

StockMaster – Tőzsdei platform

Specifikáció





Tartalomjegyzék

1	Projektleírás	4
1.1.	Célfelhasználók.....	5
1.2.	Használat.....	5
2	Általános leírás.....	6
2.1.	Rendszeresen, automatizáltan töltött adatok	6
2.2.	Kézi adatbevitellel töltött adatok.....	6
2.3.	Esetileg, egyedi feldolgozással töltött adatok	6
2.4.	Adatfeldolgozás	6
2.5.	Felhasználás	7
2.6.	Háttérben frissülő adatok.....	7
2.7.	Operációs rendszer, környezet.....	8
2.8.	Felhasználandó programozási nyelvek.....	8
2.9.	Megoldás formátuma	8
2.10.	Rendszer és gépigény	8
3	Szoftverfejlesztés	9
3.1.	A feladat.....	9
3.2.	Modulok.....	9
4	Szoftver specifikáció.....	10
4.1.	A szoftver.....	10
4.2.	Megjelenés	10
4.3.	Funkciók	10
5	Dokumentáció	11
5.1.	Technikai dokumentáció.....	11
5.2.	Forráskód dokumentáció	11
5.3.	Felhasználói dokumentáció	11
6	A projekt tesztelése és értékelése	12
6.1.	A feladat felhasználó oldali szempontjai.....	12



Specifikáció

6.2.	A feladat technikai szempontjai	12
6.3.	Projekt megvalósításának piaci jellegű értékelése	12
7	Projekt adatlap	13



1 Projekteírás

A projekt neve: StockMaster — egy felhasználóbarát és oktató jellegű alkalmazás, amely kifejezetten a kezdő tőzsdei kereskedők számára készült. Célja, hogy átfogó, mégis könnyen követhető módon vezesse be őket a részvénypiac izgalmas és összetett világába, megismertetve őket az árfolyamok mozgásával, a piaci folyamatokkal és a kereskedési stratégiák alapjaival.

A StockMaster nem csak egy egyszerű kereskedési platform, hanem egy olyan interaktív tanulási környezet, amely lehetővé teszi a felhasználóknak, hogy lépésről lépésre, saját tempójukban ismerkedjenek meg a piac működésével, így biztosítva az **elméleti tudás** és a **gyakorlati tapasztalat** együttes megszerzését.

Stratégiai kulcsok kiépítése a felületen vagyis felnyitható PDF-ek (oktató anyagok beágyazása a menürendszerben). Rendszeres **piaci riportok feltüntetése** a részvény ablakok alatt, ahol az **aktuális pozícióhoz tartozó** hírek és információk ugranak fel. A program implementáljon API-kat (*Application Programming Interface*), melyek **képesek ábrán vizualizálni a termékek ármozgását** egy meghatározott ablakban. Az ablak alatt legyen egy **buy** (vétel) és **sell** (eladás) gomb, melyben a részvény eladási és vételi ára is legyen megjelenítve.

Az ablak alján húzódó **sávon be lehessen állítani** a diagram **gyertyák időközét** (hogy milyen időszávot foglaljon el egy gyertya kimutatása) 1 perces, 5 perces, 15 perces 1 órás, 4 órás, 1 napos, 1 hetes állítási skálában.

Továbbá a **pozíciókat tartó felhasználók tartási átlagát is jelenítse meg egy csúszkán**, (eladó/vevő) %-os arányban mutatva. A felület tartalmazzon **két színt**, ez a két szín a **piros** és a **zöld** legyen. A **piros a lefelé mutató és haladó (bearish)** gyertyákat jelölje és az eladás színe is ez legyen. A **zöld és a felfelé mutató és haladó (bullish)** gyertyákat jelölje és a vétel színe is ez legyen. A csúszka is osztozzon a két színen az éppen aktuális %-os arányban.

A megnyitás pillanatában a bal felső sarokban helyezkedjen el. A menürendszer úgy épüljön fel, hogy könnyedén eltudjon kalauzolni a beállításokig, ahol a felhasználó igényeit lehessen beállítani.

- Sötét vagy világos mód
- jelszóval legyen a bejelentkezés vagy automatikusan maradjon bejelentkezve első bejelentkezés után,
- küldjön-e értesítéseket az alkalmazás az üzenő falára
- kártyaadatok kezelése, módosítása
- kijelentkezés



1.1. Célfelhasználók

A projekt fő célja, hogy egy **virtuális, valós piaci adatokra épülő kereskedési szimulátort** hozzon létre, ahol a felhasználók kizárolag virtuális pénzzel, kockázat nélkül próbálhatják ki magukat. A **modern pénzügyi piacokon** való eligazodás **nem egyszerű feladat**, és sok kezdő befektető esik abba a hibába, hogy tapasztalatlanságuk miatt jelentős pénzügyi veszteségeket szenvednek el.

A **StockMaster** ezt a problémát kívánja orvosolni, hiszen a felhasználók egy '**béta környezetben**', virtuális keretek között tanulhatnak a hibáikból, így **fokozatosan fejlesztve kereskedési tudásukat** és üzleti stratégiájukat. Ezzel a biztonságos tanulási folyamattal szeretnénk hozzájárulni ahhoz, hogy a későbbiekben minél több felhasználó **magabiztosan és sikeresen** mozogjon a valódi pénzügyi piacokon.

1.2. Használat

Azok a felhasználók, akik a tőzsdei **árfolyamokat figyelemmel kísérnek**, esetleg megakarnak ismerkedni a piaci mozgások rejtelméivel, tökéletes használati felületet biztosít a **StockMaster**.

A regisztrációt követően a felhasználót megilleti egy ingyenes '**demo**' számla, ahol **1 € - 10.000 €-ig** választhat kezdő összeget. Ezzel a kezdőösszeggel kezdheti a felhasználó a kereskedését, **ha esetleg elveszíti a kezdőösszeget bármikor újra beállíthatja** a kereskedésre szánt összeget.

A továbbiakban a felhasználó vásárolhat **részvényeket**, **CFD-ket** (*Contract for Difference*), **ETF-ket** (*Exchange-Traded Fund*) és **piaci index -ket** is. Sőt a menüpontok között megkeresheti az oktatóanyagot is, amely célzottan segít megismerni **a tőkebefektetés alapjait** a **befektetés lényegét**, **különböző stratégiáit** és **alapfogalmakat** is.

A felhasználót **biztosító árfolyam adatok folyamatosan frissülnek** a zavartalan és minőségi kereskedés biztosítása érdekében, ezáltal a felhasználó élő adatokkal képes használni a platformon megjelenő kereskedési eszközöket.

A **vétel és eladás gombokra kattintva** hajtja végre kereskedését. Elsősorban be kell állítani a kívánt mennyiséget egy kis ablakban. A **mennyiség mellett** kiírásra kerül a mennyiség költsége és hogy **elegendő fedezet áll-e rendelkezésre**. Ezt követően a **vétel vagy eladás gombra kattintva nyithatja meg pozíóját**.

A **vétel gomb megnyomásakor** a piac felfelé mozgásakor profitot termel, ellenkező esetben veszteséget. Az **eladás gomb megnyomásakor** a piac lefelé mozgásakor profitot termel, ellenkező esetben veszteséget.

A nyitott pozíció felületén egy **X-el tudja lezárni a pozíciót**, miután a felugró ablakos „*Biztos lezárja a pozíciót*” kérdésben a **biztosra** válaszol, ezáltal **realizálta a profitot** vagy **veszteséget** és az egyenlegéhez lesz hozzárendelve.



2 Általános leírás

2.1. Rendszeresen, automatizáltan töltött adatok

Az automatizált adattöltés az ár mozgásokat helyettesítő API-al történik folyamatosan, meghatározott időközönként. A **statisztikai adatok folyamatosan frissülnek** a kereskedés **profitja** vagy **vesztesége** összeadásával és kiszámításával. Az adatok nagy része folyamatosan helytálló, mert **megengedhetetlen** a **késés vagy csúszás** egy pénzpiaci rendszerben.

2.2. Kézi adatbevitellel töltött adatok

A **stratégiai mutatószámrendszer** kialakításához szükségesek olyan adatok is, amelyek nem származtathatóak közvetlenül **gazdálkodási** illetve **tanulmányi** rendszerekből, mivel azon adatok nem mindenkor, vagy egyetlen rendszerben sem kerülnek tárolásra. Az adatbevitelre egy **WEB**-es rendszert kell fejleszteni.

2.3. Esetileg, egyedi feldolgozással töltött adatok

Az adattárban lehet tárolni egyéb olyan adattartalmakat is, amelyek hasznos információtartalommal bírnak, ugyanakkor nem szükséges vagy nem lehetséges rendszeres gyűjtésük illetve betöltésük. Készülhetnek felmérések, egyedi adatgyűjtések, illetve **különböző részvények** rendelkezhetnek olyan **adatvagyonnal**, amely nagyban **segítheti a döntéshozatalt**. Ezen adatokat az adattárházon keresztül szabályozott módon lehet a felhasználók számára elérhetővé tenni.

2.4. Adatfeldolgozás

Az adatfeldolgozás főbb elemei:

- Adatkitöltés és formai ellenőrzés
- Kézi adatbevitel az oktatóanyagokhoz
- Adatfeldolgozások, adattranszformációk
- Adatellenőrzések
- Oktató anyagok illesztése
- Aggregálás
- Statisztikai számok előállítása



2.5. Felhasználás

A felhasználás első megközelítésben 5 területre koncentrál. Ez az öt terület az alábbi:

- Stratégiai kulcsok kiépítése;
- Rendszeres riportok;
- Hírek;
- Oktatóanyagok
- Jól átlátható diagrammok

2.6. Háttérben frissülő adatok

Az alkalmazás a háttérben folyamatosan, **élő adatokkal szolgálja ki a felhasználókat**. A háttérben futó **API-k** és árazási rendszerek együttes munkája végzi a felhasználó zavartalan kereskedését, naprakész információkkal. A **naptár rendszer** is folyamatosan **frissül**, amelyekben céghírek, várható nagy bejelentések például:

- **Kamatláb mozgások**
- **Piaci kitermelések**
- **Központi Banki információk**
- **Cég bejelentések**
- **Negyedéves, féléves és éves beszámolók**
- **Államkötvények aukciója**
- **Export/Import árindex**
- **Fogyasztói árindexek**
- **Fogyasztói kiskereskedelmi/ nagykereskedelmi indexek**

Várhatóak a naptári felsorolásban. Továbbá meglehet tekinteni **rákattintással** a közelgő eseményből származó és előző információkat korábbi tevékenységekhez mérve. Ezáltal a felhasználó **spekulálhat** és **dönthet**, hogy mi várható a **publikálás után**. Az ebből származó információ többlettel **pénzt** és **biztos pontokat** kereshet a piacon, korábbi tevékenységekhez mozgásokhoz viszonyítva. Ez előnyként szolgálhat, különösen akkor, ha nem régóta mozog a piacon.



2.7. Operációs rendszer, környezet

- Windows 10, 11,
- Mobiltelefon böngésző

2.8. Felhasználandó programozási nyelvek

- C#,
- SQL,
- HTML,
- CSS,
- Javascript

2.9. Megoldás formátuma

- C# forrásállomány
- Teljes projekt környezet
- Forráskód dokumentáció
- Technikai dokumentáció (pdf formátumban)
- Erőforrás-terv és munkaidő nyilvántartás
- Felhasználói dokumentáció

2.10. Rendszer és gépigény

Kliens oldal

- **Minimális:**

- Eszköz: középkategóriás Android / iOS telefon (3–4 GB RAM), vagy PC / laptop (2 magos CPU, 4 GB RAM).
- Kijelző: legalább 720p felbontás.

Szerver oldal

- **Minimális**

- CPU: 2 mag
- RAM: 2–4 GB
- Tárhely: 10–20 GB SSD, Operációs rendszer: Windows Server (64-bit)
- Internet: 10 Mbps feltöltés



3 Szoftverfejlesztés

3.1. A feladat

A program képes **valós idejű árfolyamok kimutatására** lehetővé teszi a vételt és az eladást a platformon, amely eltárolja az adott felhasználónak portfóliójában. A program **listázza mettől meddig tartja a pozíókat**, ki és befizetéseket listáz és számol ki **napi, heti, havi, éves vagy akár összes bontásban**. A jegyzett adatokból a szoftver készítsen statisztikákat, melyeknek paramétereit és tartalmát a felhasználó szabadon követhesse.

A fejlesztés közben be kell tartani a meghatározott kódolási konvenciókat. Az elkészült megoldásnak maradéktalanul meg kell valósítaniuk a megfogalmazott követelményeket. Fontos, hogy a választott megoldás **megfelelő színvonalú legyen** mind **felhasználói, mind fejlesztői szempontból**.

3.2. Modulok

A projekt keretében történő megvalósítás felbontási lehetősége az alábbi:

- Adatbázis tervezése, kivitelezése, interfész megírása
- Felhasználóbarát front end (**GUI**) tervezése:
 - Applikáció felület, grafikai elemek
 - Menürendszer
 - Beállítások
 - Összetett árazási grafikonok
 - Statisztika
 - Oktatóanyagok
 - Pénzkezelési felületek
 - Rendszer üzenetek pozíciókkal
 - Piaci naptár
- Back end kivitelezése
- Projektvezetéssel kapcsolatos dokumentációk, nyilvántartások vezetése, feladatak összehangolása, felhasználói dokumentáció elkészítése, tesztelés.

Lehetőség szerint a **fejlesztői dokumentációt** minden esetben a **ténylegesen fejlesztést végző projekttagok** készíték el. A felhasználói dokumentáció külön egységet képezhet, melyet azonban érdemes a teszteléssel összekapcsolni a **megfelelő minőség biztosítása** érdekében.



4 Szoftver specifikáció

4.1. A szoftver

A szoftver a háttérben végezze működését, az árfolyamok aktivitásának **pontos dátumát** és **időtartamát** kell nyomon követnie. A felhasználói felületnek alkalmasnak kell lennie arra, hogy az addig kereskedett részvényekből **különféle statisztikákat jelenítsen meg**. A kirajzolt adathalmaz időtartamát a **felhasználó adhassa meg**. A részvénydiagrammok folyamatosan fussanak és a részvény **nyitvatartási idején belül lehessen kereskedni**. A **gazdasági naptár felület** és a **menürendszer** zavartalan működéséért is a szoftver a felelős.

4.2. Megjelenés

- A program indításakor jelenjenek meg az éppen aktuális adatok, egyenleg, záró egyenleg, fedezet és a nyitott pozíciók.
 - A begyűjtött adatok, illetve diagramok jelenjenek meg a jobb felső sarokba kattintással. A megjelenítéshez lehessen minél több releváns paramétert meghatározni, nap, időszaktartomány.
- Az adott tartott pozíciók jelenjenek meg a főoldalon és a hozzájuk tartozó záró egyenleg.

4.3. Funkciók

A nyitott pozíciók idejét folyamatosan mentse el.

- A **mentett adatokból** paraméterezéshez szükséges szűrők segítségével lehessen megjeleníteni a **nyitott pozíciók használatának statisztikáit**. A paraméterek alapesetben a statisztika **kezdeti- és végdátuma**, valamint a statisztika tartalma (pl.: nyitva tartott idő, mennyiség, átlagos profit vagy deficit, egy részvényre jutó nyereség vagy veszteség).
- A statisztikák igény szerint **diagramon is kerüljenek megjelenítésre**, mely tartalmazza a szükséges adatokat. Pl.: részvények megnevezése, mennyisége, nyitva tartás ideje.
- **Oktatóanyagok interaktívak**, próbálnak rávilágítani a felhasználó **hiányos** kereskedési területeire, **alap tudására építkezik**. A menürendszerben az oktatóanyag gombra kattintva válogathat a felhasználó az anyagokból. A rendszer ajánlani fog neki **3 kérdés után speciálisan** a számára legfontosabbakat.



5 Dokumentáció

5.1. Technikai dokumentáció

Az elkészült **kódot, függvényeket és osztályokat megfelelő kommentekkel kell ellátni**, az átláthatóságra törekedve, továbbá el kell készíteni a **teljes alkalmazás dokumentációját**. A dokumentáció a feladat bonyolultságától függő hosszúságúnak kell lennie, **maximális terjedelem nincs meghatározva**.

A technikai dokumentáció szövegezésénél előírás, hogy a **nem hozzáértő személyek** számára is **feldolgozható legyen**, így az egyes fogalmak, rövidítések, idegen kifejezések magyarázatát a dokumentumnak maradéktalanul tartalmaznia kell. Jól átlátható **ábrákkal** és **jelmagyarázatokkal** lehet színesíteni a dokumentáció tömörségét.

5.2. Forráskód dokumentáció

A fontosabb **függvények** és **osztályok** előtt szerepelnie kell **megjegyzéseknek**, melyeknek tartalmazniuk kell az azt követő metódus rövid szöveges – akár magyar nyelvű – leírását. A forráskód dokumentációt a munka során **folyamatosan kell készíteni** a projekt előrehaladtával. A **kódnak tisztának kell lennie**, nem szabad tömören elhelyezni az eljárás halmozokat, hogy a **nem hozzáértők is könnyedén megértsék** a program működését.

5.3. Felhasználói dokumentáció

Az alkalmazás használatának részletes bemutatása, képernyőképekkel, funkciók pontos leírásával például:

- **Navigáció**
Hogyan lehet belépni, kilépni, regisztrálni, profil adatokat kezelní.
- **Fő funkciók**
árfolyamok megtekintése, grafikonok kezelése (időintervallum választása), kedvencek / portfólió követése, értesítések / figyelmeztetések beállítása.
- **Interaktív elemek magyarázata**
Melyik gomb mit csinál, milyen ikonok jelennie meg (pl. + jel = hozzáadás kedvencekhez).
- **Hibaüzenetek és visszajelzések**
Mit tegyen a felhasználó, ha hibaüzenetet kap (pl. „Nincs internetkapcsolat” → ellenőrizze a hálózatot).
- **Tippek és példák**
Pl. „Ha gyorsan szeretne egy részvény árfolyamát ellenőrizni, használja a keresőt a felső sávban.”



6 A projekt tesztelése és értékelése

6.1. A feladat felhasználó oldali szempontjai

A működő alkalmazás tesztelése alapján az alábbiak a legfontosabb jellemzők:

- Kiírást teljes egészében lefedő funkcionálitás
- Ötletgazdag kialakítás
- Funkcionális, egyszerű használat
- Letisztult felhasználói felület
- Stabil működés
- Részletes felhasználói dokumentáció

6.2. A feladat technikai szempontjai

Informatikai szakmai szempontból a megoldás értékelésének alapja:

- Kódkép, a kód tisztasága, komment minősége
- Kódolási konvenciók betartása
- Fejlesztői dokumentáció színvonala
- Dokumentált tesztelés
- Erőforrás felhasználásának pontos nyilvántartása

6.3. Projekt megvalósításának piaci jellegű értékelése

A projekt lezárulásakor összehasonlításra kerül a kezdeti erőforrás-terv, valamint a megvalósítás során dokumentált munka. Ezen dokumentumok elemzéséből vezetésre kerülnek azok a jellemző problémák, melyek a piaci környezetben jellemzően megjelennek. Feltárva a problémák okait, következményeit, lehetséges elkerülésüknek vagy hatásuk mérséklésének módjait. A jellemző hibák ebből a megközelítésből:

- Határidő csúszása
- Nem megfelelő minőség
- Hiányos, vagy elmaradó tesztelés
- Használhatatlan, pontatlan dokumentáció
- Pontatlan erőforrás becslés
- Aránytalanul magas önköltség
- Az elkészült termék továbbfejlesztésének, karbantartásának nehézségei

Elemezzük a megvalósítás tapasztalatait, a lehetséges továbbfejlesztés, átalakítás, támogatás kérdéseit és piaci lehetőségeit.



7 Projekt adatlap

Projekt neve: StockMaster

Feladat rövid ismertetése: egy tőzsdei platform, amely „demo” pénzzel lehetővé teszi a tőzsde megismerését, tanulását és az oktatóanyagokból erőt meríthet a sikeres kereskedés érdekében.

Felhasznált programozási nyelvek: C#, SQL, HTML, CSS, Javascript

Projektet összeállította: Csáki Balázs, Zsigó Róbert József