

Projektarbete - Mönster, Databaser och Arkitektur

Information

Projektarbetets syfte är att validera era kunskaper inom databaser, JDBC och enhetstester. 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 detta projektarbete ska ni konstruera ett pragmatiskt ärendehanteringssystem. Ni kommer att bygga det lager som har hand om lagring av data samt det lager som har hand om affärslogiken (business logic).

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 använder uteslutande JDBC 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 MovieService och dess repositories. 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
- 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 inaktiveras 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 till: Unstarted, Started eller 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

Affärslogik:

- En WorkItem kan inte tilldelas en User som är inaktiv
- En User kan max ha 5 WorkItems samtidigt

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

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

Affärslogik:

- Ett 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 något (exempelvis en WorkItem) behöver ni fundera på hur detta ska påverka eventuellt relaterad data*

Funktionskrav - Väl Godkänd

Förutom nivå Godkänd ska även följande funktionalitet finnas:

- Dina service-klasser ska testas. Eftersom dina service-klasser har beroenden till repositories kommer dessa behöva mockas. Du behöver med andra ord **inte** använda dig av någon databas.

Redovisning

Redovisning sker i grupp **måndag den 10/10 klockan 09:30** för nivå Godkänd och enskilt **fredagen den 14/10 klockan 09:30** för nivå Väl Godkänd. **Notera:** alla i gruppen måste vara med och redovisa för att bli godkänd.

Lycka till,

Anders