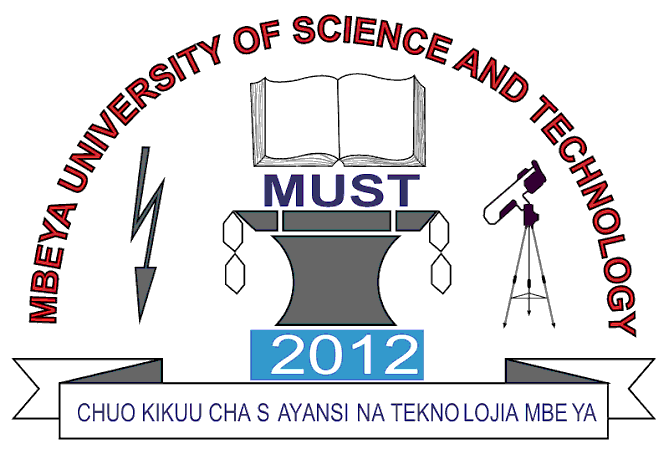
**MBEYA UNIVERSITY OF SCIENCE AND TECHNOLOGY**



**MODULE NAME : SYSTEM INTEGRATION AND DEPLOYMENT**

**MODULE CODE : IT 8209**

**FACILITATOR : SIR NCHIA**

**LEVEL : UQF8 SECOND YEAR SEMISTER II**

**NAME OF STUDENT : MOKINI SIMIONI MKWAMA**

**REGISTRATION NO : 21101133370039**

**INDIVIDUAL ASSIGNMENT**

**QN: What is relationship between service and repository in java EE.**

In Java EE, a service package typically contains business logic that encapsulates specific functionalities of an application, while a repository package deals with persistence of data and database operations.

The service layer acts as an intermediary between the presentation layer (such as the user interface) and the repository layer. It is responsible for processing requests, performing business logic, and coordinating interactions between different repository classes or entities.

On the other hand, the repository layer is responsible for handling the persistence of data to a database. It consists of classes that implement data access operations and interact with the database using technologies like JDBC or JPA.

**Repository layer is implemented to access the database and helps to extend the CRUD operations** on the database. Whereas a service layer consists of the business logic of the application and may use the repository layer to **implement certain logic involving the database**.

The relationship between the service and **repository packages is typically one of dependency**, where the **service layer depends on the repository layer to perform database operations**. The service layer calls the appropriate repository methods to retrieve, update or delete data, and the repository layer provides the necessary data access operations to perform these tasks.

**Repository collection** of different object in a database while service is nothing but help in ad hoc queries

Overall, separating business logic into the service layer and persistence logic into the repository layer provides a clean separation of concerns and allows for better maintainability and extensibility of the application.

Example: The teller can deposit or withdraw from the vault, the teller is the repository. The customer is the one who asks the teller to deposit or withdraw, the customer is the service.