

SZAKDOLGOZAT

Bánszki András

Debrecen

2010

Debreceni Egyetem
Informatikai Kar

Levezős órarend tervezése és publikálása

Témavezető:
Aszalós László
Egyetemi Adjunktus

Készítette:
Bánszki András
PTI-BSc

Debrecen
2010

Tartalomjegyzék:

1. Bevezetés
2. Követelmények
 - 2.1 Követelmények felderítése
 - 2.2 Felmerülő problémák egy órarend összeállításánál
 - 2.3 Követelmények meghatározása
3. Az adatbázis
 - 3.1 Az adatbázis megtervezése
 - 3.2 Adatok tárolása
 - 3.3 Az adatbázis leírása
4. Program tervezése
 - 4.1 Felhasználói Csoportok
 - 4.2 Használati esetek
 - 4.3 Képernyő tervek
5. Implementációs kérdések
 - 5.1 SQLite adatbázis használata:
 - 5.2 Az Implementáció:
6. Felhasználói dokumentáció
7. Továbbfejlesztési lehetőségek
8. Összefoglalás
9. Irodalomjegyzék

1. Bevezetés

Szakedolgozat témája:

Szakedolgozatom témája a levelezős hallgatók órarendjének megtervezése.

A levelezősök órái általában csütörtök, péntek, szombati napokra esik, ezek közül csütörtök-pénteken rendszerint délután, a szombati pedig egész napos elfoglaltságot kíván, ugyanis a levelezősök nagytöbbsége munka mellett folytatja tanulmányait. Egy-egy tárgyból csak három-négy előadásuk van egy félévben, amik több órák is lehetnek, így fontos hogy ne maradjon el az előadások, gyakorlatok megtartása.

Természetesen tisztában vannak az egyetemeken, hogy hogyan is épül fel egy levelezős órarend, mikor vannak a levelezős órák, gyakorlatok, azt viszont lehet, hogy nem mindenki tudja, mekkora munka van a levelezősök órarendjének összeállításával. A levelezősök órarendjének sajátosságainak kezelése megkívánna egy célalkalmazást, amivel egységesen lehetne kezelni az órarendet érintő változtatásokat, és az oktatók is központosítva jelezhetnék, ha bármi igényük támadna.

A feladat meghatározása:

Szakedolgozatom témája hogy bemutassam, mikre kell odafigyelnie az órarendszerkesztőknek, mikor összeállítják a levelezősök órarendjét, milyen határok közé vannak szorítva, milyen igényeket kell, hogy kielégítsenek, és hogy hogyan működhetnek együtt az órarendszerkesztők az oktatókkal, mindezt egy PHP nyelven írt, SQLite adatbázist használó webes alkalmazás használatával.

Ezen kívül fontos, hogy az alkalmazás biztosítson egy felületet a diákok számára, hogy időben tájékozódhassanak az órarendjükéről. A feladat nem igazán igényelte egy korszerű alkalmazás-generátor felhasználását, a PHP technológiával való alapos megismerkedés céljából próbáltam minden apróságot saját magam megoldani, a kész megoldások felhasználása helyett. Természetesen egyes segítségüket igénybe vettem, mint például a jQuery programkönyvtárét, és pár kiegészítését, mellyel a külsín professzionálissá alakítható.

2. Követelmények

2.1 Követelmények feltárása

Konzultáció a témavezetővel:

Az alkalmazással szembeni elvárásokat a témavezetőm segítségével tártam fel. Többszöri személyes megbeszélés alapján pontosítottam az alkalmazás felé támasztott követelményeket.

Az alkalmazás fő feladata az órarendet összeállítók munkájának a segítése. Az órarendszerkesztők feladata a féléves órarend összeállítása, összehangolása a levelezős oktatásra beosztható oktatókkal, és a szabadon felhasználható termek listájával. A szükséges adatokat, a levelezős oktatásban résztvevő oktatók, igénybe vehető termek, és az adott félévben tartandó tárgyak listáit egy adatbázisban kell tárolni. Ezeket az órarendszerkesztők töltik fel az adatbázisba, és ők is végzik az adatok karbantartását, kiegészülve egy adminisztrátorral.

Az órarend összeállításánál kisebb szerepet kapnak az alkalmazás során az oktatók. Nekik csak minimális funkciókat kell biztosítani, mint például igényelhesse nek nagyobb termet, vagy éppen előre jelezhessék, ha valami miatt nem tudnak bemenni egy már, meghirdetett előadás/gyakorlati órára.

2.2 Felmerülő problémák egy órarend összeállításánál:

Egy órarend összeállításakor sok mindenre kell odafigyelni: a rendelkezésre álló szabad termek, ráérő oktatókon kívül, például közbejöhethetnek különféle oktatói elfoglaltságok, amelyeket figyelembe kell venni az alkalmazás tervezésekor. Ezeket tételesen felsorolom az alábbiakban.

Szabad termek korlátjai:

Az egyetem erőforrásai – mint minden más – végesek. A levelezősök ugyanott tanulnak, és ugyanazok az egyetemi tanárok oktatják őket, mint a nappalisokat, mégse lehet összevonni a nappalisok óráival, mert más ütemben haladnak, ám mégis ugyanazok az oklevél követelmények.

Korlátozott a levelezős oktatásra igénybe vehető termek száma; elsőbbséget élveznek a nappali tagozatos előadások, gyakorlatok. Ha egy termet már lefoglaltak nappalis oktatás céljából, az akkor nem használható levelezős oktatásra ugyanabban az időben. Szükséges, hogy az alkalmazásban valahogy jelezni lehessen, mely termek vehetők igénybe illetve melyek nem.

Az oktatók száma és ideje véges:

Az oktató se lehet egyszerre több helyen: ha nappalis órát tart, akkor abban az időpontban nem számolhatunk vele az órarend összeállításakor. Az alkalmazásnak ezt is kezelnie kell, ha egy oktató nem ér rá.

Egyéb oktatói elfoglaltságok:

Az oktató megbetegedhet, vagy lehetnek más, váratlan elfoglaltságai is. Ebben az esetben találni kell egy helyettesítő oktatót, aki ráér az adott időpontban, ami nehezen megoldható. Az órarend átszervezése gyakran könnyebben járható út. Az oktatóknak lehetnek még olyan visszajelzései is, hogy például nagyobb termet kíván, helyettesítést is kérhet, mivel sürgős dolga akadhat, el kell utazni, stb. Ez nem csak a félév eleji órarend összeállítást határozza meg, hanem később is lehetőséget kell biztosítani, az oktatói igények jelzésére, azok kezelésére. Ha lehetséges, az igényeket elfogadhatják az órarendszerkesztők, illetve felülbírálnak.

Félévek kezelése:

Mikor odáig jut az alkalmazás használata, hogy másik tanítási félévbe lépünk át, az alkalmazással tudatni kell, hogy adott félévben mely oktatókra számíthatunk, és melyekre nem az előzőekhez képest. A rendszer folyamatosan legyen képes az éppen elérhető oktatók, termek, szabad időpontok kijelezésére.

Szakok eltérő tematikája:

A különböző szakokhoz tartozó tantárgyakat meg lehessen különböztetni a többi szakhoz tartozó tantárgyaktól, lehetséges legyen azok előadásait is elválasztani egymástól. Elsősorban a hallgatók számára lényeges, hogy a későbbiekben szak szerint specifikáltan, bejelentkezés nélkül megtekinthessék az óráik időpontjait.

Szakok megegyező tematikája:

Egy-egy tantárgy szerepelhet több szak oklevél követelményében is. A hallgatóknak más kóddal jelenik meg az oklevél-követelményeknél, még ha az adott tantárgy neve kísértetiesen hasonlít is. Az alkalmazásnak lehetőséget kell biztosítani arra, hogy minden előadásnál jelezzük, ha különböző szakos hallgatóknak ugyanakkor van egy órájuk.

Órarend szerkesztői igények:

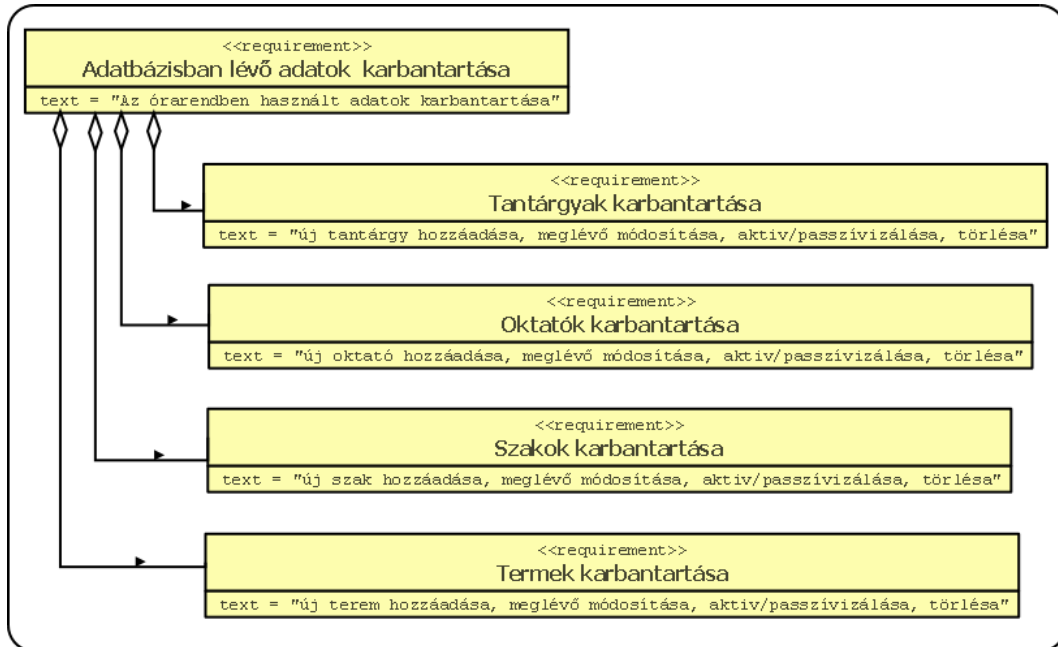
Számukra lényeges hogy a munkájukat megkönnyítsük. Könnyen kezelhető legyen a nekik kialakított kezelői felület. Fontos még az ütközések kezelése. Ha két óra ugyanabban az időben és ugyanabban a teremben lenne, kapjanak jelzést, és javíthassák. A rendszer engedje meg az ütközéseket, ám jelezze, ha két órát ugyanakkor ugyanabba a terembe szerveztünk.

2.3 Követelmények meghatározása:

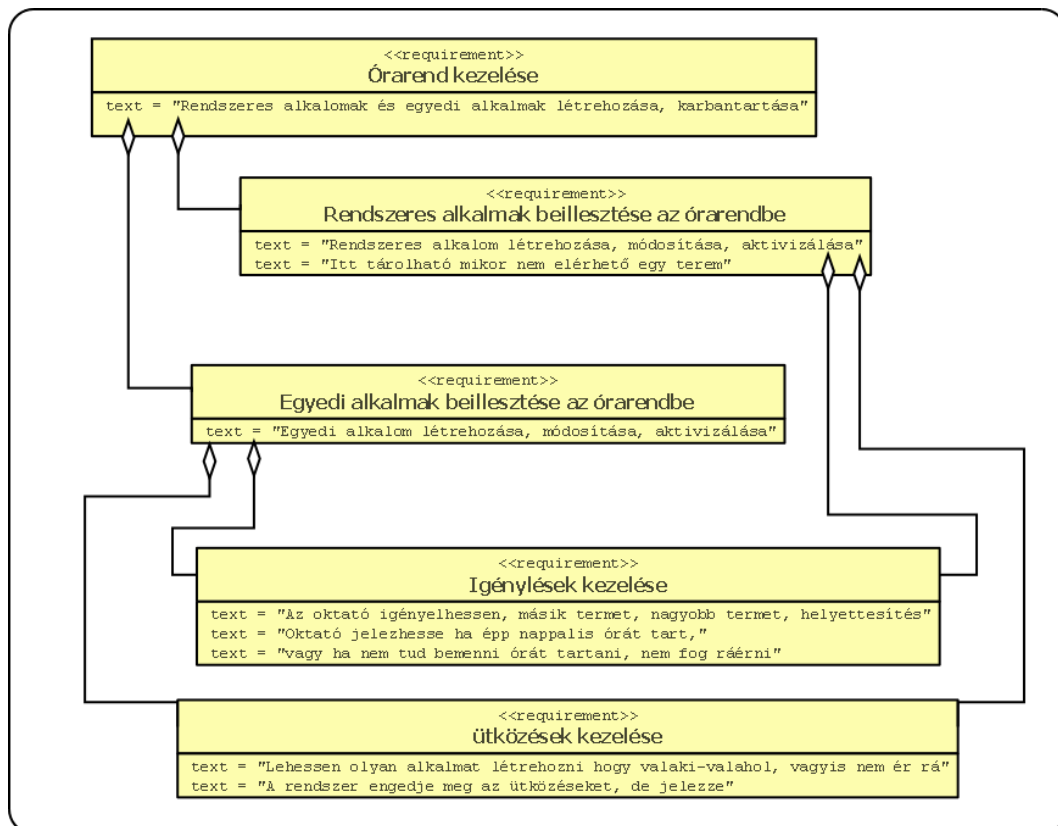
Funkcionális követelmények:

- Oktatók felvitele az adatbázisba, meglévő oktatók adatainak módosítása, törlése
- Félév során tartandó tantárgyak adatainak felvitele az adatbázisba, meglévő tantárgyak módosítása, törlése
- Levelezős oktatásra felhasználható termek hozzáadása, módosítása
- A tantárgyaknál megadható legyen, mely szakos hallgatók tanulják
- A tartandó órákat tárolni kell az adatbázisban, és karban lehessen tartani őket
- Legyenek tárolva mikor vehetők igénybe a termek
- Oktatók igényeket nyújthassanak be nagyobb teremre, helyettesítés kérésére
- Az alkalmazás által szolgáltatott funkciókhoz csak bejelentkezés után lehessen hozzáférni
- A hallgatók egy nyilvános felületen keresztül férjenek hozzá az összeállított órarendhez
- Az órarend összeállításánál megengedhető két óra ütközése, de jelezni kell
- A már feltöltött Oktatók, termek, tartandó tantárgyak esetén jelezni lehessen mely oktatókra, termekre, tantárgyakra számíthatnak az órarendszerkesztők az aktuális félévben

- Az alkalmazás a tárolásnál különböztesse meg a heti rendszerességű órákat, és az egyedi időpontra eső órákat. Ezekre a későbbiekben egységesen *alkalom* névvel hivatkozunk.



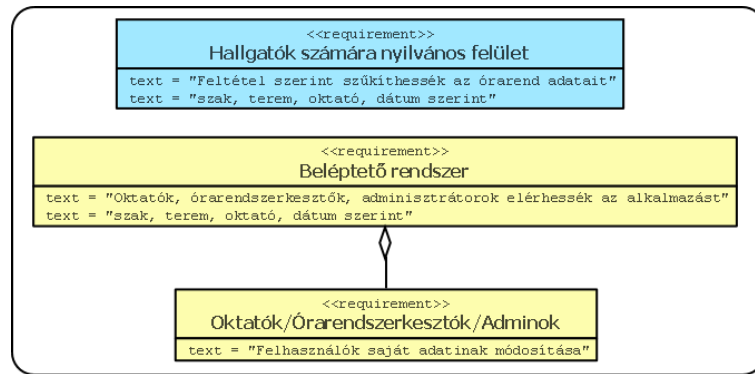
Funkcionális követelmények (adatok karbantartása)



Órarend szerkesztői követelmények

Nem funkcionális követelmények:

- Egyszerű kinézet
- Könnyű kezelhetőség
- Több felhasználós alkalmazás
- Megbízható gyors adatbázis



Nyilvános felület elválasztása a felhasználóitól

3. Az adatbázis

3.1 Az adatbázis megtervezése

Az alkalmazás által nyújtott szolgáltatások nagy része visszavezethető adatbázis kezelő funkciókra, ezért fontos volt az adatbázis szerkezetének alapos megtervezése. A karon jelenleg alkalmazott, előző változat is hosszas tervezések után készült el, de a használat során újabb és újabb követelmények merültek fel. Jelen rendszer tervezése során természetesen ezeket is figyelembe vettük.

Kiinduló pontként szolgált az egyedi és rendszeres alkalmak megadásához igényelt szükséges adatok. Egy alkalmat elsődlegesen öt adat határoz meg:

- 1) a tartandó tantárgy neve
- 2) az oktató tanár
- 3) a terem
- 4) az időpontja (kezdet-vég)
- 5) a típusa (előadás, gyakorlat, labor, egyéb, vizsga, zh)

Rendszeres alkalom:

A levelező képzés abban tér el a nappalitól, hogy az órák nem heti rendszerességgel vannak, hanem *tömbösítve*. Ennek ellenére szükség van rendszeres alkalmak kezelésére, például a tantermek foglaltsági adataira a nappali képzésből, oktatók rendszeres óráira.

Az oktatók, termek adataiból nekünk elég csak egy azonosítót tárolni az alkalmak táblában. A táblában tárolódik egy *aktív-e* mező is, amellyel jelezni lehet, hogy adott félévben él-e az adott óra. Mivel többen használják a programot, végtelen törlések esetén sose lehet megtalálni a bűnöst, ezért is döntöttünk úgy, hogy a programban az órarendszerkesztők számára csak logikai törlés létezzon. Kiegészítésképp egy megjegyzés mező is tartozik az alkalmakhoz, amellyel az órarendszerkesztők egymás felé üzenhetnek.

Fontos hogy egy alkalomhoz két vagy több különböző tárgykódú tantárgyat is lehessen illeszteni, mint többértékű attribútumokat. A normalizálás során ezért elkészült egy külön táblába kerülnek, *tantárgyak_alias* névvel. Egy rendszeres alkalomhoz a következő adatok tartoznak:

- tartandó tantárgy, (tantárgyak)
- terem

- oktató
- kezdő időpont
- befejeződés ideje
- alkalom típusa
- megjegyzés
- aktív-e

Egyedi alkalom:

Egyedi alkalomnak minősül minden levelezős tanóra, de ide tartoznak az oktatók egyéb elfoglaltságai, például konferencia miatti távollét, vagy nappali képzés zárthelyi dolgozatához lefoglalt termek.

Az egyedi alkalmak táblába kerülnek be az oktatói igénylések is. Ez csupán egy státusz mezővel van kezelve. A felvehető értékei: *igényelt, elfogadott, elutasított*. Ebben az esetben a következőket tartjuk nyilván:

- tartandó tantárgy (, tantárgyak)
- terem
- oktató
- kezdő időpont
- befejeződés ideje
- alkalom típusa
- aktív-e
- megjegyzés
- státusz (igényelt, elfogadott, elutasított)

Szak:

A szakok a tantárgyak csoportosíthatóságáért vannak, egy rövidített név, egy teljes név, egy azonosító, és itt is a jelölésre szolgáló aktív-e mezőkből áll.

- szak azonosítója
- szak neve röviden
- szak teljes neve
- aktív-e (*az adott félévben*)

Tantárgy:

A tantárgyak esetében, szükséges a név, a tantárgykód, aktív-e, és hogy melyik szakhoz tartozik. A tantárgy kódja már önmagában meghatározná egy tantárgyat, de amellet a megoldás mellett döntöttem ahol a tantárgykód csak egyszer kerül letárolásra, és az azonosítójára hivatkozok. Kényelmesebb a tantárgy adatainak javítása, és más táblákban külső kulcsként hivatkozva, csak egy azonosítót kell tárolnunk.

- tantárgy azonosító
- tantárgy neve
- tantárgy kódja
- szak
- aktív-e

Terem:

Külön tárolandók a termék adatai, azonosító, terem neve, épület ahol megtalálhatják a hallgatók, saját karhoz tartozik-e (ha igen rendelkezhetünk felette, ha nem csak kölcsönbe kaptuk), és itt is egy aktív passzív mező, ami jelzi, adott félévben felhasználható-e a terem.

- terem azonosító
- terem neve
- épülete
- saját karhoz tartozó terem
- aktív-e

Oktató:

Oktatók adatainál az adatbázis, tárolja az oktató nevét, a hozzá tartozó iroda és honlap adatait, hogy a hallgatók könnyebben felvehessék a kapcsolatot az oktatóval. Ha arra lenne szükségünk, hogy könnyen átlátható órarendet generáljunk az órarendszerkesztők számára, akkor érdemes egy-egy oktatóhoz színeket is rendelni, megadva, hogy a hozzá tartozó órák hogyan jelenjenek meg. Ezt az utóbbi adatot kihagytam a tervekből, egy későbbi verzióban kívánom felhasználni. Jelenleg a következőket tároljuk az oktatókról:

- oktató azonosító
- név (vezetéknév + keresztnév)
- iroda

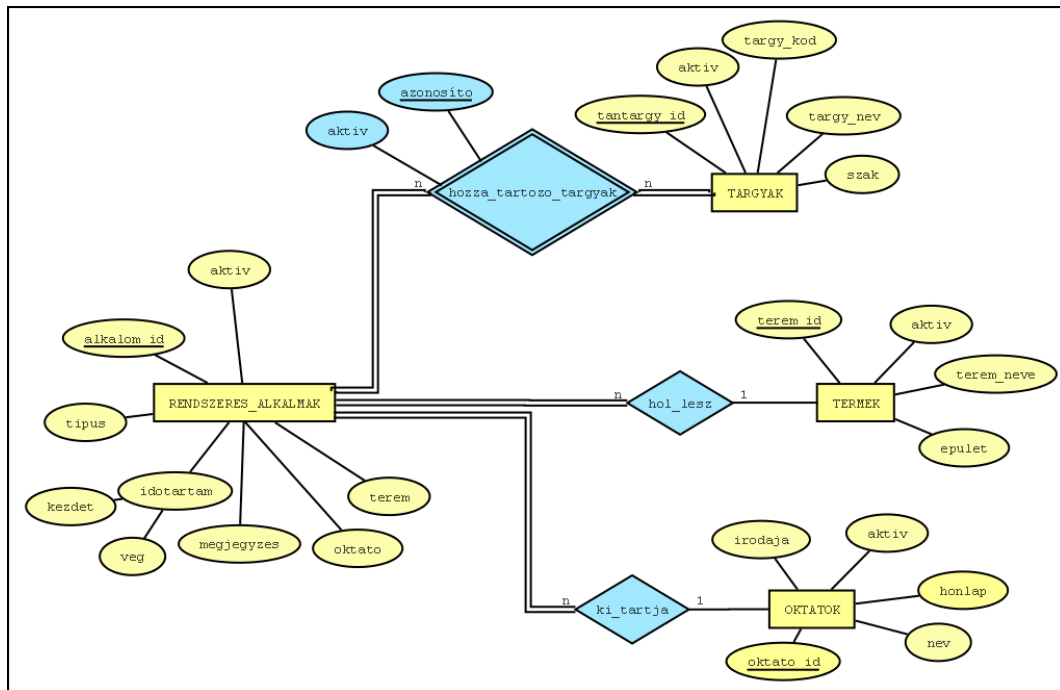
- honlap (*hallgatóknak hasznos lehet*)
- aktív-e

3.2 Adatok tárolása

Az adatbázis megtervezése során feltártam milyen adatokat kell tárolnia, és kezelni az alkalmazásnak. Az órarendhez szükséges adatokat 6+1 táblában tárolja a rendszer. Ezek nevei: Szakok, Tárgyak, Termek, Oktatók, Rendszeres alkalmak, Egyedi alkalmak, Tantárgy alias. Az utóbbi táblában, az alkalmakhoz tartozó tantárgyak kódjait tárolom. Egy alkalomhoz tartozhat két, vagy több tantárgykód is, tehát több értéket is felvehet a tartandó tantárgy kódja. Szükséges még egy táblát létrehozni a felhasználók belépéséhez szükséges azonosítók, jelszavak, jogkörök, és oktatói azonosító szám tárolására is.

3.3 Az adatbázis leírása:

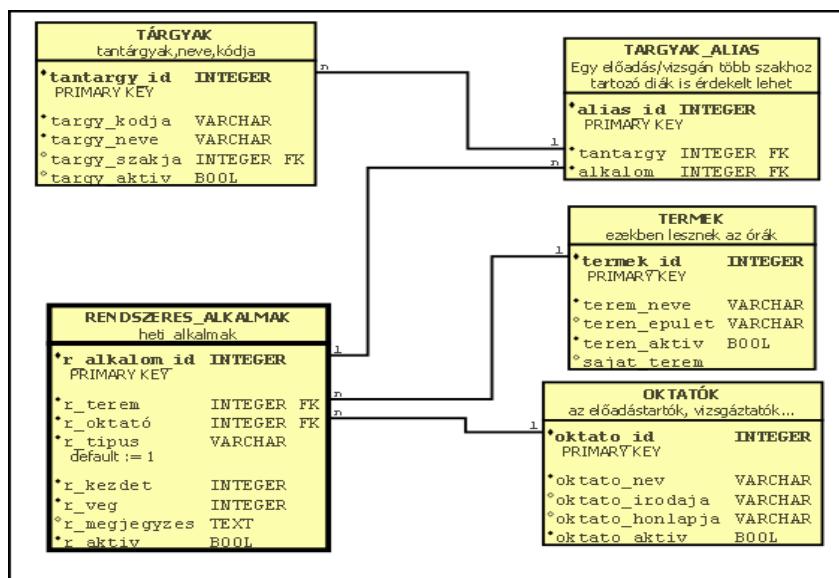
Az adatbázis sémájának ábrázolására Dia diagram szerkesztőt használtam.



Rendszeres alkalmak kapcsolatai

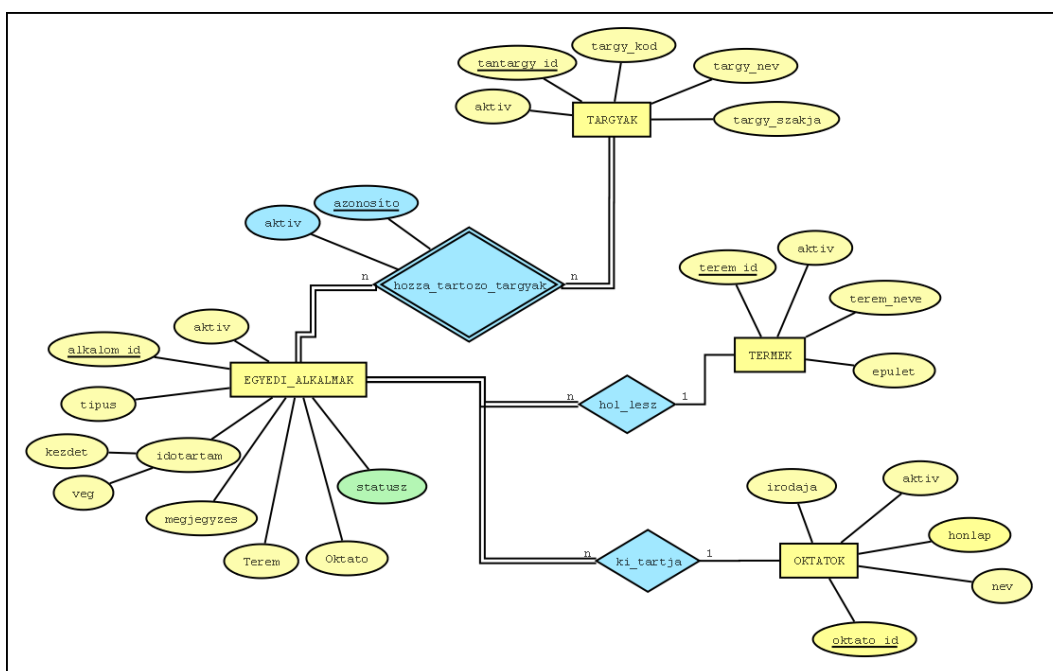
Rendszeres alkalmak tábla:

- Elsődleges kulcs: `alkalom_id`
- Külső kulcsok: `terem_id`, `oktató_id`



Rendszeres alkalmak adatbázis szerkezete

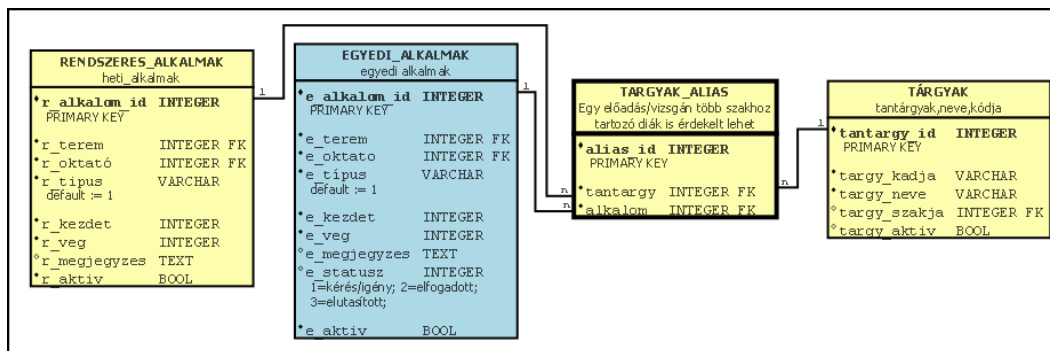
Egyedi alkalmak, tantárgyak, termek, oktatók közti reláció:



Egyedi alkalmak kapcsolata

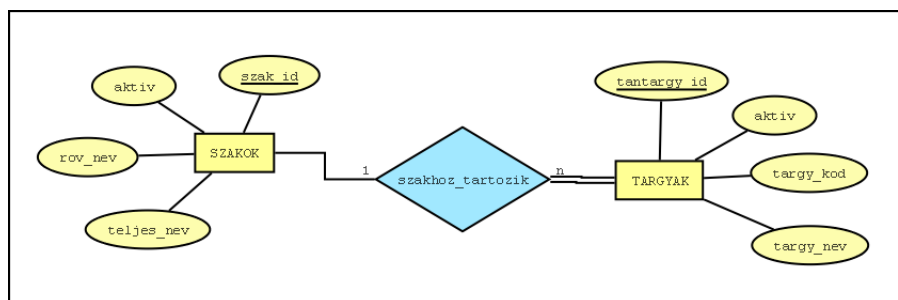
Egyedi alkalmak tábla:

- Elsődleges kulcs: alkalom_id
- Külső kulcsok: terem_id, oktató_id



Tantárgy-aliasok leképezése

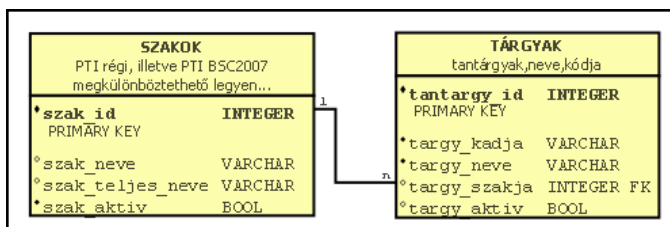
Szakok és tantárgyak közti reláció:



Szakok és tárgyak közti kapcsolat

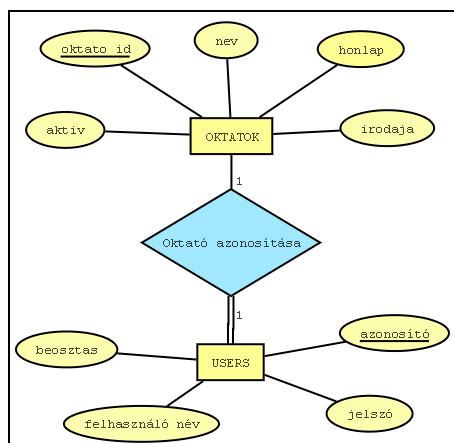
Tárgyak tábla:

- Elsődleges kulcs: tantargy_id
- Külső kulcs: targy_szakja



Szakok, tantárgyak adatbázis szerkezete

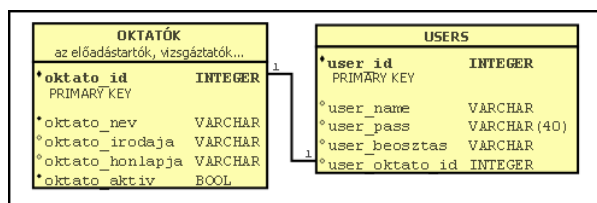
Oktatók és felhasználók közti reláció:



Felhasználók és oktatók közti kapcsolat

Users tábla:

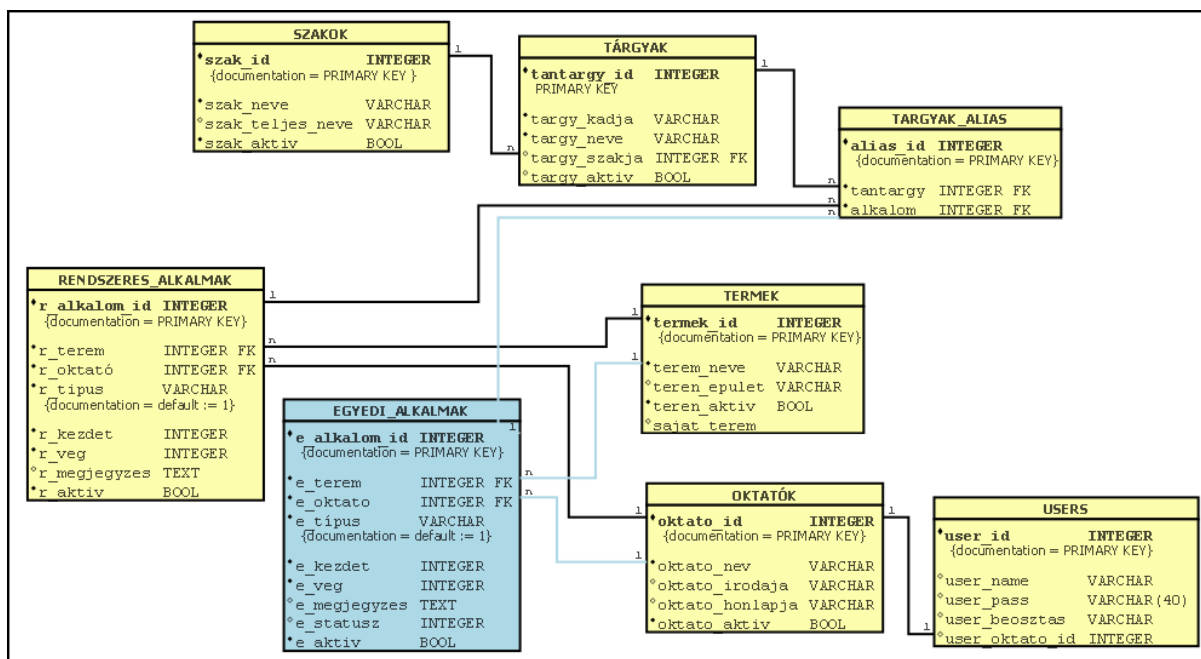
- Elsődleges kulcs: user_id
- Külső kulcs: user_oktato_id



Users oktatók tábla

A végleges adatbázis:

Az SQLite sajátosságai miatt a bool értékeket számként tároljuk: 1 = true, 0 = false. A dátumokat is számként tárolja az adatbázis, az 1970-től eltelt másodperceket tartjuk nyilván. Dátumokra az alkalmak kezdete, vége nyilvántartásánál van szükségünk. Az adatbázis karakter kódolása UTF-8-as, ezzel is próbálom követni a jelenlegi trendeket.



Végleges adatbázis kinézet

4. Program tervezése

4.1 Felhasználói Csoportok

A következőkben megadom a különböző jogosultsági csoportokat, és azok jogosultságait is. Az előzetes konzultációkból kiderült, hogy legalább 3 féle felhasználói csoportra lesz szükség, a negyedik egy adminisztrátor lesz, aki a többi felett áll.

Az alkalmazás felhasználóit négy csoportra lehet osztani:

- Hallgatók
- Oktatók
- Órarendszerkesztők
- Adminisztrátorok

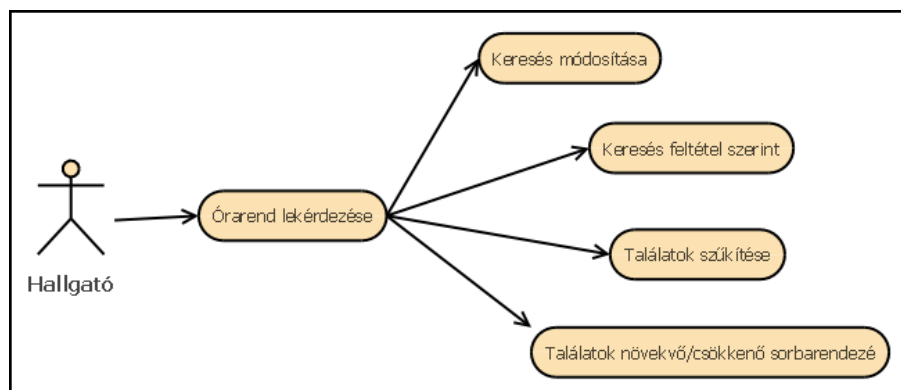
A hallgató rendelkezik a legkevesebb jogosultsággal, és tevékenységgel és az adminisztrátor a legtöbbel. Hierarchikusan egymásra épülve öröklik egymás használati lehetőségeit.

Hallgatók:

A hallgatók számára elsődleges a gyors információszerzés. Számukra egy könnyen kezelhető felületet kell biztosítani az órarend böngészéséhez, adott órák adatainak gyors kikeresésére. A felületnek bárki számára hozzáférhetőnek kell lennie.

Az alkalmazás által adott találatok számát a felhasználók szűkíthessék néhány alapvető feltétel szerint: adott szak szerinti feltétel, tantárgy szerinti keresés, oktatók, termék és napok szerint.

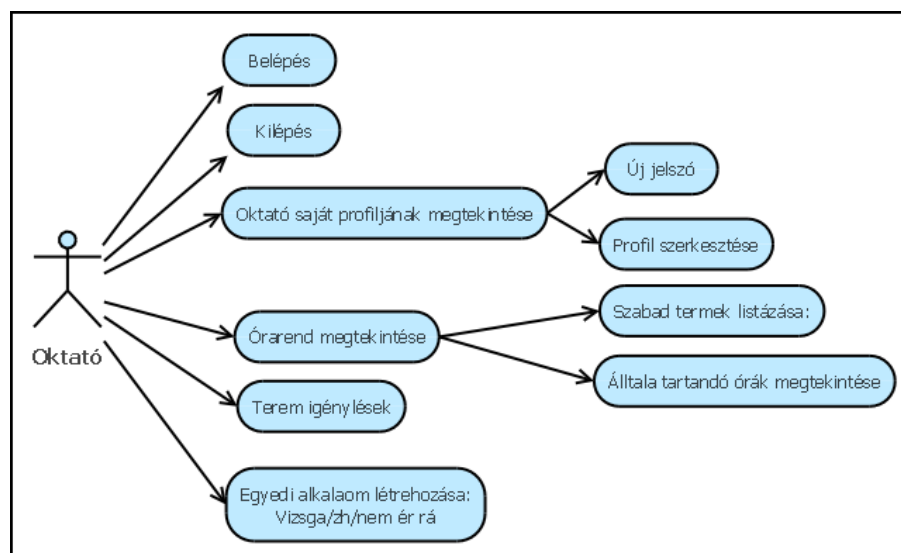
Másodlagos feltételünk, hogy a találatokat különféle szempontok szerint rendezhessék a felhasználók.



Hallgatói használati eset

Oktatók:

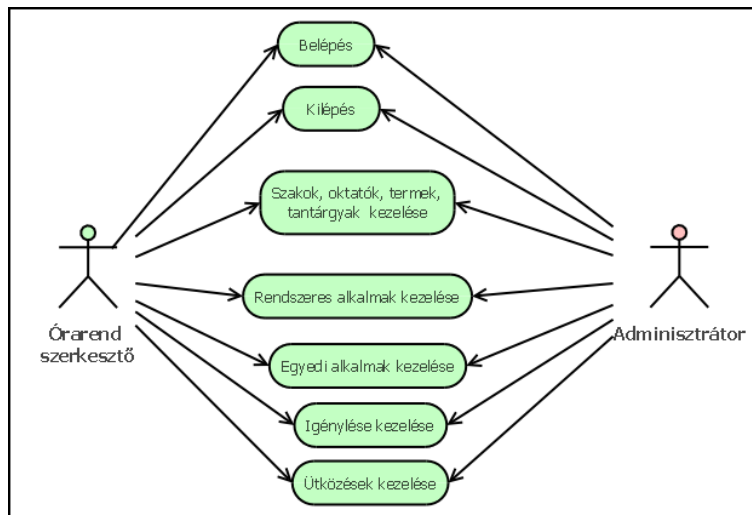
Egy bejelentkezési felület után férjenek hozzá az alkalmazáshoz. Szerkeszthessék profiljukat, új jelszót adhassanak meg, ők is megtekinthessék az órarendet; lekérdezhessék, hogy személy szerint mikor kell órát tartaniuk. Az oktatóknak igénylési lehetőséget kell biztosítani nagyobb terem, vagy helyettesítés kérésére.



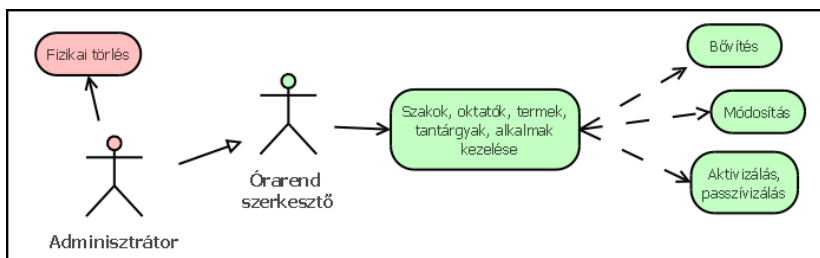
Oktatói tevékenységek használati eset

Órarend szerkesztők és adminisztrátorok:

Az órarend szerkesztők és az adminisztrátorok közti különbség kimerül abban, hogy az adminisztrátor törölhet fizikailag adatot, ezért őket együtt említem. Rendszerint az adminisztrátor és az órarendszerkesztő egy személyben oktatók is. Az oktatók minden funkcionalitását öröklik, kiegészülve az órarend szerkesztéséhez, karbantartásához, menedzseléséhez szükségesekkel.



Órarendszerkesztő és adminisztrátori tevékenységek használati eset



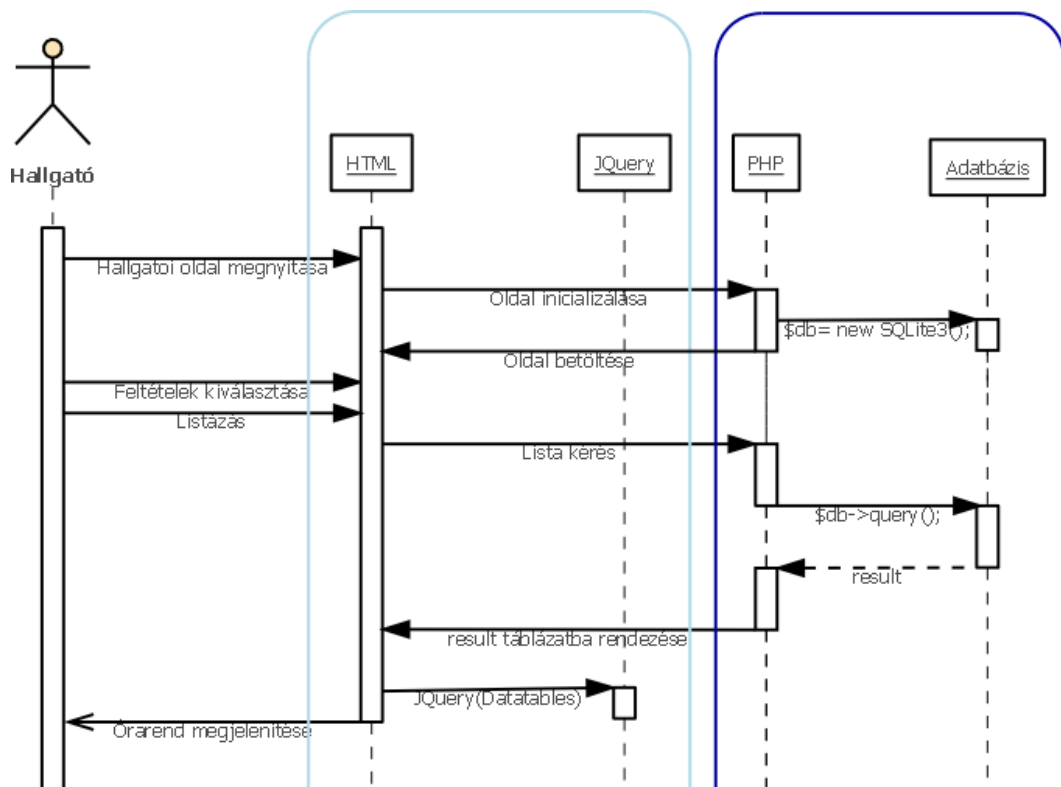
Táblák kezelése használati eset

4.2 Használati esetek

A használati esetek közül, az órarend lekérdezéséhez és a belépéshez készítettem szekvencia diagramot. Az órarend karbantartásához tartozó bővítés, módosítás, aktiválás, passzív tétel, törlés műveletek nagyon hasonlóak a lekérdezés szekvencia ábrájához, annyi különbséggel, hogy más és más parancsot futtat a PHP kód.

Órarend lekérdezése:

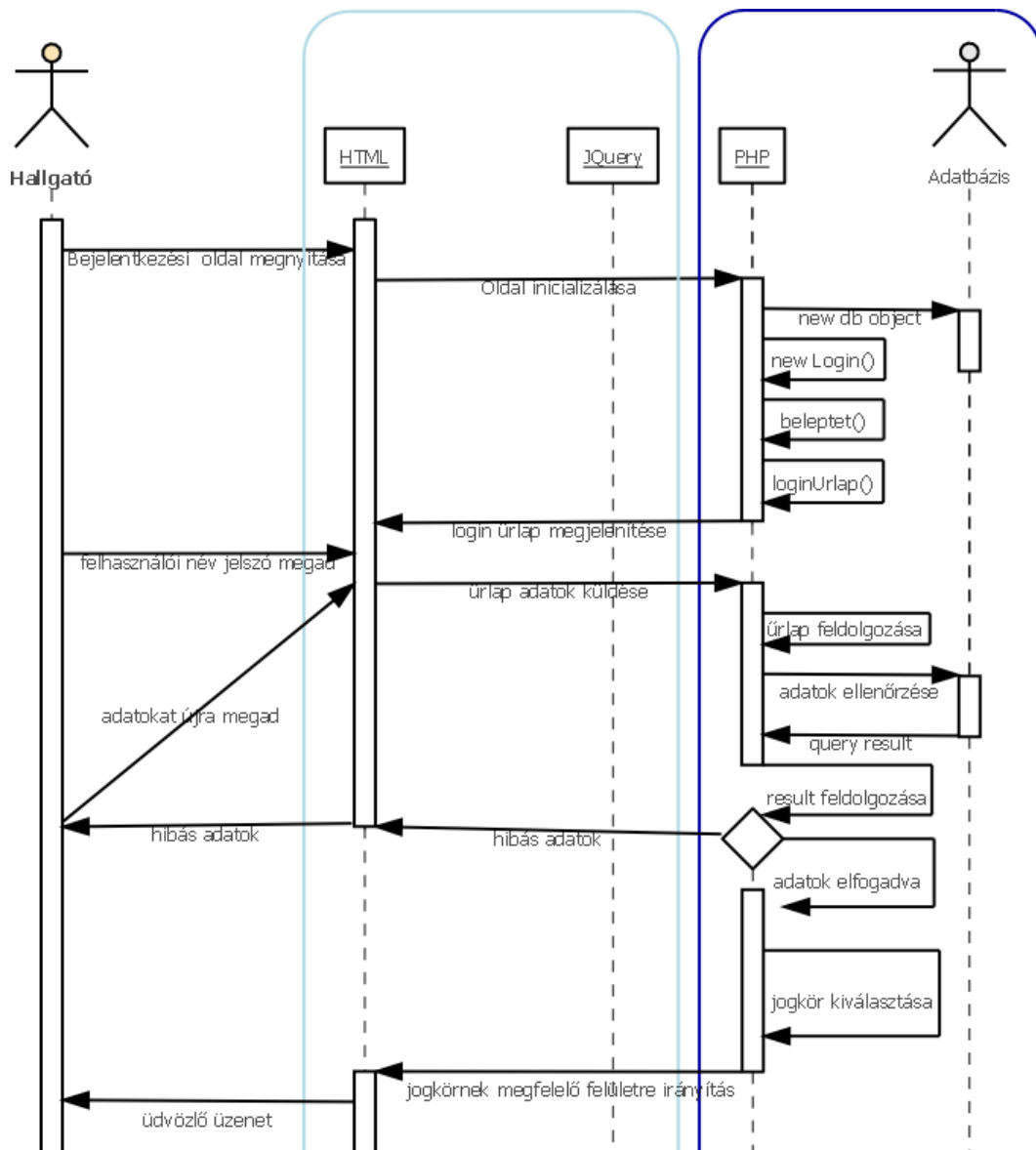
Órarend lekérdezése forgatókönyv	Leírás	Feltételek
A hallgató megnyitja az órarend honlapját. Kiválasztja milyen feltételek szerint szeretné szűkíteni az órarendet és megnyomja a listázás gombot	Az órarend nézetből lekérdezi a php modul a feltételek szerinti adatokat, és megjeleníti egy táblázatban.	<i>Alapfeltétel:</i> adatbázis megléte. <i>Előfeltétel:</i> órarend adatok megléte. <i>Utófeltétel:</i> Kilistázott órarend.



Órarend lekérdezése szekvencia diagram

Belépés kilépés:

Belépés forgatókönyv	Leírás	Feltételek
A felhasználó a belépés oldalon áll, megadja a felhasználói nevet, jelszót. Ha a megadott adatok jók, akkor megtörténik a belépés. Ha a megadott adatok nem jók, akkor a belépés nem történik meg.	A felhasználó megadja az azonosítóját és jelszavát és belép a programba	<i>Előfeltétel:</i> felhasználó rendelkezik jelszóval és azonosítóval <i>Utófeltétel:</i> belépett felhasználó



Kilépés:

Kilépés forgatókönyv	Leírás	Feltételek
A felhasználó kilép az alkalmazásból. Befejezi a program használatát	A felhasználó kilép az alkalmazásból.	<i>Alapfeltétel:</i> a felhasználó be van lépve. <i>Utófeltétel:</i> Kilépett felhasználó

Jelszó megváltoztatása:

Új jelszó forgatókönyv	Leírás	Feltételek
A felhasználó megadja kétszer az új jelszót. Ha egyezik a két jelszó, beállítódik a jelszó sha1() kódolt értéke.	A php modul átírja az új jelszóra, kódoltan a felhasználó jelszavát.	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Utófeltétel:</i> Új jelszó rögzítése

Profil módosítása:

Profil módosítása forgatókönyv	Leírás	Feltételek
A felhasználó beállítja a honlapjának elérhetőségét, és a szobája számát.	A php modul frissíti az adatbázisban az adatokat.	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Utófeltétel:</i> új adatok rögzítése

Új szak, tantárgy, terem, oktató felvitele

Forgatókönyv	Leírás	Feltételek
A felhasználó megadja a szükséges adatokat majd megnyomja a hozzáadás gombot	A php modul hozzáadja az új adatot az adatbázishoz.	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Utófeltétel:</i> új adat létrejön az adatbázisban

Új szak, tantárgy, terem, oktató módosítása

Forgatókönyv	Leírás	Feltételek
A felhasználó kiválasztja a listából a módosítandó rekordot, majd rákattint a módosításra, és megadja az új adatokat.	Az alkalmazás a Lista elemekben tárolt id alapján módosítja az adatbázis egy rekordját	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Előfeltétel:</i> létező rekord a táblában. <i>Utófeltétel:</i> új adatok rögzítése

Szak, tantárgy, terem, oktató aktivizálása, passzívizálása

Forgatókönyv	Leírás	Feltételek
A felhasználó kiválasztja a listából a módosítandó rekordot, majd rákattint az	Az alkalmazás a Lista elemekben tárolt id alapján módosítja az	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Előfeltétel:</i> létező rekord a táblában.

aktivizál/passzivizál gombra	adatbázis aktív passzív mezőjét.	<i>Utófeltétel:</i> új adatok rögzítése
------------------------------	----------------------------------	---

Új szak, tantárgy, terem, oktató törlése

Forgatókönyv	Leírás	Feltételek
Az felhasználó kiválasztja a listából a módosítandó rekordot, majd rákattint a törlésre.	Az alkalmazás a Lista elemében tárolt id alapján törli az adott rekordot	<i>Alapfeltétel:</i> a felhasználó be van lépve mint admin <i>Előfeltétel:</i> létező rekord a táblában. <i>Utófeltétel:</i> a kiválasztott rekord törölve lett

Alkalmak kezelése

Alkalmak bővítése forgatókönyv	Leírás	Feltételek
A felhasználó megadja az új alkalom adatait: kiválasztja a dátumot, a szak alapján a tárgyat, a termet, az oktatót, az alkalom típusát, és megjegyzést ír ha akar.	Az alkalmazás létrehozza az új alkalmat, majd az alias táblába beszúrja az új alkalomhoz tartozó azonosítókat.	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Előfeltétel:</i> létezik, szak, oktató, tantárgy, terem az adatbázisban <i>Utófeltétel:</i> új adatok rögzítése

Alkalmak módosítása

Alkalmak módosítása forgatókönyv	Leírás	Feltételek
A felhasználó kiválasztja a módosítandó alkalmat, és beírja az új adatokat ahol változtatni akar..	Az alkalmazás az id alapján módosítja az adott alkalmat.	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Előfeltétel:</i> létezik alkalom az adatbázisban <i>Utófeltétel:</i> adatok frissítése.

Alkalmak aktivizálása passzívizálása

Alkalmom módosítása forgatókönyv	Leírás	Feltételek
A felhasználó kiválasztja a módosítandó alkalmat a listából és megnyomja az aktivizál/passzívizál gombot.	Az alkalmazás az id alapján módosítja az adott alkalmat.	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Előfeltétel:</i> létezik alkalom az adatbázisban <i>Utófeltétel:</i> adatok frissítése.

Alkalmak törlése

Alkalmom módosítása forgatókönyv	Leírás	Feltételek
A felhasználó kiválasztja a módosítandó alkalmat, és beírja az új adatokat ahol változtatni akar..	Az alkalmazás az id alapján módosítja az adott alkalmat.	<i>Alapfeltétel:</i> a felhasználó be van lépve <i>Előfeltétel:</i> létezik alkalom az adatbázisban <i>Utófeltétel:</i> adatok frissítése.

Igények kezelése

Alkalmom módosítása forgatókönyv	Leírás	Feltételek
A órarend szerkesztő vagy az adminisztrátor listázza az igényléseket, és átállítja a státuszukat, majd igyekszik azt teljesíteni.	Az alkalmazás listázza, az igénylés alatt státuszú alkalmakat.	<i>Alapfeltétel:</i> a felhasználó be van lépve, és adminisztrátor vagy órarend szerkesztő. <i>Előfeltétel:</i> van igénylés <i>Utófeltétel:</i> frissítet státuszú igénylés.

Ütközések kezelése

Alkalmom módosítása forgatókönyv	Leírás	Feltételek
A órarend szerkesztő vagy az adminisztrátor felhasználói felületén figyelmeztetést kap ha ütközik két alkalom. Egy menüből lekérdezheti az egymást ütő alkalmakat.	Az alkalmazás, megszámlolja, majd kérésre listázza azokat az alkalmakat ahol a ütközés lép fel.	<i>Alapfeltétel:</i> a felhasználó be van lépve, és adminisztrátor vagy órarend szerkesztő. <i>Előfeltétel:</i> két alkalom ütközik <i>Utófeltétel:</i> 0 vagy 2 vagy több alkalom ütközik

4.3 Képernyő terv:

A felhasználói felületet egyhasábos, középre illesztett elrendezést használtam. Ezt három részre osztottam, melyet grafikai jellemzőkkel élesen elválasztok egymástól. Ezek a részek a következők: fejléc, lábléc, és a tartalom rész. A jelenleg divatos csak CSS megoldás és a régi iskola (HTML táblázatok mindenhol) között az arany középutat kerestem, ahol érdemes volt táblázatokat használni, ott azokat alkalmaztam, egyéb elrendezésekre pedig a CSS nyújtotta lehetőségeket.

A fejléc és a lábléc szerepe a HTML formázás megtartása, valamint a fejlécben lett elhelyezve egy elsődleges menüsor, ahol az alapvető funkciócsoportok közül választhat a felhasználó. A fejléc területe nincs teljesen kihasználva, igény szerint tovább fejleszhető.

A tartalmi rész mindig az aktuálisan meghívott funkciót jeleníti meg, valamint helyet kapott még egy másodlagos menüsor, az adott funkciócsoporthoz tartozó műveletek elérésére.

The screenshot shows the 'Levelező Órarend' web application. The header (fejléc) includes the title 'Levelező Órarend' and 'DEBRECENI EGYETEM INFORMATIKAI KÁR'. The main content area (tartalmi rész) is a form for adding a lesson, with fields for 'Szak', 'Tantárgy', 'Oktató', 'Terem', 'Kezdet', 'Vég', 'Típus', 'Megjegyzés', and 'Aktív legyen?'. The footer (lábléc) contains the copyright information: 'Copyright © 2010 Bánszki András. All Rights Reserved.'

felhasználói képernyő felület séma

Külön fejléceket alakítottam ki az órarendszerkesztőknek, adminisztrátoroknak, oktatóknak. Ezzel megszabtam az alkalmazás használata során elérhető tevékenységeket. A hallgatók felhasználói felületén és a belépéshez is ugyanezt a sémát használtam, csak kivettem a fejlécből a menüsört, a tartalmi rész pedig itt is dinamikus.

5. Implementációs kérdések

5.1 SQLite adatbázis használata:



Az adatok tárolására SQLite adatbázist választottam. Az SQLite maga csak egy programkönyvtár ami tartalmaz egy fájl alapú, konfigurálást és szerveret nem igénylő, SQL tranzakciós adatbázis motort. Komplet SQL adatbázis egy fájlban, amit közvetlenül ír és olvas. Az adatbázis fájl formátuma megengedi a 32 bites és 64 bites rendszerek közötti átvitelt. Előnyei közé sorolható hogy kicsi a helyigénye, gyors, megbízható, valamint ingyenes, akár a forrásfájljait is elérhetjük a hivatalos honlapon.

Több nagy cég is használja az SQLite-ot: Adobe, Apple, Mozilla, Google, Symbian. Valamint a PHP, Python, Realbasic, C, Java nyelvek is támogatják az SQLite-ot. Megemlíteném, hogy jelenleg a 3. verziónál jár, ami inkompatibilis a korábbiakkal, és hogy minimum PHP 5.3.0 verzió szükséges a használatához, vagy annál újabb.

5.2 Az Implementáció:

Az alkalmazást PHP nyelven írtam meg, NetBeans IDE 6.8 PHP fejlesztő eszközben. Felhasználásra került két érdekes jQuery Plugin, valamint az alkalmazás egységes kinézetét CSS-el formáztam. Az SQLite adatbázis szerkezetét Mozilla Firefox SQLDeveloper-rel készítettem el, és személyesen ezt a programot használtam a tesztadatbázis feltöltésére is.

A fejlesztés során Apache HTTP szerveren teszteltem a már elkészült alkalmazás részleteket, Windows XP környezetben. A nyelv objektumorientáltságát csak a bejelentkeztetés, és regisztráció során használtam ki. A funkciók egyszerűsége miatt feleslegesnek, lassúnak éreztem mindenhez külön osztályt létrehozni.

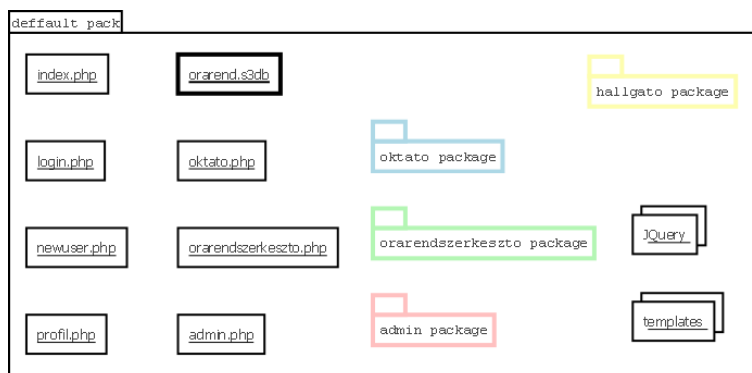
Néhány érdekes kód

Mivel az SQLite-ban a dátumok számként tárolódnak, ezeket oda vissza kell konvertálni a felhasználó által értelmezhető adatokká.

```
strtotime($_POST['kezdet'])  
strftime("%Y. %B %d. %A %H:%M ", $datum);
```

A felhasználók belépéshez szükséges adatainál a jelszót magát nem, csak az SHA-1 szabványnak megfelelő negyven karakteres kódját tároltam. A mai PHP verziók, 4.3.0-tól kezdve támogatják a megvalósítását: `sha1($jelszo);`

Az alkalmazás forrásait a felhasználók jogköre szerint csoportosítottam. A következő ábrán az alkönyvtárakat kis csomagként jelöltem.



források elhelyezkedése

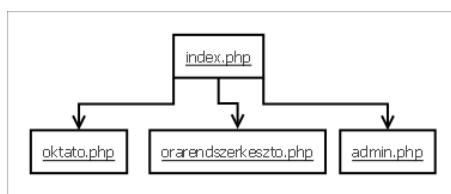
A fő csomag tartalmazza, az `index.php`-t. Betöltődésekor beillesztem az adatbázis elérést, és a `login.php`-t:

```
include 'database.php';
include("login.php");
```

Majd meghívódik a beléptetés, és azon belül a bejelentkeztetési űrlap.

```
$login = new Login();
$login->beleptet();
```

Sikeres bejelentkezés esetén a jogkörnek megfelelő felhasználó felületre vált az alkalmazás, sikertelen esetén marad a bejelentkeztető képernyő.



Az `oktato.php`, `orarendszerkeszto.php`, és az `admin.php`, az `index.php`-hez hasonlóan egy controller osztályt helyettesít meg.

```
if (isset($lap)){
    switch ($lap){
        case "profil": include 'profil.php'; unset($lap);break;
        ...
    }
}else{ include 'profil.php'; }
```

Az `admin.php`-keresztül érhetőek el az adminisztrátorok által elvégezhető tevékenységek. Ezeket egy-egy php forrás testesíti meg. Ennek a mintájára működik az `oktato.php`, és az `orarendszerk.php` is.

A dátumok bevitelére egy JQuery datetimepickert használtam, ami leegyszerűsíti a dátumok megadását és egyben validálását is. Az táblázatos formában megjelenített adatokat pedig a dataTables pluginnel sorba rendezhetővé, kereshetővé, átláthatóbbá tettem.

Ütközések kezelése:

Az órarend helyes létrehozásához szükséges, hogy két különböző óra ne legyen ugyanabban az időben, ugyanabban a teremben. Egy oktató se lehet egyszerre két helyen. Ha mégis bekövetkezne valamelyik eset, akkor ütközés történne.

Ütközést vizsgálni nagyon egyszerűen tudunk SQL lekérdezéssel is. A vizsgálatnál nem elég a két alkalom kezdeti és befejező időpontjának megegyezését vizsgálni, hanem a következőt kell figyelni:

```
két alkalom: c, d ütközik egymással ha:  
c.kezdőIdőpont < d.vége ÉS d.kezdőIdőpont < c.vége
```

Ebben az esetben c és d is az adatbázisban szereplő egyedi alkalmakat jelenti. Azaz nekünk nincs másra szükségünk, mint az adattáblát saját magával összekapcsolni, ami nem egy gyakran alkalmazott technika. Természetesen egy időben több óra is lehet, ami számunkra érdekes, hogy ugyanaz az oktató tartja mindkettőt két különböző teremben; két oktató tartja ugyanabban a teremben. Van még egy olyan eset is, amikor egyszerre két oktató tart órát két különböző teremben, de ugyanannak a szaknak. Léteznek olyan esetek, mikor ezt ütközésnek tekinthetjük, viszont van olyan is, amikor ez nem számít ütközésnek, például párhuzamos gyakorlati csoportok. Mivel a rendszer nem rendelkezik megfelelő mesterséges intelligenciával, hogy döntsön róla, hogy egy ilyen eset ütközés, vagy sem, érdemes diszkrét módon jelezni az ilyen eseteket az órarendszerkesztőnek. A korábban említett valódi ütközésekre viszont harsányan fel kell hívni a figyelmüket.

6. Felhasználói dokumentáció

Az alkalmazás használatának megkezdéséhez a felhasználóknak az `index.php` oldalon kell bejelentkezniük. Mind az adminisztrátorok, mind az oktatók és órarendszerkesztők ugyanazon a belépési felületen keresztül jelentkezhetnek be.

- A felhasználónak meg kell adnia a felhasználói nevét, amit vagy saját maga választott ki a regisztráció során, vagy az adminisztrátor hozta létre.
- A felhasználónak meg kell adni a jelszavát
- Belépés gombra kattintva, helyesen megadott név-jelszó páros esetén beléphet.



The screenshot shows the login page for 'Levelezős Órarend'. The header includes the logo and name 'Levelezős Órarend' with 'DEBRECENI EGYETEM INFORMATIKAI KAR' below it, and a navigation menu with 'Levelezős órarend tervezése és publikálása' and 'TÉMAVEZETŐ: ASZALÓS LÁSZLÓ'. The main content area has a light blue background with the text 'Be kell lépned ahhoz, hogy megnézhesd ezt az oldalt!'. Below this are input fields for 'név' and 'jelszó', a 'Belépés' button, and a link 'Nem vagy regisztrálva? Regisztráltasd magad ITT'. The footer contains the copyright notice 'Copyright © 2010 Bánszki András. All Rights Reserved.'

login képernyő

Amennyiben az adott felhasználó még nem jelentkezik saját azonosítóval és jelszóval, regisztrálhat is. Később a felhasználót az adminisztrátor hozzárendelheti egy-egy oktatóhoz, és akkor már teljes mértékben elérhetővé válnak számára az oktatói funkciók.

- Regisztráció során elegendő egy felhasználó nevet és jelszót megadni.
- Regisztráció akkor sikeres, ha jelszó és a jelszó ellenőrzés megegyezik valamint a megadott login azonosítót nem használja még senki sem.



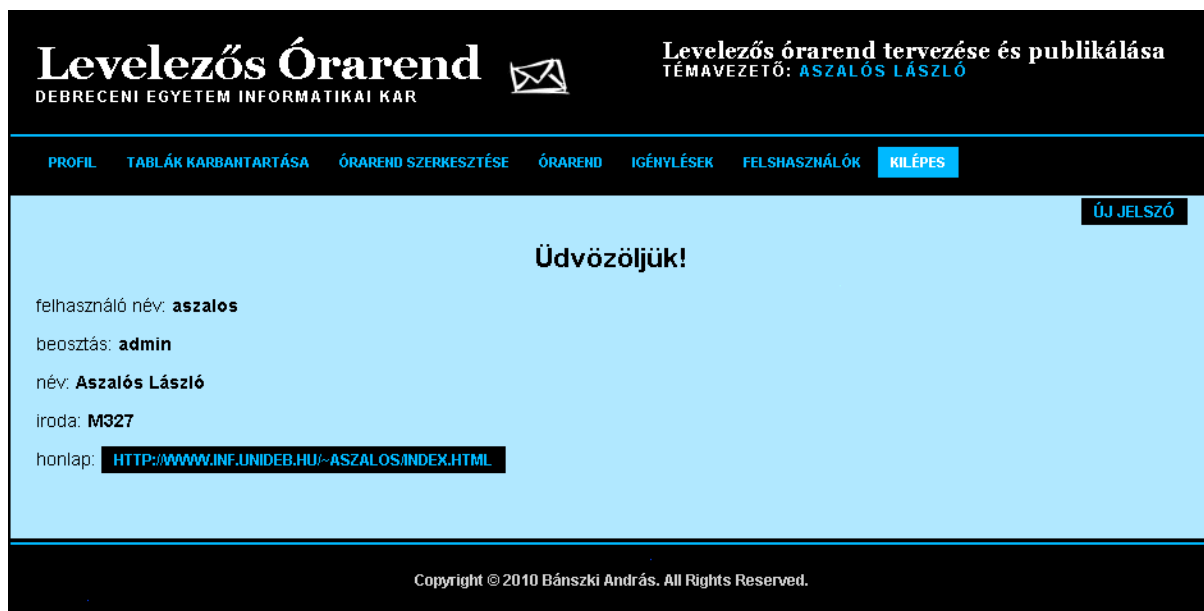
The screenshot shows the registration page for 'Levelezős Órarend'. The header is identical to the login page. The main content area has a light blue background with the title 'Regisztráció'. Below the title are input fields for 'Login név:', 'Jelszó:', and 'Jelszó (mégegyszer):', followed by a 'Regisztrálok magam' button. A 'VISZTA A FŐOLDALRA??' button is also present. The footer contains the copyright notice 'Copyright © 2010 Bánszki András. All Rights Reserved.'

regisztráció

Sikeres bejelentkezés után a felhasználót a profil oldal fogadja. Itt megváltoztathatja jelszavát, ha már hozzá lett csatolva a felhasználói lapjához hogy melyik oktató is valójában.

A következőkben egy adminisztrátornak mutatom be az alkalmazás használatát. Az oktatók, és az órarendszerkesztők, ezt vegyék kiinduló pontnak.

- A profil linkre kattintva az elsődleges menüsorban visszatérhet a profiljához.
- A táblák karbantartása linken elérhető az oktatók, tantárgyak, termek, szakok táblák adatainak a szerkesztése, bővítése
- Az órarend szerkesztésére kattintva az egyedi, és rendszeres alkalmakat kezelheti
- Az órarend menüpont alatt tekinthető meg az órarend aktuális állapota.
- Az igénylések menüpont alatt érhető el az oktatói igények listája, és azok kezelése.
- A felhasználók menüponton belül módosíthatja a regisztrált felhasználók jogosultságát, és beállíthatja az oktató azonosítókat a felhasználók számára.
- A felhasználó bármikor kiléphet a kilépés gombra kattintva.



Bejelentkezés után

Az oktatók és órarendszerkesztők a jogosultságuknak megfelelő menüpontokat találnak a menüsorban. Így például az oktató a saját profilját szerkesztheti, igényelhet új órarendet, és megtekintheti az órarendet. Az órarendszerkesztő pedig megkap minden adminisztrátori menüelemet, kivéve a törlés lehetőségét a táblák karbantartása almenüpontokban.

A táblák karbantartása menüpont alatt kiválasztható, hogy melyik táblát akarja a felhasználó kezelni, a jogosultságának megfelelően.

- A táblázatban az összes sora megjelenik az adott táblának.
- Az egyszerre 10, 25, 50 elemet lehet megjeleníteni.
- Az egyszerre megjelenített elemek számát kiválaszthatja a táblázat első sorában.
- A többi elem a táblázat jobb alsó sarkában elhelyezett nyilakra kattintva, lépkedve érhető el.
- Másik tábla szerkesztése az almenüpontból választható ki.
- Az adott elem aktivizálását/passzívizálását, törlését, módosítását az adott elem sorában található gombbal lehet elérni.
- A felhasználó oszlopokként sorba rendezheti a táblázat elemeit az oszlop nevére kattintva.
- A felhasználó kereséssel szűkíthet a megjelenített elemek között.

PROFIL TABLÁK KARBANTARTÁSA ÓRAREND SZERKESZTÉSE ÓRAREND IGÉNYLÉSEK FELSHASZNÁLÓK KILÉPÉS

SZAKOK LISTÁZÁSA OKTATÓK LISTÁZÁSA TARGYAK LISTÁZÁSA TERMÉK LISTÁZÁSA

Show 10 entries Search:

Targy kodja	Targy neve	Szak neve	Aktív/Passzív			
IL3972	Az internet eszközei és szolgáltatásai	PTI lev régi	aktív	aktív/passzív	módosít	töröl
ILBC302	Őnálló laboratórium 2	MI lev	aktív	aktív/passzív	módosít	töröl
ILBC401	Információ- és kódelmélet	MI lev	aktív	aktív/passzív	módosít	töröl
ILBC411	Adatbiztonság	MI lev	aktív	aktív/passzív	módosít	töröl
ILBC722	Bevezetés a CISCO eszközök programozásába 2	MI lev	aktív	aktív/passzív	módosít	töröl
ILBK101	Diszkrét matematika 1	MI lev	aktív	aktív/passzív	módosít	töröl
ILBK111	Kalkulus 1	MI lev	aktív	aktív/passzív	módosít	töröl
ILBK201	Bevezetés az informatikába	MI lev	aktív	aktív/passzív	módosít	töröl
ILBK211	Operációs rendszerek	MI lev	aktív	aktív/passzív	módosít	töröl
ILBK311	Programozás labor 1	MI lev	aktív	aktív/passzív	módosít	töröl

Showing 1 to 10 of 77 entries

Táblák listázása

Új rendszeres vagy egyedi alkalom hozzáadása

A rendszer megengedi az ütközést, nem ajánl fel másik időpontot; csupán jelzi, ha ütközik két vagy több alkalom.

- Először válassza ki az új alkalomhoz tartozó tantárgy szakját
- Majd válassza ki a tantárgyat
- Majd válassza ki az oktatót
- Majd válassza ki a termet
- Válassza ki az alkalom típusát
- Az alkalom kezdetének megadása: a kezdő dátum input mezőjébe kattintva, felugrik egy dátumválasztó. Jelölje ki az évet, hónapot, napot, órát, percet.
- Írjon megjegyzést, ha szükséges; majd jelölje be, hogy az újonnan létrehozni kívánt alkalom aktívként kerüljön e be az adatbázisba.

The screenshot shows the 'Levelező Órarend' web application interface. The header includes the title 'Levelező Órarend' and 'DEBRECENI EGYETEM INFORMATIKAI KAR'. The main navigation bar contains links: PROFIL, TABLÁK KARBANTARTÁSA, ÓRAREND SZERKESZTÉSE, ÓRAREND, IGÉNYLÉSEK, FELSHASZNÁLÓK, KILÉPÉS. The current page is 'RENDSZERES ALKALOM HOZZÁADÁSA'. The form includes fields for 'Szak' (dropdown), 'Tantárgy' (dropdown with 'Előbb válasszon!!!' message), 'Oktató' (dropdown), and 'Terem' (dropdown). The 'Kezdet' (start) field is set to '2010-01-01 00:00:00' and the 'Vég' (end) field is set to '2010-01-01 01:00:00'. A date picker is open, showing 'Jan 2010' and a calendar grid. The 'Típus' (type) dropdown is set to 'előadás'. There is a 'Megjegyzés' (remarks) text area and an 'Aktív legyen?' (active?) checkbox. A 'beallit' button is present. A time picker shows 'Time 01 : 00' with 'Hour' and 'Minute' spinners. A 'Close' button is at the bottom right of the date picker. The footer contains 'Copyright © 2010 Bánszki András.' and a 'Today' button.

új óra hozzáadása

7. Továbbfejlesztési lehetőségek

A rendszerben további lehetőségek rejlenek, még több kényelmi funkciót lehetne bevezetni az órarendet szerkesztő felhasználók számára. A Datatables jQuery plugint tovább lehet paraméterezni, hogy ne csak a táblázatosan megjelenített lista egészét lehessen kereséssel szűkíteni, hanem oszlopokként is.

A hallgatók számára további információkat lehetne hozzáférhetővé tenni, az órarenden kívül: hol, melyik épületben találhatóak meg a termek. Az adatbázis szerkezetben tárolható lehetne még a tantárgyak, oktatók, termek további adatai is.

Biztonsági szempontból az oktatók regisztrációját ki lehetne venni az alkalmazásból, és az oktatók belépéshez szükséges felhasználói név és jelszó létrehozását az adminisztrátorok gondjára lehetne bízni.

8. Összefoglalás

A konzultációkat követően, ismerve a problémát; bizonyossá vált, hogy fontos lenne egy ilyen alkalmazás bevezetése, az órarendbeli változtatások egységesítésére. Az órarend karbantartásához szükséges követelmények, és a levelezős órarendek egyediségének feltárása után megterveztem az adatbázist. Ennek újdonságát számomra az SQLite adatbázis használata jelentette. Majd a felhasználói csoportok, használati esetek megtervezése után kezdetét vette a tényleges alkalmazás megvalósítás. A karon jelenleg alkalmazott, előző változat használata során felmerült újabb követelményeket figyelembe véve, a mai fejlesztő, és adatbázis kezelő eszközök segítségével sikerült egy olyan alkalmazást elkészíteni, amely a levelezősöknek tartandó órák változtatásainak egységesítését orvosolja.

9. Irodalom jegyzék

- [1.] <http://www.php.net/manual/en/> - PHP manual
- [2.] <http://hu2.php.net/manual/en/book.sqlite3.php> - PHP SQLite3 manual
- [3.] <http://www.sqlite.org/version3.html> SQLite Version 3 Overview
- [4.] Sági Gábor: Webes adatbázis kezelés MySQL és PHP használatával
BBS-INFO KÖNYVK. ÉS INFORM. KFT.(2005)
- [5.] <http://datatables.net/>
Allan Jardine, 2010, *Datatables – table plug-in for jQuery*,
- [6.] <http://blog.w3visions.com/2009/04/date-time-picker-with-jquery-ui-datepicker/>
Date Time Picker with jQuery UI DatePicker, 2010