

# Development of the Registry software for the purpose of maintaining the IT equipment of the Szent Pantaleon Hospital

David Fribas <sup>a</sup>

*Britannica International School, Kakukk ut 1-3., Budapest, 1121, Hungary, david.firbas@gmail.com*

---

## Abstract

The article, entitled Development of the Registry software for the purpose of maintaining the IT equipment of the Szent Pantaleon Hospital, is intended for the preparation of the register of the workstations connected to the internal computer network of the Szent Pantaleon Hospital in Dunaújváros. I had to carefully plan the registry system, then implement the program with a suitable programming language. I have divided these articles into several main parts, after presenting the development environment and task specification, I will describe the IT network of the hospital, which helps to present the location of my work. Later, I will design the database that serves as the basis of the registry and implement it in the database manager. I used the popular E/K model for the design, while the database was implemented in the MySQL database system. As a final step, I dealt with the creation of the program itself, which accesses the database using a web interface. I created the registry system using the PHP language embedded in a web page. During the development, I also encountered countless problems, which I tried to solve and to correct new errors. It happened more than once that I had to rework a part of the design because I thought it was better to use a newer solution. I created an early version of the registry system back in 2007, which I further developed in 2014, so the user interface became clearer. With the completion of the plan, a server-based and web-based registry program was implemented, which records the workstations of the hospital, finding the maintenance work internally. The IT department manager decided to use the registry to maintain the network.

*Keywords:* programming; PHP; MySQL

---

## Nyilvántartó szoftver fejlesztése a Szent Pantaleon Kórház informatikai eszközeinek karbantartása céljából

Fribás Dávid <sup>a</sup>

*Britannica International School, Kakukk ut 1-3., Budapest, 1121, Hungary, david.firbas@gmail.com*

---

## Absztrakt

A cikk, mely a Nyilvántartó szoftver fejlesztése a Szent Pantaleon Kórház informatikai eszközeinek karbantartása céljából címet viseli, a dunaújvárosi Szent Pantaleon Kórház belső számítógépes hálózatára kötött munkaállomások nyilvántartásának elkészítését hivatott bemutatni. A nyilvántartó rendszert gondosan meg kellett terveznem, hogy utána egy megfelelő programnyelv kiválasztásával, a programot megvalósítsam. Jelen cikket több fő részre osztottam, a fejlesztői környezet bemutatása és feladat specifikációja után a kórház informatikai hálózatát ismertetem, mely segít bemutatni munkám helyszínét. A későbbiekben a nyilvántartó alapjául szolgáló adatbázist tervezem meg, és az adatbázis-kezelőben megvalósítom. A tervezésre a népszerű E/K modellt használtam fel, míg az adatbázist MySQL adatbázis-rendszerben realizáltam. Végző lépésként magával a program elkészítésével foglalkoztam, amely egy webes felület segítségével fér hozzá a létrehozott adatbázishoz. A nyilvántartó rendszert weblapba ágyazott PHP nyelv segítségével készítettem el. A fejlesztés során számtalan problémával is szembe kerültem, melyet igyekeztem megoldani, és az újabb hibákat kijavítani. Nem egyszer fordult elő, hogy a tervezésnél újra kellett dolgoznom egy részt, mivel jobbnak láttam az újabb megoldást alkalmazni. A nyilvántartó rendszer egy korai verzióját még 2007-ben készítettem el, amit

2014-ben tovább fejlesztettem, így a kezelőfelület is áttekinthetőbb lett. A terv elkészülésével, megvalósításra került egy szerver-alapú és web-es felületű nyilvántartó program, mely a kórház belső munkaállomásait tartja nyilván, ezáltal jelentősen megkönnyítve a karbantartói munkát. Az informatikai osztályvezető úgy döntött, hogy a nyilvántartót használni fogják a hálózat karbantartásához.

*Kulcsszavak: programozás; PHP; MySQL*

---

## 1. Bevezető

Nyilvántartó szoftverünk sikeres megvalósításához elengedhetetlen lesz két szerveralkalmazást és egy fordítóprogramot telepítenünk. Szerver-oldali alkalmazás lévén, szükség van egy web-szerverre, melynél választásom a sokak által legelterjedtebbnek tartott Apache lett. Ugyanezeket a szoftvereket kell majd a cél szerver számítógépre is telepíteni, ahol a nyilvántartó rendszer futni fog:

- Apache webszerver
- MySQL szerver
- PHP fordító

Mindhárom szoftvert a feltelepítés után körültekintően kell bekonfigurálni, de létezik egy alternatíva, mellyel könnyedén tudjuk ezt megvalósítani összesített csomagok révén. Én a XAMPP nevű web szerver csomagot választottam segítségül nyilvántartó programom kifejlesztésére, mely mindhárom általam felsorolt szoftvert pár kattintással feltelepíti, ezzel is időt spórolhatunk meg. Ahhoz, hogy jobban átláthatóvá váljon az adatbázis tervezés, a XAMPP programcsomagban megtalálható az ún. phpMyAdmin is, mely grafikus felületet biztosít számunkra.

## 2. A követelmények feltárása és a feladat specifikációja

### 2.1. A kórház hálózatának története napjainkig

A cikk témája egy webes felületen működő nyilvántartó program megtervezése, mely a dunaújvárosi Szent Pantaleon kórház hálózati eszközeit tartja számon. Mielőtt belemerülnénk a program elkészítésének fázisaiba, szükség van arra, hogy röviden végignézzük a kórház hálózatának fejlődését, és megértsük miért elengedhetetlen ott napjainkban egy ilyen adatbázis létrehozása.

1980-ban alakult meg a kórházban az informatikai osztály, mely akkor még csak egy karbantartót alkalmaztatott. Az osztály feladata, hogy az informatikai, majd később, a hálózati berendezéseket üzemeltesse, illetve karbantartsa. Ekkor még önálló munkaállomásként

működtek a betelepített úgynevezett MOD típusú számítógépek, melyeket 1987-ben felváltottak az XT típusúak.

1988-ban egy nagyobb bővítésen ment keresztül a számítógéppark, és ahogy növekedett a berendezések száma, úgy merült fel egyre jobban az igény egy belső hálózat kialakítására. A hálózat hasznos megoldás, mert:

- az erőforrásokat közösen megoszthatják
- költség-hatékony
- kommunikációs eszközök, gyors információcsere
- megbízhatóság növekedése

A hálózat hasznosságáról oldalakat lehetne még írni, én csak párat említettem a legfőbb tulajdonságaiból.

1990 után gyűlt össze annyi számítógép a kórházban, hogy mindenképp szükségessé tette a belső hálózat létrehozását. Így létrejött az ARCNET típusú helyi hálózat, mely abban az időben közkedvelt volt.

Akkor még csak belső hálózat működött, melynek három darab Novell szerver volt a lelke. Az adatbázis kezelőt Clipper-ben írták meg, amelyek a Novell rendszerére voltak optimalizálva.

1993-94-be újfent nagyobb szabású fejlesztések voltak az informatikai berendezések terén, amikor is pályázat útján új hálózati eszközöket tudott beszerezni a kórház. Felmérték azokat a helyeket az épületen belül, ahol ideális lehet az adatáramlás, így megállapították, hogy 400 darab munkaállomásra lenne szükség. Erre nem volt lehetőség, egyrészt anyagi keret hiányában, másrészt az akkori hálózati technológia mindössze 128 végpontos volt, amire értelemszerűen legfeljebb 128 gépet lehet rákötni. A későbbi fejleszthetőséget figyelembe véve kiépítették a 400 végpontot, amiből csak 128 volt használva. Ezért később szükségessé vált egy ötös szintű ETHERNET hálózat kiépítése.

#### **MAN hálózat (Metropolitan Area Network):**

A kórháznak több telephelye létezik, mit például a Zöld rendelő, melyek túl messze vannak ahhoz, hogy UTP kábelt lehessen használni. A szervezeti egységek közötti kapcsolatra optikai megoldást kellett találni.

1994 decemberében felszereltek egy szünetmentes tápegységet, mely a következő 1999-es nagyobb hálózati fejlesztésig karbantartás nélkül működött.

1999-ben egy újabb közbeszerzési pályázatot hirdetett a kórház teljes körű informatikai felújításra, mely alapjaiban a mai rendszer részét képezi. A nyertes cég kibővítette 250 darab számítógéppel az informatikai parkot, illetve a hálózati berendezéseket is lecserélték. Így az új hálózatnak már 180 működő végpontja volt, melyet a Switch-ekre lecserélt gócpontok kötöttek össze, felváltva ezzel a HUB-okat.

A kórház fő hálózati központjánál lecserélték a kábeleket gigabites sávszélességűekre, majd bevezették az Internet-t TCP/IP protokollon keresztül. A kórház teljesen egészében le lett fedve, és minden munkaállomás egy szerverre, illetve egy adatbázisra dolgozott.

## *2.2. A nyilvántartással kapcsolatos igény felmerülése*

A sok számítógép karbantartása egyre nagyobb problémát okozott, mivel a csekély számú informatikus mérnök lassan tudta csak így ellátni feladatát. A kórházon belüli informatikai számítógéppark az elmúlt évek alatt jelentősen felfejlődött, és a munkaállomások folyamatos üzemeltetése, karbantartása, hibakeresése már nehezen követhető, ha nincsen rendesen nyilvántartva. A hálózat struktúrájáról összeszedett információt felhasználva, egy szoftver jelentősen megkönnyítené a karbantartók munkáját. A könnyen kezelhetőség érdekében egy olyan programot kellett írni, amit nem kell minden egyes munkaállomásra feltelepíteni, hiszen az újfent felesleges munkához vezetne, tekintve a számítógépek darabszámát. A kórház informatikai osztályvezetője egy szerver-oldali program el-készítését javasolta, a továbbiakban ennek a megtervezésével és kivitelezésével foglalkozom.

## **3. Az alkalmazás statikus és dinamikus megtervezése**

A programom elkészítésének kettő főbb fázisa lesz, először megtervezem és kivitelezem az adatbázist, mely tulajdonképpen a nyilvántartás lelke, illetve elkészítem és bemutatom magát a programot, mely az adatbázist fogja lekérdezni, módosítani. A szerver-oldali alkalmazást a PHP programnyelv segítségével fogom megvalósítani, mely a MySQL adatbázis-kezelő szerverhez fog kapcsolódni. A webes programfelülettel részletesen a következő fejezetben fogok foglalkozni, jelen fejezetben csak magára az adatbázis létrehozására térek ki.

Minden adatbázis elkészítése először egy tervezési fázissal kezdődik, melyet úgy hívunk, hogy adatmodellezés. Ebben a szakaszban a feldolgozandó adatok, információk elemzésével, és ezek közötti kapcsolatok meghatározásával foglalkozunk. Az adatmodellezés grafikus megközelítését, az egyed/kapcsolat diagrammot (E/K) használom. (Ullman, J. D., & Widom, J., 2009), Gábor, S. (2005), Nixon, R. (2014).

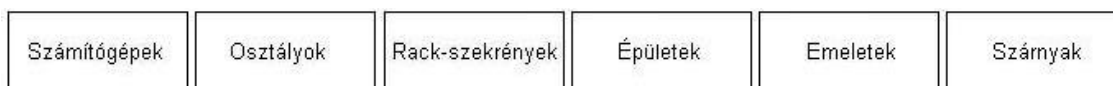
### 3.1. Az E/K modell létrehozása

A kórházban kikötötték, hogy egy olyan adatbázist kell létrehoznom, mely a hálózatra kötött munkaállomásokat tartja nyilván, és ezáltal a következő adatokat kell tartalmaznia:

- a számítógép alapvető tulajdonságait (IP cím, gépnév stb.)
- fizikai hely (hol helyezkedik el a gép fizikailag)
- logikai hely (hol helyezkedik el a gép a hálózatban)
- Rack szekrény (csatlakozási pont)

Az adatok közül a Rack szekrény rövid magyarázatra szorul. A Rack szekrény önmagában Switch-eket és úgynevezett Patch-paneleket tartalmaz. A Switch-ek, mint már korábban írtam, adatáramlást irányító kapcsolók, melyek elosztóként szerepelnek a hálózatban. A Patch-panel egy 12 vagy 24 csatlakozóból álló címkével ellátott panellap, mely nagymértékben segít beazonosítani a szekrénynél, hogy melyik csatlakozóra van rádugva az adott számítógép. Természetesen a két eszköz össze van kötve egymással, de kapcsolatukat az adatbázisunk szempontjából el fogjuk hanyagolni, mivel a hálózat nyilvántartásában nem képviselnek lényeges szerepet, csak és kizárólag a Patch-panel számát tartjuk fontosnak.

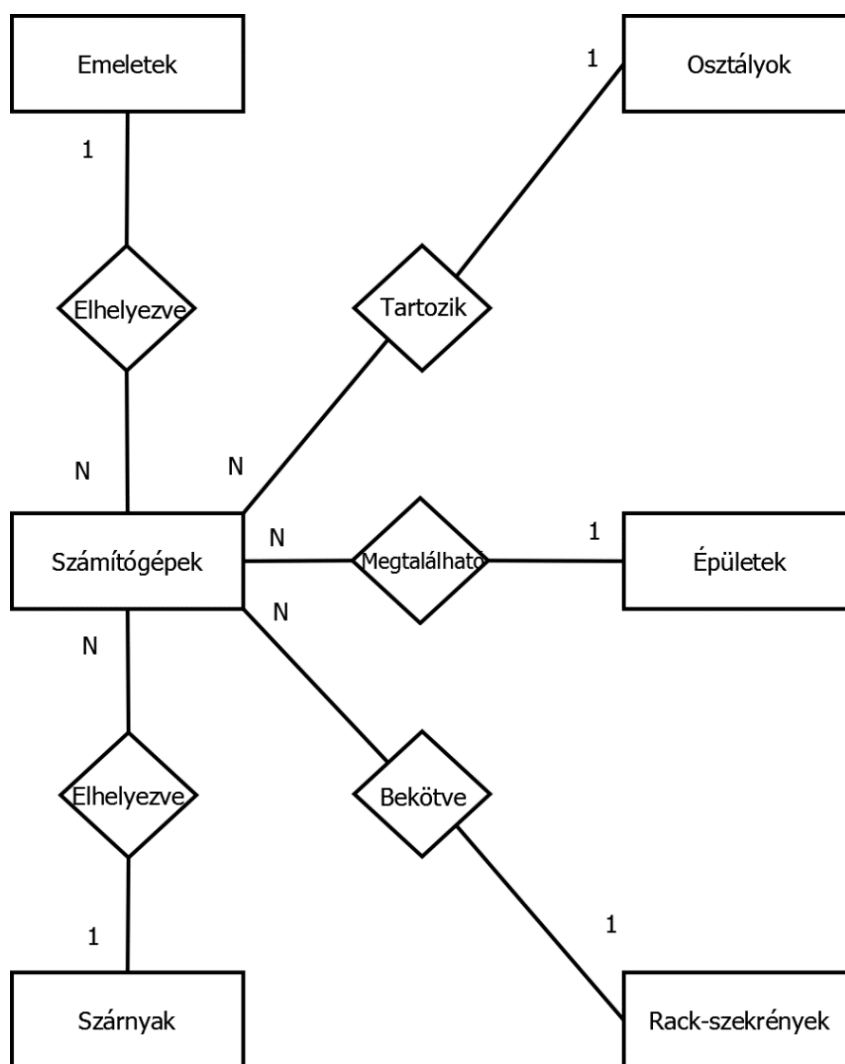
Az elvárások függvényében, és az adatbázis normalizálása után létrehoztam hat darab egyedhalmazt. Az adatok normalizálása azért fontos, hogy elkerüljük a redundanciát. Meg kell említenem, hogy az egyedhalmazok létrehozása másodjára nyerte el ezt a formát, az először meghatározott E/K modell kevésbé működött hatékonyan a PHP program alatt. A létrejött egyedhalmazok:



1. ábra: Az adatbázis egyedhalmazai

Az adatbázis alapját maga a számítógép képezi, az egész információrendszer e köré fog felépülni, ezért a Számítógépek egyed mindenképp szükséges. Az Osztályok tartalmazza magát a számítógépek logikai helyét, az Épületek, Emeletek és Szárnyak pedig a fizikai helyét. Lényeges volt a fizikai helyet tovább bontani épületei, emeletei és szárnyai szerint, mivel ezek az egyedhalmazok nem fognak sok adatot tartalmazni, és a web-es felületen is könnyebben tudunk rájuk hivatkozni adatkeresés, felvétel és módosítás céljából. Végül a Rack-szekrény pedig a jelenlegi hat darab hálózati elosztót fogja tartalmazni.

Most, hogy ismerjük az egyedeket, jelöljük ki a köztük lévő kapcsolatokat. A Számítógépek és a többi egyedhalmaz között sok-1 kapcsolat áll fent, mivel több számítógép tartozhat egy osztályhoz, épülethez, szárnyhoz, és rack-szekrényhez. Az első tervezésnél egy Szoba egyedhalmazt is létrehoztam, de később kiderült, hogy ez csak bonyolítja a rendszert, mivel meglehetősen sok szobát tartalmazhat egy épület, így ez a Számítógépek egy attribútumaként szerepel a végső modellben.



2. ábra: Az adatbázis egyedhalmazai

Ahhoz, hogy teljesen elkészüljön az E/K diagramunk, már csak az egyedhalmazokhoz tartozó tulajdonságokat, illetve kulcsokat kell meghatároznunk. A következőkben előállítom hat táblázatba (1-6. táblázatok), hogy az egyes egyedekhez milyen tulajdonságokat kívánok hozzárendelni. A tulajdonságneveket úgy állítom össze, hogy azokat később a MySQL adat-bázisban is használhassam, tehát ne tartalmazzon ékezeteket, mivel az problémát okozhat egyes rendszereken (például, ha csak angol nyelvű MySQL van telepítve).

1. táblázat A Számítógép egyed attribútumai

SZG_ID	A Számítógép egyed egyértelmű azonosítója, kulcs
IP_CIM	A számítógépek fix IP címe
GEP_NEV	A számítógépek neve
OEP_KOD	A számítógépek kórházi OEP azonosítója
CONTROL_KOD	A számítógépek kórházi CONTROL azonosítója
PATCH_SZAM	A számítógépek Patch-panel száma
SZOBA_SZAM	Szobaszám, ahol el van helyezve a számítógép

2. táblázat Az Osztály egyed attribútumai

O_ID	Az Osztály egyed egyértelmű azonosítója, kulcs
OSZTALY_NEV	Az osztályok neve

3. táblázat A Számítógép egyed attribútumai

RACK_ID	A Rack-szekrények egyed egyértelmű azonosítója, kulcs
RACK_NEV	A Rack-szekrények neve
PATCH_HELY	A Rack-szekrények patch panelén lévő csatlakozók száma

4. táblázat A Számítógép egyed attribútumai

EP_ID	Az Épületek egyed egyértelmű azonosítója, kulcs
EP_NEV	Az épületek neve

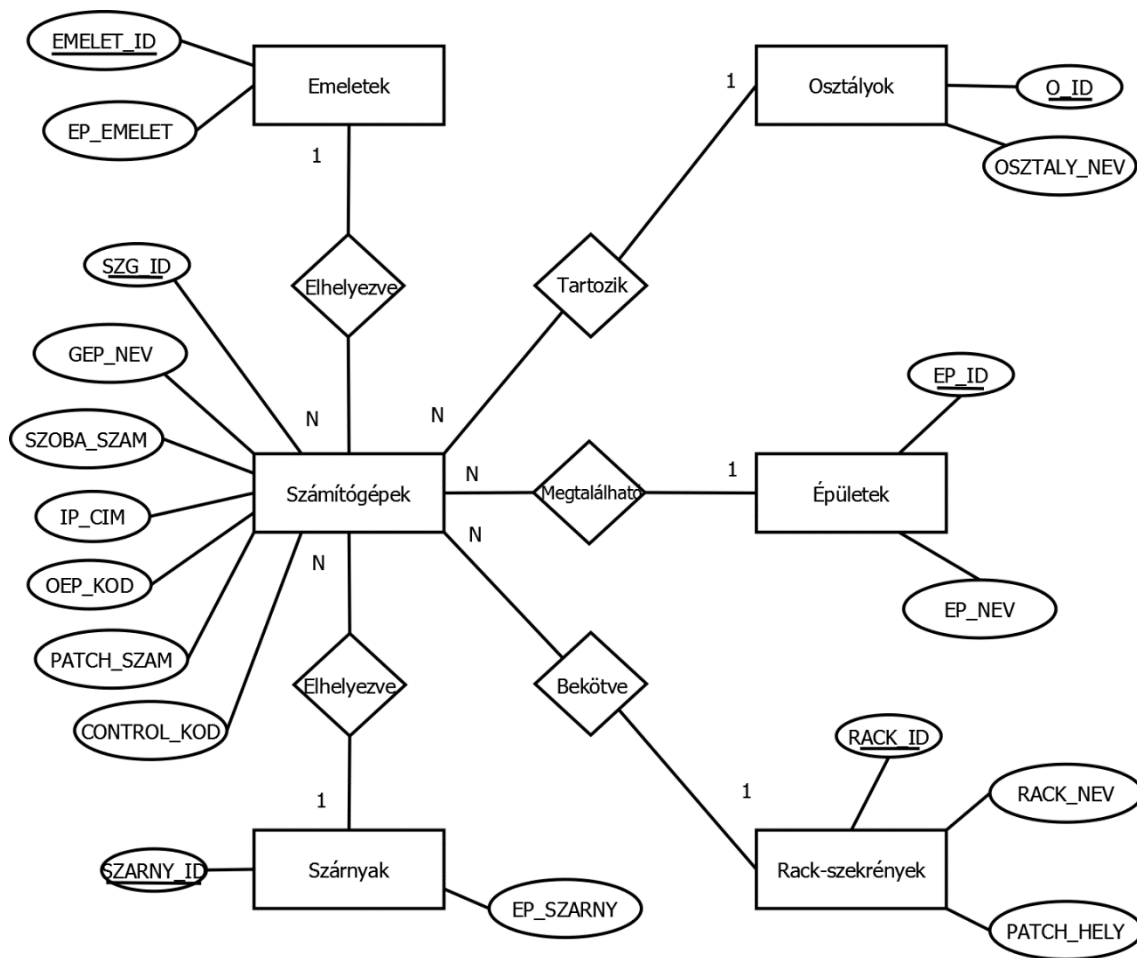
5. táblázat A Számítógép egyed attribútumai

EMELET_ID	Az Emeletek egyed egyértelmű azonosítója, kulcs
EP_EMELET	Az épület emelete

6. táblázat A Számítógép egyed attribútumai

SZARNY_ID	A Szárnyak egyed egyértelmű azonosítója, kulcs
EP_SZARNY	Az épület szárnya

Definiáltuk ezzel az egyedek attribútumait, és kijelöltük az elsődleges kulcsokat. Nincs más hátra, mint lefektetni az elkészült egyed/kapcsolat diagramot, melynek végleges formája a következőképpen néz ki:

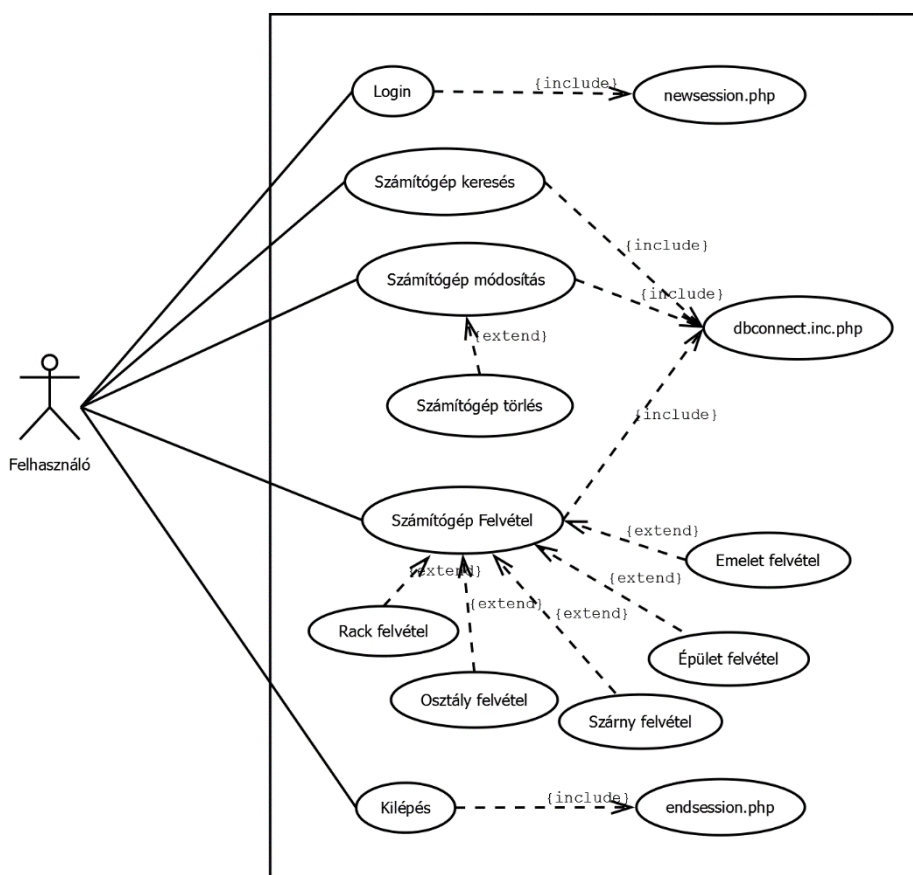


3. ábra: Az adatbázis egyedhalmazai

### 3.2. Nyilvántartó felépítésének tervezete

A következő Használati eset ábra (USE CASE diagram) segítségével mutatom be a nyilvántartó szoftverem szerkezeti felépítését:





4. ábra: Use Case diagram a nyilvántartóról

#### 4. Az alkalmazás implementálása és integrálása

Az eddig bemutatott adatbázis-tervnek megfelelően már létre tudjuk hozni nyilvántartó rendszerünk SQL kódját. Ez a kód rögzül a MySQL szerverbe, amihez majd a nyilvántartó programunkból tudunk hozzáférni, lekérdezni és módosítani az adatokat.

Elérkezett az idő, hogy elkészítsük a programot, ami a korábban létrejött adatbázisban fogja az adatokat felvenni, keresni és módosítani. Ez a felület fogja biztosítani majd a könnyebb használhatóságot, mert bonyolult SQL lekérdezések helyett, egy tudatosan megszerkesztett „kezelőfelületet” kell használnunk. Így az is könnyedén tudja majd kezelni az adatbázist, aki nem ismer semmiféle programnyelvet.

Egy nyilvántartó program elkészítésére sok fajta programnyelv áll rendelkezésre, mondhatni szinte mindegyik korszerű nyelv képes erre, de a kórháznak olyan program kellett, ami könnyedén elérhető a belső hálózat minden pontjáról. Egy hagyományos fejlesztés esetén az elkészült nyilvántartót minden munkahelyre fel kellene telepíteni, ez viszont időigényes tekintve, hogy több száz számítógépről van szó.

A megoldást az úgynevezett szerver-oldali program kifejlesztése jelentette, ami csak egy példányban, magán a szervergépen fut, és a hálózatról bárhonnét meghívható egy Web-böngésző segítségével. Erre az egyik legalkalmasabb nyelvet, a PHP-t választottam, mely teljes egészében támogatja az adatbázis rendszerekkel való kapcsolatot, köztük a MySQL-t is.

#### 4.1. Paraméterátadás Form-ok segítségével

Egy általános, egyszerű weboldal statikus parancsok segítségével nyeri el végső külalakját. Ezek a parancsok nem képesek kommunikálni más programokkal, illetve oldalakkal, csak egyszerűbb feladatokra alkalmasak (mint például a szöveg és kép elhelyezése). Ahhoz, hogy az oldalakat össze tudjuk hangolni, kétirányú kommunikációra képes dinamikus weboldalakat kell létrehozunk. Ez úgy jöhet létre, hogy az oldalak képesek felhasználói inputok fogadására és továbbítására. Erre biztosítanak nekünk megoldást az űrlapok, vagy eredeti nevén a Form-ok.

#### 4.2. A weblap szerkezeti felépítése

A weboldalak elkészítése során az egyszerű felépítésre törekedtem, a lapokon képeket, illetve egyéb design elemeket mellőztem, egyrészt a kórház teljesen megelégszik a „fa-pados” kinézettel, másrészt kevesebb helyet foglal így a nyilvántartó, az oldalak közötti betöltési idő is gyorsabb. Ezen irányelvek betartásával egy egyszerű kinézetű, könnyen kezelhető, és célnak megfelelő program megtervezését tűztem ki célul.

A weboldalak .php kiterjesztéssel ellátott állományok, melynek nagy része HTML kód-ból, és közé ágyazott PHP kódból áll. Szerkezetileg hat fő fájlra osztottam a nyilvántartót, mindegyik egyedi célt képvisel.

#### 4.3. A Főmenü

Baloldalon helyezkedik el a nyilvántartó főmenüje, mely mindig látható marad, az egyszerű kinézet jegyében felirattal ellátott gombok formájában. A négy darab közül, az első három különböző űrlapokra (keresés, adatfelvétel, módosítás) irányít minket, míg az utolsó kiléptet a nyilvántartóból, és megszakítja a kapcsolatot az adatbázis szerverrel. Mindegyik gomb saját Form-ban található meg, egy egysoros táblázat részeként.

Keresés – továbbítja a felhasználót a *keres.php* oldalra

Felvétel – átirányítja a felhasználót az *uj.php* oldalra

Módosítás/Törlés – a *modosit.php* oldalra linkel

Kilépés – az *endsession.php*-t hajtja végre



5. ábra: A nyilvántartó főmenüje.

#### 4.4. MySQL műveletek PHP-ben

Egy mai modern programnyelv egyik legfontosabb tulajdonsága, hogy lehetőséget biztosítson adatbázisokhoz való hozzáféréséhez. A PHP rengeteg adatbázis-kezelőt támogat, a MySQL eléréséhez is több beépített parancsot tartalmaz.

#### 4.5. Az űrlapok formai, belső felépítése

A továbbiakban végigveszem, és részletesen bemutatom a nyilvántartó program által végzett műveleteket. Először az űrlap oldali részt tárgyalom ki, majd rátérek a keresésre, illetve az egyéb SQL műveletekre, melyet a program hajt végre.

A nyilvántartó program három különböző bekérő oldalt tartalmaz. Ezek az űrlapok az adatok bevételére alkalmasak, melyet egy másik oldal ki fog értékelni (vagy ugyanaz az oldal,

amennyiben a paramétert közvetlenül önmagának küldte el). Korábbiakban már említettem a form-okat, segítségükkel jönnek létre a kereső-, adatfelvevő- és adatmódosító űrlapok.

#### *4.6. Keresés a program segítségével*

A Keresés űrlapon kitöltött keresési feltételeket megadva, és a „submit” típusú gombra kattintva, a program elküldi önmagának a megadott keresési feltételeket, ami alapján táblázatba soronként kiírja az eredményt. Ez a fájl csak magával a kereséssel és kiíratással foglalkozik, a többi művelet elvégzését a `query.php` végzi.

#### *4.7. Adatmanipuláció a program segítségével*

A `query.php` elnevezésű állományról eddig még csak néha tettem említést, pedig nagy méretet képvisel, annak ellenére, hogy kevés felületi elemet tartalmaz. Ennek az oka, hogy ez az oldal végzi el az SQL műveletek jelentős részét (kivéve a `select-et`), tehát a programozás jelen része volt a legösszetettebb.

A `query.php` fájlra az `uj.php` és a `modosit.php` oldalakról lehet eljutni, adatmanipuláció céljával. Ez az állomány végzi el az adatok felvételét, módosítását és törlését, valamint hibát jelez, ha üresen hagyunk mezőket. Erről az oldalról nem tartottam fontosnak képet mutatni, mivel csak a művelet sikeréről, illetve a hiba okáról tájékoztat minket.

Két weboldaltól is érkehetnek feldolgozandó adatok, illetve azokon belül is különbözőek, így el kell különíteni a műveleteket az állományon belül, hogy a megfelelő feladat legyen teljesítve. Emiatt lett bevezetve az űrlapoknál tárgyalt „címkék”, mely egy rejtett input mező segítségével azonosítja a kívánt művelet végrehajtását.

## **5. Verifikáció és validáció**

Több táblázaton keresztül mutatok be néhány tesztelési eredményt, melyből kiderül, hogy sikerült-e teljesíteni a megfogalmazott követelményeket.

1. táblázat Keresés oldal tesztje

Végrehajtott művelet	Várható eredmény	Végleges eredmény	Megfelelt?
„Keres” gomb megnyomása	Teljes nyilvántartás listázása	Teljes nyilvántartás listázódik	Igen
Keresés egy szűréssel	Szűrt lista megjelenítése	Szűrt lista helyesen megjelenik	Igen
Keresés két szűréssel	Szűrt lista megjelenítése	Szűrt lista helyesen megjelenik	Igen
Keresés három szűréssel	Szűrt lista megjelenítése	Szűrt lista helyesen megjelenik	Igen
Combobox-ok ellenőrzése	Mind az öt combobox listája helyes	Mind az öt combobox lista helyesen feltöltve	Igen
„Módosít/Töröl” gombra kattintás	modosit.php-ra ugrás helyes adatokkal	modosit.php-ra ugrás a helyes adatokkal	Igen

2. táblázat Felvétel oldal tesztje

Végrehajtott művelet	Várható eredmény	Végleges eredmény	Megfelelt?
„Felvesz” gombra kattintás adatbevitel nélkül	Hibaüzenet	Hibaüzenetet dob az első hiányos mezőre	Igen
„Felvesz” gomb, helyes űrlap kitöltéssel	Sikeres felvétel	Üzenet: Adatfelvétel sikerült	Igen
Új „Osztálynév” felvétele	Sikeres felvétel	Üzenet: Adatfelvétel sikerült	Igen
Új „Szárny” felvétele	Sikeres felvétel	Üzenet: Adatfelvétel sikerült	Igen
Combobox-ok ellenőrzése	Mind az öt combobox listája helyes	Mind az öt combobox lista helyesen feltöltve	Igen

3. táblázat Módosítás/Törlés oldal tesztje

Végrehajtott művelet	Várható eredmény	Végleges eredmény	Megfelelt?
„Módosít” gombra kattintás adatbevitel nélkül	Hibaüzenet	Hibaüzenetet dob az első hiányos mezőre	Igen
„Módosít” gomb, helyes űrlap kitöltéssel	Sikeres felvétel	Üzenet: Adatmódosítás sikerült	Igen
„Számítógép törlése” gombra kattintás adatbevitel nélkül	Hibaüzenet	Hibaüzenetet dob: hiányolja az IP címet	Igen
„Módosít” gomb, csak 'IP cím' módosítás	Sikeres módosítás	Üzenet: Adatmódosítás sikerült	Igen

„Számítógép törlése” gomb, csak 'IP cím' segítségével	Sikeres törlés	Üzenet: Adattörlés sikerült	Igen
Combobox-ok ellenőrzése	Mind az öt combobox listája helyes	Mind az öt combobox lista helyesen feltöltve	Igen

A fenti tesztnapló eredményei szerint sikerült a nyilvántartó szoftvert hibamentessé tenni, a vizsgált nyilvántartó program teljesíti azokat a követelményeket, melyeket a második fejezetben specifikáltunk.

## 6. Az alkalmazás bevezetése és nyomon követése

A program jelenlegi állapota megfelel az informatikai osztályvezető által támasztott követelményeknek, de még rengeteg elemmel lehetne bővíteni. A szoftvert betanított adminisztrátorok fogják kezelni, így nem szükséges a jelenleginél bonyolultabb felépítés, de ez nem kizáró ok, hogy a jövőben bővülő modulok és egyéb újítások belekerüljenek.

A nyilvántartó nyílt forráskód révén bárki által - aki a kórházban dolgozik és van hozzá jogosultsága - hozzáférhető és továbbfejleszhető, nem pedig egy lezárt, kódolt program. Számomra ez volt az első komolyabb PHP-ben írt program megtervezése és kivitelezése, így a fejlesztés közben hibák és új dolgok megismerése lassította a munkám.

A nyilvántartó továbbfejlesztésénél az első, amit meg lehetne tenni, a különböző jogok bevezetése, tehát csak bizonyos felhasználók tudjanak felvenni, módosítani és törölni adatokat (phpMyAdmin segítségével nélkül). Emellett létre kellene hozni egy új oldalt, mely Karbantartás címszó alatt, a jogok megvizsgálása után lehetne az egyéb táblákat módosítani vagy törölni.

Nem tartottam fontosnak sügöt építeni a programba, mert véleményem szerint egyszerű és egyértelmű a felépítése, így egy gyors tanulmányozás elég a kiismeréséhez.

A program anyagi értékét nehéz meghatározni, mivel nem általános célra, hanem kimondottan a kórház számára lett elkészítve, így pár tízezer forint környékére lehetne kalkulálni az árát.

A cikkem célja, miszerint egy működő nyilvántartó-rendszert kell megtervezni és elkészíteni sikeresnek bizonyult, mivel a program által támasztott követelményeknek eleget tettem, de a felhasználók igényei mindig újabb és újabb fejlesztésre adnak okot, így egy ilyen projektet sosem tekinthetünk tökéletesen befejezettnek.

## 7. Konklúzió, avagy a projekt értékelése

Konklúzió. Következtetés levonás. Ez olyan dolog, melyet az ember önmaga nem feltétlen tud helyesen megítélni, mondom ezt azért, mert a saját munkája iránt az ember mindig is kicsit elfogult lesz. Megpróbálok ennek ellenére reálisan értékelni a munkámat.

Elérkeztem egy nagyon hosszú út végéhez. Az út során sok fejtörések, megoldandó problémák közepette megszületett egy nagyszerű alkalmazás, mely azért közel sem tökéletes jelenleg, de a jövőben még minden adott ahhoz, hogy sikeres legyen. A tesztelők közül sokan azt mondták, már az első pillanat után, hogy mennyire letisztult, mégis átlátható és informatív a felülete. Ezzel úgy gondolom elértem a célokat, és megvalósítottam az elvárásokat, melyeket kitűztem magamnak az alkalmazással szemben.

A másik nehéz feladat ezen dolgozat megírása volt. Az alkalmazás komplexitásának, és sokrétűségének köszönhetően bőven akadt írni való, ám még ezek után is úgy érzem tudtam volna miről írni. A lehető legfontosabb elemeket megpróbáltam érthetően, és minél részletesebben bemutatni, annak érdekében, hogy egy kerek képet alkothassak arról, hogy miképpen áll össze, épül fel egy Android-os alkalmazás.

Megmutattam mi szükséges ahhoz, hogy valaki elkezdjen fejleszteni erre a remek platformra, készítettem egy tervet, mely alapján elkészítettem egy alkalmazást, aminek a legfontosabb elemeit ismertettem. Ezek után teszteltem is azt, és végül a Google Play Áruház segítségével elérhetővé tettem a nagyközönség számára is.

### Irodalomjegyzék

- [1] Ullman, J. D., & Widom, J. (2009) *Adatbázisrendszerek–Alapvetés*, Második, átdolgozott kiadás. Panem kiadó, Budapest, ISBN 963-5454-81-5.
- [2] Gábor, S. (2005). *Webes adatbáziskezelés MySQL és PHP használatával*. BBS INFO Kft.
- [3] Nixon, R. (2014). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. " O'Reilly Media, Inc."