**Task 1. (40 points) Full power of Hibernate**

Implement DAO layer for your pet app or domain model defined with your mentor

You application should cover next features:

1. (4 points) Inheritance: select appropriate strategy for you model.
2. (2 points) Some entities should have One-to-Many relationship.
3. (2 points) Some entities should have Many-to-One relationship.
4. (2 points) Try to involve lazy and eager initialization.
5. (2 points) Map embedded object and enum field.
6. (4 points) DAO layer should support CRUD operation
7. (3 points) DAO should have several methods which select data with help of:
   1. NamedQuery
   2. Criteria API
   3. Native Query
8. (1 points) Enable logging of SQL queries with parameters
9. (4 points) Provide solution for n+1 problem
10. (4 points) Support composite key
11. (2 points) Properly override equals/hashcode
12. (2 points) Generate meta-model and use it with Criteria
13. (2 points) Enable 2nd level cache and use it
14. (4 points) Introduce optimistic lock with versioning and try to involve pessimistic lock
15. (2 points) Try to map value objects: List, Map<String, SomeNotEntityObject>

**Task 2. (30 points) Real project with Hibernate**

Create Spring application with Hibernate or JPA.

List of models:

* Employee
* EmployeeStatus
* Address
* EmployeePersonalInfo
* Project
* Unit

Requirements:

1. EmployeeStatus should be Enum type
2. Employee has attribute ‘external’
3. Address should be embedded to Employee object
4. Relationship between Employee and EmployeePersonalInfo should be one-to-one
5. Relationship between Employee and Project should be many-to-many
6. Relationship between Unit and Employee should be one-to-many

Implement Service API which provides:

1. Take into account the following:
2. Create Employee / Unit / Project
3. Find employee / Unit / Project by id
4. Delete employee / Unit / Project by id
5. Update Employee / Unit / Project by id
6. Add Employee to Unit by id’s
7. Assign Employee for Project by id’s
8. Get Projects with Employees which are not external

**Task 3. (40 points) Books viewer DAO with Spring Projects**

Create web-application with Hibernate or JPA

Your application should cover next features:

1. CRUD operation on Book entities
2. Uploading Book files (.epub, .fb2, .pdf)
3. Books list view with pagination
4. Create required views (as JSP or HTML with AJAX as you wish)
5. Summary email is asynchronously sent to configured recipient with latest uploaded books
6. Use docker and spring boot to deliver the project.
7. Implement data-provider layer with "spring-data-jpa"
8. Implement REST API with "spring-data-rest"

{

"caseData":

{ "submittedDate":"2020-06-24T01:31:15.890Z", "filedDate":"2020-06-24T01:31:15.890Z", "caseClassificationID":"1000102", "foreignTransfer":false, "security1":false, "security2":false, "security3":false, "security4":false, "security5":false, "locationID":"1000001" }

,

"caseParticipants":[

{

"statusDate":"2020-06-24T12:31:15.890Z",

"startDate":"2020-06-24T12:31:15.890Z",

"proSe":false,

"participantStatusTypeID":"10001",

"participantSubTypeID":"1000001",

"caseParticipantRef":"newDriverParticipant",

"address":

{ "line1":"123 Test St", "city":"SomeCity", "zipCode":"19999", "international":false, "addressType":"21", "countryType":"561", "regionType":"1000011", "scopeID":"10000" }

,

"actor":{

"actorCategoryID":"1",

"scopeID":"10000",

"actorTypeDetails":

{ "actorTypeID":"1", "tribalAffiliation":false }

,

"firstName":"John",

"lastName":"Doe"

},

"security1":false,

"security2":false,

"security3":false,

"security4":false,

"security5":false

}

],

"charges":[{

"security1":true,

"security2":true,

"security3":true,

"security4":true,

"security5":true,

"chargeCount": 0,

"chargeDetail":

{ "chargeComment": "string", "chargeCount": 0, "chargeType": "50146", "statuteDetailID": "2", "offenseStartDate":"2020-06-24T04:00:00.000+0000" }

,

"chargeableParticipantRef":"newDriverParticipant",

"chargeRef":"charge"

}

]

}

When user is trying to save more than one case classification record on the configuration screen there should be some kind of validation I mean errorMessage. Do you have any preferences about this message?