Szoftverarchitektúrák

12. Sorozat portál  
(követelmény specifikáció)

Kiss Zoltán (BUS1FJ)  
Balázs Zoltán (X0ELSN)

# A feladat szövege

A cél egy olyan szoftver tervezése és implementálása, amely lehetőséget biztosít sorozatokkal kapcsolatos információk megosztására (sorozat fórum).

A regisztrált felhasználók új sorozatokat, új epizódokat adhatnak meg. Egy sorozat esetén meg kell adni a nevét, rendezőjét, főbb szereplőket, kulcsszavakat, amiről az adott sorozat szól, és egy rövid, szöveges leírást a sorozatról. Egy sorozathoz több epizód tartózhat, az epizód esetén is elérhető a címe, szereplői, kulcsszavai, szöveges leírása, hossza. Az epizódhoz feliratot is fel lehet tölteni, amit a regisztrált felhasználók le is tölthetnek.

A regisztrált felhasználók értékelhetik (1-10) a sorozatokat, és az egyes epizódokat is, megjegyzéseket fűzhetnek hozzájuk és „lájkolhatják” az egyes részeket, vagy az egész sorozatot.

A portál keresési lehetőséget biztosít a felhasználóknak, hogy keressenek cím, leírás, kulcsszavak, színész vagy rendező alapján. A legjobbra értékelt sorozatokat egy összefoglaló oldalon tekinthetik meg a felhasználók. A hozzászólásokat az adminisztrátornak jóvá kell hagynia, hogy azok megjelenhessenek az oldalon.

# Rendszer áttekintés

## Architektúra

A rendszer klasszikus kliens-szerver architektúrát valósít meg.

* A felhasználók a böngészőn keresztül érik el a szerveren futó webalkalmazást.
* Ez a webalkalmazás a szerver EJB-in keresztül vezérli a rendszert.
* Az EJB-k túlnyomórészt adat perzisztálásával vagy lekérdezésével kapcsolatos funkcióit delegálja egy DAO rétegnek
* A DAO réteg végzi a JPA lekérdezések futtatását
* A JPA réteg végzi az adatok tényleges perzisztálását az adatbázisba

## Biztonság

## Felhasználói interfészek

A felhasználók a rendszert webes felületen keresztül érhetik el.

## Szoftver interfészek

A szoftver interfészek azokat az interfészeket takarják, amik a felhasználó által rejtve maradtak, mégis meghatározó szerepük van a funkcionalitásban. Ilyen interfészek például a Java EE által nyújtott szolgáltatások (pl.: EJB container, Java Persistence API).

A szoftveres interfészt a szerver szolgáltatásaihoz EJB-k biztosítják.

# Felhasznált technológiák

## Java EE 5

A biztos alapokon nyugvó Java Standard Edition (Java SE) platformra épülő Java Enterprise Edition (Java EE) ipari szabványként funkcionál a vállalati rendszerek fejlesztésében.

A Java EE különböző technológiák, specifikációk halmaza, amelyek együttesen lecsökkentik a többrétegű, szervercentrikus alkalmazások fejlesztési költségeit és a fejlesztéshez szükséges időt.

### Java Persistence API

A rendszer a működéséhez szükséges információkat adatbázisban tárolja, az adatbázis elérését pedig nem közvetlenül végzi, hanem egy perzisztens réteg, a Java Persistence API (JPA) közbeiktatásával, entitások használatával.

A Java Persiszence API ORM (Object-Relational Mapping) perzisztencia modellt biztosít az adatbázis eléréséhez. Ez azt jelenti, hogy az adatbázis tábláit egyszerű Java objektumok (POJO, Plain Old Java Object) segítségével érjük el, melyek annotálva vannak, és a JPA az ezekben tárolt információt rögzíti az adatbázisba.

Használata azért előnyös, mert segítségével az egyes adatbázisok közötti különbségek rejtve maradnak, ezáltal a kód hordozhazó lesz. Az egyik legelterjedtebben használt JPA implementáció a Hibernate, amit sok Java EE szerverhez mellékelve adnak.

Annak egyszerűsége és kis erőforrás igénye miatt HSQLDB-t használunk, JPA implementációnak pedig a Hibernate-et választottuk.

### Enterprise Java Beans

Az Enterprise JavaBeans (EJB) specifikáció tranzaktált, elosztott és biztonságos szerveroldali üzleti komponensek létrehozását teszi lehetővé rövid idő alatt.

Ideális választás akkor, ha távoli gépről kell üzleti logikákat elérni és ha skálázható megoldást szeretnénk.

### Java Servlet Technology

A Java Servlet Technology egyszerű, konzisztens mechanizmust biztosít a fejlesztőknek egy web-szerver funkcionalitásának kibővítéséhez, valamint már meglévő üzleti rendszerek eléréséhez.

Ezt a technológiát csak érintőlegesen fogjuk használni arra, hogy a GWT számára biztosítson egy web-container-t.

### JBoss

A programot futtató Java EE alkalmazásszerver a JBoss 5.1.0 GA verziója. Konfigurálása elsőre nehézkesnek hat, mert rengeteg XML-t kell beállítani hozzá, de a fejlesztés gyorsan megy vele, az úgynevezett hot-deploy funkciója miatt: a telepítendő szerverkomponenseket elég egy könyvtárba bemásolni és azokat automatikusan elindítja.

Nyílt forráskódú, jól támogatott alkalmazásszerver.

## GWT

Webportálok fejlesztésére kiválóan alkalmas a Google által fejlesztett Google Web Toolkit. Hatalmas előnye, hogy olyan módon teszi lehetővé a webalkalmazások fejlesztését, hogy a funkcionalitás Java nyelven implementálható, a kinézet pedig a webfejlesztők körében már megszokott CSS-ekkel alakítható. Alapelve, hogy a Java nyelven írt kódot lefordítja a böngészők által is támogatott JavaScript-re. Azzal, hogy a fejlesztés Java nyelven történik, a kód egyben robosztusabbá is válik.

Rendkívül jól használható Eclipse plugin készült a GWT-hez, ami a fejlesztést még problémamentesebbé teszi. Ennek segítségével a kliens oldali kód is debug-olhatóvá válik, rengeteg időt és energiát megspórolva ezzel a fejlesztőknek.

## Fejlesztőkörnyezet

### Eclipse

A kódot Eclipse 3.5 használatával fejlesztjük. Nagyon jól használható, elterjedt fejlesztőeszköz, rengeteg fejlesztést segítő funkcióval és kiegészítővel.

### Ant

A kód fejlesztése során a monoton ismétlődő feladatokat Ant 1.7.0 segítségével automatizáljuk. Ilyenek például a kód fordítása és release-elése, szerverre telepíthető formába hozása.

### Hibernate Tools

Az üzleti logika része az adatok bizonyos feltételek szerinti prezentálása, lekérdezése, kinyerése. Ehhez különböző JPA lekérdezéseket kell írni, amiket tesztelni kell.

A Hibernate Tools egy Eclipse plugin, amivel JPA lekérdezések futtathatóak, valamint JPA specifikus feladatok futtathatóak, pl a séma exportálása adatbázisba az entitások alapján.

# Funkcionalitás

## Használati eset diagram

## Használati esetek

# Követelmények kódokkal

## Hálózat

|  |  |
| --- | --- |
| **Sorszám** | **Követelmény leírása** |
| SP-1 | A kliensek és a szerver között IP hálózatot kell kiépíteni |

## Szoftver interfészek

|  |  |
| --- | --- |
| **Sorszám** | **Követelmény leírása** |
| SP-2 | A rendszer szolgáltatásai API-n keresztül elérhetőek, ezek az interfészei a rendszer EJB-inek |

## Szerver funkciók

|  |  |
| --- | --- |
| **Sorszám** | **Követelmény leírása** |
| SP-3 | A rendszerbe a új felhasználó regisztrálható |
| SP-4 | A rendszer funkcióit csak regisztrált felhasználók érhetik el, belépés után |
| SP-5 | Új sorozat vihető fel a rendszerbe |
| SP-6 | Új epizód vihető fel egy sorozathoz |
| SP-7 | Sorozathoz, epizódhoz címke rendelhető |
| SP-8 | Sorozat, epizód like-olható |
| SP-9 | Epizódhoz megadhatóak a szereplői |
| SP-10 | Epizódhoz feliratok tölthetőek fel |
| SP-11 | A feltöltött feliratok letölthetőek |
| SP-12 | Sorozat, epizód értékelhető (1-10) |
| SP-13 | Sorozathoz, epizódhoz megjegyzéseket lehet adni |
| SP-14 | A megjegyzések csak adminisztrátori jóváhagyás után jelennek meg az oldalon |
| SP-15 | Sorozatok, epizódok kereshetőek cím, leírás, kulcsszavak, színész, rendező alapján |
| SP-16 | A legjobbra értékelt sorozatok egy összefoglaló oldalon megtekinthetők |

## Web portál funkciók

|  |  |
| --- | --- |
| **Sorszám** | **Követelmény leírása** |
| SP-17 | A szerver funkciói elérhetőek a webes felületen keresztül, miután egy felhasználó sikeresen bejelentkezett |