the calculated data such as total dissolved solids and the safety flag.

I have implemented a Gateway which helps to redirect the requests for the requested microservice and also a circuit breaker if there is any error with the Water Quality Service the request will be redirected to a fallback Route.

Except the core components of the project I have also implemented Logging feature at various levels of Criticality to check the output and errors in the processes and keep logs for further troubleshooting in a Log File created at the root of project folder. Apart from that I tried to use the Do not repeat yourself (DRY) principle at every stage of my application and followed a class based approach in the microservice and in the testing of those services.

Some areas of improvement would have been better error handling such as making a centralized class for exception handling which could have made the project more readable and maintainable Parsing of CSV file could have been optimised by better handling for more exception.

To conclude, the water quality microservice is very improved by collect real time data which can be inserted via an IOT sensor which will further enhance the features of the application also by making the necessary improvement in my application mentioned could enhance the flow and further minimize the possibilities of errors in the application. Further the application could be deployed and scaled on a cloud platform with a better and enhanced gateway to increase the bandwidth of the requests.

References

Basics of Java:

- https://www.w3schools.com/java/default.asp
- https://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html#newtojava

Documentation:

- https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html
- https://spring.io/projects/spring-boot
- Swagger Documentation

SpringBoot:

- Youtube Videos for SpringBoot REST

ChatGPT:

- Used for for some debugging some error, of which root cause not clear.

University Blackboard:

- Access to course materials, assignments, and project instructions.