

***HIBERNATE***

**Training Assignment**

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

**Hanoi, 09/2020**

RECORD OF CHANGES

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

Contents

[Hibernate Mapping 4](#_Toc53956752)

[Recruitment System 5](#_Toc53956753)

[Objectives: 5](#_Toc53956754)

[Technical Requirements: 5](#_Toc53956755)

[Problem Descriptions: 5](#_Toc53956756)

[Questions to answer: 6](#_Toc53956757)

[Unit Testing Requirements: 6](#_Toc53956758)

[Guidelines: 6](#_Toc53956759)

|  |  |
| --- | --- |
|  | **CODE: ORM.M.A301**  **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!

Recruitment System

Objectives:

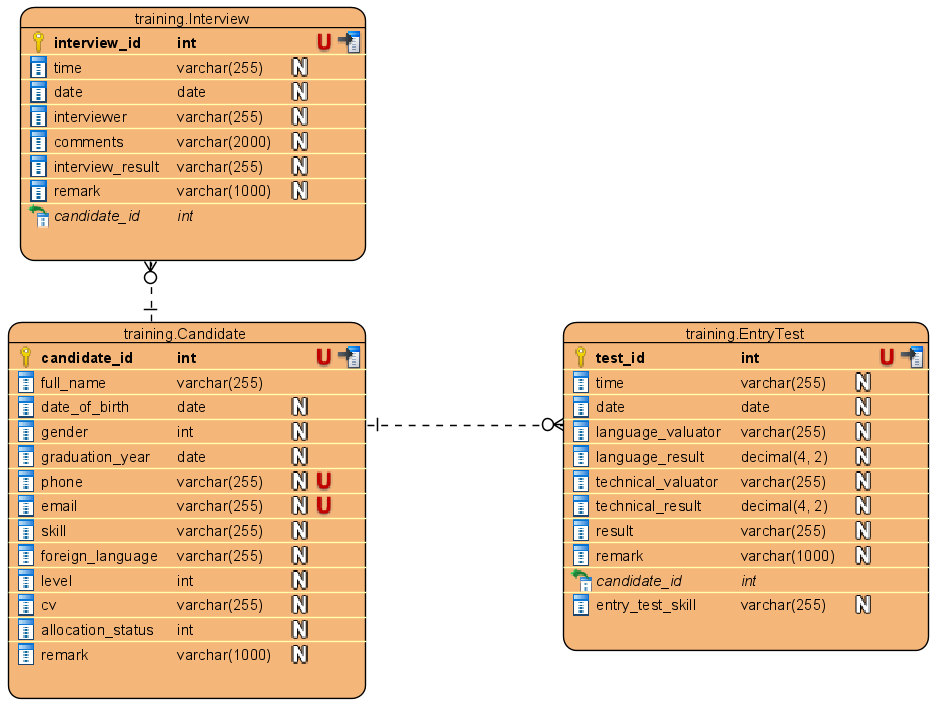
* Understand what is the hibernate criteria query language.
* Understand what is the hibernate validator.
* Able to use the basic of Bean validation annotations to validate user input.
* Able to implement the hibernate criteria to manipulate with database.

Technical Requirements:

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

Problem Descriptions:

A company wants to have an application to manage their recruitment candidates and test/interview results. The project team designed the ERD as below:



Each candidate can apply for one or many positions with different respective skills such as Java, .NET, FrontEnd, Embedded, .. Therefore, the candidates may have no or many recruitment exam and interview results of them.

*Bussiness rules:*

* **Candidate** entity:
  + fullName, dateOfBirth, graduationYear, phone, email: are not null
  + phone, email: are unique
  + gender: accepts value 0 (female) and 1 (male) only.
  + level: skill level of candidate (accepts value range from 1 to 7 only)
* **EntryTest:**
  + languageResult, techinicalResult: accepts value range from 0 to 10 only.
  + result: 'pass' or 'fail' value.

Questions to answer:

The trainees need to create a new project structure to use Maven managing libraries, project name as **orm.m.a301**.

* Creating a package with name as **fa.training.entities** in this project that contain 3 above entities: Candidate, Interview, EntryTest.
* You must create the appropriate DAO classes for the above models (**CandidateDao, InterviewDao**, **EntryTestDao**) to proceed CRUD and some operations as following (using HQL-a,b,c and Criteria-d,e,f)
  1. Find all of the candidate that has skill is 'Angluar’ and skill level is 2.
  2. Find all of the candidate that has foreign language is 'Japanese' and skill is 'Python/ML'.
  3. Find all of the candidate by skill and entry test result (that has skill is ‘Java’ and pass entry test on '1-Oct-2020').
  4. Find all of the candidate that pass interview on '15-Oct-2020'.
  5. Update remark is inactive for candidates who do not have either phone, email and cv.
  6. Create a method to proceed paging operation for Candidate use Hibernate Criteria Query Language.

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 --**