

# *Szakedolgozat*

*Szoboszlai Sándor*

Debrecen

2011

Debreceni Egyetem

Informatikai Kar

## **Nyilvántartó rendszer fejlesztése Java nyelven**

Témavezető:

Kollár Lajos

Egyetemi tanársegéd

Készítette:

Szoboszlai Sándor

Prog. tervező inf. BSc

Debrecen

2011

## **Plágium - Nyilatkozat**

Szakedolgozat készítésére vonatkozó szabályok betartásáról nyilatkozat

Alulírott (Neptunkód: BXIINB) jelen nyilatkozat aláírásával kijelentem, hogy a

Nyilvántartó rendszer fejlesztése Java nyelven

című szakdolgozat/diplomamunka

(a továbbiakban: dolgozat) önálló munkám, a dolgozat készítése során betartottam a szerzői jogról szóló 1999. évi LXXVI. tv. szabályait, valamint az egyetem által előírt, a dolgozat készítésére vonatkozó szabályokat, különösen a hivatkozások és idézések tekintetében.

Kijelentem továbbá, hogy a dolgozat készítése során az önálló munka kitétel tekintetében a konzulenszt, illetve a feladatot kiadó oktatót nem tévesztettem meg.

Jelen nyilatkozat aláírásával tudomásul veszem, hogy amennyiben bizonyítható, hogy a dolgozatot nem magam készítettem vagy a dolgozattal kapcsolatban szerzői jogsértés ténye merül fel, a Debreceni Egyetem megtagadja a dolgozat befogadását és ellenem fegyelmi eljárást indíthat.

A dolgozat befogadásának megtagadása és a fegyelmi eljárás indítása nem érinti a szerzői jogsértés miatti egyéb (polgári jogi, szabálysértési jogi, büntetőjogi) jogkövetkezményeket.

.....

hallgató

Debrecen, 2011. 04. 16.

# Tartalomjegyzék

1. Bevezetés .....	4
2. Szoftver tervezés.....	6
2.1 Követelmények összegyűjtése .....	6
2.2 A rendszer funkcionális követelményei .....	7
2.2.1 Az adminisztrátor funkcionális követelményei .....	7
2.2.2 A gondozó funkcionális követelményei .....	8
2.3 A rendszer környezete .....	10
2.4 Adatbázis terv .....	10
2.5 Rendszer tervezése .....	14
2.5.1 Rendszer csomag .....	15
2.5.2. Felületek csomag .....	15
3. Felhasználói kézikönyv .....	17
3.1 A program indítását megelőző teendők .....	17
3.2 A program indítása .....	17
3.3 Bejelentkezés .....	17
3.3.1 Adminisztrátor által végezhető feladatok .....	18
3.3.2 A gondozó által végezhető feladatok.....	20
4. Összegzés .....	30
Irodalomjegyzék .....	33
Köszönetnyilvánítás .....	34

# 1. Bevezetés

Azért választottam a nyilvántartó rendszernek egy hajléktalanellátó központ adatkezelésének az elektronikus megvalósítását, mert manapság a hajléktalanság, mint probléma napi szinten jelenik meg a televízióban és a sajtóban. Az egyik legnagyobb problémát a hajléktalanok számának meghatározása okozza. Jelenleg sincs kidolgozva olyan egységes rendszer, ami elősegítené ezt. A legtöbb intézmény papír alapú nyilvántartást vezet, amit a törvény előír, de 2011. január 1.-el módosultak a szociális igazgatásról és szociális ellátásokról szóló 1993. évi III. tv. valamint a személyes gondoskodást nyújtó szociális ellátások igénybevételéről szóló 9/1999. (XI. 24.) SzCsM. rendelet adminisztrációs kötelezettségekre irányadó rendelkezései. Ezáltal lehetővé válhat egy olyan online nyilvántartó rendszer kidolgozása, ami helyettesítheti, sőt a későbbiekben esetleg ki is válthatja a papír alapú nyilvántartást, illetve ügyintézését.

Dolgozatom célja, hogy egy könnyen használható, mindenki által áttekinthető és a nyilvántartás szempontjából formalizált program létrehozása volt. A szoftver megírásakor a következő szempontokat tartottam szem előtt:

- adatbázis használata,
- az adatbázis bárholnan legyen elérhető,
- platformfüggetlenség,
- egyszerű kezelhetőség.

Azért választottam adatbázist, mert így nincs szükség szerializált állományokra, melyek nemcsak a programot lassítanák, hanem adateltéréseket is okozhatnának. Mivel központosított adatbázist használtam, emiatt a program maga hozza létre az adatbázissémát, a program legelső indításakor. Emiatt a program csak egyfajta adatbázissal fut. Ennek az adatbázisnak én az Oracle Database 11g-t választottam.

A külön adatbázisszerver miatt az adatbázis internet kapcsolaton keresztül bárholnan elérhető. Ez a hajléktalan ellátásban fontos, ugyanis sokszor erdőszélen, közterületen, parkokban lehet megtalálni őket. Így a rendszernek hála azonnal fel lehet venni az adatait, nem kell külön papíron rögzíteni, majd később egy elektronikus nyilvántartásba bevezetni.

A helyszínen kiderül, hogy nyilvántartásba van-e már véve vagy eddig még nem került a hajléktalanellátás látókörébe.

Az eddigi elektronikus nyilvántartások általában excel táblák, amelyek azonban csak Windows környezetben használhatók, emiatt fordultam a platformfüggetlen rendszer felé. Ezen okból kifolyólag választottam a program megírásához a Java nyelvet. Így a forráskód fordítása után előálló tárgykód, bármely olyan operációs rendszeren elfut, amelyen található Java futtató környezet (Java Runtime Environment - JRE).

Az egyszerű kezelhetőséget azért tartottam szempontnak, mert a szociális szféra dolgozói közül nem mindenkinek van informatikai képzése, sőt általában az a megszokott, hogy nincs. Jelenleg még sokan dolgoznak olyanok, akik a 80-as évek előtt végezték el iskoláikat, amikor még az informatika nem volt tananyag az általános és középiskolákban. Azért próbáltam egyszerűsíteni, hogy ezek az emberek is könnyen elsajátíthassák a program használatát. Ezért a program kialakításánál törekedtem arra, hogy informatikai háttér nélkül is lehessen kezelni, emiatt lettek a kezelőfelületek egyszerűek.

Szakedolgozatom további részében megpróbálom bemutatni a program technológiai háttérét, majd a fejlesztés folyamatát, valamint segítségnyújtásként adni egy egyszerű felhasználói kézikönyvet.

## 2. Szoftver tervezés

### 2.1 Követelmények összegyűjtése

Minden szoftverfejlesztés a követelmények feltárásával kezdődik. Ebben a fázisban a fejlesztő és a megrendelő tisztázza, hogy a szoftvernek milyen követelményeknek kell megfelelnie.

A papír alapú rendszerben előfordulhat, hogy ha egy ügyfél két vagy több intézményben is megfordul, mindkét helyen nyilvántartásba veszik, és akár elindításra kerülhet mindkét helyen az iratpótlás is. Azaz a hajléktalannak valamilyen személyes irata nincs meg, pl. Taj.kártya, személyi igazolvány. Ezek elkészítését a hajléktalanellátó centrum hivatalból kezdeményezi. Az online nyilvántartási rendszer ezen hibák egy részét próbálja meg megszüntetni azáltal, hogy ugyanazt az adatbázist használja mindegyik intézmény, így elkerülhető egy - egy gondozott többszöri felvétele.

A szoftver fejlesztésénél a következő követelményeket vettem figyelembe:

1. A rendszer legyen képes a hajléktalanok adatainak tárolására és módosítására.
2. Lehessen a hajléktalanokkal szerződést kötni,
3. Ha a hajléktalan gondozásra szorul, akkor a gondozási tevékenységet lehessen felvinni a rendszerbe,
4. Az adatokat egy külön adatbázisszerveren tárolja, hogy bárhonnán elérhető legyen.
5. Lehessen a fontosabb adatokra keresni,
6. A rendszer a nemi eloszlásra, iskolázottságra, családi állapotra, életkorra, jövedelem forrásra, és lakhely típusára (pl. közterület, telek vagy lakás) szolgáltatson statisztikát.
7. Egyszerű használat és könnyű tanulhatóság.

Ezen követelmények úgy álltak elő, hogy szemben a papír alapú vagy az excel táblázatos nyilvántartással azonnali információ álljon rendelkezésre bármely ellátásban. A kialakítandó új, felhasználóbarát nyilvántartási rendszer újdonsága, hogy a hajléktalan ellátás intézményeit egyszerre vagy eltérő időben használó ügyfelek adatai illetve a részükre nyújtott szolgáltatások, a szociális munka lépései, együttműködési adatok stb. nem vesznek el a

rendszerben, bizonyos korlátozások között, de rendelkezésre állnak a többi intézmény részére is. Egy - egy ügyfél esetében nem kell újra és újra felvenni az adatait, hanem azok bármely hajléktalanellátó központhoz tartozó intézményben rendelkezésre állnak. Belső információ áramlás mellett a kifelé történő adatszolgáltatás megkönnyítését is szolgálja a rendszer, hiszen a korábbi gyakorlattól eltérően, pl. a Központi Statisztikai Hivatal részére szolgáltatott adatokat nem papírokból kell kikeresgélni, hanem akár azonnal rendelkezésre állnak. Ezen kívül különböző pályázatokhoz is egyszerűen rendelkezésre állnak az adatok, melyekkel a pályázati programot, annak szükségességét könnyebben alá lehet támasztani.

## **2.2 A rendszer funkcionális követelményei**

A funkcionális követelményeket két részre kell bontanunk, mert függ attól, hogy a felhasználó adminisztrátorként vagy gondozóként használja a rendszert. A használati eset diagramokat a függelékben helyeztem el.

### **2.2.1 Az adminisztrátor funkcionális követelményei**

Az adminisztrátornak négy tevékenységi köre van. Be tud jelentkezni a rendszerbe, felhasználót tud hozzáadni a rendszerhez, felhasználót tud törölni a rendszerből és ki tud lépni. Ezek közül a felhasználó hozzáadását és törlését emelném ki.

A felhasználó regisztrálásánál az adminisztrátornak bejelentkezett állapotban kell lennie, majd a felhasználó regisztrálása menüpont kiválasztása után megadja a felhasználó nevét és jelszavát majd regisztrálja a felhasználót. Ha a felhasználó már regisztrálva van, akkor a felhasználó hiba üzenettel tér vissza.

A felhasználó törlésénél szintén bejelentkezett állapotban kell lenni az adminisztrátornak, a felhasználó törlése menüpont kiválasztása után a felhasználó nevének megadásával törölheti a felhasználót a rendszerből.

## 2.2.2 A gondozó funkcionális követelményei

A gondozó foglalkozik a hajléktalan adatainak kezelésével és azok karbantartásával.

A gondozó funkcionális követelményei:

- Bejelentkezés
- Hajléktalan hozzáadása
- Hajléktalan keresése
- Hajléktalan adatainak módosítása
- Hajléktalan törlése
- Gondozási lap hozzáadása
- Szerződéskötés
- Statisztikák

A gondozó be tud jelentkezni a rendszerbe, abban az esetben, ha az adminisztrátor előtte már regisztrálta. Legelső bejelentkezéskor a rendszer kéri a jelszava megváltoztatását. Minden más esethez bejelentkezett állapotban kell lenni a gondozónak.

A hajléktalan hozzáadása egy űrlapon történik, amely űrlap kitöltése után a rendszerbe bekerülnek a hajléktalan adatai. Ha a hajléktalan már benne van a rendszerben, akkor a program hibüzenettel tér vissza.

A keresésnél a gondozó kiválaszthatja, hogy mi alapján szeretne keresni, megadja ezt az adatot és a rendszer kilistázza a keresési kritériumnak megfelelő személyek adatait.

Adatmódosításkor a rendszer kéri a hajléktalan bizonyos adatait, mely alapján azonosítja a módosítandó személyt. Ha az adatbázisban nincs ilyen személy, akkor hiba üzenettel tér vissza, ha talál megfelelő személyt, akkor az adatbázisban szereplő adatokkal kitöltött űrlapot jelenít meg, melyen a gondozó megváltoztathatja a módosult adatokat, majd a rendszer frissíti azokat.

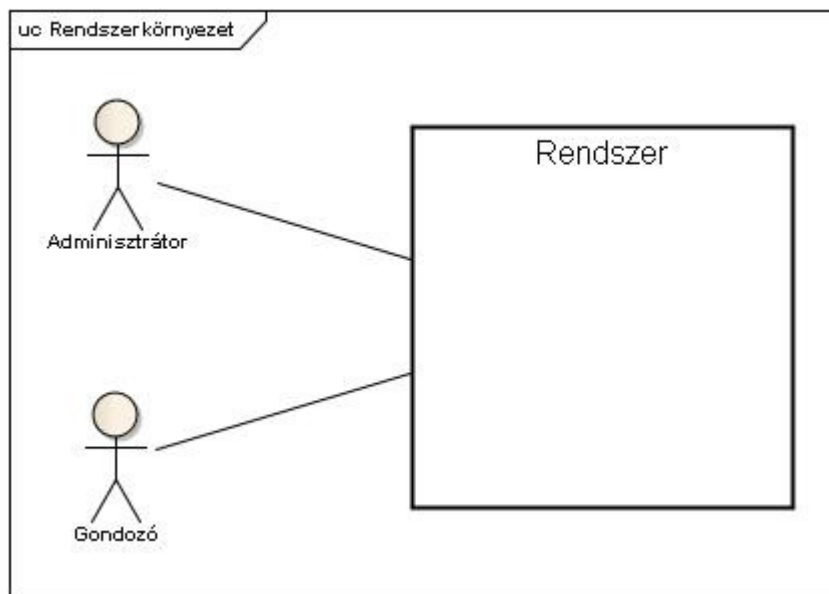
Gondozási lapot csak olyan hajléktalanhoz rendelhetünk hozzá, akinél az adataiban szerepel, hogy gondozásra szorul. Ekkor a gondozó kap egy űrlapot, melyet kitöltve az adatok bekerülnek a rendszerbe.

A szerződés kötésnek kétfajta esete van, az egyik ha új szerződést akarunk kötni a másik, ha egy meglévő szerződést akarunk lezárni. A lezárást csak olyan hajléktalannál kezdeményezhetjük, akinek van aktív szerződése, míg aktív szerződéssel rendelkező hajléktalannál nem köthetünk új szerződést addig, amíg a régit le nem zártuk.

A statisztikákat a gondozó a megfelelő menüpont kiválasztása után éri el, ahol a program kilistázza a fő statisztikai adatokat. Ez megkönnyíti a kifelé történő adatszolgáltatást, főleg a Központi Statisztikai Hivatal részére, mert a statisztikai adatokat egyszerűen elérhetővé teszi.

## 2.3 A rendszer környezete

A rendszer az adatbázis és a felhasználók között áll, és felületet biztosít a felhasználóknak az adatbázis eléréséhez.



1. ábra

A rendszer két felhasználói típust különböztet meg az adminisztrátort, aki a gondozókat tudja a rendszerhez hozzáadni, vagy eltávolítani. A másik felhasználói típus a gondozó, aki a rendszert használó hajléktalanellátó központ alkalmazottja, és aki a hajléktalanok adatait kezeli.

## 2.4 Adatbázis terv

A rendszer által használt adatbázis egy relációs adatbázis. A rendszer fejlesztésekor az Oracle 11g adatbázist használtam.

Adatbázis séma:

1. Tábla: **Alapadatok** (Ez tartalmazza az ellátott személyek alapadatait.)
  - **Azonosító** – elsődleges kulcs.
  - **Nev** – az ellátott személy neve.

- Szuletesi\_nev – nőknél a leánykori név, míg férfiaknál ez általában azonos a névvel.
- Neme
- Szul\_hely – az ellátott születési helye.
- Szul\_ido – a születés évszáma.
- Anyja\_neve
- Apja\_neve
- Csaladi\_allapot – az ellátott családi állapota.
- Kivel\_el
- Miota\_hajlektalan
- Felvetel\_datuma – az az időpont, amikor a gondozott, ellátott bekerül a rendszerbe.
- Utolso\_talalkozas – az utolsó személyes kontaktus az ellátott személlyel.

2. Tábla: Azonosítók (Az ügyfelek azonosítóit tartalmazza.)

- Azonosito – külső kulcs, az alapadatok tábla elsődleges kulcsára mutat.
- Taj\_szam – az ellátott taj száma ha van neki.
- Nygd\_tsz – az ellátott személy nyugdíjbiztosítási száma.
- Adoszam – az ellátott személy adószáma.
- Sz\_ig\_sz – az ellátott személy személyi igazolványának száma.

3. Tábla: Egeszsegugyi\_adatok ( Az ügyfelek egészségügyi adatai )

- Azonosito – külső kulcs az alapadatok tábla elsődleges kulcsára mutat.
- Gondozasi\_naplo – ha az ellátott személy rendszeres gondozásra szorul, akkor szükséges a megléte, ami itt van jelölve, hogy van vagy nincs.
- Kornyezet\_tanulmany – itt van megadva, hogy készült-e környezet-tanulmány a személy környezetéről.
- Mikor\_keszult\_az\_utolso – a környezettanulmány készítésének időpontja

- Hazi orvos neve címe – ha ismert az ellátott személy orvosa, itt tárolhatóak az orvos adatai.
- Ellátott személy állapota – ezt általában az orvos határozza meg, ez a személy állapotára vonatkozik.
- Orvosi javaslat – ha orvosi kezelésre szorul, akkor az orvos által megadott kezelések itt vannak tárolva.
- Utolsó vizsgálat – a személy utolsó orvosi vizsgálatának időpontja.

4. Tábla: Gondozasi\_naplo (csak akkor szükséges, ha a személy gondozásra szorul.)

- Azonosito – külső kulcs az alapadatok tábla azonosítójára hivatkozik .
- Datum – a kezelés időpontja.
- Tevekenyseg – a kezelés megnevezése.

5. Tábla: Szuresek\_ideje (a kötelező szűrővizsgálatok időpontjai.)

- Azonosito – külső kulcs az alapadatok tábla azonosítójára hivatkozik.
- Tudoszures\_idopontja – az utolsó tüdőszűrés időpontja.
- ANTSZ\_szures\_idopontja – az utolsó ANTSZ szűrés időpontja.

6. Tábla: Iskolazottsag (Ellátott személy iskolai végzettsége és jövedelme.)

- Azonosito – külső kulcs az alapadatok tábla elsődleges kulcsára mutat.
- Iskolai\_vegzettseg\_fokozata – a személy iskolai végzettsége.
- Szakma – a személy által megszerzett szakmák.
- Rendszeres\_jov\_tipusa – ha részesül rendszeres jövedelemben, itt tárolódik, hogy az milyen típusú jövedelem.
- Havi\_osszege – rendszeres jövedelem havi összege.
- Egyeb\_jov\_tipusa – ha egyéb juttatásokat is kap, itt adható meg, hogy az milyen típusú.
- Egyeb\_havi\_osszege – egyéb juttatások havi összege.
- Jovedelem\_osszesen – a személy által kapott összes jövedelem.

- **Munkakepesseg\_csokkenes** – a rokkantság %-ban megadott értéke.
7. Tábla: **Pontos\_tartozkodasi\_hely** (A személy állandó tartózkodási helye.)
- **Azonosito** – külső kulcs, az alapadatok tábla elsődleges kulcsára mutat.
  - **Hely\_azonosito** – elsődleges kulcs, a pontos tartózkodási hely azonosítója.
  - **Varos** – az ellátott személy mely városban található.
  - **Utca** – a városon belül mely utcában,
  - **Hsz** – milyen házszám alatt vagy
  - **Hrsz** - milyen helyrajzszám alatt található meg.
  - **Lakhely\_tipusa** – lakás, nem lakás vagy közterület a személy tartózkodási helye.
  - **Altipus** – a lakhely típusának pontosabb megadása.
8. Tábla: **Kozmu** (Az ellátott személy tartózkodási helyén milyen közművek érhetőek el.)
- **Azonosito** – külső kulcs, a pontos tartózkodási hely tábla elsődleges kulcsára mutat.
  - **Víz** – van-e ivóvíz a tartózkodási helyén.
  - **Viz\_milyen** – a víz típusa (vezetékes vagy más).
  - **Gaz** – van-e gázvezeték a tartózkodási helyén.
  - **Futes** – rendelkezik-e valamilyen fűtéssel.
  - **Futes\_milyen** – ha rendelkezik fűtéssel, akkor annak típusa.
  - **Villany** – van-e tartózkodási helyén elektromos hálózat kiépítve.
9. Tábla: **Szerzodes** (A hajléktalan és a hajléktalanellátó központ között megkötött szerződés.)
- **Azonosito** – külső kulcs, az alapadatok tábla elsődleges kulcsára mutat.
  - **Szerzodes\_azonosito** – elsődleges kulcs, a szerződés azonosító száma.
  - **Celja** – a szerződésben megkötött célkitűzés.
  - **Mit\_vallal\_a\_kliens** – a cél elérése érdekében a kliens vállalása.

- `Mit_vallal_a_szocialis_munkas` – a hajléktalan ellátó központ munkatársai által vállalt feladatok.
- `Szerzodes_letrejottenek_ideje` – a szerződés megkötésének dátuma.

10. Tábla: `Eset_lezarasa` (Teljesült vagy felbontott szerződések lezárása.)

- `Azonosito` – külső kulcs, a szerződés tábla elsődleges kulcsára mutat.
- `Teljesult_e_a_szerzodes` – a szerződés teljesült-e vagy nem.
- `Miert_nem_teljesult` – ha nem teljesült itt az indoklás, hogy miért nem teljesült.
- `Felbontasanak_oka` – a szerződés lezárásának vagy felbontásának oka.

11. Tábla: `Dolgozok` (A rendszert használó felhasználók adatai.)

- `Nev` – elsődleges kulcs, a rendszert használó személy felhasználó neve.
- `Jelszo` – a rendszert használó személy jelszava.
- `Statusz` – a felhasználó milyen minőségben használja a programot, adminisztrátorként vagy gondozóként.
- `Belepesszam` – a felhasználó hányszor lépett be a rendszerbe.

Hajléktalanok esetében általában nem áll rendelkezésre minden adat, emiatt az adatbázist úgy kellett felépíteni, hogy a ki nem töltött adatokat is fel lehessen vinni.

## 2.5 Rendszer tervezése

Rendszertervezéskor a követelményekből próbáljuk meg valamilyen minta szerint összeállítani a rendszernek egy absztrakt modelljét. Majd ezt a modellt felhasználva implementálni a szoftvert.

A rendszert a tervezéskor két oldalról próbáltam meg felépíteni a felhasználói felületekről és a rendszer kommunikációjáról az adatbázissal. Ez a felbontás a programban is kitűnik, mivel a program felépítésében is ezt az elvet próbáltam meg követni. Konkrét tervezési mintát nem alkalmaztam, a tervezéskor legfőképpen azért nem, mert a szoftver mérete ezt nem indokolta.

A program struktúráját, úgy próbáltam meg felépíteni, hogy a felhasználói felületek elkülönüljenek a rendszer magjától, azonban a szoftver írásakor az üzleti logika nagy része a felhasználói felületekbe került beépítésre. Bár ez a megoldás nem a legegyszerűsebb, ugyanakkor programozás közben ez tűnt a megfelelő választásnak.

### **2.5.1 Rendszer csomag**

Ebben a csomagban valósítottam meg a legalapvetőbb adatbázis műveleteket, valamint ez a csomag tartalmazza az adatbázis csatlakozáshoz szükséges osztályokat.

Osztályai:

- Users: Ez az osztály a kezelő és adminisztrátor osztályok őszintája.
- Gondozó: Ennek az osztálynak a példányosításával hozható létre kezelő szerepkörű felhasználó.
- Adminisztrátor: Ez az osztály tartalmazza az adminisztrátor feladatköreit.
- Rendszer: Ez az osztály tartalmazza az adatbázisba való bejelentkezést, valamint az adatbázishoz való csatlakozást és az adatbázis lezárását.
- TableCreator: Ez az osztály ellenőrzi az adatbázis csatlakozáskor, hogy az adatbázis táblák léteznek-e. Ha a táblák nem léteznek, ezzel az osztállyal hozható létre az adatbázis séma.

A további osztályok a csomagban adattárolási célt szolgálnak, valamint ezek az osztályok valósítják meg az adatbázis táblákba való beszúrást, törlést és frissítést. A csomag osztály diagramja a 2. ábrán látható.

### **2.5.2. Felületek csomag**

A felületek csomag osztályai valósítják meg a felhasználói felületeket és ezekbe az osztályokba került megvalósításra a kezelő minősítésű, felhasználó által végezhető műveletek.



## 3. Felhasználói kézikönyv

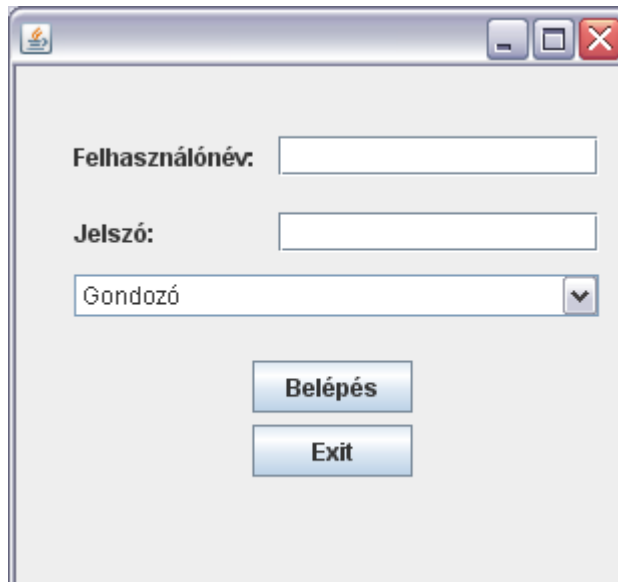
### 3.1 A program indítását megelőző teendők

Ahhoz, hogy a programot futtatni tudjuk, szükséges a Java futtató környezet megléte valamint internet kapcsolat. A program futtatásához 6-os Java futtató környezet szükséges. A futtató környezet legfrissebb verziója letölthető a következő internet címről: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

### 3.2 A program indítása

A program indításakor a rendszer csatlakozik az adatbázishoz, majd automatikusan ellenőrzi, hogy az adatbázistáblák léteznek-e. Ha nem léteznek, akkor automatikusan létrehozza őket és létrehoz egy adminisztrátori felhasználót, admin felhasználónévvel és jelszóval.

### 3.3 Bejelentkezés



The image shows a Java Swing window titled "Bejelentkezés" (Login). It contains three input fields: "Felhasználónév:" (Username), "Jelszó:" (Password), and "Gondozó" (Role) with a dropdown arrow. Below the fields are two buttons: "Belépés" (Login) and "Exit". The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

3. ábra: Bejelentkezés

A program indításakor a bejelentkező képernyő jelenik meg (3. ábra). Első indításkor csak az admin felhasználó névvel és jelszóval tudunk bejelentkezni, mint adminisztrátor. Ha már vannak hozzáadott felhasználók, akkor bármely felhasználó felhasználónevével és jelszavával valamint státuszának kiválasztásával be tudunk jelentkezni. Bejelentkezés után a program a bejelentkezett felhasználónak megfelelő menüt jeleníti meg.

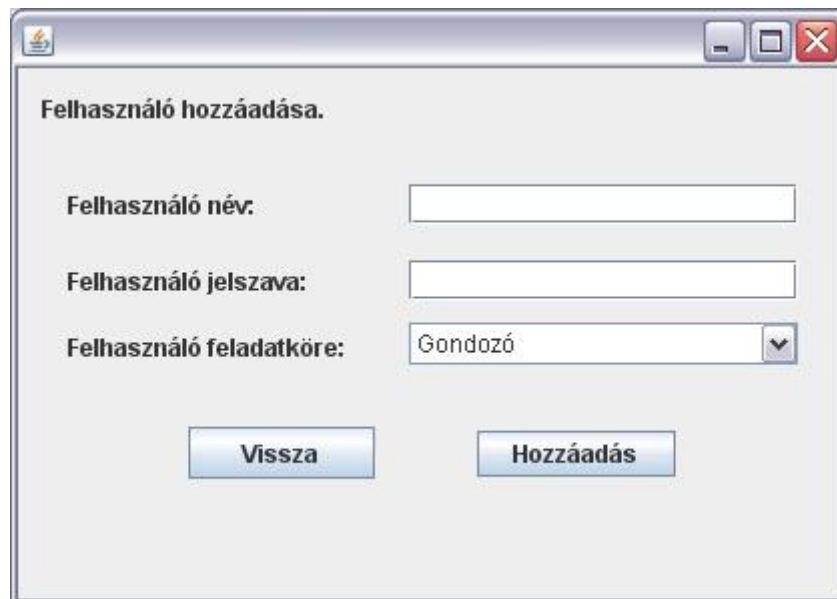
### 3.3.1 Adminisztrátor által végezhető feladatok

Adminisztrátorként két feladatot tudunk elvégezni. Ezek a felhasználó hozzáadása valamint a felhasználó törlése.



4. ábra: Adminisztrátor menü

### Felhasználó hozzáadása:



Felhasználó hozzáadása.

Felhasználó név:

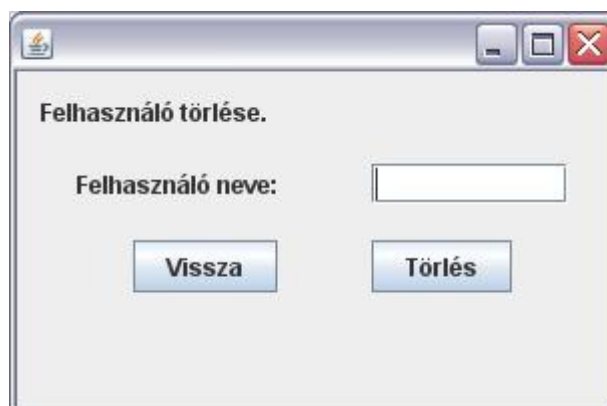
Felhasználó jelszava:

Felhasználó feladatköre: Gondozó

5. ábra: Felhasználó hozzáadása

Ha az adminisztrátor felhasználó hozzáadása menüpontot választja, akkor a 5. ábrán látható felület jelenik meg, melyben megadhatja a felhasználó nevét, egy kezdeti jelszót valamint hogy gondozóként vagy adminisztrátorként regisztrálja az illetőt. Majd a hozzáadás gomb megnyomásával az illető bekerül a rendszerbe és beléphet a programba. A vissza gomb megnyomásával az adminisztrátor visszajut a menübe.

### Felhasználó törlése:



Felhasználó törlése.

Felhasználó neve:

6. ábra: Felhasználó törlése

A felhasználó törlése nagyon egyszerű, mert elég a felhasználó nevét megadni, valamint a törlés gombra kattintani és a felhasználó törölve lesz a rendszerből. A vissza gomb itt is a menübe juttat vissza.

Mindkét esetben egyszerre csak egy felhasználót kezelhetünk, azaz sikeres hozzáadás vagy törlés után a menübe jutunk vissza.

### 3.3.2 A gondozó által végezhető feladatok

Gondozóként van lehetőség a gondozottak adatainak kezelésére. A menüből választhatja ki a gondozó, hogy milyen feladatot szeretne elvégezni a programban.



7. ábra: Gondozó menü

Azaz új gondozottat akar felvenni, keresni szeretne, meglévő gondozottnak az adatait szeretné módosítani, vagy gondozottat akar törölni. Ezen kívül lehetősége van még gondozási lap hozzáadásához azon gondozottnaknál, akiknél betegségükből adódóan állandó gondozás miatt már van gondozási napló. Lehetősége van még a gondozottakkal szerződést kötni vagy meglévő szerződéseket lezárni és statisztikai adatokat lekérdezni.



A 8. ábra a gondozottak adatainak felvitelére szolgáló űrlap, amelyen a gondozott legfontosabb adatait tudjuk felvinni. Ezt az űrlapot a menüből a hajléktalan hozzáadása menüpont kiválasztása után érjük el. Az azonosítókat az alapadatoknál és a tartózkodási helynél a program maga szolgáltatja és azt a kitöltő személy nem tudja megváltoztatni. Az űrlapon azon mezők közül amelyeknek nincs alapértelmezett értéke, bármely mező üresen maradhat. Azonban az alapértelmezett értékű mezők kitöltése sem kötelező, hiszen nem mindig vannak meg a kitöltéshez szükséges adatok, vagy az értékük az alapértelmezett értékkel megegyezik. Abban az esetben, ha a gondozott személyről nem készült környezettanulmány, az űrlapon alatta szereplő dátum mezőkben a dátumot nem tudjuk megadni. Az űrlapon szereplő egymás mellett levő radioButton-ok közül csak az egyik jelölhető ki.

Az összes jövedelmet a program a két jövedelem havi összegéből számítja ki és azonnal megjeleníti az űrlapon. Az űrlap kitöltése után a felvitel gombra kattintva az adatok bekerülnek az adatbázis megfelelő tábláiba és a gondozó visszakérül a menübe. A menübe való visszatérés után, a hajléktalan hozzáadása menüpont újbóli kiválasztásával tudunk új személyt felvenni a rendszerbe. A vissza gomb csak visszaléptet a menübe.

A dátumok közül az 1900. január 1. az alapértelmezett, amely ugyan bekerül az adatbázisba, de ezzel a dátummal művelet nem végezhető, azaz nem lehet rákeresni. Abban az esetben, ha a gondozott személy már benne van a rendszerben, akkor a program hibaiüzenetet ad, mégpedig „A hajléktalan már benne van a rendszerben!” üzenettel.

### **Hajléktalan adatainak módosítása:**

A gondozott adatainak módosítása ugyanezen űrlap segítségével végezhető, mely űrlap a menüből az adatmódosítás menüpont kiválasztásával érhető el. Az adatmódosítás űrlapja annyiban különbözik az adatfelvétel űrlapjától, hogy mielőtt elérnénk, meg kell adnunk, kinek az adatait akarjuk módosítani, ehhez elég a személy nevét, születési helyét és születési idejét megadni, majd a tovább gombra kattintva megjelenik az űrlap a gondozott eddigi adataival kitöltve. Abban az esetben, ha a rendszerben nem találja a gondozottat, a „Nincs ilyen gondozott!” hibaiüzenetet jeleníti meg. Ezen adatokat módosítani tudjuk, majd a módosítás gombra kattintva az adatok az adatbázisban is frissülnek. A módosítás elvégzése után a program szintén visszatér a menübe és az adatmódosítás gomb újbóli kiválasztása után van

lehetőségünk újabb személy adatainak a módosítására az előzőekhez hasonlóan. Ez a menüpont nagyon fontos, hiszen a hajléktalanok életmódjából adódóan többször elveszítik, elhagyják irataikat, ami miatt például személyi igazolvány számuk mindig megújul. A másik adat, ami folyamatosan változhat, főleg az utcán, közterületen élő hajléktalanok körében a tartózkodási hely.

### Gondozott adatainak keresése:

Név	Születési idő	Taj szám	iskolai végzettség	tudószűrés	ANTSZ szűrés	utolsó találkozás	Tartózkodási hely
-----	---------------	----------	--------------------	------------	--------------	-------------------	-------------------

9. ábra: Keresés

A keresés a menüből a hajléktalan keresése menüpont kiválasztásával érhető el. A menüpont kiválasztása után a fenti képernyőre jutunk, ahol külön kereshetünk a névre, születési időre, TAJ.számra, iskolázottságra stb. adatokra. Egyszerre csak egyfajta adatra kereshetünk rá, méghozzá úgy, hogy azt az adatot adjuk meg, majd a keresés gombra kattintva a program a képernyő alsó részén látható mezőben táblázatos módon adja vissza az adatokat. A tartózkodási helyre való keresés talán a legfontosabb opció, mivel egy esetleges riasztásnál, hogy valahol ismeretlen személyt találtak, kihűlt, nincs hol aludnia. Ha a gondozó rá tud keresni a tartózkodási helyre, akkor már láthatja, hogy esetleg kiről is lehet szó, ami sok esetben többlet információ lehet a további tevékenységhez, megoldáshoz.

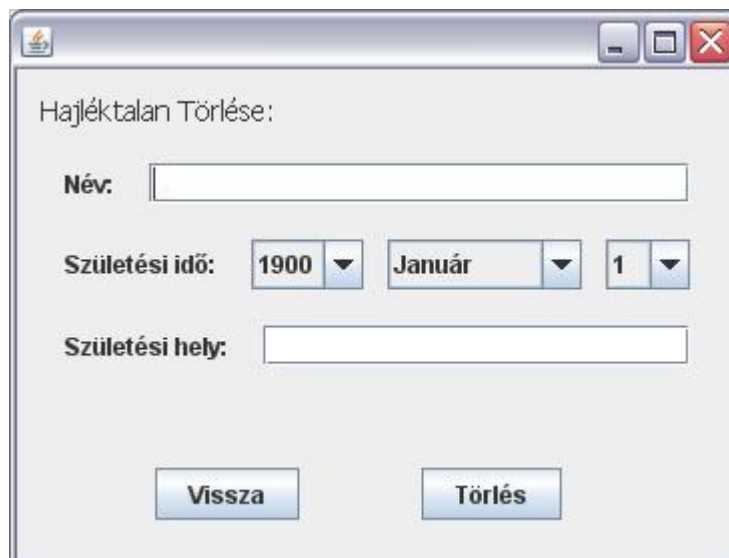
A névre való kereséskor lehetőségünk van csak név részletek keresésére, azaz nem szükséges a gondozott teljes nevét beírni, elég ha csak annak egy részét pl. csak vezetéknév, keresztnév vagy akár csak egy betűt adunk meg.

Az iskolázottságra való kereséskor is lehetőségünk van erre az opcióra.

Keresésnél figyelni kell arra, hogy a dátumoknál az 1900.-as év dátumaira nem tudunk keresni, mert az a dátum mezők alapértelmezett értéke.

Ennél az opciónál mivel megjelenítjük a keresés eredményét, így a gondozó nem tér vissza azonnal a menübe, valamint egyszerre akár többször egymás után is kereshet. Ebben az esetben a keresés gomb megnyomása után a táblázat törlődik, majd az új adatokkal töltődik fel. A keresés végeztével a menübe az ok gomb lenyomásával tudunk visszatérni. Ha a keresési kritériumnak nincs megfelelő személy a rendszerben, akkor a program a „Nincs a keresésnek megfelelő gondozott az adatbázisban.” hiba üzenetet szolgáltatja. Abban az esetben, ha nem éri el az adatbázist, vagy valamilyen probléma lép fel az adatbázissal való kommunikáció során, az „Adatbázis hiba történt!” hibáüzenettel tér vissza.

### **Gondozott adatainak törlése:**



Hajléktalan Törlése:

Név:

Születési idő: 1900  Január  1

Születési hely:

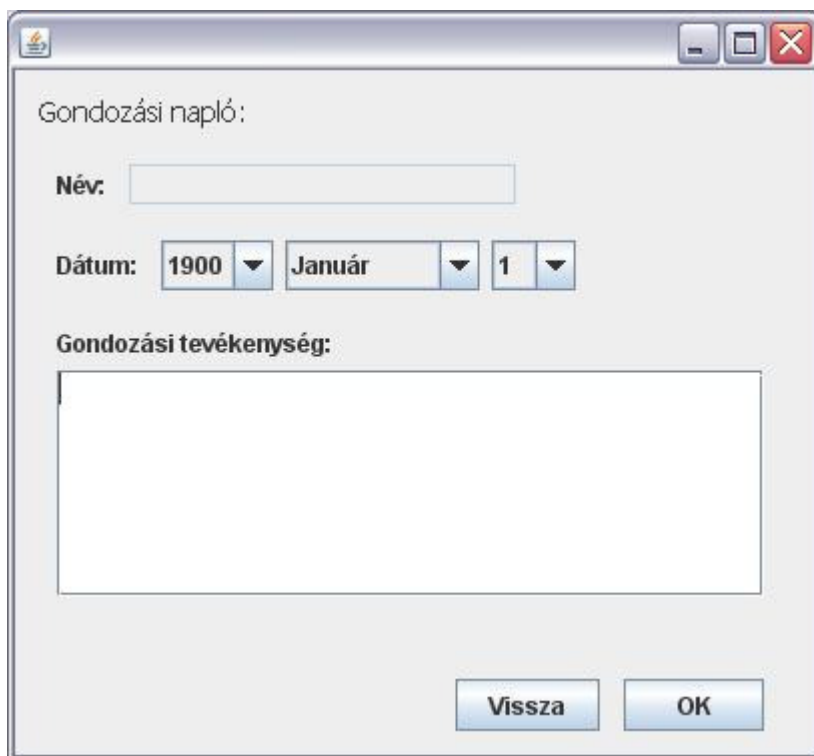
10. ábra: Hajléktalan törlése

Gondozott törlését a hajléktalan törlése menüpont kiválasztásával érhetjük el, ekkor egy hasonló képernyőt kapunk, mint az adatszűrés lekérésekor. Az adatok megadása után a törlés gombra kattintva a gondozott adatai törlődnek a rendszerből, beleértve a

szerződéseket, azok lezárását valamint, - ha a gondozott egészségügyi ellátásra szorult -, a gondozási naplót is. Sikeres törlés esetén a program a „Törlés sikerült!” üzenetet írja ki. Amennyiben az adatbázis nem tartalmazza a gondozottat, a „Nincs ilyen gondozott!” hiba üzenettel tér vissza, akkor ha a törlés nem sikerül a program a „Törlés sikertelen!” hiba üzenetet írja ki a képernyőre. A menübe a vissza gomb megnyomásával léphetünk vissza.

### **Gondozási napló kitöltése:**

A megfelelő menüpont kiválasztása után egy adatlekérő űrlapra jutunk, amelyen azon személy adatait adhatjuk meg, akihez a gondozási naplóját ki akarjuk tölteni. Ekkor a rendszer ellenőrzi, hogy a személy benne van-e az adatbázisba és hogy rendelkezik-e gondozási naplóval. Ha nem rendelkezik, vagy nincs benne, az adatbázisban a rendszer hiba üzenettel tér vissza.

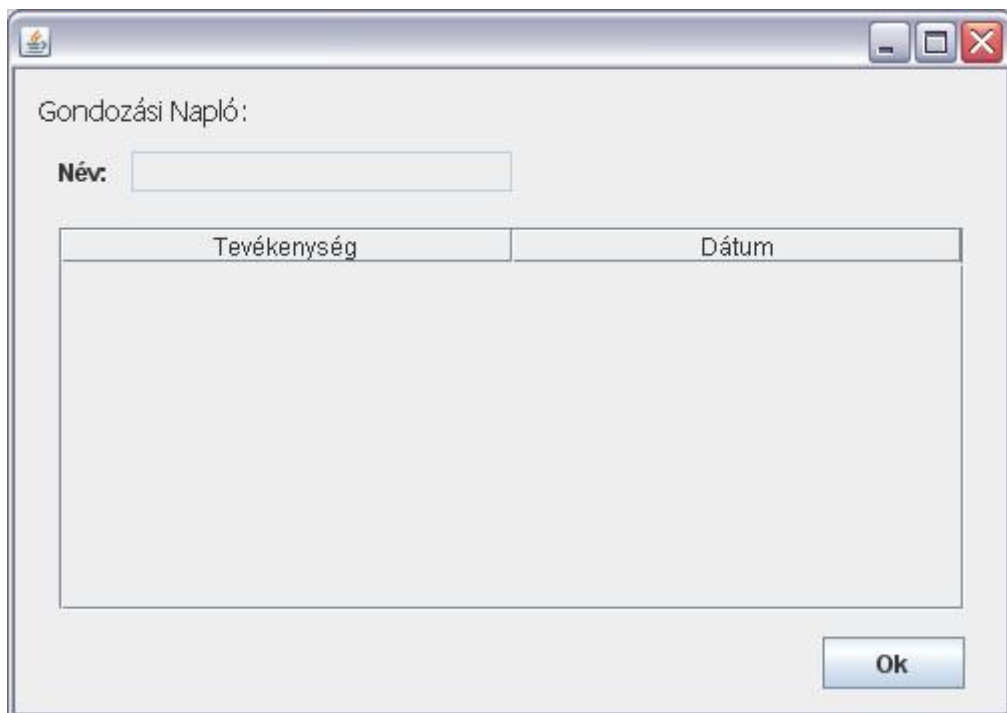


The screenshot shows a standard Windows-style dialog box with a title bar containing a small icon and window control buttons (minimize, maximize, close). The main area is titled "Gondozási napló:". Below the title, there is a text input field labeled "Név:". Underneath that is a date selection section labeled "Dátum:" with three dropdown menus: the first is set to "1900", the second to "Január", and the third to "1". Below the date fields is a large, empty text area labeled "Gondozási tevékenység:". At the bottom right of the dialog, there are two buttons: "Vissza" and "OK".

11. ábra: Gondozási napló hozzáadása

Amennyiben a gondozott rendelkezik gondozási naplóval, a 11. ábra képernyője jelenik meg, ahol a nevet a program beírja és azonos a gondozott nevével. A dátum a gondozás idejének dátuma, míg a gondozási tevékenységet a szöveg mező kitöltésével tudjuk megadni. Az OK gomb megnyomására az adatbázisba bekerül a gondozási napló bejegyzése. Amit a program „A beszúrássikerült.” üzenet megjelenítésével nyugtáz. Ha bármilyen hiba történik „A beszúrássikertelen.” üzenet jelenik meg.

Ugyanazon az adatlekérő űrlapon, választhatjuk ki egy már létező napló megtekintését, ha a gondozott rendelkezik gondozási naplóval. Ekkor a 12. ábrán látható képernyőre jutunk, ahol a táblázatban láthatjuk a gondozási tevékenységet és annak dátumát a táblázatban. A táblázat felett a gondozott neve jelenik meg.



Tevékenység	Dátum
-------------	-------

12. ábra: Gondozási napló megtekintése

## Szerződéskötés és esetlezárás:



Ügyfél adatai:

Név:

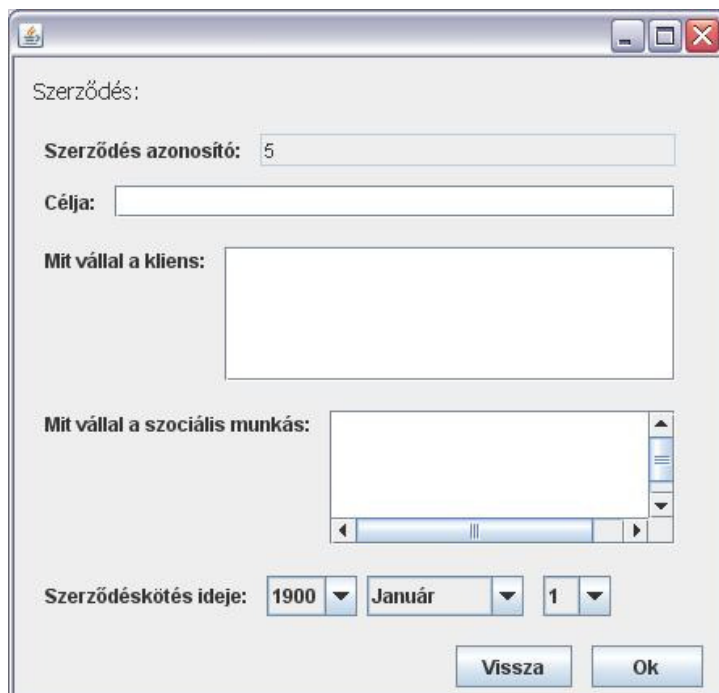
Születési hely:

Születési idő: 1900 ▼ Január ▼ 1 ▼

Vissza Szerződések megtekintése Eset lezárása Szerződés

13. ábra: Szerződéskezelés adatlekérése

A szerződéskötés menüpont kiválasztásával 13. ábrán látható képernyőhöz jutunk, melyen az ügyfél adatainak megadása után kiválaszthatjuk, hogy új szerződést akarunk vele kötni, vagy a régi szerződését akarjuk lezárni. Abban az esetben, ha új szerződést akarunk kötni, a rendszer ellenőrzi, hogy ezt megtehetjük-e, azaz hogy nincs-e lezáratlan szerződése a gondozottnak. Ha a gondozott rendelkezik lezáratlan szerződéssel, akkor új szerződést nem köthetünk, amíg azt le nem zárjuk és a program ezt „A gondozottnak van aktív szerződése!” hiba üzenet megjelenítésével tudatja a felhasználóval.



Szerződés:

Szerződés azonosító: 5

Célja:

Mit vállal a kliens:

Mit vállal a szociális munkás:

Szerződéskötés ideje: 1900 ▼ Január ▼ 1 ▼

Vissza Ok

14. ábra: Szerződés kötés

Amennyiben a gondozottnak nincs aktív szerződése, akkor a szerződés gomb megnyomása után a fenti képernyőhöz jutunk. A szerződés azonosítóját a program szolgáltatja, mindig a soron következőt. A szerződésnél megadhatjuk, hogy mi miatt kötjük a szerződést, ez bármi lehet pl. munkahely keresés, lakhatás megoldása, stb. A másik két szövegmezőbe a hajléktalan vagy gondozott személy és a hajléktalan ellátó központ dolgozója által a cél elérése érdekében felvállalt feladatait lehet megadni. A képernyő dátum rovatába beírt idő a szerződéskötés időpontját jelenti.

Abban az esetben, ha a gondozottnak van szerződése és az még nincs lezárva, választhatjuk az esetlezárása gombot, mellyel lezárhatjuk a szerződést.



15. ábra: Eset lezárás

Ekkor a fenti ablak jelenik meg, melyben a szerződés azonosító a lezárandó szerződésnek az azonosító száma. Megadhatjuk, hogy teljesült-e a szerződés. Amennyiben nem teljesült a szerződés célkitűzése, akkor tölthetjük ki a „Miért nem teljesült?” rovatot, valamint megadhatjuk a szerződés lezárásának az okát. Ezután az OK gombra kattintva az adatok bekerülnek az adatbázisba és a szerződés lezárásra kerül, majd visszatérünk a menübe.

Ha a 13. ábrán látható képernyőn az ügyfél adatainak megadása után a szerződés megtekintése gombot választja a gondozó, akkor megtekintheti a gondozott személlyel kötött összes eddigi szerződést és azok lezárását, ha le van zárva.

## **Statisztikák:**

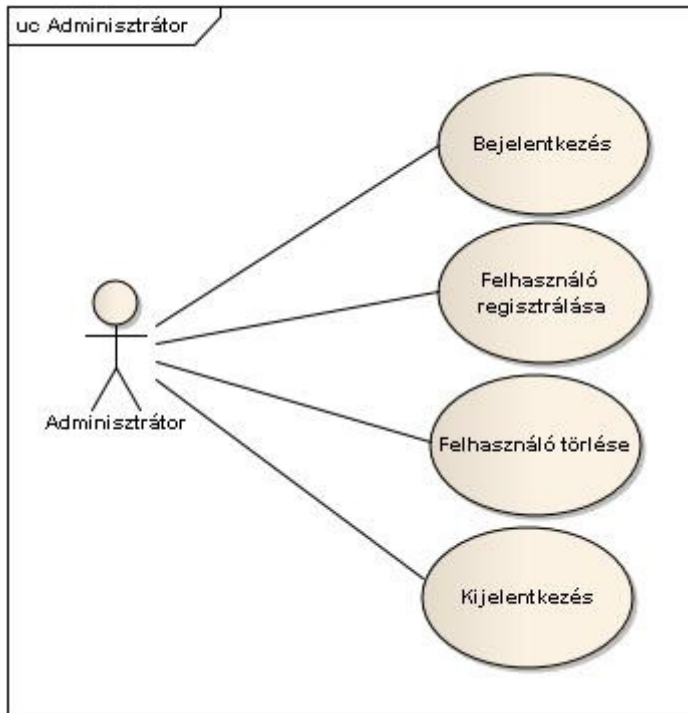
A statisztikákat a menüből a „Statisztikák” menüpontra lehet elérni. Ez a modul az éppen aktuális adatbázis állapot alapján számolja ki a statisztikák értékeit. A program a nemi eloszlás, az életkor szerinti eloszlás tízévenkénti osztásban, iskolai végzettség szerint, családi állapot szerint, tartózkodási hely típusaként és a jövedelemforrás alapján számolt statisztikákat szolgáltatja. A jövedelemforrás statisztikájánál egy személy akár kétszer is bekerülhet abban az esetben, ha rendszeres havi jövedelme és alkalmi jövedelme is van. A statisztikáknál a program nem csak számszerűen, hanem százalékosan is megjeleníti az adatokat, ezzel megkönnyítve a gondozó munkáját és segítve a féléves és éves beszámolók megírását a Központi Statisztikai Hivatal felé.

## 4. Összegzés

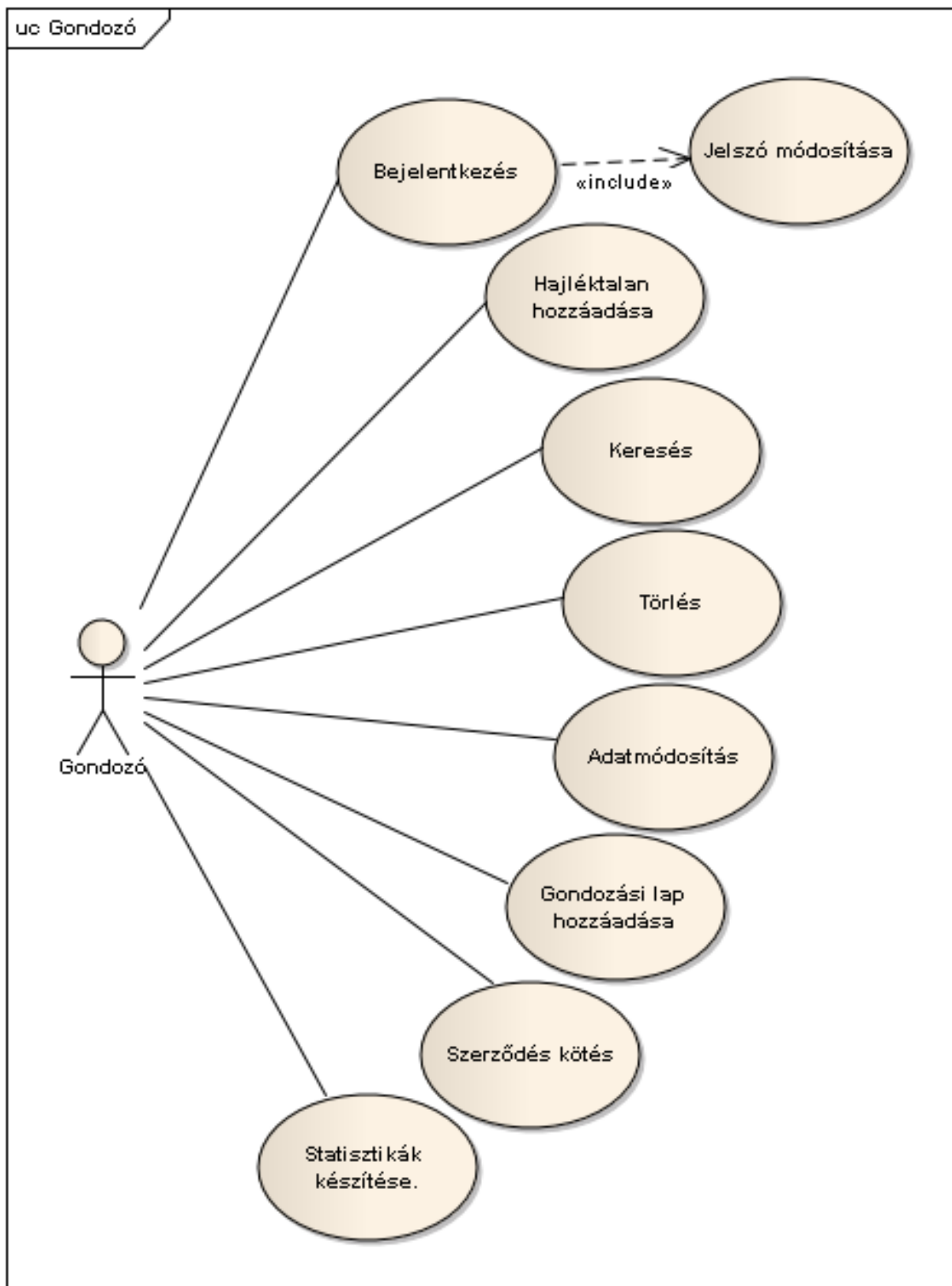
Az online nyilvántartási rendszer továbbra is a szociális munka menetéhez igazodik, a papír alapú dokumentációval összhangba került kialakításra, azonban néhány pontban eltér tőle. Az online program lehetőséget teremt arra, hogy az ügyfelek adatait ill. a velük végzett szociális munkát, minél egyszerűbben rögzíthessék a munkatársak. Az adatfelvitelt a szociális munka menetéhez igazodó felületek kialakításával valamint legördülő listákkal, választó gombokkal segíti. A központi adatbázis lehetőséget nyújt arra, hogy az ellátó központ munkatársai a központ bármely intézményében elérhessék a naprakész adatokat, és bármikor, ha változás történik az adatokban, azonnal tudják azt korrigálni. Valamint egy plusz lehetőségként megjelenik a terepen dolgozók számára, hogy amikor kiszállnak egy hajléktalanhoz, az adatait akár azonnal rögzíthessék a rendszerbe. A program továbbfejlesztésére jó irány lehet az, hogy internet kapcsolat nélkül offline módban is lehessen adatokat rögzíteni, majd amikor rendelkezésre áll internet kapcsolat, az adatbázis szerveren is frissülhessenek az adatok. Természetesen az online nyilvántartás nem fogja megoldani az összes fennálló problémát, de jó alkalmat teremt az eddigi információáramlás hibáinak kiküszöbölésére. Meg kell említenünk azt a lehetőséget is, hogy idővel egy ehhez hasonló rendszer akár ki is válthatja teljesen a jelenleg használt papír alapú nyilvántartást és központosítottan, minden hatóságnak elérhetővé teszi az adatokat.

## Függelék

### Az adminisztrátor use case



## Gondozó use case



## **Irodalomjegyzék**

1. Vég Csaba Instant Java/ Java EE/ SOA I-II
2. Ian Sommerville Szoftverrendszerek fejlesztése
3. <http://www.cab.u-szeged.hu/WWW/java/kiss/article.html>
4. <http://www.cs.ubbcluj.ro/~bittologatok/eloadasok/JavaKurzus/J2EE5.pdf>

## **Köszönetnyilvánítás**

Ezúton szeretném megköszönni a Debreceni Egyetem tanárainak, előadóinak, akik az évek során tudásukat átadták és ezzel hozzájárultak dolgozatom elkészítéséhez.

Külön köszönöm Kollár Lajos tanár úrnak a témavezetőként nyújtott segítségét.