Tervezés I.

# Tartalomjegyzék

# Bevezetés

A személyes pénzügyi szoftverek arra a célra lettek kifejlesztve, hogy segítsék a felhasználók pénztárcáját beosztani. Mivel a személyes vagyon nyilvántartása pénzügyi alkalmazások nélkül hosszú és odafigyelést igénylő folyamat, így az emberek többsége nem is veszi a fáradságot e nyomon követések elkészítésére. Ezen alkalmazások több célra is használhatóak többek között számon tarthatjuk, mennyi pénzünk van jelenleg, mennyi pénzt tudtunk félretenni bizonyos hónapokban, milyen kategóriájú termékekre költöttünk a legtöbbet. Ezekből az adatokból olyan következtetéseket tudunk levonni amelyek segítik a sikeres pénzbeosztást, esetleges spórolást és a pénzügyi tudatosság kialakulását.

Többek közt készíthetünk egy képzeletbeli pénztárcát, ami azt a célt szolgálja, hogy kedvünk szerint megadhatjuk, hogy egy bizonyos kategóriájú termékcsoportra egy adott hónapban maximum mennyit fogunk költeni. Ezt a funkciót teljes mértékben a felhasználó szerkeszti és változtathatja kedve szerint. Az alkalmazások csupán értesíthetik a felhasználót a pénztárca jelenlegi helyzetéről például, hogy túlléptük a megszabott keretünket vagy a hónap lejártával mennyi pénzt sikerült megtakarítanunk a pénztárca funkció segítségével.

Bizonyos applikációk felajánlanak pénzspórolási tanácsokat is, amiket a felhasználó tranzakcióinak statisztikai elemzéseiből nyernek ki. Habár e tanácsok többsége fizetős funkció. A legnépszerűbb pénzügyi szoftverek legfőbb tulajdonsága hogy a felhasználók egy helyen láthatják az összes bankszámlájuk tranzakcióját, költési szokásukat és egyéb kimutatásokat. Így egy átfogó képet kapva a személyes pénzügyeikről, ezáltal segítve annak nyomon követését és esetleges jobbá tételét is.

Fontos megjegyezni, hogy manapság az alkalmazások nagy része elérhető telefonos applikáció formájában is. Mivel a mai emberek azonnali elérhetőséget kívánnak a nap bármely pontjában, főként, ha a személyes pénzügyeik alakulásáról van szó. Tehát a sikeres pénzügyi alkalmazások mind elérhetőek telefonos készülékre is.

Az első fejezetben a feladat rövid leírása található, leírva a fő problémát, annak okát, illetve tisztázva azokat a funkciókat, amivel a programnak rendelkeznie kell.

A második fejezetben bemutatásra fog kerülni pár elterjedtebb pénzügyi alkalmazás, azok funkcióival, pozitív illetve negatív tulajdonságukkal egyaránt. Illetve ezen alkalmazások összehasonlítása táblázat formájában.

A harmadik fejezetben bemutatásra kerülnek a szoftver képernyőtervei, illetve az azokhoz kötött funkcionális követelményeiket is, figyelembe véve a második fejezetben olvasottakkal.

A negyedik fejezetben leírásra kerülnek a program megírása alatt használt szoftvereket és e szoftverek ismertetése, illetve hogy miért ezeket az alkalmazásokat használtam a szakdolgozatok megírásához.

Az ötödik fejezetben felsorolásra kerülnek a jövőbeli tervek, fejlesztési irányok a jelenleg meglévő szoftvert illetően.

TÖBB

# Feladat leírása

A jelenleg piacon lévő pénzügyi alkalmazások mind nagyon különbözőek, mind megjelenítésben, mint funkciók terén. De egy közös tulajdonsággal mind rendelkeznek, méghozzá azzal hogy mind csak az amerikai FURA bankkártyák adatainak feldolgozására lettek kifejlesztve. Ennek az az oka, hogy ott minden bank egységes felülettel rendelkezik. Ezáltal sokkal könnyebb dolguk van a fejlesztőknek megírni egy számlaaggregátor applikációt, mint más országokban. Sajnos Magyarországon minden banktól exportált tranzakciós fájl felépítése különböző. Célunk egy olyan importáló szoftver írása, amely képes feldolgozni bármely magyar banktól exportált számlakivonat fájl adatait, és ezen adatokkal dolgozva megvalósítani a pénzügyi szoftverek egyéb funkcióit is.

Tehát a programnak képesnek kell lennie megjeleníteni az importált adatokat táblázatok formájában, a tranzakciókat képesnek kell FURA . Végül, de nem utolsó sorban az applikációnak tudnia kell kezelni mind a vásárolt és az eladott részvényeket, egy táblázatban megjeleníteni a nyereség illetve veszteség jelenlegi állását, a már megvásárolt részvény fajtáinak jelenlegi árfolyamát is FURA .

Az idei év (2017) szeptemberében a parlamentnek be lett nyújtva egy törvényjavaslat a bankok és a fintech-cégek számára vonatkozóan. Ha a PSD2 nevű irányelv elfogadásra kerül az a bankok számára három fő dolgot jelenthet.

Elsőként nagyobb versenyt és a bankolást megújító innovatív cégeknek pedig könnyebb piacra lépést. Az irányelv elsődleges célja a bankok és fintech cégek ügyfeleinek hatékonyabb védelme valamit nem utolsó szempontban a belső piac további erősítése. A belső piac erősítése előnyös, mint a fogyasztók, mind a kereskedők és a vállalkozók számára is.

Az irányelv új típusú szolgáltatók megjelenését is lehetővé teszi. Az úgynevezett külső szolgáltatók (TTP-k) kétféle új szolgáltatást nyújthatnak: számlainformációs és fizetéskezdeményezési szolgáltatókká válhatnak. Ezekhez az ügyfelek engedélyével banki adatokat lehet lekérdezni, vagyis a fintech szolgáltatók is hozzáférést kapnak a banki rendszerekhez.

A gyakorlatban ez azt jelenti, hogy új, például a kártyatársaságokat az értékláncból kihagyó fizetési szolgáltatások jöhetnek, illetve olyan számlaaggregátor szolgáltatók, akiknél egyszerre több számlát is kezelni lehet. Ezáltal rengeteg felhasználót segítve számlái és egyéb tranzakció átlátásaiban és nyomon követésével MORE.

# Pénzügyi alkalmazások bemutatása

Mivel a személyes pénzügyek a legtöbb ember számára fontosak így nagy igény van ilyen szoftverekre, főleg napjainkban, ahol minden alkalmazásnak van telefonos változata és az emberek kényelemben szeretik élni mindennapjaikat, tehát, csak pár újmozdulat segítségével láthatjuk pénzügyeinket.

Természetesen jó-pár pénzügyeket beosztó, segítő és nyomon követő applikáció létezik. Nem szokatlan az sem hogy egy felhasználó egyszerre használ több ilyen alkalmazást, mivel amit az egyik változat tud azt nem biztos, hogy a másik is. Vagy az is előfordulhat, hogy az egyik alkalmazás használata könnyen elsajátítható vagy esetleg jobban személyre szabható, jobb grafikus felülettel rendelkezik, míg a másik nem rendelkezik e tulajdonságokkal.

Habár a pénzügyi applikációk mindegyike rendelkezik egy Tranzakciók Importálása menüpontal, ahol pár kattintás segítségével új adatokat tudunk felvinni a rendszerbe, valamint Táblázat menüponttal, ami tartalmazza a beimportált, már az adatbázisban lévő tranzakciókat, aminek segítségével könnyűszerrel átláthatjuk a bizonyos időközben végbement változásokat a bankszámlán(in)kon. Kijelenthetjük, hogy e funkcióik alapkövetelményei a pénzügyi programok számára. Illetve könnyedén előfordulhat, hogy egy család használja ugyan azt az alkalmazást és a közös kasszájukat követik nyomon programban. Tegyük fel hogy, a család minden tagja külön banknál rendelkezik számlával (ami nagyon könnyen előfordulhat). Így egy szoftvernek képesnek kell lennie több bank adatait is feldolgozni, hogy eredményesen tudjon működni. A mai felhasználóknak nagy erénye a kényelmesség és ehhez hozzátartozik a kimutatások könnyű átlátása, ezalatt azt kell érteni, hogy látványos grafikonokkal illetve színekkel várják el a fontos adatok megjelenítését. Ha egy alkalmazás nem rendelkezik e funkciókkal, nem lesz túl népszerű a felhasználók körében MORE.

Rengeteg apró dolog tehet egy pénzügyi alkalmazás sikeressé, avagy kevésbé népszerűvé. A következőekben négy darab szoftvert mutatok be azok pozitív illetve negatív tulajdonságait is leírva.

## Mint.com

A Mint.com egy olyan alkalmazás, amely többek között segít beosztani a pénzt, emlékeztet a számlák befizetésére, sőt egy régi frissítésnek köszönhetően már be is fizetheti helyettünk is azokat. Több bankkártya adatait is tudja tárolni és kezelni. Teljesen biztonságos alkalmazás, több lépcsős beléptető rendszerrel rendelkezik. Csak a számlaforgalmat látja, azt kezelni nem tudja (számlabefizetésen kívül, ha a felhasználó igény tart erre a funkcióra). Mikor regisztrál egy ügyfél és megadja, a bankkártyája adatait a szoftver egy algoritmus segítségével megvizsgálja a felhasználó korábbi tranzakcióit, ezáltal megtudva költési szokásait. A Mint elérhető telefonra is. Sajnos a magyar bankok kezelésére nem képes, csak amerikai bankkártyákat ismer fel, viszont a PayPal fiókokat támogatja, ami Magyarországon is elérhető. A Mint.com rendelkezik a legjobb osztályzással a felhasználók körében. Amit a barátságos kezelőfelületének, könnyen használhatóságának és főként a sokrétű pénzügyi funkciók nyújtásáért érdemelt ki.



Ábra: Mint.com kezdőoldala.

Az ábrán látható fül a Mint.com kezdőoldala, ahogy látjuk a szoftver ezen része tartalmazza az éppen aktuális pénztárca állását figyeljük meg ahogy a program kiírja hogy mennyi pénzt költhetünk még az adott kategóriában, illetve ha túlköltöttünk akkor pirosra színezve azt így felhívva a felhasználó figyelmét a költési korlát átlépésére, természetesen láthatjuk mennyivel haladtuk túl a keretünket. Továbbá láthatjuk az esedékes számlabefizetések dátumát (négy hétre előretekintve) illetve azok pénzösszegeit is megjelenítve. Illetve az elmúlt fél év statisztikáit (jövedelem szempontjából). Nem utolsó sorban láthatjuk az osztályzást, amit a rendszer ad nekünk MORE

**Pozitív tulajdonságok**

* Könnyen személyre szabható.
* Könnyen átlátható.
* Automatizált rendszer.
* Többlépcsős beléptető rendszerrel rendelkezik.
* Ingyenes.
* Az osztályzások alapján a Mint.com a legjobb és legnépszerűbb személyes pénzügyi alkalmazás.
* Létezik telefonos alkalmazás formájában is.

**Negatív tulajdonságok**

* Csak az Amerikai bankkártyák adatait tudja kezelni.
* Ha fizetünk bizonyos összeget, akkor tanácsokat kaphatunk, hogyan spóroljunk több pénzt.

## YNAB (You Need A Budget)

A YNAB a nevéhez hűen, „csak” egy pénztárca létrehozásához és kezeléséhez jó. De arra tökéletesen megfelel feltéve, ha az ember elég időt szán az alkalmazás megismerésére. A program regisztrálástól számítva 34 napig ingyenes ezen idő letelte után havi 5$ a felhasználási díja. Barátságos kezelőfelülettel rendelkezik. Van telefonos változata is. Sajnos, ez az alkalmazás inkább csak a költésre fokuszál, a jövedelem nem jelenik meg a szoftverben. Habár a felhasználók véleménye szerint a szoftver hatékony használatának megtanulása sok időt emészthet fel, segítségünkre lehetnek az útmutató videók, amiből az interneten rengeteg található. A tranzakciókat a felhasználók importálják, majd később kategorizálhatják is azokat. A tranzakciók megfelelő kategóriába besorolása után a rendszer kiszámolja a pénztárca és azon korlátainak aktuális állapotát. Amikor először használjuk ezt az applikációt meg kell adnunk, mennyi pénzel rendelkezünk jelenleg. A YNAB a pénzünk jelenlegi helyzetével foglalkozik, és nem tekint előre esetleges jövőbeli tranzakciókra (a Mint.com-al ellentétben) MORE.



Ábra: a YNAB kezdőoldala

Az ábrán láthatjuk a felhasználó által létrehozott kategóriákat. Vegyük, észre hogy bal oldalt láthatjuk a fontosabb adatokat:

* Jelenleg rendelkezünk 3,362.01 $-al
  + Amiből jövedelem 1,740.00 $
  + A spórolt pénzösszeg 5,247.23 $
  + A tartozásunk (hitelkártyán) $3,625.22$

Illetve láthatjuk a pénztárcánk maximális korlátait, az ebből már elköltött pénzösszeget illetve kategóriánként az aktivitásunkat az elmúlt hónapban. A rendszer első induláskor már rendelkezik kategóriákkal, de természetesen ezeket lehet bővíteni is

Észre vehetjük, hogy a szoftver felajánlja a több bankszámla egyszeri kezelését is, ami például egy családi kassza kezeléséhez elengedhetetlen funkció.

**Pozitív tulajdonságok**

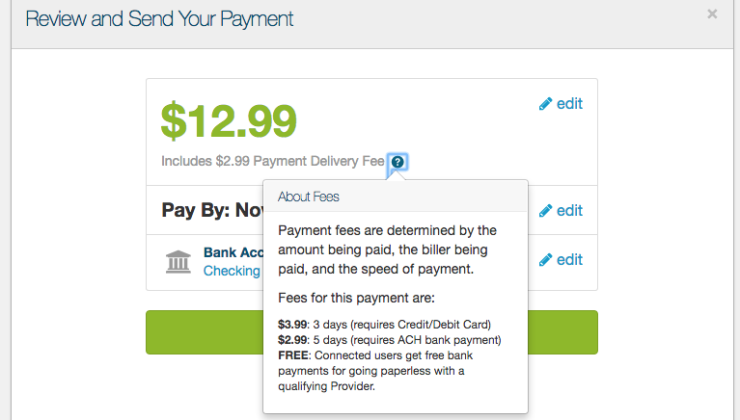
* Személyes „pénztárca” alkalmazás.
* Jó felhasználói felület.
* Sok oktató irodalom található róla.

**Negatív tulajdonságok**

* Sok idő, míg meg tanulja használni az ember.
* Nincs többlépcsős belépés.
* Drágább, mint a konkurens alkalmazások (34 napig ingyenes).
* Nincs nyomtatás funkció.
* Csak a költésre van kiélezve, nem kezel jövedelmet.

## Doxo

A Doxo egy olyan pénzügyi alkalmazás, ami főként a számla befizetésre és a számlával kapcsolatos dolgok tárolására van kiélezve. Habár az alkalmazás használata ingyenes, rendelkezik fizetős funkciókkal is pl. számlabefizetés. Automatikusan frissül számlaforgalom esetén. Egy kifejezetten jó alkalmazás azon emberek számára, akik nem szeretnének többé papír alapú tranzakciós dokumentumokat, számlákat tárolni. Automatikusan importál tranzakciókat és egyéb dokumentumokat, amik kapcsolódnak bankszámláinkhoz.



Ábra: Doxo számla befizetés

A fenti ábrán láthatjuk a Doxo egyik meghatározó funkcióját a számla befizetést, ami egy nagyon hasznos funkció, képzeljük el milyen kényelmes így csekket befizetni, hogy kattintunk párat és már kész is. Míg az alkalmazás használata ingyenes, ezért a funkcióért befizetésenként különböző összeget kell fizetni. Sajnos ezek az összegek csak a befizetés jóváhagyása előtt jelennek meg MORE.

**Pozitív tulajdonságok**

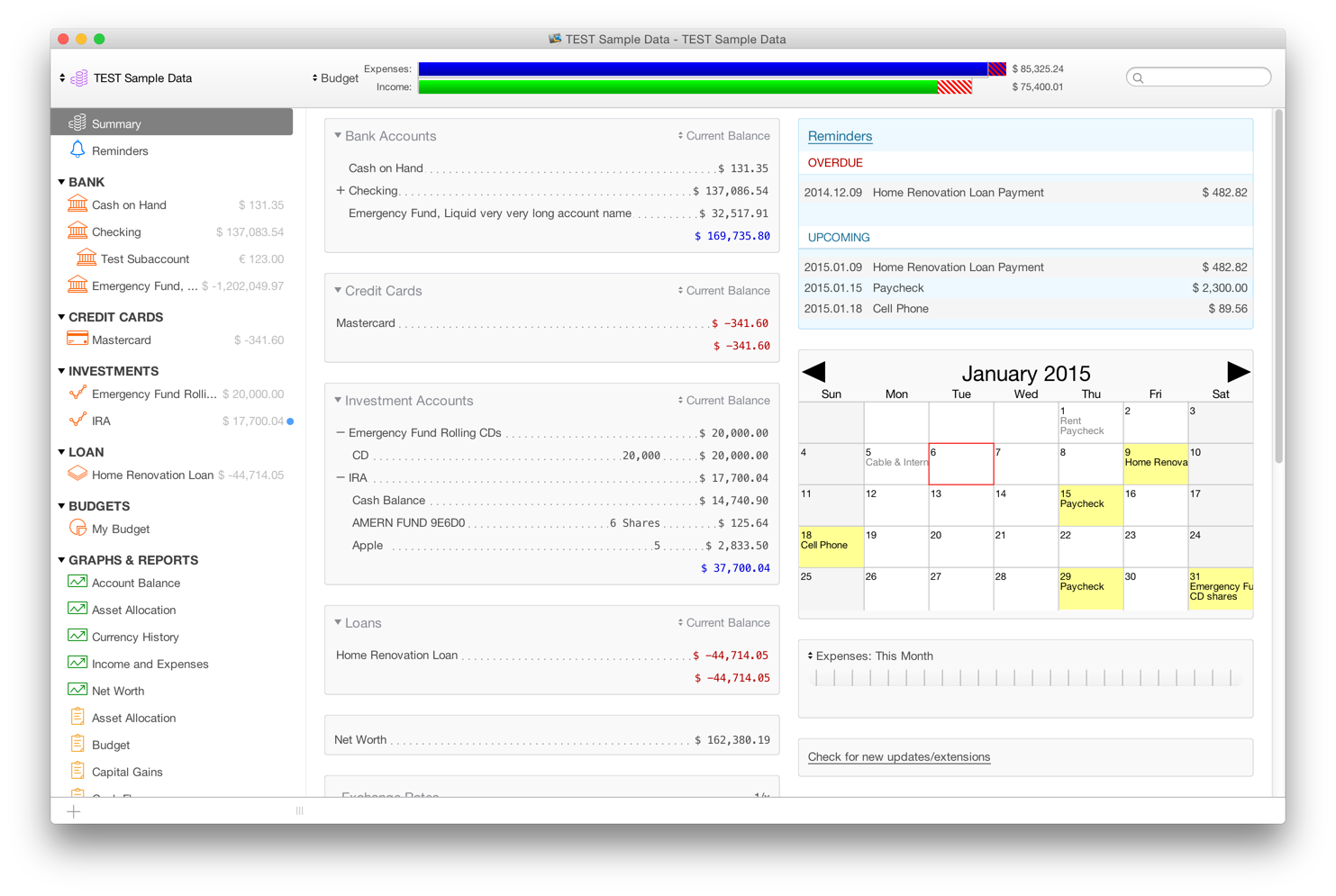
* Rendelkezik számlafizetési opcióval.
* Rugalmas.
* A használata nagy részben ingyenes.
* Nyomtatás opcióval is rendelkezik.
* Tud tárolni blokkokat és egyéb dokumentumokat.

**Negatív Tulajdonságok**

* Lassú lehet bizonyos funkciók esetén például számla befizetés, importálás.
* Sok negatív vélemény (felhasználók befizették a számláikat, de a pénz sosem érkezett meg).
* Nem ismer fel bizonyos bankkártyákat.

## Quicken Premier

A Quicken Premier inkább azoknak a felhasználóknak ajánlott, akik már ismeretesek egyéb számla menedzselős programokkal és tudják, mit keresnek. Ennek az alkalmazásnak nagy előnye az automatizálás, például automatikusan letölti tranzakcióinkat, melyeket utána automatikusan kategorizál. A többi alkalmazáshoz hasonlóan mutatja a befizetéseket, a számla jelenlegi állását. Habár a felület talán kicsit barátságtalanabb, mint a többi alkalmazásé, több funkcióit, megjelenítési módot is tartalmaz, mint konkurens társai. Sajnos ennek az alkalmazásnak nincs ingyenes változata, éves díja 80 $. Az alkalmazás rendelkezik pénztárca opcióval, különböző diagramokkal segíti az átláthatóságot. Valamint rendelkezik befektetés követéssel is, ami csak nagyon ritkán található meg személyes pénzügyi alkalmazásokban, mint funkció. Eredetileg csak személyes számítógépre tervezték. De ma már elérhető telefonra is.



Ábra: Quicken Premier kezdőoldala

A fenti ábrán láthatjuk a Quicken Premier kezdőoldalát és annak sokrétű információ megjelenítő részeit. Ahol többek közt láthatjuk az összes kiadásunkat illetve bevételünket. A befektetéseink jelenlegi helyzetét. A kiválasztott hónapban várható számlabefizetéseket, ezen számlák árait, fizetés időpontját és egyéb számlánkat érintő eseményeket. Az ábrán látható adatok szerint 15.-én jön meg a fizetésünk és 18.-án kell befizetnünk a telefonszámlánkat. Valamint baloldalon láthatjuk, mennyi grafikon megjelenítését kínálja fel a rendszer.

A lenti ábrán láthatjuk a Quicken Premier egyik fő funkcióját a befektetés kezelés. Az alsó grafikonon az Apple részvények árszerinti változását láthatjuk, ezzel is segítve a felhasználót a részvények változásának nyomon követésében. Láthatjuk azt is hogy a felhasználó rendelkezik 5 darab Apple részvénnyel, amit 2012.03.31.-én vásárolt 42,5$ értékben, összesen 212,50$ ért. Ettől jobbra pedig láthatjuk az Apple részvények jelenlegi állását és egy kimutatást, hogy 1233,41%-al többet ér jelenleg az öt darab Apple részvényünk, mint mikor vásároltuk őket.



Ábra: Quicken Premier befektetés követő funkció

**Pozitív tulajdonság**

* Rendelkezik befektetés követő funkcióval.
* Gyakran jönnek ki új verziói.
* Habár főként a befektetésekre fókuszál megtalálhatóak benne a többi alkalmazásoknál már jól ismert pénztárca, importálás, kimutatás funkciók is.
* Elsősorban asztali alkalmazásnak tervezték, de tudjuk irányítani a fiókunkat telefonról is.

**Negatív tulajdonságok**

* Csak egy számítógépre tudjuk telepíteni.
* Nincs ingyenes változata

## Összehasonlító táblázat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Alkalmazás neve | Univerzális importálás | Telefonos változat | Nyomtatás funkció | Fizetős alkalmazás | Befektetések kezelése |
| Mint.com | + | + | + | tanácsok fizetősek | + |
| YNAB | + | + | - | csak 34 napig ingyenes | - |
| Doxo | + | + | - | számla befizetés opció fizetős | - |
| Quicken Premier. | + | + | + | éves díja 80$ | + |

# Az alkalmazás menüpontjai

## Kezdőlap



Ábra: Az alkalmazás bejelentkező felülete

A bejelentkező oldalon láthatunk két szövegmezőt amelyek segítségével a felhasználó be tud jelentkezni a rendszerbe. A felső szövegmező a felhasználónév azonosításért felel, míg az alatte lévő a jelszó azonosításért, a jelszót azonosító szövegdobozban minden karakter egy pont formájában jelenik meg, megvédve a felhasználót az esetleges jelszó lopástól. A sikeres bejelentkezéshez ezen mezők kitöltése természetesen kötelező Illetve a Kezdőlapon láthatunk még két gombot is, amik a Bejelentkezés végrehajtásáért, valamint a Regisztrációért felelősek. A billentyűzet fókusza az applikáció indulásánál alapértelmezetten a felhasználónév mezőben van.

Ha a felhasználó úgy kattint rá a Bejelentkezés gombra, hogy a felhasználónév mezőt és/vagy a jelszó mezőt üresen hagyta akkor az oldal, értesítést (Messagebox) küld az érvénytelen próbálkozás után.(A felhasználónév és a jelszó mezők üresek!).

Ha a felhasználó kitöltötte a felhasználónév a jelszó mezőt, de a felhasználónév vagy a jelszó nem egyezik az adatbázisban szereplő adatokkal a rendszer, értesítést (Messagebox) küld a bejelentkezni kívánó személynek a hibáról (Nem megfelelő felhasználónév vagy jelszó). Ezután a felhasználó újrapróbálkozhat, de maximum három lehetősége van hibázásra, utána nem próbálkozhat bejelentkezhetni két percig.

A bejelentkezés folyamata (felhasználónév illetve jelszó ellenőrzés) elkezdődik a Login gombra kattintva vagy egy Enter gomb lenyomásával is, feltéve, ha e mezők nem üresek. A rendszer küld egy kérést a belső adatbázis számára a felhasználónév és a hozzá tartozó jelszóval. A bejelentkezés akkor lesz sikeres, ha a felhasználónév és jelszó kombináció megegyezik a tárolt adattal a felhasználók adatbázisában. Ha az adott személy még nem regisztrált, akkor a Regisztrálás gombra kattintva létrehozhat egy új fiókot.

Ha a felhasználó rákattint a Regisztrálás gombra, egy új ablak jelenik meg. Ahol látni fog három szövegmezőt ahova be kell írnia a leendő felhasználónevét, amivel később be tud jelentkezni illetve a bejelentkezéshez szükséges jelszavát is.



Ábra: Az alkalmazás bejelentkező felülete

A jelszó beírásra kétszer van szükség, természetesen e jelszavaknak egyezniük kell különben a regisztráció nem lesz sikeres.

A regisztráció abban az esetben sem sikeres, ha az adatbázisban már szerepel ilyen nevű felhasználó. Ilyen esetben az oldal értesítést (Messagebox) küld a regisztrálni kívánó felhasználónak, hogy az a felhasználónév, amivel regisztrálni kíván már foglalt és újat kell választania.

A leendő felhasználónévnek legalább négy karakter hosszúnak kell lennie, míg a jelszó hossza legalább hat karakter és tartalmaznia kell számot is. A felhasználónév és a jelszó formájára (bizonyos karakterek megkövetelt használatára) nincsenek egyéb megkötések (például nagybetűk használata a jelszóban illetve speciális karakterek használata).

Ha a felhasználó kitöltötte a három szövegmezőt és rákattint a Regisztrálás gombra hat lehetséges esemény következhet be, amiről a regisztráló fél minden esetben értesítést fog kapni (Messagebox).

* A regisztráció sikeres volt, ezután a felhasználó már be tud lépni az új fiókjával.
* A regisztráció nem volt sikeres, mivel a felhasználónév már foglalt.
* A regisztráció nem volt sikeres, mivel a jelszómezők tartalma nem egyezik.
* A regisztráció nem volt sikeres mivel a felhasználónév nem elég hosszú (legalább 4 karakter).
* A regisztráció nem volt sikeres mivel a jelszó nem elég hosszú (legalább 6 karakter).
* A regisztráció nem volt sikeres mivel a jelszó nem tartalmaz legalább egy számot.

Ha a regisztrálás sikeres volt, akkor a felhasználó újra a kezdőoldalt fogja látni.

Ha a bejelentkezés sikeres a felhasználó az alkalmazás főmenüjét fogja látni, ahol a jobb felső sarokban láthatja az értesítéseit és az aktuális felhasználó nevét. Egy felhasználó értesítései attól függenek, hogy az aktuális dátumhoz mérten mikor importált legalább egy új tranzakciót. Az új tranzakció alatt jelen esetben egy olyan vagy banki, vagy tőzsdei tranzakciót kell érteni, ami még nem szerepelt az importálás előtt az alkalmazás adatbázisában.

A felhasználónak szóló értesítésnek három színe lehet:

* Piros: A felhasználó nem importált új tranzakciót több mint egy hónapja.
* Sárga: A felhasználó legalább tizenöt, de legfeljebb egy hónapja nem importált
* Zöld: Ha az értesítések száma nulla, tehát a felhasználó importált új legalább egy új tranzakciót az elmúlt tizenöt napban egy zöld pipa fog megjelenni az értesítések helyén ezzel jelezve a felhasználónak, hogy napra készen áll a személyes pénzügyeivel.

## Tranzakciók Importálása

Amint sikeresen belépett a felhasználó az alapértelmezett oldal, ami automatikusan megjelenik neki az a Tranzakciók importálása menüpont. Amely tartalmaz két almenüt:

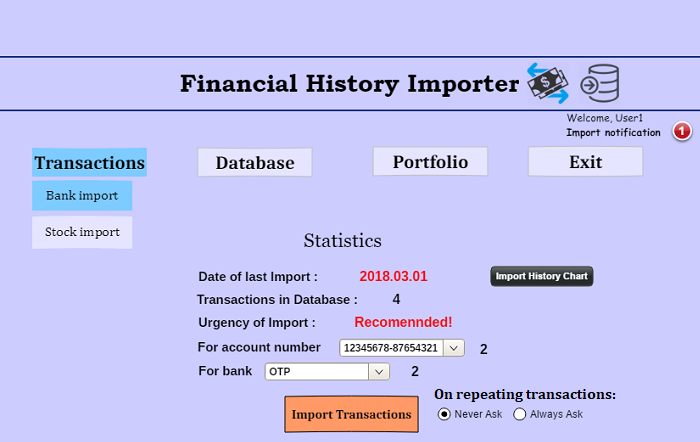
* Banki Tranzakciók importálása
* Tőzsdei Tranzakciók Importálása

A két almenü között az almenük nevét tartalmazó gombra való kattintás segítségével tudunk váltani. Az aktuálisan kiválasztott menü gombja különböző háttérszínnel rendelkezik, mint a többi gomb, ennek segítségével a felhasználó könnyen tudja, melyik menü van kiválasztva jelenleg. A sikeres bejelentkezést követően az alapértelmezetten kiválasztott almenü a Bank Tranzakciók importálása.

### Banki Tranzakciók importálása

A felhasználó a Banki Tranzakciók importálása menüre rákattintva láthat egy rövid áttekintést az általa már beimportált és a szoftver adatbázisában lévő banki tranzakciókról.

* Utolsó importálás ideje
* Tárolt banki tranzakciók az adatbázisban összesen
* Importálás szükségessége
* Különböző számlákhoz tartozó tranzakciók száma (számlaszám vagy különböző bankok száma egy legördülő listában).
* Különböző bankokhoz tartozó tranzakciók száma



Ábra: Banki Tranzakciók importálása

Az utolsó importálás ideje azt a dátumot jeleníti, meg amikor a felhasználó sikeresen importált banki tranzakciókat. A sikeres importálás jelen esetben azt jelenti, hogy a felhasználó egy olyan tranzakciót vitt fel a rendszer adatbázisába amit eddig az nem tartalmazott, tehát ha egy olyan fájlt importálunk, amiben csak olyan adatok vannak, amik már szerepelnek az adatbázisban akkor ez a dátum nem fog frissülni. (Kivétel erre az esetre, ha a felhasználó ettől függetlenül is importálni szeretné az adott tranzakciókat.).

A Tárolt banki tranzakciók azt a számot jeleníti, meg hogy hány tranzakciót importáltunk összesen, banktól, számlaszámtól, dátumtól függetlenül.

Az importálás szükségessége egy jelzés a felhasználó számára, hogy ha eredményesen szeretné nyomon követni pénzügyeit, akkor ajánlott egy bizonyos időközönként importálnia új tranzakciókat. A "Szükséges" felírat, akkor jelenik meg, ha a legfrissebb újonnan beimportált banki tranzakció dátuma az aktuális dátum előtt van legalább egy hónappal. Különben ha az utolsó banki tranzakció felvitele óta legalább tizenöt, de legfeljebb egy hónap telt el, az "Ajánlott" felirat fog megjelenni. Ha a legfrissebb banki tranzakció importálása óta még nem telt el legfeljebb tizenöt nap a felhasználó a "Naprakész" feliratot láthatja. A feliratoktól függően változik a felhasználó értesítése.

A Különböző tárolt számlák felirat és a mellette lévő legördülő lista csak abban az esetben látható a felhasználó számára, ha már sikeresen importált legalább egy darab tranzakciót. Ebben a sorban a felhasználó láthatja az általa beimportált fájlból kinyert bankszámlaszámo(ka)t, amelyeket egy legördülő lista segítségével választhat ki. Egy elem kiválasztása esetén megjelenik egy szám a lista mellet, amely jelzi az kiválasztott számlaszámtól importált tranzakciók számát. Ezen információ segítségével könnyen láthatjuk melyik bankszámlánkról, hányszor importálunk már banki tranzakció(ka)t a rendszerbe.

A Különböző tárolt bankok felirat és a mellette lévő legördülő lista csak abban az esetben látható a felhasználó számára, ha már sikeresen importált legalább egy darab tranzakciót. Ebben a sorban a felhasználó láthatja az általa beimportált fájlból kinyert bankneveket, amelyeket egy legördülő lista segítségével választhat ki. Egy elem kiválasztása esetén megjelenik egy szám a lista mellet, amely jelzi az kiválasztott banktól importált tranzakciók számát. Ezen információ segítségével könnyen láthatjuk melyik banktól, hányszor importálunk már banki tranzakció(ka)t a rendszerbe.

Ezen négy importálási statisztikát tartalmazó szövegmező mellet láthatunk egy gombot, ami csak abban az esetben jelenik meg, ha a felhasználó legalább egy tranzakciót sikeresen importált az adatbázisba. Ez a gomb Importálási Történet diagram felirattal rendelkezik. Ha erre a gombra rákattint a felhasználó egy új ablak fog megjelenni, ami tartalmaz egy legördülő listát és egy diagramot. A legördülő lista tartalmazza azon banko(ka)t amely fájljait felhasználva a felhasználó sikeresen importált a rendszer adatbázisába legalább egy tranzakciót. A lista aktuálisan kiválasztott elemétől függ a diagram által megjelenített adat. Alapértelmezetten az Összes opció van kiválasztva, ami azt jelenti hogy az összes banktól származó importálások számát fogja megjeleníteni a diagram. A diagram X tengelye tartalmazza a tranzakció importálásának dátumát, növekvő sorrendben. Az Y tengely pedig jelzi az adott napon felvitt tranzakciók számának összegét. Továbbá diagram jobb felső sarkában láthatunk egy gombot, ami a vissza felirattal rendelkezik. Erre rákattintva a felhasználó visszakerül a Banki Tranzakciók Importálása menübe.

Az Importálási Történet diagram gomb alatt láthatunk két rádió gombot, amik felett az alábbi felirat szerepel: "Ismétlődő tranzakció esetén". Ezen rádió gombok változtatása kihatással van az importálás folyamán fellépő esetleges ismétlődő tranzakciók felismerésére és szoftver adatbázisába történő elmentésükre. A rádió gombok két opciót ajánlanak fel:

* Mindig kérdezzen rá
* Soha ne kérdezzen rá

A rádió gomboktól függetlenül, a szoftver egy algoritmust használ a beimportálni kívánt fájlok adataira, ami összehasonlítja a már eltárolt tranzakció adataival és megpróbálja kiszűrni azokat az adatokat, amik nagy valószínűséggel már el vannak tárolva a rendszer adatbázisában, többek közt figyelembe veszi a tranzakció végbemenetelének dátumát, tranzakció pénzösszegét és az akkori számla egyenleget és nem utolsó sorban a tranzakció bankszámlaszámát is.

Ha a Mindig kérdezzen rá rádió gombot választja ki a felhasználó, abban az esetben, az importálás folyamata alatt egy valószínűleg ismétlődő tranzakciót talál a szoftver, értesíti a felhasználót, hogy valószínűleg az aktuális fájlból importált bizonyos tranzakciók már szerepelnek az adatbázisban, kiírva a tranzakció végbemenetelének dátumát és tranzakció pénzösszegét is. Ezek után az alkalmazás két lehetőséget kínál fel a felhasználó számára:

* Figyelmeztetés figyelmen kívül hagyása, a felhasználó tisztában van az esetleges redundanciával de ezt figyelmen kívül hagyja és szeretné beimportálni az adott tranzakciót a rendszerbe.
* Ne importálja az aktuális tranzakciót, ez esetben a felhasználó nem importálja a tranzakciót mivel az már szerepel a rendszer adatbázisában így az redundáns adat lenne.

Soha ne kérdezzen rá rádió gomb kiválasztása esetén a felhasználó átadja az irányítást a szoftvernek ami saját kezűleg dönti el hogy importáljon-e egy bizonyos tranzakciót, figyelembe véve annak a lehetőségét hogy már eltárolt adatról van-e szó.

Alapértelmezetten a Soha ne kérdezzen rá rádió gomb van kiválasztva.

Tranzakciók Importálása gomb: Erre a gombra rákattintva megjelenik a felhasználó számára két választási lehetőség egy Messagebox formájában. Amin az alábbi két lehetőséget kiválasztására van lehetősége a felhasználónak:

* Automatikus importálás
* Felhasználó által deklarált importálás

Ha a felhasználó az Automatikus importálás opciót választja akkor a következőekben az általa kiválasztott fájlra az alkalmazás bizonyos algoritmusokat használva megpróbálja azokat a cellákat és oszlopokat beazonosítani, amelyek szükségesek egy tranzakció egyértelmű azonosítására. Amik a következőek:

* Fájlhoz tartozó bankszámlaszám
* Tranzakció dátuma
* Tranzakció összege

Tehát ha ez a három adat sikeresen kiolvasható egy fájlból akkor már van beolvasható tranzakció amit beilleszthetünk az adatbázisba. Ha ezt a három adatot nem sikerül megtalálnia az algoritmusnak akkor értesítést küld a felhasználónak hogy az adott fájlra alkalmazza a Felhasználó által deklarált importálást. Természetesen egy tranzakcióhoz több adat is tartozhat, ez teljes mértékben a banktól függ ahonnét a tranzakció történetet tartalmazó fájl származik. Ilyen kiegészítő adat lehet:

* Tranzakció leírása
* Jelenlegi egyenleg

Fájl típusától függően több cella is tartalmazhat tranzakcióhoz tartozó leírást. Az alkalmazás által használt beolvasó algoritmus megpróbálja felismerni a legtöbb lehetséges leírásra illeszkedő cellát.

A jelenlegi egyenleget a legtöbb fájl megjeleníti, de természetesen a bankoktól letöltött fájlok mint már korábban is leírtam nagyon különböznek egymástól. Így hát az is megtörténhet hogy bizonyos fájlok nem rendelkeznek ezzel az adattal.

Ha az algoritmus sikeresen felismerte a kellő cellákat és kiolvasta az összes lehetséges tranzakciót, ezek után elkezdődik a kiolvasott tranzakciók adatainak összehasonlítása a már eltárolt tranzakciók adataival.

Ha az ismétlés megtalálásáért felelő algoritmus talál egy valószínűleg már eltárolt tranzakciót a felhasználó korábbi döntéseit figyelembe véve reagál (Ismétlődő tranzakciók esetén rádió gombok).

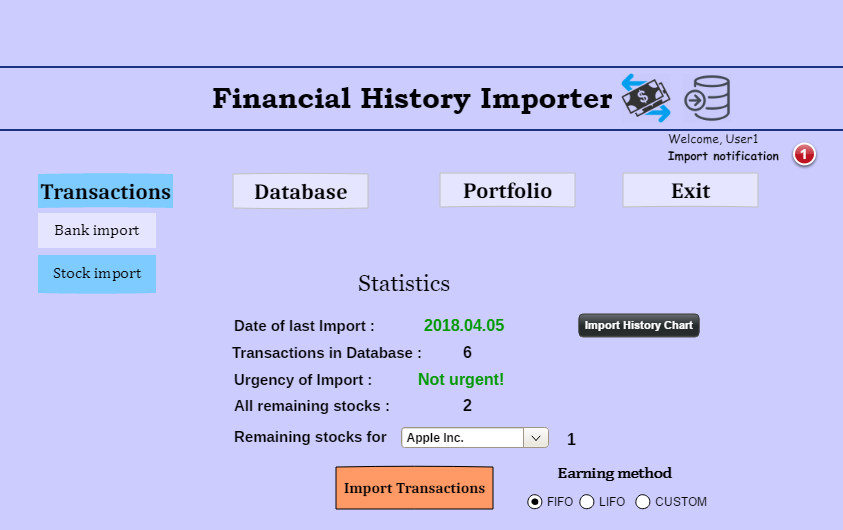
Miután az összes beimportált tranzakció átvizsgálásra került, az alkalmazás exportálja a kellő adatokat az adatbázisba. Így azok már később megtekinthetőek a szoftverben táblázat formájában, illetve frissítésre kerül a felhasználóhoz tartozó statisztika is.

//todo user-specified import

### Tőzsdei Tranzakciók importálása

A felhasználó a Tőzsdei Tranzakciók importálása menüre rákattintva láthat egy rövid áttekintést az általa már beimportált és a szoftver adatbázisában lévő tőzsdei tranzakciókról.

* Utolsó importálás ideje
* Tárolt tőzsdei tranzakciók az adatbázisban összesen
* Importálás szükségessége
* Jelenleg birtokolt részvények összes száma
* Jelenleg birtokolt részvények száma különböző cégekre



Ábra: Tőzsdei Tranzakciók importálása

Az utolsó importálás ideje azt a dátumot jeleníti meg amikor a felhasználó sikeresen importált tőzsdei tranzakció(ka)t. A sikeres importálás jelen esetben azt jelenti, hogy a felhasználó egy olyan tranzakciót vitt fel a rendszer adatbázisába amit eddig az nem tartalmazott, tehát ha egy olyan fájlt importálunk, amiben csak olyan adatok vannak, amik már szerepelnek az adatbázisban akkor ez a dátum nem fog frissülni. (Kivétel erre az esetre, ha a felhasználó ettől függetlenül is importálni szeretné az adott tranzakciókat.).

A Tárolt tőzsdei tranzakciók azt a számot jeleníti, meg hogy hány tranzakciót importáltunk összesen, cégtől, tranzakció típusától, dátumtól függetlenül.

Az importálás szükségessége egy jelzés a felhasználó számára, hogy ha eredményesen szeretné nyomon követni pénzügyeit, akkor ajánlott egy bizonyos időközönként importálnia új tranzakciókat. A "Szükséges" felírat, akkor jelenik meg, ha a legfrissebb újonnan beimportált tőzsdei tranzakció dátuma az aktuális dátum előtt van legalább egy hónappal. Különben ha az utolsó tőzsdei tranzakció felvitele óta legalább tizenöt, de legfeljebb egy hónap telt el, az "Ajánlott" felirat fog megjelenni. Ha a legfrissebb tőzsdei tranzakció importálása óta még nem telt el legfeljebb tizenöt nap, a felhasználó a "Naprakész" feliratot láthatja. A feliratoktól függően változik a felhasználó értesítése.

A Jelenleg birtokolt részvények összes száma felirat mellett egy darabszámot láthat a felhasználó, méghozzá azon már a szoftver adatbázisába beimportált vásárolt részvények darabszámát amelyekből az esetleges eladott tranzakciók darabszámai levonásra kerültek. Így a felhasználó könnyűszerrel láthatja hogy hány darab vásárolt részvénnyel rendelkezik még az adott időszakban összesen.

A Jelenleg birtokolt részvények száma különböző cégekre felirat mellett egy darabszámot láthat a felhasználó, méghozzá azon már a szoftver adatbázisába beimportált vásárolt részvények darabszámát amelyekből az esetleges eladott tranzakciók darabszámai levonásra kerültek azon termékre leszűrve, amit a felhasználó kiválasztott a felirat mellet lévő legördülő listából. Természetesen a legördülő lista azon részvény termékneveket tartalmazza amit a felhasználó már sikeresen importált a rendszer adatbázisába.

Ezen négy importálási statisztikát tartalmazó szövegmező mellet láthatunk egy gombot, ami csak abban az esetben jelenik meg, ha a felhasználó legalább egy tranzakciót sikeresen importált az adatbázisba. Ez a gomb Importálási Történet diagram felirattal rendelkezik. Ha erre a gombra rákattint a felhasználó egy új ablak fog megjelenni, ami tartalmaz egy legördülő listát és egy diagramot. A legördülő lista tartalmazza azon részvény termékneveket, amelyet a felhasználó egy fájl importálása által sikeresen hozzáadott a rendszer adatbázisához. A lista aktuálisan kiválasztott elemétől függ a diagram által megjelenített adat. Alapértelmezetten az "Összes" opció van kiválasztva, ami azt jelenti hogy az összes terméknévtől származó importálások számát fogja megjeleníteni a diagram. A diagram X tengelye tartalmazza a tranzakció importálásának dátumát, növekvő sorrendben. Az Y tengely pedig jelzi az adott napon felvitt tranzakciók számának összegét. Továbbá diagram jobb felső sarkában láthatunk egy gombot, ami a vissza felirattal rendelkezik. Erre rákattintva a felhasználó visszakerül a Tőzsdei Tranzakciók Importálása menübe.

Az Importálási Történet diagram gomb alatt láthatunk három rádió gombot, amik felett az alábbi felirat szerepel: "Nyereség kiszámítási módszer". Ezen rádió gombok változtatása kihatással van az importálás alatt kiszámolt esetleges vásárlás-eladás nyereségek illetve veszteségek összegének kiszámolási módszerére. A három lehetséges opció:

* FIFO
* LIFO
* CUSTOM

A **FIFO (First In-First Out)** azaz amit időrendi sorrendben előbb vásároltunk meg részvényt, azt fogjuk először "felhasználni" tehát eladni. Az alábbi ábrán, egy egyszerű példával szemléltetem a FIFO elv működését.



Ábra: FIFO elv működése

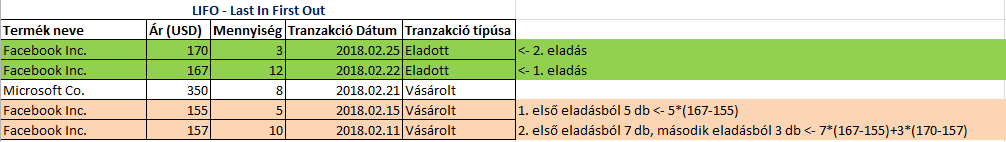
Mivel FIFO elvet alkalmazunk ezért a vásárolt tranzakciókat alulról-felfelé dolgozzuk fel. A fenti ábrából láthatjuk hogy 2018.01.22.-én vásároltunk tíz darab Apple Inc. részvényt, darabonként 170 dollár áron, hívjuk el ezt a tranzakciót Első tranzakciónak. Majd 2018.01.25.-én vásároltunk öt darab Apple Inc. részvényt, darabonként 165 dollár áron hívjuk el ezt a tranzakciót Második tranzakciónak, illetve 2018.02.02.-án vásároltunk nyolc darab Coca Cola Co. részvény darabonként 350 dollár áron, de mivel ezeket a részvényeket nem adtuk el ezért a későbbiekben nem foglalkozunk vele. Majd láthatjuk hogy eladtunk tizenkét darab Apple részvényt 2018.02.02.-án méghozzá 180 dollár értékben. Ha FIFO elvet alkalmazunk a nyereség-veszteség kiszámolására akkor azt a részvényt adtuk el amit legelőször megvettünk. Tehát mivel tizenkettő darab részvényt adtunk el ezért mind az Első tranzakció illetve a Második tranzakcióval számolnunk kell, mivel az Első tranzakcióhoz csak tíz darab részvény tartozik. Mivel FIFO elvet alkalmazunk ezért az Első tranzakcióból adtunk el tíz darab részvényt, aminek a nyeresége a következő: 10\*(180-170) ahol a 10 jelenti az eladott részvények számát, a 180 az eladási árat és 170 a vételi árat. Még maradt két eladott részvényünk az első eladásból amit a Második tranzakcióból adtunk el, mivel az Első tranzakcióból eladtuk minden részvényt. Ennek az eladásnak a nyeresége a következő: 2\*(180-165) ahol kettő az eladott részvények száma, 180 a részvények eladási ára, 165 a Második tranzakcióból származó vételi ár.

Tehát a 12 darab eladott Apple Inc. részvény nyeresége összesen 10\*(180-170)+2\*(180-165) ami egyenlő száz-harminccal.

Az első eladás nyereség-veszteség kiszámolása után van összesen három darab Apple Inc. részvényünk a Második tranzakcióból. De a második eladás során pont három darabot adunk el. Így a második eladáshoz tartozó nyereségét a következő módon számoljuk ki: 3\*(182-165) ahol három az eladott részvények száma, 182 a részvények eladási ára 165 a részvények vételi ára.

Tehát második eladáshoz tartozó nyereség egyenlő ötveneggyel. Illetve rendelkezünk még nyolc darab Coca Cola Co. részvénnyel.

A **LIFO (Last In-First Out)** azaz amit időrendi sorrendben később vásároltunk meg részvényt, azt fogjuk először "felhasználni" tehát eladni. Az alábbi ábrán, egy egyszerű példával szemléltetem a LIFO elv működését is.



Ábra: LIFO elv működése

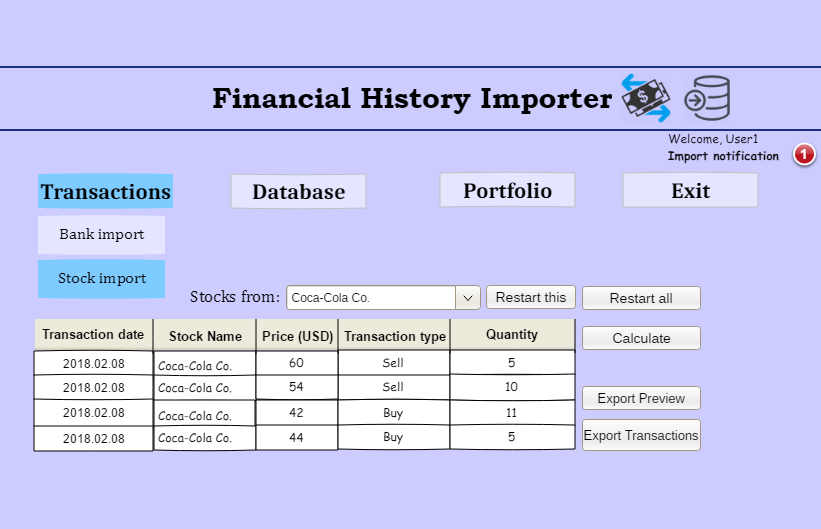
Az ezelőtti példához hasonlóan itt is nevezzük el a vásárolt tranzakciókat. De mivel most LIFO elvet alkalmazunk a vásárolt tranzakciókat fordított időrendi sorrendben fogjuk feldolgozni. Az ábráról leolvashatjuk hogy 2018.02.21.-én vásároltunk nyolc darab Microsoft Co. részvényt háromszáz-ötven dollár értékben, de mivel ezt nem adtuk el, nem foglalkozunk vele a későbbiekben. Láthatjuk 2018.02.15.-én vásároltunk Facebook Inc. részvényt, öt darabot, darabonként száz-ötvenöt dollár áron, nevezzük el Első tranzakciónak. Valamint 2018.02.11.-én is vásároltunk tíz darab Facebook Inc. részvényt, darabonként száz-ötvenhét dollár áron, nevezzük el Második tranzakciónak. Észrevehetjük hogy 2018.02.22.-én eladtunk tizenkét darab Facebook Inc. részvényt, ennek a nyereség-veszteség értékét a következő módon tudjuk kiszámolni LIFO elv használatával: mivel tizenkét darab részvényt adtunk el, és az Első tranzakció csak öt darab részvényt tartalmaz. Ezért az első öt darab eladott részvényhez számolt nyereség a következő: 5\*(167-155) ahol öt az eladott részvények száma, 167 az eladott részvények ára és 155 a vásárolt részvények ára. Mivel tizenkét darabot adtunk el és eddig csak az első ötre számoltuk ki a nyereséget a Második tranzakcióból kell kiszámolnunk maradék hét eladott részvény nyereségét. Az első eladásból vett hét részvény és a Második tranzakcióból számolt nyereség a következő: 7\*(167-157) ahol hét az eladott részvények száma, 167 az eladott részvények ára, 157 a vásárolt részvények ára.

Tehát az első eladáshoz tartozó nyerség egyenlő 5\*(167-155)+7\*(167-157) ami egyenlő százharminccal.

A második eladáshoz tartozó nyereség kiszámolása a következő módon történik: mivel a Második tranzakcióhoz már csak három részvény tartozik, és a második eladás során három részvényt adtunk el így a következő képletet alkalmazhatjuk: 3\*(170-157) ahol három az eladott részvények száma, 170 az eladott részvények ára és 157 a vásárolt részvények ára.

Tehát a második tartozó nyerség egyenlő harminckilenccel. Illetve rendelkezünk még nyolc darab Microsoft Co. részvénnyel.

Ha a felhasználó a **CUSTOM** nyereség-veszteség kiszámítási módot választja az alábbi táblázat és a gombok segítésével a saját íze-kedve szerint köthet össze eladott és vásárolt tranzakciókat.



Ábra: CUSTOM nyereség-veszteség kiszámítás

Az oldalon láthatunk egy táblázatot ami tartalmazza a frissen beimportált tranzakciókat. Méghozzá öt különböző adatot megjelenítve minden tranzakció esetében:

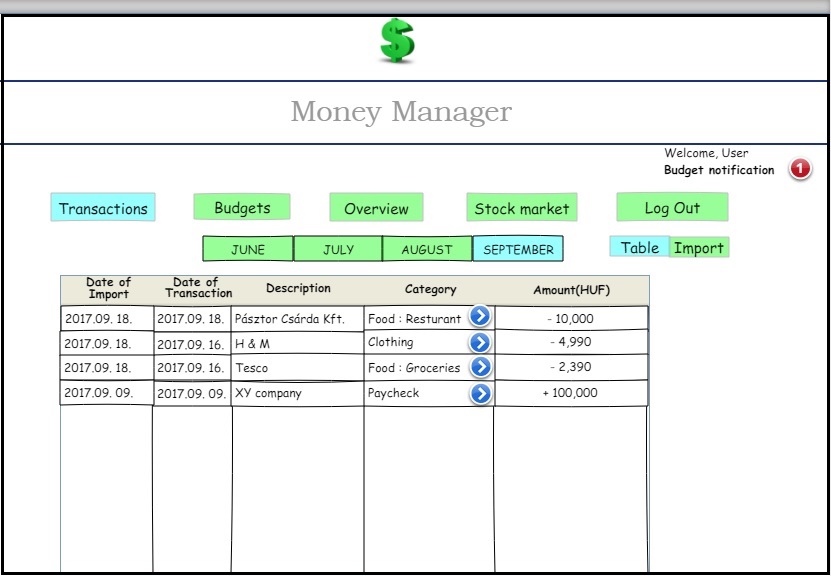
* Tranzakció dátuma: a tranzakció végbemenetelének napja.
* Termék neve: Azonosítja a céget amely a részvények kibocsájtásáért felel.
* Ár: a részvény akkori árát jeleníti meg amikor megvásároltunk vagy eladtuk.
* Tranzakció típusa: a részvény típusa, két értéket vehet fel, eladott vagy vásárolt.
* Mennyiség: megjeleníti az adott áron vett vagy eladott részvény mennyiségét.

A táblázat felett elhelyezkedik egy legördülő lista amely tartalmazza az összes különböző cégekhez tartalmazó részvényt, amelyeket a felhasználó által beimportált fájlból olvas ki a szoftver.

Táblázat és Importálás néven. Alapértelmezetten a Táblázat gomb aktív. Aminek hatására a Tranzakciók menüpont megjelenít egy táblázatot, amely tartalmazza a felhasználó által beimportált fájlból kiolvasott tranzakciókat, a táblázat tartalmaz minden fontos információt a tranzakciókat illetően.

A táblázat öt oszloppal rendelkezik:

* Importálás dátuma: A tranzakció importálásának ideje (ÉÉÉÉ. HH. NN) például 2017.11.11
* Tranzakció dátuma: A tranzakció „végbemenetelének” ideje (ÉÉÉÉ. HH. NN) például 2017.11.11
* Leírás: Tartalmazza a tranzakció közleményét (vásárlás eseten ez a cég neve ahol vásároltunk) vagy az ellenoldali nevet (például fizetés megkapása esetén a cég neve ahol dolgozunk).
* Kategória: Itt a felhasználó beállíthatja milyen típusú volt az adott tranzakció egy legördülő listából. Alapértelmezett érték szerepel ebben a cellában, ha a felhasználó nem állított be értéket.
* Érték: Tartalmazza a tranzakció értékét (Forintban). Ha vásárlunk például -5,000. Ha fizetésünk jön +5,000



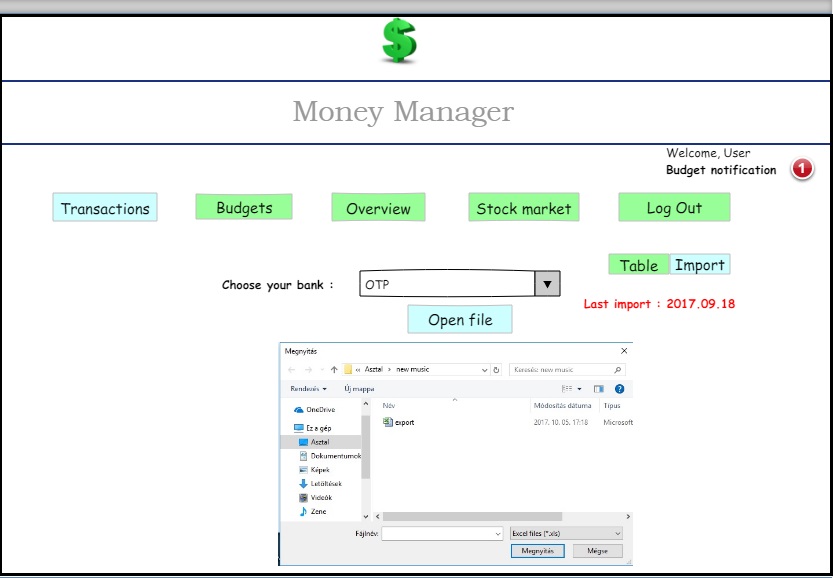
Ábra: Tranzakció menüpont Táblázat menüpontja

Ha a felhasználó be szeretné állítani a Kategória cella értékét az ő általa felvitt „Pénztárca” típusokat választhatja, vagy ha nem szerepel köztük a megfelelő kategória hozzá is adhat új kategóriákat.

Fontos, hogy a bizonyos Kategóriák több alkategóriára oszthatóak. Például az Ételnek lehetnek olyan alkategóriái például:

* Étterem
* Bevásárlás.

Az Importálás gomb konkrétan az alkalmazás fő eleme. Ami azt a célt szolgálja, hogy a felhasználó (saját bankjától) letöltött tranzakciós adatokat tartalmazó Excel fájlt feldolgozza és felvigye a táblázatba illetve a rendszer adatbázisába. Ezáltal tudunk felvinni új tranzakciókat az alkalmazásba. Ha rákattintunk az Importálás gombra. Egy legördülő lista fog megjelenni ahol ki kell választanunk melyik Banktól szeretnénk tranzakciókat importálni.



Ábra: Tranzakció menüpont Importálás funkciója

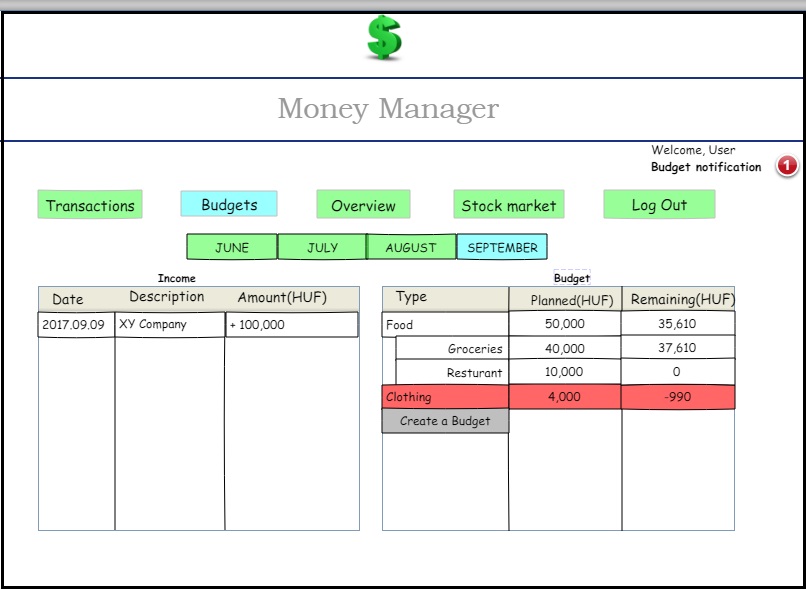
Miután kiválasztottunk bankunkat megjelenik egy Fájl megnyitása gomb, ezt megnyomva kiválaszthatjuk sajátgépünkről a beimportálni kívánt Excel fájlt. A sikeres kiválasztás után a rendszer feldolgozza a fájlban lévő adatokat, amiket elment az adatbázisába is.

A rendszer kiszűri azokat az importálni kívánt tranzakciókat, amik már szerepelnek az adatbázisban, ezzel megoldva a redundáns adattárolás problémáját. Ellenkező esetben a felhasználó hibásan követné nyomon a személyes pénzügyeit. Az ismétlődő tranzakciókról az alkalmazás értesítést is küld a felhasználónak (Messagebox), illetve az értesítésben láthatjuk az új, sikeresen elmentett tranzakciók számát is.

Az Importálás gomb alatt megjelenítésre kerül az utolsó importálás dátuma is, de csak abban az esetben, ha beszúrásra került egy új tranzakció.

## Pénztárca

A Pénztárca menüpont a jobb átláthatóság kedvéért két táblázatot tartalmaz. Míg az egyik táblázat a számlánkra bejövő pénzt tartalmazza (Fizetés táblázat) a másik táblázat a felhasználó által beállított „Pénztárcát” és annak jelenlegi állását tartalmazza (Pénztárca táblázat).



Ábra: Pénztárca menüpont két táblázata

A Fizetés táblázat: tartalmazza azokat a tranzakciókat az aktuálisan kiválasztott hónapból ahol a felhasználónk számlájára pénz érkezett. A táblázat tartalmaz három oszlopot:

* Tranzakció dátuma: A tranzakció „végbemenetelének” ideje (ÉÉÉÉ. HH. NN) például 2017.11.11
* Leírás: Tartalmazza a tranzakció közleményét (vásárlás eseten ez a cég neve ahol vásároltunk) vagy az ellenoldali nevet (például fizetés megkapása esetén a cég neve ahol dolgozunk).
* Érték: Tartalmazza a tranzakció értékét (Forintban). Ha fizetésünk jön például +5,000.

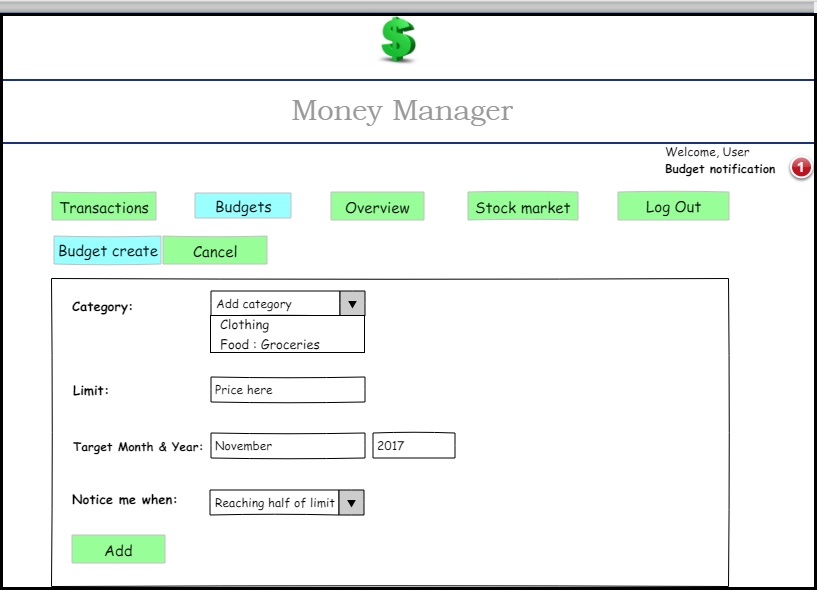
A Pénztárca táblázat: teljes mértékben a felhasználó készíti el és a neki megfelelő értékekkel töltheti fel. A Pénztárca táblázat három oszloppal rendelkezik:

* Típus: A „Pénztárca” típusa, ebben az oszlopban szerepelhetnek főkategóriák és alkategóriák is. Például Autó a főkategória az alkategóriák pedig üzemanyag és tisztítás.
* Tervezett: Ez az oszlop az aktuális kategória tervezett maximális költését tartalmazza forintban. A költés értékét teljes mértékben a felhasználó tölti ki és szerkesztheti kedve szerint. A felhasználó nem tud ebbe az oszlopba negatív értéket beírni, ha megpróbálja az alkalmazásértesítést küld.
* Még rendelkezésre áll: A még rendelkezésre álló oszlop tartalmazza az aktuális kategóriára beállított még rendelkezésre álló pénzösszeget forintban. Ez az oszlop a rendszer által kiszámított értéket tartalmazza, tehát a felhasználó nem szerkesztheti. A felhasználó által beállított Tervezett összegből levonja azoknak a tranzakcióknak az összegét, amelyeknek megegyezik a kategóriája az aktuális kategóriával.

Ha a felhasználó túllépi az általa megszabott korlátot az aktuális sor pirosra színeződik és a jobb felső sarokban frissül a felhasználó értesítéseinek száma, vagy ha közel van a korlát átlépéséhez színváltozás következik be (zöldről-sárgára). A zöld szín azt az esetet jelöli, amikor a felhasználó egyik pénztárca korlátot sem lépte túl és még közel sincs hozzá.

A Pénztárca menüpont még tartalmaz négy gombot a táblázatok felett ahol az aktuális illetve az ezelőtti hónapok szerepelnek, ezen gombok által a felhasználó meg tudja nézni a múlt havi pénztárcáit is. Alapértelmezés szerint az oldal mindig az aktuális hónapot választja ki.

Valamint a táblázat tartalmaz egy Új Pénztárca hozzáadása gombot, amire ha a felhasználó rákattint, egy új oldal jelenik meg ahol új Pénztárca korlátokkal tudja bővíteni a táblázatát. Egy új Pénztárca létrehozásánál a felhasználó megadhatja a tetszőleges kategóriát egy legördülő listából, a limitet, hogy maximálisan mennyit költhetünk az adott kategóriában, illetve az időpontot hogy meddig legyen aktív az adott Pénztárca, valamint egy legördülő listából választhatunk, hogy mikor értesítsen minket a rendszer, például mikor elértük a limitünk felét. Ha a Pénztárca kategóriánál a legördülő listából a kategória hozzáadását választjuk lehetőségünk van új kategóriákkal bővítjük rendszerünket.



Ábra: Új Pénztárca hozzáadása

Ha a felhasználó túllépi az általa megszabott korlátot a Pénztárca táblázat aktuális sora pirosra színeződik és a jobb felső sarokban frissül a felhasználó értesítéseinek száma, vagy ha közel van a korlát átlépéséhez színváltozás következik be az értesítés színe szempontjából zöldről-sárgára. (A zöld szín azt az esetet jelöli, amikor a felhasználó egyik pénztárca korlátot sem lépte túl és még közel sincs hozzá.)

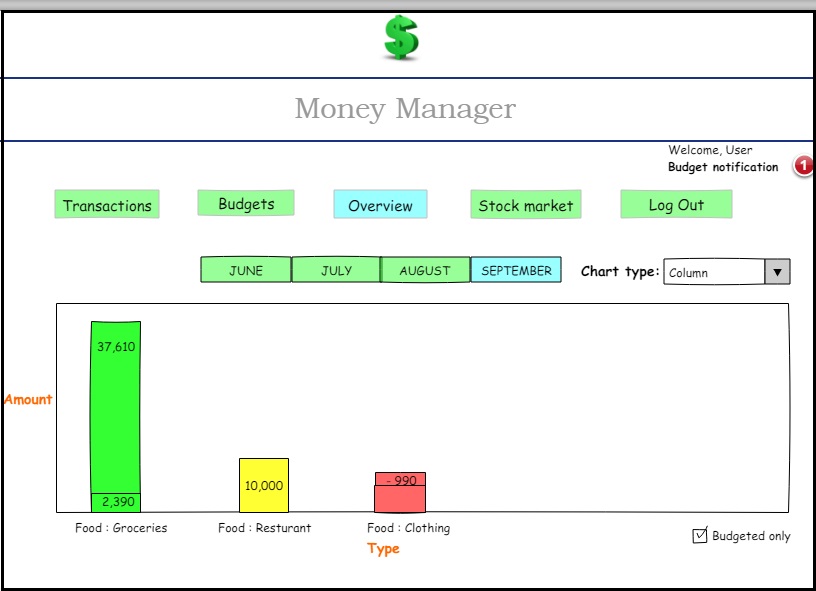
A Pénztárca menüpont még tartalmaz négy gombot a táblázatok felett ahol az aktuális illetve az ezelőtti hónapok szerepelnek, ezen gombok által a felhasználó meg tudja nézni a múlt havi pénztárcáit is. Alapértelmezés szerint az oldal mindig az aktuális hónapot választja ki..

## Áttekintés

Az áttekintés menüpont tartalmaz egy grafikont, amiben szerepel a felhasználó által beállított „Pénztárca” jelenlegi helyzete az adott hónapban.

A felhasználó a grafikon felett láthat négy gombot, amik a hónapokat jelölik. Ezekre rákattintva válthat bizonyos hónapok között és nézheti meg a pénztárcák alakulását. Alapértelmezés szerint az oldal mindig az aktuális hónapot választja ki. Ezzel is segítve a pénzügyeink alakulásainak nyomon követését. Meg tudja nézni mire költött a legtöbbet, milyen kategóriában lépte túl az általa megszabott keretet, mire költött a legkevesebbet, mennyi pénzt spórolt az adott hónapban.

Továbbá a táblázatok fölött láthatunk egy legördülő listát ahol a megjelenített grafikon stílusát választhatjuk ki.



Ábra: Áttekintés menüpont oszlopos grafikon megjelenítés

A fenti ábrán láthatjuk, hogy jelen esetben az oszlopos megjelenítés van kiválasztva, vízszintes tengelyen láthatjuk a bizonyos oszlopokban megjelenített pénztárcák típusát, valamint a függőleges tengelyen láthatjuk a költött pénz mennyiségét.

Továbbá a grafikont tartalmazó táblázat alatt láthatunk egy jelölőnégyzetet, ami jelen esetben be is van pipálva, ezáltal a mi esetünkben csak azok a költések jelennek, meg amikre pénztárca korlát is szabva van. Ha ez a funkció nincs kipipálva az összes tranzakció, költés, jövedelem megjelenik a táblázatunkban.

Az oszlopok négy színt vehetnek fel a pénztárca jelenlegi helyzete alapján:

* Piros: akkor, ha a felhasználó túlköltött az általa létrehozott korláton, ha a felhasználó az oszlopra húzza, a kurzort láthatja mennyivel költött többet a kelleténél.
* Sárga: akkor az azt jelenti, hogy a felhasználó pontosan annyit költött az adott hónapban, mint amennyit beállított magának az adott kategóriára.
* Zöld: akkor a felhasználó az adott kategóriában még nem érte el az általa megszabott korlátot, annak csak egy bizonyos részét használta fel.
* Fehér: akkor a felhasználó az adott kategóriában még nem költött az aktuális hónapban.

## Részvények

A részvények menüpontban a felhasználó két opció közül választhat:

* A rendszer listázza ki a Vásárolt részvényeket.

Vagy

* A rendszer listázza ki az Eladott részvényeket.

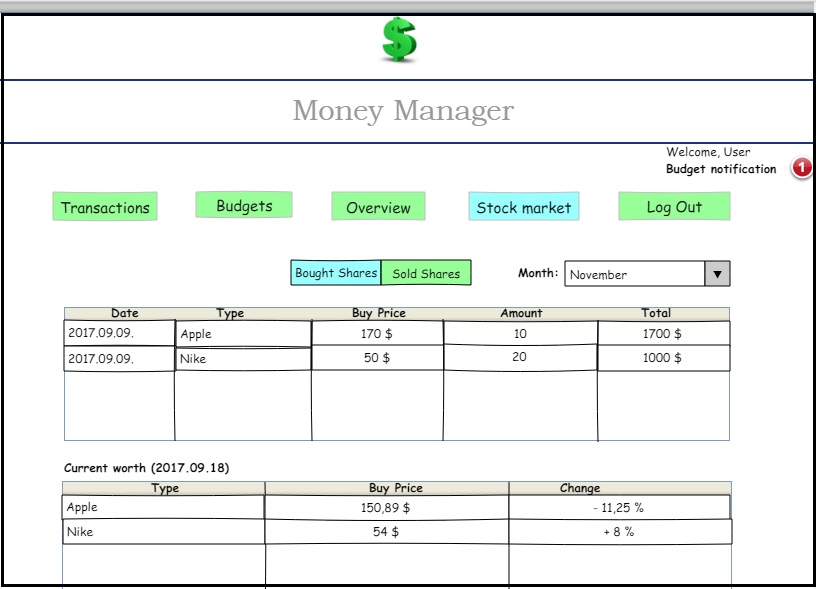
A Vásárolt részvények gombra kattintva a felhasználónak két táblázat fog megjelenni. Az felső táblázat tartalmazni fogja (Vásárolt részvények) a részvényeket, amiket vásárolt. Méghozzá öt oszlopban:

* Dátum: Mikor vásárolta a részvényeket. (ÉÉÉÉ. HH. NN például 2017.09.31)
* Típus: A vásárolt részvény neve, milyen cég részvényei. (például IBM, Apple, Adidas)
* Vételár: A részvény akkori árfolyama, amikor a felhasználó vásárolta őket (dollárban).
* Mennyiség: Hány darab részvényt vásárolt a felhasználó.
* Végösszeg: Az akkori árfolyam és a darabszám szorzata a végösszeg.

Az alsó táblázat tartalmazza az aktuális napi részvényárfolyamokat, méghozzá azon részvényeket, amiket a felhasználó is vásárolt. E táblázat tartalmazni fogja az aktuális árfolyamot és a százalékos változást a felhasználó által vásárolt részvények árfolyama között.

* Típus: A vásárolt részvény neve, milyen cég részvényei. (például IBM, Apple, Adidas)
* Vételár: A részvény akkori árfolyama, amikor a felhasználó vásárolta őket (dollárban).
* Változás: A részvény vásárlási árfolyamához képest milyen százalékos eltérése van az aktuális árfolyamhoz képest.

Nem utolsó sorban láthatunk a Vásárolt részvények táblázat fölött egy legördülő listát, aminek segítségével ki tudjuk, választani melyik hónap adatait szeretnénk kilistázni.



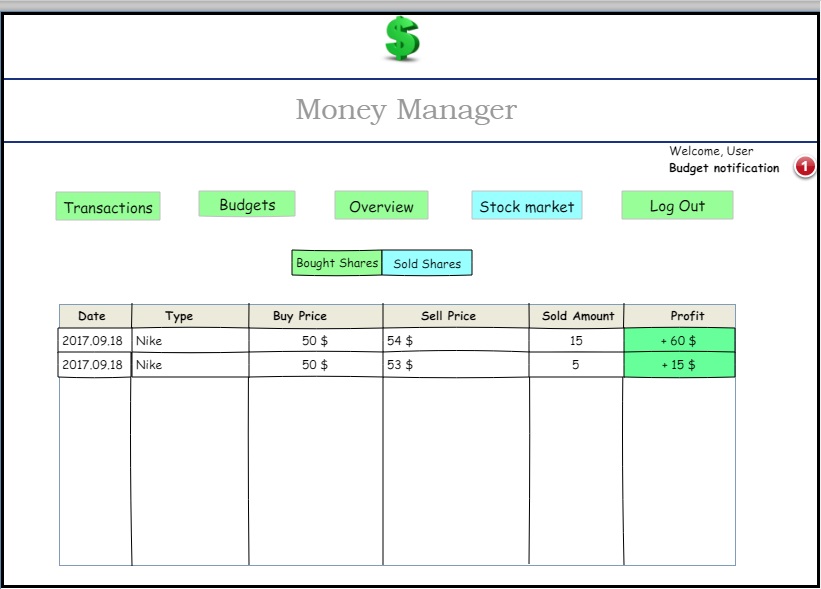
Ábra: Részvények menüpont, Vásárolt részvények

Ha a felhasználó az Eladott részvények gombra kattint egy hatoszlopú táblázat fog megjelenni:

* Dátum: A részvények eladási dátuma. (ÉÉÉÉ. HH. NN például 2017.09.31)
* Típus: Az eladott részvény neve, milyen cég részvényei. (például IBM, Apple, Adidas)
* Vételár: A részvény akkori árfolyama, amikor a felhasználó vásárolta őket (dollárban).
* Eladási ár: A részvény akkori árfolyama, amikor a felhasználó eladta őket (dollárban).
* Darabszám: Azon eladott részvények száma, amelyek ugyan olyan Eladási áron keltek el.
* Profit: A részvény vételárához képest mennyi a nyereség az eladás után (dollárban).

A profit oszlop elemei három színt vehetnek fel:

* Piros: Ha a vételi árnál olcsóbban adtuk el a részvényeket, tehát veszteség keletkezett.
* Sárga: Ha a vételi árral azonos értékben adtuk el a részvényeket.
* Zöld: Ha a vételi árnál drágábban adtuk el a részvényeket, tehát nyereség keletkezett



Ábra: Részvények menüpont, Eladott részvények

## Kilépés

A felhasználó a kilépés menüpontra kattintva kijelentkezik az oldalról és újra a Kezdőlap oldal fog megjelenni.

# Használt technológiák

## .NET keretrendszer

Habár .NET keretrendszer nem új technológia, azonban a felhasználók közül nem sokan tudják, mire jó, vagy mire használatos. Ezen rövid leírásomban bemutatom, hogy mire is jó. Ezen felül szót ejtek olyan témákról is, amelyek a járatosabb felhasználók számára sem egyértelműek.

A Microsoft.NET rendszer többféle programozási nyelven is programozható - feltéve, ha az adott programozási nyelvhez van olyan fordítóprogram, amely alkalmazza a Common Language Specification elveit és ajánlásait, és képes a Microsoft.NET rendszer virtuális kódjára fordítani.

A Common Language Specification olyan szabályok halmaza, melyek leírják azokat az alapvető normákat, amelyre a programozási nyelveknek illeszkedniük kell. Mivel a Microsoft.NET nem mindegy, hogy milyen nyelven programozunk.

Sok ilyen programozási nyelv létezik. A Microsoft által direkt támogatott nyelvek a Visual Basic, J# , C#, Visual C++.

Mivel a Microsoft.NET rendszerben mindegy, hogy milyen nyelven programozunk, egyik programozási nyelven megírt eljárást a másik nyelvből is meg lehet hívni és lehet használni. Fontos hogy ez a funkció csak abban az esetben valósítható meg, ha a nyelvek egymással is megegyeznek bizonyos alapvető elvekben. Ilyen elvek közé tartozik a tömbök és rekordok, ábrázolása és használata és az alapvető típusok (Integer, Double, String, stb.) reprezentációja a memóriában.

 Common Language Runtime a .NET Keretrendszer része, ami lehetővé teszi a Common Intermediate Language nevű kód futtatását különböző platformokon, például Windowson,  Linuxon, Macintoshon,  és Unixon is. Common Language Runtime réteg gyakorlatilag egy processzor-szimulátor. Amely az általa kezelt virtuális mikroprocesszor virtuális gépi kódjára lefordított programokat képes futtatni. Részét képezi egy Just In Time Compiler is, amely a futó programot 'röptében', a memóriában átfordítja az adott számítógép konkrét mikroprocesszorának konkrét gépi kódjára, így a virtuális gépi kódfutás közben átalakul gépi kódra. Továbbá részét képezi egy jogosultsági rendszert kezelő rendszer, amely eldönti, hogy a program kódjának milyen tevékenységeket szabad végrehajtania. Valamint hozzátartozik a Garbage Collector (szemétgyűjtő) is, amely a nem használt memóriaterületeket felfedezi és felszabadítja,

## C# Programozási nyelv

A C# az a programozási nyelv, ami a legközvetlenebb módon tükrözi az alatta működő, minden .NET programot futtató .NET keretrendszert. A primitív adattípusai az objektumok. Szemétgyűjtést használ, valamint az absztrakcióinak többsége (osztályok, interfészek, delegáltak, kivételek…) a .NET futtatórendszert használja közvetlen módon.

A C vagy C++ nyelvhez hasonlítva a C# több korlátozást és továbbfejlesztést is tartalmaz. Például:

* A mutatók és a nem ellenőrzött aritmetika csak egy speciális, nem biztonságos módban használhatók. A legtöbb objektum-hozzáférés csak biztonságos hivatkozásokon keresztül tehető meg.
* Az objektumok nem szabadíthatók fel közvetlen módon, ehelyett a szemétgyűjtő szabadítja fel őket, mikor már nincs rájuk hivatkozás. Ez a módszer kizárja a nem létező objektumokra való hivatkozás lehetőségét (C , C++ nyelvek hiányossága)..
* A nyelv csak egyszeres öröklődést támogat, de egy osztály több interfészt is megvalósíthat.

A C# sokkal típus biztosabb, mint a C++. Az egyetlen implicit konverzió a biztonságos konverzió, úgy, mint az egészek tágabb intervallumba konvertálása vagy a leszármazott osztályok alaposztályba konvertálása. Nincs implicit konverzió az egészek és a logikai típus (boolean) között, a felsorolás tagok és az egészek között. Nincsenek void mutatók (bár az Object osztályra mutató mutatók hasonlóak), valamint bármely, a felhasználó által definiált implicit konverziót explicit módon meg kell jelölni.

### WPF (Windows Presentation Foundation)

C# nyelvi környezetben használhatunk Windows Form Designer (röviden WinForm) vagy Windows Presentation Foundation ( röviden WPF) dizájnereket hogy gyorsan és könnyedén készíthessünk grafikus felületet.

A fő különbség a két felület között hogy a WinForm egy olyan függ a Windowstól, viszont a WPF et akár a semmiből is felépíthetjük és csak nagyon kevés helyzetben függ Windowstól. Egy jó példa erre a különbségre egy Gomb elem készítése, amiben van egy kép és egy TextBox is, ez nem egy alapértelmezett Windows funkció igy a WinForm nem ajánlja fel nekünk ezt a lehetőséget egyből. Tehát nekünk kell implementálnunk azt ezt megvalósító metódust, ami a következő lépésekből áll Meg kell szerkesztenünk a képet, létre kell hoznunk egy olyan gombot, ami támogatja a képek megjelenítését vagy importálunk kell egy harmadik féltől származó kiegészítést. Ellenben a WPF felületen egy Gomb komponens tartalmazhat bármit, mivel ez az elem végülis egy keret, ami bizonyos állapotokkal rendelkezhet (például érintetlen,kattintott, egér alatt helyezkedik el). Ez a tulajdonság nem csak a Gombról mondható el, hanem az összes WPF-ben kreálható elemről is. Ha az akarjuk megoldani WPF felületen, hogy egy Gomb tartalmazzon egy képet és egy TextBoxot nincs más dolgunk, mint a Gomb tartalmába beilleszteni ezeket az elemeket. A WinForm ilyen szintű "kényelmet" nem nyújt a felhasználó számára. Viszont ennek a kötetlenségnek és a semmiből való "felépítésnek" az a hátránya hogy bizonyos esetekben több időt kell fordítanunk valamire, hogy ugyan azt az eredményt elérjük, amit esetleg könnyű megvalósítani a WinForm Windows által korlátozott elemein.

Az XAML az XML nyelv egy változata, amit a Microsoft fejlesztett ki a GUI ( Grafikus Felhasználó Felület) leírására. A WinForm vagy korábbi GUI részei ugyan azon a nyelven íródtak mint maga a program ami a megjelenítés mögött van például C#,vB.NET és a dizájnerek felügyelik, viszont az XAML esetén külön van választva a megjelenítés és a programkód.

WPF előnyei:

* Újabb, tehát jobban szinkronban van az új technológiákkal.
* A Microsoft rengetek alkalmazáshoz használja pl. Visual Studio
* Rugalmas, több dolgot csinálhatunk anélkül, hogy egyéb dolgokat is implementálnunk kell.
* Ha harmadik féltől származó elemet importálunk nagy valószínűséggel támogatni fogja mivel újabb technológia.
* XAML segítségével könnyen el tudjuk különíteni a megjelenítését és a programozás részét a programunknak.
* Dabinding, ami segítségével jobban el tudjuk választani a kódot és a megjelenítést.
* Hardveres gyorsítást használ a GUI kirajzolására , jobb teljesítmény.
* Lehetővé teszi a Felhasználó Felületek készítését Windowsos szoftverekre és Web applikációkra is (Silverlight/XBAP).

WinForm előnyei:

* Régebbi technológia , ezáltal többet tesztelték és próbálták.
* Sok harmadik féltől származó kiegészítés létezik, amit ingyen letölthetünk.
* A Visual Studio jelenleg is a WinForm-ot támogatja inkább. Tehát WPF-ben többet kell dolgozni bizonyos esetekben ugyan azért az eredményért.

## Databinding

A Databinding egy olyan technológia, ami veszi az adat forrását az adat küldőjétől és a vevőjétől és szinkronizálja őket. A WPF (Windows Presentation Foundation) felületen a Databinding egyszerű és konzisztens útja az adatok megjelenítésének és változtatásának.



Ábra: Databinding fajtái WPF esetén

Egyszeri: (Az ábrán nincs illusztrálva) Az adat forrás irányból megváltozhat, de a későbbi változtatások nem jutnak vissza a forráshoz. Ami azt jelenti, ha a cél adat megváltozik , arról a forrás nem fog tudomást szerezni. Ez a módszer akkor használatos, amikor szükségünk van a jelenlegi állapotra vagy az adatok ténylegesen statikusak.

Egy irányú: (Az Ábrán OneWay) Az egy irányú összerendelés a cél tulajdonság megváltoztatását eredményezi, ha megváltoztatjuk a forrás adatot, de a célobjektum tulajdonságainak módosítása nem terjed vissza a forrás tulajdonságra. Ez a fajta adat összekötés akkor megfelelő, ha az ellenőrzött ellenőrzés implicit módon csak olvasható. Például kötődhet egy olyan forráshoz, mint például egy tőzsdei figyelő vagy esetleg a célobjektumhoz nincs olyan vezérlő felület, amely lehetővé tenné a változtatásokat, például az asztal háttérszínét vezérlő adatok. Ha nem szükséges figyelni a céladat változásait, akkor a OneWay kötési mód használatával elkerülhető a TwoWay kötési mód túlterhelése.

Kétirányú: (Az ábrán TwoWay) összerendelés megváltoztatja a forrásadatot vagy a céladatot a másik frissítéséhez. Ez a fajta kötés alkalmas a szerkeszthető formák vagy egyéb, teljesen interaktív felhasználói felület forgatókönyvekhez. A legtöbb tulajdonság alapértelmezett a OneWay összerendeléshez, de bizonyos függőségi tulajdonságok (jellemzően a felhasználó által szerkeszthető vezérlők, például a TextBox Szöveg tulajdonsága és a CheckBox IsChecked tulajdonsága) alapértelmezetten TwoWay összerendelés. Egy programszerű módszer annak meghatározására, hogy egy függőségi tulajdonság alapértelmezés szerint egyirányú vagy kétirányú-e, az GetMetadata használatával megkapja az tulajdonság metaadatait, majd ellenőrizze a BindsTwoWayByDefault tulajdonság Boolean értékét egy irányú a forrás felé: ahol a felhasználó csak írási joggal rendelkezik.

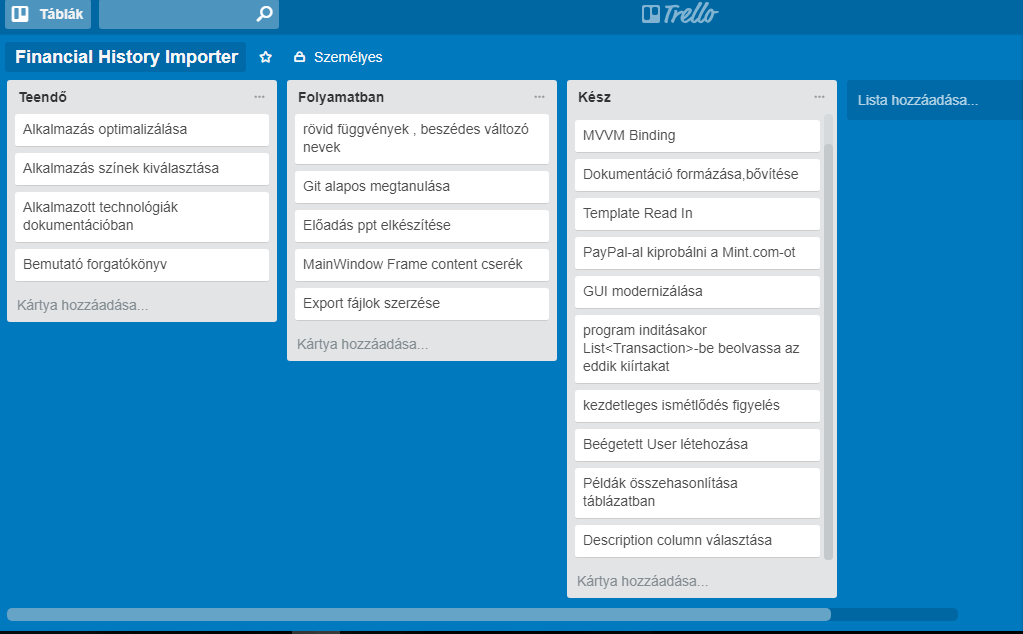
## Trello

A Trello egy web alapú projektmenedzsment alkalmazás, aminek segítségével nyomon tudjuk követni, hogy állunk a számunkra kitűzött célok elérésében. A Trello tábla alapvetően egy weboldal, amely az oldalakon vízszintesen felsorolt ​​listákat tartalmaz, így egy madártávlatos képet kaphat a projektről. A listákon belüli, kártyákkal ellátott elemek húzhatók és más listákra visszahúzhatók vagy listákon belül rendezhetők.

Az egyes kártyák magukban tartalmazhatnak ellenőrző listákat, képeket, csatolmányokat, határidő dátumokat, színes címkéket és vitafelvételeket másoktól, akik osztoznak a fórumon. Annyi táblát csinálhatunk, amennyit csak szeretnénk - az egyik például "Otthoni teendők" ,a másik "Szakdolgozat" és így tovább.

Trello kártyák olyanok, mint a ragadós jegyzetek, amelyeket egy parafa fedélzeten rendeznek - azaz olyan digitális ragadós jegyzetek, amelyek kereshetők, megoszthatók, és emlékeztetők. Kártyákat is létrehozhat e-mailben.

Ha több fős a project csapatunk akkor különböző emberekhez rendelhetünk bizonyos részfeladatokat, erről ők is értesítést kapnak, ha kész vannak áthúzhatják a kártyát a kész oszlopban igy jelezve a többiekkel hogy készen van a feladatával.



Ábra: Trello folyamatok állapotának áttekintése

## GitHub

A GitHub megértéséhez először meg kell értenünk a Git-et. A Git egy nyílt forrású verzióvezérlő rendszer, amelyet Linus Trovalds indított - ugyanaz a személy, aki Linuxot hozott létre. A Git hasonlít a többi verzióvezérlő rendszerhez - a Subversion, a CVS és a Mercurial, hogy csak néhányat említsünk.

Tehát a Git egy verziókezelő rendszer, de mit jelent ez? Amikor a fejlesztők valamit hoznak létre (például egy alkalmazást), folyamatosan módosítják a kódot, új verziók kiadása az első hivatalos (nem béta) kiadásig és után.

A verzióellenőrző rendszerek ezeket a módosításokat nyomon követik, a módosításokat egy központi adattárban tárolják. Ez lehetővé teszi a fejlesztők számára, hogy könnyedén működjenek együtt, letölthetik a szoftver új verzióját az módosíthatják és feltölthetik a legújabb verziót. Minden fejlesztő láthatja ezeket a változásokat, letöltheti őket és esetleg hozzá is járulhat.

Hasonlóképpen azok az emberek, akiknek semmi köze a projekt fejlesztéséhez, továbbra is letölthetik és felhasználhatják a fájlokat. A legtöbb Linux-felhasználónak ismernie kell ezt a folyamatot, mivel a Git, a Subversion vagy más hasonló módszer használata elég gyakori a szükséges fájlok letöltéséhez - különösen egy program forráskódból történő összeállításának előkészítésében.

A Git a legtöbb fejlesztő számára előnyben részesített verzióvezérlő rendszer, mivel számos előnye van a többi rendelkezésre álló rendszerhez képest. Fájlváltozásokat hatékonyabban tárolja, és jobb fájlintegritást biztosít.

Megállapítottuk, hogy a Git verziókezelő rendszer, hasonló, de jobb, mint a rendelkezésre álló alternatívák. Tehát mi teszi a GitHub-ot annyira különlegesnek? A Git egy parancssori eszköz, de a központ, amely körül minden olyan dolgot, amely a Git-et forgatja, az a hub-GitHub.com, ahol a fejlesztők hasonlóan gondolkodó emberekkel tárolják projektjeiket és hálózatukat.

## Pencil mockup

A Pencil nevű mockup lehetővé teszi, hogy gyorsan és egyszerűen hozzunk létre szórakoztató és interaktív grafikus felhasználói felületet maketteket. Az alkalmazás rengeteg beépített elemmel rendelkezik és nagyon könnyű kezelni. Valamint egy egyszerű "behúzós" felülettel rendelkezik. Mivel cross-platform és teljes offline támogatást nyújt, bárhol létrehozhatunk maketteket. Nagy előnye ennek a szoftvernek, hogy teljesen ingyenes. A Pencil természetesen engedi, hogy saját képeinket illetve egyéb elemeinket is beimportáljuk.

# Jövőbeli tervek

A Kiolvasó algoritmus optimalizálása illetve gyorsítása, hatékonyság növelése illetve sok számlakivonat típuson történő tesztelés.

Tőzsdei adatok importálási lehetősége a programba. Ami tartalmazza az eladott illetve vásárolt részvények adatait. Valamint a nyereséget és a veszteséget, illetve a jelenlegi árfolyamokat is láthatják majd a felhasználók.

Tranzakciók dátum szerint történő szűrése illetve a kiválasztott hónap alapján történő megjelenítése az Adatbázis menüpontban a felhasználó számára, amiket gombokkal tud kiválasztani.

Grafikonok készítése a felhasználó pénzügyi szokásairól a kiválasztott hónapban , bizonyos keresési feltételeket figyelembe véve.

**Végső cél:** Mivel rengeteg bank van Magyarországon és az exportált Excel fájlok formátuma mindegyiknél különböző, ezért a fő jövőbeli cél egy olyan beolvasó algoritmus írása, amelyik banktól függetlenül be tudja olvasni az exportált Excel fájlok tartalmát. Ezzel megspórolva azt a hosszadalmas munkát ahol mindegyik Banktól exportált Excel fájlokat különböző algoritmusokat kelljen be olvasni.