

Projektmunka

A feladat csapatonként egy adott nyilvántartó rendszer létrehozása. A feladat megoldása során a törekedjete a tanult eszközök széleskörű felhasználására. A szembejövő feladatokat az arra alkalmas technológiai elemek segítségével oldjátok meg. A megoldás során használt összes form kitöltött adatainak érvényességét minden esetben ellenőrizni kell mind kliens (form-validation), mind server (bean-validation) oldalon. A nem érvényes adatokból származó nem várt működést, kivételekkel és kivételkezelő mechanizmusokkal kezeljétek le, amik hibaüzenetek formájában jelenjenek meg a felületeken.

1. Felhasználók

Az oldalon legyen lehetőség regisztrálni és bejelentkezni. A rendszerbe regisztrált felhasználókat az Admin jogkörrel rendelkező felhasználók láthatják és kezelhetik, egy erre a célra létrehozott admin oldalon. Ezen az oldalon legyen lehetősége az adminoknak a regisztrált felhasználókat különböző csoportokba rendelni, illetve onnan kiszedni őket. Fontos, hogy az admin oldalt csak az admin jogkörrel rendelkező felhasználók érhetik el. Egy felhasználót egy másik, admin jogkörrel rendelkező felhasználó nevezhet ki adminnak. Adminról csak admin vonhatja vissza a jogkört. Önmagát senki nem nevezheti ki adminnak. Legalább 1 admin felhasználónak mindenképpen lennie kell a rendszerben minden esetben.

Minden felhasználó rendelkezik egy profil oldallal, ahol böngészheti és szerkesztheti a saját adatait.

2, Törzsadatok rögzítése

A rendszerben szereplő adatokat egy admin oldal segítségével kell kezelni. A törzsadatok rögzítésére az alábbi felületeket valósítsátok meg:

- Táblázatszerű, oldalakra bontott, szűrhető lista
- Adat rögzítő form
- Adat szerkesztő form.

Táblázatszerű, oldalakra bontott, szűrhető lista

Ezen az oldalon a rögzített adatok jelennek meg egy táblázatban. Az egy oldalon megjelenő elemek száma (10, 50, 100, 200) a felületről állítható és felhasználóként különböző lehet. Legyen lehetőség szöveges alapú keresésre. A felhasználó által meghatározott szöveg szűrőként hat a megjelenített elemekre. Minden elemhez tartozzon egy szerkesztés és egy törlés hivatkozás. A törlés kérdezze meg, hogy valóban szeretnénk -e törölni a kijelölt elemet.

A szerkesztés navigáljon át az adat szerkesztő felületre (az url-ben jelenjen meg az adat azonosítója). Továbbá helyezzünk el egy "Hozzáadás" hivatkozást az oldalon, amire kattintva eljuthatunk arra a felületre, ahol új adatot tudunk rögzíteni.

Adat rögzítő form

Ezen az oldalon legyen lehetőség új adatot létrehozni. Minden a felhasználó által megadott adatot ellenőrizzetek és jelenítsetek meg értelmes hibaüzeneteket, ha valamelyik bevitt adat nem érvényes. A sikeres mentés után navigáljatok vissza a listához.

Adat szerkesztő form

Ezen az oldalon egy azonosító alapján meghatározott elemet lehessen szerkeszteni. Sikeres mentés esetén egy üzenetben értesítsétek a felhasználót a mentésről. Sikertelen mentés esetén egy hibaüzenettel jelezzétek, hogy mi miatt nem lehetett lementeni az adatokat.

Az adatok közötti kapcsolatokat is ezen az oldalon kell kezelni. Amennyiben egy adat egy, vagy több adathoz is kapcsolódik, akkor legyen lehetőség itt, (vagy ennek az oldalnak egy aloldalán) a további adatok összerendelésére, illetve szétválasztására.

3. Összehasonlítás

Bejelentkezett felhasználóknak legyen lehetősége egy adott típusú fő adat példányait összehasonlításra kijelölni és a kijelölt elemeket egymás mellett megjelenítve összehasonlítani.

4. Rendszerspecifikus lekérdezések

Minden rendszer tartalmaz 4+1 rendszer-specifikus lekérdezést. Ezek megvalósításához hozzatok létre aloldalakat, ahol a felhasználó meg tudja adni a lekérdezéshez szükséges paraméterek értékeit. Az eredményeket listában jelenítsétek meg, ahonnan könnyen át lehet navigálni az adott adat oldalára.

5. Interceptor használata

Minden EJB metódus hívásra ki kell logolni legalább az EJB osztály nevét és a meghívott metódus nevét

6. Messaging használata

Minden projekthez használni kell 2 topicot. A topicokra küldi a fő alkalmazás az üzeneteket a **[jms]** tag-gel megjelölt entitás létrehozásáról valamint törléséről (külön topicok, az egész entitást el kell küldeni rá). Ezekre a topicokra egy másik webalkalmazással lehet feliratkozni a következő módon:

- Az alkalmazásban kiválasztjuk, hogy entitás létrehozására vagy törlésére szeretnénk feliratkozni (melyik topicra)
 - o Létrehozás topicra feliratkozva a felületen be kell tudni állítani a két **JMS-property** értékét (pl text field, combo boksx segítségével), ami alapján a megfelelő selectorral iratkozik fel a topicra az alkalmazás. (Pl minden olyan terméket kapjak meg a topicról, aminek price adattagja kisebb mint 250 és típusa monitor).
 - o Törlés esetén az összes üzenetre (selector nélkül) kell feliratkozni, MessageDrivenBean használatával
- Feliratkozást követően az alkalmazás navigáljon át egy táblázatos nézetre, ahol egy refresh gomb is megtalálható
- A refresh gombra kattintva az alkalmazás jelenítse meg a topicról megkapott üzeneteket

Üzenetben nem egyszerű sztring-et, hanem egy osztály objektumát kell elküldeni, ezért érdemes a két alkalmazás által közösen használt osztálynak egy külön projektet csinálni, ami mindkét alkalmazásnak függősége (dependency), így mindkét alkalmazás pontosan ugyanazt az osztályt fogja látni és használni.

A JMS-entitást és a hozzájuk tartozó 2 propertyt a következő módon jelöljük: **[JMS:property1,property2]**

7 Oldal látogatottsága

Legyen egy olyan komponens, ami számolja, hogy adott napon összesen hány látogatás történt az oldalon és ennek értékét jelenítsük meg a főoldalon.

PROJEKT- és FELADATKEZELŐ RENDSZER

Megjegyzés: Egyes adatkapcsolati, vagy adattárolási módok eltérhetnek a lentebb leírtaktól. Az alábbi modell kiegészíthető a feladat megoldása érdekében.

A rendszer az alábbi adatokat kezeli

- Projekt (id, név, ügyfél id, határidő, kezdési dátum, állapot, leírás, projekt vezető id) **[fő adat]**

- Ügyfél (id, név, telefonszám, email cím)
- Feladat (id, projekt id, prioritás, felelős id, szülő feladat id, típus, státusz) [JMS:projekt név, típus]
- Felhasználó (id, név, bejelentkezési név, jelszó, születési év, beosztás, kép)
- Fórum (id, téma, projekt id, tulajdonos id)
- Jegyzet (id, dátum, tartalom, szülő jegyzet id)
 - Komment (feladat id)
 - Fórum bejegyzés (fórum id)

Az adatok közötti kapcsolatok:

- Egy Ügyfélnek több projektje lehet.
- Egy projektnek több feladat lehet.
- Egy feladatnak egy felelőse van
- Egy felhasználó több projekten dolgozhat, és egy projekten több felhasználó dolgozhat.
- Egy projektnek több fóruma lehet.
- Egy fórumhoz több bejegyzés tartozhat.

Projekt specifikus feladatok

Legyen minden felhasználónak (bejelentkezés után) egy "értesítések" oldala/menüpontja, ahol listában látja az értesítéseit a "feladat"-okról, amiknek ő a felelőse. A következő feladateseményekről kapjon értesítést a felhasználó: feladat felelősének beállítása/átállítása (ha ő az új felelős, vagy ő volt a régi felelős). A feladathoz új komment érkezett. Az értesítések részről közvetlenül át lehessen navigálni ahhoz a feladathoz, amiről az adott értesítés szól. (Az értesítéseknek nem kell live-nak lennie, tehát az oldal frissítésekor, menüpont újbóli betöltésekor kell megjelenennie az értesítéseknek, nem facebook-szerűen)

A projektek összehasonlításánál jelenjen meg (az alap adatokon túl), hogy melyik projektnek hány be nem fejezett feladata van még.

Projekthez tartozó lekérdezések

- Adott felhasználó fórum bejegyzései projekt szerint.
- Adott felhasználóhoz tartozó top prioritású feladatok neve és határideje.
- Azok a feladatok, melynek alfeladatai között van "blokkolt" állapotú.
- Adott projekthez tartozó nyitott feladatok.
- Plusz egy a csapat által definiált lekérdezés (legalább 1 where meg egy join).

