Tartalomjegyzék

[1. Bevezető 3](#_Toc131087853)

[2. Választott téma indoklása 3](#_Toc131087854)

[3. Fejlesztői dokumentáció 4](#_Toc131087855)

[Fejlesztés menete és lépései: 5](#_Toc131087856)

[Követelménymeghatározás 5](#_Toc131087857)

[Szoftverspecifikáció 6](#_Toc131087858)

[Alkalmazás megtervezése 8](#_Toc131087859)

[Látványtervek és arculati elemek készítése 8](#_Toc131087860)

[Adatbázisterv elkészítése 9](#_Toc131087861)

[Adatbázis felépítése 10](#_Toc131087862)

[A szoftver osztálydiagramja 11](#_Toc131087863)

[API kiválasztása és konfigurálása 12](#_Toc131087864)

[Felhasználóifelület tervezése WPF-ben 13](#_Toc131087865)

[Regisztrációs felület 14](#_Toc131087866)

[Jelszó titkosítás 15](#_Toc131087867)

[Bejelentkezési felület 15](#_Toc131087868)

[Kezdőképernyő felülete 16](#_Toc131087869)

[Portfolió kezelő felület 17](#_Toc131087870)

[Árfolyam átváltó felület 19](#_Toc131087871)

[Jelszó módosítás 19](#_Toc131087872)

[API hívás 20](#_Toc131087873)

[PDF generálás 22](#_Toc131087874)

[Felhasznált NuGet Package-ek 23](#_Toc131087875)

[Newtonsoft.JSON (13.0.2) 23](#_Toc131087876)

[LiveCharts 23](#_Toc131087877)

[iTextSharp (5.5.13.3) 23](#_Toc131087878)

[Tesztelési, hibafeltárási folyamat 24](#_Toc131087879)

[Elvégeztt Unit Test-ek 24](#_Toc131087880)

[4. Továbbfejlesztési lehetőségek 25](#_Toc131087881)

[Bitcoin-tárca (Bitcoin Wallet) 25](#_Toc131087882)

[Jutalompontok beváltása 25](#_Toc131087883)

[Aktuális kriptovaluta árfolyamok azonos időközönként történő frissítése 25](#_Toc131087884)

[5. Felhasználói dokumentáció 26](#_Toc131087885)

[Szoftver célközönsége 26](#_Toc131087886)

[Rendszerkövetelmények: 26](#_Toc131087887)

[Szoftveres követelmények 26](#_Toc131087888)

[Hardveres követelmények 26](#_Toc131087889)

[Telepítési útmutató: 27](#_Toc131087890)

[CoinAmy telepítése 27](#_Toc131087891)

[XAMPP szoftvercsomag telepítése 27](#_Toc131087892)

[Az alkalmazás használata: 28](#_Toc131087893)

[Bejelentkezés 28](#_Toc131087894)

[Felhasználói regisztráció 29](#_Toc131087895)

[Árfolyamok ablak (Kezdőképernyő) 30](#_Toc131087896)

[Portfolió kezelő felület 31](#_Toc131087897)

[Befektetés hozzáadása 32](#_Toc131087898)

[Befektetés eltávolítása 32](#_Toc131087899)

[Befektetése szerkesztése 33](#_Toc131087900)

[Exportálás PDF dokumentumként 33](#_Toc131087901)

[Árfolyamátváltó felület 34](#_Toc131087902)

[Belépési jelszó megváltoztatása 35](#_Toc131087903)

[Hibaüzenetek 35](#_Toc131087904)

[Fejlesztői elérhetőségek 36](#_Toc131087905)

[6. Záródolgozat összegzése 37](#_Toc131087906)

[7. Hivatkozásjegyzék 38](#_Toc131087907)

[8. Ábrajegyzék 39](#_Toc131087908)

[9. Köszönetnyilvánítás 40](#_Toc131087909)

# Bevezető

A napjainkban egyre inkább, széles körben terjedésnek indult digitális és kriptovaluták iránt rohamosan növekvő kereslet magával hozta a magánszemélyek és a legkülönfélébb intézményi, institúciós befektetők érdeklődést is. Ez a folyamatosan növekvő kereslet magában hordozza az igényt egy kellően megbízható, átlátható és felhasználóbarát portfoliókezelő szoftver fejlesztésére, amely kellő mértékben kiszolgálja a felhasználó igényeit. Záródolgozatom témájaként ezen szoftvertermék megtervezését, fejlesztését és dokumentálását választottam.

# Választott téma indoklása

Záródolgozatom központi témaadójaként azért esett a választásom erre a meglehetősen széles témakörre, mert több, mint két éve rendszeresen nyomon követem a kriptovaluták piacát, az aktuális árfolyamok és hírek alakulását. Mivel a kriptopénzek piaca gyors ütemben gyarapszik, és a befektetők száma is növekvő tendenciát mutat, így egyre inkább szükségessé válnak az olyan szoftverek, amelyek segítenek az embereknek követni a kriptovaluta befektetéseiket.

Számomra meglehetősen szignifikáns szempont volt, hogy olyan témakört válasszak a záródolgozatom témájaként, amelyet a hétköznapi érdeklődésikörömhöz valamilyen formában kapcsolni tudok. Választásom során közrejátszott, hogy meglehetősen izgalmas feladatnak véltem egy olyan szoftver elkészítését, amelynek témája a mindennapjaimban is szerepet játszik. Emellett azt is pozitívumként vélem, hogy ezirányú érdeklődésem kellő mértékű motivációt biztosított a fejlesztés során, ezért projekttel történő foglalkozást gyakran a hobbim részeként tekintettem.

# Fejlesztői dokumentáció

Az alkalmazás fejlesztésének megkezdése előtt döntést kellett hoznom abban a kérdésben, hogy mely programozási nyelvet és keretrendszert kívánom használni az implementáció során. Ennek során a C# nyelv és a Windows Presentation Fountation keretrendszer mellett tettem le a voksomat.

Az alkalmazás fejlesztéséhez a következő eszközöket használtam:

* Fejlesztőkörnyezet (IDE): Microsoft Visual Studio Community 2022
* Adatbáziskezelő: phpMyAdmin 5.2.0
* Diagramkészítő: Microsoft Visio 2021 Professional

Programnyelv és keretrendszer:

* C#
* .NET Framework 4.8
* Windows Presentation Foundation (WFA)

Fejlesztéshez használt hardverkörnyezet:

* Processzor: Intel® Core™ i5-10400
* Memória: 32 GB DDR4
* Videókártya: GeForce RTX™ 3060

A CoinAmy – Kriptovaluta portfolió- és befektetéskezelő egy olyan asztali alkalmazás, amely Windows operációsrendszeren történő futtatásra készül.

A szoftver egy mögöttes MySQL adatbázishoz kapcsolódva tárolja a felhasználó által rögzített adatokat. Ezen lokális adatbázis használatához a XAMPP szoftvercsomag megléte és háttérben történő futtatása szükséges.

Ezenfelül számos adatot egy API felhasználásával szerez be a szoftver, úgy, mint a különböző kriptovaluták adatai és aktuális árfolyamai. Mindebből adódóan az alkalmazás használatához aktív internetkapcsolat léte is elengedhetetlen.

## Fejlesztés menete és lépései:

### Követelménymeghatározás

A szoftver fejlesztése során a cél egy kriptovaluta portfolió kezelő asztali alkalmazás elkészítése, amely kiszolgálja a témában érdekelt felhasználók igényeit.

Az alkalmazással szemben támasztott funkcionális követelmények:

**Regisztráció**: A felhasználó a program elindítása után lehetőséget kap a regisztrációra, ahol megadhatja a továbbiakban, a bejelentkezéshez használni kívánt adatokat.

**Bejelentkezés:** A későbbiekben, a felhasználói bejelentkezéshez, a regisztráció során megadott felhasználónév, e-mail cím és jelszó megadása után léphető tovább az alkalmazásba.

**Kriptovaluták adatainak megjelenítése:** Főbb, nagyobb piaci kapitalizációval[[1]](#footnote-1) rendelkező kriptopénzek aktuális árfolyamainak, azok 24 órán belüli százalékos elmozdulásának, és egyéb kapcsolódó információk listázása az alkalmazásban, egy API felhasználásával.

**Adatábrázolás:** A kiválasztott kriptovaluta árfolyamának visszamenőleges megtekinthetősége árfolyam grafikonon.

**Befektetések kezelése:** Lehetőség a felhasználó egy-egy befektetésének adatainak rögzítésre, eltávolítására, módosítására azok későbbiekben történő visszatekinthetőségének céljából.

**Statisztikák:** A szoftver, néhány hasznos információt kínál a felhasználónak a befektetéseivel kapcsolatosan, mint például a különböző befektetések átlaga.

**PDF generálás:** A rendszerben rögzített befektetések adatainak PDF fájlba történő exportálására is lehetőséget nyújt.

**Valuta átváltó:** A kész rendszer lehetőséget kínál két eltérő kriptovaluta mennyiségének egyikből, a másikba történő átváltására.

**Jelszó módosítás:** A bejelentkezett felhasználó bejelentkezési jelszavának cseréjére, megváltoztatására.

### Szoftverspecifikáció

Funkcionális elvárások:

A szoftver működésének célja a kriptovaluták aktuális árfolyamainak nyomon követhetősége, a felhasználók befektetéseinek biztonságos környezetben történő tárolása, azok statisztikáinak figyelemmel kísérése, a befektetések PDF fájlba történő exportálhatósága és az árfolyamok közti átváltás lehetősége.

A felhasználó a felhasználónevének, e-mail címének és jelszavának az elvártak szerinti helyes megadása után regisztrálhat az alkalmazásba.

A regisztrációt követően akkor léphet az alkalmazás kezdőképernyőjére, ha a beviteli mezőket megfelelően kitöltötte, megadta a regisztráció során rögzített adatokat és már létezik regisztrált felhasználói fiók a megadott adatokkal.

A kriptovaluták árfolyamai között szabadon válthat a felhasználó és az éppen aktuális árfolyamgrafikon kerül megjelenítésre.

Tárolhatóak a szoftverben a felhasználóhoz tartozó befektetések. Egy befektetés megadása során megadandó a kriptopénz neve, amelybe befektetés történt, a kriptovaluta ticker azonosítója, a befektetett pénzösszeg, a vásárlás árfolyama és dátuma. Az adatok hiánytalan megadása esetén rögzíthető a befektetés.

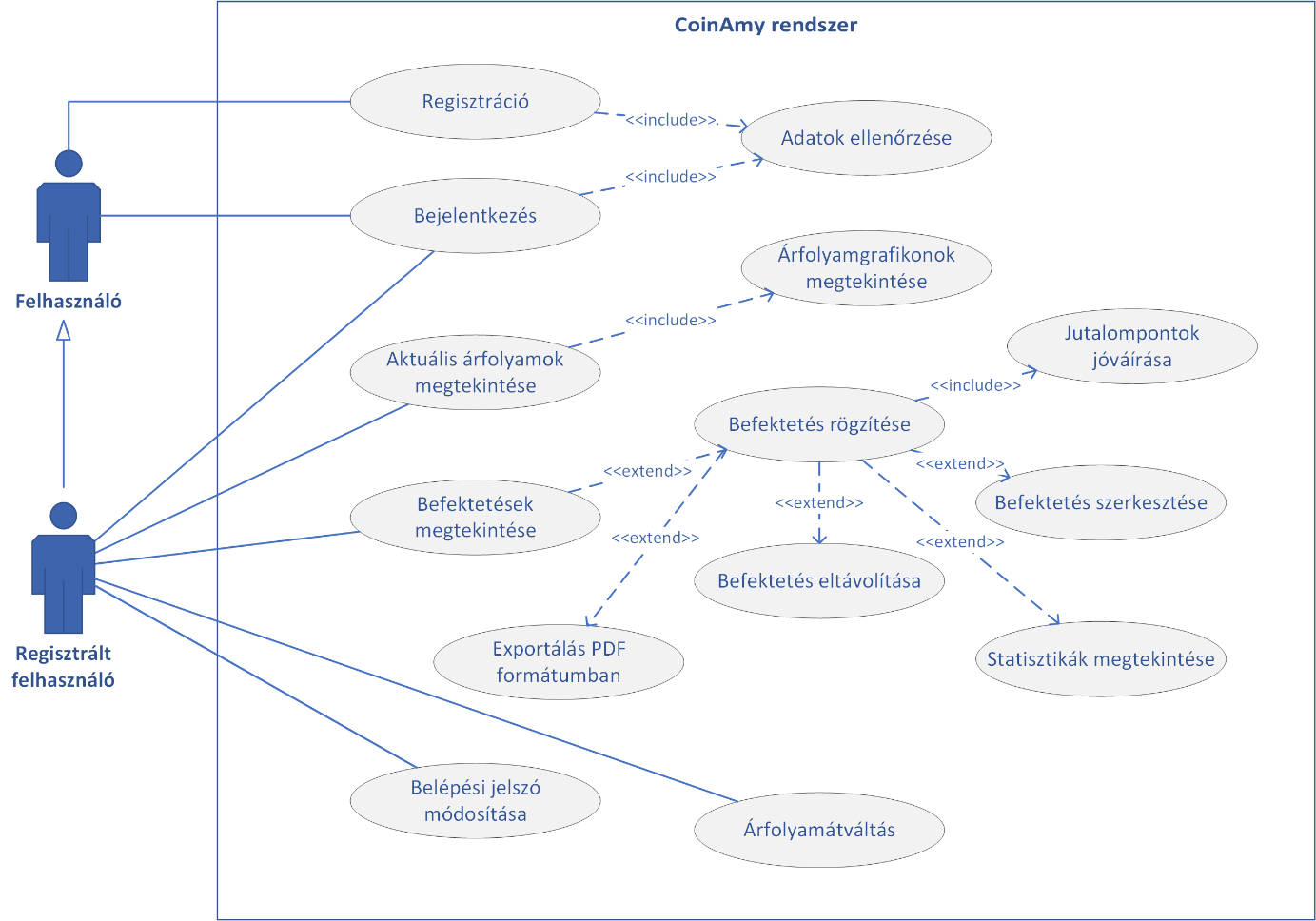
Amennyiben a felhasználó már rendelkezik rögzített befektetéssel, törölheti is azt, az adott befektetés kiválasztását követően. Ezenfelül, ha egy értéket módosítani szeretne, arra is nyílik lehetősége, szintén az adott befektetés kiválasztását majd a módosítani kívánt érték megváltoztatását követően.

Az árfolyamok közti átváltásra is kínál lehetőséget az alkalmazás a váltandó és a kapni kívánt valuta kiválasztása és a váltandó valutából birtokolt mennyiség megadása után.

Továbbá a felhasználó megváltoztathatja a felhasználói fiókjához tartozó belépési jelszavát az aktuális jelszó és az új jelszó kétszeri megadását követően.

Technikai elvárások:

* Asztali alkalmazásként, Windows operációsrendszer alatt történő felhasználhatóság.
* Mögöttes adatbázis alkalmazása.
* Adatok biztonságos környezetben való tárolása.

A szoftver elindítása után, a még nem regisztrált felhasználónak a bejelentkezés és a regisztráció funkciójának elérése van lehetősége. A regisztrációt követően a felhasználó az alkalmazásban implementált összes funkció elérésére jogosulttá válik. Megtekintheti az aktuális árfolyamokat, mellyel egyidőben a választott valuta árfolyamgrafikonja is betöltésre kerül. Lehetősége nyílik a saját befektetésinek az alkalmazásban történő rögzítésére, amelyeket ezután szabadon megtekinthet, módosíthat és el is távolíthat. A befektetés rögzítése esetén, az azért járó jutalompont mennyisége automatikusan kiszámításra kerül és hozzáadása megtörténik a felhasználó aktuális egyenlegéhez. Amennyiben a felhasználó rendelkezik rögzített befektetéssel, alkalma nyílik rá, hogy ezeket az adatokat PDF fájlformátumban az alkalmazáson kívül is megtekinthesse az exportálás funkció segítségével és az azokra vonatkozó statisztikákat is megnézhesse. Ezenfelül az éppen aktuális árfolyamokon az átváltó funkció használatával azt is megtudhatja, hogy egy bizonyos összegű kriptovalutáért mekkora mennyiséget kaphat egy másik kriptopénzből. Mindemellett módosíthatja a felhasználói fiókjához tartozó, bejelentkezéshez szükséges jelszavát.

1. ábra: Szoftver használati eset diagramja

### Alkalmazás megtervezése

Minek okán már léteznek hasonló célra szánt szoftverek a piacon, ezért a fejlesztés megkezdése előtt már rendelkeztem egy tervvel, azzal kapcsolatban, hogy miként szeretném az alkalmazást megvalósítani és milyen funkciókkal szeretném azt felruházni. Míg a piacon jelenleg is fellelhető megoldások szinte egytől-egyig csak webes környezeteben vagy mobilalkalmazás formájában érhetők el, fontos szempont volt számomra egy asztali alkalmazásként futtatható alternatíva megvalósítása.

### Látványtervek és arculati elemek készítése

A fejlesztés megkezdése előtt vizualizálni kezdtem a terveimet, mert ezáltal még inkább el tudtam képzelni és át tudtam gondolni, hogy a fejlesztési folyamat során előreláthatólag milyen fázisokra kell figyelmet fordítanom.

Ennek során készítettem néhány látványtervet, amelyek a szoftver felhasználóifelületének különböző ablakait és azoknak elrendezését ábrázolják. A kialakítás során szem előtt tartottam az ergonomikus, intuitív elrendezést és a könnyen átlátható felület megtervezését.

Ebben a fázisban kerültek elkészítésre a szoftver arculati elemei is, mint például a szoftver parancsikonja és a logó, amelyek igényes, kifinomult és egyedi megjelenést biztosítanak a használat során.

2. ábra: Felhasználói felület kezdetleges terve

### Adatbázisterv elkészítése

A program által használt adatok egy részét egy mögöttes, lokális adatbázis biztosítja, melyre a MySQL relációs adatbáziskezelőt választottam, melyben több szempont figyelembevétele is közrejátszott. Többek között jelentős tényező volt, hogy az említett adatbáziskezelő díjmentesen használható, emellett magas hatékonysággal képes lekérdezéseket futtatni. Emellett említésre méltó, hogy a szakmai tanulmányaim során leginkább ezzel az adatbáziskezelő rendszerrel foglalkoztam a legnagyobb részben, ennek okán volt elegendő időm a megismerésére.

A MySQL jelenleg az Oracle tulajdonában lévő relációs adatbáziskezelő, mely a világ több nagy méretű vállalatánál is, mint a Facebook és Google, jelenleg is aktív használatban áll.

A program adataink eltárolására egy kettő relációból álló adatbázist készítettem, melynek előzetes tervezése során fontos szerepet játszott a logikus felépítés és az adatredundancia elkerülésére történő törekvés.

Az adatbázis tervezésének kezdetén meghatároztam, hogy szükségem lesz egy táblára, amely eltárolja a felhasználókat és a hozzájuk tartozó különféle adatokat. Továbbá egy másik reláció is létrehozandó, mely az adott felhasználóhoz tartozó befektetéseik adatait tárolja.

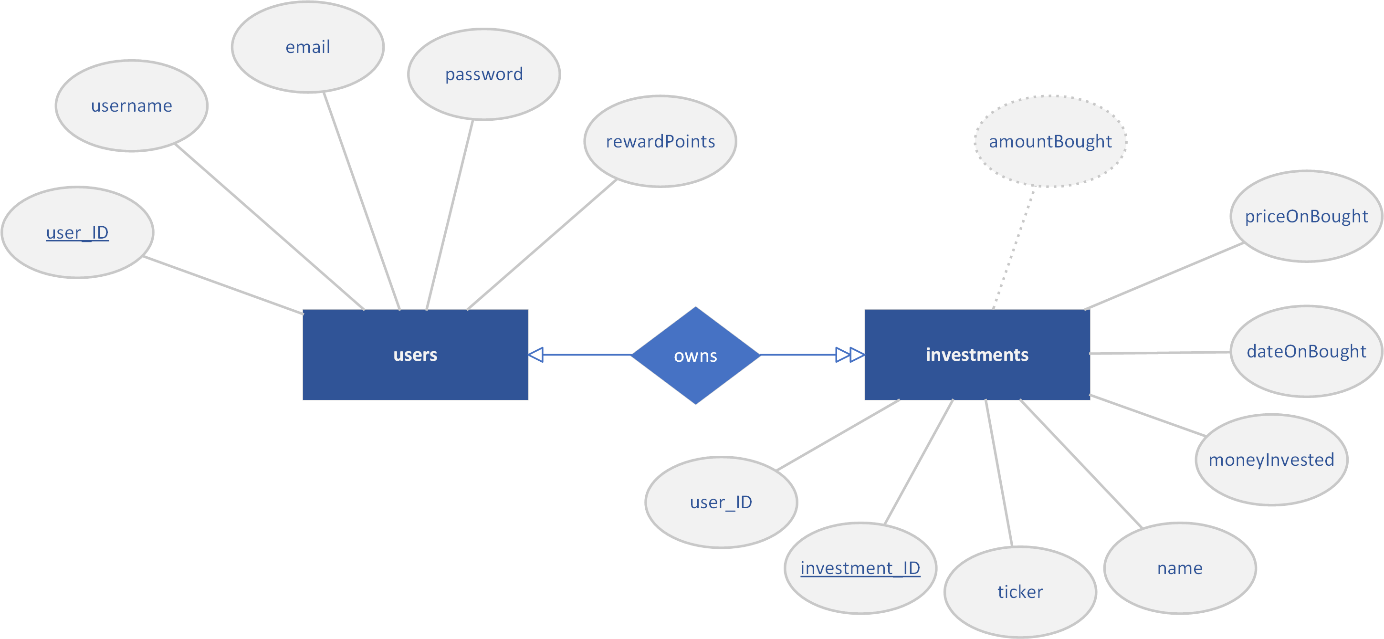
Ezt követően meghatároztam a mezők nevét, amiket az adott táblában szerepeltetni szeretnék, majd ezután kiválasztottam az elsődleges kulcsként használandó mezőket, amelyeket automatikus számlálóként definiáltam.

### Adatbázis felépítése

A CoinAmy adatbázis relációi az alábbi mezőket tartalmazzák:

A users tábla magában foglalja a felhasználóhoz tartozó felhasználónevet, e-mail címet, jelszót és a felhasználóhoz tartozó jutalompontok összegét. A relációban szereplő username és e-mail mezőket UNIQUE tulajdonsággal ruháztam fel, mivel ezekből csak egy érték szerepelhet az adott mezőben.

Az investments tábla tartalmazza az adott kriptovaluta nevéhez tartozó ticker[[2]](#footnote-2)-t, a kriptovaluta nevét, a befektetett FIAT pénzösszeg mennyiségét, a vásárlás árfolyamát és dátumát.

A két reláció között egy a többhöz hozzárendelés (1:N) áll fenn.

3. ábra: Adatbázis egyed-kapcsolat diagramja

Minek okán a C# beépítetten az MS SQL használatát támogatja, a MySQL-el történő adatbáziskapcsolat kialakításához szükség volt a mysql.data.dll 6.7.10 verziójának Visual Studio projektbe történő importálására.

Az adatbázis számára nem állítottam be egyéni jelszóval történő védelmet, azonban ez a későbbiek során, éles környezetben történő felhasználás mellett megfontolandó döntés lehet.

### A szoftver osztálydiagramja

4. ábra: A szoftver osztálydiagramja

### API kiválasztása és konfigurálása

Az alkalmazás bizonyos információkat, mint például a kriptovaluták aktuális árfolyamait és egyéb adatait egy API felhasználásával szerzi be. A fejlesztés megkezdése előtt meghatároztam, hogy melyik API felhasználása lenne a legoptimálisabb választás a jelenre és az esetleges későbbi továbbfejlesztést tekintve.

Különféle cégek által készített termékek elérhetőek az interneten, melyek közül számos költségterítéses, míg néhány bizonyos mértékben díjmentesen is felhasználható. A választás során fontos szerepet játszott, hogy az API felhasználásának jogáért ne kelljen díjat fizetni. Mindezen kondíciók meghatározása után a CoinGecko és a CoinStats által biztosított lehetőségekre szűkült a kör. Mindkét API esetében elmondható, hogy bizonyos korlátozásokkal biztosítanak lehetőséget díjmentes felhasználásra és nem igényelnek autentikációt, tehát API kulccsal történő azonosítást a szerver felé.

Választásom végül a CoinGecko API-ra esett, mivel számos fórumon ezt ajánlották a leginkább, illetve személyes véleményem szerint is ez felelt meg a leginkább az elvárásaimnak. Ez az API számos információhoz nyújt hozzáférést, ennek ellenére a fejlesztés során csak a számomra szükséges adatokat dolgoztam fel.

Fontos megjegyezni, hogy az API által a díjmentesen használható csomagban biztosított adatok csak korlátozott mértékben hozzáférhetőek, hiszen a hivatalos dokumentáció szerint 10 és 30 közötti hívás engedélyezett percenként.

Az API konfigurálása nem okozott különösebb nehézséget, hiszen az elérését biztosító URL címben található paraméterek módosításával állítható be az igényeknek megfelelően.

### Felhasználóifelület tervezése WPF-ben

Az alkalmazás felhasználói felületének tervezésese megkezdése előtt meghatároztam, hogy milyen ablakokra lesz szükségem, illetve azok milyen funkcionalitással bírnak majd a későbbiekben. Tervezés során végigondoltam, hogy mely ablakok esetén nincs szükség azok átméretezhetőségét biztosítani, ilyen például a bejelentkezési és a regisztrációs ablak.

A felület elrendezésének kialakítása során fontos szerepet játszott a könnyen áttekinthető, intuitív, felhasználóbarát elrendezés megalkotása.

A könnyen átlátható felülethez nagymértékben hozzájárul a hasonló funkcióval rendelkező vezérlők egységes megjelenésének kialakítása, ezért a gombok számára azonos megjelenést biztosítottam. Ezenfelül odafigyeltem, hogy a program működésében szerepet játszó vezérlők, a szoftver háttérszínétől jól elkülönülő színt kapjanak, ezáltal egyértelművé téve a használatot.

Az oldalsó felhasználói menü oszlop kialakításánál figyelembe vettem, hogy az azonos célt szolgáló menüpontok jól rendezett formában, egymáshoz közel legyenek megtalálhatóak, melyhez a XAML leíró nyelv által biztosított StackPanel felhasználásával, vertikálisan középre igazított elrendezést választottam.

Mindemellett a betűtípus kiválasztásánál törekedtem a könnyen olvasható, kellően nagy méretű betűstílus alkalmazására.

A program által biztosított hibaüzenetek, visszajelzések megjelenésére is nagy hangsúlyt fektettem, hiszen fontos szempont, hogy a felhasználó számára minél könnyebben érthető legyen a hibát kiváltó ok, ezért igyekeztem ezeknek a megjelenésükkel is nyomatékot adni.

A szoftver ablakainak reszponzívvá tételének jegyében a felhasználói felületek létrehozásánál Grid-eket is alkalmaztam.

Az XAML kód könnyebb átláthatóságának érdekében a különböző vezérlők stílusbeállításait az App.xaml fájlban igyekeztem elvégezni.

### Regisztrációs felület

Az alkalmazás funkcióinak eléréséhez minden esetben felhasználói regisztráció szükséges. A felhasználói felületen szereplő beviteli mezők és vezérlők a felhasználó által megadni kívánt felhasználói adatok rögzítésére szolgálnak. Ezután a megadott értékek az adatbázisban kerülnek tárolásra.

Mindezek előtt azonban fontos, hogy ellenőrizzük a felhasználó által rögzíteni kívánt adatok helyességét, tehát validáljuk azokat.

A beveti mezők tartalmának a következő kritériumoknak kell megfelelnie:

* A mezők értéke nem lehet üres.
* A felhasználónév hosszának el kell érnie a legalább 3 karaktert.
* Az e-mail címnek tartalmazni kell az arra jellemző karaktereket, tehát egy ’@’-t, illetve ’.’-t.
* A jelszónak legalább 8 karaktert, kis és nagy betűt, számot, valamint speciális karaktert kell tartalmaznia.
* A jelszó és a jelszó megerősítésére szolgáló mezőnek karakterre pontosan egyeznie kell.

A felhasználó által bevitt adatok a Regisztráció gombra történő kattintás esetén összehasonlításra kerülnek a UserDataValidator osztályban meghatározott feltételekkel és amennyiben nem tesznek eleget, a hibaüzenet megjelenítésre kerül az erre a célra szolgáló TextBlock-ban.

Az e-mail cím és jelszó helyességének ellenőrzésére reguláris kifejezést (Regex) használtam, mert hatékonyságával, minimális erőforrás felhasználása mellett teszi lehetővé a validációt.

Ezenfelül implementáltam a RegistrationViewModel osztályt, amely megvalósítja az INotifyPropertyChanged interfészt. Ez osztály az interfész segítségével a beviteli mezők minden változását folyamatosan figyelemmel kíséri. Továbbá elkészítettem a UsernameValidator és az EmailValidator osztályokat, melyek a ValidationRule beépített osztályból öröklődnek és feladatuk, hogy folyamatosan visszajelezzenek a felhasználónak, amíg a bevitt értékek meg nem felelnek az elvártaknak. Az adatok feldolgozása közben a TextBox-ban megjelenő TextBlock-on keresztül kerülnek megjelenítésre az információk, emellett a TextBox kerete mindaddig narancssárga színű, míg a mező nem veszi fel a várt értéket.

A UsernameValidator és EmailValidator osztályok az XAML kódban, a Text Property-n keresztül kerültek összeköttetésre a TextBox vezérlőkkel. A jelszó beviteli mezők esetében, biztonsági okokból nem alkalmazható ’Binding’, ezért azok validációjára, ebből az okból kifolyólag nem alkalmaztam ValidationRule osztályt.

A regisztrációs feltételek maradéktalan teljesülése esetén azt is megvizsgálom, hogy az adatbázisban létezik-e már regisztráció a megadni kívánt felhasználónévvel vagy e-mail címmel. Amennyiben az IsUsernameExist() és IsEmailExist() függvények ’true’ értékkel térnek vissza, a regisztráció megtörténik és a szoftver átirányít a bejelentkezési felületre. Ennek oka, hogy a tervezési szakasz folyamán biztonságosabb megoldásnak láttam a frissen regisztrált felhasználó adatait a bejelentkezés során ismételten bekérni a szoftver tényleges használattának megkezdése előtt. Ezzel a döntéssel ellenőrizhető, hogy a felhasználó megjegyezte-e a bejelentkezési adatait.

### Jelszó titkosítás

A felhasználó által regisztrációkor megadott jelszó titkosítására az SHA256 titkosítási algoritmust választottam, mivel ezt a fajta titkosítást gyakorlatilag lehetetlen visszafejteni.

A felhasználó által bevitt jelszó a Hash osztályban található, HashPassword statikus függvény felhasználásával kerül titkosításra. A későbbi felhasználói bejelentkezések során a felhasználó által megadott jelszó ismételten titkosításra kerül és a titkosított karakterlánc kerül összehasonlításra az adatbázisban szereplő értékkel.

### Bejelentkezési felület

Ebben az ablakban szintén rendelkezésre állnak a felhasználói interakcióhoz szükséges beviteli mezők és vezérlők.

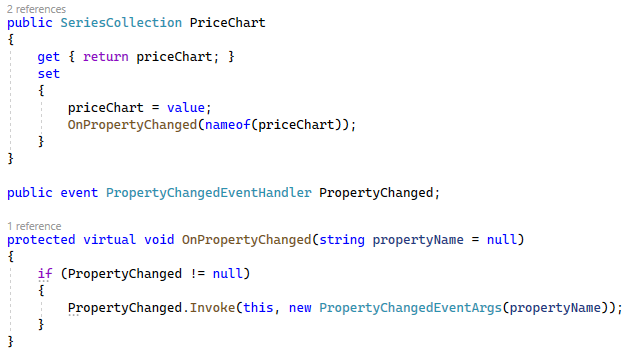
Abban az esetben, ha a felhasználó a bejelentkezés gombra kattint, a UserDataValidator osztály segítségével ellenőrzésre kerül, hogy a beviteli mezők megfelelően vannak-e kitöltve. Amennyiben igen, a mezők értéke alapján lefut egy MySQL lekérdezés, amely ellenőrzi, hogy az adatbázisban szerepel-e regisztráció a megadott adatokkal. Abban az esetben, ha létezik a megadott felhasználónév / email és jelszó kombináció, a szoftver továbblépteti a felhasználót a kezdőképernyőre. Ezzel ellenkező esetben hibaüzenetben értesíti a felhasználót a bevitt adatok helytelenségéről.

### Kezdőképernyő felülete

A sikeres bejelentkezés után megjelenik a kezdőképernyő felülete, ahol a felhasználó tájékozódhat a piaci kapitalizáció szerint csökkentő sorrendben álló ötven kriptovaluta aktuális adatairól. Minek okán a kapitalizáció folyamatosan változik, a listában megjelenő kriptovaluták sorrendje ennek függvényében cserélődhet.

Az információk egy ListView vezérlőben kerülnek megjelenítésre, melynek előnye a ListBox-al szemben, hogy támogatja a többoszlopos elrendezést, ezáltal egy átláthatóbb és esztétikus megjelenést biztosít, emellett lehetőséget nyújt a képek megjelenítésére is.

A ListView-ban megjelenő adatok a WPF-en belül gyakran használt DataBinding-al kerültek megvalósításra.

A ListView SelectedIndex tulajdonságának függvényében kerülnek megjelenítésre a grafikonon az adott kriptovalutához tartozó árfolyamadatok. A coinDatas lista tartalmazza az API-től beérkező kriptopénzek összes adatát. Mivel ListView adatai is ebből a kollekcióból származnak, ezért annak kiválasztott index száma megegyezik a listáéval, ebből kifolyólag a választott indexen szereplő adatok kerülnek lekérésre. Az adatokat biztosító API URL címébe string interpoláció alkalmazásával, paraméterként kerül beszúrásra a kiválasztott indexhez tartozó kriptovaluta azonosítója (id).

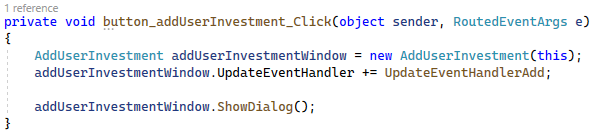
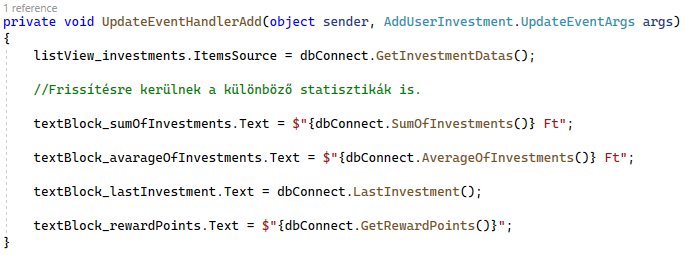
A grafikon felhasználói interakció után történő azonnali frissítésére az INotifyPropertyChanged interfészt alkalmaztam, amely lehetővé teszi a felhasználói felület (UI) számára, hogy értesítést kapjon, amint a háttérben a mező vagy kollekció tulajdonsága (property) megváltozik.

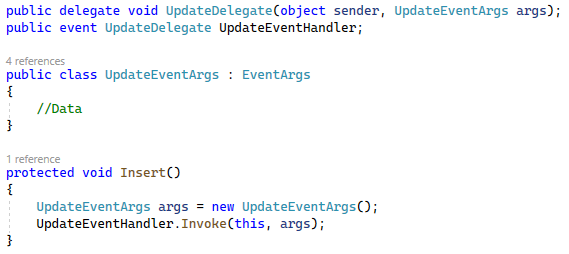
### Portfolió kezelő felület

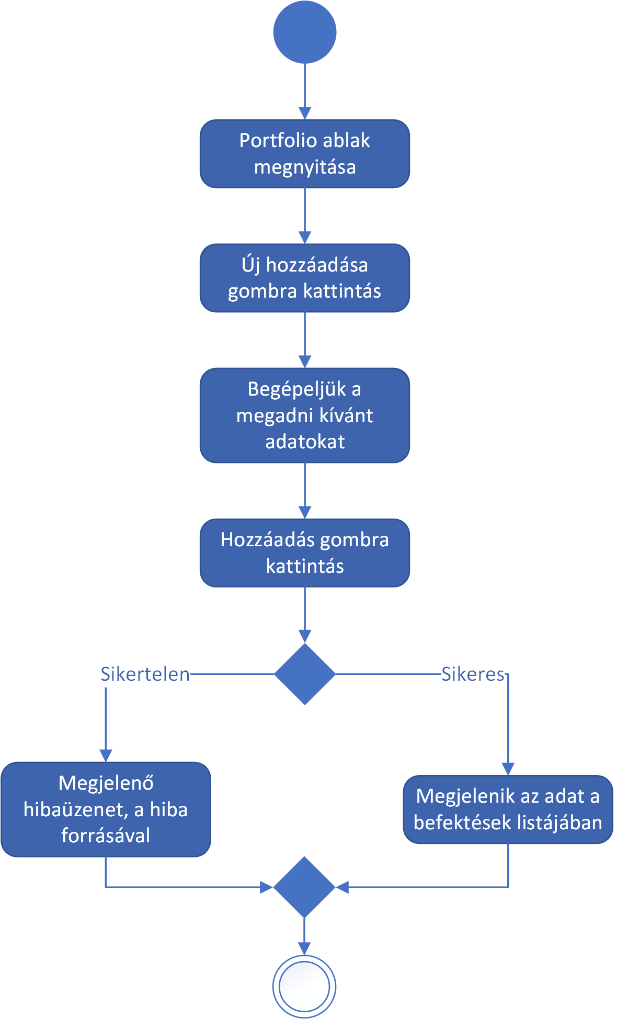
A felhasználói befektetések megjelenítésére, az előzőhez hasonlóan egy ListView megjelenítőt alkalmaztam.

Ebben az ablakban, a felhasználó által történő adatrögzítésre és adatmódosításra a választott funkció gombjára történő kattintást követően, új ablakban nyílik lehetőség. Az adatok megadását követően azok rögtön a mögöttes adatbázisban kerülnek rögzítésre.

A fejlesztés során azt is kivitelezni szerettem volna, hogy az adatbázisban történő rögzüléssel egyidőben a felhasználói felületen is jelenjenek meg a frissen megadott befektetési információk és az ezekhez tartozó statisztikák. Ennek megvalósítására eseményt (event) implementáltam.

Az Új hozzáadása gomra történő kattintáskor példányosításra kerül az adatok rögzítésére lehetőséget biztosító ablak, majd az UpdateEventHandler() metódus által hozzáfűzésre kerülnek a már az adatbázisban szereplő adatok.

A megnyíló ablakban, a Hozzáadás gombra történő kattintást követően meghívásra kerül az Insert() metódus, melynek következtében az UpdateEventHandler() átadja az információkat az UpdateEventHandlerAdd() metódusnak frissíti a befektetések adatait tartalmazó ListView-t és a további statisztikai adatokat az adatbázisban rögzítésre került információk alapján.

A választott befektetés módosítása után a ListView frissítését ehhez hasonlóan, event felhasználásával valósítottam meg.

5. ábra: Befektetés hozzáadásnak aktivitás diagramja

Jutalompont összegének kiszámítása

A felhasználó számára adott jutalompontok mennyisége a befektetett pénzösszegtől függően kerül kiszámításra. Az alkalmazás minden esetben egy-egy kriptovaluta vásárlás után növeli az adott felhasználó pontegyenlegét a következőek szerint:

* Ha a befektetés összege kisebb, mint 300.000 Ft, a jutalom mértéke 0,5%
* Amennyiben nagyobb, mint 300.000 Ft, de kisebb, mint 500.000 Ft, a jutalom 0,8%
* 500.000 Ft-nél magasabb összeg befektetése esetén a jutalom mértéke eléri az 1%-ot.

### Árfolyam átváltó felület

A funkció megvalósításához szintén az API által biztosított adatok kerültek felhasználásra, ezáltal mindig aktuális árfolyamokat használja a szoftver.

Az API-ból származó adatok listájából az első ComboBox választott indexén lévő kriptovaluta árfolyamának szorzata osztásra kerül a második ComboBox-ból választott, kapni kívánt valuta aktuális árfolyamával.

Váltandó mennyiség \* váltandó valuta aktuális árfolyama / várt valuta aktuális árfolyama

A két legördülő listába a valuták neve mellett a hozzájuk tartozó logok is megjelenítésre kerülnek, amelyekhez StackPanel-t használtam fel.

### Jelszó módosítás

A felhasználói jelszó módosításához egy SQL lekérést írtam, amely megvizsgálja, hogy létezik-e olyan felhasználói fiók, ahol a bejelentkezett felhasználó azonosítója (user\_ID) és jelszó módosítás során megadott régi jelszó megegyezik. Abban az esetben, ha létezik ilyen, az újként megadni kívánt jelszó beállításra kerül.

A jelszó módosítása során a következő szempontok kerülnek figyelembevételre:

* A felhasználó helyesen adta meg a jelenlegi jelszavát.
* Az új jelszó megfelel-e az elvárt kritériumoknak (Az új jelszó tartalmaz legalább 8 karaktert, kis és nagy betűt, számot, valamint speciális karaktert.)
* Az újként beállítani kívánt jelszó nem egyezhet meg a régi jelszóval.
* Megerősítésre került-e az újként beállítani kívánt jelszó az erre szolgáló mezőben.

### API hívás

Az alkalmazás a különböző kriptovaluták adatait egy API felhasználásával szerzi be. Ennek megvalósítását az objektumorientált programozás konvenciói szerint különálló osztályban implementáltam.

Az API által szolgáltatott adatok beszerzésére egy Task típusú függvényt hoztam létre, amely lehetővé teszi az aszinkron műveletek kezelését. Aszinkron műveletek alkalmazására azért van szükség, mivel a szerver irányába történő kérés indításának pillanatában nem tudható, hogy a szerver milyen válaszidő mellett tér vissza a válasszal. Aszinkron műveletek felhasználásával a program futása több szálon folyik, ezáltal elkerülhető, hogy a program futása látszólagosan szüneteljen.

A CoinAmy, a CoinGecko API által biztosított kétféle végpontot használ:

* /coins/markets

Ez a végpont a paraméterben megadott mennyiségű kriptovalutával és azok adataival, JSON formátumú válasszal tér vissza. Ezenfelül van lehetőség további paraméterek megadásával testre szabni.

* /coins/{id}/market\_chart

A paraméterül adott kriptovaluta API ’id’ szerint adja vissza a kért kriptovaluta árfolyamát és az árfolyamhoz tartozó időpontot UNIX időformátumban. A válasz, az előzőhez hasonlóan JSON formátumban érkezik.

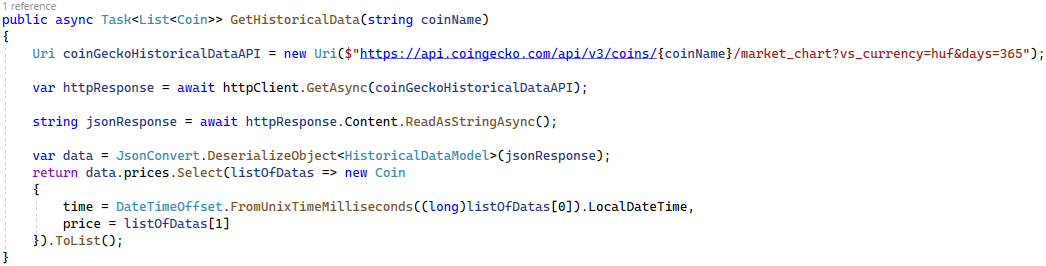
Az API által szolgáltatott JSON visszatérésű objektumok C# osztállyá történő átalakítására a <https://json2csharp.com/> weboldalt használtam, bár újabban már a Visual Studio is biztosít lehetőséget a vágólapon lévő JSON-t, C# osztályként a projektbe illeszteni.

Az API díjmentes verziója nem igényel API kulcs által történő hitelesítést, emellett ebben az ingyenes csomagban a hívások száma percenként 10 és 30 közötti számban van korlátozva.

A CoinGecko API fejlesztői dokumentációja a következő hivatkozáson keresztül érhető el: <https://apiguide.coingecko.com/getting-started/introduction>

Az API hívását egy, a szerver felé irányúló aszinkron GET kéréssel indítottam el. Amennyiben az erre küldött válasz sikeres, 200-as státuszkóddal érkezik vissza, a kapott JSON formátumú szöveg a jsonResponse változóban kerül tárolásra.

Ezt követően, a JSON fájl feldolgozását a Newtonsoft.JSON NuGet Package segítségével valósítottam meg a CryptocurrencyDataModel osztály szerint, majd az adatokat a coinDatas listában tároltam el.

Amennyiben mégis hibát jelző státuszkód érkezne a szervertől, megjelenítésre kerül a hibát kiváltó HTTP státuszkód és a hibát kiváltó ok.

A választott kriptovaluta adatait az elsőzőhöz hasonlóan GET kéréssel szereztem be. Mivel az API által biztosított adat két tulajdonságot, az árfolyamot és a hozzá tartozó, éppen aktuális időpontot tartalmazza UNIX időformátumban, ezért létrehoztam a Coin osztályt, amely ezzel a két tulajdonsággal rendelkezik. Az érkező adatok egy listává alakítva kerülnek visszadására.

### PDF generálás

A felhasználónak lehetősége kínálkozik a befektetések PDF fájlba történő generálására is a szoftver segítségével. Ennek megvalósítására elkészítettem a PDF\_generate osztályt, amely az iTextSharp NuGet Package segítségével teszi lehetővé a funkció megvalósítását.

Az exportált PDF tartalmát a dbConnect osztályban szereplő LoadDataToExport() függvény adja, amely az iTextSharp ’PdfPTable’ speciális típusával tér vissza, amely kifejezetten egy táblázatot ad vissza PDF generálását végrehajtó osztályban szereplő példánynak.

A generálandó fájl mentésére szolgáló elérési útvonalat egy felugró SaveFileDialog ablak segítségével kerül átadásra az osztály konstruktorának.

## Felhasznált NuGet Package-ek

A szoftver rendelkezik olyan funkcionalitásokkal, amelyre a C# és Windows Presentation Foundation (WPF) alapértelmezetten nem, vagy csak korlátozott mértékben nyújt lehetőséget. Az alkalmazás képességeinek bővítése érdekében, ezek a funkciók .NET környezetben jól ismert NuGet Package-ek felhasználásával kerültek implementálásra.

Fejlesztés során felhasznált NuGet Package-ek:

### Newtonsoft.JSON (13.0.2)

A kriptovaluták adatait tartalmazó információk HTTP (HyperText Transfer Protocol) kérések segítségével, JSON (JavaScript Object Notation) formátumban érkeznek meg az API részéről. A Newtonsoft.JSON NuGet Package hívatott a JSON formátumban lévő adatokat lebontani (deszerializálni), azután egy választott adatszerkezetben eltárolni.

Habár manapság már a Microsoft is biztosít lehetőséget a System.Text.Json NuGet Package segítségével történő JSON formátumban tárolt adatok feldolgozására, a fejlesztés során mégis a Newtonsoft.JSON-t választottam annak szélesebb körű ismertsége és felhasználtsága okán.

### LiveCharts

A WPF keretrendszer alapvetően nem biztosít lehelőséget a különböző grafikonok és diagramok ábrázolására. Ennek kiküszöbölésére a LiveCharts NuGet Package felhasználását találtam ideálisnak, mivel véleményem szerint a mai napig ez a leginkább karbantartott és legtöbbek által felhasznált megoldás. Az API árfolyamadatainak és a felhasználó által biztosított adatok megjelenítésére szolgál.

### iTextSharp (5.5.13.3)

A szoftverből történő PDF fájl generálás megvalósításához láttam szükségességét a felhasználásának. Ennek a funkciónak az implementálásához szükség volt az iTextSharp felhasználására, amely nagymértékű testreszabhatóságával, díjmentes felhasználhatóságával, megbízhatóságával és elterjedtségével előnyt élvezett a választásom során.

### Tesztelési, hibafeltárási folyamat

Az elkészült szoftver tesztelésére manuális és automatizált tesztelési módszert választottam.

A regisztrációs, bejelentkezési és további adatbeviteli felületeken a legkülönfélébb típusú adatok kerültek megadásra, amellyel ellenőriztem, hogy eltérő adatok megadása esetében hogyan viselkedik a szoftver. Ezenfelül ellenőriztem, hogy hibamentes-e a bevitt adatok ellenőrzése.

Az egyik eset során kipróbálásra került, hogy hogyan reagál a szoftver az internetkapcsolat hiányára. A tesztelés során felmerült hibák javításra kerültek.

A szoftver manuális tesztelésének lefolytatására külső személy bevonását is alkalmaztam egy családtagom közreműködésével. A tesztelés a fejlesztési hardverkörnyezettől eltérő eszközön történt. A tesztelő meglátásai szerint, az alkalmazás megfelelően működik, a tesztelés során nem talált a stabil működést befolyásoló hibát.

### Elvégeztt Unit Test-ek

A fejlesztési folyamat végeztével komponens teszteket (Unit Test) is végeztem a szoftveren. A regisztrációs és bejelentkezési felületeket találtam a leginkább alkalmasnak az efféle tesztek futtatására. A tesztek eredményei egytől-egyig megfeleltek az előre elvártaknak.

A tesztek megírása során törekedtem a Unit Testelésre vonatkozó konvenciók alkalmazására, többek között a különböző tesztesetek elnevezésénél és az Arrange-Act-Assert hármas alkalmazásával. Emellett minden tesztelt függvény kipróbálásra került az elvártnak megfelelő és nem megfelelő értékek megadásával.

Főként a regisztráció során megadandó adatok validációját végző függvények működésének helyességét ellenőriztem komponens tesztek alkalmazásával.

# Továbbfejlesztési lehetőségek

A CoinAmy asztali alkalmazás számos hasznos funkcióval rendelkezik, amelyek a felhasználó segítségére lehetnek a mindennapos használat során, azonban létezik még néhány funkció, amely szintén helyet kaphatott volna az alkalmazásban, azonban a kellő idő hiányában nem került implementálásra. A következőkben erre látható néhány konkrét példa:

### Bitcoin-tárca (Bitcoin Wallet)

A kriptovaluták működése a blokklánc technológia működési elvén alapszik. A bányászok különböző bonyolult matematikai műveletek elvégzésével hitelesítenek egy-egy felhasználók között végbemenő kriptovaluta tranzakciót, utalást, melyekért cserébe jutalomban részesülnek. Ezek a tranzakciók a végbemenetelük után hozzáíródnak egy blokk végéhez és ezen blokkok összességét nevezzük blokkláncnak. A különböző kriptovaluta tranzakciók és bizonyos típusú kriptovaluták tárcáinak egyenlege bárki számára szabadon hozzáférhető az interneten. Egy tranzakció két felhasználó között megy végbe, amikor egyikük a másik fél publikus kriptovaluta tárca címére utal egy szabadon megválasztott mennyiséget.

Ennek a funkciónak a későbbi megvalósításához szintén egy NuGet Package, az NBitcoin felhasználására lesz szükség, mivel a Bitcoin, mint szoftver C++ nyelven íródott.

### Jutalompontok beváltása

A szoftver fejlesztése során már implementálásra került olyan funkció, amely a felhasználó kriptovaluta vásárlásai után egy matematikai képlet alapján adott mennyiségű pontjutalomban részesíti őt. Az alkalmazás továbbfejlesztésével beépíthető egy olyan funkció, amely ezeknek a pontoknak a beváltására biztosít lehetőséget különböző digitális jutalmakért cserébe, például különböző NFT termékekre.

### Aktuális kriptovaluta árfolyamok azonos időközönként történő frissítése

Az általam felhasznált, CoinGecko API dokumentációja szerint, percenként 10-30 API hívás engedélyezett az ingyenes felhasználhatóságú csomagban. Az alkalmazás későbbiekben történő fejlesztése során megvalósítható egy olyan funkció, amely mindenféle felhasználói interakció nélkül, bizonyos időközönként hívást végez az API felé és ezután frissíti a kriptovaluták különböző adatait tartalmazó listát.

# Felhasználói dokumentáció

## Szoftver célközönsége

Az alkalmazás minden olyan felhasználó számára hasznosnak bizonyul, akik rendszeres időközönként vagy egyszeri alkalommal a kriptovaluták piacán fektetik be a megtakarításaikat és egy helyen kívánják nyomon követni azokat. A szoftver hivatott a felhasználók naprakészen tartása aktuális árfolyamokkal, piaci adatokkal, emellett az egyén befektetéseihez tartozó adatok biztonságban történő tárolására szolgál.

## Rendszerkövetelmények:

A CoinAmy asztali alkalmazás futtatásához szükséges hardveres és szoftveres követelmények a következőkben olvashatók:

### Szoftveres követelmények

* Operációsrendszer: Windows 10 (64-bit) vagy ettől újabb kiadás
* .NET Framework 4.8
* MySQL lokális adatbázis futtatására alkalmas szoftvercsomag (Ajánlott: XAMPP)

### Hardveres követelmények

* Processzor: 1 GHz vagy ennél magasabb órajel
* Memória: 512 Megabájt
* Tárhely: 200 Megabájt szabad tárterület
* Megjelenítő: 1280 x 1024 pixel
* Aktív szélessávú internetkapcsolat

## Telepítési útmutató:

### CoinAmy telepítése

Az alkalmazás telepítése nem igényel különleges felhasználói ismereteket. A CoinAmy szoftver telepítése az általános asztali alkalmazásokkal megegyező módon történik.

A telepítésre szolgáló futtatható állomány (.exe) elindítását követően a telepítési varázsló navigálja végig a felhasználót a telepítési folyamaton.

### XAMPP szoftvercsomag telepítése

Ezen felül szükséges a XAMPP szoftvercsomag telepítése is, mely a szoftver hátterében futó adatbázis kiszolgálására hivatott. A szoftver telepítésére szolgáló fájl a következő hivatkozáson keresztül érhető el: <https://www.apachefriends.org/hu/index.html>

A XAMPP telepítése hasonlóan egyszerű módon történik, a telepítővarázsló utasításait követve telepíthető a szoftver.

A sikeres telepítést követően, a XAMPP futtatása szükséges, ahol az Apache és MySQL szerverek elindítása szükségeltetik, a mellettük található gombok segítségével.

A továbbiakban a számítógépen meglévő, szabadon választott internetböngésző megnyitása után, a phpMyAdmin felület indítása szükséges, amely a következő hivatkozáson keresztül érhető el: <http://localhost/phpmyadmin/index.php>

Ezt követően, az oldal tetején található navigációs sávban megjelenő Importálás lehetőség választása szükséges, ahol lehetőség nyílik a CoinAmy szoftverhez mellékelt SQL kiterjesztésű adatbázislétrehozó állomány importálására.

Mindezek után, a szoftverek telepítésével további teendő nincs.

Amennyiben nem kívánja minden rendszerindítást követően a XAMPP-ot és a hozzá tartozó kiszolgálókat manuálisan futtatni, beállítható annak operációsrendszerrel történő automatikus indítása a szoftver beállításaiban.

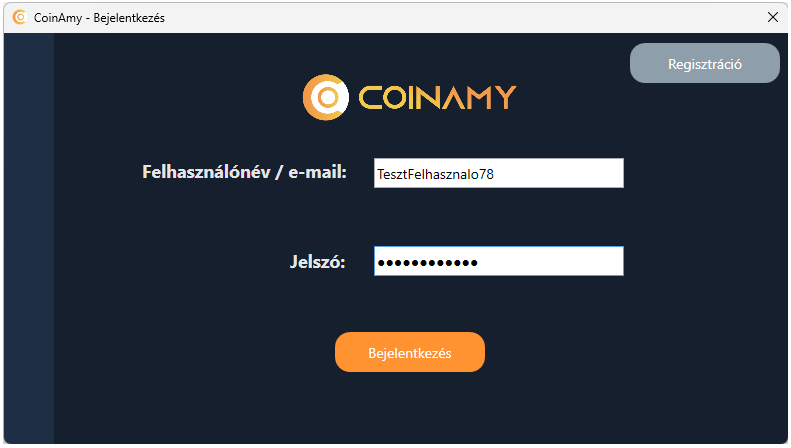
## Az alkalmazás használata:

### Bejelentkezés

A szoftver elindítása után a bejelentkezési felület jelenik meg, ahol a felhasználónak lehetősége nyílik a bejelentkezéshez szükséges felhasználónév / e-mail és jelszó kombinációjának megadására.

Amennyiben létezik a megadott adatokkal regisztráció, a felhasználó bejelentkezése sikeres, a szoftver átlép az Árfolyamok kezdőoldalra. Ellenkező esetben a hiba leírását tartalmazó hibaüzenet jelenik meg a képernyőn, amely javítását követően a bejelentkezés szintén sikeres.

Abban az esetben, ha a felhasználó még nem rendelkezik saját felhasználói fiókkal, az ablak jobb felső sarkában található szürke színű Regisztráció gombra történő kattintással átléphet a regisztrációs ablakba.



6. ábra: Bejelentkezési felület

### Felhasználói regisztráció

A szoftver funkciói csupán egy felhasználói regisztrációt követően érhetők el teljeskörűen. A megjelenő regisztrációs ablakban szükséges megadni a felhasználó által szabadon választott felhasználónevet, e-mail címet és jelszót.

Az adatok megadása során a következő szempontok szem előtt tartandók:

* A mezők nem maradhatnak kitöltetlenül.
* A felhasználónév hosszának el kell érnie a legalább 3 karaktert.
* Az e-mail címnek valódinak kell lennie.
* A jelszónak legalább 8 karaktert, kis és nagy betűt, számot, valamint speciális karaktert kell tartalmaznia.
* A jelszó és a jelszó megerősítésére szolgáló mezőnek karakterre pontosan egyeznie kell.

A regisztrációs adatok megadása során a felhasználónév és e-mail cím begépelése közben folyamatosan látható, hogy a begépelt adat megfelel-e az elvártaknak.

A program használatához a későbbiekben ezen adatok megadása szükséges, ezért érdemes az adatokat megjegyezni, különben a bejelentkezés a már meglévő felhasználói fiókba nem lehetséges.

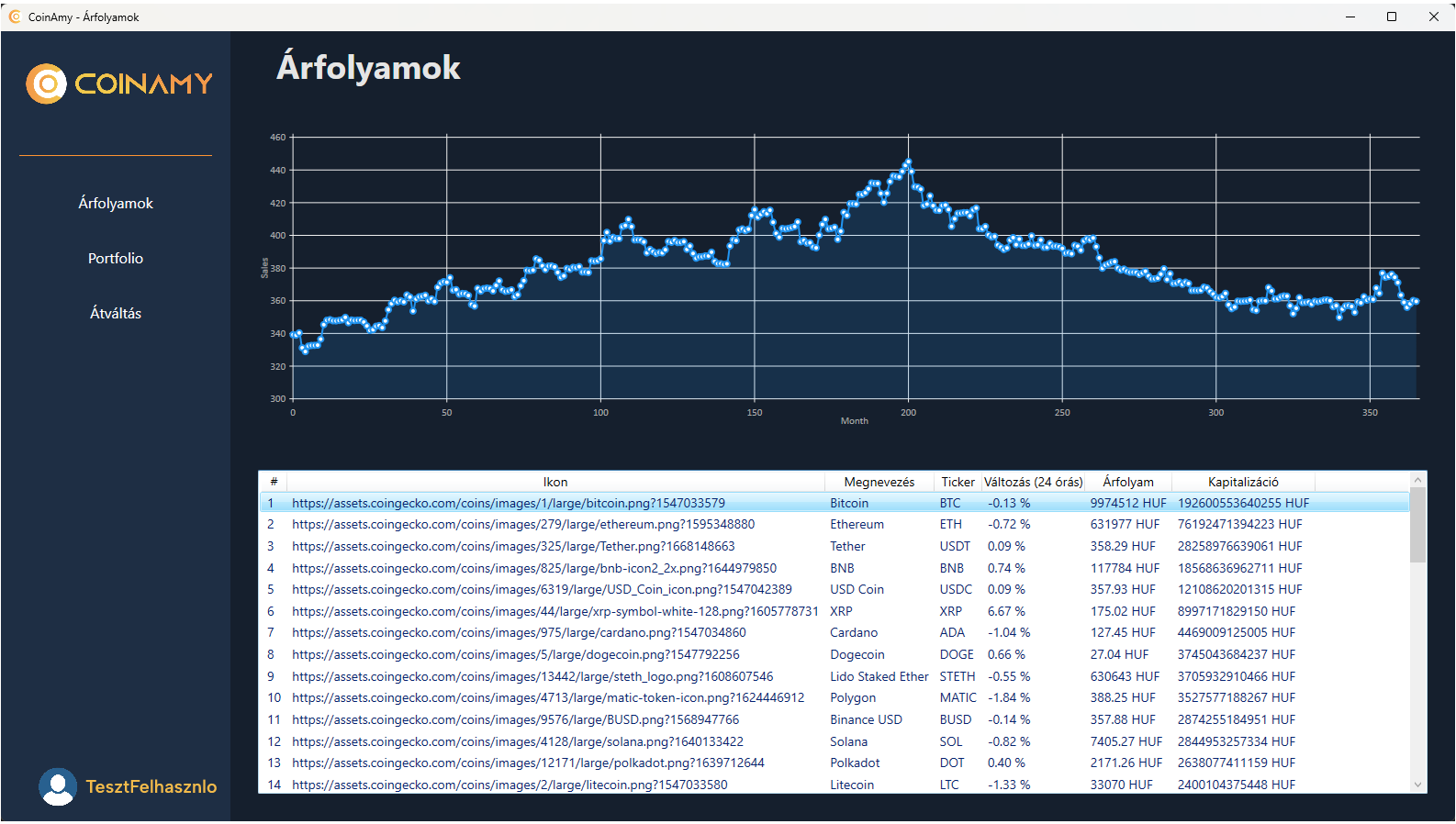
7. ábra: Regisztrációs felület

### Árfolyamok ablak (Kezdőképernyő)

A sikeres bejelentkezést követően megjelenő ablakban látható az 50 legnagyobb piaci kapitalizációval rendelkező kriptovaluta, kapitalizáció szerinti csökkenő sorrendbe rendezve. Az ablak felső részében található a választott kriptovaluta árfolyamát ábrázoló grafikon. Közvetlenül ezalatt található a kriptovaluták adatait részletesen tartalmazó lista.

A felhasználónak az említett listából van lehetősége kiválasztani az általa megnézni kívánt kriptovalutát, amelynek a grafikonját meg kívánja tekinteni. Amikor a listában szereplő kriptopénz sorára kattint, a grafikonon a választott kriptovaluta árfolyamadatai jelennek meg 365 napra visszamenőlegesen.

Amennyiben részletesebb információkra kíváncsi, a kurzort a grafikon fölé mozgatva, megtekintheti az adott ponthoz tartozó árfolyamadatot.

A szoftver további funkcióit a bal oldalt található menüsorból, a választott menüpontra történő kattintást érheti el.

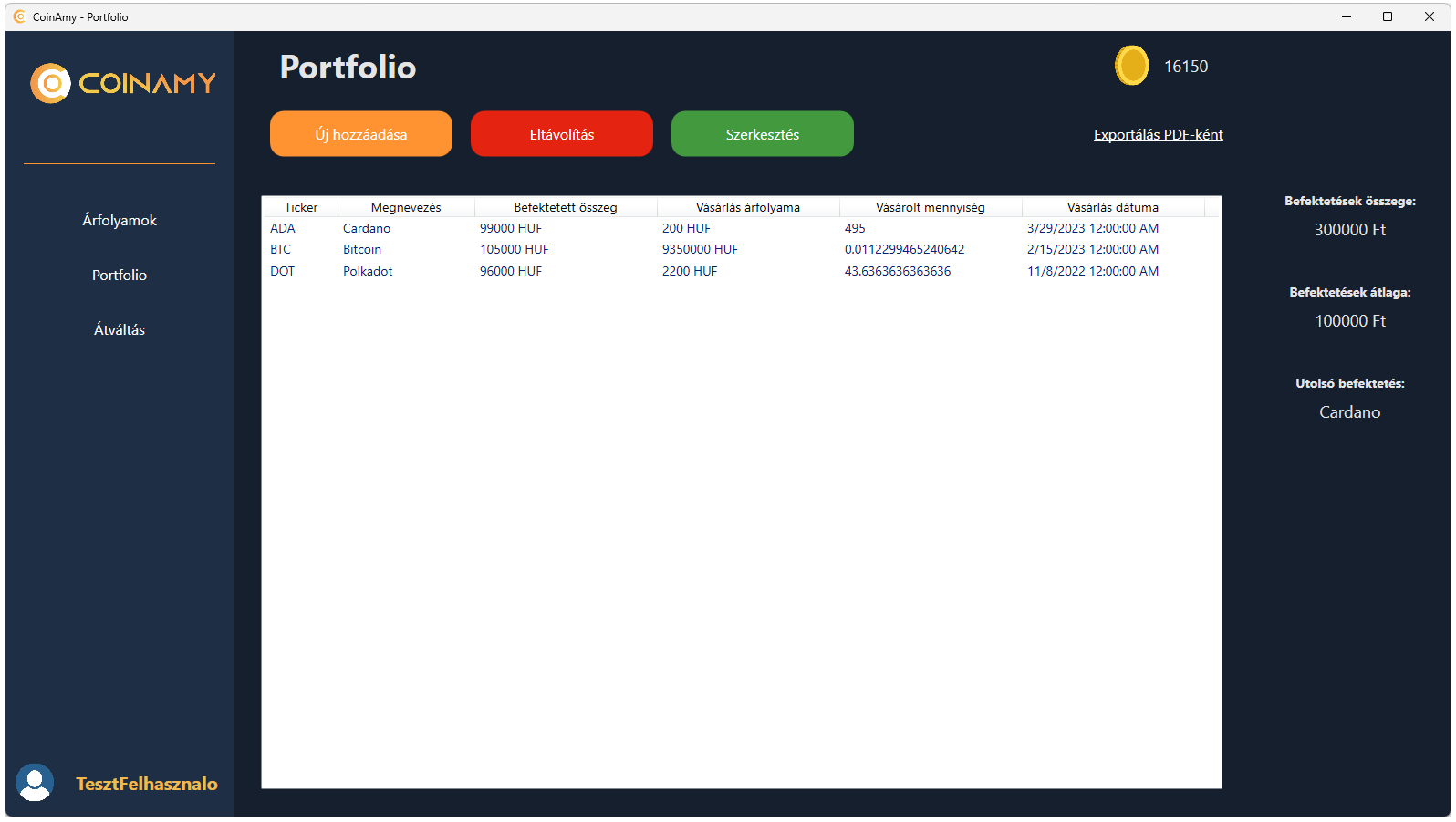
8. ábra: Kezdőképernyő felülete

### Portfolió kezelő felület

Az ablak megnyitását követően megjelennek a felhasználó által, már a korábbiakban rögzített befektetések. Amennyiben a felhasználó még nem rögzített semmilyen adatot, a befektetéseit tartalmazó lista üres.

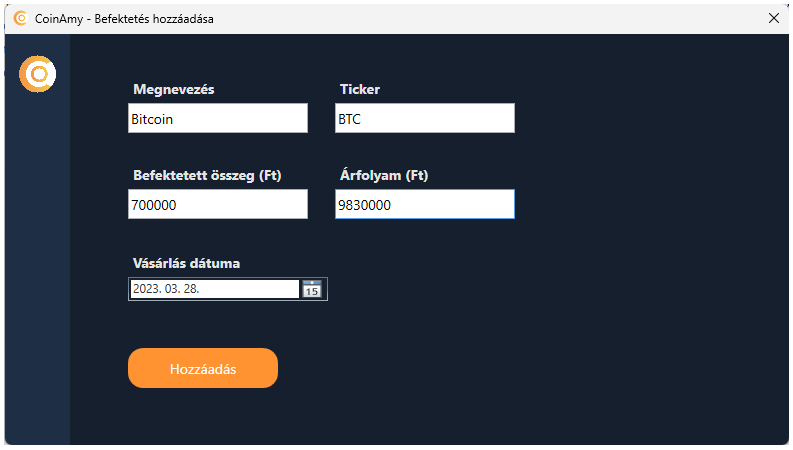
A már hozzáadott befektetések soronként jelennek meg az ablakban, különböző adattípusonként oszlopokra bontva.

A felhasználó láthatja, az általa vásárolt kriptovaluták nevét, azok rövidített ticker azonosítóját, a befektetés idejében aktuális árfolyam értéket, a befektetett pénzösszeg mennyiségét, a befektetés során a felhasználó birtokába jutott kriptopénz mennyiségét és a befektetés dátumát.

A felület jobb szélső oldalán találhatók meg a különböző befektetések összességére vonatkozó statisztikák, melyek segítségével hasznos információk birtokába juthat a felhasználó.

9. ábra: Portfolió kezelő felület

### Befektetés hozzáadása

Új befektetés rögzítésére, az ablak felső sorában megtalálható narancssárga Új hozzáadása gombra történő kattintás után nyílik lehetőség. A felugró ablakban a felhasználó rögzítheti az általa megadni kívánt befektetés adatait.

10. ábra: Befektetés hozzáadására szolgáló ablak

### Befektetés eltávolítása

Amennyiben a megadott befektetés tévesen került rögzítésre, a lehetőség biztosított annak törlésére. Ennek megvalósításához elengedhetetlen kiválasztani a befektetéseket tartalmazó listából az eltávolítandó adatot, majd a felső sorban megtalálható piros Eltávolítás gombra kattintani. Ezt követően a felugró ablakban a kiválasztott befektetés törlési szándékának megerősítése szükséges. Abban az esetben, ha mégsem kíván törölni, kattintson a Nem gombra.

### Befektetése szerkesztése

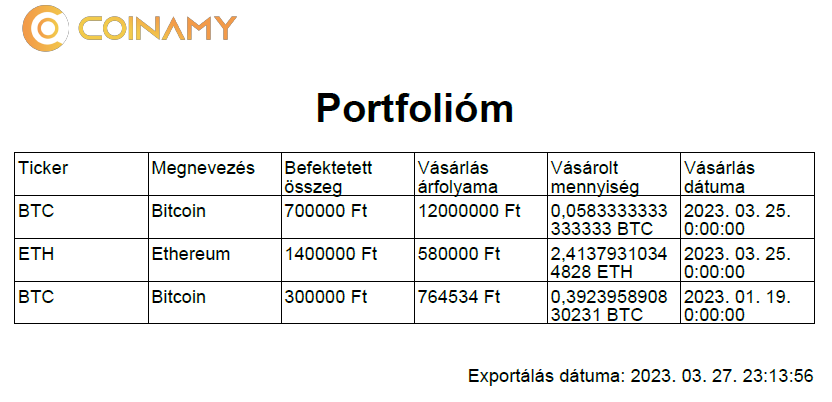
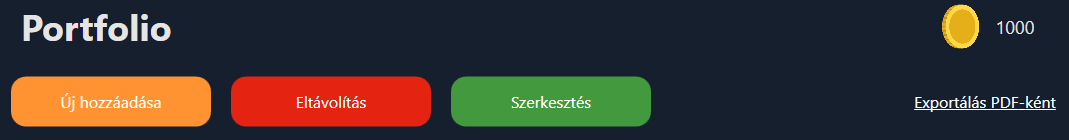
Ha a befektetés rögzítése során egy adat helytelenül került megadásra, az adott befektetés kiválasztásával, majd a Szerkesztés gombra történő kattintással szerkeszthető a bejegyzés. A felhasználónak csak egy befektetés kiválasztása után van lehetősége a befektetés módosítására. A gombra történő kattintást követően betöltődnek a különböző értékek az ablakban található szövegbeviteli mezőkbe.

A helytelenül megadott érték módosításához javítsa a már megadott értéket, majd ezután kattintson a Szerkesztés gombra.

A szerkesztési funkció felülete a 8-as számú ábrán található elrendezéshez hasonló.

### Exportálás PDF dokumentumként

A szoftverben rögzített befektetések exportálására az ablak felső sorában található Exportálás PDF-ként hivatkozásra szükséges kattintani. A megnyíló ablakban kiválasztható, hogy melyik, a számítógépen megtalálható könyvtárba szeretné menteni a PDF fájlt. Az alkalmazás alapértelmezetten a Dokumentumok mappába történő mentést ajánlja fel.

Miután kiválasztásra került a fájl mentésére szolgáló könyvtár, a mentéshez az ablak jobb alsó sarkában található mentés gombra szükséges kattintani.

11. ábra: Menüsor

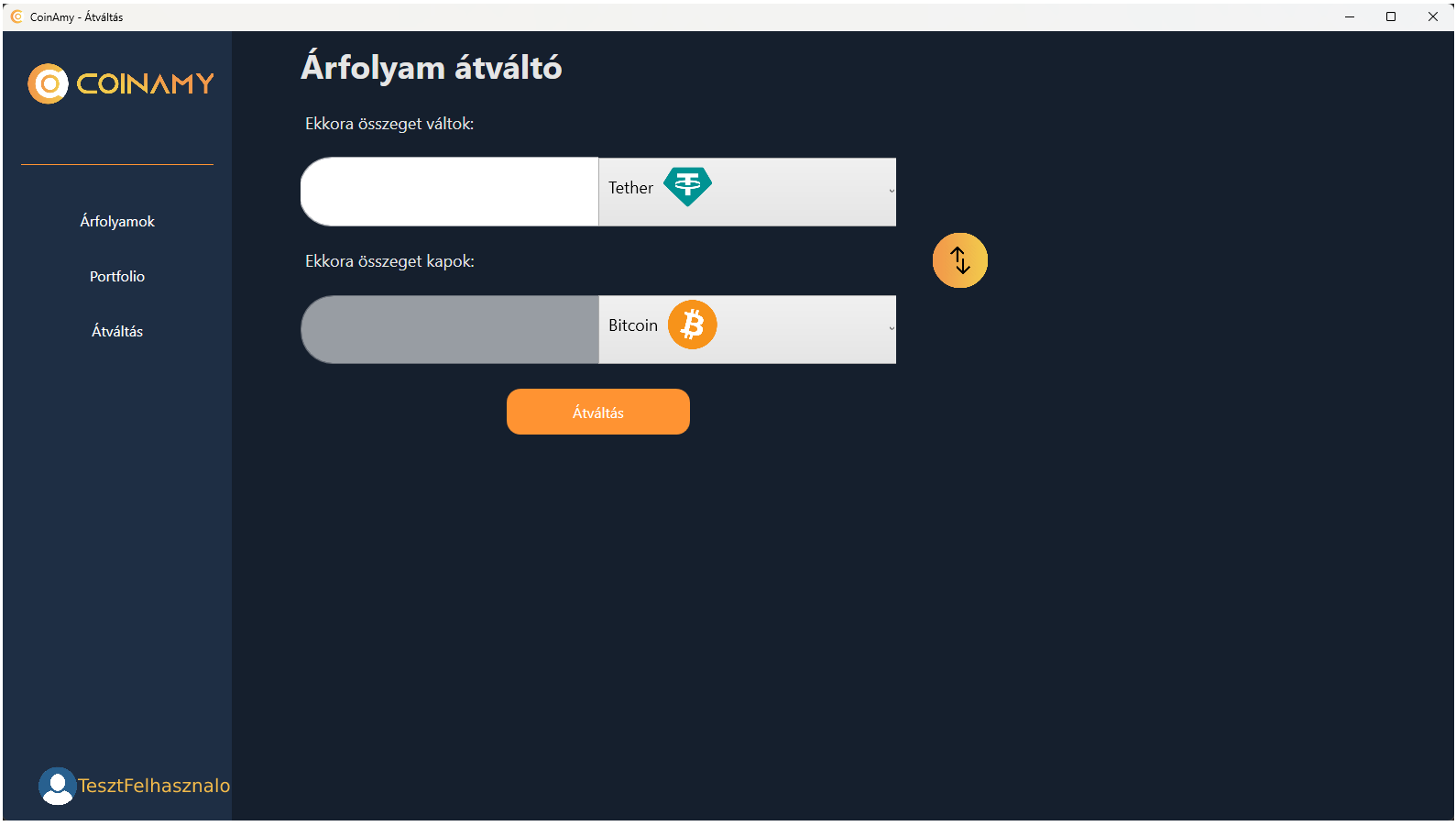
12. ábra: Exportált dokumentum

### Árfolyamátváltó felület

Az Átváltás menüpontra történő kattintással megnyílik a különböző kriptovaluták konvertálására lehetőséget nyújtó felület.

Az Ekkora összeget váltok felirat alatti mezőben megadható az átváltani kívánt kriptovaluta mennyisége. A mezőben egész- és tizedesvesszős számok is megadhatóak.

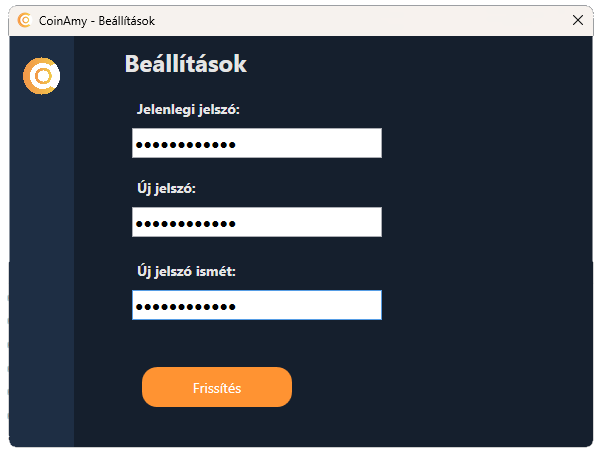
A mező mellett található legördülő listából kiválasztható az átváltandó kriptovaluta megnevezése. A könnyebb tájékozódást a kriptopénzekhez tartozó logók is elősegítik. A listában való görgetéssel érhetőek el a listában szereplő további választási lehetőségek. Ezután az alsó sorból kiválasztható az ellenértékként szolgáló valuta.

Amennyiben váltani szeretne a két mező tartalma között kattintson a mezők mellett található két irányú nyílt ábrázoló ikonra. Az Átváltás gombra történő kattintás után megtörténik a két kriptovaluta mennyisége közötti váltás az aktuális árfolyamokon. Az átváltandó kriptovalutáért kapható ellenértékű kapható mennyiségé az Ennyit kapok feliratú mezőben jelenik meg.

13. ábra: Árfolyam átváltó felület

### Belépési jelszó megváltoztatása

Ha úgy éri, hogy felhasználói fiókjának jelszava illetéktelen kezekbe került, megváltoztathatja a fiókjához tartozó jelszavát.

A módosításhoz a fő ablak bal alsó sarkában megtalálható felhasználónév mellett kattintson a profilképet ábrázoló ikonra. A megjelenő ablakban gépelje be jelenlegi jelszavát, majd adja meg a választott új jelszót, majd kattintson a Frissítés gombra. Kérem vegye figyelembe, hogy a jelszó megváltoztatásához az aktuális jelszó pontos megadása szükséges, ezenfelül a jelenlegi és a beállítani kívánt új jelszó nem egyezhet meg.

14. ábra: Felhasználói jelszó változtató ablak

### Hibaüzenetek

A program használata közben a legkülönfélébb hibaüzenetek tájékoztatják a felhasználót, ennek oka, hogy ezáltal tájékoztatás kapható, hogy hogyan kerülhetők el vagy javíthatók ki a szoftver működése közben jelentkező hibák. Ezek közül néhány a következőkben olvasható:

Hibás érték került megadásra:

A felhasználó által olyan adat került megadására, amely nem megfelelő az aktuális mező elvárt értékének. Többek között ilyen lehet, amikor egy mezőbe, ahol szám megadását várja a szoftver, szám érték kerül megadására.

Hiba a kapcsolódás során:

A szoftver a kriptovaluták aktuális árfolyamot az internet segítségével szerzi be, azonban az alkalmazás díjmentes felhasználhatósága okán az adatok frissítésére rendelkezésre álló keret korlátozott mértékben áll rendelkezésre. Ez a hibaüzenet abban az esetben jelenik meg, ha a felhasználó elérte az adott időszakra vonatkozó, internetről történő adatlekérések számát. A hiba orvoslásához nagyjából egy percet szükséges várni, míg az adatok újra lekérdezhetővé válnak.

Az előzőekben felsoroltaktól eltérő hibaüzenetek is megjelenhetnek az alkalmazásban, amelyek esetén mindig a hibaüzenetben foglaltak szerint szükséges eljárni.

### Fejlesztői elérhetőségek

Amennyiben szoftver fejlesztőjét bármilyen, a program használatával, fejlesztésével vagy bővítésével kapcsolatos okból szándékozik elérni, a következő elérhetőségeken megteheti:

* Vaju Ákos
* E-mail: [vaju.akos@gmail.com](mailto:vaju.akos@gmail.com)
* Telefonszám: +36 70 225 9804

# Záródolgozat összegzése

A CoinAmy egy Windows operációsrendszer alatt történő futtatásra készült asztali alkalmazás, amely lokális adatbázist használ.

Az alkalmazása fejlesztése során számos olyan dolgot ismertem meg még inkább részletekbe menően, amire mindaddig csak kevésbé részletes rálátásom nyílt. Példának okán a fejlesztés során volt lehetőségem első alkalommal komolyabb mennyiségű időt fektetni az API megoldások használatára.

Ezenfelül lehetőségem nyílt különféle beépített interfészek használatára a szoftverben, amelyek tanórai kereteken belül történő megismerésére nem adódott alkalom.

Emellett érdekes volt számomra néhány vezérlő és azok elrendezésének módosítását optimálisabbnak bizonyult az XAML kódból elvégezni.

A fejlesztés során a harmadik féltől származó LiveCharts NuGet Package alkalmazásával kapcsolatban merült fel a leginkább nehézség, mivel meglátásom szerint nem rendelkezik könnyen átlátható, jól értelmezhető dokumentációval, emellett a felhasználók közössége sem túlságosan aktív. Mindezek ellenére még mindig ez a leginkább karbantartott és legfejlettebb díjmentesen használható kiegészítő a WPF-ben való grafikonok készítésére. Mindemellett az sem könnyítette a helyzetemet, hogy a díjmentes felhasználhatóságot biztosító API csomag nem mindig működött az elvártaknak megfelelően és volt, amikor hosszú órákra leállt a működése.

Továbbá a záródolgozathoz szükséges kutatásaim alkalmával lehetőségem volt a végleges szoftverben ugyan nem felhasznált és alkalmazott, de ismereteim gyarapítására alkalmas tudás megszerzésére.

Az alkalmazás fejlesztésével a célom volt egy könnyen kezelhető, jól átlátható felülettel rendelkező és megbízható portfolió kezelő szoftver elkészítése, amely használata a felhasználó számára véleményem szerint sikerrel megvalósult.

# Hivatkozásjegyzék

1. API hívás és JSON feldolgozás I.

Hogyan válj jól fizetett C# programozóvá? – Koncz Balázs (112. oldal)

1. API hívás és JSON feldolgozás II.

<https://www.youtube.com/watch?v=N6JBjzPssQI> – (Megtekintve: 2022. 12. 20.)

1. Aszinkronitás

<https://www.youtube.com/watch?v=5a6WCBftjvw> – (Megtekintve: 2022.12.20.)

1. SHA-256 titkosítás

<https://www.youtube.com/watch?v=ZbUCgU3G1z4> - (Megtekintve: 2023.02.25.)

1. ValidationRule beépített osztály használata

<https://www.youtube.com/watch?v=5KF0GGObuAQ> – (Megtekintve: 2023.03.04.)

1. ListView tartalmának frissítése másik ablakból

<https://www.youtube.com/watch?v=BHdNd6ojeDI> – (Megtekintve: 2023.03.07.)

1. LiveCharts grafikon használata

<https://v0.lvcharts.com/App/examples/v1/Wpf/Line> - (Megtekintve: 2023.03.26)

1. Telepítő állomány készítése

<https://www.youtube.com/watch?v=7iVIfkVHKII> – (Megtekintve: 2023.03.27.)

# Ábrajegyzék

[1. ábra: Szoftver használati eset diagramja 7](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089258)

[2. ábra: Felhasználói felület kezdetleges terve 8](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089259)

[3. ábra: Adatbázis egyed-kapcsolat diagramja 10](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089260)

[4. ábra: A szoftver osztálydiagramja 11](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089261)

[5. ábra: Befektetés hozzáadásnak aktivitás diagramja 18](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089262)

[6. ábra: Bejelentkezési felület 28](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089263)

[7. ábra: Regisztrációs felület 29](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089264)

[8. ábra: Kezdőképernyő felülete 30](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089265)

[9. ábra: Portfolió kezelő felület 31](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089266)

[10. ábra: Befektetés hozzáadására szolgáló ablak 32](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089267)

[11. ábra: Menüsor 33](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089268)

[12. ábra: Exportált dokumentum 33](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089269)

[13. ábra: Árfolyam átváltó felület 34](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089270)

[14. ábra: Felhasználói jelszó változtató ablak 35](file:///C:\Users\vajua\Desktop\vaju_akos_istvan_zarodolgozat.docx#_Toc131089271)

# Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani Luczek Zsuzsa tanárnőnek a számtalan alkalomért, amikor tanítási időn túl is rendelkezésre állt, segített és azon volt, hogy minél többet legyek képes kihozni a záródolgozatomból.

Ezenfelül Kardos Krisztián tanárúrnak is köszönettel tartozom az egész éves osztályfőnöki tevékenységéért és azért, hogy a szoftverfejlesztés gyakorlat órákon átadott tudásával nagy mértékben hozzájárult a záródolgozatom elkészültéhez.

Továbbá hálás vagyok Csiszár Tamás, a konzulensem munkájáért, minden idő és energiaráfordításért, amivel a segítségemre volt.

Valamint köszönöm édesapám támogatását is, aki az elkészült szoftver tesztelésében volt a segítségemre.

1. Egy pénzügyi eszköz összességének értéke, a kibocsátott eszközök száma és azok árfolyamának szorzata [↑](#footnote-ref-1)
2. Rövidített kód vagy szimbólum, amelyet egy pénzügyi eszköz azonosítására használnak. [↑](#footnote-ref-2)