**Baranya Vármegyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

**Vizsgaremek**

Készítették: Németh Noel,

Lőrincz Noel,

Rapcsák János

**Pécs**

**2025**

**Baranya Vármegyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

*Szakma megnevezése: Szoftverfejlesztő és –tesztelő*

*A szakma azonosító száma: 5 0613 12 03*

**Vizsgaremek**

Aliasly

Készítették: Németh Noel,

Lőrincz Noel,

Rapcsák János

**Pécs**

**2025**

**2025**

**EREDETISÉGI NYILATKOZAT**

Alulírottak: Németh Noel, Lőrincz Noel, Rapcsák János

A Baranya Vármegyei SzC Simonyi Károly Technikum és Szakképző Iskola, Szoftverfejlesztő és tesztelő végzős tanulói, büntetőjogi és fegyelmi felelősségünk tudatában nyilatkozunk és aláírásunkkal igazoljuk, hogy a(z)

Aliasly

című vizsgaremek saját, önálló munkánk, az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul vesszük, hogy vizsgaremek esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;

- tartalmi idézet hivatkozás megjelölése nélkül;

- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentjük, hogy a plágium fogalmát megismertük, és tudomásul vesszük, hogy plágium esetén a vizsgaremekünk visszautasításra kerül.

Tanulók aláírásai

## Tartalomjegyzék

## Bevezetés

### Projekt bevezető

#### 1.1 Megoldandó feladat és probléma

A modern vállalati környezetekben gyakori, hogy egy dolgozónak több céges fiókja is van. Ezek kezelése nehézkes, időigényes, és biztonsági kockázatot jelenthet, ha a jelszavak vagy hozzáférési jogok nem megfelelően vannak kezelve. A projekt célja egy olyan centralizált fiókkezelő alkalmazás létrehozása, amely segíti a rendszergazdákat a felhasználói adatok biztonságos és átlátható kezelésében.

#### 1.2 Miért ezt választottuk

A vállalati adatvédelem egyre nagyobb hangsúlyt kap, de a meglévő eszközök gyakran túlbonyolítottak vagy költségesek. Saját fejlesztéssel rugalmasan lehet alkalmazkodni a cégek egyedi igényeihez.

A csapatunk számára lehetőség volt a biztonsági protokollok (pl. titkosítás, szerepkör-alapú hozzáférés) és a skálázható architektúra tervezésének gyakorlati alkalmazására.

#### 1.3 Célunk

Egy egyszerű, intuitív felület biztosítása a rendszergazdáknak. Biztonság: AES-256 titkosítás és mesterkulcsos beléptető rendszer alkalmazása. Felhasználóbarát felület: Intuitív kezelőfelület a rendszergazdák számára. Skálázhatóság: Több ezer felhasználó kezelése kis és közepes vállalatoknál.

### Fejlesztési folyamat és Tapasztalatok

#### 2.1 Mit tanultunk és új ismeretek

Közös döntéshozatal: Rájöttünk, hogy a különböző szakmai háttérrel rendelkező csapattagok (pl. backend vs. frontend fejlesztők) más szempontokat hangsúlyoznak. Például egy biztonsági funkció implementálása során a backend csapat a teljesítményre, míg a frontend a felhasználói élményre fókuszált. Rendszeres design sprint ülésekkel sikerült közös nevezőre jutni.

Feedback kultúra: A napi minimum 15 perces meetingek segítettek a blokkoló problémák gyors feloldásában.

Konfliktuskezelés: Megtanultuk, hogy a technikai nézeteltérések nem személyesek, hanem a projekt minőségét szolgálják.

### Csapat és Munkamegosztás

#### 3.1 Csapatszerepek

**Németh Noel – Backend; Adattitkosítás; Adatbázis:** N. Noel a programon belüli adatbázis kezelési funkciókat tervezte és implementálta, emellett az adatbázis finomításán is dolgozott. Ő volt a felelős az adat titkosítás teljes kivitelezésére.

**Lőrincz Noel – Frontend; Design; Backend:** L. Noel a frontend Design és UI kialakításához készített munkát, az átláthatóság és könnyedén megérthető UI érdekében. A backend részen is hozzáadott a projekthez a funkcionalitás terén.

**Rapcsák János – Frontend; Backend; Adatbázis; Dokumentáció:** János készítette el az applikáció felhasználói felületét és azoknak a funkcióit a háttérkódban, valamint a kód logikáján is dolgozott. Végül, ő tervezte és készítette el az adatbázis felépítését. A dokumentációt szerkesztette össze és formázta meg.

### Műszaki megvalósítás

#### 4.1 Architektúra

Backend: A szerveroldali logikát a C# .NET 8.0 keretrendszerrel fejlesztjük.

Frontend: A kliensoldali felület Windows Presentation Foundation (WPF) keretrendszerrel készül. Data Binding és XAML használatával a dinamikus felületek kialakításához.

Adatbázis: Az adatok tárolására MySQL relációs adatbázist használunk, amely megbízható és nagy teljesítményű megoldást nyújt. A phpMyAdmin felület segítségével könnyedén kezelhetők az adatbázisok.

#### 4.2 Biztonsági intézkedések

Adattitkosítás: Az adatok védelmére AES-256 titkosítást használunk.

Hozzáférés-vezérlés: A nem kívánt belépések megelőzéséhez egy mesterkulcsos belépés rendszert hoztunk létre.

Naplózás: Minden fontos műveletet pontosan és átlátható módon naplózunk az adatbázison belül (pl. bejelentkezések, adatmódosítások).

### Következő lépések

Multi-Faktor Hitelesítés (MFA): A rendszergazdák számára kötelezővé tehető MFA bevezetése.

Integrációs Lehetőségek: Kapcsolódás vállalati HR rendszerekhez a felhasználói adatok automatikus szinkronizálásához.

Mobile Admin Felület: Alkalmazás mobilra, hogy a rendszergazdák akár távolról is kezelhessék a fiókokat.

Felhasználói Önkiszolgáló Portál: Jelszó-változtatás vagy jogosultságkérelmek önálló kezelése.

## Felhasználói dokumentáció

### A program általános specifikációja

#### 1.1 Áttekintés

Az Aliasly egy olyan szoftveralkalmazás, amelyet kifejezetten rendszergazdák számára fejlesztünk ki, hogy biztonságosan kezelhessék a felhasználói fiókokat. A program egy mesterkulcs alapú hitelesítési rendszerrel védi a hozzáférést, lehetővé téve a felhasználói adatok könnyű kezelését, új fiókok tárolását és meglévők törlését. A felület egyszerű és intuitív, így minimális betanulási idővel használható. A rendszer három részből áll:

- Bejelentkezési felület, ahol a rendszergazda mesterkulccsal azonosítja magát.

- Új mesterkulcs létrehozási felület, ahol a rendszergazda létrehozhat egy vagy több különböző kulcsot.

- Főfelület, ahol a felhasználói adatok megjelennek, szerkeszthetők és törölhetők.

A program célja, hogy egy központi helyen biztosítsa a felhasználói adatok biztonságos tárolását és kezelését, miközben a lehető legegyszerűbb működést nyújtja.

#### 1.2 Mesterkulcs kezelése

A program elindításakor a felhasználó egy bejelentkezési képernyővel találkozik, ahol meg kell adnia a mesterkulcsot. Ha ez az első indítás, és még nincs mentve mesterkulcs, a rendszer felkínálja annak létrehozását. A kulcsot ezután biztonságosan titkosítva tárolja az adatbázisban.

Ha a felhasználó helyes mesterkulcsot ad meg, a program betölti a főfelületet, ahol az összes hozzá tartozó felhasználói fiók látható. Hibás kulcs esetén a rendszer figyelmeztető üzenetet jelenít meg, és lehetőséget ad újra próbálkozásra.

#### 1.3 Felhasználói fiókok kezelése

A főfelületen a rendszergazda áttekintheti az összes tárolt felhasználói fiókot. Az adatok táblázatos formában jelennek meg, lehetővé téve a gyors áttekintést. Minden felhasználóhoz tartozik egy egyedi azonosító, felhasználónév, jelszó, további metaadatok (például. weboldal url, leírás), valamint egy törlés gomb.

Az új felhasználó hozzáadásához a bal oldalon egy űrlap található, ahol meg kell adni a szükséges adatokat (például felhasználónév, jelszó, e-mail cím). A rendszer ellenőrzi, hogy minden kötelező mező ki legyen töltve. Az "Adat felvitele" gombra kattintva az új felhasználó hozzáadódik az adatbázishoz, és azonnal frissül a lista.

Ha egy felhasználót törölni kell, a megfelelő sorban lévő törlés gombra kattintva a rendszer megerősítést. A művelet végleges, így a felhasználó adatai visszavonhatatlanul törlődnek.

#### 1.4 Kilépés

A főfelületen található egy kilépés gomb, amelyre kattintva a rendszer visszairányítja a felhasználót a bejelentkező képernyőre. A kilépéskor a rendszer ellenőrzi, hogy minden változtatás mentésre került-e, és biztonságosan zárja le a munkamenetet.

#### 1.5 Biztonsági intézkedések

A program kiemelt figyelmet fordít az adatvédelemre. A mesterkulcsot és a felhasználó összes adatát titkosítva tárolja a program, így akár adatbázis-szintű illetéktelen hozzáférés esetén is védve maradnak.

#### 1.6 Naplózás

A naplózási rendszer minden jelentős eseményt rögzít, kezdve a belépési és kilépési műveletekkel. Amikor egy rendszergazda sikeresen bejelentkezik a rendszerbe, a napló pontosan rögzíti a belépés időpontját, a használt mesterkulcshoz tartozó azonosítót, valamint a munkamenet egyedi azonosítóját. A kilépéskor a rendszer automatikusan menti a kilépés pontos időpontját.

Az adatmódosítások részletes dokumentálása a naplózási rendszer másik alapvető feladata. Új felhasználói fiók létrehozásakor a rendszer pontosan rögzíti a létrehozás időpontját, az új felhasználóhoz tartozó azonosítót és a műveletet végző rendszergazda azonosítóját. Felhasználói fiók törlésekor a napló rögzíti a törlés időpontját, a fiók azonosítóját és a használatban lévő mesterkulcs azonosítóját

#### 1.7 Felhasználói felület

A program célja, hogy a lehető legegyszerűbb legyen a használata. A felület egyszerű, a fontos funkciók könnyen elérhetők, és minden művelethez egyértelmű utasítások vagy eszköztippek jelennek meg. A lista nézet lehetővé teszi a gyors szűrést és rendezést, így nagy számú felhasználó kezelése is hatékony marad.

#### 1.9 Összegzés

Ez a specifikáció részletesen leírja az Aliasly működését, biztonsági mechanizmusait és felhasználói felületét. A program célja, hogy a rendszergazdák számára hatékony és biztonságos eszközt nyújtson a felhasználói fiókok kezelésére, miközben a lehető legegyszerűbb marad a használata. A továbbfejlesztési lehetőségek garantálják, hogy a rendszer a jövőben is igazodhasson a változó igényekhez.

### Rendszerkövetelmények

#### 2.1 Hardverkövetelmények

Minimum:

Processzor: Intel Core i3 2.Generációs vagy AMD Ryzen 3 2.Generációs

Memória: 2 GB ddr3 RAM

Videókártya: Integrált

Tárhely: 2 – 6 GB Szabad tárhely

Ajánlott:

Processzor: Intel Core i5 5.Generációs vagy AMD Ryzen 5 3.Generációs

Memória: 4 – 8 GB ddr3 vagy ddr4 RAM

Videókártya: Integrált vagy Dedikált

Tárhely: 4 – 8 GB Szabad tárhely

#### 2.2 Szoftverkövetelmények

Operációs rendszer: Windows 10 vagy Windows 11

.Net keretrendszer: .NET Runtime 8.0.15 vagy .NET Runtime 9.0.4

Adatbáziskezelés: XAMPP 8.2.12 vagy MAMPP 4

### A program telepítése

### A program használatának a részletes leírása

## Fejlesztői dokumentáció

### Témaválasztás indokolása

### Az alkalmazott fejlesztői eszközök

### Tervezési módszer

### Adatmodell leírása

### Részletes feladatspecifikáció algoritmusok

### Forráskód

### Tesztelési dokumentáció

### Továbbfejlesztési lehetőségek

## Összegzés

### Irodalomjegyzék, forrásmegjelölés