Dokumentumnyilvántartó

Technikai dokumentáció

Informatikai Biztonsági és Adatvédelmi Tanácsadó Kft.

Tartalom

A program rövid leírása

Az alkalmazás egy dokumentum nyilvántartó rendszer, mely segítségével egy szerverre tudunk feltölteni és letölteni fájlokat.

A rendszer a feltöltött elemeket dupla titkosítással tárolja a szerveren, a fájlokat a felhasználó csak e program segítségével tudja újra dekódoltan letölteni. A programba való feltöltést követően az eredeti fájl kitörlődik a számítógépről. Mindenki csak a saját maga által feltöltött adatokat láthatja és töltheti le.

A felhasználóknak két különböző típusú mappájuk lehet a rendszerben. A „mappa”, melyben almappákat lehet létrehozni, illetve „dokumentumtároló mappa”, melyben nem lehet almappát létrehozni, ez fájlok tárolására szolgál.

Navigálás

A programban a menüpontok közti navigálást a fel le nyilakkal és az enter billentyűvel tudjuk megvalósítani. A programban a szoftveres kilépés gomb helyett használhatjuk még a CTRL+C billentyűkombinációt a program bezárására. Továbbá ehhez hasonlóan a vissza gomb helyett a mappák közti navigáláskor még a backspace-t használhatjuk a visszalépésre. Az F2 billentyű lenyomására nyitja meg a program a mappa átnevezése menüpontot.

Technikai információk

A program csak Microsoft Windows operációs rendszeren működik. Console Application ablakban fut. .NET környezetet használ. A programkód c# nyelven íródott. A kód VS2017 fejlesztőkörnyezetben lett fejlesztve, illetve abban is fejleszthető tovább.

A program normál működése során 16MB memóriát használ, azonban feltöltésnél illetve letöltésnél, nagyobb fáj esetén, ez jelentősen meg is növekedhet. A program és a hozzá tartozó útmutatók közel 35 MB méretőek.

A szoftver normál működése esetén szükségünk van még hálózati csatlakozásra is mellyen keresztül el tudjuk érni a központi szerver/adatbázist.

Titkosítás és biztonság

A program az adatokat a kliensoldalon titkosítja, így a szerverre már titkosítva küldi el, ezáltal csökkentve a feltöltés esetén az esetleges adatlopást és ezzel a szerver terhelését is csökkenti.

A fájlok titkosítására dupla titkosítást használ a program. Egy külső AES 256-os szimmetrikus titkosítást és minden felhasználó esetén egy általa megadott jelszó által kódolt titkosítást. Ezen rendszerek használatából fakadóan két különböző felhasználó által feltöltött ugyanazon dokumentum titkosított kódja más.

A feltöltött adatok egy mappában helyezkednek el a szerveren, ezáltal könnyű róla biztonsági mentést készíteni. A felhasználók adatai szintén a szerveren kerülnek eltárolásra SQL adatbázisban.

SQL

Az adatbázis futatására Microsoft SQL szervert használ a program. Az adatbázis 3 táblából áll Login, Folders és egy Documents táblából.

A Login táblában vannak eltárolva a felhasználók adatai, a Folders táblában a felhasználók által létrehozott mappák és dokumentumtárolók, a Documentums táblában a feltöltött dokumentumok adatait tárolja a program.

Folders:

Folder_id: A mappa id-ja

username: A feltöltő felhasználóneve

FolderName: A mappa neve

```
CREATE TABLE [dbo].[Folders] (
    [Folder_id]     INT          IDENTITY (1, 1) NOT NULL,
    [username]      VARCHAR (50) NOT NULL,
    [FolderName]    VARCHAR (50) NOT NULL,
    [Parent_id]     INT          NULL,
    [Doc_Container] BIT          NOT NULL,
    PRIMARY KEY CLUSTERED ([Folder_id] ASC)
);
```

Parent_id: Melyik mappa alá tartozik, id alapján?

Doc_container: „Dokumentumtároló” vagy „mappa”

Login:

username: A felhasználó felhasználóneve

Name: A felhasználó vezeték és keresztneve

Email: A felhasználó email címe

Password: A felhasználó jelszava

```
CREATE TABLE [dbo].[Login] (
    [username] VARCHAR (50) NOT NULL,
    [Name]     VARCHAR (50) NOT NULL,
    [Email]    VARCHAR (50) NOT NULL,
    [Password] VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([username] ASC)
);
```

Documents:

Folder_id: Melyik mappában van a fájl?

username: A felhasználó felhasználóneve

FileName: A feltöltött fájl neve

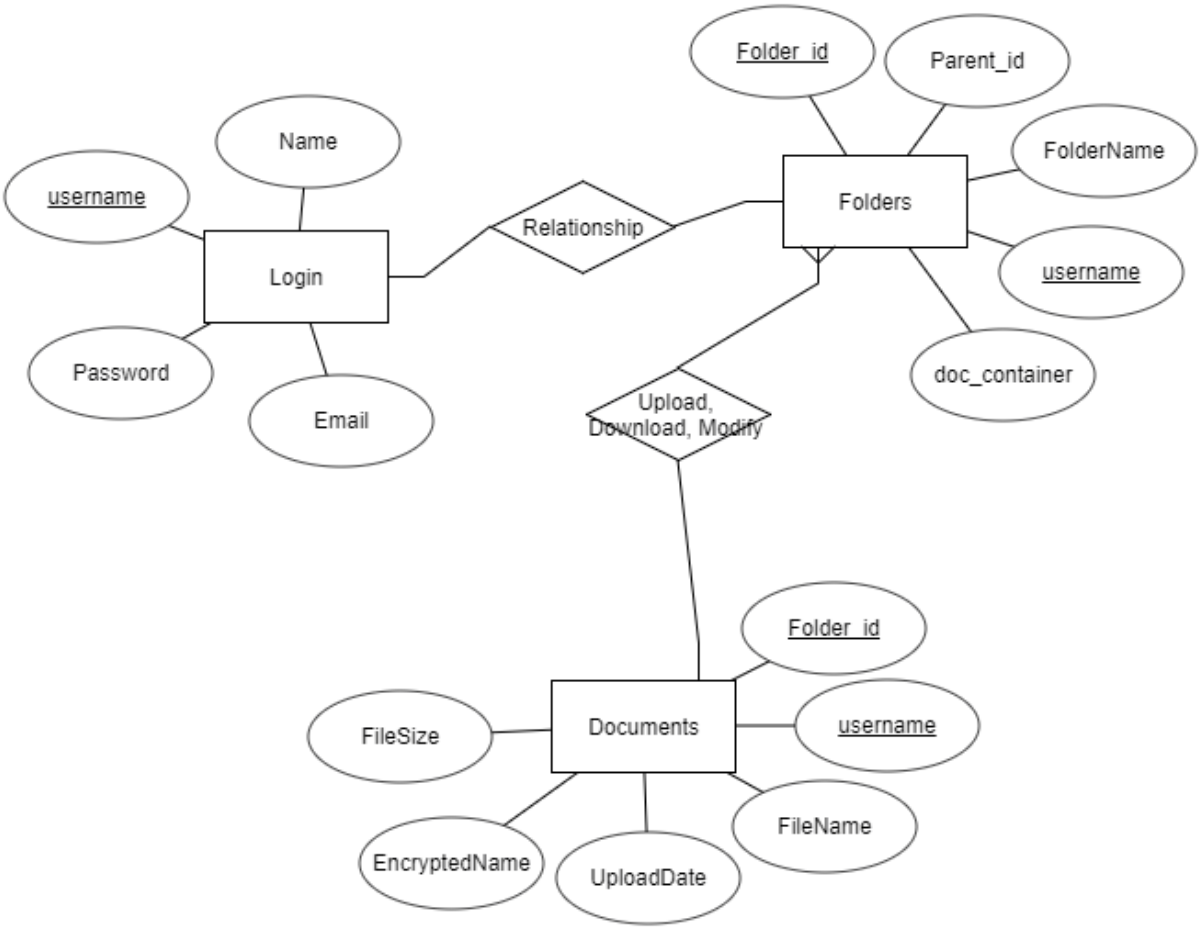
UploadDate: A feltöltés dátuma

EncryptedName: A feltöltött fájl titkosított neve

FileSize: A fájl méret (byte)

```
CREATE TABLE [dbo].[Documents] (
    [Folder_id]     INT           NOT NULL,
    [username]      VARCHAR (50)  NOT NULL,
    [FileName]      VARCHAR (50)  NOT NULL,
    [UploadDate]    VARCHAR (50)  NOT NULL,
    [EncryptedName] VARCHAR (100) NOT NULL,
    [FileSize]      INT           NOT NULL
);
```

SQL terv:


Folder_id
Parent_id
Name
FolderName
username
Folders
Relationship
Login
username
Password
doc_container
Email
Upload,
Download, Modify
Folder_id
FileSize
Documents
username
FileName
EncryptedName
UploadDate

Programkód

aes.cs

Ezen osztály a fájlok titkosításáért, illetve dekódolásért felel. Ennek segítségével valósul meg a letöltés, feltöltés, és a módosítás.

Az osztályban először két külső forrásból szerzett függvény található, először a *public byte[] AES_Encrypt* nevű függvény, ami a titkosításért felelős, majd a *public byte[] AES_Decrypt* , ami a dekódolásért.

Ezek a függvények a titkosításokat duplán végzik el, a második titkosítás egy plusz jelszó alapján történik, ami minden esetben egyenlő az aktuálisan bejelentkezett felhasználó jelszavával.

Ezek után négy saját függvény található, a fájlműveletek elvégzéséhez.

1. *public string EncryptText*
2. *public string DecryptText*
3. *public void EncryptFile*
4. *public void DecryptFile*

1.Szövegtitkosítás

Ez a függvény meghíváskor megkap egy adott szöveget, és az éppen bejelentkezett felhasználó jelszavát.

A szöveget bájt típusba konvertálja, majd ezeket az értékeket továbbítja az *AES_Encrypt* függvénynek. Ennek az eredményét visszakonvertálja string-é, és visszaadja a kapott, már titkosított string-et.

2.Szövegdekódolás

Ez a függvény teljesen hasonlóan működik, mint az előző.

Meghíváskor egy string-et kap, amiben egy titkosított string található, és megkapja az éppen bejelentkezett felhasználó jelszavát.

A szöveget bájt típusba konvertálja, majd az adatokat továbbítja az *AES_Decrypt* függvénynek, aminek az eredményét visszakonvertálja string-é és visszaadja az így már dekódolt szöveget.

3.Fájltitkosítás

Meghíváskor megkapja a felhasználó jelszavát és egy fájl elérési útját.

Beolvassa a fájl tartalmát, amit az *AES_Encrypt* függvénnyel titkosít. Megállapítja a fájl nevét, amit rögtön titkosít az *EncryptText* függvénnyel, majd kiszűri a „/” karakterekből adódó lehetséges hibát azzal, hogy ezeket lecseréli „_” karakterekre.

Majd ezek után létrehoz egy új fájlt, a titkosított tartalommal, a titkosított néven a program „szerver” mappájába.

4.Fájldekódolás

Meghíváskor ugyan azon értékeket kapja meg, mint társa, az *EncryptFile*. A titkosított fájl adatait beolvassa, majd az *AES_Decrypt* függvény segítségével dekódolja. Megállapítja a titkosított fájl nevét, amit a *DecryptText* függvénnyel dekódol, majd a dekódolt tartalommal és a dekódolt fájlnévvel létrehoz egy új fájlt, az eredeti teljesen dekódolt változatát, a rendszer alapértelmezett „Letöltések” mappájában.

LoginMenü.cs

Ez az osztály a bejelentkezési felületért felelős.

Első lépésként az IB Controll logóját jeleníti meg a program a *PrintLogo();* metódus segítségével, amely a „1.txt” fájlból való beolvasás után ASCII karakterek segítségével kirajzolja a logót.

A függvényen belül, külön lekezelésre kerül az, hogy jó fájlt olvastunk-e be, azaz ha nem jelenik meg a logó (nem létezik az 1.txt fájl), akkor alternatív szövegként az „IB Controll” jelenik meg a képernyőn.

ArraySelect(string chosen): Ez a metódus közvetlenül a bejelentkezés előtt kerül meghívásra. Eldönti, hogy a „Menuk1” tömbből a "Bejelentkezés", "Regisztráció", "Kilépés” funkciókat, vagy esetlegesen a hibás bejelentkezés után fellépő "Újrapróbálkozás", "Regisztráció", "Kilépés" lehetőségek között tudjunk választani.

public static void DoMenu(string[] selected,string chosen): A menü navigációját megvalósító függvény, ami láthatóvá teszi, hogy éppen melyik menüponton van a kurzor (szürkére festi az éppen kiválasztani kívánt menüpont hátterét). Továbbá a menüpontok közötti választásért is ez a függvény felel, lekezeli a fel- és le billentyűket és az enter billentyűt.

További függvények:

Console.SetCursorPosition(0, Console.CursorTop - selected.Length); Ez a függvény a kurzort állítja be a kezdő pozícióba.

p.ClearConsole(); A konzol törléséért felel - letörli az adott tartalmat a képernyőről.

sql.GetRootDirectory(); A gyökérkönyvtárba lehet visszalépni a segítségével.

LoginSql.cs

Ez a .cs állomány felel a bejelentkezés, illetve a regisztrációs felületért. Itt lekezeli a program, hogy szerepel-e már a felhasználónév az adatbázisban, valós-e az e-mail cím (megfelelő-e a formátum, tartalmazza-e a szükséges karaktereket). Értelemszerűen, ha ezek az adatok hibásan kerülnek bevitelre, akkor a program addig fogja bekérni az adatokat, amíg a felhasználó nem adja meg helyesen a bejelentkezéshez/regisztrációhoz szükséges adatokat (például: üres karaktert nem lehet megadni, legalább egy karakter szükséges a bejelentkezéshez).

Ebben a részben kerül lekezelésre az is, hogyha a program nem tudja elérni az adatbázist, akkor közli a felhasználóval: "Nem sikerült csatlakozni az adatbázishoz."

A LoginSql.cs –ben előforduló függvények:

public bool IsUsernameAlreadyTaken(string username) Bool típusú függvény, amely eldönti, hogy a felhasználónév már szerepel-e az adatbázisban, ha igen, akkor másikat kell választani.

public bool IsValidEmail(string email) Ez a metódus felel az e-mail cím helyes formátumának bekéréséért.

public void StartLogin() A bejelentkezést biztosítja, ha sikerül csatlakozni az adatbázishoz, akkor következő lépésként bejelentkezhet a felhasználó.

public void DoLogin() A sikeres bejelentkezés feltétele a helyes felhasználónév, illetve jelszó megadása, ha ezek teljesülnek (nem hagyjuk egyik mezőt sem üresen), akkor tudunk csak a teljes programhoz hozzáférni.

public void DoReg() Regisztrációért felel, itt adhatunk meg felhasználónevet, vezetéknevet, keresztnevet, jelszót és e-mailt, majd a függvény megvizsgálja, hogy minden adat megadása helyesen történt-e. Ezen belül a *ToLower()* illetve a *ToUpper()* felelnek a kis és nagybetűs átalakításért a nevek kezdőbetűinél, ha valaki teljesen kisbetűvel adja meg a vezeték illetve a keresztnevét, akkor ez automatikusan átállítja azok kezdőbetűit nagybetűvé.

A *namespace Orb.App* névtér felel azért, hogy a jelszavak bekérése során, a jelszó csillagozott(rejtett) formában jelenjen meg a konzolon, ezzel elrejtve. A függvény, amely visszaadja mindegy egyes bekért karakter helyett a *karaktert, az a *public static string ReadPassword()*.

Tartalmaz még továbbá SQL injection-t is, ami az adatbázisba beírja a regisztrált felhasználó adatait.

Menu.cs

Amikor a felhasználó már bejelentkezett, akkor lehetősége nyílik mappát vagy dokumentumtárolót létrehozni, dokumentumot feltölteni, letölteni, visszalépni az előző könyvtárba (ha nem a gyökér könyvtárban van) és kijelentkezni.

Itt van lekezelve az is, hogy a „Backspace” billentyűre automatikusan visszalépjen egyet a könyvtárban a felhasználó, valamint az új mappa ellenőrzött bekérése is itt történik meg.

Továbbá a nyilak navigációja, kezelése és az Enter szerepe a menüpontban. Az F2 billentyű lenyomására átnevezhetjük a mappát.

Ezen feladatok mindegyikéről a *public void DoMenu(List<string> selected*) gondoskodik.

A fájlban előforduló függvények:

public void SetInRoot(bool tf) Ez a függvény fogja módosítani az „inroot” változó értékét(bool értékű paramétere van).

public void DebugTable(DataTable table) Ez a függvény teszteléshez van használva, mely arra hivatott, hogy kiírathassuk egy DataTable minden értékét.

public void Clear(int db) Segítségével tudunk sorokat a konzolról törölni.

public bool Yesnomenu() Az igen nem és vissza menüpontokért felel, illetve ezek között végzett navigálásért.

public bool Modify_Download_Menu(string filename) A letöltés módosítás és vissza menüpontok megvalósításáért felel. Hasonlóan a *YesNoMenu()*-höz.

public bool Rename_Exit() EZ a függvény teszi lehetővé, hogy átnevezzünk egy mappát illetve egy dokumentumtárolót.

Program.cs

Ebben az osztályban található a Main függvény. Itt állítjuk be a program és az adatbázis közti kapcsolatot illetve itt tárolunk Listákat és változókat, amiknek segítségével navigálhatunk majd a mappák között.

1. Main függvény

Ebben a függvényben kezdődik minden, először is meghívjuk az *u.CreateDirectory* nevű függvényt, ami ha nincsen előre létrehozott „szerver" mappa az asztalon, akkor létrehoz egyet a program számára, ahova majd később a titkosított, fájlok kerülnek.

Ezután az *m.PrintLogo* függvény megpróbál kiíratni egy ASCII-art-ot, ami az IB Controll logóját ábrázolja, ha ez nem történne meg, ehelyett kiír egy egyszerű „IB Controll" szöveget.

Végül elindul a belépési folyamat, az *l.StartLogin* függvénnyel, ami után lehetőségünk lesz bejelentkezni illetve regisztrálni egy új felhasználót a program rendszerébe.

2.Listek, változók

Ebben az osztályban két List található.

- *public List<string> FolderList*
- *public List<int> IdList*

FolderList, ahol a mappák nevei lesznek eltárolva, és *IdList*, ahol a mappák azonosítására szolgáló „Folder_id"-k vannak eltárolva.

Négy változó került deklarálásra.

- *public int actualparentid;*
- *public string actualparent;*
- *public int actualchildid;*
- *public string actualchild;*

Ezek a mappák közötti szülő-gyerek kapcsolat tárolására alkalmasak. „Szülő" nevet és id-t, illetve „gyerek" nevet, és id-t tárolnak. Ezekhez a változókhoz tartoznak beállító és lekérdező függvények is.

3.Adatbázis kapcsolat

Itt adunk meg egy „connection string"-et, ami arra szolgál, hogy a program megtalálja az adatbázist, és kommunikálni tudjon vele.

Ez a cím egy relatív hivatkozással van beállítva, ezzel elkerülve annak a hibának a lehetőségét, hogy ha a programot áthelyezzük, vagy egy másik számítógépen futtatjuk, akkor ne találná meg az adatbázist.

SQL_query.cs

Ezt a .cs-t értelmezhetjük úgy is mint a program motorját. A program nagy része az adatbázissal való kommunikációra épül. Ebben az osztályban csak adatbázissal való kommunikáció van, ez segíti az összes többi .cs-nek a munkáját, a bejelentkezéstől kezdve a fájlműveletekig.

Tíz nagyobb függvény található benne, ezen felül még kissebb változó értéklekérdező függvények is találhatók benne.

A függvényekben főleg SQL lekérdezések szerepelnek, amik paraméteresen kapnak értékeket így védve a programot az „SQL Injection" ellen.

1.GetUsersFullName

Az egyik legegyszerűbb függvény ebben az osztályban, amit a sikeres bejelentkezés után láthatunk működni. A függvényben egy SQL lekérdezés szerepel, ami lekérdezi az adatbázisból a felhasználónk teljes nevét, a felhasználóneve alapján. Ezt a függvényt az üdvözléskor használjuk.

2.GetUsersPassword

Lekérdezi az adatbázisból a bejelentkezett felhasználó jelszavát, a felhasználónév alapján.

3.FileOrFolder

Komoly szerepet játszik ez a függvény a mappák közti navigálásban. Mivel a menü magában nem tudja, hogy amit épp kiválasztunk az egy fájl, egy mappa, vagy egy dokumentumtároló. Ez a függvény megmondja neki, és ez alapján kínál fel nekünk lehetőségeket a menü, mint például egy mappa megnyitása, egy fájl letöltése vagy módosítása.

4.UjMappa

Ellenőrzötten bekér a felhasználótól egy mappanevet, majd megkérdezi, hogy a létrehozni kívánt mappa dokumentumtároló legyen-e vagy sem, majd ezen értékeket felhasználva létrehoz az adatbázisban egy új mappát, a felhasználót meg visszalépteti a főmenübe.

5.FileInfoToDb

Ezt a függvényt a fájlok feltöltéséhez használjuk. A fájl feltöltésnél kinyerjük az adott fájl, különböző információit, mint például a név, a méret, vagy a titkosított név. Ez a függvény ezeket kapja meg és egyszerűen létrehoz az adatbázisban egy új elemet ezekkel az adatokkal.

6.OpenFolder

Az *OpenFolder* függvény megnézi, hogy melyik mappában vagyunk jelenleg, majd két lekérdezéssel kigyűjti, hogy ahhoz a mappához milyen almappák és milyen fájlok tartoznak, ezeket átadja egy listának, amit a menü függvény kiír nekünk.

7.GetRootDirectory

Rögtön a bejelentkezés után használjuk. Megnézi, hogy az éppen bejelentkezett felhasználónknak milyen mappái vannak a gyökérkönyvtárban, és ezekkel indítja el a menüt.

8.GetFName

Ha egy elemnek tudjuk az azonosítóját, de a neve nincs, eltárolva ezt a függvényt hívjuk meg, ami az azonosító, és a bejelentkezett felhasználó alapján lekérdezi, hogy ahhoz az azonosítóhoz milyen név tartozik.

9.GetDokTároló

Fontos részét képezi a programnak ez a függvény. Dokumentumtárolóba nem hozhatunk létre almappát, csak dokumentumot tölthetünk fel. Ha a mappa nem dokumentumtároló, nem tölthetünk fel bele fájlt, csak almappát hozhatunk létre bele. Ez a függvény vezérli ezt a folyamatot.

10.ChangeFolderName

Megkérdezi a felhasználót, hogy melyik mappát szeretné átnevezni, ezután megkérdi, hogy mi legyen a mappa új neve, amennyiben az újnév nem foglalt még, egy módosító lekérdezést futtat le, a mappa neve megváltozik, majd visszalépteti a felhasználót a menübe.

UDMD.cs

Az UDMD.cs állomány a mappákban lévő fájlok letöltéséért, feltöltéséért, módosításáért, illetve dokumentumtároló és mappa létrehozásáért felel. Egy mappán belül nem hozható létre ugyanolyan nevű mappa.

A UDMD.cs-ben előforduló függvények:

public void Feltolt() A feltöltésért felel, ennek segítségével lehet feltölteni fájlokat a dokumentumtárolókba, ha sima mappába szeretnénk dokumentumot elhelyezni a program hibát fog dobni. Ha sikeres volt a fájl feltöltése a szerverre, vagyis a kívánt útvonalra (kiválasztott könyvtárba), akkor erről tájékozat minket a program, hogy ez sikeresen megtörtént. A program megvizsgálja, hogy sikeres volt a feltöltés és amennyiben sikerült, akkor kitörli a számítógépről az eredeti fájlt.

A *public void Letolt(string fajlnev)* a letöltésért felel, segítségével a dokumentumokat letölthetjük a Letöltések mappába. Hasonlóképpen a feltöltéshez, ha sikeresen megtörtént a letöltés, akkor a program ezt kiírja a képernyőre a felhasználónak.

public void ModifyFile(string fajlnev) A fájlokat módosítja, törli a régit és a helyébe feltölt egy újat.

Support

Amennyiben bármi működési rendellenességet tapasztal, vagy észrevétele lenne, kérem, keresse fel a fejlesztőket.

A program folyamatos fejlesztés alatt áll. A jövőben várható plusz funkciók, hibajavítások:

- Szerveren tárolt dokumentumok szelektív válogatása
- Dokumentumok hash-elése
- Fájlmegosztás
- Fájl törlése
- Automatikus Backup
- Vizuális felület
- Logolás

A programot az IB Controll megbízásából készítették a Neumann János Egyetem GAMF karának hallgatói:

Antal Gergő
Antal Máté
Tímár Roland