



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
AUTOMATIZÁLÁSI ÉS ALKALMAZOTT INFORMATIKAI TANSZÉK

Tőzsdei rendszer

SZOFTVERARCHITEKTÚRÁK (VIAUM105)
HÁZI FELADAT

Készítették

Kovács Dávid Balázs
JWMYX
davidkovaccs@gmail.com

Konzulens

Dévai István
tanársegéd
istvan.devai@aut.bme.hu

Barta Pál
M6E1QL
brazil.hu@gmail.com

2011. november 27.

HALLGATÓI NYILATKOZAT

Alulírott *Kovács Dávid Balázs* és *Barta Pál*, a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatói kijelentjük, hogy ezt a házi feladatot meg nem engedett segítség nélkül, saját magunk készítettük el.

Budapest, 2011. november 27.

Kovács Dávid Balázs
hallgató

Barta Pál
hallgató

Tartalomjegyzék

Hallgatói nyilatkozat	I
Tartalomjegyzék	II
Bevezető	1
1. Követelményspecifikáció	2
1.1. Definíciók	2
1.1.1. Felhasználó	2
1.1.2. Ügyfél	2
1.1.3. Tőzsdei instrumentum	3
1.1.4. Számla	3
1.1.5. Egyenleg	3
1.1.6. Megbízás	3
1.1.7. Ügylet	3
1.1.8. Bróker	3
1.1.9. Adminisztrátor	4
1.2. Felhasználói szerepkörök	4
1.2.1. Ügyfél	4
1.2.2. Bróker	4
1.2.3. Adminisztrátor	4
1.3. Leggyakoribb felhasználói esetek	4
1.3.1. Regisztráció kérelme	4
1.3.2. Megbízás adása	4
1.3.3. Megbízás visszavonása	4
1.3.4. Ügyfél megbízásainak listázása	5
1.3.5. Ügyfél ügyleteinek historikus listázása	5
1.4. Funkciók	5
1.4.1. Regisztráció kérelme	5
1.4.2. Felhasználó regisztrációja	5
1.4.3. Felhasználó kinevezése	5
1.4.4. Felhasználó deaktiválása	5
1.4.5. Ügyfél számla létrehozása	5
1.4.6. Ügyfél számla módosítása	5
1.4.7. Ügyfél számla törlése	5
1.4.8. Megbízás adása	6
1.4.9. Megbízás módosítása	6
1.4.10. Megbízás visszavonása	6
1.4.11. Ügyfél ügyleteinek historikus listázása	6
1.4.12. Ügyfél számláinak megtekintése	6

1.4.13.	Ügyfél számlaegyenlegének megtekintése	6
1.4.14.	Ügyfél élő megbízásainak megtekintése	6
1.4.15.	Ügyfél inaktív megbízásainak megtekintése	6
1.5.	Felhasználói interfész tervek	7
1.5.1.	Regisztráció	7
1.5.2.	Bejelentkezés	7
1.5.3.	Felhasználók menedzselése	8
1.5.4.	Ügyfél kezdőlapja	8
1.5.5.	Ügyfél számlái	9
1.5.6.	Ügyfél megbízásai	9
1.5.7.	Ügyfél ügyletei	10
1.5.8.	Új megbízás	10
1.6.	Biztonsági követelmények	11
1.6.1.	Authentikáció	11
1.6.2.	Authorizáció	11
1.6.3.	Beviteli mezők védelme	11
1.6.4.	Biztonságos kommunikáció	11
1.7.	Tervezett architektúra	11
2.	Rendszerterv	13
2.1.	Architektúra	13
2.2.	Rétegek	13
2.2.1.	Megjelenítési réteg	14
2.2.2.	Webszerver réteg	14
2.2.3.	Adatbázis réteg	14
2.3.	Funkciók	15
3.	Telepítési dokumentáció	16
	Értékelés	17

Bevezető

A tőzsde egy központosított nyilvános szervezett piac, ahol különböző papírokkal, árukkal kereskedhetnek a brókerek. Régen a tőzsdéken nem használtak számítógépes rendszereket, a megbízások kötése minden esetben papíron, emberi erővel történt, így nyilván nem tudtak olyan sebességgel, akkora mennyiségben (annyi áruval) kereskedni mint manapság. Napjainkban a tőzsde elképzelhetetlen bonyolult számítógépes rendszerek nélkül, amik automatikusan végzik a megbízások kezelését és az ügyletek kötését, az áruk árát pedig valós időben követhetjük az bárhol a világon (emiat ma már egyes számítások szerint a hálózati kábelek sebességén is múlhat egy-egy megbízás sikeressége).



1. ábra. Tőzsde

A tőzsdéknek a következő típusai léteznek:

- árutőzsde
- értéktőzsde
 - devizatőzsde
 - értékpapírtőzsde
 - hírpiac

Jelen web alapú tőzsdei applikációval egy értékpapír tőzsde működését kívántuk megvalósítani, úgy mint a megbízások, ügyletek kezelését, valós idejű értékpapír árfolyam kijelzés, számla adminisztráció.

1. fejezet

Követelményspecifikáció

A feladatunk egy tőzsdei kereskedési rendszer megvalósítása egy fejlett alkalmazás keretrendszer és többretegű architektúra segítségével. A fejezet elején 1.1. definiáljuk az alkalmazás fejlesztése közben előkerülő fogalmakat, majd bemutatjuk a különböző felhasználó típusokat akik használhatják a rendszert. A 1.3. fejezetben bemutatjuk a legfontosabb felhasználói eseteket, majd a 1.4. fejezetben felsoroljuk az összes funkciót amit a rendszer támogatni fog. A 1.5. részben felvázoljuk hogyan fognak kinézni a legfontosabb funkciók a képernyőn, illetve milyen lesz a képernyő elrendezése. A fejezet végén röviden kitérünk az alkalmazás használata során felmerülő biztonsági problémákra 1.6., majd bemutatjuk a tervezett architektúrát 1.7..

1.1. Definíciók

1.1.1. Felhasználó

A rendszert felhasználók használják akiknek autentikálniuk kell magukat a rendszerbe való belépéshez. Minden felhasználóról tároljuk a nevét, a jogosultági körét és a jelszavát. Három féle felhasználót különböztetünk meg: *ügyfél*, *bróker*, *adminisztrátor*.

Adatok: email cím, előnév, utónév, jelszó, szerepkör

Megkötések: kötelező a név, jelszó, email cím, szerepkör megadása

1.1.2. Ügyfél

Az ügyfél egy felhasználó. Miután beléptek a rendszerbe látják az árfolyamok aktuális alakulását és *megbízásokat* köthetnek (vétel-eladás) bizonyos *tőzsdei instrumentumra*. Lehetőségük van több *számlát* fenntartani és mindegyikről külön *megbízásokat* kötni, de legalább egy *számlájuknak* kötelező lenni. Listázhatja éppen futó *megbízásait* vagy nyomon követheti aktuális *ügyleteit*.

Megkötések: legalább egy számlája van

Adatok: számlák, megbízások, ügyletek

1.1.3. Tőzsdei instrumentum

A tőzsdén jegyzett és kereskedhető eszközök, amikkel az ügyfelek kereskedhetnek a rendszer segítségével, eladási vagy vételi *megbízásokat* köthetnek rájuk. Az áruk folyamatosan valós időben megjelenik az ügyfél előtt.

Adatok: aktuális ár

1.1.4. Számla

Minden ügyfélnek van legalább egy számlája, de akár több is. A számlának van egy *egyenlege* ami az aktuálisian felhasználható pénz mennyisége. Minden számlához tartozhatnak *megbízások*, *ügyletek*. Egy vételi megbízás indításakor a számláról lekerül a fedezet, majd ha az ügylet záros határidőn belül nem jön létre, a pénz visszakerül, egyébként végleg lekerül. Ha egy eladás ügylet létrejön az áru ára rákerül a az adott számlára (ahonnan az véve lett).

Megkötések: nem létezhet ügyfél nélkül

Adatok: egyenleg, megbízások, ügyletek

1.1.5. Egyenleg

A számlán aktuálisan található *pénz összege*. Maximum ekkora értékű megbízás indítható az adott számláról.

Adatok: pénz összeg

Megkötések: minden számlához tartozik

1.1.6. Megbízás

Egy megbízás lehet *eladás* vagy *vétel* egy adott tőzsdei instrumentumra, amelyet az ügyfél kezdeményez adott áron, adott mennyiségre. Ha két ellentétes típusú és ugyanolyan áru megbízás találkozik akkor létrejön egy *ügylet*. A mennyiségek oszthatódnak több ügyletre.

Megkötések: számlához tartozik (így ügyfélhez is)

Adatok: számla, instrumentum neve, mennyiség, ár, típus(vétel/eladás), kezdeményezés dátuma

1.1.7. Ügylet

Ha két ellentétes típusú *megbízás* történik egy idő intervallumban létrejön egy ügylet és a tőzsdei instrumentum gazdát cserélnek. A vevő számlájáról végleg lekerül a pénz, míg az eladóéra rákerül.

Adatok: két megbízás, mennyiség, dátum amikor létrejött

Megkötések: két ellentétes típusú ügylet kell hozzá

1.1.8. Bróker

A bróker egy felhasználó. Megnézheti egy adott *ügyfél számláit* élő vagy inaktív *megbízásait*, *ügyleteit*.

1.1.9. Adminisztrátor

Az ügyfelek, brókerek *regisztrációját* végzi a rendszerbe. Akár *deaktiválhatja* is felhasználókat, ő a rendszer ura.

1.2. Felhasználói szerepkörök

Mint fentebb már említettük három különböző felhasználói szerepkört különböztetünk meg a rendszerben.

1.2.1. Ügyfél

Miután belép látja az aktuális árfolyamokat és eladási vagy vételi megbízást indíthat akár azonnal. Listázni tudja a megbízásait, visszavonni, módosítani, látja a régebbi ügyleteit. Szintén listázni tudja a számláit, látja az aktuális egyenlegét, új számlát tud létrehozni, tud utalni pénzt a számlájára.

1.2.2. Bróker

Látja az összes ügyfél adatait, megbízásait, ügyleteit, számláit.

1.2.3. Adminisztrátor

Rendelkezik ugyanazokkal a szerepekkel, mint egy bróker, de képes regisztrálni brókert, ügyfelet vagy akár törölheti is őket a rendszerből.

1.3. Leggyakoribb felhasználói esetek

1.3.1. Regisztráció kérelme

A felhasználó egy felületen a szükséges adatok bevitelével regisztrációt kérvényez ami az adminisztrátorhoz kerül felülvizsgálatra. Az adminisztrátor engedélyezheti, elvetheti a kérelmet.

Szerepkör: Ügyfél, Adminisztrátor, Bróker

1.3.2. Megbízás adása

A felhasználó kitölt egy megbízási formulát, melyhez megadja a szükséges adatokat (instrumentum, mennyiség, eladás/vétel, ár), majd ezt beregisztrálja a rendszerbe.

Szerepkör: Ügyfél

1.3.3. Megbízás visszavonása

A felhasználó eltávolít egy megbízást.

Szerepkör: Ügyfél

1.3.4. Ügyfél megbízásainak listázása

A felhasználó kilistázza egy lapozható listában az összes aktív/inaktív megbízását.

Szerepkör: Bróker, Ügyfél (csak a sajátját)

1.3.5. Ügyfél ügyleteinek historikus listázása

A felhasználó kilistázza egy lapozható listában az összes ügyletet, ami az ügyfél megbízásai által jött létre. Az ügyleteket szűrhetjük aszerint, hogy eladók vagy vásárló az ügyfél.

Szerepkör: Bróker, Ügyfél (csak a sajátját)

1.4. Funkciók

1.4.1. Regisztráció kérelme

A felhasználó egy felületen a szükséges adatok bevitelével regisztrációt kérvényez

Szerepkör: Ügyfél, Adminisztrátor, Bróker

1.4.2. Felhasználó regisztrációja

A felhasználó a beérkezett regisztrációs kérelmet feldolgozza és létrehoz egy új felhasználót, a role alapvetően ügyfél típusú lesz

Szerepkör: Adminisztrátor

1.4.3. Felhasználó kinevezése

Az felhasználó kinevezhet egy ügyfelet brókernek vagy adminisztrátornak

Szerepkör: Adminisztrátor

1.4.4. Felhasználó deaktiválása

A felhasználó eltávolíthat egy felhasználót és annak összes bejegyzését a rendszerből

Szerepkör: Adminisztrátor

1.4.5. Ügyfél számla létrehozása

Az felhasználó az adott ügyfél számára létrehoz egy számlát

Szerepkör: Adminisztrátor

1.4.6. Ügyfél számla módosítása

A felhasználó módosítja az ügyfélhez tartozó számlát

Szerepkör: Adminisztrátor

1.4.7. Ügyfél számla törlése

A felhasználó eltávolít egy ügyfél számlát

Szerepkör: Adminisztrátor

1.4.8. Megbízás adása

A felhasználó kitölt egy megbízási formulát, melyhez megadja a szükséges adatokat, majd ezt beregisztrálja a rendszerbe.

Szerepkör: Ügyfél

1.4.9. Megbízás módosítása

A felhasználó módosítja az egyik megbízást.

Szerepkör: Ügyfél

1.4.10. Megbízás visszavonása

A felhasználó eltávolít egy megbízást.

Szerepkör: Ügyfél

1.4.11. Ügyfél ügyleteinek historikus listázása

A felhasználó kilistázza egy lapozható listában az összes ügyletet, ami az ügyfél megbízásai által jött létre. Az ügyleteket szűrhetjük aszerint, hogy eladók vagy vásárló az ügyfél.

Szerepkör: Bróker, Ügyfél (csak a sajátját)

1.4.12. Ügyfél számláinak megtekintése

A felhasználó lekéri egy adott ügyfél összes számlájának listáját.

Szerepkör: Bróker, Ügyfél (csak a sajátját)

1.4.13. Ügyfél számlaegyenlegének megtekintése

A bróker lekéri egy adott ügyfél adott számlájának egyenlegét.

Szerepkör: Bróker, Ügyfél (csak a sajátját)

1.4.14. Ügyfél élő megbízásainak megtekintése

A bróker lekéri egy adott ügyfél élő megbízásait.

Szerepkör: Bróker, Ügyfél (csak a sajátját)

1.4.15. Ügyfél inaktív megbízásainak megtekintése


A bróker lekéri egy adott ügyfél inaktív megbízásait.

Szerepkör: Bróker, Ügyfél (csak a sajátját)

1.5. Felhasználói interfész tervek

1.5.1. Regisztráció

Tőzsdei kereskedő rendszer



Vezetéknév

Utónév

E-mail cím

Jelszó

Jelszó újra

1.1. ábra. Regisztráció

1.5.2. Bejelentkezés

Tőzsdei kereskedő rendszer



E-mail cím

Jelszó

1.2. ábra. Bejelentkezés

1.5.3. Felhasználók menedzselése

Az adminisztrátor ezen az oldalon tudja karbantartani a felhasználókat.

Kijelentkezés

Tőzsdei kereskedő rendszer

Adminisztrációs felület

Ügyfelek
Brókerek
Adminisztrátorok

Vezetéknév	Családnév	E-mail cím	Törlés	
Szabó	Béla	szabobela@gmail.com	✕	Ügyfél funkciók... ▾
Török	Gábor	torokgabor@gmail.com	✕	Ügyfél funkciók... ▾
Kovács	Ádám	kovacsadam@gmail.com	✕	Ügyfél funkciók... ▾
Karsai	László	karsailaszlo@gmail.com	✕	Ügyfél funkciók... ▾

1.3. ábra. Felhasználók menedzselése

1.5.4. Ügyfél kezdőlapja

Kijelentkezés

Tőzsdei kereskedő rendszer

Kezdőlap
Számlák
Új megbízás
Megbízások
Ügyletek

Aktuális számla: euroSzámla

Eladás
Vétel

Név	eladás	vétel
EURUSD	1.40	1.41
USDHUF	0.1	0.2
USDCHF	1.2	1.3

EURUSD

Instrumentum	Típus	Mennyiség	Ár	Számla	Profit
EURUSD	eladás	10	1.41	euroSzámla	-10
USDHUF	eladás	10	1.41	euroSzámla	100
USDCHF	eladás	10	1.41	euroSzámla	-100
JPNUSD	eladás	10	1.33	euroSzámla	20

1.4. ábra. Ügyfél kezdőlapja

1.5.5. Ügyfél számlái

Tőzsdei kereskedő rendszer

Kijelentkezés

Kezdőlap	Számlák	Új megbízás	Megbízások	Ügyletek	Q
----------	---------	-------------	------------	----------	---

Aktuális számla: euroSzámla ▾

számla neve	egyenleg	kiválasztás
forintSzámla	300	
dollárSzámla	4000	
fontszámla	300000	
euroSzámla	566	

1.5. ábra. Ügyfél számlái

1.5.6. Ügyfél megbízásai

A bróker is egy hasonló oldalt lát az ügyfelek megbízásairól.

Tőzsdei kereskedő rendszer

Kijelentkezés

Kezdőlap	Számlák	Új megbízás	Megbízások	Ügyletek
----------	---------	-------------	------------	----------

Aktuális számla: euroSzámla ▾

típus	instrumentum	mennyiség	ár	megbízás lejártának időpontja
eladás	EURUSD	10	1100	2011.03.23 11:54
eladás	EURUSD	10	2400	2011.04.19 15:11
eladás	EURUSD	10	5300	2011.05.03 13:34
eladás	EURUSD	20	4800	2011.05.11 14:49
vétel	EURUSD	12	1100	2011.03.23 11:54
eladás	EURUSD	30	2400	2011.04.19 15:11
vétel	EURUSD	40	5300	2011.05.03 13:34
eladás	EURUSD	10	4800	2011.05.11 14:49

1.6. ábra. Ügyfél megbízásai

1.5.7. Ügyfél ügyletei

A bróker is egy hasonló oldalt lát az ügyfelek ügyleteiről.

Tőzsdei kereskedő rendszer

Kijelentkezés

Kezdőlap	Számlák	Új megbízás	Megbízások	Ügyletek
----------	---------	-------------	------------	----------

Aktuális számla: euroSzámla

eladó fél	vevő fél	mennyiség	ár	ügylet időpontja
Szabó Béla	Kovács Ádám	10	1100	2011.03.23 11:54
Török Gábor	Kovács Ádám	1	2400	2011.04.19 15:11
Kovács Ádám	Karsai László	3	5300	2011.05.03 13:34
Tóth Géza	Kovács Ádám	20	4800	2011.05.11 14:49
Szabó Béla	Kovács Ádám	1	1100	2011.03.23 11:54
Török Gábor	Kovács Ádám	3	2400	2011.04.19 15:11
Kovács Ádám	Karsai László	4	5300	2011.05.03 13:34
Tóth Géza	Kovács Ádám	111	4800	2011.05.11 14:49

1.7. ábra. Ügyfél ügyletei

1.5.8. Új megbízás

Tőzsdei kereskedő rendszer

Kijelentkezés

Kezdőlap	Számlák	Új megbízás	Megbízások	Ügyletek
----------	---------	-------------	------------	----------

Aktuális számla: euroSzámla

Instrumentum

Ár

Mennyiség

Vétel

Megbízás feladása

1.8. ábra. Új megbízás

1.6. Biztonsági követelmények

1.6.1. Authentikáció

Biztonsági szempontból a legfontosabb, hogy illetéktelen személyek ne tudjanak belépni mások felhasználói fiókjába, így számukra kedvezőtlen megbízásokat indítani vagy a számlájukról pénzt lopni. Természetesen az adminisztrátori szerepkörben lévő felhasználó védelme a legfontosabb, így ott érdemes akár valamilyen fizikai (token kulcs) autentikáció használata. Mivel a rendszer nagyon sok pénzt kezelhet, így ilyen fizikai kulcsot akár minden felhasználó kaphatna a végleges rendszerben.

1.6.2. Authorizáció

A felhasználói szerepeknek élesen, pontosan el kell különülniük és implementálás után az egyes jogokat megfelelően ki kell tesztelni, így megakadályozható, hogy egy felhasználó egy másik felhasználó számlájához hozzáférhessen, esetleg megbízásokat indítson onnan vagy egyszerűen csak olyan adatokhoz jusson hozzá ami nem az ő tulajdona.

1.6.3. Beviteli mezők védelme

Természetesen a különböző beviteli mezőket védeni kell, mind SQL Injection , mind XSS ellen, így megakadályozva, hogy illetéktelen kód fusson az adatbázison vagy a kliens böngészőjében. A kirívóan magas (a felhasználó számlájához képest) összegű megbízásokra érdemes külön visszaigazolást kérni, így elkerülhetők figyelmetlenségből, elgépelésből adódó nagyobb veszteségek.

1.6.4. Biztonságos kommunikáció

Annak érdekében, hogy egy harmadik fél ne halgathassa le a szerver-böngésző kommunikációt érdemes titkosított kapcsolatot alkalmazni, így megakadályozhatjuk mind, hogy harmadik félhez illetéktelen adatok kerüljenek ki, mind, hogy egy harmadik fél a kommunikáció közepébe álljon és hamis információkat adjon a felhasználónak vagy a szervernek (pl: magasabb összegű megbízás a felhasználó tudta nélkül). A rendszernek rendelkeznie kell egy hivatalos harmadik fél által kiadott tanúsítvánnyal ami biztosítja a felhasználót, hogy a biztonságos hálózati kommunikáció során nem egy hamis rendszerrel kommunikál, hanem a valódival.

1.7. Tervezett architektúra

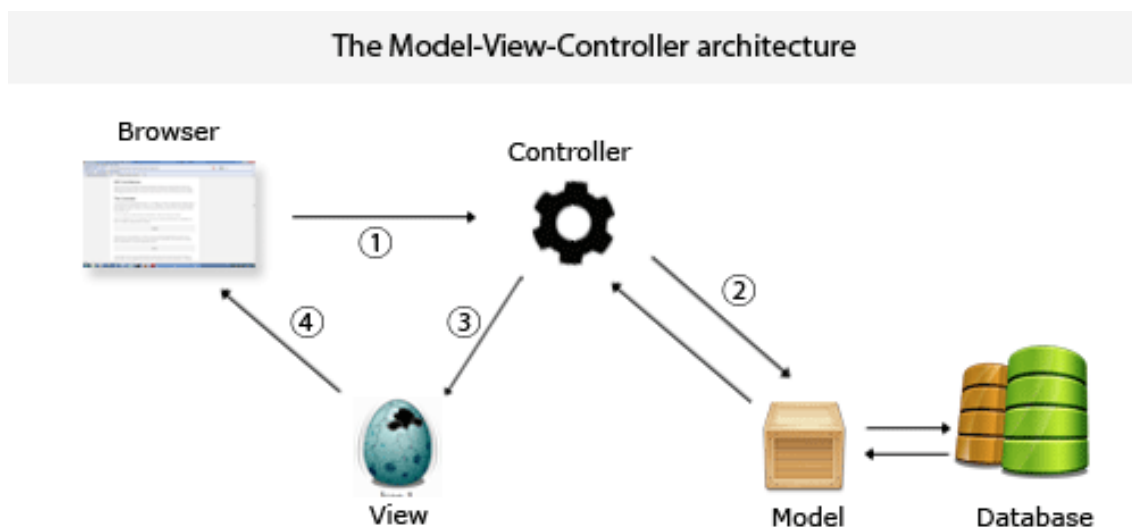
A rendszert egy három rétegű architektúrában valósítjuk meg. A megjelenítést a felhasználó böngészőjében (kliens oldalon) végezzük, az üzleti logikát (megbízások, ügyletek kezelése) egy dedikált szerveren, míg az adatokat egy adatbázis rétegben tároljuk. Az alkalmazás egy webalkalmazás lesz, a Ruby on Rails keretrendszer lesz az alapja, így a működése platformtól független lesz.

Az alkalmazás frontendje és üzleti logikája mögött egy SQL (MySQL vagy PostgreSQL) adatbázis fog futni.

Az adatbázisban tároljuk a felhasználók rekordjait, az ügyfelek számláit és az ezekhez tartozó megbízások és ügyletek rekordjait. Új felhasználó regisztrációjakor, új számla létrehozásakor, egy megbízás feladásakor vagy egy ügylet megkötésekor egy új bejegyzés jön létre az adatbázisban, ezeket különböző felületeken a felhasználók listázzák, egyes mezőket szerkeszthetnek, felülírhatnak vagy akár törölhetnek is. A különböző entitások között tartalmazó kapcsolatok lehetnek, melyeket figyelembe kell venni az adatbázis tervezésekor.

Az adatbázisból kiolvasott és éppen használatban lévő adatokat a memóriában tároljuk, mint amilyen az aktuálisan belépett felhasználó adatai vagy az aktuálisan kiválasztott számlához tartozó információ. Ezekkel az adatokkal folyamatosan dolgozunk, így megéri őket a memóriában tartani, adatbázis hívást csak entitások listázása, új entitás létrehozása, módosítása vagy törlése esetén hívunk, módosító műveleteknél a memóriában és az adatbázisban tárolt adatok szinkronizálásáról gondoskodunk.

A részvények aktuális árfolyamát a Google Finance API fogja nyújtani JSON formátumban, ezt az adatot nem szükséges az adatbázisba kimenteni, hiszen folyamatosan változik, a web service által visszaadott értékek érvényességi idejük alatt a memóriában lesznek tárolva, innen kérdezheti őket a webalkalmazás. A memóriában tárolt értékek periodikusan frissítve lesznek.



1.9. ábra. Architektúra

2. fejezet

Rendszerterv

Jelen fejezetben szeretnénk bemutatni az elkészítendő rendszer tervét. A következő részek kitérnek az architektúra kiválasztására és megvalósítására (2.1. szakasz), az itt szereplő rétegek felépítésére és funkciójára (2.2. szakasz), valamint végül a program funkcióinak ismertetésére (2.3. szakasz).

2.1. Architektúra

«««< HEAD Az alkalmazás három rétegből épül fel a klasszikus MVC architektúra alapján. A kliens oldalon a felhasználó webböngészőben futtatja az alkalmazást. A szerver oldali kód egy Apache Web Serveren fog futni, mely többszálú kliens kiszolgálást tesz lehetővé és képes titkosított csatornán végezni a kommunikációt a köztes rétegben, nagyban növelve a privát adatok védelmét. Az modell réteg a PostgreSQL adatbázis kezelő rendszeren alapul, mely egy széles körben támogatott open-source megoldás, mely megfelelő hatékonysággal képes szolgálni a rendszert. ===== A program architektúrájának a ?? szakaszban említett háromrétegű (multitier) architektúrát választottuk. Ezzel a megoldással a felhasználói felület adminisztrálásának terhére le tudjuk venni a kiszolgáló válláról, az csak az alkalmazás hosztolásában, illetve az esetleges működés közbeni kommunikációban vesz részt. »»»> bfa5d678ee66be46eed7925d4601ca683ec0f221

2.2. Rétegek

A megjelenítési réteg a kliens oldalon játszik fő szerepet, ahol a rendszer felhasználója egy tetszőleges böngészőben futtathatja a webalkalmazást. Az üzleti logikai funkciók a szerver oldalon futnak, a webalkalmazáshoz egy dedikált szerver tartozik, mely képes több kliens egyidejű kiszolgálására. Ide érkeznek be kliens oldalról a kérések, az üzleti logikai rétegen keresztül tudunk szolgáltatásokat nyújtani a kliens felé. A felhasználók perzisztens adataira, a különböző megbízások és ügyletek adatainak tárolására egy adatbázis és a programozható kezelőfelülete áll rendelkezésünkre.

2.2.1. Megjelenítési réteg

Az felhasználó számítógépén, egy tetszőleges böngészőben fut az alkalmazás kliens rétege, ami a vizuális megjelenítésért felelős és kapcsolódásokat definiál az üzleti logika réteg felé.

Az alkalmazás a legmodernebb webes trendeket követi, szem előtt tartva a minél nagyobb körű kompatibilitást különböző böngészőkkel. A felhasználóhoz érkező végső termék egy HTML lap CSS stíluslapokkal és Javascript kódokkal megtámogatva. A CSS fájlok felelnek a felhasználói felületen elhelyezett elemek beállításáért, segítséget nyújtanak egy koherens design megalkotásában, ahol az elemek vizuális tulajdonságai teljes mértékben lecsatolódnak az elemek definiálásától. A HTML kódban hozunk létre elemeket a kliens alkalmazás felhasználói felületén. Az elemek lehetnek statikus vagy interaktív elemek, utóbbiakat használva a felhasználó az üzleti logika felé kérésekkel fordulhat, melyekre a szerver válaszol egy újabb HTML fájlal. A felhasználó nem csak újabb oldalakra léphet tovább, hanem adatokat is küldhet, az üzleti logika által nyújtott felületen keresztül manipulálhatja a modell rétegben található adatokat, ha rendelkezik az ehhez megfelelő jogosultságokkal.

A webalkalmazás a kliens-szerver alapokon nyugvó REST architektúrán alapul, amelynek legfontosabb tulajdonsága, hogy minden fogalmát és objektumát a rendszernek erőforrással reprezentáljuk, melyeken különböző műveleteket végezhetünk, ezzel manipulálva a reprezentációt. A kliens oldalon a webböngésző különböző HTTP metódusokat (GET, POST, PUT, DELETE) küldve végezheti ezeket a műveleteket, melyeket az üzleti logika rész megfelelő funkciókhoz irányít.

2.2.2. Webszerver réteg

A kontroller réteget egy webszerver valósítja meg, melynek feladata a kienstől beérkező kérések és adatok feldolgozása, adatbázis manipulációk kezdeményezése és a válaszok visszaküldése a kliens réteg felé.

A webszerver legkülső felülete egy routing tábla, mely a kienstől érkező, egyes erőforrásokon végzett kéréseket a megfelelő funkcióhoz rendeli. Ezek a funkciók szintén az üzleti logika rétegben vannak definiálva, a különböző erőforrás reprezentációkhoz megfelelő modellek vannak definiálva. A modelleken az objektum-orientált szemléletet követve attribútumok és funkciók vannak definiálva. Az osztályok és attribútumok közvetlen kapcsolatban állnak az adatbázis egyes tábláival, definiálhatunk ezen túl egy-egy, egy-több és több-több kapcsolatokat, melyek az adatbázis rétegben kapcsolótáblák segítségével jelennek meg, de az üzleti logika réteg számára ez a működés transzparens.

2.2.3. Adatbázis réteg

Az adatbázisban az üzleti logika részben definiált osztályok tábla reprezentációi szerepelnek, a köztük levő kapcsolatokat megvalósító kapcsolótáblák és különböző segéd táblák, melyek csupán a működést támogatják. Az adatbázisban egy tábla egy erőforrás osztálynak felel meg, ezen osztályok példányai a megfelelő tábla egy sorában vannak tárolva, az osztályok attribútumainak a tábla oszlopai felelnek meg.

A USER tábla tartalmazza a regisztrált felhasználókat, a jelszavakat lekódolva tároljuk az esetleges sikeres támadás hatásainak tompítása érdekében. A bejelentkezéshez kapcsolatos adatok és a felhasználó egyéb adatai tárolódnak ebben a táblában, továbbá egy flag, mely jelzi, hogy a felhasználó regisztrációját aktiválva van-e.

A ROLE tábla tartalmazza a rendszerben megtalálható szerepeket, az adminisztrátor, bróker és felhasználó szerep elérhető jelenleg a rendszerben. A felhasználók és szerepek között több-több kapcsolat van, ezt a $USER_{ROLE}$ kapcsolótáblával tájékoztatjuk.

Minden felhasználóhoz tartozhat több számla, így az ACCOUNT tábla a számla adatai mellett egy referenciát is tartalmaz a megfelelő felhasználóra.

A STOCK tábla különböző részvényeket reprezentál, mely alapvető adatokat tartalmaz a részvényről (név, aktuális árfolyam, különböző időbélyegek).

A TRANSACTIONS tábla tartalmazza a már megkötött ügyleteket egy referenciával a megfelelő részvény típusra. Ezen kívül különböző időbélyegeket és az ügylet árát tartalmazza.

Az ORDERS tábla a megrendeléseket tartalmazza, egy megrendelés egy számlához tartozik, így a megfelelő oszlopban hivatkozik egy számlára. Ezen kívül egy ügylet referenciát is tartalmaz, mely NULL értéket tartalmaz, ha az ügylet még nem jött létre. A harmadik referencia a megrendelt részvényre mutat, ezentúl tartalmazza a megrendeléshez tartozó egyéb információkat (ár, vétel-eladás flag, időbélyegek).

2.3. Funkciók

3. fejezet

Telepítési dokumentáció

Az alkalmazást természetéből adódóan egy webserverre szükséges telepíteni, ez célszerűen a Microsoft IIS 7-es verziója. A telepítéshez a webprojekt könyvtárának tartalmát kell elérhetővé tenni a webserveren. Célszerű a könyvtárban található HTML, vagy ASPX fájlt a könyvtár default kezdőlapjának beállítani, így a **kiszolgáló/elérésiút** címen az alkalmazás könnyen indítható.

A program működése emellett természetesen igényli a felhasználó számítógépén a Silverlight környezet meglétét. Azonban ha nem érzékeli, a kezdőlapon automatikusan felajánlja a letöltést és telepítést, ami után a rendszer rögtön használhatóvá válik.

Értékelés

A házi feladat elkészítése során bepillantást nyerhettünk, az architektúrális szoftvertervezés és -fejlesztés lépéseibe. Munkánk során sikerült egy olyan programot fejlesztenünk, amely – egyelőre alapszinten – teljesíti a követelményspecifikációban lefektetett, majd a tervezés során pontosított igényeket. A munkánk során szerzett tapasztalatokat reményeink szerint a későbbiekben kamatoztathatjuk, amennyiben a mostanihoz hasonló, komplex tervezést és kivitelezést igénylő feladattal találkozunk.

Továbbfejlesztési lehetőségek

Lehetséges, hogy a szoftverünk funkciói, kezelői felülete több potenciális felhasználónak elnyeri a tetszését, mint ahányan valójában elkezdik használni. Ennek egyik okát abban láthatjuk, hogy sokan nem szeretnék veszni hagyni egy másik családfaszerkesztő-programban¹ fáradságos munkával létrehozott családfájukat. Erre egy lehetséges megoldásként kínálkozik, hogy a szoftverünk továbbfejlesztése során beépítsünk GEDCOM importálási/exportálási lehetőséget.

A GEDCOM a *GEnealogy Data COMmunication* kifejezésből képzett mozaikszó, és a genealógiai adatok cseréjének de facto szabványát képezi a származástannal kapcsolatos szoftverekben. A GEDCOM szabvány egy egyszerű szöveges struktúrában teszi lehetővé a leszármazási kapcsolatok modellezését és tárolását.

A szabvány támogatásával programunk bekerülhet a családfa-modellezési szoftverek vérkeringésébe. A fent lábjegyzetben hivatkozott összehasonlító táblázat tanúsága szerint ez korántsem olyan kiterjedt és centralizált, hogy egy-egy jól megvalósított ötlettel ne lehetne a téma iránt érdeklődő felhasználók jelentékeny hányadának bizalmát elnyerni.

¹lásd: http://en.wikipedia.org/wiki/Comparison_of_genealogy_software