**LAB 3**

This lab you will continue your previous lab that contains the User and Post and entities. You will be adding another entity with similar functionalities based on the following:

public class Comment {  
  
 long id;  
 String name;  
  
}

1. Make a domain model using the domain model above.
2. Make a repository and database that will hold the Comments.
3. Make a service layer that will inject the repository.
4. Make an ORM relation between the Post and Comment entities, where a Post holds a collection of Comments. Create it using JoinColumn
5. Make it possible to add a comment that will be associated with its post.
6. Cascade all the operations from the User to the Posts and Posts to the comments. So, if you delete a User, it should delete all related data. (Delete a User to verify)
7. Make a query that will retrieve all the users that have more than (n) posts.
8. Make a query that will find all the posts that match a given title.

**Optional**

1. Make a query that will find the users that made posts within a given title
2. Make it possible to navigate from your user to a post, then to a comment like the following example: localhost:8080/api/v1/users/111/posts/1/comments/1

After hitting this URI, it should give us the comment with **id 1** in the post with **id 1** for the user with **id 111**

Make sure you execute all these requests using postman and follow best practices and conventions..