# Fejlesztői dokumentáció

## Adatbázis

## Backend

## Frontend

## LMS Desktop (asztali alkalmazás)

### Osztályok

#### Connection

* Az adatbázis adatai bele vannak írva a kódba, nincsenek titkosítva.
* Saját Select függvény, ami egy lekérdezést fogad és visszaad egy string tömbökből álló listát.
* RunSqlCommand függvény, ami a kapott lekérdezést egyszerűen lefuttatja.

#### HandleFiles

* Egy képfájl feltöltéséért felel a benne lévő Upload függvény.
  + Itt a kép átküldésre kerül a backendnek HttpClient segítségével.
  + Az esetleges hibákat json formátumban küldi vissza a backend.
  + Kivételkezelést is tartalmaz, ahol egyéb hibákat szűrünk ki.

#### HandleQueries

* Ez az osztály felel az egyes lekérdezések, beszúrások és frissítések megvalósítására.
  + A sima lekérdezések csak egy lekérdezést vagy egy fájlnevet várnak el. A fájlok az SqlQueries mappában találhatók.
  + A Delete függvény egy lekérdezés alapján kitörli kijelölt elemeket az adatbázisból.
  + Az összes többi függvény minden egyes Insert és Update műveletet valósít meg, ahol paraméterekben várja a feltöltendő / frissítendő adatokat.

#### Egyéb osztályok

* GeneratePassword -> Egy függvényt tartalmaz, ami a megadott karakterek és hossz alapján generál egy véletlenszerű jelszót.
* HandleGrids -> Függvényeket tartalmaz ComboBox és dataGridView feltöltésére, plusz egy megadott griden belüli keresésre.
* BorderPaint -> Keretet rajzol az adott form köré.
* CloseWindow -> Gombhoz rendelhető ablakbezárás, a megadott formot bezárja.
* DragWindow -> Lehetővé teszi az ablakok mozgatását címke (title) és panelek megadásával.
* HandleKeys -> A megadott gombra meghív egy megadott függvényt (pl.: Az Enter lenyomásakor elmenti a beállításokat, Escape gomb megnyomásakor pedig bezárja az ablakot).
* HandleFonts -> Beállítja a megadott form összes elemére a megadott betűtípust.

### Formok

#### Login

* Ez az első oldal, ami az alkalmazás indításakor megjelenik.
* Az alkalmazás az adatbázis nélkül nem fog elindulni.
* A LoginCheck függvényen belül ellenőrzésre kerülnek a felhasználó adatai. Ha a felhasználónév alapján található ilyen felhasználó, akkor BCrypt Verify segítségével összehasonlítjuk az adatbázisban tárolt titkosított jelszót a felhasználó által megadott jelszó titkosított változatával. Ha a két titkosított jelszó egyezik, akkor a jelszavak egyeznek. Így nincs szükség a titkosítás feloldására, ezáltal biztonságosabb a program.
* Ha a felhasználó rossz adatot ad meg, arról tájékoztatja a program.
* A UsernameTextChanged és PasswordTextChanged függvények felelnek a valós idejű hibák kiírására a felhasználónév és jelszóval kapcsolatban.
* A ShowPassword függvény pedig a szem ikont megváltoztatja, és megjeleníti a jelszót.

#### Main

* A Main form felel az összes művelet megvalósítására.
* Minden adat kezelhető külön oldalakon (Dashboard, Books, Borrowings, Reservations, Categories, Authors, Publishers és Users).
* Itt kerülnek meghívásra az egyes SQL lekérdezések.
* A címre kattintva a Dashboard nyílik meg
  + Itt külön-külön lekérdezés van a statisztikák kiírtatására (Könyvek, Felhasználók és Kölcsönzések száma)
  + Külön lekérdezés a 10 legtöbbet kölcsönzött könyv kiíratására.
* A jobb felső sarokban található üdvözlő szövegre kattintáskor tudja a felhasználó szerkeszteni saját fiókját.
* A Logout gomb pedig kijelentkezteti a felhasználót, majd megnyitja a Login formot.

#### Profile Settings

* A form megnyitásakor betöltésre kerülnek a bejelentkezett felhasználó adatai, amiket tud módosítani is.
* A validáció minden adatot ellenőríz, beleértve, ha a felhasználónevét módosítja, akkor ne tudja olyanra módosítani, ami már létezik.
* A validációt követően lefut a UpdateProfile függvény, ami frissíti a felhasználó adatait az adatbázisban.
* A ChangePassword felel a felhasználó jelszavának megváltoztatására.
  + Itt a validáció során ellenőrzésre kerül a Login-ban említett módon, hogy megadott jelenlegi jelszó tényleg megegyezik e a felhasználó jelszavával. Ezt követően a másik két szövegdoboz is ellenőrzésre kerül.
  + A ShowPassword lehetővé teszi, hogy a felhasználó meg tudja jeleníteni a beírt jelszót.
  + A jelszó írása közben a gombok lenyomására kiírja a felhasználónak, hogy minek és minek nem felel meg a beírt jelszó.
  + Végül a mentésre kattintva a jelszó frissül az adatbázisban (UpdatePassword függvény).

#### Books

* Az AddBook-ban meg lehet nyitni a ChooseAuthor és ChooseCategory ablakokat, ahol ki lehet választani egy vagy több szerzőt és kategóriát.
* Az adatok validálása után a könyv adatai feltöltésre kerülnek az adatbázisba (InsertBook függvény).
* Opcionálisan képet is lehet feltölteni:
  + A kép kiválasztásakor validálásra kerül a kép, itt ellenőrizve lesz a kép formátuma (png, jpg, jpeg), illetve a kép mérete (Max 5 MB).
  + Az ellenőrzés után át lesz másolva a temp mappába. A fájl neve itt a könyv isbn száma lesz.
  + Ezt követően meghívásra kerül a HandleFiles osztály Upload függvénye.
* Az EditBook szinte teljesen megegyezik az AddBook formmal, azzal a különbséggel, hogy ide betöltődnek a kijelölt könyv adatai (UpdateBook függvény).
* A RemoveBook egész egyszerűen eltávolítja a kiválasztott könyve(ke)t az adatbázisból (Delete függvény).

#### Borrowings

* Az AddBorrowing-ban meg lehet nyitni a könyvek kiválasztására szolgáló ChooseBooks formot, ebből a SelectedISBNs nevű string lista mezőből betöltésre kerülnek az ISBN számok a megfelelő szövegdobozba.
* A validációt követően a dátumok megfeleő formátumba lesznek váltva (InsertBorrowing függvény).
* Az ExtendBorrowing-ban a jelenlegi határidőhöz hozzáadásra kerül a felhasználó által megadott hónapok száma (UpdateBorrowing függvény).
* A ReturnBorrowings-ban a könyv visszahozásának dátuma mindig aznap lesz (Update Borrowing overloaded függvény).

#### Reservations

* Az AddReservation felel egy vagy több foglalás felvételére.
  + Az OpenChooseBooks megnyitja a könyvek kiválasztására szolgáló formot, ami egyébként megegyezik a Borrowings részben használt könyv kiválasztó formmal. Itt, ha több könyvet választ ki a felhasználó, azok külön rekordokként jelennek majd meg az adatbázisban.
  + A mentésre kattintva lefut a validáció, és ha minden rendben van feltöltésre kerül az adatbázisba (InsertReservation függvény).
* Az ExtendReservation form felel egy foglalás meghosszabbítására előre meghatározott 5 nappal. Az adott foglalás lejárati dátumához hozzáad 5 napot és frissíti azt az adatbázisban (UpdateReservation függvény).
* A LendReservation a kiválasztott foglalás alapján kölcsönbe adja az adott könyvet. A foglalás tartalmazza a felhasználónevet és az ISBN számot, a könyvtáros meg kell adjon egy határidőt, és ezek alapján bekerül a kölcsönzés az adatbázisba (InsertBorrowing függvény), majd törlésre kerülnek a foglalások táblából (Delete függvény).
* A RemoveReservations pedig eltávolítja a kiválasztott foglalás(oka)t az adatbázisból (Delete függvény).

#### Categories

* Az AddCategory validációja után, beleértve annak ellenőrzését, hogy létezik e már a megadott kategória, az feltöltésre kerül az adatbázisba (InsertCategory).
* Az EditCategory betölti a kiválasztott kategória nevét, majd az előző pontban említett validáció után frissítve lesz az adatbázisban (UpdateCategory).
* A RemoveCategory pedig eltávolítja a kiválasztott kategóriá(ka)t az adatbázisból (Delete függvény).

#### Authors

* Az AddAuthor-ban a validációt követően feltöltésre kerül a szerző (InsertAuthor függvény).
* Az EditAuthor-ban betöltődik a kiválasztott szerző, majd a validációt követően módosításra kerül (UpdateAuthor függvény).
* A RemoveAuthor a paraméterben kapott szerző(ke)t eltávolítja (DeleteFüggvény).

#### Publishers

* Az AddPublisher a mentésre kattintást követően végrehajtja a validációt, ami többek között leellenőrzi, hogy a megadott kiadó létezik e már az adatbázisban. Amennyiben a validáció sikeres, a kiadó feltöltésre kerül az adatbázisba (InsertPublisher függvény).
* Az EditPublisher ablakban a kiválasztott kiadó neve betöltésre kerül a szövegdobozba, majd az előző pontban említett validációt követően frissítésre kerül az adatbázisban (UpdatePublisher függvény).
* A RemovePublishers a kiválasztott kiadó(ka)t eltávolítja az adatbázisból (Delete függvény).

#### Users

* Az AddUser a validációt követően, beleértve azt, hogy a felhasználónév biztosan egyedi e, feltölti a felhasználó adatait az adatbázisba (InsertUser függvény).
  + Kötelező adatok: Vezetéknév, Keresztnév, Felhasználónév, Születési dátum, Lakcím.
  + Kötelező adatok adminként: szerep, ami lehet
    - Member – Sima felhasználó, tud kölcsönözni és foglalni.
    - Librarian – Könyvtáros, tudja kezelni a könyvtári rendszer jelentős részét (kivéve a felhasználók felhasználónevét és rangját).
    - Admin – Adminisztrátor, a teljes rendszerhez hozzáfér.
  + A felhasználó felvételét követően egy véletlenszerűen generált jelszó társul a fiókhoz, amit a webes felületen később meg tud változtatni.
* Az EditUser a kiválasztott felhasználó datainak betöltése után az előző pontban említett validációt követően frissíti a felhasználó adatait az adatbázisban (UpdateUser függvény).
  + Itt ugyan azok az adatok módosíthatóak, viszont csak az Admin tudja megváltoztatni a felhasználónevet vagy a rangot.
  + A felhasználót meg lehet jelölni megerősítettként, ami azt jelenti, hogyha a személyes adatai valósak, csak akkor tud kölcsönözni.
  + A Reset Password (jelszó visszaállítása) gomb arra szolgál, ha a felhasználó elfelejtette vagy elhagyta az ideiglenes jelszavát, akkor a könyvtárosnak lehetősége van újat generálnia számára.
* A RemoveUser pedig eltávolítja a kiválasztott felhasználó(ka)t az adatbázisból (Delete függvény).