

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

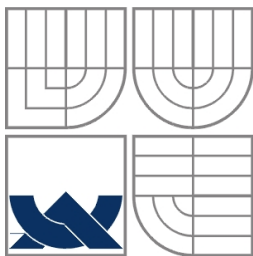
WEBOVÁ APLIKÁCIA PRE SPRÁVU INVESTIČNÉHO  
PORTFÓLIA

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

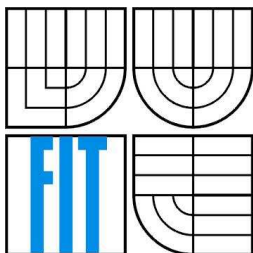
AUTOR PRÁCE  
AUTHOR

MARTIN KASSAY

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# WEBOVÁ APLIKACE PRO SPRÁVU INVESTIČNÍHO PORTFÓLIA

WEB APPLICATION FOR PORTFOLIO MANAGEMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN KASSAY

VEDOUcí PRÁCE

SUPERVISOR

ING. LADISLAV RUTTKAY

BRNO 2009

## **Abstrakt**

Investování je určitá činnost, kterou je možné provádět na základě relevantních informací o trhu na kterém se investice uskuteční. Je třeba aby tyto informace byly co nejpřesnější a dostatečně obsáhlé. Rozvoj informačních technologií a tedy i rozvoj webových aplikací přináší nové možnosti šíření respektive získávání potřebných informací o trhu. Důležité je aby samotné data o vývoji trhu a jednotlivých finančních nástrojích byly převedeny do informací, tj. grafů, tabulek, simulačních výstupů, tak aby byly zřetelné. Technologie NET.Framework přináší množství nástrojů, které lze využít na vývoj a implementaci webových aplikací, právě o investování na finančních trzích.

## **Abstract**

Investment is an activity that can be performed on the basis of relevant market information on which the investment is made. It is necessary that this information is as accurate and sufficiently comprehensive. Development of information technology and the development of web applications brings new opportunities for disseminating or obtaining the necessary information on the market. It is important that the data itself on the market and various financial instruments are appropriately transformed into information, i.e. graphs, tables, simulation output, so that clear. Technology NET.Framework brings a number of tools which can be used for the development and implementation of web applications just about financial investment.

## **Klíčová slova**

Webová aplikace, webová služba, ASP.NET, NET.Framework, investování, investiční nástroj, ER diagram, objekt

## **Keywords**

Web application, Web services, ASP.NET, NET.Framework, investing, investment instrument, ER diagram, object

## **Citace**

Kassay Martin: Webová aplikácia pre správu investičného portfólia, bakalárska práca, Brno, FIT VUT v Brně, 2009

# Webová aplikácia pre správu investičného portfólia

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Ladislava Ruttkaya.

Uviedol som všetky literárne pramene a publikácie, zo ktorých som čerpal.

.....  
Martin Kassay

20.5.2009

## Pod'akovanie

Ďakujem vedúcemu práce p. Ing. Ladislavovi Ruttkayovi za ochotu, cenné rady a usmernenie pri písaní záverečnej práce.

© Martin Kassay, 2009

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah .....	1
1 Úvod.....	4
2 Investovanie a informácie .....	6
2.1 Investovanie do akcií.....	6
2.1.1 Akcia.....	6
2.1.2 Cenné papiere .....	6
2.1.3 Obchodovanie s cennými papiermi.....	6
2.1.4 Burza cenných papierov .....	7
2.1.5 Investičné portfólio .....	7
2.2 Vplyv informačných technológií na rozvoj investovania.....	7
2.2.1 Historický pohľad .....	7
2.2.2 Rozvoj investovania v dôsledku inovácie informačných technológií .....	8
2.3 Potrebné informácie a ich dostupnosť .....	8
2.4 Prínos aplikácie pre užívateľa.....	9
2.4.1 Úloha a cieľ webovej aplikácie .....	9
2.4.2 Popis webovej aplikácie .....	10
2.5 Webové služby ako zdroj informácií .....	10
3 Technológie vývoja webových aplikácií.....	11
3.1 Webová aplikácia.....	11
3.1.1 Definícia webovej aplikácie.....	11
3.1.2 História aplikácií klient - server .....	11
3.1.3 Technológie webových aplikácií.....	11
3.2 Trojvrstvová architektúra .....	12
3.2.1 Dvojvrstvová verzus Trojvrstvová architektúra.....	12
3.2.2 Prezentačná vrstva tenký klient.....	13
3.2.3 Dátová vrstva .....	13
3.2.4 Aplikačná vrstva .....	13
3.3 Technológia .NET .....	13
3.3.1 Úvod do technológie .NET .....	13
3.3.2 Webové aplikácie v ASP.NET .....	15
3.4 XML a webové služby .....	16
3.5 Vývojový model.....	17

3.5.1	Proces vývoja softwaru.....	18
3.5.2	Modely životného cyklu softwaru.....	18
3.6	Objektovo orientovaný návrh.....	19
3.6.1	Objektovo orientovaný model.....	19
3.6.2	Diagram tried.....	20
3.6.3	Trieda.....	20
3.6.4	Asociačná väzba.....	20
3.6.5	Agregačné väzba.....	20
3.6.6	UML.....	21
4	Analýza a návrh.....	22
4.1	Požiadavky na systém.....	22
4.1.1	Informácie.....	22
4.1.2	Zdroje informácií.....	22
4.1.3	Funkcie aplikácie.....	23
4.1.4	Užívatelia.....	23
4.2	Prípady použitia.....	24
4.2.1	Popis prípadov použitia.....	24
4.2.2	Modelovanie prípadov použitia.....	24
4.3	Databázový model.....	25
4.3.1	Analýza dát.....	25
4.3.2	ER diagram.....	25
4.3.3	Dostupné dáta na webe.....	26
5	Implementácia.....	27
5.1	Vytvorenie databázy.....	27
5.2	Dizajn Aplikácie.....	28
5.2.1	Dátová vrstva.....	28
5.2.2	Vrstva prístupu k dátam DAL.....	29
5.2.3	Logická vrstva BLL.....	29
5.2.4	Prezentačná vrstva.....	30
5.3	Rozloženie aplikácie.....	30
5.3.1	Informačné celky.....	30
5.3.2	Konfiguračné nastavenia aplikácie.....	31
5.3.3	Adresár zdrojového kódu.....	31
5.4	Užívateľské rozhranie.....	32

5.4.1	Registácia .....	32
5.4.2	Autorizácia .....	32
5.4.3	Cenné papiere .....	32
5.4.4	Portfólio .....	32
5.4.5	Prehľad.....	33
5.4.6	Ostatné .....	33
5.4.7	Navigácia po webovom sídle.....	33
5.4.8	Témy .....	33
5.5	Použité nástroje.....	34
5.5.1	AJAX Toolkit .....	34
5.5.2	Chart.NET .....	34
5.5.3	FCKeditor.....	34
6	Verzia a rozšírenie aplikácie.....	35
7	Vývojový softvér.....	36
8	Záver .....	37
	Použitá literatúra.....	38

# 1 Úvod

Každodenne sa stretávame s rastúcim záujmom investorov o kapitálové trhy. Napredovanie informačných technológií a rozvoj siete internet podporujú šírenie informácií a tak podmieňujú dostupnosť investovania. V súčasnej dobe má takmer ktokoľvek možnosť investovať svoj kapitál na finančných trhoch po celom svete. Rozvoj technológií webových stránok a webových aplikácií tak prinášajú takmer neobmedzený priestor finančného trhu do osobných počítačov každého užívateľa. Napríklad v súčasnosti nie je problém aby investor v českej republike investoval do japonského Yenu, akcií spoločnosti General Motors na burze v New Yorku, nakúpil ropu na Londýnskej burze alebo špekuloval o vývoji cien spoločnosti Zentiva na Pražskej burze. Základným predpokladom úspešného investora sú jeho kvalitné znalosti problematiky a trepezlivosť učiť sa na vlastných investičných rozhodnutiach. Práve pre takéto potreby existujú webové serveri, ktoré svojim užívateľom vytvárajú prostredie pre získavanie informácií a sprístupňujú najrôznejšie webové aplikácie pre testovanie svojich rozhodnutí a simulovanie možného zisku alebo straty. Práve technológia ASP.NET prináša účinné nástroje na tvorbu takýchto webových aplikácií. Ide o technológiu vývoja implementácie webových aplikácií, ktorý pracuje na základe vývojového nástroja NET.Framework od spoločnosti Microsoft.

## **Investovanie a informácie**

Táto kapitola vytvára jednoduchý úvod do problematiky investovania. Pojednáva o možnostiach, ktoré sú dostupné a podáva vysvetlenie potreby webových aplikácií v investorskom kontexte.

## **Technológie vývoja webových aplikácií**

V tejto kapitole je popísaný prehľad technológií a procesov pre návrh, vývoj a implementáciu webových aplikácií. Sú popísané také technológie, ktoré budú využité pri celom procese vývoja webovej aplikácie

## **Analýza a návrh**

Uvedené informácie popisujú technológiu návrhu dátovej štruktúry a približujú použité jazyka UML pre túto potrebu. Uvedená kapitola popisuje konkrétny návrh tejto dátovej štruktúry, vrátane E-R diagramov(entitno-relačných diagramov) a Use Case diagramov (diagramov prípadov užitia).

## **Implementácia**

Táto kapitola popisuje implementáciu celej aplikácie. Ide teda o vytvorenie databázy, tvorbu užívateľského rozhrania a použité nástroje. Hlavnou časťou kapitoly je popis dizajnu celej aplikácie, ktorý vychádza z modelu trojvrstvovej architektúry.

## **2 Investovanie a informácie**

### **2.1 Investovanie do akcií**

#### **2.1.1 Akcia**

Akcia je cenný papier, sú s ňou spojené práva akcionára ako spoločníka podieľať sa na riadení spoločnosti, jej zisku a na likvidačnom zostatku pri zániku spoločnosti. Patrí do kategórie dlhodobých cenných papierov. Majiteľ akcie nemá právo vyžadovať od emitenta - vydavateľa cenného papiera - zaplataenie nominálnej hodnoty akcie. Akciu (akcie) je možné predať iba podľa aktuálneho trhového kurzu. Výnosom z držby akcie môže byť dividenda. Jej výška je závislá na zisku akciovej spoločnosti. Ďalej je to kapitálový zisk, rozdiel medzi nákupnou a predajnou cenou akcie.

Hovorí sa, že ten kto chce dobre spať, by sa mal akciám vyhnúť, naopak ten, kto chce dobre jesť, by mal investovať do akcií. Realita je však taká, pokiaľ váš investičný horizont trvá viac ako 10 rokov, tak by akcie mali tvoriť väčšiu časť investičného portfólia. Investovanie sa zaraďuje medzi dôležité peňažné operácie, ktoré uskutočňuje subjekt, právnická alebo fyzická osoba. Podstata investície je v nákupe dlhodobého finančného, nehnuteľného poprípade hnuiteľného majetku a krátkodobého majetku. Pri investičných operáciách je potrebné analyzovať a rozložiť riziko (minimalizovať ho), stanoviť trvanie investície a vyhodnotiť výšku možného výnosu.

#### **2.1.2 Cenné papiere**

Cenné papiere sú listiny predstavujúce pohľadávky vlastníkov voči vystavovateľovi týchto listín. Vznik, trvanie, plnenie a zánik pohľadávok je viazaný na tieto cenné papiere.

Obchodník s cennými papiermi je právnická osoba, ktorá má na vykonávanie obchodov s cennými papiermi licenciu, a to prostredníctvom fyzickej osoby - makléra, ktorý koná v mene obchodníka s cennými papiermi. Môže mať iba formu akciovej spoločnosti.

#### **2.1.3 Obchodovanie s cennými papiermi**

Obchodovanie s cennými papiermi ponúka vysoký výnos, avšak nesie so sebou určité riziko straty, pričom platí: čím vyšší výnos, tým väčšie riziko. Niekde medzi týmito dvoma spomínanými spôsobmi zhodnocovania finančných prostriedkov je práve kolektívne investovanie, ktoré v sebe spája požiadavku bezpečnosti investície a zároveň primeraného výnosu.



Obrázok 1: Akcia na majiteľa<sup>1</sup>

## 2.1.4 Burza cenných papierov

Burza cenných papierov je špecificky organizovaný trh, na ktorom predávajúci, kupujúci a sprostredkovatelia realizujú obchody pomocou cenných papierov. Miesto, kde sa organizuje dopyt a ponuka cenných papierov v určenom čase podľa platných pravidiel a burzových predpisov.

## 2.1.5 Investičné portfólio

Investičné portfólio je štruktúrovaná skladba držaných cenných papierov a iných investičných nástrojov. To umožňuje investorovi obmedziť riziko vyplývajúce predovšetkým z pohybu trhových cien obligácií a akcií a zo zmien v úrokovej a dividendovej výnosnosti cenných papierov a docieľiť tak maximálny zisk.

Portfólio fondu je súbor všetkých majetkových položiek podielového fondu. Najčastejšie sa používa v súvislosti s cennými papiermi v majetku vo фонде.

## 2.2 Vplyv informačných technológií na rozvoj investovania

### 2.2.1 Historický pohľad

Postupným vývojom informačných technológií sa investovanie rozšírilo medzi širokú verejnosť. V minulosti musel investor vložiť svoj kapitál do rúk makléra, ktorý mu poskytoval informácie o aktuálnom dianí na trhu a radil mu aké investičné rozhodnutia by mal urobiť. Jediným zdrojom informácií boli špecializovaná denná tlač, ktorá poskytovala aktualitu zo sveta investovania a burzy.

<sup>1</sup> E-Trend : Webový portál o podnikaní, dostupné na: <http://www.etrend.sk/podnikanie/uctovnictvo-dane-odvody/zmenky-su-casto-praktickejsie-ako-hotovost/20903.html>

Práve rozvoj informačných technológií a počítačovej siete internet, zapríčinili masívny rozvoj investovania. Informácie, kedysi dostupné len priamym zúčastneným, boli sprístupnené každému, kto mal chuť a čas sa niečo o takejto činnosti dozvedieť. V súčasnosti môžeme na internete dohľadať informácie najrôznejšieho charakteru k tejto téme. Od aktuálnych cien jednotlivých investičných titulov, ktoré sú relatívne okamžité, cez psychológiu a techniky obchodovania na burze až po „realtime“ obchodovanie cez obline aplikácie.

## **2.2.2 Rozvoj investovania v dôsledku inovácie informačných technológií**

Webové stránky, ktoré sú dostupné z akéhokoľvek miesta na svete, kde je dostupnosť siete internet, prinášajú investorom po celom svete aktuálne informácie priamo z burzy. Napríklad investor z českej republiky, má možnosť sledovať aktuálne dianie na svetoznámej burze NYMEX (New York Merkantile Exchange ) priamo na ich webových stránkach a obchodovať cenné papiere na tejto burze napríklad cez európsku investičnú banku Saxobank, priamo z pohodlia domova prostredníctvom počítača s pripojením do siete internet.

Investovať svoj kapitál na finančnom trhu môže doslova ktokoľvek, kto má chuť a čas. Na internete sú dostupné webové stránky, v akomkoľvek jazyku, ktoré začínajúceho investora naučia základné pravidlá a techniky investovania. Existujú diskusné fóra, kde sa stretávajú investori z najrôznejších kútov sveta a vymieňajú si svoje skúsenosti. Príkladom takéhoto webového servera je v českej republike napríklad financnik.cz. Pre milovníkov štatistik, grafov a najrôznejších analýz existujú takéto webové stránky, ktoré ponúka napríklad webový server quotes.com.

Informačné technológie a rozvoj siete internet, umožnil vznik a rozvoj veľmi úspešných elektronických trhov. Sú to v podstate burzy cenných papierov, ktoré však nemajú podobu veľmi známych burzových parketov, kde jednotliví obchodníci mávajú s objednávkami na kúpu alebo predaj obchodovaných titulov, ale existujú len vo virtuálnom svete. Príkladom takejto burzy je Forex, kde môžeme obchodovať s menami a vytvárať zisk, alebo stratu na základe vývoja kurzov svetových mien. Veľmi populárny menový pár, ktorý sa najčastejšie obchoduje je EUR/USD, t.j. kurz euro meny voči doláru alebo naopak.

Vstup na burzu v minulosti podmieňoval kapitál vo výške 100 000 dolárov. V súčasnosti tomu nie je inak, ale práve informačný boom vytvoril rôzne príležitosti pre rozvoj maklérskeho domov, ktoré za mierny poplatok z investovanej časti sprístupnia hru na poli investícií aj investorom s kapitálom 10 000 Kč. Príkladom takejto burzy je práve Forex, ktorý je dostupný v českej republike prostredníctvom desiatky maklérskeho spoločností.

## 2.3 Potrebne informacie a ich dostupnost'

Na to aby sa bežný človek stal dobrým investorom, potrebuje veľa študovať danú problematiku, tak ako je to aj v ostatných odvetviach. Prvá vec, o ktorú sa musí začínajúci investor zaujímať, je podstata celého investovania. Je treba sa naučiť čo všetko popisuje pojem investovanie a obchodovanie na kapitálovom trhu, zohľadniť svoje možnosti, či už finančné alebo časové. Druhým a zároveň najdôležitejším krokom k úspešnému investovaniu je učenie a testovanie svojich investorských zručností

Pre takéto účely existujú rôzne webové aplikácie, ktoré umožňujú investovať fiktívny kapitál a učiť sa tak na vlastných chybách. Aj najúspešnejší investori potrebujú spravidla jeden rok fiktívneho investovania na to aby sa odvážili vložiť do tejto „populárnej hry“ svoj nemalý a ťažko zarobený kapitál. Vytúženým výsledkom úspešného investora sú určite čo najvyššie zisky. Avšak investovanie nie je snaha o vysoké zisky za krátky čas, ale sústredený a rozvážny proces rozhodnutí určený trpezlivým investorom. Nerozvážny investor sa po čase stáva dokonalým zdrojom kapitálu pre tieto trhy a zdrojom zisku pre tých úspešnejších. Z tohto dôvodu sa masívne rozšírili webové stránky, ktoré majú naučiť investora úspešne zvládnuť celý proces. Ich primárnym účelom je práve poskytnúť začínajúcemu investorovi priestor pre testovanie správnosti ich investičných rozhodnutí. Takmer každý broker(maklérska spoločnosť) poskytuje takzvané demo aplikácie, ktoré umožnia investovať fiktívny kapitál pre vzdelávacie účely. Motívom takýchto možností je prilákať čo najväčšiu pozornosť širokej verejnosti o investovanie, a tým si vytvoriť väčšie množstvo potencionálnych investorov, ktorý budú investovať svoj kapitál práve prostredníctvom ich služieb. Na druhú stranu existuje veľa nadšencov, ktorý vytvárajú najrôznejšie webové aplikácie, za účelom rozšírenia takejto vzrušujúcej investorskej hry. V kruhu úspešných investorov môžeme často počuť názor, že každý nový investor prináša na kapitálový trh nový kapitál, a tým zvyšuje množstvo peňazí, o ktoré môžu investori súperiť.

Na to aby začínajúci investor zdokonalil svoju techniku, potrebuje veľké množstvo aktuálnych aj historických dát, či už v podobe grafov, alebo neuveriteľne rozsiahlych štatistík, a aktuálne novinky zo sveta politiky, ekológie, obchodu a priemyslu. Zdrojom takýchto informácií sú špecializované webové serveri, ktoré poskytujú takéto informácie prostredníctvom webových stránok, RSS kanálov alebo webových služieb.

## 2.4 Prínos aplikácie pre užívateľa

### 2.4.1 Úloha a cieľ webovej aplikácie

Úlohou tejto webovej aplikácie je vytvoriť nový priestor pre začínajúcich investorov, tak aby mali možnosť testovať svoje investorské aktivity a mali prístup k základným dátam potrebným pre

správne rozhodovanie. Väčšina webových aplikácií, ktoré sa venujú danej problematike, poskytuje veľké množstvo informácií, ktoré môžu začínajúceho investora odradiť. Dôvodom je práve preinformovanie a následné nepochopenie poskytnutých informácií čo z psychologického hľadiska vytvorí nechť a nezáujem investovanie.

Táto webová aplikácia si dáva za cieľ poskytnúť také množstvo relevantných informácií, ktoré povedú k lepšiemu pochopeniu celého procesu pre začínajúcich investorov.

## 2.4.2 Popis webovej aplikácie

Webová aplikácia prinesie užívateľom základné informácie o investičných nástrojoch a jednotlivých tituloch, ktoré sú dostupné na finančných trhoch. Aplikácia umožní poskladanie vlastného portfólia investičných nástrojov v danom investičnom portfóliu. Užívateľ bude mať tak možnosť testovať výnosnosť navrhnutého portfólia a analyzovať tak správnosť rozloženia rizika v takto vytvorenom portfóliu jak na relatívne aktuálnych dátach, tak aj na dátach historických.

## 2.5 Webové služby ako zdroj informácií

Získavanie aktuálnych a historických dát o kapitálových trhoch a ich sprístupnenie je náročný proces. Existujú rôzne spoločnosti, ktoré takéto dáta zbierajú a poskytujú ich investorom, za určitý poplatok. O distribúciu takýchto informácií investorom sa starajú práve maklérske spoločnosti prostredníctvom vlastných webových stránok a webových aplikácií. Pre možnosť takejto distribúcie dát sú práve webové služby, poskytujúce suché dáta. Prezentáciu takýchto dát zabezpečujú najrôznejšie webové serveri podľa ich zamerania a špecializácie na oblasť investovania na kapitálových trhoch.

Ľudstvo prežíva informačný vek a práve informácie sú tou najcennejšou devízou. Aktuálne informácie zo sveta investovania majú na trhu nemalú cenu. Avšak neaktuálne dáta, respektíve dáta staršie niekoľko dní či hodín sú k dispozícii zadarmo. Práve tieto dáta, napríklad v podobe webových služieb sú zdrojom pre testovanie a zdokonaľovanie investorskej techniky. Takéto informačné spoločnosti ich poskytujú v podobe „free služieb“, a slúžia ako priamy dátový zdroj pre voľne dostupné webové aplikácie a stránky.

*Dáta sú akékoľvek vyjadrenie (prezentácia) skutočnosti, schopné prenosu, uchovania, interpretácie či spracovania. Informácie sú dáta, ktoré majú sémantiku(význam).<sup>2</sup>*

Definícia týchto pojmov vysvetľuje, že dáta sa stávajú informáciami až po ich prezentácii v určitom kontexte. Podobne aj v tomto prípade, existujú zdroje informácií a prezentácie týchto dát na webových stránkach a aplikáciách. Pre tieto potreby sú najčastejšie využívané práve webové služby a technológia XML.

---

<sup>2</sup> Prof. Ing. Tomáš Hruška, CSc., Opora k predmetu *Informačné systémy* IIS, FIT VUT Brno 2007.

# 3 Technológie vývoja webových aplikácií

## 3.1 Webová aplikácia

### 3.1.1 Definícia webovej aplikácie

Webová aplikácia je v softvérovom inžinierstve definovaná ako aplikácia poskytovaná užívateľom z webového serveru prostredníctvom počítačovej siete Internet, respektíve prostredníctvom podnikovej siete Intranet. Webová aplikácia je založená na technológii klient server. Webové aplikácie sú dostupné prostredníctvom webového prehliadača na zariadení klienta. Takúto klientskú aplikáciu nazývame tenký klient, pretože logika aplikácie mu nie je známa.

Hlavnou prednosťou takejto aplikácie je možnosť jej spravovania a aktualizácie bez nutnosti inštalácie. Webové aplikácie sú využívané pre implementáciu podnikových systémov, iných informačných systémov, ale aj internetových obchodov, diskusných fór, voľne dostupných emailových aplikácií.

### 3.1.2 História aplikácií klient - server

V minulosti si každá aplikácia typu klient - server vyžadovala svoj klientský program, ktorý slúžil ako užívateľské rozhranie a musel byť inštalovaný na osobnom počítači užívateľa. Každá aktualizácia serverovej časti, vyžadovala aktualizáciu aj na strane klienta. Táto skutočnosť viedla k zvyšovaniu nákladov na podporu a znižovaniu efektivity celej aplikácie. Naopak webové aplikácie umožňujú využívať webový prehliadač ako užívateľské rozhranie.

### 3.1.3 Technológie webových aplikácií

Webové aplikácie generujú webové stránky vo formáte HTML/XHTML a posielajú ich webovému prehliadaču ako statický dokument. Klientská interakcia užívateľa je sprístupnená prostredníctvom formulárových prvkov vložených do kódu webovej stránky.

Webové rozhranie v určitom smere obmedzuje funkčnosť a možnosti klienta v porovnaní s desktopovou aplikáciou. Z tohto dôvodu boli zavedené skriptovacie jazyky na strane klienta, ktoré prinášajú interaktivitu užívateľského rozhrania. Dynamické prvky stránky sú implementované napríklad v jazyku Java Script alebo AJAX, ktorý umožňuje spoluprácu skriptov na strane klienta so serverovou časťou aplikácie. Pre implementáciu aplikačnej logiky na strane servera sa používa technológia spoločnosti Microsoft ASP.NET alebo veľmi obľúbená open source technológia PHP.

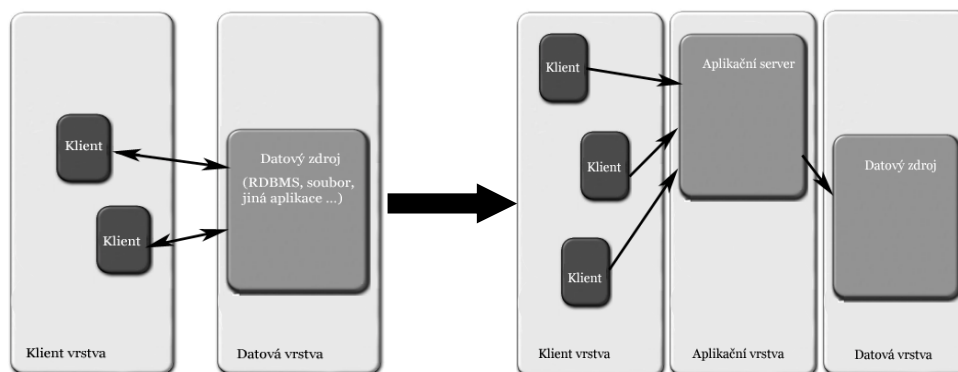
Dáta sú webovej aplikácii sprístupnené na dátových serveroch prostredníctvom technológií SQL databázových serverov.

## 3.2 Trojvrstvová architektúra

### 3.2.1 Dvojvrstvová verus Trojvrstvová architektúra

Dvojvrstvová architektúra vychádza z modelu aplikácie, ktorá pozostáva z dvoch vrstiev. Ide o vrstvu dátovú, ktorá sprístupňuje dáta v databáze alebo súboroch pre klientskú vrstvu. Klientská vrstva obsahuje aplikačnú logiku a zabezpečuje prezentáciu užívateľovi. Nevýhoda dvojvrstvovej architektúry sa prejavila pri implementácii zložitejších aplikácií, ktoré so sebou priniesli rastúce nároky na výkon klientských počítačov. Ďalším faktorom, ktorý prispel k vytvoreniu trojvrstvovej architektúry boli rastúce požiadavky užívateľov na flexibilitu a aktualizáciu aplikácií. Potreba pružne reagovať na požiadavky zákazníkov a rozvoj business aplikácií si vyžadujú väčšie nároky na výkon klientských zariadení. Tento problém rieši prídanie ďalšej tretej vrstvy.

Trojvrstvová architektúra je priamym nástupcom architektúry dvojvrstvovej. Vytvorenie ďalšej, aplikačnej vrstvy umožnilo presunúť logiku celej aplikácie na stranu serveru, a tak odbremeniť klientskú časť, pretože celý výpočtový výkon aplikácie bol presunutý na stranu výkonného serveru. Táto zmena so sebou priniesla možnosti lepšieho zdieľania, správy a dostupnosti aplikácie. Presun aplikačnej logiky aplikácie na aplikačný server redukoval nároky na dátový prenos. Takýto dátový prenos sa obmedzil len na transfer medzi aplikačným a dátovým serverom, čo znížilo nárok na prenos medzi prezentačnou(klientskou) vrstvou a celou aplikáciou na strane serveru, na rozdiel od dvojvrstvovej architektúry, kde bol nutný prenos jak dát tak interakcia užívateľa medzi klientom a serverom. Táto evolúcia sprístupnila takéto aplikácie aj užívateľom s obmedzenou prenosovou rýchlosťou pripojenia.



Obrázok 2: Dvojvrstvová vs. Trojvrstvová architektúra <sup>3</sup>

<sup>3</sup> Flex3G : Ekonomický webový portál <http://www.flexis.sk/main.php?page=flexG3tech>

### **3.2.2 Prezentáčná vrstva, tenký klient**

Klientská časť aplikácie zabezpečuje prezentáciu a interakciu užívateľa s aplikáciou. Táto časť aplikácie sa nazýva aj ako tenký klient (Thin client). Tento názov bol odvodený od jeho predchodcu v dvojvrstvovej architektúre, nazývaný ako tlstý klient. Tieto názvy vychádzajú zo samotnej podstaty jednotlivých vrstiev. Tenký klient teda neobsahuje žiadnu aplikačnú logiku. Tenký klient prístupuje k aplikačnému serveru, ktorý mu logiku celej aplikácie sprostredkúva. Naopak tlstý klient v dvojvrstvovej architektúre obsahuje väčšinu aplikačnej logiky. V skutočnosti musí obsahovať nejakú logiku aj tenký klient, avšak ide o zanedbateľnú časť oproti tlstému klientovi.

Príklad tenkého klienta môžeme určiť v prípade webových aplikácií. Webová aplikácia využíva webový prehliadač ako tenkého klienta. Komunikáciu medzi klientskou a aplikačnou vrstvou zabezpečuje prenos HTML či XML dokumentov s podporou CSS a JavaScriptov, prostredníctvom komunikačného protokolu HTTP. Jedná sa teda o komunikáciu klient - server.

### **3.2.3 Dátová vrstva**

V prípade dátovej vrstvy nedošlo k žiadnym zmenám v evolúcii k trojvrstvovému modelu. Táto vrstva poskytuje dátovú základňu aplikačnej vrstve, ktorá tieto dáta využíva. Dátová vrstva môže poskytovať dáta či už z relačnej databáze, súborového systému, webovej služby alebo inej aplikácie.

### **3.2.4 Aplikačná vrstva**

Aplikačná vrstva tvorí jadro aplikačnej logiky celej aplikácie. Táto aplikačná vrstva zabezpečuje prístup k dátam, prácu s dátami a ich vystavenie vo vhodnom formáte (HTML,XML) prezentačnej vrstve. Ako najjednoduchšiu realizáciu aplikačnej vrstvy môžeme považovať webový server s CGI (Common Gateway Interface) rozhraním a serverovými skriptami (PHP). Rozhranie CGI slúži na volanie externých programov webovým serverom, ako napríklad volanie funkcií z DLL(Dynamic Link Library) v technológii ASP.NET od firmy Microsoft.

## **3.3 Technológia .NET**

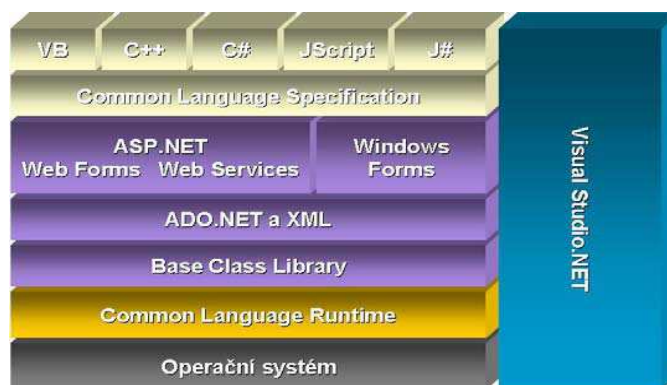
### **3.3.1 Úvod do technológie .NET**

Technológia .NET je platforma vyvíjaná spoločnosťou Microsoft. Účelom technológie .NET je zjednodušiť a urýchliť vývoj aplikácií. Platformu .NET tvorí množina technológií, ktorá je dostupná ako ucelený rámec pod názvom .NET Framework. Táto platforma je určená najmä pre prostredie operačných systémov Windows.

Architektúra .NET Framework pozostáva z jednotlivých častí, ktoré na seba funkčne nadväzujú. Na najnižšej úrovni sa nachádza CLR (Common Language Runtime), ktorý realizuje základnú infraštruktúru na ktorej je celý rámec vytvorený. CLR je zdieľaný všetkými aplikáciami postavenými na technológii .NET Frameworku. Dôležitým prvkom CLR je podpora spoločného typového systému CTS (Common Type System). Implementácia CLR teda zahrnuje mechanizmus zabezpečujúci typovú správnosť. Ďalšie vlastnosti, ktoré zahrnuje implementácia CLR je automatický manažment pamäti. Nad vrstvou CLR sa nachádzajú hierarchicky usporiadané knižnice, ktoré sú rozdelené do menných priestorov. Základ tvorí knižnica Base Class Library, nad ktorou je knižnica pre prístup k dátam ADO.NET a knižnica pre prácu s XML súbormi.

Nasledujúca vrstva sa skladá zo skupiny knižníc, ktoré podporujú a uľahčujú implementáciu užívateľského rozhrania. Je rozdelená do dvoch skupín, a to na knižnice podporujúce webové aplikácie ASP.NET a knižnice podporujúce klasické aplikácie založené na WinForms v prostredí Windows. Poslednú vrstvu tvorí množina programovacích jazykov. Základné vlastnosti tejto vrstvy definuje CLS (Common Language Specification). V prostredí .NET sú zdrojové súbory napísané v ľubovoľnom programovacom jazyku kompilované do jazyka MSIL (Microsoft Intermediate Language). V prípade že má byť takáto aplikácia spustená, systém detekuje aplikáciu v MSIL a spustí kompiláciu, ktorá vygeneruje príkazy cieľovej platformy.

Jedným z hlavných prínosov technológie .NET je podpora rôznych programovacích jazykov, čo prináša vývojárom nezávislosť v použití toho ktorého programovacieho jazyka, ktorý CLR podporuje. Táto vlastnosť umožňuje vývojárom lepšiu tímovú integráciu pri vývoji rozsiahlejších projektov. V súčasnej dobe firma Microsoft podporuje päť programovacích jazykov: Visual Basic, C++, C#, Jscript, J# . Takáto množina nie je uzavretá a ktorýkoľvek výrobca ju môže rozšíriť o nové. Táto škála programovacích jazykov je v súčasnej dobe rozšírená aj o mutácie programovacích jazykov Perl a Python.



Obrázok 3: .NET Framework architektúra<sup>4</sup>

<sup>4</sup> Microsoft: webové stránky spoločnosti MS, dostupné z <http://msdn.microsoft.com/en-us/netframework/default.aspx>.

### 3.3.2 Webové aplikácie v ASP.NET

Pre vývoj webových aplikácií v technológii .NET je definovaná knižnica ASP.NET (Active Server Pages). Názov tejto technológie vychádza zo staršej technológie ASP pre vývoj webu, napriek tomu je technológia ASP.NET odlišná od jej predchodcu. Táto technológia uľahčuje prechod od klasických aplikácií pre Windows do prostredia webu. Webové stránky sú poskladané z objektov a ovládacích prvkov, ktorým môžeme priradiť vlastnosti a zachytávať na nich udalosti, podobne ako pri vývoji klasických WinForms aplikácií vo Windows.

Tak ako sa vykresľujú jednotlivé ovládacie prvky aplikácie do formulárov na obrazovku vo Windows, tak aj webové aplikácie generujú HTML kód, ktorý obsahuje webové ovládacie prvky. Takto vygenerovaný HTML súbor je zaslaný serverom do klientskej aplikácie, respektíve webového prehliadača prostredníctvom protokolu HTTP. Pretože webový protokol HTTP je sám o sebe bezstavový, je potrebné zachovať kontext medzi jednotlivými požiadavkami. ASP.NET tento problém rieši pomocou dvoch základných techník, kombináciou HTML a JavaScriptu.

**ViewState** uchováva informácie medzi opakovaným odosielaním formulárov na server v zakódovanom tvare. Výhodou takejto techniky je použitie len HTML, bez žiadnej špeciálnej podpory na strane klienta ani na strane serveru. Technika ViewState prenáša medzi serverom a klientom väčší objem dát.

**SessionState** ukladá všetky informácie na strane serveru a predáva klientovi jednoznačný identifikátor vo forme cookie alebo ako súčasť URL. Takáto technika na jednej strane umožní prenos menšieho počtu dát, no kladie vyššie nároky na výkon serveru. Výhodou je ukladanie informácií do samostatného procesu alebo na SQL server, čo umožní zachovanie dát aj po reštarte serveru.

Technológia ASP.NET zastáva významnú pozíciu na poli vývoja webových aplikácií a prináša hneď niekoľko výhod:

- Kód je na rozdiel od iných technológií kompilovaný, preto umožňuje aplikácii fungovať rýchlejšie a identifikovať množstvo chýb pri implementácii.
- Užívateľsky definované prvky je možno použiť ako šablóny, čím sa redukuje duplicitný kód.
- Podobný prístup ako pri vývoji formulárových aplikácií vo Windows, čo zjednodušuje prechod medzi jednotlivými vývojovými prostrediami.
- Široké spektrum ovládacích prvkov a knižníc tried zrýchľuje vývoj aplikácie.
- Technológia .NET podporuje veľké množstvo programovacích jazykov.
- Webové aplikácie nie sú závislé na operačných systémoch (Windows, Linux), webových prehliadačoch (IE, Firefox, Opera) ani na webových serveroch (IIS, Apache).

## 3.4 XML a webové lužby

XML znamená eXtensible Markup Language, v preklade rozšíriteľný značkovací jazyk, ktorý bol vyvinutý a štandardizovaný konzorciom W3C (World Wide Web Consortium) ako pokračovanie jazyka SGML a HTML. Umožňuje jednoduché vytváranie konkrétnych značkovacích jazykov na rôzne účely a široké spektrum rôznych typov údajov.

Jazyk je určený predovšetkým na výmenu údajov medzi aplikáciami a na publikovanie dokumentov. Jazyk umožňuje popísať štruktúru dokumentu z hľadiska vecného obsahu jednotlivých častí a nezaobera sa sám o sebe vzhľadom dokumentu alebo jeho časti. Prezentácia dokumentu (vzhľadu) sa potom definuje pripojeným štýlom. Ďalšou možnosťou je pomocou rôznych štýlov vykonať transformáciu do iného typu dokumentu alebo do inej štruktúry XML.

Pôvodný jazyk na publikovanie HTML už prestal vyhovovať predovšetkým pre svoju zložitosť, ktorá vznikla jeho postupným (a svojvoľným) rozširovaním. Jazyk XML nemá žiadne preddefinované značky (tagy, názvy jednotlivých elementov) a tiež jeho syntax je podstatne prísnejšia ako syntax HTML.

Sila XML je najmä v jeho hierarchickej štruktúre a pomerne jednoduchom spôsobe kódovania. Umožňuje popisovať - označovať ľubovoľné dáta a prenášať ich medzi rôznymi aplikáciami a platformami. Hlavnou ideou XML je oddelenie obsahu a dizajnu dát. Spojovacím mostom je XSL, ktorý umožňuje vytvárať množstvo výstupných formátov.

XML je formát súboru obsahujúci dáta. Zvláštnym ho robí to, že je široko prijímaný ako štandard so schopnosťou existencie na akejkoľvek platforme a používaný masívnym, stále rastúcim počtom aplikácií a programovacích jazykov.

XML znamená Extensible Markup Language (rozšírený jazyk so značkami). Jeho špecifikácia je popísaná s odporúčením konzorcia World Wide Web Consortium. Jazyk XML nám dovoľuje vytvárať štruktúrované dokumenty veľmi flexibilným spôsobom. Dá sa použiť k vytváraniu dokumentov, ktoré sa zdanlivo podobajú dokumentom HTML. Jazyk XML sa však od HTML líši a používa sa k zásadne iným účelom. Táto podobnosť pochádza od rovnakých rodičov. Obidva jazyky sú odvodené od štandardu definície jazyka SGML. Ten sám o sebe nie je žiadny jazyk, ale iba spôsob definovania jazykov vyvinutých podľa jeho obecných princípov. Existuje tu však dôležitý rozdiel v tom, akým spôsobom sú jazyky XML a HTML odvodené od SGML. XML je podmnožina SGML, ľahšia verzia, ktorá bola oproti SGML zjednodušená, aby umožňovala použitie na sieti WWW, ale aj ako formát výmeny dát. Toto zjednodušenie robí zápis dokumentov vychádzajúcich z XML jednoduchším. Oproti tomu jazyk HTML je aplikáciou SGML. Je to teda konkrétny jazyk, ktorý dodržiava štandard SGML.

XML a HTML v skutočnosti slúžia veľmi odlišným účelom. Dokument HTML obsahuje značky indukujúce, ako má byť dokument naformátovaný, neobsahuje však žiadne informácie o vlastnom obsahu. Inými slovami HTML obsahuje iba informácie o zobrazení dát a neobsahuje žiadne údaje o

tom, o aké dáta sa jedná. XML sa pokúša oddeliť obsah a jeho reprezentáciu. Dokumenty XML obsahujú informácie v štruktúrovanej forme, ktorá sa zdanlivo veľmi podobá HTML. Dôležitým rozdielom je fakt, že môžeme definovať svoje vlastné značky, ktoré určujú štruktúru dát.

Dokumenty XML sa dajú previesť do všetkých ostatných variantov XML tým, že sa k dokumentu priradí štýl v jazyku XSL (Extensible Style Language – rozšírený jazyk štýlov. Štýl bude obsahovať informácie o tom, ako túto stránku zobrazit'. Každý dokument XML sa skladá z kombinácie dát značiek a znakov. Značky dávajú dokumentu XML štruktúru, zatiaľ čo znaky predstavujú iba vlastný obsah. Všetky dokumenty XML spĺňajúce špecifikáciu XML musia dodržiavať požiadavky DTD, potom sú považované za správne štruktúrované.

## 3.5 Vývojový model

### 3.5.1 Proces vývoja softwaru

Vývoj softwaru môžeme definovať ako proces, ktorého kroky vedú k vytvoreniu plne funkčnej aplikácii na základe stanovených cieľov. Prvý krok tohto procesu tvorí analýza potrieb zákazníkov resp. užívateľov. Na základe definície takýchto potrieb sa vytvorí predbežná predstava o aplikácii. Samotný proces vývoja aplikácie začína dekompozíciou celého projektu. Ide o rozloženie celej aplikácie na jednotlivé podproblémy, ktoré sú jednoduchšie riešiteľné. Je však dôležité zabezpečiť následnú funkčnosť systému ako celku, zloženého z jednotlivito riešených podsystémov.

Vývoj softwaru špecifikujú rôzne prístupy, ktoré sú popísané modelmi životného cyklu. Takýto model popisuje jednotlivé etapy vývoja:

- Analýza a špecifikácia požiadaviek
- Návrh
- Implementácia
- Integrácia a testovanie
- Použitie a údržba

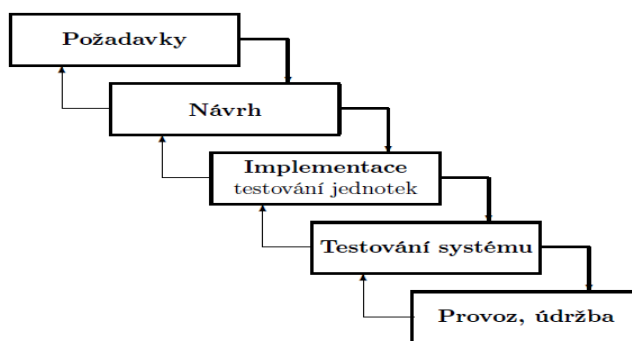
Najväčší podiel vynaloženého úsilia v tomto procese má etapa použitia a údržby, preto je veľmi dôležité klásť dôraz na čo najdokonalejšiu analýzu a návrh. Tieto tvoria základný kameň úspechu celej aplikácie.

## 3.5.2 Modely životného cyklu softwaru

Model životného cyklu softwaru definuje jednotlivé kroky, ktoré je nutné vykonať a ich časovú následnosť. Nedefinuje však dĺžku trvania jednotlivých krokov ani ich rozsah. Každá etapa musí byť definovaná tak, aby bolo možné vyhodnotiť jej správnosť.

Základným modelom životného cyklu softwaru je vodopádový model. Tento model je použiteľný v prípade, že požiadavky na software sú presne a vopred známe. V opačnom prípade je lepšie použiť niektorý z modelov, ktoré počítajú s neskorším spresnením požiadaviek na software.

**Vodopádový model:** Jednotlivé etapy sú zoradené za sebou. Každá etapa začína až po ukončení predchádzajúcej. Nevýhodou tohto modelu je, že ak užívateľ nepresne definuje svoje požiadavky v začiatkových fázach, tak po dokončení aplikácie a identifikácii jej nedostatkov je potrebné celý proces opakovať od začiatku.



Obrázok 4: Vodopádový model<sup>5</sup>

**Iteratívny model:** Tento model odstraňuje nedostatky vodopádového modelu. Podstatná výhoda oproti vodopádovému modelu spočíva v tom že aplikácia je dostupná užívateľovi ako spustiteľná čiastočne fungujúca verzia. Tento prístup umožňuje identifikovať nedostatky už počas vývoja. Ide vlastne o iteráciu vodopádového modelu v priebehu času. Takýto prístup však môže zhoršiť celú štruktúru systému, tým že špecifické požiadavky nie sú presne definované od začiatku vývoja softwaru.

**Inkrementálny model:** Tento model vychádza konceptuálne z iteratívneho modelu. Na základe špecifikácie celého systému sa určujú ucelené časti systému, ktoré sa vytvárajú a odovzdávajú užívateľovi postupne. Výhodou modelu je že celý systém môže byť navrhnutý lepšie, pretože eliminuje negatívne dopady neskorších požiadaviek užívateľa.

**Špirálový model:** Tento model využíva v procese vývoja softwaru prototypy a kladie veľký dôraz na analýzu rizík. Jednotlivé etapy sa vo vývoji softwaru opakujú, vždy na vyššej úrovni problematiky. Na rozdiel od iteratívneho modelu, kde sa vytvára čiastočne fungujúca verzia, v procese špirálového modelu sa vytvára prototyp. Ten po odskúšaní a analýze sa zahodí a vytvára sa nový, obsiahlejší prototyp. Tento model kladie veľké časové nároky na vývoj softvéru, čo

<sup>5</sup> KŘENA, B. – KOČÍ, R.: *Úvod do softwarového inženýrství*, opora k predmetu IUS, FIT VUT Brno, 2006.

znižuje riziko nedokonalosti a špatnej funkčnosti po dokončení, a tým ušetrí čas a náklady na prípadné opravy a údržbu.

## 3.6 Objektovo orientovaný návrh

### 3.6.1 Objektovo orientovaný model

V programovacích metódach sa kladie primárny dôraz na správne a efektívne využitie mechanizmov konkrétneho jazyka. Oproti tomu návrhové metódy kladú dôraz na správne a efektívne štruktúrovanie zložitého systému.

Objektovo orientovaný návrh je metóda návrhu zahrňujúca proces objektovo orientovanej dekompozície a notáciu zobrazenie logického a fyzikálneho ako aj statického a dynamického modelu navrhovaného systému.

Pre celý objektovo orientovaný návrh je pojmovým rámcom objektový model. Tento model pozostáva zo štyroch hlavných prvkov:

- Abstrakcia
- Zapuzdrenie
- Modularita
- Hierarchia

Objektovo orientovaná analýza a návrh je snáď momentálne jedinou metódou, ktorá sa dá použiť na prekonanie prekážok, ktoré predstavuje zložitosť v rozsiahlych systémoch. Ak hovoríme o objektovom návrhu, je dôležité ozrejmiť niekoľko základných pojmov:

**Analýza:** dôraz sa kladie na to, čo má systém robiť. Cieľom analýzy je pochopiť, čo treba vytvoriť nezávisle od implementácie a technológie. V etape analýzy sa uskutočňuje transformácia funkčných požiadaviek do koncepcie požadovaného programového systému. Modelovacie prostriedky pre etapu analýzy obsahujú diagramy pre opis požiadaviek používateľa, jeho aktivity, a diagramy tried.

**Návrh:** predstavuje zušľachtenie analýzy. Modelovanie návrhu znamená definovanie toho, ako je štruktúrovaný a napísaný zdrojový kód programu. Modelovacie prostriedky pre etapu návrhu obsahujú diagramy balíkov (packages) tried (classes), a iné útvary, ktoré prezentujú záujmovú oblasť alebo riešený problém.

**Model** - reprezentuje podstatu komplexného problému, z ktorého sú vylúčené nepodstatné detaily. Vylúčenie nepodstatných detailov umožňuje pochopiť podstatu problému aj spôsob jeho

riešenia. Model obsahuje jeden alebo viac diagramov a jeden alebo viac modelových prvkov (model elements).

**Diagramy** - poskytujú spôsob vizualizácie modelových prvkov. Rozdielne diagramy predstavujú rozdielne pohľady na vyvíjaný systém. Pre popis systému je možné použiť viacero typov diagramov niektorého z modelovacích jazykov, napr. UML, ktorý rozlišuje medzi štruktúrnym a behaviorálnym pohľadom na modelovaný systém. My sa budeme zaoberať práve štruktúrnym pohľadom prostredníctvom diagramu tried (Class diagram).

### 3.6.2 Diagram tried

Diagram tried zobrazuje statickú štruktúru systému. Primárne je určený k popisu tried a ich vzťahov, avšak v diagrame tried môžu byť zobrazené i rozhrania (interface), balíky (package), objekty (objects) a taktiež súvisiace vzájomné väzby. Výstižnejšie by bolo preto nazývať tento diagram diagramom statickej štruktúry, používanie termínu diagram tried sa však v tomto kontexte ustálilo.

### 3.6.3 Trieda

Trieda je popisom množiny objektov, ktoré zdieľajú rovnaké atribúty, metódy, vzťahy a sémantiku, je teda prvkom modelu s vyššou mierou abstrakcie než prvok modelu nazývaný objekt. Trieda je v rámci diagramu jednoznačne určená svojim názvom. Pre triedu je možné definovať atribúty a operácie, ktorých názvy musia byť jednoznačne určené v rámci triedy. Reprezentáciou triedy je obdĺžnik rozdelený na tri časti pre zobrazenie atribútov a operácií.

### 3.6.4 Asociačná väzba

Asociačná väzba medzi triedami je vyjadrená abstrakciou vzťahu medzi objektmi (inštanciami tried). Asociácia je špecifikovaná viacerými vlastnosťami, z ktorých niektoré sú viazané priamo na samostatnú väzbu (napríklad názov väzby), ostatné k zakončeniu väzby (napríklad rola). Asociácia je určená svojim názvom, ktorému môže byť priradený smer, ktorý uľahčuje jeho interpretáciu. Väzbu asociácie je možné uviesť ako orientovanú (navigability), pričom neorientovaná väzba je považovaná za obojsmernú (teda dva jednosmerné vzťahy). Triedy v asociácii môžu voči sebe vystupovať v roliach (role). V asociácii je možné určiť násobnosť väzby (multiplicity), ktorá vyjadruje počet možných väzieb objektov tried v asociácii (0, 0..1, 0..\*, 1, 1..\*, \*, M..N, ...).

### 3.6.5 Agregáčné väzby

Agregácia je vyjadrením abstrakcie vzťahu medzi objektmi, ktorý charakterom zodpovedá vzťahu celku a časti. Agregácia je špeciálnym prípadom asociácie, a preto všetky vyššie uvedené vlastnosti týkajúce sa asociácie sú aplikovateľné i na agregáčnú väzbu. Jazyk UML rozlišuje medzi dvoma

typmi agregácií, jednoduchou agregáciou a kompozíciou. Väzba kompozície je silnejšia než väzba jednoduchej agregácie, požaduje vlastníctvo objektu agregovanej triedy (tzv. časť) a vyjadruje možnosť objektu agregujúcej triedy (tzv. celok) vytvoriť a zrušiť objekty agregovaných tried (častí).

### 3.6.6 UML

Unified Modeling Language alebo UML je v softvérovom inžinierstve grafický jazyk na vizualizáciu, špecifikáciu, navrhovanie a dokumentáciu programových systémov. UML ponúka štandardný spôsob zápisu tak návrhov systémov vrátane konceptuálnych prvkov ako sú business procesy a systémové funkcie, tak konkrétnych prvkov ako sú príkazy programovacieho jazyka, databázové schémy a znovu použiteľné programové komponenty.

UML podporuje objektovo orientovaný prístup k analýze, návrhu a popisu programových systémov. UML neobsahuje spôsob, ako sa má používať, ani neobsahuje metodiku, ako analyzovať, špecifikovať alebo navrhovať programové systémy.

Štandard UML definuje štandardizačná skupina Objekt Management Group (OMG). UML je podporným nástrojom na komunikáciu medzi vývojármi a na zaznamenávanie myšlienok a návrhov. Do diagramov sa kreslia len veci podstatné na grafické vyjadrenie návrhu alebo časti návrhu predtým, ako sa začne programovať. Dôležitá je zrozumiteľnosť, rýchlosť nakreslenia a možnosť jednoduchej zmeny alebo alternatív riešenia.

Cieľom je zaznamenať kompletný návrh alebo kompletnú realizáciu. Pri kreslení návrhu by mal analytik obsiahnuť všetky prvky tak, aby programátor bol schopný vytvoriť program bez veľkého premýšľania nad vecnou oblasťou (pre programátora by nemala vzniknúť potreba konzultácie s užívateľom). Pri kreslení detailných návrhov sa obvykle používajú špecializované programy (CASE), ktoré sú schopné zdieľať informácie medzi jednotlivými modelmi a kontrolovať konzistenciu návrhu. Pri dokumentácii programu sa často používajú nástroje na generovanie diagramov z vlastného kódu aplikácie.

Pri opačnom použití vývojár nakreslí UML diagramy, z ktorých sa vygeneruje priamo spustiteľný kód. Toto vyžaduje špecializované nástroje a veľmi presné vyjadrovanie v UML diagramoch. V tejto súvislosti sa veľmi často používa pojem Model Driven Architecture (MDA), čo je ďalší štandard skupiny OMG, ktorý sa snaží štandardizovať použitie UML ako programovacieho jazyka.

## 4 Analýza a návrh

Je dôležité analyzovať všetky dostupné zdroje informácií a pretransformovať na výstupné informácie, ktoré slúžia ako základ pre implementáciu aplikácie. Na základe špecifikovaných požiadaviek užívateľov je možné navrhnuť databázový systém aplikácie a užívateľské rozhranie.

### 4.1 Požiadavky na systém

Aby webová aplikácia plnila svoj účel, je potrebné definovať skupinu užívateľov, ktorý budú používať danú aplikáciu. Predpokladá sa že užívateľ systému má potrebné znalosti z oblasti investovania na kapitálových trhoch, a preto je potrebné zamerať sa na také informácie a nástroje, ktoré užívateľovi poskytujú pridanú hodnotu.

#### 4.1.1 Informácie

Webová aplikácia má za úlohu poskytovať konkrétne a reálne dáta o cenných papieroch a ich obchodných výsledkoch na burzách. Informácie, ktoré webová aplikácia poskytuje, sú formulované pre užívateľa, ktorý má potrebné znalosti na ich použitie. Preto je potrebné aby pre užívateľov, ktorým dané znalosti chýbajú, boli dostupné zdroje pre ich získanie. Poskytované informácie musia zahŕňať podrobné informácie o cenných papieroch obchodovaných na burzách a ich aktuálne a historické výsledky. Tie sú dôležité na analyzovanie vývoja cien jednotlivých finančných nástrojov.

#### 4.1.2 Zdroje informácií

Informácie o cenných papieroch a obchodné výsledky zverejňuje každá burza cenných papierov. Existuje veľa rôznych spoločností, ako napríklad Xintigie, ktoré poskytujú najrôznejšie štatistiky a výsledky z ekonomiky a finančných trhov prostredníctvom webových služieb. Nevýhodou takýchto informačných zdrojov je že sú vysoko spoplatnené, pretože sú poskytované na komerčné účely. Pre nekomerčné účely sú dobre dostupné informácie vo forme "CSV súborov", ktoré poskytuje centrálna banka (ČNB, ECB, NBS) alebo burza cenných papierov (PSE, NYSE). V základnej verzii tejto webovej aplikácie budeme vychádzať výhradne z voľne dostupných informačných zdrojov pražskej burzy cenných papierov (Prague Stock Exchange), ktoré sú k dispozícii na <ftp.pse.cz/ft>.

### 4.1.3 Funkcia aplikácie

Zmyslom každej webovej aplikácie je priniesť užívateľovi, síce už verejne dostupné, informácie ale s určitou pridanou hodnotou. Hlavnou funkciou tejto webovej aplikácie je analyzovať jednotlivé cenné papiere a simulovať obchodovanie s nimi. Súbor finančných nástrojov definujeme ako portfólio. Základnou funkciou tejto aplikácie je vytvoriť a spravovať portfólio investičných nástrojov resp. simulovať ich reálne obchodovanie na burze cenných papierov. Finančné a obchodné dáta sú veľmi obsiahle a podrobné, preto je treba užívateľom poskytnúť takú formu prístupu k nim, aby ich čítanie bolo jednoduchšie. Na tento účel slúži zobrazenie dát formou grafov a prehľadných tabuliek.

### 4.1.4 Užívatelia

K aplikácii môžu pristupovať dva typy užívateľov. V prvom rade neregistrovaný užívateľ, ktorý má prístup k informáciám. Ďalej môže pristupovať registrovaný užívateľ, ktorý má možnosť okrem získavania informácií aj využívať nástroje na ich spracovanie. Povolené funkcie jednotlivým užívateľom:

#### **Neregistrovaný užívateľ môže:**

- Prezerat' povolené stránky systému
- Zaregistrovať sa
- Prezerat' zoznam cenných papierov
- Prezerat' podrobné informácie o cennom papieri
- Sledovat' obchodné výsledky jednotlivých nástrojov

#### **Registrovaný užívateľ môže:**

- Všetko, čo je povolené neregistrovanému užívateľovi
- Prihlásiť sa do systému, odhlásiť sa zo systému
- Po prihlásení do systému prezrieť vlastný profil, zmeniť svoje prihlasovacie údaje
- Písať do diskusného fóra, pridať novú podskupinu (tému) do diskusných skupín
- Vytvoriť a editovať portfólio
- Prezerat' obchodné výsledky portfólia

## 4.2 Prípady užitia

Prípad použitia (Use case) popisuje správanie systému v rôznych situáciách vo forme reakcií systému na požiadavky jedného z užívateľov systému.

Prípady užitia sa používajú v systémovej inžinierstve pre zachytenie funkčných požiadaviek a popisujú vzťahy medzi systémom a užívateľom systému vo forme postupnosti jednoduchých krokov. Užívateľ systému sa nazýva primárny aktér a vyvoláva interakciu so systémom za účelom dosiahnutia určitého cieľa. Každý prípad užitia predstavuje sériu činnosti a udalosti z pohľadu aktéra.

### 4.2.1 Popis prípadov užitia

Neregistrovaný užívateľ sa môže prihlásiť do systému. Ak nie zaregistrovaný musí sa registrovať. Neprihlásený užívateľ môže sledovať komentáre k jednotlivému portfóliu a sledovať vývoj investičných titulov. Registrovaný užívateľ má možnosť sa odhlásiť, meniť registračné údaje alebo zrušiť registráciu. Ďalej má možnosť vytvárať portfólio a pracovať ním.

### 4.2.2 Model prípadov užitia

#### Registrovaný užívateľ

- Prihlásiť / Odhlásiť
- Pridať Portfólio / Zrušiť Portfólio
- Pridať Účet / Odstrániť účet / Predať účet
- Publikovať portfólio
- Pridať komentár
- Diskutovať na fóre

#### Neregistrovaný užívateľ

- Registrovať
- Prezerat' investičné nástroje
- Prezerat' články

## 4.3 Databázový model

Databázový model databázového systému je množina syntaktických pravidiel a množina metód, ktoré slúžia na modelovanie databázového systému. Databázový model vznikne na základe dátovej a funkčnej analýzy systému. Výsledkom dátovej analýzy v databázovom modeli je dátový model (model bázy dát) systému.

Malá zmena požiadaviek na navrhnutý databázový systém vedie obvykle k veľkej zmene aplikácie. Hlavnou úlohou projektanta informačného systému podľa štruktúrovanej analýzy je navrhnuť entitno relačný (ER) model, funkčný model a model riadenia systému. S realizáciou daného modelu systému súvisí entitno relačný diagram (ERD), diagram dátových tokov (DFD - data flow diagram) a diagram stavov a prechodov (STD - state transition diagram).

Najznámejšou a najčastejšie používanou modelovacou technikou pre návrh relačných databáz je entitne vzťahové modelovanie, jeho výsledkom je entitne vzťahový diagram (ER diagram alebo ERD). Vytváraný model a odpovedajúci diagram slúži ako podklad pre návrh databázy.

### 4.3.1 Analýza dát

Na základe požiadavkou na systém možno definovať typy dát ktoré budú potrebné. Je zrejmé že aplikácia bude pracovať s celými aj reálnymi číslami. Je potrebné aby aplikácia dokázala pracovať s menovými, percentuálnymi a textovými hodnotami. V prípade webovej aplikácie je nutné dobre definovať, ktoré dáta budú čerpané z webu a ktoré je nutné ukladať v databáze. V tejto aplikácii je nutné brať do úvahy nástroje na prácu s portfóliom. Ide o simuláciu obchodov, ktorá vyžaduje pracovať a ukladať vlastné dáta. Informácie o cenných papieroch a ich obchodných výsledkoch nie je potrebné ukladať, pretože sú dostupné prostredníctvom webu.

### 4.3.2 E-R Diagram

Aplikácia potrebuje ukladať informácie o registrovaných užívateľoch a ich registračných údajoch. Práve jeden užívateľ môže vlastniť 0 až n portfólií a každý užívateľ môže pridať 0 až n príspevkov práve k 1 portfóliu. Každé portfólio je zložené z 1 až n položiek investičných titulov. Informácie o hodnotách a tituloch sa získavajú z webových služieb.

ER diagram popisuje entity použité vo webovej aplikácii:

**Portfolio:** Táto tabuľka nesie informáciu o jednotlivých portfóliách, ktoré si užívateľ vytvorí. Slúži ako skupina investičných nástrojov, ktoré definuje užívateľ.

**Account:** Tabuľka nesie informácie o investičnom nástroji, ktorý bol pridaný do portfólia. Atribúty, ktoré uchováva popisujú obchodnú transakciu na finančnom trhu.

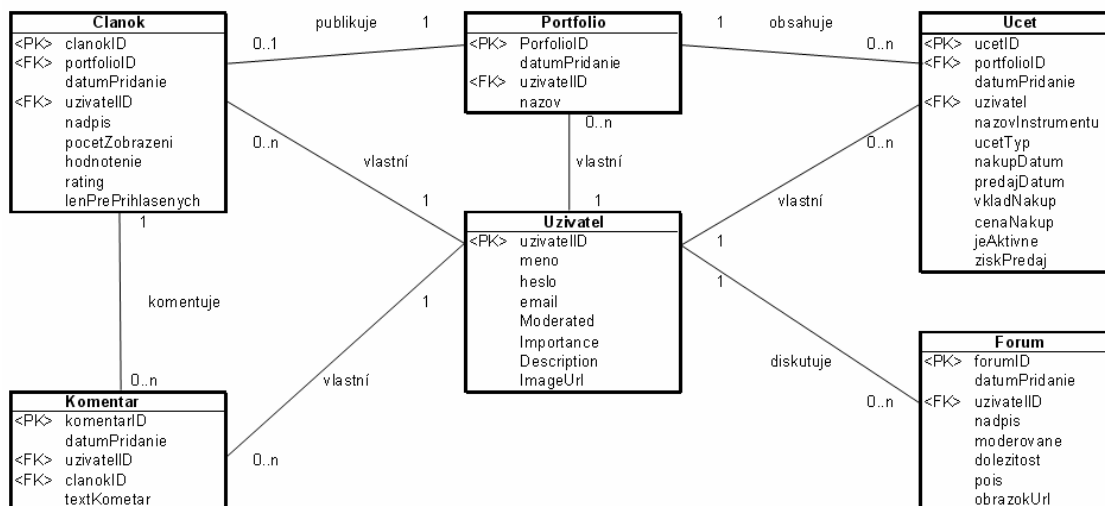
**Forums:** Tabuľka uchováva informácie potrebné pre správu diskusného fóra

**Articles:** Tabuľka je použitá na ukladanie jednotlivých článkov vo webovej aplikácii.

**Categories:** Tabuľka nesie dáta o kategóriách jednotlivých článkov v aplikácii

**Coments:** Uchováva informácie o komentároch k jednotlivým článkom

Pre správu a uchovávanie údajov o užívateľoch webovej aplikácie sú použité prednastavené tabuľky, ktoré aplikácia ASP.NET využíva.



**Obrázok 5:** Entitno Relačný diagram

### 4.3.3 Dostupné dáta na webe

Táto webová aplikácia pracuje s voľne dostupnými dátami a s webovými službami dostupnými na webových stránkach burzy cenných papierov . V základnej verzii sú dáta čerpané výhradne z pražskej burzy cenných papierov, dostupné na <ftp.pse.cz/ftp>. Aktuálne obchodné dáta sú dostupné v adresári “Results.ak/AK.csv“ a historické obchodné dáta sú dostupné v adresári “Results.ak/AKddmmrr.csv“ . Popis dát v súboroch “CSV“ je dostupný v súbore Popis\_AK.txt v adresári “Results.ak/“.

# 5 Implementácia systému

Táto kapitola obsahuje podrobný popis implementácie jednotlivých častí systému. Implementácia vychádza z podrobnej analýzy a návrhu celej aplikácie. Táto kapitola popisuje jednotlivé funkcie a časti systému, počnúc vytvorením databázy až po užívateľské rozhrnie, ktoré funguje ako transformátor dát na informácie.

## 5.1 Vytvorenie databázy

Na základe návrhu databázy je možné takúto databázu vytvoriť. Technológia ASP.NET poskytuje dobré nástroje pre prácu s dátami (ADO.NET) a spojenie s databázovým SQL serverom. Dátové typy, potrebné pre túto webovú aplikáciu, ktoré poskytuje SQL server sú:

DECIMAL: percentuálne hodnoty

MONEY: mena resp. cena cenných papierov

INT: počet obchodov, kusov cenných papierov

DATETIME: dátum nákupu, dátum splatnosti

VARCHAR: identifikácia cenných papierov ISIN<sup>6</sup>, BIC

Ďalším krokom je samotné vytvorenie databázových tabuliek v databáze SQL podľa návrhu databázy:

- mp\_Account
- mp\_Forum
- mp\_Articles

Pre prácu s jednotlivými tabuľkami je vhodné implementovať jednotlivé procedúry na spracovanie a vyhľadávanie dát v tabuľkách. Tieto procedúry sú potom využívané na prístup k dátam v databázových tabuľkách pre objekty dátovej vrstvy. Väzby medzi tabuľkami sú vyriešené pomocou cudzích kľúčov (FOREIGN KEY). Pomocou klauzule ON DELETE CASCADE je zabezpečené kaskádové zmazanie záznamov s rovnakou hodnotou cudzieho kľúča. Väzby medzi tabuľkami sú vyriešené pomocou cudzích kľúčov (FOREIGN KEY). Pomocou klauzule ON DELETE CASCADE je zabezpečené kaskádovo zmazanie záznamov s rovnakou hodnotou cudzieho kľúča.

Na úrovni SQL serveru je možné zaistiť niektoré požiadavky, ktoré sú na jednotlivé položky kladené. Napr. môžeme požadovať, aby určité položky boli vždy zadané. Ďalej môžeme chcieť, aby nejaká

---

<sup>6</sup> ISIN je identifikačný reťazec, ktorý presne definuje určitý cenný papier, vydáva PSE.

položka mala jedinečnú hodnotu (väčšinou takou položkou bude primárny kľúč), alebo mala nejakú hodnotu defaultnú, v prípade, že ju užívateľ nezadal. Tieto obmedzenia nazývame integritnými obmedzeniami. Poznáme niekoľko typov integritných obmedzení, ako napr:

DEFAULT <hodnota> – znamená, že sa použije uvedená hodnota, keď užívateľ nezadá žiadnu hodnotu

NOT NULL - hodnota musí byť povinne zadaná

PRIMARY KEY – primárny kľúč, jeho hodnota musí byť jedinečná

UNIQUE – jedinečná hodnota

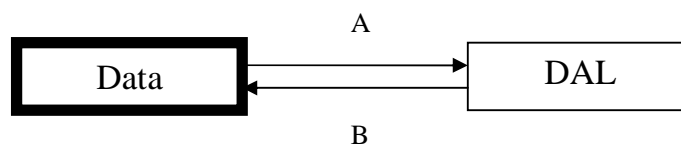
Po určení všetkých potrebných integritných obmedzení sa pomocou príkazu CREATE TABLE vytvoria všetky tabuľky.

## 5.2 Dizajn aplikácie

### 5.2.1 Dátová vrstva

Dátovú vrstvu môžeme definovať ako zdroj dát. Teda ide o databázový server alebo databázový súbor, ktorý je nosičom dát webovej aplikácie. Ide o základnú a najnižšiu vrstvu celého dizajnu aplikácie. Táto vrstva obsahuje databázový systém, ktorý poskytuje databázové tabuľky a jednotlivé procedúry, ktoré slúžia na naplnenie databázových tabuliek a prácu s dátami v týchto tabuľkách. Pre prácu s dátami na dátovej vrstve sa používa jazyk SQL.

**Rozhranie vrstvy:**



*Obrázok 6: Dátová vrstva*

Dátová vrstva komunikuje s vrstvou prístupu k dátam (ďalej len DAL). Výstupom procedúry sú získané dáta z databázovej tabuľky a tie sú poskytnuté vyššej vrstve, vrstve DAL. [rozhranie A]

Naopak dátová vrstva prijíma požiadavky vyššej vrstvy na poskytnutie dát alebo príkazy na zápis alebo zmenu dát v databázovej tabuľke. [rozhranie B]

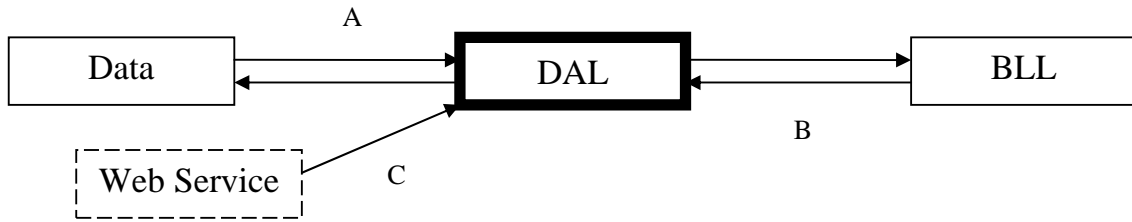
**Databázové Objekty:**

*Data Table:* **mp\_Account**

*Procedures:* **mp\_Portfolio\_InsertAccount** , **mp\_Portfolio\_GetAccountByType**

## 5.2.2 Vrstva prístupu k dátam DAL (Data Access Layer)

Vrstva DAL hrá významnú úlohu vo trojvrstvovom dizajne aplikácie. Úlohou tejto vrstvy je sprostredkovať logickej vrstve BLL potrebné dáta z databázy. Na jednej strane získava dáta z databázy, ktoré poskytuje vyššej logickej vrstve v potrebnom tvare a zložení. [Rozhranie A] Na druhej strane prijíma dáta od BLL, ktoré je potrebné uložiť resp. zapísať do databázy alebo požiadavky na dáta v určitom formáte, ktoré si BLL vyžiada od DAL. Tie následne DAL získava z databázy a pretransformuje ich do konkrétnej podoby. [Rozhranie B]



Obrázok 7: Vrstva prístupu k dátam

Táto vrstva sa skladá zo súboru tried (CLASS) objektov, ktoré disponujú rôznymi metódami pre sprostredkovanie dát z databázy alebo webovej služby. Tieto metódy využívajú na prenos dát špeciálne objekty triedy "Details". Ide o objektovú triedu, ktorá obsahuje rôzne atribúty v určitom formáte podľa typu objektivej triedy. Základné triedy, od ktorých sú objekty odvodené slúžia na definovanie spoločných konfiguračných atribútov a metód. Trieda SiteProvider sprístupňuje všetky triedy, ktoré prístupujú k dátam.

### Objekty:

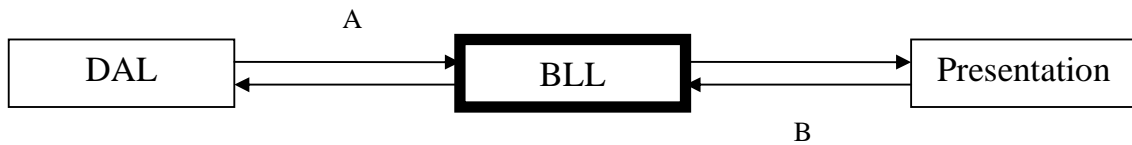
*Class:* (namespace)DAL.Portfolio: **.SqlPortfolioProvider** , **.PortfolioProvider**  
(namespace)DAL.StockEx: **.WebStockExProvider** , **.StockExProvider**  
*Common base Class:* (namespace).DAL **.SiteProvider()** , **.DataAccess()**  
*Method:* **GetAccountsByType()** [A] , **InsertAccount()** [B]  
**GetPSEResults()** [C]

*Class Details:* (namespace).DAL.Portfolio.AccountDetails  
(namespace).DAL.StockEx.InstrumentDetails

## 5.2.3 Logická vrstva BLL (Business Logical Layer)

Logická vrstva sa nachádza nad DAL vrstvou a pod prezentačnou vrstvou. Vrstva BLL je súbor tried objektov, ktorých úlohou je získať naformátované dáta v "Details object", použiť ich a vytvoriť konkrétne ucelené informácie, ktoré poskytuje prezentačnej vrstve. [Rozhranie A] Naopak BLL prijíma požiadavky na informácie od prezentačnej vrstvy alebo prijíma informácie, ktoré treba uložiť,

a posielala ich na spracovanie vrstve DAL. [Rozhranie B] Základnou triedou od ktorej sú triedy BLL odvodené, BizObject dáva všetkým objektom týchto tried spoločné atribúty a metódy.



Obrázok 8: Logická vrstva

#### Objekty:

*Class:* (namespace).BLL.Portfolio: **Account** : **Portfolio** : **BasePortfolio**

(namespace).BLL.StockEx: **Instrument** : **StockExchange** : **BaseStockExchange**

*Common base Class:* (namespace).BLL.**BizObject**

*Method:* **GetAccounts()** , **GetInstruments()** [A] , **DeleteAccount()** [B]

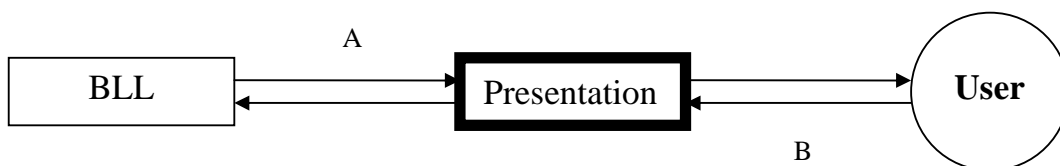
**GetPSEResults()** [C]

*Class Details:* (namespace).DAL.Portfolio.AccountDetails

(namespace).DAL.StockEx.InstrumentDetails

## 5.2.4 Prezentačná vrstva

Najvyššia vrstva je vrstva prezentačná. Úlohou tejto vrstvy je získať ucelené a logické informácie a tie poskytnúť prostredníctvom užívateľského rozhrania užívateľovi [Rozhranie A]. Naopak užívateľ môže špecifikovať požadované informácie od systému, alebo vkladať informácie do systému prostredníctvom webových formulárov. Tie potom prezentačná vrstva odošle objektu BLL informáciu, ten ju uloží do databázy prostredníctvom DAL [Rozhranie B]. Prezentačná vrstva využíva webové formuláre ku komunikácii s užívateľom. Vlastnosti a správanie týchto formulárov sú popísané v tzv. súboroch na pozadí. Jednotlivé webové formuláre sú prezentované prostredníctvom návrhového vzoru “Master Page File“.



Obrázok 8: Vrstva prezentácie užívateľovi

#### Objekty webových formulárov:

*Form:* (namespace).UI: **BrowseAccount.aspx** (**BrowseAccount.aspx.cs**) : **BasePage**

*Control:* (namespace).UI: **InstrumentListing.ascx** (**InstrumentListing.ascx.cs**)

: **BaseWebPaet**

## 5.3 Rozloženie aplikácie

### 5.3.1 Informačné celky

Celú webovú aplikáciu je možno rozdeliť tematicky, t.j. podľa zamerania informácií na niekoľko sekcií. Tieto sekcie sú dostupné ako hlavné položky menu navigácie. Sú to:

- Domov – úvodné informácie
- Investičné nástroje – aktuálne obchodné výsledky PSE
- Portfólio – práca s portfóliom
- Fórum – diskusie
- Pomocník – nápoveda
- O aplikácii – verzia, informačné zdroje, kontaktné informácie

### 5.3.2 Konfiguračné nastavenia aplikácie

Technológia ASP.NET poskytuje rôzne nástroje na konfiguráciu webovej aplikácie. Medzi takéto konfiguračné súbory patrí **web.sitemap**, ktorý popisuje navigáciu po webovom sídle, ďalej **web.config**, ktorý nesie základné konfiguračné údaje aplikácie ako napríklad spôsob autorizácie užívateľa, nastavenia kompilácie apod. Webová aplikácia obsahuje objektové triedy: Globals, ConfigSection, ktoré sú globálne a slúžia na konfiguráciu aplikácie ako napríklad “enabling Caching“ alebo poskytujú globálne vlastnosti a parametre ako napríklad maximálny počet riadkov na stranu, formát menových alebo percentuálnych hodnôt. Trieda Helpers a trieda CSVReader sú tridy objektov, ktoré slúžia ako nástroje na kompiláciu, lokalizačné nastavenia alebo čítanie “CSV“ súborov.

### 5.3.3 Adresár zdrojového kódu

Technológia ASP.NET ponúka preddefinované položky stromového adresára zdrojového kódu aplikácie. Stromová adresárová štruktúra je rozčlenená podľa metodológie trojvrstvovej architektúry:

- Root** – obsahuje všetku priečinky a zdrojové kódy aplikácie (Images, FCKEditor, Uploads)
  - UI – obsahuje zdrojové kódy užívateľského rozhranie (About.aspx, PxListing.ascx)
- App\_Code** – obsahuje všetky zdrojové kódy tried objektov, ktoré aplikácia využíva
  - BLL – triedy BLL vrstvy (Portfolio.cs, StockExchange.cs)
  - DAL – triedy DAL vrstvy (SqlPortfolioProvider.cs, WebStockExProvider.cs)
- App\_Data** – obsahuje databázové súbory (ASPNETDB.mdf, InstrumentType.xml)
- App\_WebReferences** – obsahuje odkazy na webové služby

**App\_GlobalResourcesData** – obsahuje globálne konštanty (SiteMap.resx)

**App\_LocalResourcesData** – obsahuje lokálne konštanty (Template.master.resx)

**bin** – obsahuje vložené knižnice nástrojov (AJAXToolKit, FCKEditor)

## 5.4 Uživateľské rozhranie

### 5.4.1 Registrácia

Neregistrovaný užívateľ má možnosť sa zaregistrovať po kliknutí na odkaz “Vytvoriť nový účet” na paneli prihlásenia. Ak tak neurobí má možnosť prezerat’ dostupné informácie, ale nemôže využívať nástroje portfólia. Druhou možnosťou je registrácia po varovnom oznámení o neautorizovanom prístupe. Pri registrácii je možné nastaviť si svoj vlastný užívateľský profil. Tento je možné upraviť po prihlásení ako autorizovaný užívateľ odkazom “Zmeniť profil” v panely prihlásenia

### 5.4.2 Autorizácia

Na prihlásenie do systému ako autorizovaný užívateľ slúži prihlasovací panel, ten umožňuje ďalej vyžiadať zabudnuté heslo alebo zapamätať si prihlasovacie údaje. Prístup na také časti, ktoré sú dostupné len pre registrovaných užívateľov, je chránený a vracia hlásenie o neplatnom prihlásení registrovaného užívateľa. K dispozícii dáva odkaz na prihlásenie alebo registráciu užívateľa.

### 5.4.3 Cenné papiere

Informácie o aktuálnych obchodných výsledkoch cenných papierov na burze sú prístupné v sekcii „Investičné nástroje“ v hlavnom menu aplikácie. Táto sekcia poskytuje výpis jednotlivých cenných papierov, ich popis, a aktuálne obchodné údaje. Aplikácia umožňuje obmedzenie výpisu cenných papierov podľa typu cenného papiera, v základnej verzii PSE. Každá položka nesie odkaz na zobrazenie detailov o vybranom cennom papieri. Tento prehľad obsahuje základné údaje, údaje o emitentovi a grafický prehľad vývoja ceny. Každá položka v prehľade obchodných výsledkov obsahuje samozrejme aktuálny kurz resp. cenu investičného nástroja a odkaz na jeho zaradenia do portfólia.

#### **5.4.4 Portfólio**

Táto sekcia dostupná z hlavného menu ako “Portfólio“ umožňuje výpis jednotlivých položiek zostaveného portfólia. Okrem ich aktuálnych hodnôt zobrazuje aj informácie o obchodnej transakcii resp. o simulovanej kúpe daného nástroja. Sekcia portfólio umožňuje pridávať nové nástroje do portfólia. Editácia jednotlivých položiek v tomto smere nemá zmysel. Užívateľ má možnosť odstrániť položku úplne alebo ju previesť do stavu ukončená, čo zachováva informácie o uzavretej obchodnej transakcii buď to so ziskom alebo stratou. Portfólio ponúka rôzne grafické informácie a štatistiky o vývoji a zložení celého portfólia ako aj finančné toky. Užívateľ má možnosť vygenerovať si informácie o jeho portfóliu do súboru “PortfolioResults.xml“ a tak vytvoriť nosič pre ďalšie použitie

#### **5.4.5 Prehľad**

Na bočných paneloch sa nachádzajú rôzne prehľady finančných informácií. Panel “PX/GPX“ zobrazuje aktuálne informácie o vývoji indexu pražskej burzy<sup>7</sup> (PX a GPX). Ďalší takýto panel “Bilancia Portfólia“ poskytuje bilanciu ziskov a strát portfólia daného užívateľa, čo umožňuje stály prehľad nad vývojom portfólia. Tretím v poradí je panel zloženia portfólia “Prehľad Portfólia“, ktorý informuje o aktuálnom zložení daného portfólia z jednotlivých inštrumentov a ich váhový podiel na portfóliu.

#### **5.4.6 Ostatné**

Pre registrovaných užívateľov je k dispozícii jednoduché fórum, ktoré umožňuje pridávanie nových diskusných vlákien a príspevkov, a to v sekcii “Fórum“ z hlavného menu. K dispozícii sú aj odkazy na iné informačné zdroje na danú tému v sekcii “Odkazy“.

#### **5.4.7 Navigácia po webovom sídle**

Navigáciu po webovom sídle poskytuje hlavný panel menu a navigačná lišta, ktorá informuje o mieste aplikácie v ktorom sa užívateľ nachádza. Podrobný návod na použitie a vysvetlivky jednotlivých položiek sú obsiahnuté v sekcii “Pomocník“. Informácie o autorovi, verzii a účele aplikácie sú dostupné v sekcii “O aplikácii“.

---

<sup>7</sup> Px resp. GPx je index pražskej burzy PSE, ktorý reflektuje zmenu cien cenných papierov obchodovaných na burze cenných papierov PSE

## 5.4.8 Témy

Užívateľ má možnosť výberu z dvoch rôznych tém, ktoré dávajú možnosť variability užívateľského rozhrania. Táto funkcia webovej aplikácie neprináša žiadny informačný tok, no napriek tomu je to nástroj na prilákanie nových užívateľov k tejto webovej aplikácii.

## 5.5 Použité nástroje

Technológia ASP.NET je veľmi rozsiahla a poskytuje množstvo rôznych riešení implementácie. Rozšírenie tejto technológie prináša množstvo hotových komponentov ako napríklad textový editor alebo grafické nástroje. Takýmto spôsobom sa stáva technológia ASP.NET jednoduchým a efektívnym implementačným a vývojovým nástrojom , pričom efektívnosť vývoja aplikácií má vysokú úroveň

### 5.5.1 AJAX Toolkit

Nástroj AJAX Toolkit je súbor nástrojov s použitím skriptovacieho jazyka AJAX na grafické efekty prezentácie webovej aplikácie. V tejto webovej aplikácii je nástroj použitý na zobrazovanie panela kalendára pri vyplňaní dátumu, ktorý tak dáva jednoduchý prostriedok na špecifikovanie dátumu. Tento nástroj je dostupný na webových stránkach [www.asp.net](http://www.asp.net) .

### 5.5.2 Chart.NET

Nástroj Chart.NET je novinkou z dielne spoločnosti Microsoft. Obsahuje jednoduché a účelné prostriedky pre tvorbu akýchkoľvek typov grafov. K dispozícii sú jednotlivé objekty rôznych typov grafov, ktoré dokážu premeniť surové dáta na zaujímavú informáciu. Tento nástroj bude súčasťou nasledujúcej verzie NET.Frameworku. V súčasnosti je dostupný na webových stránkach [www.microsoft.com](http://www.microsoft.com) ako rozšírenie NET.Rameworku 3.5.

### 5.5.3 FCKeditor

Ide o jednoduchý textový editor pre použitie v diskusných fórach, ako nástroj na písanie a formátovanie textových polí resp. príspevkov vo fóre. Tento nástroj je taktiež voľne dostupný na webových stránkach [www.asp.net](http://www.asp.net) .

## 6 Verzia aplikácie a rozšírenia

Verzovanie je nástroj na označenie rozšírenia aplikácií. V podstate ide o označenie aplikácie príslušným číslom, ktoré definuje jednotlivý stupeň v rozširovaní aplikácie. Napríklad počas implementácie je možné testovať tzv. spustiteľné verzie resp. beta verzie:

### **Investičné portfólio v0.6 BETA**

Pre každú verziu je potrebné definovať zmeny v aplikácii a rozšírené funkcie aplikácie. Prvá dokončená a použiteľná aplikácia sa označuje väčšinou ako číslom jedna. Táto aplikácia teda nesie označenie:

### **Investičné Portfólio v1.0**

Táto verzia pracuje výhradne s informáciami z pražskej burzy (Prague Stock Exchange), avšak dizajn aplikácie je vytvorený tak aby umožňoval rozšírenie o nové nástroje analýzy cenných papierov alebo o nové burzy a ich finančné nástroje

## 7 Vývojový softvér

Táto webová aplikácia bola implementovaná využitím technológie ASP.NET, ktorá je súčasťou vývojového rámca NET.Framework v3.5 od spoločnosti Microsoft. Tieto nástroje sú voľne dostupné na webových stránkach [www.microsoft.com](http://www.microsoft.com) ako SDK(Software Development Kit)<sup>8</sup>. Ako databázový systém bol pri vývoji aplikácie využívaný SQL Server 2005 Express, ktorý je taktiež voľne dostupný na webových stránkach spoločnosti Microsoft. Webová aplikácia bola implementovaná použitím vývojového nástroja Visual Studio 2008 od spoločnosti Microsoft, ktorého používanie vyžaduje licenciu. Pre vzdelávacie účely je tento nástroj dostupný s licenciou študenta FIT VUTBR na webových stránkach [http://msdn61.e-academy.com/elms/Storefront/Home.aspx?campus=vut\\_fit](http://msdn61.e-academy.com/elms/Storefront/Home.aspx?campus=vut_fit) .

---

<sup>8</sup> SDK ( Software Development Kit ) je súbor nástrojov na kompiláciu a zostavenie aplikácií .NET

# Záver

Webová aplikácia je trendom vo vývoji aplikácií a informačných systémov. Svojou technológiou umožňujú užívateľom mať rýchly a neobmedzený prístup k aplikácii bez veľkých nárokov na zariadenie na ktorom je aplikácia spustená. Investovanie je spôsob zhodnotenia kapitálu. Investovanie so sebou prináša nároky na informácie a skúsenosti v tejto oblasti. Práve webové aplikácie sú nástrojom pre šírenie interaktívnych informácií, a tak tvoria podstatnú časť rozvoja práve v tejto oblasti. Na základe popisu požiadaviek je možné implementovať návrh aplikácie, jej dizajn a užívateľské rozhranie. Cieľom tejto bakalárskej práce bolo vytvorenie funkčného prototypu webovej aplikácie, ktorej úlohou je simulácia investovania na kapitálovom trhu. Aplikácia tak prináša užívateľovi možnosť bližšie sa zoznámiť s jednotlivými finančnými nástrojmi, ktoré sú dostupné a vyskúšať si schopnosti investovania na kapitálovom trhu. Najväčším prínosom tejto bakalárskej práce je zoznámenie sa s technológiou NET.Framework resp. ASP.NET a vývojovým nástrojom VISUAL STUDIO, ktoré tak podľa môjho názoru tvoria veľmi jednoduchý a veľmi silný nástroj na implementáciu akokoľvek náročných či už webových alebo tzv. formulárových aplikácií. Toto všetko je umocnené využitím programovacieho jazyka C# a skriptovacích jazykov ako napríklad AJAX, JavaScript. Tieto nástroje a mnohé iné tak dávajú nekonečné možnosti či už v dizajne samotnej aplikácie alebo grafike užívateľského rozhrania

# Literatúra

- [1] KŘENA, B. – KOČÍ, R.: Opora k predmetu Úvod do softwarového inženýrství, FIT VUT Brno, 2006.
- [2] ZENDULKA, J. – RUDOLFOVÁ, I.: opora k predmetu Databázové systémy, FIT VUT Brno, 2006.
- [3] HRUŠKA, T.: Opora k predmetu Informačné systémy, FIT VUT Brno 2007.
- [4] KAČMÁŘ, D.: Programujeme .NET Aplikace, Computer Press, 2001.
- [5] PROSISE, J.: Webové aplikace v .NET Framework, C# a ASP.NET, Computer Press, 2003
- [6] Wikipédia – otvorená encyklopédia. Dokument dostupný na URL: <http://cs.wikipedia.org/>,

# Zoznam príloh

Príloha 1. CD