

**Laborfeladatok a Rendszerintegráció tárgyhoz**

Célok.....	2
A modell .....	3
Központi autentikáció .....	3
Az alkalmazások .....	3
Kidolgozandó feladatok.....	4
KA API és alkalmazás .....	5
“Intranet” alkalmazások.....	5
Alkalmazások listája .....	5

## **Célok**

A labor célja, hogy a hallgatók ill. hallgatói csoportok egyszerűbb, de demonstratív programok segítségével részletesebben megismerjék az előadáson elhangzottakat. A tárgy hangsúlyos részei az alábbiak voltak:

- elosztott rendszerek, Grid alapfogalmak
- elektronikus levelezés
- Internet/intranet
- Web technológiák, benne
  - cgi
  - servlet
  - JSP
  - CSS
- adatbázis és web kapcsolata
- XML.

A labormunka 2-3 fős csoportokban folyik. A félév során minden csoportnak két házi feladatot kell megoldani, és dokumentálni. Az egyik feladat az elosztott párhuzamos rendszerek használatát és programozását hivatott bemutatni.

A másik feladat egy webes alkalmazás elkészítése. Minden webes alkalmazás jelszavas elérésű. Az alkalmazások önálló, általában önmagában is megálló, működőképes rendszerek, kivéve azt, hogy az autentikáció központi megoldású. (Lásd később!)

A munkák kidolgozásánál nemcsak a programozáson van a hangsúly, de a koncepciózus kialakítás is feltétlen követelmény. Bonyolultabb feladatok esetén az előadáson kiadott anyagokra lehet hivatkozni.

A megoldásoknál az adatbázis célszerűen MySQL és csak olyan SQL akciókat kell használni, melyek MySQL-lel elkészíthetők. A munkához minden csoport saját MySQL-t installál, és abban önálló adatbázisokat hoz majd létre.

A hallgatói csoportok önálló web- ill. alkalmazáserverrel dolgoznak. Ezek Apache-Tomcat 4.x ill. 5.x (servlet/JSP megoldás esetén), de lehet használni Orion 2.x alkalmazáservert is. A szerverek installációját önállóan el kell tudni végezni.

A programozási nyelv Java; a hangsúly a Java-s, JSP-t használó alkalmazásokon van. A servlet-ek mellett feltétlenül szükséges különböző hatásugarú bean-ek előírásos alkalmazása is.

A web oldalakon kerülni kell a böngésző-specifikus megoldásokat. Az oldalakon CSS használata kötelező! Javascript-et nem célszerű használni, de használata megengedett.

Az adatbáziskezelések JDBC-vel történjenek.

A web-oldalaknak nem kell túl komplikáltaknak lenniük, de a szövegszerkesztős rendszerfejlesztés helyett Eclipse fejlesztői környezetet és benne Lombos JSP-

plugint kell használni. Az Eclipse és a Lomboz használatáról bemutatóval egybekötött ismertetést adunk.

### **A modell**

A laborfeladatok egy modell alapján határozhatók meg ill. oszthatók ki a hallgatói csoportoknak. Ez a modell

- általános
- könnyen bővíthető
- koncepciózus
- modern rendszerelképzelésen alapul.

A labormodell a **vállalati intranetek** egy egyszerűsített modellje. Ennek lényege, hogy az intranetben központi autentikáció és alkalmazásfüggő autorizáció van. Ez az alábbiakat jelenti:

#### Központi autentikáció

A központi autentikáció egy autentikációs adatbázison alapul. Ebben van bent a vállalati intranet felhasználó azonosítója és aktuálisan érvényes jelszava. A felhasználó bármely jelszavas intranet alkalmazáshoz ezzel az (egyetlen) azonosítóval és jelszóval tud csak belépni. A jelszó adott periódusonként lejár, ezért azt időnként meg kell újítani.

Ebbe az adatbázisba akkor kerül be új bejegyzés, ha egy új munkatárs belép ahhoz a vállalathoz, amelyiknek az intranetjéről szó van, és akkor törlődik egy bejegyzés, amikor egy munkatárs kilép onnan.

#### A Központi Autentikáció (KA) általános szolgáltatásai

- megmondja, hogy érvényes-e egy adott azonosító-jelszó páros
- jelszócserét engedélyez (adott szabályok szerint)
- a hozzáférésekről naplót vezet
- extra szolgáltatásként az elfogadás esetén figyelmeztet, hogy hány nap van vissza a lejáratig.

Speciális szolgáltatás az új felhasználó felvétele ill. az élő – valamilyen okkal való - törlése.

#### Az alkalmazások

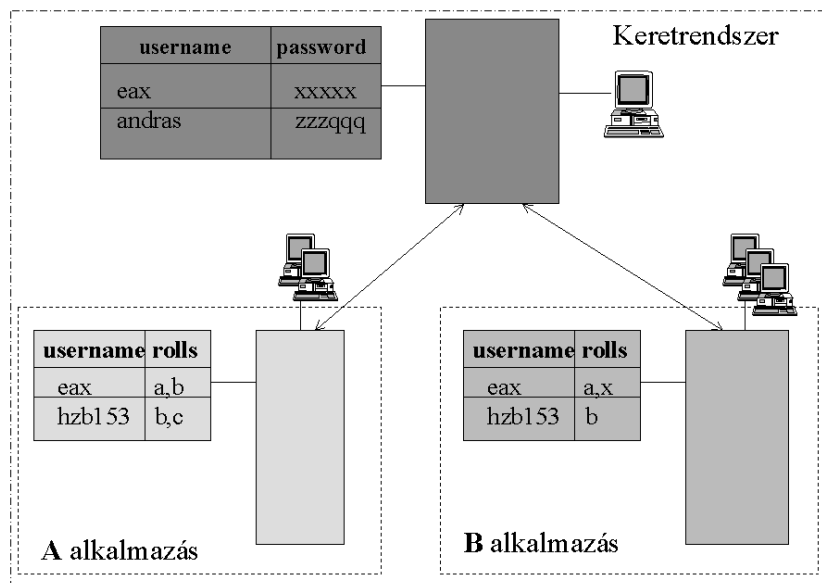
Az alkalmazások úgy működnek, hogy mindegyik először a központi autentikációhoz fordul, hogy megnézze az azonosító-jelszó páros érvényességét. Ha érvényes a belépési kísérlet, akkor saját, azaz a konkrét alkalmazással – esetleg alkalmazáscsoporttal kapcsolatos - alkalmazásfüggő autorizációs adatbázisából megnézi, hogy az adott felhasználó milyen jogokkal rendelkezik, és ennek megfelelő szolgáltatást engedélyez a belépő felhasználónak.

Ha a jelszó nem érvényes, akkor két megoldás lehetséges:

- az alkalmazás egyszerűen visszaírja a felhasználónak, hogy lépjen be a jelszóváltó alkalmazásba. (Azaz ekkor egy külön jelszóváltó alkalmazást is kell készíteni!)
- lehetővé teszi a jelszóváltást, ami kicsivel bonyolultabb megoldást eredményez.

A sikeres belépés után tehát elkezdődik a tényleges „intranet” munka, ami igen különféle lehet. Fő jellemzője, hogy adatbázis akciókkal kapcsolatos, ami egyáltalán nem különleges a mai, modern rendszerek esetében.

A rendszer elvét jól mutatja az alábbi ábra.



### **Kidolgozandó feladatok**

A fent bemutatott keretbe lehet beilleszteni a Célok szekció által megfogalmazott feltételeknek megfelelő alkalmazásokat. Olyan alkalmazást kell készíteni tehát, amely

- központi autentikáción alapul
- az alkalmazás „intranet” jellegű,
  - saját autorizációs adatbázist használ,
  - a felhasználók jogosultságai alapján az alkalmazás különféle funkciói érhetők el,
  - az alkalmazás saját ügyviteli adatbázisán operál.

Az egyszerűsítés érdekében a központi autentikációs rendszert szimuláljuk egy külön adatbázissal és az arra épülő szolgáltatásokkal. Ez gyakorlatilag azt jelenti, hogy minden alkalmazást minimum két elkülönült adatbázison dolgozik, mivel az autentikációs adatbázis és az ügyviteli adatbázist el kell egymástól választani

egymástól. Célszerű az autorizációs adatokat is külön kezelni, de ennek az ügyviteli adatbázisba történő bevonása megengedett.

#### KA API és alkalmazás

Elsőként a központi autentikációs (KA) rendszert kell kialakítani. A háttérben egy viszonylag egyszerű adatbázis van, bár a feladatot a valóságban igen összetett módon oldható csak meg.

Az API segítségével a program lekérdezheti a belépő felhasználó jelszavának érvényességét, képes egy felhasználó saját jelszavát átírni. A jelszóváltáshoz „policy” is tartozik; pl. egy nap csak max. három jelszóváltást lehet elvégezni, továbbá egy jelszót újra csak akkor szabad használni, ha n darab másik eltérő volt előtte. (Pl. n legyen 7!)

Az adatbázis kapcsolatot JDBC alapon kell megoldani, és a jelszóváltást megfelelő biztonsággal kell kialakítani.

Ennek az alkalmazásnak a kidolgozása – időhiány miatt - elhagyható, de az autentikációs adatbázis használata kötelező!

#### “Intranet” alkalmazások

Az alkalmazás beléptetéssel kezdődik. Sikeres belépés után az autorizációellenőrzésre kerül sor, majd az autorizációs opciók alapján megállapítja a program, hogy a felhasználónak milyen jogosultságai vannak.

Minden alkalmazásnak legalább két jogosultsági csoportot kell megkülönböztetnie. Az egyik a közönséges felhasználó, aki az intranet szolgáltatásokat igénybe veszi, a másik pedig az alkalmazás-adminisztrátor, aki valamilyen adminisztrációs feladat megoldása érdekében speciális jogosultságokkal rendelkezik.

Az alkalmazások különféle típusúak lehetnek, azonban mindenképpen adatbázist kell használniuk. Ez az adatbázis lehet

- SQL típusú adatbázis vagy
- „XML adatbázis”.

Az adatbázis kapcsolatot itt is JDBC alapon kell megoldani, erősen javasoljuk a datasource-típusú használatot.

Ezt az alkalmazást – a fent említett különféle jogokkal rendelkező felhasználók esetére – mindenképpen ki kell dolgozni.

#### **Alkalmazások listája**

A konkrét alkalmazások az alábbi listából választhatók. Az egyes feladatokhoz különféle külső specifikációk tartoznak. A csoportok a listából választanak, majd a választás után rövid rendszertervben leírják, hogy konkrétan hogyan kívánják az alkalmazást kidolgozni.

	Alkalmazás neve	Leírása	Normál ügyfél	Adminisztráció
--	-----------------	---------	---------------	----------------

1.	Hírszerver	Intranet híreket lehet beadni, melyet a vállalat dolgozói olvashatnak. A hírszöveg mellett web-referencia is beadható.	Különbéféle szintű hírek olvasása, pozícióhoz kapcsolt jogosultságtól függően.	Hírek szerkesztése, törlése.
2.	Szerződés kezelés	Vállalati szerződés adatbázis menedzselése.	Szerződés adatok változtatása, lekérdezése, csoportosítása.	Új szerződés bevitele, régi archiválása.
3.	Tárgyalás-adminisztráció	Tárgyalók lefoglalása, munkatársak bejelentkezése.	Lekérdezések, bejelentkezések.	Termék beállítása, törlése.
4.	Adatbázis feltöltése	Upload jelleggel fix méretű rekordokból álló file-ok felolvasása. Elemzés után adatbázisba beírás.	Feltöltés, sikeres feltöltés után visszajelzés.	Naplók olvasása, adatbázis kiolvasása.
5.	Adatbázis feltöltése excel file-ból	Upload jelleggel excel file-okból adatok felolvasása (POI package segítségével). Elemzés után adatbázisba beírás.	Feltöltés, sikeres feltöltés után visszajelzés.	Naplók olvasása, adatbázis kiolvasása.
6.	Adatbázisból kiolvasás excel file-ba	Adatbázisból kiolvasás, majd excel file generálása (POI package segítségével). Letöltés adminisztrálása	Kiolvasás, visszajelzés (adott session-ben csak egyszer lehet kiolvasni).	Naplók olvasása, adatbázis feltöltése, aktualizálása.
7.	Fióklista kezelése.	Vállalati fióklista rendszer kezelése. A fiókok magasabb szintű profitcentrumokba csoportosíthatók.	Lekérdezések, keresések.	Új fiók beszúrása, egy élő kiszedése.
8.	Település-irányítószám	Hazai települések neve és irányítószámai közötti kapcsolat.	Lekérdezések, ellenőrzések.	Adattábla feltöltése, módosítása.
9.	Díjszámítás	Casco díjszámító alkalmazás.	Díjszámítás elvégzése.	Gépjármű adatbázis feltöltése, módosítása.
10.	Ügyféladatbázis	Ügyfelek adminisztrációja	Ügyfél beléptetése, kiléptetése.	Naplók olvasása, összesítő kimutatások.
11.	KA adatbázis kezelése	Munkatársak adminisztrációja	Munkatársak beléptetése, kiléptetése. Kezdeti jelszó beállítása.	Naplózások olvasása, összesítő kimutatások.

Budapest, 2004.09.15.

Ercsényi A.