

***Szakképesítés neve: Szoftverfejlesztő***

***OKJ száma: 54 213 05***

# **ZÁRÓDOLGOZAT**

## **Hibajegy kezelő rendszer**

Lipák Tibor  
Témavezető

Oszvald Gergő  
2/14. E

# Tartalom

Bevezetés .....	3
Felhasznált technológiák .....	3
Felhasználói dokumentáció .....	4
Általános specifikáció .....	4
Rendszerkövetelmények .....	4
Telepítési útmutató .....	5
A program használata .....	5
Bejelentkezés .....	5
Menüpontok .....	6
Fejlesztői dokumentáció .....	12
Az adatbázis adatmodellje .....	12
Táblák szerkezete .....	12
Fájlszerkezet .....	14
Konfigurálási fájlok .....	14
Oldalak .....	16
Tesztelési dokumentáció .....	19
Továbbfejlesztési lehetőségek .....	20
Zárszó .....	21
Hivatkozások .....	22

## Bevezetés

Azért választottam a hibajegykezelő rendszert a vizsgamunkám témájának, mert bele akartam mélyedni a webprogramozásba, és az ilyenféle rendszerek tökéletesen illeszkednek egy webes felületre. Illetve az adatbáziskezelés már régóta foglalkoztat, ezért mindenképpen egy olyan témát akartam választani, amihez elengedhetetlen egy jól megkonstruált adatbázis. Szerettem volna minél jobban belelátni az adatbázisfelépítés folyamatába, illetve egy meglévő adatbázis szerkesztési műveleteibe.

Egy ilyen rendszer megfelelően felépítve rendkívül hasznos lehet számos intézmény és vállalat számára, mivel célja, hogy felgyorsítsa az információáramlást és problémamegoldást, ezzel növelve az intézmény eredményességét. Továbbá, mivel webes felületen van felépítve, ezért nem kell a felhasználóknak további programokat letölteni és telepíteni az eszközeikre, elég, ha rendelkeznek egy böngészőt futtató eszközzel. Ezzel költséghatékony is lesz, mivel nem kell különböző platformokra lefejleszteni az alkalmazást.

Az alkalmazás célcsoportja olyan intézmények, amelyek számos munkaállománnyal és egyéb eszközzel rendelkeznek, és ezek a fokozott használat miatt gyakran meghibásodhatnak. Ilyenek például a tanintézmények, amelyeknek majdnem minden termében megtalálható legalább egy eszköz, vagy ilyenek még az infokommunikációs vállalatok irodái.

## Felhasznált technológiák

- Windows 10 operációs rendszer
- Microsoft Word 2016
- XAMPP
  - Apache webservert
  - MySQL adatbázis kezelő
- Notepad++
  - PHP
  - HTML
  - CSS
  - Javascript
- Google Chrome

# Felhasználói dokumentáció

## Általános specifikáció

Ez a program egy webes felületre szánt hibajegy kezelő szoftver. Elkészítése közben mindvégig az *egyszerűség* volt szem előtt tartva, így a leegyszerűsített felhasználói felület kezelése bárki számára könnyen megtanulható és használható, illetve a webes felületnek köszönhetően nem szükséges a felhasználónak további programokat telepíteni eszközeire. A rendszer felépítése szintén egyszerű: termekből, eszközökből és hibajegyekből áll. A termék létrehozhatóak név alapján, melyekhez hozzáadhatóak eszközök, és ezekre az eszközökre adhatóak fel a hibajegyek. Ezen felül még létrehozhatóak hírek a főoldalra, így egyszerűen adható át a felhasználóknak bármilyen közérdekű információ. További felhasználó létrehozásához pedig egyszerűen meg kell adnia pár adatot egy adminisztrátornak, és hozzá is van adva a rendszerhez. A felhasználók két csoportra bonthatók, melyek az **Adminisztrátor** és a **Dolgozó** elnevezést kapták.

A **dolgozó** csoport jogai közé tartozik a hírek olvasása, a termék böngészése, a termékben lévő eszközök böngészése, az eszközök eddigi hibajegyeinek megtekintése, és a rájuk való hibajegy feladása, illetve a saját hibajegyeik megtekintése.

Az **adminisztrátor** csoport továbbá képes a főoldalra új hírt írni, és a meglévőket módosítani, illetve törölni. Terméket tud létrehozni, módosítani és törölni. A terméken belül képes eszközöket hozzáadni, módosítani és törölni. Képes megnézni az összes feladott hibajegyet, illetve ezeknek tudja módosítani a státuszát, a prioritását, és törölheti őket. Végül képes felhasználókat létrehozni, módosítani és törölni.

## Rendszerkövetelmények

### Minimális rendszerkövetelmények

Intel Core 2 Duo 2.0 Ghz

Integrált videokártya

2 GB RAM

5Mb tárhely

Microsoft Windows 7 operációs rendszer

Google Chrome-on tesztelve, de kompatibilis a többi böngészővel is

## Ajánlott rendszerkövetelmények

Intel Core i5-7200 2.5Ghz

Integrált videokártya

8 GB RAM

10Mb tárhely

Microsoft Windows 10 operációs rendszer

Google Chrome-on tesztelve, de kompatibilis a többi böngészővel is

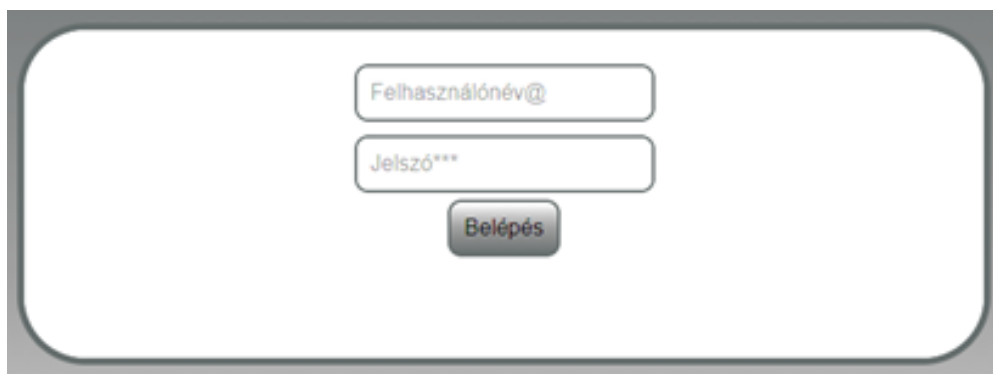
## Telepítési útmutató

Mivel a program webes alapú, így telepítés nem szükséges. Elég, ha a felhasználó egy böngészővel rendelkezik.

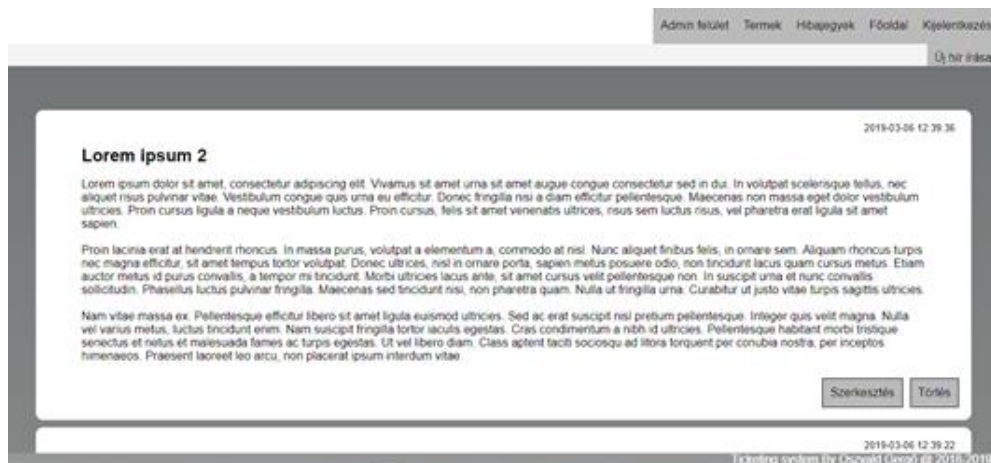
## A program használata

### Bejelentkezés

A program használatához elsősorban be kell jelentkezni a felhasználónak a bejelentkezőfelületen (1. ábra). Az első használatbavételkor egyetlen adminisztrátor felhasználó létezik, melynek a felhasználóneve “admin”, illetve a jelszava szintén “admin”. A bejelentkezést követően ez a felhasználó módosítható, vagy törölhető. A bejelentkezéshez be kell írni a felhasználónevet a “Felhasználónév@” mezőbe, majd a jelszavat a “Jelszó\*\*\*\*” mezőbe. Ha létezik a felhasználónév-jelszó páros, akkor a Belépés gombot megnyomva sikeresen bejelentkezett, és megjelenik a Főoldal (2. ábra). Ha a bejelentkezés sikertelen volt, akkor tájékoztat, hogy rossz a felhasználónév, vagy a jelszó.



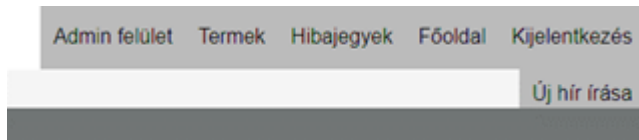
1. ábra, Bejelentkező felület



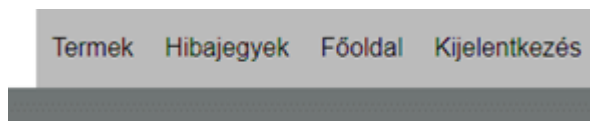
2. ábra, Főoldal

## Menüpontok

A bejelentkezést követően, az adminisztrátor felhasználói csoport tagjai (innenről: Admin) a 3. ábrán látható menüpontokat láthatják, míg a dolgozó felhasználói csoport tagjai (innenről: Dolgozó) a 4. ábrán lévő menüpontokat.



3. ábra, menüpontok az Adminnak



4. ábra, menüpontok a Dolgozónak

Egyes menüpontokhoz tartoznak almenüpontok is, mint ahogy a 3. ábrán is látható, a Főoldal menüpontnak az Admin számára létezik még egy „Új hír írása” almenüpont.

A menüpontok felépítése az Admin számára a következő:

- Főoldal
  - Új hír írása
- Hibajegyek
  - Kezeletlen hibajegyek
  - Összes hibajegy
- Termek
  - Új terem hozzáadása
- Admin felület
  - Felhasználók

## Új hír írása

Az Admin új hírt adhat hozzá a főoldalhoz, melyet minden felhasználó láthat és olvashat. Ehhez rá kell kattintani az „Új hír írása” almenüpontra, majd megjelenik két szövegdoboz. A „Hír címe...” szövegdobozba a hír címét kell írni, majd a „Hír szövege...” átméretezhető szövegdobozba lehet írni a hír szövegét. Végül a Hír kiírása gombra kattintva megjelenik a hír a Főoldalon. Mindkét szövegdobozt ki kell tölteni, különben nem jelenik meg a hír.

A meglévő hírek szerkesztése, illetve törlése azoknak bal alsó sarkában található Szerkesztés és Törlés gombbal történhet. Törlés előtt a program rákérdez, hogy valóban törölni akarja-e a hírt.

## Hibajegyek

Ezen a menüponton tekintheti meg a felhasználó a feladott hibajegyeit. Ha a felhasználó még nem adott fel hibajegyet, akkor a lista egy üzenettel jelzi, hogy nincsenek még feladott hibajegyei. A lista egyszerre 5 hibajegyet jelenít meg egy oldalon. Utána elérhetővé válik a lapozás az oldal alján a Következő oldal, illetve az Előző oldal gombokra kattintva.



ticketID	Feladó	Eszköz neve	Hiba leírása	Terem neve	Dátum	Státusz	Prioritás
WQM866 teszt (Kiss Géza)		Router	Hiába van bedugva a LAN1 portba a kábel, nem működik, és nem működik a reset gomb	Szerverterem	2019-03-06 13:59:33	Feldolgozás alatt...	Általános
KDA791 teszt (Kiss Géza)		Vetítő	Hibás színeket ad	Terem 005	2019-03-06 13:55:46	Kész	Általános
WN992 teszt (Kiss Géza)		Számítógép 004	Nem tölt be az operációs rendszer	Terem 204	2019-03-06 13:54:59	Ellenőrzés alatt...	Általános
CSE670 teszt (Kiss Géza)		Számítógép 012	Hibás a monitor	Terem 005	2019-03-06 13:54:32	Új hibajegy	Általános

Oldal: 1/1

5. ábra, Hibajegyek a Dolgozónak

A lista elemeihez tartozik a ticketID, egy egyedi azonosító, mely a hibajegy feladásakor jön létre. A Feladó, ami tartalmazza a hibajegyet feladó felhasználó felhasználónevét, illetve teljes nevét. Az eszköz neve, amihez fel lett adva a hibajegy. A Hiba leírása, egy rövid leírás a hibáról, melyet a feladó írt le. A Terem neve, melyben az eszköz található. A Dátum, a pontos idő, amikor a hibajegy feladásra került, illetve a Státusz és a Prioritás.

Ha a feladó felhasználója időközben törlésre került, akkor „Törölt felhasználó” jelenik meg Feladónak.

Egy hibajegynek négy különböző státusza van. Az „Új hibajegy”, amit akkor kap, amikor feladják. „Feldolgozás alatt...” státuszt kaphat, ha az Admin már tud a problémáról, és hozzájárul az elhárításához. „Ellenőrzés alatt...” státusza lehet, ha az eszköz elméletileg javítva van,

gyakorlatban viszont még tesztelésre vár. „Kész” státusz pedig akkor lehet, amikor a hiba elhárítódott. A státuszok visszafele is módosíthatóak, olyan esetekben, mint például ha az ellenőrzés nem volt sikeres, ezért tovább kell dolgozni a probléma javításán.

Az Admin változtathatja még a hibajegy prioritását *Általános* és *Fontos* között. A Fontos prioritású hibajegyek kerülnek előre a kezeletlen hibajegyek listájában, jelezve, hogy ezeket a hibákat minél hamarabb el kell hárítani.

A lista végén található Szerkesztés gombra kattintva változtatható a hibajegy státusza, illetve prioritása. Változtatás előtt az Admin a hibajegy jelenlegi állapotát láthatja. A Törlés gombra kattintva pedig törölheti az aktuális hibajegyet. Törlés előtt a program rákérdez, hogy valóban törölni akarja-e.

A lista rendezhető dátum, státusz és prioritás szerint növekvő és csökkenő sorrendbe. Alapjáraton dátum szerint rendez csökkenő sorrendbe, vagyis a legújabb kerül legelőre. Mikor a felhasználó beállít egy alaptól eltérő rendezést, az addig úgy marad, míg el nem navigál a Hibajegyek menüpontról, így az almenüpontok közti navigálásnál is megmarad.

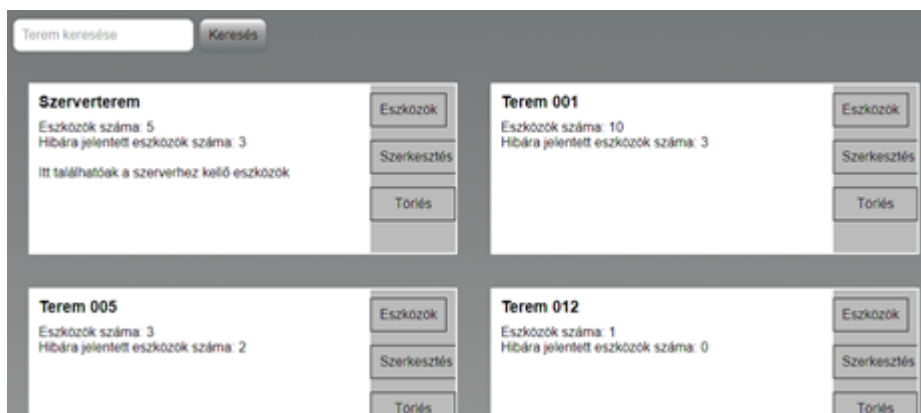
Az Admin számára további két almenüpont létezik. Az egyik a „Kezeletlen hibajegyek” almenüpont, melyen megtekintheti a még nem Kész státuszban lévő hibajegyeket. Ez a lista minden esetben előre teszi a Fontos prioritású hibajegyeket. A másik almenüpont az „Összes hibajegy”, melyen megtekintheti az összes eddigi feladott hibajegyet, státusztól és prioritástól függetlenül.

A listában egy eszköz nevére kattintva láthatjuk annak az összes hibajegyét. A terem nevére kattintva pedig megnézhetjük a teremben lévő eszközöket.

## **Termek**

Ebben a menüpontban találhatóak a rendszerhez hozzáadott termék. Minden terem egy külön doboz, a dobozon belül a bal felső sarokban látható a terem neve, alatta a teremben lévő eszközök száma, a hibás eszközök száma, majd a teremről egy rövid leírás. A doboz jobb oldalán a teremhez tartozó gombok találhatóak. (6. ábra)





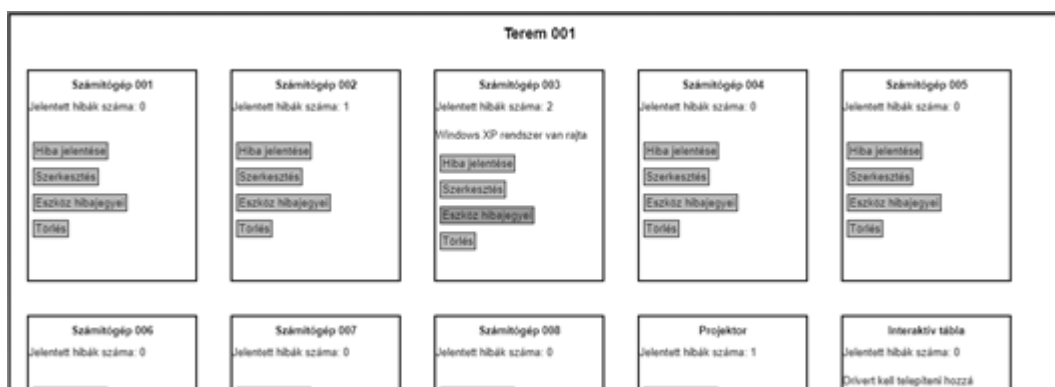
6. ábra, Termek

Új terem hozzáadni az Admin az „Új terem hozzáadása” almenüpontból képes. Létrehozás-kor meg kell adni a terem nevét, illetve nem kötelezően megadható róla egy rövid leírás. Két terem neve megegyezhet. A terem létrehozása a Létrehoz gombra kattintva véglegesíthető.

Az Admin a Törés gombra kattintva képes a termet, és ezzel az azon belüli eszközöket és a hozzájuk tartozó hibajegyeket is törölni. Törés előtt rákérdez a program, hogy biztos törölni akarja-e a termet. A Szerkesztés gombra kattintva képes módosítani a terem címét és leírását, amit a Módosít gombra kattintva véglegesíthet.

Az Eszközök gombra kattintva nyitható meg a terem, és tekinthetőek meg a benne lévő eszközök.

Minden doboz a termen belül egy-egy eszközt jelöl. A doboz felső részében található az eszköz neve, alatta pedig az eszközre feladott aktív hibajegyek száma. (7. ábra)



7. ábra. Eszközök egy teremben

Az almenüpontban található „Új eszköz hozzáadása” gombbal új eszköz adható hozzá a kiválasztott teremhez. Meg kell adni az eszköz nevét, és opcionálisan adható hozzá egy rövid leírás is, majd a Létrehoz gombbal hozható létre az eszköz.

Az Admin a Törlés gombra kattintva törölheti a kiválasztott eszközt. Törlés előtt a program rákérdez, hogy biztos törölni akarja-e az eszközt.

A Szerkesztés gombra kattintva szerkeszthető a kiválasztott eszköz. A Módosítás gombra véglegesíthető az eszköz módosítása.

Az Eszköz hibajegyei gombra kattintva megtekinthető az összes hibajegy, melyet a kiválasztott eszközre adtak fel. Itt az almenüpontból létrehozható egy új hibajegy az eszközhöz az „Új hibajegy” gombra kattintva, illetve ugyanerre jó a Hiba jelentése gomb is az eszközön.

Egy hiba feladásához csupán meg kell adni egy leírást a problémáról, majd a Küldés gombra rákattintva fel is lett adva a hibajegy.

### **Admin felület**

Ebben a menüpontban az Admin egy statisztikát találhat a rendszerben lévő termekről, eszközökről és hibajegyekről. Az almenüpontban található „Felhasználók” menüben találhatóak a rendszerben létrehozott felhasználók.

### **Felhasználók**

Itt az Admin egy listában láthatja az összes létrehozott felhasználót. Kereshet benne Felhasználónév, Vezetéknév, illetve Keresztnév szerint, és rendezheti a listát Felhasználónév, Keresztnév és Vezetéknév szerint csökkenő, vagy növekvő sorrendbe. A lista egyszerre 10 felhasználót jelenít meg egy oldalon, utána elérhetővé válik a lapozás, melyet a Következő oldal, illetve az Előző oldal gomb tesz lehetővé.

Az almenüpontban található „Új felhasználó hozzáadása” gombbal az Admin új felhasználót képes hozzáadni a rendszerhez. Meg kell adni az új felhasználó felhasználónevét, jelszavát, vezetéknévét, illetve keresztnévét, és ki kell választani, hogy melyik felhasználói csoportba tartozzon. Ezután a Létrehoz gombbal létrehozható a felhasználó a megadott adatokkal. Minden adatot kötelezően meg kell adni, illetve a felhasználónév nem lehet megegyező egy már létező felhasználó névvel. Létrehozás után a program nem navigál vissza a Felhasználók menübe. A sikeres létrehozásról az tájékoztat, hogy üresek lettek a beviteli mezők, esetleges további felhasználók hozzáadásához. A Vissza gombra kattintva visszakerülhet a Felhasználók menübe.

A lista végén található Törlés gombra kattintva törölheti a kiválasztott felhasználót. Törlés előtt rákérdez a program, hogy valóban törölni akarja-e a felhasználót. A felhasználó törlése

után a feladott hibajegyei továbbra is megmaradnak. Ha a törlésre került felhasználó be volt jelentkezve, akkor a rendszer kidobja őt a bejelentkezési felületre.

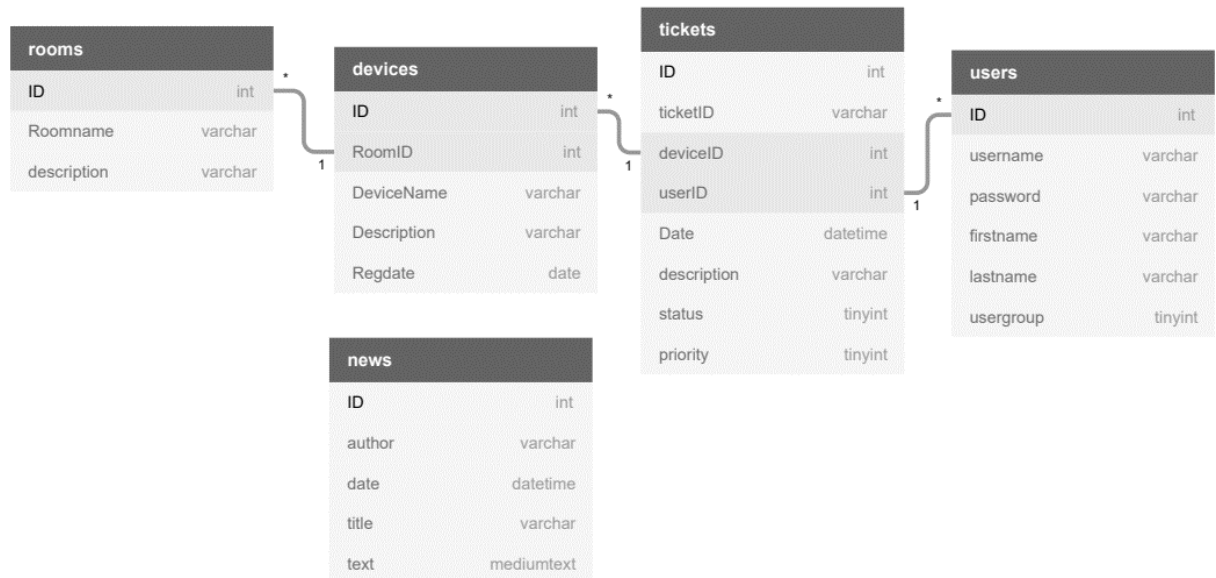
A Szerkesztés gombra kattintva szerkesztheti a felhasználó adatait. Szerkesztéskor a mezőkben a felhasználó aktuális adatai láthatóak. Módosításhoz a jelszó mezőn kívül minden mező ki kell legyen töltve, különben nem módosítható a felhasználó. A Mentés gombra kattintva véglegesíthetők a módosítások. Mentés után kap egy üzenetet a mentés sikerességéről, viszont manuálisan kell visszalépni a Felhasználók menübe a Vissza gomb segítségével.

### **Kijelentkezés**

A Kijelentkezés menüpontra kattintva kijelentkezhet a felhasználó az aktuálisan bejelentkezett fiókjából.

# Fejlesztői dokumentáció

## Az adatbázis adatmodellje



8. ábra, Adatbázis diagram (dbdiagram.io)

### Táblák szerkezete

Fontos, hogy a devices.RoomID – rooms.ID, tickets.deviceID – devices.ID mezők közötti megszorítások létrehozásánál a DELETE utasításnál CASCADE utasítás, a tickets.userID – users.ID között pedig SET NULL utasítás történjen.

### users (Felhasználók tábla)

ID	int, Auto Increment, elsődleges kulcs
username	varchar(30), utf8_hungarin_ci, egyedi kulcs, NULL
password	varchar(32), utf8_hungarin_ci
firstname	varchar(50), utf8_hungarian_ci
lastname	varchar(50), utf8_hungarian_ci
usergroup	tinyint

A password mező az MD5 kódolási algoritmus segítségével van titkosítva.

Az usergroup mezőben két érték tárolódhat, az 1-es érték a Dolgozó felhasználói csoport, a 2-es érték az Adminisztrátor felhasználói csoport.

**news (Hírek tábla)**

ID	int, Auto Increment, elsődleges kulcs
author	varchar(30), utf8_hungarian_ci
date	datetime
title	varchar(100)
text	mediumtext, utf8_hungarian_ci

**rooms (Termek tábla)**

ID	int, Auto Increment, elsődleges kulcs
Roomname	varchar(50), utf8_hungarian_ci
description	varchar(100), utf8_hungarian_ci

**devices (Eszközök tábla)**

ID	int, Auto Increment, elsődleges kulcs
RoomID	int, Index
DeviceName	varchar(50), utf8_hungarian_ci
Description	varchar(50), utf8_hungarian_ci
Regdate	date

**tickets (Hibajegyek tábla)**

ID	int, Auto Increment, elsődleges kulcs
ticketID	varchar(6), utf8_hungarian_ci, egyedi kulcs
deviceID	int, Index
userID	int, Index
Date	datetime
description	varchar(200), utf8_hungarian_ci
status	tinyint
priority	tinyint

A status mezőnek 4 értéke lehet, 1-4-ig terjedően. Az értékek elnevezését a config.php-ban a ticketstatus nevű változó tárolja.

A priority mezőnek 2 értéke lehet, 1 és 2. Az értékek elnevezését a config.php-ban a ticketprior nevű változó tárolja.

## Fájlstruktúra

- config.php
- staticelements.php
- login.php
- index.php
- tickets.php
- rooms.php
- admin.php
- users.php
  - css
    - mystyles.css

## Konfigurációs fájlok

Minden oldal első utasításai között kell szerepeltetni az alábbi fájlok betöltését, ebben a sorrendben:

- config.php
- staticelements.php

A login.php esetben csak a config.php-t kell betölteni.

## Lényeges változók:

- *“LoggedIn” session* – Ha létezik, akkor a felhasználó be van jelentkezve. Bejelentkezéskor az értéke true-ra állítódik.
- *“UserGroup” session* – A felhasználó felhasználói csoportját tárolja el, így megspórolva számos lekérdezést a felhasználói csoportokkal kapcsolatban.
- *“UserID” session* – A felhasználó azonosítóját tárolja az users tábla ID mezőjéből.
- *ROWPERPAGETICKET* – Ezzel a konstans változóval a hibajegyek listához a lapozásonként megjelenített sorok állíthatóak be.
- *ROWPERPAGEUSERS* – Ezzel a konstans változóval a felhasználók listához a lapozásonként megjelenített sorok állíthatóak be.
- *ticketstatus* – Egy egydimenziós tömb, mely eltárolja a hibajegyek státuszainak nevét és színeit. Fixen 4 méretű, mivel kizárólag 4 státusz beállítást kezel a rendszer.

- *ticketprior* – Egy egydimenziós tömb, mely eltárolja a prioritáshoz tartozó elnevezéseket és színeket. Mérete fixen 2, mivel kizárólag 2 prioritási beállítást kezel a rendszer.

### **config.php**

A `config.php` fájlban találhatóak azok az algoritmusok, melyeknek minden oldal megnyitásakor le kell futniuk, illetve itt vannak definiálva egyéb konfiguráláshoz létrehozott változók, és itt töltődik be a stíluslap is. Ezt a fájlt minden további oldalon be kell tölteni, a betöltésnek kell lennie az első lefutó feladatnak.

Ebben a fájlban fut le a `session_start()` metódus, így nem szükséges a további fájlokban lefuttatni.

Itt csatlakozik a weboldal a MySQL serverhez. A server adatait az `SQLHOST` (kiszolgáló címe), `SQLUSER` (felhasználónév), `SQLPASS` (jelszó), `SQLDB` (adatbázis neve) elnevezésű konstans változókban kell megadni.

Az `“alert($str)”` metódus php kódon belüli javascript alert metódus. Ez segíti, hogy php kódban is könnyen megjeleníthető legyen egy üzenetdoboz.

A `“DeleteWarning()”` nevű rövid javascript metódus arra jó, hogy bármelyik oldalon található Törlés gombra kattintva a felhasználónak megjelenítsen egy dialógus dobozt, így egy félrekattintás esetén nem törlődik azonnal egyetlen adat sem. Mindegyik HTML kódban található Törlés gomb formjánál az `“onsubmit”` tag ezzel a metódussal tér vissza.

### **staticelements.php**

Ez a fájl tartalmazza azokat a statikus elemeket, melyeknek – a `login.php`-t kivéve – minden oldal megnyitásakor meg kell jelenniük. Ezek az elemek ebben az esetben a menüpontok.

A fájl elején ellenőrzésre kerül, hogy a felhasználó be van-e jelentkezve, és ha igen, akkor az `“UserGroup”` session értékét állítsa be a megfelelő értékre. Viszont ha a felhasználó nem létezik, de mégis be volt jelentkezve, akkor az azt jelenti, hogy a felhasználó időközben törlésre került, így a `“LoggedIn”` session is törlésre kerül, ezzel kijelentkeztetve a felhasználót. Ha a változó nem létezik, akkor átirányítja a `login.php`-re.

A menüpontokat a header tag-en belül a `<nav></nav>` tag-ek közé kell tenni.

Itt az Admin felület nevű menüpont akkor jelenik meg, ha a felhasználói csoport értéke nagyobb, vagy egyenlő, mint 2.

## Oldalak

Az oldalak fő tartalma a `<main></main>` tag-ek között található meg.

A `<nav></nav>` tagen belül vannak megadva az almenüpontok. Az adminisztrátoroknak megjeleníthető almenüpontok előtt a felhasználói csoport értékét kell ellenőrizni, hogy nagyobb, vagy egyenlő legyen mint 2.

### Lényeges változók:

- *“webpage” session* – A változó értéke tárolja, hogy a felhasználó éppen melyik oldalon jár. Ha a felhasználó almenüpontokon belül navigál, az értéke nem változik, így a listáknak a rendezése addig marad a beállított értékeken, ameddig el nem kattint egy másik menüpontra.
- *GET tömb “show” indexe* – Ettől az értéktől függ, hogy az aktuális oldalon mi jelenik meg. Gyakoribb értékei: new, edit.

### login.php

Ezen az oldalon jelentkezhetnek be a felhasználók. Erre az oldalra navigál a Kijelentkezés menüpont is, így ez az oldal ellenőrzi, hogy a felhasználó rákattintott-e a Kijelentkezés gombra. Ha létezik a POST tömbben átadott kijelentkező gomb neve, akkor minden sessiont töröl.

Bejelentkezéskor egy sql utasítással ellenőrzi a program, hogy létezik-e a felhasználónév és – md5-tel titkosított - jelszó páros. Ha létezik, akkor a “LoggedIn” sessiont true értékre állítja, és a felhasználó ID-jét eltárolja az “UserID” session-ben, és átirányít az index.php-re. Ha nem található sor az adatbázisban a megadott értékekkel, akkor a message nevű változónak átad egy hibaüzenetet, amit kiír a felhasználónak.

### index.php

Ez a főoldal, itt találhatóak a hírek. Ez a fájl kezeli a hírek létrehozását, szerkesztését és törlését.

Amikor egy adminisztrátor hozzáad egy hírt, a program ellenőrzi, hogy léteznek-e a hír hozzáadásához szükséges változók. A “snewnewsbutton” nevű session létezése ahhoz szükséges, hogy a hír hozzáadása után a weboldal frissítésekor ne adja újra hozzá a hírt. A session bármilyen értékkel létrehozható, amikor az adminisztrátor megnyitja a hír hozzáadás oldalt, majd a hír hozzáadása után törölni kell.

A hír írójának a firstname és az username mező lesz megadva, majd egy sql utasítással új sor lesz beillesztve a news táblába a megfelelő változók értékeivel.



Egy hír szerkesztéséhez eltároljuk a Szerkesztés gomb értékeként a hír ID-jét, majd ezt tovább adjuk a Módosít gombnál a newseditid változónak, és a POST tömbből visszkapott értékekkel frissítjük a sort, ahol az ID mező értéke egyenlő a newseditid változó értékével.

Törlésnél, hogy ne azonnal törölje a hírt, a form onsubmit tag értékének “return DeleteWarning(this)”-nek kell lenni, így először egy dialógus dobozt dob fel a felhasználónak. A Törlés gomb értékének át kell adni a hír ID-jét, és azt a sort kell törölni, ahol az ID mező értéke egyenlő a newsdeletebtn változó értékével.

Ha a GET tömb “show” indexe nem létezik, akkor a főoldal a hírsávot jeleníti meg, a hírek lekérdezése dátum szerint van rendezve csökkenő sorrendben.

A Szerkesztés gomb létrehozásánál két hidden típusú input tag-et hozzá kell adni, melyek a hír címét, és a hír szövegét tárolják el, így a szerkesztés oldalon azokra az indexekre kell hivatkozni a POST tömbben, és az értéküket át kell adni a megfelelő szövegdoboznak.

Ha a GET tömb “show” indexe egyenlő “new” értékkel, akkor jelenik meg az új hír hozzáadása, ha az értéke “edit”, akkor pedig a hír szerkesztése. Mindkét esetben, ha a felhasználónak a felhasználói csoport értéke kisebb mint 2, akkor az oldal nem jelenik meg.

### **tickets.php**

Ebben a fájlban találhatóak a hibajegyek létrehozásával, szerkesztésével és törlésével kapcsolatos utasítások, illetve ez foglalkozik a hibajegyek listába rendezésével.

Új hibajegy hozzáadásánál létre kell hozni egy 6 karakterű hibajegy ID-t, ami még nem létezik az adatbázisban, utána egy sql utasítással beszúrni egy új sort a megfelelő adatokkal.

Hibajegy törlés és módosítás után a „lastpage” nevű session az előzőleg megtekintett almenüpontot tárolja, és arra irányít vissza.

Az almenüpontokhoz akkor kerül hozzá az „Új hibajegy” nevű almenüpont, ha létezik a GET tömb „deviceid” nevű index, mivel ennek létezésekor tudható, hogy a felhasználó éppen egy eszköz hibajegyeit nézi.

Amikor a GET tömb „show” index értéke „edit”, akkor az adminisztrátor egy hibajegyet szerkeszt. A státusz és a prioritást beállító legördülő menü értéke mindig a hibajegy aktuális értéke.

A „show” érték „own”, „all” és „half” értékeinél van felhasználva a ROWPERPAGE-TICKET nevű konstans változó. Egy sql lekérdezéssel összekötve a lista mindig a megfelelő oldalt jeleníti meg.

Ha az sql lekérdezés után az username mező üres értéket ad vissza, akkor a felhasználó nem létezik, ezért a változónak, vagyis a feladó nevének az értéke „Törölt felhasználó” lesz.

A listában az „Eszköz neve” és a „Terem neve” elnevezések átirányítanak a megfelelő linkre a megfelelő GET értékek megadásával.

A Szerkesztés és Törlés gombok csak akkor jelennek meg, ha az „UserGroup” session értéke nagyobb vagy egyenlő, mint 2.

A Következő oldalra lapozás csak akkor jelenik meg, ha a jelenlegi oldal értéke kisebb, mint a maxpage változó értéke, az Előző oldal pedig akkor, ha a jelenlegi oldal értéke nagyobb, mint 1. A maxpage értéke a tickets tábla sorainak számából, és a ROWPERPAGETICKETS változó értékéből van kiszámolva.

### **rooms.php**

Ez a fájl kezeli a termeket és az eszközöket egyszerre. Ezen belül történik a terem létrehozás, szerkesztés és törlés, illetve az eszköz létrehozás, szerkesztés és törlés. Ebben van kezelve a termék és eszközök megjelenítése is.

Terem törlésnél itt lényeges, hogy megfelelő megszorítások legyenek létrehozva a táblák között, így elég csak a terem törléséhez tartozó utasítást végrehajtani, és a hozzátartozó eszközök és hibajegyek automatikusan törlődni fognak.

Termék közötti keresésnél a keresett szó a terem nevében bárhol megjelenhet. a GET tömb „searchr” nevű index létezése jelzi, hogy a felhasználó keresett, az értéke pedig maga a keresett kifejezés.

A Szerkesztés és Törlés gombok csak akkor jelennek meg, ha az „UserGroup” session értéke nagyobb, vagy egyenlő, mint 2.

### **admin.php**

Ezt az oldalt csak adminisztrátorok érhetik el, így ha a felhasználó felhasználói csoport értéke kisebb, mint 2, akkor az oldal nem jelenik meg.

Ez a fájl statisztikákat közöl a rendszer adataiból kiszámolva, illetve innen léphet az adminisztrátor a Felhasználók almenüre.

Statisztikaszámításnál figyelembe van véve, hogy osztásnál a nevező értéke ne legyen 0.

### **users.php**

Ezt az oldalt csak adminisztrátorok érhetik el, így ha a felhasználó felhasználói csoport értéke kisebb, mint 2, akkor az oldal nem jelenik meg.

Ez a fájl kezeli a felhasználók listában való megjelenítését, felhasználó hozzáadását, szerkesztését és törlését.

Felhasználó hozzáadásánál ellenőrizni kell, hogy létezik-e már a megadott felhasználónév, mivel a username mező egyedi, így nem létezhet két sor ugyanazzal az értékkel. Sikeres hozzáadás után, mivel a program nem navigál el a hozzáadás oldalról, a beviteli mezők értékeit le kell nullázni. Az oldalon maradás egy kényelmi funkció, ha esetlegesen egyszerre több felhasználót hozna létre egymás után egy adminisztrátor.

Egy felhasználó szerkesztésénél, ha a jelszó beviteli mező értéke nem üres, akkor a jelszavat is frissíteni kell, ellenkező esetben csak a többi adat kerül frissítésre.

Felhasználó törlésénél, ha a törölni kívánt felhasználó ID-je megegyezik a felhasználó saját ID-jével, akkor nem engedi törölni.

## **Tesztelési dokumentáció**

### **Teszt eset: Hibajegy feladás tesztelése**

Elvárt eredmény: Új hibajegy jelenik meg a Hibajegyek menüpontban, és átirányít oda

Kapott eredmény: Megjelent az új hibajegy, és át is irányított

### **Teszt eset: Eszköz szerkesztés tesztelése**

Elvárt eredmény: Az eszköz neve és leírása átmódosul a megadottra, és az eszköz nevét kötelező megadni

Kapott eredmény: Eszköznév nélkül nem engedi módosítani, és módosításnál a leírás és a név is frissült

### **Teszt eset: Admin felületi statisztikák tesztelése**

Elvárt eredmény: Ha nincs hibajegy, akkor a hibajegyekkel számoló statisztikák „Nincs adat” értéket adnak. Ha nincs terem, a termekkel számoló statisztikák szintén nincs adat értéket adnak.

Kapott eredmény: Ha nincs hibajegy, „Nincs adat” értéket adnak. Ha nincs terem, „Nincs adat” értéket adnak.

### **Teszt eset: Saját felhasználó törlése**

Elvárt eredmény: Nem engedi törölni a felhasználót

Kapott eredmény: Frissült az oldal, egy hibaüzenet ablak jelent meg, és nem töröltődött a felhasználó

### **Teszt eset: Felhasználótörlés tesztelése**

Elvárt eredmény: A felhasználó hibajegyei megmaradnak, a Feladó helyén „Törölt felhasználó” jelenik meg. Ha a felhasználó be van jelentkezve, akkor kijelentkeződik.

Kapott eredmény: A hibajegyei megmaradtak „Törölt felhasználó ( )” feladóval. A felhasználót kidobta az oldal, ha át akart navigálni egy másik menüpontra.

### **Teszt eset: Felhasználói csoportok tesztelése**

Elvárt eredmény: Ha egy Dolgozó fh. csoportú egy csak adminisztrátor által elérhető oldalra próbál navigálni a böngésző url sávjában, akkor nem jelenik meg neki az oldal.

Kapott eredmény: Az oldal üresen jelent meg.

Teszt közben felmerült problémák: A felhasználó adminisztrátori jogok nélkül is hozzáférhetett a „tickets.php?show=half”, a „tickets.php?show=all”, és a „rooms.php?show=new” oldalhoz.

## **Továbbfejlesztési lehetőségek**

### **Értesítés új hibajegy feladásáról**

Ha egy felhasználó új hibajegyet adott fel, akkor az adminisztrátorok kapjanak emailt, hogy melyik eszközre, milyen hibajegyet adtak fel. Ehhez új felhasználó hozzáadásakor meg kell adni egy email címet is. Illetve ha egy adminisztrátor még nem tekintette meg az új hibajegyet, akkor az oldalon a Hibajegyek menüpont mellett jelenjen meg zárójelben az „új” szó.

### **Rendezések, megjelenítések**

Lehessen keresni a hibajegyek között TicketID, feladó, eszköznév, teremnév és dátum szerint. A termék is lapozhatóak legyenek, és lehessen rendezni őket.

A felhasználó be tudja állítani, hogy mennyi sor jelenjen meg oldalanként a listákban.

### **Oldal dinamikussá tétele**

Az AJAX és jQuery használatának bevezetése, hogy az oldal sokkal innovatívabb, és még felhasználóbarátabb legyen.

### **MVC szerkezeti minta szerinti felépítés**

Az MVC (Model View Controller), magyarul MNV (Model Nézet Vezérlő) egy szoftvertervezésben használatos szerkezeti minta. Lényege, hogy szétválasztja a felhasználói felületet az adatok elérésétől és az üzleti logikától egy közjük tett vezérlő segítségével. (1)

Az adatok és a felhasználói felület szétválasztása lehetővé tenné, hogy a rendszert egy már meglévő felületbe lehessen integrálni.

## **Zárszó**

Sok újat tanultam az eltervezett feladatom megoldása során. Elsősorban, hogy sokkal több időt kell befektetni a tervezésbe. A program megalkotásának kezdeténél nem volt egy kész vázlatom az egésztől, nem volt egy út, amin haladhattam volna, hogy mit mi után kéne megcsinálnom. Így egyes ötletek később jöttek, majd egy ponton realizáltam, hogy az egészen kicsit változtatni kell, mert nem jó irányba haladtam, ezzel több munkát adva saját magamnak. Így rájöttem, hogy egy nagyobb kaliberű program esetében a tervezésre is sok időt kell szánni, ellenkező esetben a megalkotása sokkal nagyobb kihívás lesz.

A tervezéssel ellentétben a webes programozás nem volt akkora kihívás. Ez talán annak tudható be, hogy a PHP nyelvet már máskor is használtam egy hasonló kaliberű oldal megalkotására, így majdnem minden funkciót a meglévő tudásommal oldottam meg. Ezért talán kicsit bánom, hogy nem használtam benne a számomra kevésbé ismert jQuery könyvtárat, vagy Ajax-ot.

A MySql adatbázishasználat során viszont sok újat tanultam. Megtanultam összetettebb sql lekérdezéseket írni, és hogy hogyan kell helyesen felépíteni egy adatbázist.

Összességében számomra egy izgalmas kihívás volt, mivel a program megalkotása alatt felmérhettem a jelenlegi tudásomat, és bővíthettem azt új ismeretekkel.

## Hivatkozások

1. *Model-Nézet-Vezérlő.* [Online] [Hivatkozva: 2019. Március 13.]  
<https://hu.wikipedia.org/wiki/Modell-n%C3%A9zet-vez%C3%A9rl%C5%91>.