Projektarbete - Spring Data JPA

Information
Projektarbetets syfte är att validera era kunskaper inom JPA/Spring Data JPA.
Laborationen är betygsgrundande och har två nivåer.
Nivå **Godkänd** görs i grupp (3 - 5st) och nivå **Väl Godkänd** görs enskilt.
Nedan hittar du en beskrivning av vad som ska ingå för att uppnå respektive nivå.

Beskrivning
Under denna laboration ska ni konstruera ett pragmatiskt ärendehanteringssystem. Ni kommer att bygga det lager som har han om lagring av data.

Systemet ska tillåta en användare att lägga upp ärenden som sedan kan prioriteras och hanteras. I projektarbetet ingår att i grupp ta reda på hur ett sådant system fungerar. Utmärkta exempel på sådana system kan hittas här: https://www.pivotaltracker.com och http://leankit.com/

Generella krav
Projektet ska vara ett Maven-projekt och hantera alla sina externa beroenden m.h.a. Maven.
Projektet använder uteslutande SpringData (och SpringDataJPA) för att hantera datalagring.
Koden **MÅSTE** vara formaterad korrekt och får **INTE** innehålla några System.out (förutom i en Main-klass om ni har en sådan). All funktionalitet ska exponeras via en eller flera Service-klasser. Dessa Service-klasser kommer att innehålla er affärslogik. Ni kommer alltså ha en liknande arkitektur som webb-shoppen. I övrigt **MÅSTE** koden vara väl strukturerad och följa en bra objektorienterad design.

Funktionskrav - Godkänd
User *- en användare i systemet som tillhör ett team*

Funktioner:
- ~~Skapa en User~~
- ~~Uppdatera en User~~
- ~~Inaktivera en User~~
- ~~Hämta en User baserat på user id/number (inte entity id)~~
- ~~Söka efter en User baserat på förnamn *och/eller* efternamn *och/eller* användarnamn~~
- ~~Hämta alla User som ingår i ett visst team~~

Affärslogik:
- ~~En User måste ha ett användarnamn som är minst 10 tecken långt~~
- ~~När en User inaktivares ändras status på alla dennes WorkItem till *Unstarted*~~

Team *- en gruppering av User*

Funktioner:
- ~~Skapa ett team~~
- ~~Uppdatera ett team~~
- ~~Inaktivera ett team~~
- ~~Hämta alla team~~
- ~~Lägga till en User till ett team~~

Affärslogik:
- ~~Det får max vara 10 users i ett team~~
- ~~En User kan bara ingå i ett team åt gången~~

Work item - *ett ärende som tilldelas en User*

Funktioner:
- ~~Skapa en work item~~
- ~~Ändra status på en work item [Unstarted, Started, Done]~~
- ~~Ta bort* en work item~~
- ~~Tilldela en work item till en User~~
- ~~Hämta alla work item baserat på status~~
- ~~Hämta alla work item för ett Team~~
- ~~Hämta alla work item för en User~~
- ~~Söka efter work item som innehåller en viss text i sin beskrivning~~

Affärslogik:
- ~~En WorkItem kan inte tilldelas en User som inte är aktiv~~
- ~~En User kan max ha 5 WorkItems samtidigt~~

Issue - en anmärkning som kan ges en work item när den inte accepteras

- Skapa en Issue och lägga till den till en work item
- Uppdatera en Issue
- Hämta alla work item som har en Issue

Affärslogik:
- En Issue ska bara kunna läggas till work item som har status Done
- När en Issue läggs till en work item ändras status för work item till Unstarted

Exakt vilka datamedlemmar som varje entitet har samt hur dessa hänger ihop är upp till er att bestämma. Det finns dock några givna datamedlemmar som behöver finnas utifrån de funktionskrav som finns. Det är tillåtet att ha fler entiteter än dessa om ni anser att ni behöver det.

** När ni tar bort en entitet behöver ni fundera på hur detta ska påverka eventuellt relaterade entiteter, dvs, vilken/vilka cascade type(s) ska användas*

Funktionskrav - Väl Godkänd

Förutom nivå Godkänd ska även följande funktionalitet finnas:

- Du ska på egen hand ta reda på hur Auditing i SpringData fungera och implementera detta i ditt datalager på passande ställen. Du bestämmer själv var.
  Se: http://docs.spring.io/spring-data/jpa/docs/current/reference/html/#auditing (stycke 4.8 + 4.9)

- Implementera paging för följande funktionalitet:
  - Hämta alla users
  - Hämta alla work items
  - Hämta alla Issues

- Implementera en funktion för att få reda på history över vilka work items som klarades av under en viss period. Det ska exempelvis gå att fråga efter alla work items mellan perioden 2015-08-01 till 2015-08-30 som har status 'Done'

Redovisning

Redovisning sker i grupp **fredagen den 19/2 klockan 09:30** för nivå Godkänd och enskilt **fredagen den 19/2 klockan 13:00** för nivå Väl Godkänd. **Notera:** alla i gruppen måste vara med och redovisa för att bli godkänd.

Lycka till,

Anders och Pablo