# # Subject

This 5-session TD aims at developing a Kanban Board, mainly with the entities of Developers and Tasks.

Students will have to develop all layers of the application :

- domain layer with JPA Entities

- data layer with repositories

- service layer

- controller layer exposing a REST API

- front-end layer, displaying screens and accessing data via the REST API

During sessions 1 and 2, students will work on domain, data, and service layers.

During session 3, students will work on controller layer.

During sessions 4 and 5, students will work on front-end layer.

## ## Back-end stack

Back-end is structured around SpringBoot, and bootsrapped by Spring Initializr.

Domain layer will be implemented with JPA Entities.

Data layer will be implemented with JPA Repositories.

Service layer will be implemented with Spring Services.

Controller layer will be implemented with Spring Rest Controllers.

Unit tests will be implemented with JUnit and SpringBoot Test infrastructure.

Maven will be used as build tool for the back-end.

## ## Front-end stack

Front-end will be bult using VueJs and Vue-CLI bootsrapper.

# # Input

Project will be launched without any input at session 1.

At session 3, a Maven project with functional service, data and domain layer will be provided, to provide appropriate layers for implementing controllers.

At session 4, a functional back-end will be provided, to provide appropriate layers for implementing front-end.

# # Output

The expected deliverable will be Git repo or a zip of your source code, with all your source files.

Send the URL of the Git repo or the zip by mail to your teacher : remy.girodon@gmail.com

Please send it at the end of session 2, then at the end of session 3, then at the end of session 5.

# # Steps

## ## Step 1

Initialize a SpringBoot Maven project with Spring Initializr, with JPA and H2 (in-memory database) as dependencies.

Import this Maven project into Eclipse.

## ## Step 2

Create classes Developer, Task, TaskType, TaskStatus, and ChangeLog in domain layer, as JPA entities.

Following class diagram shows relations between these entities :

Diagram

Description automatically generated

Create a repository for each of these classes.

Create a CommandLine Runner to populate the db before unit tests.

Write a unit test for method addDeveloper of class Task.

Write a unit test for method findAll of DeveloperRepository.

Write a unit test for method findAll, save of TaskRepository.

## ## Step 3

Create DeveloperService and TaskService interface and implementations.

DeveloperService provides methods with following signatures :

- public List<Developer> findAllDevelopers();

TaskService provides methods with following signatures :

- public Collection<Task> findAllTasks();

- public Task findTask(Long id);

- public Task moveRightTask(Task task);

- public Task moveLeftTask(Task task);

Maybe you will need additional methods.

Write unit tests for all services methods.