

***HIBERNATE***

**Training Assignment**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 20/11/2012 |

**Hanoi, 09/2019**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
| 1 | 25/07/2020 | Create assignments | Create | DieuNT1 | VinhNV |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Hibernate Mapping 4](#_Toc53255907)

[Movie Theater Application 5](#_Toc53255908)

[Objectives: 5](#_Toc53255909)

[Technical Requirements: 5](#_Toc53255910)

[Problem Descriptions: 5](#_Toc53255911)

[Questions to answer: 5](#_Toc53255912)

[Unit Testing Requirements: 6](#_Toc53255913)

[Guidelines: 6](#_Toc53255914)

|  |  |
| --- | --- |
|  | **CODE: ORM.M.A101**  **TYPE: MEDIUM**  **LOC: NA**  **DURATION: 180 Minutes** |

# Hibernate Mapping

This assignment requires trainees having Java, Maven, SQL, File and Hibernate knowledge. Technical details are:

* Java 8
* [Hibernate 4.3](https://hibernate.org/orm/releases/4.3/)
* [Maven 3](https://maven.apache.org/download.cgi)
* MS SQL Server

The important note:

* Should **avoid many-to-many** design for entities and have to know the differences between **create** and **update** for the property **hbm2ddl.auto**.
* Java classes must follow naming convention rules (for examples, class names and attribute names must not have \_)
* Tables must have names as required on the design (for examples, Employee table has first\_name column instead of firstname)
* The DAO class should have at least 5 methods as getXxxByID, getAllXxx, updateXxxByID, deleteXxxById, insertXxx where Xxx is the model class of DAO. You could choose other verbs instead of get, update, delete, and insert. However, they should be consistent among DAO classes and meaningful.

Happy coding!

Movie Theater Application

Objectives:

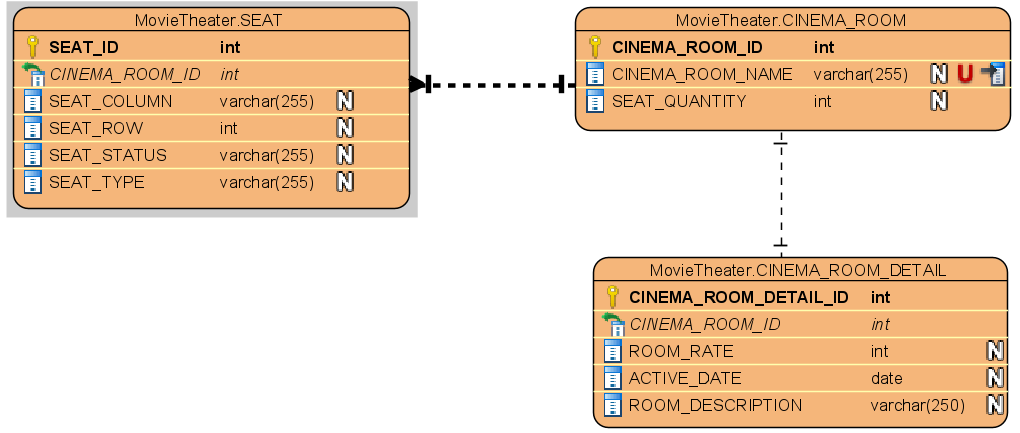
* Understand what is the mapping in Hibernate.
* Know about types of Association mapping.
* Able to use the basic annotations to map Java objects to relational tables.
* Able to implement the association relationship: OneToOne, OneToMany.
* Know how to implement CRUD operations in Hibernate.

Technical Requirements:

* Must use Eclipse to build projects.
* Must use tools with versions mentioned above.

Problem Descriptions:

A theater wants to have an application to manage their movie room as well as seat booking. The project team designed the ERD as below after asseted the requirements and seat booking operations.



A cinema room has a lot of seats, each seat includes a seat column, seat row, seat status ('Available’, 'Not Available', 'Booked' and type ('VIP', 'Normal').

A cinema room has only one detailed information that includes: room rate, active date and description.

After developing CRUD operations for this project, you must simulate one senario of all activities. The database name should be **movietheaterdb**.

Questions to answer:

The trainees need to create a new project structure to use Maven managing libraries, project name as **orm.m.a101**.

Creating a package with name as **fa.training.entities** in this project that contain 3 above entities: SEAT, CINEMA\_ROOM, CINEMA\_ROOM\_DETAIL. You should see new tables by running ‘**SELECT**’ queries after connected to **movietheaterdb** database.

You must create the appropriate DAO classes for the above models (**SetDao**, **RoomDao**, **RoomDetailDao**) to proceed CRUD operations

Unit Testing Requirements:

* You must write scripts to test all of the DAO methods that you have developed.

Guidelines:

*The Date or Time columns, it is recommended to use LocalDate or LocalTime to search for high accuracy.*

**-- THE END --**