Spring Boot Core Components Assignment (Life cycle methods and scopes)

Objective: Understand and apply the fundamental components of Spring Boot, emphasizing its core modules and lifecycle methods without relying on databases or web endpoints.

Requirements:

1. **Setup:**
    - Create a new Spring Boot project using the Spring Initializer or your preferred IDE without any additional dependencies.
2. **Entities:**
    - Create a POJO named **Student** with fields **id**, **name**, and **score**.
3. **Service:**
    - Create a **StudentService** class with methods to:
        - Add a new student (store in an in-memory list or map).
        - Fetch all students.
        - Fetch a student by ID.
        - Update a student's score.
        - Delete a student.
    - Store the students in an in-memory data structure like **List** or **Map**.
4. **Lifecycle Methods:**
    - Implement the **InitializingBean** and **DisposableBean** interfaces in the **StudentService** class.
        - In the **afterPropertiesSet()** method (from **InitializingBean**), log a message indicating that the **StudentService** has been initialized.
        - In the **destroy()** method (from **DisposableBean**), log a message indicating that the **StudentService** is being destroyed.
5. **Scoring System:**
    - Implement a simple scoring system in the **StudentService** where:
        - A student's score can be between 0 and 100.
        - If a score is below 50, log a message indicating the student is "Below Average."
        - If a score is between 50 and 75, log a message indicating the student is "Average."
        - If a score is above 75, log a message indicating the student is "Above Average."
    - These logs should be triggered whenever a student's score is updated.
6. **Main Class:**
    - In the main Spring Boot application class, demonstrate the usage of the **StudentService**:
        - Add a few students.
        - Fetch and display the students.
        - Update scores and observe the scoring system logs.
        - Remove a student.

Submission:

- Submit the source code for the entire project on git share url.

- Provide a README.md with instructions on how to compile and run your application.
- Clearly comment on the code wherever necessary.

Evaluation Criteria:

1. Proper setup and configuration of the Spring Boot project.
2. Adherence to Java and Spring Boot best practices.
3. Correct implementation of the lifecycle methods.
4. Proper implementation and handling of the scoring system.
5. Clarity and quality of the code documentation and README.

Notes:

Ensure that your application can be easily compiled and run by someone reviewing your code. Properly handle any potential errors or edge cases to provide a robust application.