# Module 5 - CRUD using collections

1. Use the front-end application Data
2. Create a collection appropriately using List/Set/Map in Java
   1. Example: Data from tables of Profile, Accounts, Transactions and CheckBookRequest
   2. Create appropriate models with setters and getters.
   3. Hint: Collection with User Defined Objects
3. Perform Read, Update and Delete in the collections.

Hint: use add, addall, get, put, foreach methods

1. Follow package structure and naming convention

**Package Structure**: com.banking -> main method to call service methods.

com.banking.service -> Create Interface object/instance. Call the interface methods and perform implementations

com.banking.models -> Object models with getters and setters

com.banking.intf -> Interface for your implementations

com.banking.impl -> Business logic for Create, read, Update and delete

com.banking.repsoitory -> To return sample data

**Naming convention:**

All Interface class names should end with Intf. All implementation class names should end with Impl. All Repository class names should end with Respository.

E.G:

Com.banking.intf.UserTntf

Com.banking.impl.UserImpl

Com.banking.db.UserRepository

Com.banking.model.User

Com.banking.service.UserService

Com.banking.BankingApplication -> main method

1. Use Test case report and handle any one of the flow neatly with inputs and ouputs
2. Show iteration in test report like fail in first iteration/pass accordingly while testing your application.

**Note: Last Submission Date:** 13 – August - 2024