CS5220 - 2 Lab 3. REST API with Spring Boot Description

1. Follow Step 1 and 2 of REST API Using Spring Boot to create a Spring Boot project.

2. Add the three model classes, Department, Search, and User, from Lab 2 to the project. Use JPA annotations to annotate the model classes to map them to relational tables. Run the project once to create the tables in the database, then populate the tables with the following data:

- Four users: John, Jane, Tom, and Kate. You may assume user names are unique.
- One department Computer Science with chairperson John.
- One search conducted by the Computer Science department for the position Assistant Professor. The search committee chair is Jane, and the search committee has one member Tom.

Note that you may add additional constructors, methods, and/or properties to the model classes, but you may not change the existing properties.

3. Add SearchDao.java to the project (feel free to change the packages), and create a `SearchDaoImpl.java` that implements the interface.

4. Design and implement REST API for the following operations (all data exchanges use JSON):

(a) Get the searches conducted by a department. The request will include department id. The response should include search id, position, the id and name of the department, and the id and name of the committee chair and the committee members.

(b) Create a search. The request will include department id, position, the id of the committee chair, and the id's of the committee members. The response should include the id of the created search.

(c) Add a user to a search committee. The request will include the id of the user and the id of the search. The response body can be empty (the status code is sufficient to indicate whether the operation is successful).

(d) Remove a user from a search committee. The request will include the id of the user and the id of the search. The response body can be empty (the status code is sufficient to indicate whether the operation is successful).

5. Use Postman to test the REST API with following test cases. Note that I use the names of the departments/users/searches in the test case, but you should replace them with the corresponding id's which are required by the API.

(a) Get the searches conducted by the Computer Science Department.

(b) Create a search with the following information:

- Department: Computer Science
- Position: Associate Professor
- Committee Chair: Tom
- Committee Members: Jane and Kate

(c) Add Kate to the search committee for the Assistant Professor position

(d) Remove Tom from the search committee for the Assistant Professor position

6. Create a zip file of the project and upload it to CSNS. Put all test requests in Postman in a collection, export the collection, then upload the exported collection (i.e. a JSON file) to CSNS.