Budapesti Műszaki Szakképzési Centrum

Neumann János Számítástechnikai Szakgimnáziuma

***Szakképesítés neve:*** Szoftverfejlesztő ***OKJ száma:*** 54 213 05

**ZÁRÓDOLGOZAT**

**Zálogházi nyilvántartó program**

|  |  |
| --- | --- |
| Várkonyi Tibor | Kövérné Gargya Viktória |
| konzulens | 14.rsze |

Budapest, 2020.

Tartalomjegyzék

[Bevezetés 5](#_Toc39426977)

[Témaválasztás 5](#_Toc39426978)

[A probléma rövid ismertetése 5](#_Toc39426979)

[Felhasználói dokumentáció 6](#_Toc39426980)

[Rendszerigény 6](#_Toc39426981)

[Beüzemelés 6](#_Toc39426982)

[Jogosultsági szabályok 6](#_Toc39426983)

[Felhasználói felület elemei, funkciói 7](#_Toc39426984)

[Zálogjegy felvétel 7](#_Toc39426985)

[Zálogjegy kezelés 8](#_Toc39426986)

[Új ügyfél felvétel 10](#_Toc39426987)

[Fejlesztői dokumentáció 11](#_Toc39426988)

[Rendszerterv 11](#_Toc39426989)

[Adatbázis terv 11](#_Toc39426990)

[Adattáblák 11](#_Toc39426991)

[Megvalósítás 11](#_Toc39426992)

[Biztonság 12](#_Toc39426993)

[Tesztelés 12](#_Toc39426994)

[Összegzés 13](#_Toc39426995)

[Fejlesztési lehetőségek 13](#_Toc39426996)

[Hivatkozások 15](#_Toc39426997)

# Bevezetés

## Témaválasztás

A záródolgozatom témájának azért választottam a zálogházi nyilvántartást, mert több éves tapasztalatom van zálogházi tevékenység működésében. Tisztában vagyok a zálogházi ügymenettel valamint működési nehézségeivel.

## A probléma rövid ismertetése

Egy zálogháznak készítek egy belső nyilvántartói rendszert, valamint ügyfelek részére egy mobil webalkalmazást zálogjegyük egyszerű nyomon követéséhez.

Sokan megfordulnak zálogházba – vállalkozók, magánszemélyek – akiknek azonnali gyors kölcsönre van szükségük.

Szeretnék egy olyan rendszert létrehozni, ami a dolgozók és az ügyfelek igényeihez rugalmasan igazodik és kezeli a zálogházba felvett tárgyak adatait. Elzálogosítás esetén új tárgy felvitele az adatbázisba, hosszabbítás esetén az adatok módosítása, illetve kiváltás esetén az adatok törlése.

Az adatbázist egy központi szerveren szeretném tárolni, mivel különböző helyeken szeretném azokat elérni. Például, azért mert több ügyfél felvetette, hogy a hitelek hosszabbítása bármelyik zálogfiókba egyszerűen lebonyolítható lehessen, ne csak ahol elzálogosította a tárgyát.

Az adatbázis különböző hozzáférési jogokkal lehessen használni. A zálogigazgató hozzáférhet az egész adatbázishoz, le tud kérdezni belőle. A zálogfiók – ahol az elzálogosítás történik – tud felvinni új adatot az adatbázisba, azt módosítani (kiváltás, hosszabbítás), lekérdezéseket végezni. A zálogfiók csak a hosszabbítást tudja elvégezni a más fiókban elzálogosított tárgynál. Az ügyfél csak a saját elzálogosított tárgyait látja a mobil webalkalmazáson keresztül – mikor, melyik zálogfiókba, milyen kölcsönösszeggel lett elzálogosítva, valamint mennyi kamat van rajta az adott napon.

# Felhasználói dokumentáció

## Rendszerigény

* Internet kapcsolat
* Webböngésző (Mozilla Firefox, Google Chrome, Opera, Microsoft Edge)

## Beüzemelés

## Jogosultsági szabályok

Ügyfél jogosultságai:

* mobil alkalmazáson való bejelentkezés
* olvasási jog a zálogjegy táblán lévő ügyfél azonosítójával rendelkező rekordokra
* saját jelszó változtatása a mobil alkalmazásra való bejelentkezéshez

Dolgozói jogosultságok munkakör alapján:

Zálogfiókban dolgozók jogosultságai:

* ügyfél tábla adatainak módosítása (az összes zálogfiók egyhez fér hozzá), új ügyfél hozzáadása
* zálogjegy tábla olvasási jog (csak az adott zálogfióké)
* megjegyzés hozzáadása a zálogjegyhez

*Pénztáros jogosultságai:*

* zálogjegy tábla olvasási és módosítási jog – hosszabbítása, kiváltása (helyi zálogfiók)
* zálogjegy tábla olvasási és módosítási jog – hosszabbítás (más zálogfiók)

*Becsüs jogosultságai:*

* zálogjegy táblába új zálogtárgy felvitele
* pénztáros jogosultságai

Ellenőrző becsüs jogosultságai:

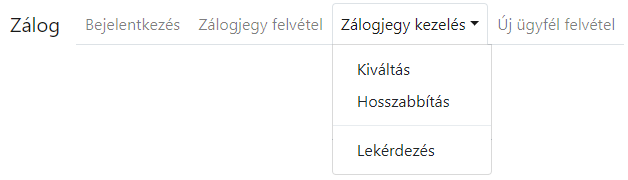
* olvasási jog a zálogjegy táblán
* megjegyzés hozzáadása a zálogjegyhez

Zálogigazgató jogosultságai:

* ügyfél- és zálogjegy tábla olvasási jog (összes fiók)
* zálogfiók- és dolgozó tábla olvasási és írási jog
* zálogjegy bevonása kényszerértékesítésre, ha a lejárati időt túllépte

## Felhasználói felület elemei, funkciói

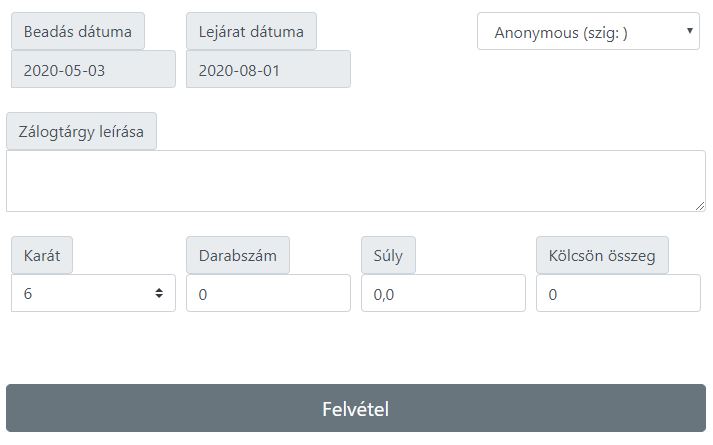
Ahogy az 1. ábra mutatja a felhasználói felületet több részre bontottam szét, azért hogy elkülönüljenek egymástól a különböző funkciók.



1. ábra: Menüsor

### Zálogjegy felvétel

A 2. ábra mutatja be az új zálogtárgy elzálogosítás folyamatát.



2. ábra: Zálogjegy felvétel

A beadás dátuma az aznapi dátum, amikor az elzálogosítás történik. A lejárati dátum a beadás dátumától számított 90. nap. Ezt a két értéket a program maga generálja le.

A dátumok melletti lenyíló mező az ügyfelek listáját tartalmazza személyigazolvány számmal együtt. Ha az elzálogosító nem adja meg az adatait, akkor az Anonymous felhasználóval lesz rögzítve. Ha megadja az ügyfél az adatait, és már regisztrálva van, akkor megjelenik a listába a neve, ellenkező esetben az „Új ügyfél felvétel” menüpont alatt kell regisztrálni. Ha nevesítve van a zálogjegy, akkor csak az válthatja ki a zálogjegyet, aki ott szerepel.

Zálogtárgy leírása egy kötelezően kitöltendő mező, aminek minimum 5 karakter hosszúnak kell lennie.

A karátnak a számát egy lenyíló listából lehet kiválasztani – 6, 8, 10, 14, 18.

A darabszámnak minimum 1-nek kell lennie, 1-esével lehet növelni vagy csökkenteni az értéket.

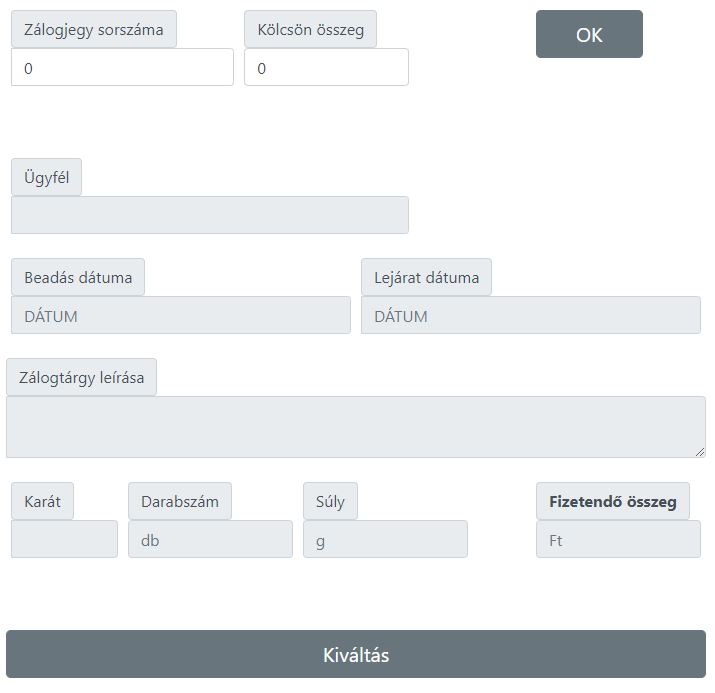
A súly század pontosságúra adható meg, minimum 0,5-nek kell lennie, 0,01-el lehet növelni vagy csökkenteni az értékét.

A kölcsön összeg minimum 2000 Ft-nak kell lennie, 500-zal lehet növelni vagy csökkenteni az értékét.

A felvétel gombra való kattintáskor megtörténik az elzálogosítás, az adatok belekerülnek az adatbázisba.

### Zálogjegy kezelés

A 3. ábra mutatja be a „Hosszabbítás” és a „Kiváltás” felületét.



3. ábra: Kezelés

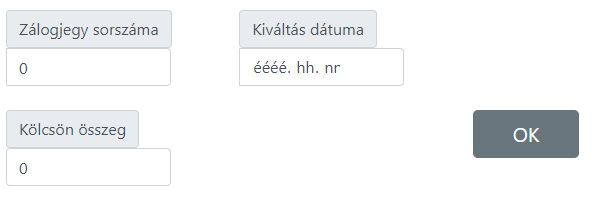
A zálogjegy sorszámát és a kölcsönösszeget kötelező megadni a zálogjegy beazonosítása érdekében. Az ügyfél, a beadás dátuma, a lejárat dátuma, a zálogtárgy leírása, a karát, darabszám és a súly mezőket az „OK” gomb lenyomásakor kitölti.

A fizetendő összeg a kiváltás esetén a kölcsön összeg, az eltelt napokra a kamat és a kezelési költség összege. Kiváltás esetén az eltelt napokra a kamat és a kezelési költség összege.

Kiváltás gombra való kattintáskor törlődik az adatbázisból az adat.

Hosszabbítás gombra való kattintáskor a beadás dátuma megváltozik az aznapi dátumra.

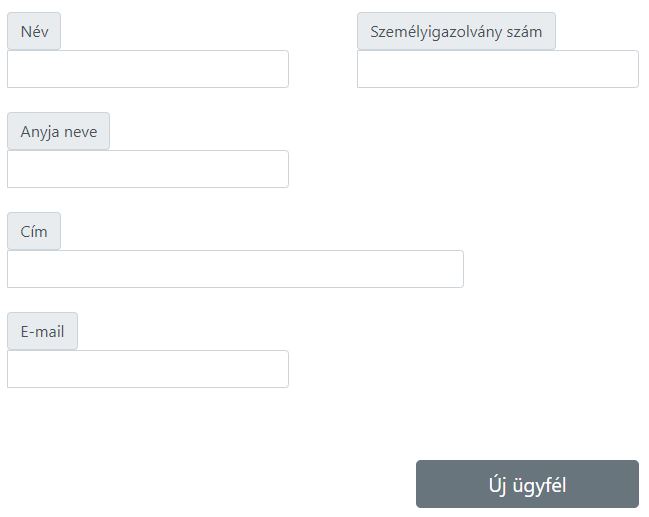
Ahogy a 4. ábra mutatja, a lekérdezésnél van egy kiváltás dátuma mező, ahol tervezett kiváltási dátumot lehet megadni, és az alapján számolja ki a hosszabbítás illetve a kiváltás összegét.



4. ábra: Lekérdezés

### Új ügyfél felvétel

Az 5. ábra új ügyfelek regisztrációját valósítja meg.



5. ábra: Új ügyfél felvétel

A név, az anyja neve, a cím és a személyigazolvány szám kötelezően kitöltendő mező. A személyigazolvány számnak minden ügyfélnek egyedinek kell lennie.

Az e-mail mező kitöltése nem kötelező, akkor kell megadni, ha az ügyfél szeretne hozzáférést a webalkalmazáshoz, ahol nyomon tudja követni zálogjegyeit.

Az „Új ügyfél” gombra kattintva felkerül az ügyfél az adatbázisba, és az „Zálogjegy felvétel” felületre kerülünk át.

# Fejlesztői dokumentáció

## Rendszerterv

A zálogházi nyilvántartó programot két különböző felületre terveztem. Az egyik az ügyfelek számára megtervezett mobil alkalmazás, a másik a zálogházban dolgozók számára létrehozott webalkalmazás, amin keresztül történik a zálogházi adminisztráció. Ez a két felület elkülönül egymástól, de ugyanahhoz az adatbázishoz férnek hozzá különböző jogosultságokkal.

Első lépések egyike átgondolni, hogy kik fogják használni a rendszert, vagyis kik férnek hozzá az adatbázishoz és milyen jogosultsági szabályok szerint. Akik használják:

* zálogházi dolgozók
  + zálogfiókban dolgozó (pénztáros, becsüs, ellenőrző becsüs)
  + zálogigazgató
* zálogházi ügyfelek

## Adatbázis terv

### Adattáblák

### Megvalósítás

A Java objektumok és a relációs adatbázis közötti adatok elérésére, kezelésére és megőrzésére **JPA** (Java Persistence API)-t használtam.

A JPA hídnak tekinthető az objektum-orientált tartománymodellek és a relációs adatbázis-rendszerek között.

Az adatbázisok csak skalár (pl stringek, egészek) értékeket tudnak tárolni és kezelni. Az ORM (Object Relational Mapping) valósítja meg, hogy az objektumokat egyszerű értékekre konvertálja, és így adatbázisban tárolhatóvá váljanak. Az ORM a hagyományos adatelérési módszerekkel szemben lecsökkenti a megírandó kód mennyiségét és a kód hordozhatóbbá válik.

Mivel a JPA a Java egy specifikációja, ezért ORM eszközként a **Hibernate**-et használtam, ami végrehajtja a JPA specifikációit a relációs adatbázisban az adatok fennmaradása érdekében.

Ha a jövőben valami oknál fogva ORM eszközt szeretnénk váltani ezt könnyen meg tudjuk tenni a JPA-nak köszönhetően. Ugyanazon specifikáció megvalósításával minden ORM eszköz követi a közös szabványokat.

Ahhoz, hogy a programot ne kelljen egy külső adatbázishoz csatolni fejlesztés és tesztelés során, **H2 Emdedded Database**-t használtam, ami egy Java nyelven írt relációs adatbázis-kezelő rendszer.

A H2 Emdedded Database futási időben hozza létre az adatbázist, minden szerver újra indulásával törlődik a felvitt adat.

## Biztonság

## Tesztelés

# Összegzés

## Fejlesztési lehetőségek

# Hivatkozások

*javaTpoint*. (2020. április). Forrás: https://www.javatpoint.com/

SanFranciscobólJöttem. (2020. március). *SFJ*. Forrás: https://sanfranciscoboljottem.com/