**Evaluation of the Credit Union Member Account Information Application**

**1. Fulfillment of the requirements: Does your application meet the requirements outlined above?**

The application meets all the outlined requirements:

* It is built using Spring Boot 3.
* It interacts with an H2 in-memory database configured in application.yml.
* It manages member account information with the specified columns like Manage credit union member account information (Account ID, First Name, Last Name, Address).
* It exposes a RESTful API with endpoints for all CRUD operations.
* Adhere to Jakarta EE specifications.
* The package structure is feature-based, separating concerns into controller, service, repository, and model packages.

**2. Code quality: Is your code well-structured, easy to read, and maintainable?**

The code is well-structured and easy to read:

* The use of Lombok reduces boilerplate code, making the classes cleaner and more maintainable.
* Each layer (controller, service, repository) has a single responsibility, adhering to the SOLID principles.
* The use of Spring Data JPA simplifies database interactions.
* The feature-based package structure helps in maintaining the code by separating different functionalities.

**3. Error handling: Does your application handle potential errors gracefully?**

Currently Basic error handling is implemented:

* The service layer methods throw a RuntimeException if an account is not found, which is caught and handled by Spring Boot's default exception handling mechanism.
* For a production-ready application, this can be improved by creating custom exception classes and global exception handlers using @ControllerAdvice to provide more meaningful error messages.

**Testing: Have you included tests for your application? Do the tests provide good coverage?**

The application currently lacks explicit tests. For better coverage and evaluation:

* Unit tests should be written for the service layer using JUnit and Mockito.
* Integration tests should be written to test the RESTful endpoints using Spring Boot's test utilities.
* Tests should cover various scenarios, including successful operations and error cases.

**5. Documentation: Is the application and its code properly documented?**

* The project structure, dependencies, and configuration are clearly explained.
* Each component's role (model, repository, service, controller) is described.
* Instructions on how to run the application are provided.
* The REST API endpoints are listed and described.

**Conclusion**

The application fulfills the basic requirements and is well-structured, making it easy to read and maintain. Error handling is basic but sufficient for a simple application. However, to achieve higher code quality, comprehensive testing should be included. The documentation is detailed and provides clear instructions and explanations. Overall, the application meets the essential criteria, but the inclusion of tests and improved error handling would enhance its robustness and reliability.

**Here are two menu, 1) Create Member and 2) Search Member**

1. Create a member can enter account number, first name, last name and address and then submit it. So record will be saved in h2-console database.

A computer screen shot of a computer

Description automatically generated

1. When we click on search member menu then you will see the inputbox so you can search by name like firstname or lastname and you will get the data accordingly.

A computer screen shot

Description automatically generated

A screenshot of a computer

Description automatically generated

Here is Database

Enter this link to browser:- <http://localhost:8080/h2-console>

A screenshot of a computer

Description automatically generated

Restful API

1. Create an Account

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Retrieve an Account by ID

A screenshot of a computer

Description automatically generated

1. Retrieve all account

A screenshot of a computer

Description automatically generated

1. Update an Account

It was Asha and after update from Asha to Rahul it successfully updated Rahul

A screenshot of a computer

Description automatically generated

1. Delete an Account

A screenshot of a computer

Description automatically generated

Delete data successfully for id = 5

And now data available 1 in database

A screenshot of a computer

Description automatically generated

Retrieve all having the same 1 record

A screenshot of a computer

Description automatically generated

Now From UI I am adding more records, after adding record in create member page then come to search member page and able to see below records. Search by Lastname as “Patel” so all Patel records will be appeared.

A screenshot of a computer

Description automatically generated

Let’s see in database now

A screenshot of a computer

Description automatically generated

Search by lastname

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

Test Cases

A screenshot of a computer

Description automatically generated