



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**WEBOVÉ ROZHRAŇNIE PRE SPRÁVU VIRTUÁLNEHO
PORTFÓLIA**

WEB INTERFACE FOR VIRTUAL PORTFOLIO MANAGEMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP BALI

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ LENGÁL, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Bali Filip**
Program: Informační technologie
Název: **Webové rozhraní pro správu virtuálního portfolia**
A Web Interface for the Management of Virtual Portfolio
Kategorie: Informační systémy

Zadání:

1. Seznamte se s programovacím jazykem Python a frameworkem pro tvorbu webových aplikací Django.
2. Seznamte se se základy fungování burzovních trhů a základními metodami predikce ceny.
3. Seznamte se s nástroji pro vizualizaci pohybu ceny na burze, například finance.yahoo.com, finance.google.com, vizualizací v platformě Degiro, případně dalšími.
4. Analyzujte a vyberte vhodné (alespoň dva, je-li to možné) zdroje burzovních informací (např. IEX, Yahoo Finance, atd.).
5. Navrhněte snadno rozšiřitelnou webovou aplikaci pro správu virtuálního portfolia, kde si uživatel může zadat virtuální portfolio s akciemi a finančními deriváty. Umožněte různé způsoby zobrazování ceny akcií či derivátů a hodnoty portfolia v čase. Aplikace bude využívat reálná data z vybraného zdroje burzovních informací.
6. Navrženou aplikaci implementujte, spolu s několika základními technikami predikce ceny a možností nastavení upozornění na význačné události (např. změna ceny pod/nad nastavený limit, prudký růst ceny). Aplikace půjde spustit jak na serveru, tak i u klienta (přes Electron JS či podobnou technologii).
7. Aplikaci pravidelně testujte na vzorku uživatelů a modifikujte dle jejich zpětné vazby.

Literatura:

- Dokumentace projektu Django, <https://docs.djangoproject.com/en/3.1/> .
- Michael Becket. *How the Stock Market Works: A Beginner's Guide to Investment*. Kogan Page. 2012.
- Dokumentace projektu Electron JS, <https://www.electronjs.org/docs> .

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Lengál Ondřej, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

Abstrakt

Táto bakalárska práca navrhuje a implementuje webovú aplikáciu pre správu virtuálnych portfólií. Hlavným cieľom je vizualizácia a analýza dát z API burzových služieb. Užívateľ môže byť notifikovaný v prípade zmeny ceny. Aplikácia využíva existujúce metódy predikcie burzových cien a podporuje vizualizáciu užívateľských rozhodnutí na burze a poskytuje mu o nich všeobecný prehľad.

Abstract

This thesis designs and implements a web application for managing virtual portfolios. Main goal of the application is visualise and analyze data from stock exchange services API. User can be notified on price change. The application also uses existing methods to predict stock prices and supports the visualization of the user's stock exchange decisions and provides him/her a general overview of them.

Klíčová slova

webová aplikácia, Python, Django, finančný trh, investičné inštrumenty, burza cenných papierov, virtuálne portfólio, predikcia ceny

Keywords

web application, Python, Django, financial market, investment instruments, stock exchange, virtual portfolio, price prediction

Citace

BALI, Filip. *Webové rozhranie pre správu virtuálneho portfólia*. Brno, 2021. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. ONDŘEJ LENGÁL, Ph.D.

Webové rozhranie pre správu virtuálneho portfólia

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Ondřeje Lengála Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Filip Bali
11. května 2021

Poděkování

Velmi rád by som sa poďakoval vedúcemu bakalárskej práce, pánovi Ing. Ondřeji Lengálovi Ph.D. za všetok čas venovaný konzultáciám a cenné rady, ktoré mi dal počas spracovávania tejto bakalárskej práce. Tiež by som sa rád poďakoval svojej rodine a okoliu, ktoré má podporovalo a motivovalo k štúdiu.

Obsah

1	Úvod	3
2	Finančné pojmy a teória akciových trhov	5
2.1	Investícia	5
2.2	Investor	6
2.3	Investičný inštrument	6
2.4	Finančný trh	7
2.5	Akcia	7
2.6	Index	8
2.7	Investičná stratégia	8
2.8	Typy grafov	9
2.8.1	Líniový graf	9
2.8.2	Sviečkový graf	10
3	Analýza dostupných existujúcich riešení	11
3.1	Všeobecná analýza aplikácií pre správu virtuálneho portfólia	11
3.2	Yahoo! Finance	12
3.3	Google Finance	14
4	Návrh webovej aplikácie	15
4.1	Návrh architektúry webovej aplikácie	15
4.2	Entitno-relačný diagram a diagram prípadov použitia	17
4.3	Návrh modulov technológie Django	19
4.3.1	Modul Autentifikácia	19
4.3.2	Modul Portfólio	19
4.3.3	Modul Historických dát	21
4.4	Návrh modulov logiky	21
4.4.1	IEX Finance Manager	22
4.4.2	Yahoo! Finance Manager	23
4.4.3	Predikcia	23
4.4.4	Iné funkcie programu	23
4.4.5	Návrh systému upozornení	23
4.5	Návrh užívateľského prostredia	24
4.5.1	Navigačná lišta	24
4.5.2	Domovská stránka	25
4.5.3	Stránka listu portfólií	26
4.5.4	Stránka portfólia	27
4.5.5	Stránka investičného inštrumentu	28

4.5.6	Stránka historických dát	33
4.5.7	Stránka analýzy inštrumentu	33
5	Použitie technológie	34
5.1	Programovací jazyk Python	34
5.2	Django	35
5.3	Pandas	37
5.4	Bootstrap 4	37
5.5	Datatables	38
5.6	Datetimepicker	38
5.7	Nativefier	38
5.8	Highcharts	38
5.9	SQL	40
5.9.1	PostgreSQL	40
5.9.2	SQLite	40
6	Implementácia webovej aplikácie	42
6.1	Štruktúra programu	42
6.2	Implementácia užívateľského prostredia	44
6.2.1	Stránka prihlásenia/odhlásenia/registrácie	44
6.2.2	Stránka profilu	45
6.2.3	Domovská stránka	45
6.2.4	Stránka listu portfólií a stránka portfólia	46
6.2.5	Stránka investičného inštrumentu	46
6.2.6	Stránka analýzy inštrumentu	47
6.2.7	Stránka upozornení	48
6.3	Implementácia získavania a spracovania dát	48
6.3.1	Knižnica iexfinance	49
6.3.2	Knižnica yfinance	49
6.3.3	Knižnica stocknews	49
6.3.4	Stránka historických dát	50
6.4	Implementácia upozornení	50
6.5	Implementácia predikcie	50
6.5.1	Jednoduchá lineárna regresia	51
6.5.2	Model Auto ARIMA	51
6.5.3	Porovnanie výstupov	51
6.6	Databázová štruktúra aplikácie	52
7	Testovanie webovej aplikácie	53
7.1	Typy testovaní	53
7.2	Proces testovania	53
8	Záver	55
	Literatura	57
A	Obsah priloženého pamäťového média	59

Kapitola 1

Úvod

V súčasnej dobe v mnohých krajinách existuje právo na súkromné vlastníctvo a podnikanie. Jednotlivci zakladajú a rozvíjajú spoločnosti, ktoré vytvárajú produkty a zároveň sú súčasťou väčšieho celku, ktorý nazývame trh. Trh považujeme za miesto, kde dve strany po vzájomnej dohode dobrovoľne obchodujú. Samotný obchod môže mať viac podôb, príkladom môže byť sprostredkovanie služby za adekvátny finančný zisk.

Úspešnosť spoločnosti na trhu je ovplyvnená viacerými faktormi z ktorých spomeniem len niekoľko. Primárnym cieľom spoločnosti je vytvárať zisk a zároveň byť úspešnou na trhu. Zisk môže byť definovaný rôzne, existuje viacero typov zisku, avšak vo všeobecnosti je ho možné popísať ako kladný rozdiel medzi príjmami a výdavkami spoločnosti za určitý časový interval. Pre spoločnosť sú to dôležité finančné prostriedky s ktorými môže v budúcnosti manipulovať a sú základným predpokladom pri jej ďalšom rozvoji, ktorý jej môže priniesť konkurenčnú výhodu na trhu v segmente, kde ponúka svoje služby/tovar alebo môžu byť investované do nového produktu, ktorý sa môže týkať iného segmentu na trhu.

Všetky vyššie spomenuté faktory majú nezanedbateľný vplyv na aktuálnu a budúcu hodnotu spoločnosti. Existuje niekoľko metód pre získanie hodnoty podniku, všeobecným cieľom je zistiť ekonomickú hodnotu spoločnosti. Táto veličina môže ďalej slúžiť na celkový odhad úspešnosti na trhu v konkrétnych segmentoch. Ak je hodnota spoločnosti v konkrétnom segmente vysoká a zároveň je vyššia než u konkurencie, môže z toho vyplývať, že produktom/službám, ktoré spoločnosť v segmente ponúka sa darí a majú významnejší podiel na trhu. Dôvera v hodnotu spoločnosti sú nerozlučne spojené s výškou ceny akcie spoločnosti na burze. Aby si investor mohol kúpiť akcie spoločnosti, musí byť spoločnosť verejne obchodovateľná.

Akcie spoločnosti patria do podmnožiny investičných nástrojov. Vlastníci akcií môžu hlasovať na valných zhromaždeniach spoločnosti, kde môžu silou svojho hlasu (často je závislý od percentuálneho počtu vlastnených akcií spoločnosti) ovplyvňovať jej ďalšie smerovanie. Zároveň je pre investora vlastníctvo akcií spôsob ako investovať a prípadne zhodnotiť svoje doterajšie finančné prostriedky.

Status obchodovateľnej spoločnosti prináša pre spoločnosť radu výhod a nevýhod. Ako už bolo popísané vyššie, parametre pre ďalšiu inováciu alebo uvedenie nových produktov na trh vyžaduje najmä dostatočné finančné prostriedky. V závislosti od spoločnosti, druhu a rozmeru plánovaných zmien môžu byť požiadavky na objem potrebných financií diametrálne odlišné. V prípade, že spoločnosť v danom momente nevládni dostatočné množstvo finančných prostriedkov na uskutočnenie plánov, môže jej výrazným spôsobom pomôcť stať sa obchodovateľnou. V princípe tak predá investorom určitý podiel vlastníctva v spoločnosti, čím získa ďalšie financie. Býva zvykom, že prípadným investorom je ponúknutý podiel

dokopy menší než pětidesiat percent. Týmto spôsobom bude pôvodná spoločnosť stále majoritným vlastníkom akcií a v prípade akýchkoľvek zmien v spoločnosti bude mať hlavné slovo.

Ako už vyplýva z textu vyššie, pozícia investora hrá dôležitú úlohu vo finančnom sektore. Investuje svoje peniaze do finančných inštrumentov, ku ktorým patria aj akcie obchodovateľnej firmy, pričom jeho hlavným motívom je svoje financie zhodnotiť. Zhodnotenie financií môže prebiehať v rôznych časových intervaloch, od krátkodobých až po dlhodobé investície. Na to aby investor svoje financie investoval správne, musí dobre zvážiť riziká investícia a zároveň mať čo najväčší možný prehľad o ďalšom vývoji v budúcnosti. Pod vývojom v budúcnosti sa v tomto kontexte myslí, aby investor mohol v budúcnosti sledovať svoje investície a vedieť tak reagovať na zmeny čo najrýchlejšie. Nakoľko na cenu akcie spoločnosti má vplyv veľmi veľa faktorov, ktoré sú prítomné teraz alebo ešte len môžu v budúcnosti neočakávane vzniknúť, stále bude investovanie do akcií rizikom, hoci toto riziko je možné dôkladnou analýzou finančného trhu a udalosti vo svete patrične minimalizovať. Preto je v záujme investora mať k dispozícii čo najlepšie nástroje pre analýzu, prehľad dát a situácie vo finančnom sektore.

Cielom tejto bakalárskej práce je vytvoriť webovú aplikáciu pre správu virtuálneho portfólia. Úlohou aplikácie je umožniť investorovi získať informácie o aktuálnych a historických vývoch cien investičných inštrumentov. Zároveň má ponúkať prehľad a analýzu už investovaných financií na niekoľkých úrovniach a z rôznych uhlov pohľadov. Ako už z popisu vyplýva, webová aplikácia bude určená užívateľom, ktorí sa buď rekreačne zaujímajú o finančné trhy alebo sú už reálnymi investormi. Cieľom je aby údaje a dáta o finančných inštrumentoch boli z prostredia reálneho sveta a tým pádom aby prínosom aplikácie bolo jej použitie aj v reálnom prostredí. S využitím jednoduchých predikčných techník na báze technickej analýzy bude môcť užívateľ pozorovať predpokladaný ďalší vývoj svojho portfólia a jednotlivých inštrumentov. Rovnako je cieľom aby bola aplikácia transparentná a aby užívateľ mohol nahliadnuť do zdrojového kódu aplikácie. Pri vývoji aplikácie sú brané do úvahy princípy voľne šíriteľného zdrojového kódu a modularita s čím je spojená jednoduchšia rozšíriteľnosť aplikácie o ďalšie funkcionality v prípadného priebehu budúceho vývoja.

Osobná motivácia pre výber takejto témy pre bakalársku prácu spočíva vo využití znalostí a prepojení informačných technológií s ďalším oborom, konkrétne finančným. Návrh a vývoj aplikácie zahrňuje nie len vedomosti informačných technológií ale je nutné pochopenie základných princípov obchodovania aby bola do aplikácie implementovaná správna a dôležitá funkcionality, ktorá má pre užívateľa ďalší prínos. Rovnako je mojou ambíciou sa zoznámiť s technológiami webových aplikácií a ich princípmi, nakoľko sa jedná o rýchlo rozvíjajúcu sa oblasť, ktorá prináša stále nové a nové možnosti a nezanedbateľne vplýva a ovplyvňuje pohľad na moderne navrhnuté aplikácie v prostredí stále sa rozvíjajúceho internetu.

Kapitola 2

Finančné pojmy a teória akciových trhov

Táto kapitola slúži ako upresnenie a pomenovanie základných pojmov a princípov, ktoré sú často spomínané v texte bakalárskej práce a ich princípy využité v jej implementácií.

Pochopenie finančných pojmov a vedomosti z oblasti finančných trhov a obchodovania s finančnými inštrumentami je nevyhnutnou súčasťou pre správne používanie aplikácií, ktoré sa na túto oblasť špecializujú. Tým pádom aplikácia nie je vhodná pre užívateľov, ktorý sa vo finančnej oblasti vôbec neorientujú, nakoľko súčasťou aplikácie sú odborné termíny a nesprávne interpretovanie výsledkov z aplikácie môže viesť aj k nemalým osobným finančným stratám.

Rovnako dôležité je objasniť, že správanie finančného trhu, špeciálne toho akciového je chaotické. Nikdy nie je možné s istotou predpovedať vývoj cien investičných inštrumentov a rovnako nie je možné s úplnou istotou ani určiť všetky vplyvy a ich váhu, ktoré v danom momente cenu menia. Z toho vyplýva, že faktory a vývoj ceny je možné iba odhadnúť. Avšak častokrát majú podobné odhady veľmi triviálny rozmer a ich výpovedná hodnota je nejasná. Nejasnosť narastá v prípade, ak započítame možnosť javov, ktoré sa stávajú náhodne a ich trend je nejasný alebo nie je žiadny (napríklad prírodné katastrofy, ľudské nešťastia, náhle zmeny správania na trhu - finančná kríza).

2.1 Investícia

Investícia je vloženie obnosu finančných prostriedkov do investičného inštrumentu za účelom zhodnotenia v budúcnosti. Investíciu vykonáva investor so svojimi financiami. Investície môžu mať viacero vlastností, uvediem tie najzákladnejšie. [11]

Riziko investície určuje aký je vzťah medzi investíciou a výsledkom investície. Typov rizík môže byť rádovo niekoľko a každý má svoj význam pri konkrétnych typoch investícií. Veľkosť rizika určuje aká je šanca, že investícia bude neúspešná, čo sa môže prinajlepšom rovnať nulovému zisku a prinajhoršom k finančnej strate pre investora. Každý investor by sa mal pred vykonaním investície dôsledne oboznámiť s možnými rizikami investície¹. [3]

Časový horizont investície môže byť krátkodobý, strednodobý a dlhodobý. Krátkodobý sa berie v rámci časového horizontu, ktorý je nanajvýš dlhý jeden kalendárny rok. Krátkodobý horizont je vhodný na investovanie do skupiny investičných nástrojov ako napríklad termínované vklady a peňažné fondy. Strednodobý má časový interval od jedného do troch

¹<https://www.lynxbroker.sk/investicne-rizika/>

kalendárnych rokov a sú vhodné ako napríklad investície do krátkodobých a dlhodobých dlhopisov a môžu tu byť prítomné aj akcie. Dlhodobé sú v horizonte päť a viac rokov a sem sa zaraďujú investície hlavne do akcií či akciové fondy. Konkrétne investície rozdelené v rámci časových horizontov nie sú striktné dané a slúžia skôr ako všeobecný prehľad z hľadiska najväčšieho podielu investície daného typu v danom časovom horizonte. [9][24]

Vzťah investície k trhu sa rozumie rozdelenie investícií medzi aktívne a pasívne. Často je tento termín skôr skloňovaný na vyššej úrovni a to priamo na aktívneho a pasívneho investora. Tento termín je upresnený v ďalších sekciách textu (Investor v časti 2.2). [17]

Investícia ako súčasť investičnej stratégie vytvára nový pohľad na investíciu ako takú, keďže samotná jedná investícia môže byť len časť súboru väčšej investície do určitého trhového segmentu. Veľkosť investície má priamy dopad na hodnotu portfólia, do ktorej je investícia zahrnutá. [18]

2.2 Investor

Pod investorom rozumieme fyzickú osobu alebo organizáciu, ktorá investovala svoje financie do ľubovlného investičného inštrumentu. Charakteristika investora je veľmi všeobecná, keďže spôsob ako sa investor správa na trhu sa vzťahuje od osobných rozhodnutí konkrétnej fyzickej osoby/organizácie. Spoločným zámerom investorov je zhodnotiť svoje finančné prostriedky pomocou investície. Je v zodpovednosti investora aby zväzil čo najviac rizík pred investíciami a eliminoval tak prípadné finančné straty. Mnoho investorov si postupom času vybuduje svoje vlastné portfólio investícií a sledovaných investičných nástrojov a postupuje podľa jeho zvolenej investičnej stratégie. Investora je možné charakterizovať podľa vybraných parametrov, pre prehľad spomeniem tie najjednoduchšie. [5]

Aktívny investor je podmnožina investora, ktorého cieľom pri investícií je prekonať priemerné výnosy trhoch s akciami. Využíva k tomu krátkodobé cenové výkyvy a časté manipulácie s svojimi investíciami. Rovnako to môže zahŕňať väčšiu rizikovosť a menšiu stabilitu investícií, ktorá je naopak vyvážená možnými vyššími ziskami oproti priemerným ziskom na akciových trhoch. V princípe, ideálny úspešný aktívny investor vie v správnu chvíľu kedy podiel predať alebo navýšiť ďalším nákupom. Vyžaduje sa pri tom časté dôkladné analýzy trhu a početné osobné skúsenosti pri investovaní. Rovnako dôležité je správny výber akcií v portfóliu a to taký, aby bolo portfólio čo najziskovejšie. [17]

Pasívny investor je podmnožina investora, ktorého cieľom je kopírovanie výnosov zvoleného indexu. Tento spôsob investície je menej náročný oproti tomu u aktívneho investora. Medzi jeho časté investície patria ETF fondy. Radí sa medzi dlhodobejšie investície a predpokladá sa zisk na úrovni priemerného akciového trhu. [17]

Pojem investor sa vzťahuje na fyzickú osobu alebo aj organizáciu, ktorá má v úmysle svoje voľné finančné prostriedky zhodnotiť. Investor zvažuje pri investícií napríklad mieru rizika, čím je riziko nižšie, tým je väčšia šanca návratnosti investície. Z hľadiska bakalárskej práce uvažujeme o užívateľovi aplikácie ako o investovi, ktorý disponuje svojimi finančnými prostriedkami a investuje ich v rámci svojho portfólia.

2.3 Investičný inštrument

Je to podmnožina finančných inštrumentov ktorý predstavuje súbor finančných prostriedkov, ktorých spoločnými vlastnosťami sú že majú určitú hodnotu a sú obchodovateľné.

Každé aktívum zakúpené investorom je možné považovať za investičný inštrument. Môžu mať podobu cenného papiera, komodity, derivátu či indexu. [14] [2]

2.4 Finančný trh

Existuje viacero typov finančných trhov. Medzi najzákladnejšie patria:

- Peňažné trhy - Možnosť požičať si na krátku dobu finančné prostriedky so splatnosťou v horizonte jedného roku.
- Kapitálové trhy - Miesto kde je možnosť obchodovať s dlhopismi, akciami, derivátmi.
- Devízové trhy - Miesto kde je obchodovaná domáca mena za zahraničné meny, prípadne zahraničné meny navzájom
- Komoditné trhy - Miesto kde sa obchoduje s prírodnými surovinami a poľnohospodárskymi výrobkami

Spoločnou vlastnosťou všetkých typov finančných trhov je že predstavujú miesto, môže byť fyzické alebo virtuálne, kde investori obchodujú s finančnými inštrumentami. [10]

2.5 Akcia

Akcia reprezentuje cenný papier spoločnosti, ktorý predstavuje dohodnutý podiel vlastníctva v spoločnosti. Osoba, ktorá akciu vlastní sa volá akcionár. Akcionár má v danej spoločnosti práva, ktoré mu umožňujú sa podieľať na rozhodovaní o jej budúcim smerovaní, ktoré môže ovplyvniť na valnom zhromaždení pomocou hlasovania. Väčšinou je jeho podiel na hlasovaní priamo úmerný s počtom akcií ktoré vlastní. O váhe hlasu, ktorú predstavuje jeden kus akcie rozhoduje spoločnosť, ktorá ich vydáva. Väčšinou ale má spoločnosť väčšinového vlastníka, ktorý má väčšinový podiel akcií v spoločnosti (viac ako päťdesiat percent) a tým pádom má sám plnú kontrolu nad spoločnosťou bez ohľadu na podiely iných akcionárov. [12]

Hlavným parametrom cennosti akcie je jej cena. Na cenu môže vplývať viacero faktorov ako napríklad vnútorné procesy v spoločnosti, historické úspechy spoločnosti, predpoveď ako sa bude spoločnosti dariť v budúcnosti, dôvere v segment v ktorom spoločnosť pôsobí a vyrába prípadne stav svetovej ekonomiky. Nakoľko dôvodov, ktoré majú menší či väčší dopad na ďalší vývoj ceny akcie je mnoho a zároveň môžu byť od seba nezávislé, spôsobuje že nie je možné zaručene predpovedať jej ďalší vývoj. Predikciou cien akcií sa zaoberajú dve disciplíny, fundamentálna analýza a technická analýza, ktoré sa od seba odlišujú spôsobom ako a s akými dátami pracujú a podľa akých kritérií vyhodnocujú výsledok.

Fundamentálna analýza vykonáva hĺbkovú analýzu spoločnosti. Využíva k tomu informácie, ktoré sú verejne dostupné širokej spoločnosti. Zaoberá sa viacerými faktormi, pre predstavu uvediem len niektoré: [16]

- Štatistické dáta, ktoré spoločnosť vyprodukovala počas svojej histórie pôsobenia.
- Štruktúra trhu na ktorom spoločnosť pôsobí. Aká veľká je na v danej oblasti konkurencia, aké je zložité a aké sú vysoké náklady aby spoločnosť mohla vstúpiť a začať v oblasti obchodovať, tým pádom predstavovať ďalšiu konkurenciu.

- V akom životnom cykle sa daná oblasť nachádza. Môže ísť o čerstvo sa rozvíjajúci trh, stabilnú časť cyklu alebo obdobie útlmu a zastarávania. Pomáha určiť dynamiku vývoja a prípadnú predpoveď ako bude daná oblasť dôležitá v budúcnosti a či bude potrebné. Taktiež je možný predpoklad ak by v období útlmu (kedy dynamika trhu v danej oblasti prestáva byť zaujímavá a postupne firmy danú oblasť opúšťajú) sa vyskytol dostatočne veľký pokrok alebo objav, ktorý posunie danú oblasť opäť do stavu dynamického rozvoja a kedy sa celý cyklus opakuje.

Technická analýza využíva k svojej predpovedi historické a súčasne dáta cien akcie. Táto analýza zakladá na tom, že všetky ostatné vplyvy na spoločnosť sú už výsledkom ceny a teda jej súčasťou. Zároveň očakáva, že cena akcie sa vyvíja v trendoch a že je trend možné rozpoznať podľa časového rámcu (krátkodobé, stredne dlhé a dlhodobé) a že história sa zvykne opakovať. Technická analýza sa zaujíma iba o cenu akcie, preto veľkú váhu prikladá ako sa cena vyvíja. Cenu môžeme rozdeliť a analyzovať na: [13]

- Otváracia cena - Cena, ktorú akcia zaznamená na začiatku otvorenia burzy. S akou cenou akcia spoločnosti vstupuje na burzu pri jej otvorení.
- Najvyššia cena - Najvyššia cena za ktorú bola akcia obchodovaná v intervale od otvorenia až po zatvorenie burzy.
- Najnižšia cena - Najnižšia cena za ktorú bola akcia obchodovaná v intervale od otvorenia až po zatvorenie burzy.
- Uzavieracia cena - Cena, ktorú mala akcia pri zatvorení burzy. Ide o hodnotu za akú bola akcia obchodovaná naposledy pred uzavretím burzy.

Cena akcie je v podstate hodnota za ktorú bola daná akcia v daný moment obchodovaná.

2.6 Index

Predstavuje portfólio zostavené z cenných papierov podľa trhového významu. Jednotlivé cenné papiere sú v rámci indexu vážené. Výkonnosť indexu naznačuje výkonnosť konkrétneho trhu na ktorý je index navrhnutý. Investori pohyby indexov analyzujú a môžu ovplyvniť ich rozhodovanie v danom segmente trhu. [4]

2.7 Investičná stratégia

Pod investičnou stratégiou rozumieme formuláciu jednotlivých investičných cieľov a ich následne dosiahnutie pomocou investičnej stratégie. Investor si v tomto prípade môže rozdeliť svoje investičné inštrumenty do priorit a podľa nich potom prebehne následná alokácia finančných prostriedkov investora. Investičných stratégií je mnoho a akýkoľvek postup investora, ktorý nekoná chaoticky ale podľa svojho alebo prevzatého vzorca, môžeme označiť za investičnú stratégiu.

2.8 Typy grafov

Pre vizualizáciu cien a trendu vývoja akcií sa používa niekoľko grafov. Každý graf má svoje špecifiká a vlastnosti, niektoré sú pre jednoduchú analýzu a nezatažujú užívateľa veľkým množstvom dát a iné naopak obsahujú značne väčšie množstvo informácií ako boli akcie obchodované a ponúkajú tak užívateľovi značne väčšie možnosti pri analýze. Predstavím často používané grafy v oblasti obchodovania na burze a ktoré bude taktiež používať aj aplikácia tejto bakalárskej práce.

2.8.1 Líniový graf

Patri k jednoduchším grafom, história dát je spojená jednoduchou čiarou. Ako dáta sú väčšinou použité uzatváracie ceny akcie na burze.



Obrázek 2.1: Líniový graf. Zdroj: yahoo.com²

²<https://finance.yahoo.com/chart/TSLA>

2.8.2 Sviečkový graf

Sviečkový graf ponúka k zobrazeniu podobné dáta ako schodový graf. Sviečkový graf patrí k jedným k najčastejšie používaných grafov vo finančníctve vďaka svojej dlhej histórii. Odčítanie dát zo sviečkového grafu je podobné k tomu schodovému. Konkrétny interval počas ktorého sa akcie obchodovali si môžeme predstaviť ako zvisle orientovaný obdĺžnik (telo sviečky) z ktorého hornej aj dolnej strany vychádzajú úsečky (horný a dolný knôt sviečky). Najvyšší bod hornej úsečky, ktorá predstavuje vrchol, predstavuje najvyššiu cenu akcie. Rovnako najnižší bod dolnej úsečky predstavuje najnižšiu cenu akcie. Horná a dolná strana sviečky predstavuje otváraciu alebo uzatváraciu cenu. O ktorú cenu ide vieme určiť podľa farby tela sviečky, ktoré ak je červenej farby potom cena klesala a ak je zelenej potom cena stúpala. Ak cena stúpala potom horná strana sviečky je uzavieracia cena a dolná otváracia cena, ak je červenej farby potom je význam ceny naopak.



Obrázek 2.2: Sviečkový graf. Zdroj: yahoo.com³

³<https://finance.yahoo.com/chart/TSLA>

Kapitola 3

Analýza dostupných existujúcich riešení

Táto kapitola je výsledkom môjho prieskumu existujúcich riešení. Analyzujem riešenia z pohľadu bežného užívateľa, ktorý nemá žiadnu skúsenosť s podobnými aplikáciami. Snažím sa zameriavať na užívateľské prostredie a funkcionálnosť. Snažím vnímať rozdiely medzi jednotlivými aplikáciami a porovnávať ich. Ako výstup analýzy uvediem len riešenia, ktoré sú zjavne rozdielne. Mnoho riešení ponúka veľmi podobnú funkcionálnosť len pod inou značkou.

3.1 Všeobecná analýza aplikácií pre správu virtuálneho portfólia

V súčasnosti existuje už mnoho webových aplikácií v oblasti spravovania osobného virtuálneho portfólia. Mnoho z nich má možnosti využitia predplatného, bez ktorého je služba značne limitovaná (napríklad sledovaním maximálneho počtu investičných nástrojov). Zároveň, pri mnohých službách je uzavretý zdrojový kód. Aplikácie nie sú v tomto prípade transparentné, na ich používanie sa môžu vzťahovať rôzne licenčné podmienky.

Jednotlivým mnou vybraným službám sa budem podrobnejšie venovať v sekciách tejto kapitoly. Rozdiely je možné vidieť v celkovej užitočnosti, aké operácie je možné v aplikáciách vykonávať. Pre hodnotenie služieb je možné použiť viacero metrík a rovnako záleží na preferenciách užívateľa.

Rovnako, veľké rozdiely medzi aplikáciami sú aj v oblasti grafického rozhrania. Rozdelil by som aplikácie medzi jednoduchšie a zložitejšie. Jednoduché rozhrania ponúkajú veľmi čistý vzhľad, bez rušivých elementov a ako nový užívateľ som sa vedel v rozhraní rýchlo zorientovať a začať ho používať. Pre príklad jednoduchšieho rozhrania uvediem Google Finance. Google Finance považujem za dobrý príklad na uvedenie oboch typov grafických rozhraní, pretože v dobe písania tejto práce má služba nové a od základu prerobené grafické rozhranie, ktoré bolo značne zjednodušené od pôvodného rozhrania.

Pôvodné rozhranie Google Finance bolo viac zložitejšie. Zložitejšie rozhranie je viac komplexné, často obsahuje viac prvkov na jednej stránke a častokrát nie sú od seba dostatočne vizuálne oddelené a pre nového užívateľa dokopy tvoria neprehľadnú sieť informácií. Výhodou avšak môže byť ich praktickosť. Užívateľ sa postupom času naučí aplikáciu ovládať a väčšie množstvo informácií na jednej stránke môže byť prínosom.

Okrem vizuálnej stránky je možné vidieť vplyv jednoduchosti/zložitosti aj na funkcionálnosť aplikácie. Google Finance pri prechode na jednoduchšie grafické rozhranie taktiež

okresal funkcionalitu. Stratila sa napríklad možnosť zmeny typu grafu z čiarového grafu na sviečkový. Ďalšie podrobnosti a nastavenia grafu boli pre čistotu dizajnu buď odstránené alebo skryté. Porovnanie grafu s iným grafom je síce stále dostupne avšak je potrebné viac interakcií užívateľa s aplikáciou. Zatiaľ čo pôvodne bolo možné určiť časový interval dát z ktorých bude graf vykreslený manuálne a tiež formou skratiek pre často používané časové obdobia (deň, 5 dni, 1 mesiac, 3 mesiace, 6 mesiacov, 1 rok, 5 rokov, 10 rokov, celé obdobie), po novom je možné využiť len skratky pre často používané obdobia, ktoré sú tiež ochudobnené o niektoré možnosti. Rovnako pri porovnávaní dvoch grafov súčasne sa možnosti zobrazenia časového úseku zmenšili (nie je dostupné porovnanie v daný deň, zatiaľ čo pri jednom grafe dostupné je).

Ďalším častým negatívom sú agresívne reklamy, ktoré odpútavajú pozornosť od dôležitých informácií aplikácie.

3.2 Yahoo! Finance

Je to služba, vďaka ktorej je možné spravovať svoje osobné virtuálne portfólio. Podporuje celú škálu investičných nástrojov (napríklad vrátane digitálnej meny, ktoré stále nebývajú samozrejmosťou). Do vlastného portfólia je možné si pridať investičné nástroje do zoznamu sledovaných položiek. Pre vybrané sledované položky sa následne zobrazia na stránke aktuality zo sveta. Pri detailnejšom zobrazení investičného nástroja poskytuje Yahoo! Finance širokú škálu informácií:

- Možnosť zobrazenia súhrnného prehľadu. Obsahuje tabuľku hodnôt, s akými bol daný investičný nástroj v ten deň obchodovaný (hodnota pri otvorení a uzavretí burzy, rozpätie hodnôt v ktorých bol obchodovaný v podobe minima a maxima). Prítomný je aj zmenšený graf, ktorého typ je možné zmeniť (obsahuje všetky často používané grafy pre finančnú oblasť) a na ktorom je možné zmeniť aj časový interval. Graf informuje o hodnote pri uzavretí burzy a tiež aký veľký bol rast alebo pokles hodnoty. Graf je možné zväčšiť a prejsť na ďalšie, podrobnejšie zobrazenie.
- Možnosť zobrazenia grafu, ktorý popisuje ako sa vyvíjala hodnota investičného nástroja za zvolený časový interval. Časový interval (X-ova os grafu) je možné zvoliť podľa predom nastavených rozpätí, skrz ktoré je možné zobraziť ako sa hodnota vyvíja minútu po minúte až po zobrazenie niekoľko ročného obdobia, ktoré ukáže aký je trend vývoja hodnoty. Je dovolené aj manuálne vybrať časového intervalu. Rovnako je možné meniť os Y, či už zväčšovať alebo znižovať interval alebo zmeniť aj typ (percentuálna, logaritmická, lineárna). Graf je možné porovnávať s inými grafmi iných investičných nástrojov. Pri zvolení je možné vybrať hrúbku čiary a jej farbu. Graf je interaktívny, zobrazuje ďalšie podrobnosti o hodnotách v konkrétny čas, nad ktorým ukáže kurzor počítačovej myši. Graf podporuje aj zobrazenie rôznych indikátorov. Posledná zaujímavá vlastnosť je podpora pre kreslenie priamok do grafu, ktoré slúžia k ďalšej analýze pre užívateľa.
- Panel všetkých historických hodnôt. Je v ňom možné vidieť aké hodnoty mal investičný nástroj v daných dňoch, pričom jednotlivé dni sú zobrazené pod sebou v tabuľke. Rozsah s akým sa budú na riadkoch ukazovať hodnoty sa dá zvoliť (každý deň, hodnoty na konci týždňa/mesiaca). Hodnoty popisujú hodnotu investičného nástroja pri otvorení/zatvorení burzy a minimum/maximum, ktorý v daný deň

dosiahol. K dispozícii je aj priemerná hodnota za daný deň a počet, ktorý popisuje aký veľký objem bol obchodovaný.

Yahoo! Finance podporuje zobrazenie investičných nástrojov podľa kategórií. Je možné napríklad zobrazit investičné nástroje, ktorých zameranie je výhradne energetika. Takýchto podkategórií je dostupná celá škála - zdravotníctvo, technológie, priemysel a podobne. Ďalšia možnosť rozdelenia do kategórií je podľa trhu, príkladom môžu byť často obchodované investičné nástroje alebo tie ktoré dosiahli významné zisky či straty na obchodovanej hodnote.

Dostupné sú aj filtre, ktoré pomáhajú investorovi zobrazit len tú množinu investičných nástrojov, ktorá spĺňa parametre nastaveného filtra. Možnosti filtra sú špecifické pre kategórie v ktorých chce investor hľadať.



Obrázek 3.1: Ukážka služby Yahoo! Finance. Zdroj: yahoo.com¹

¹<https://finance.yahoo.com/quote/TSLA/>

3.3 Google Finance

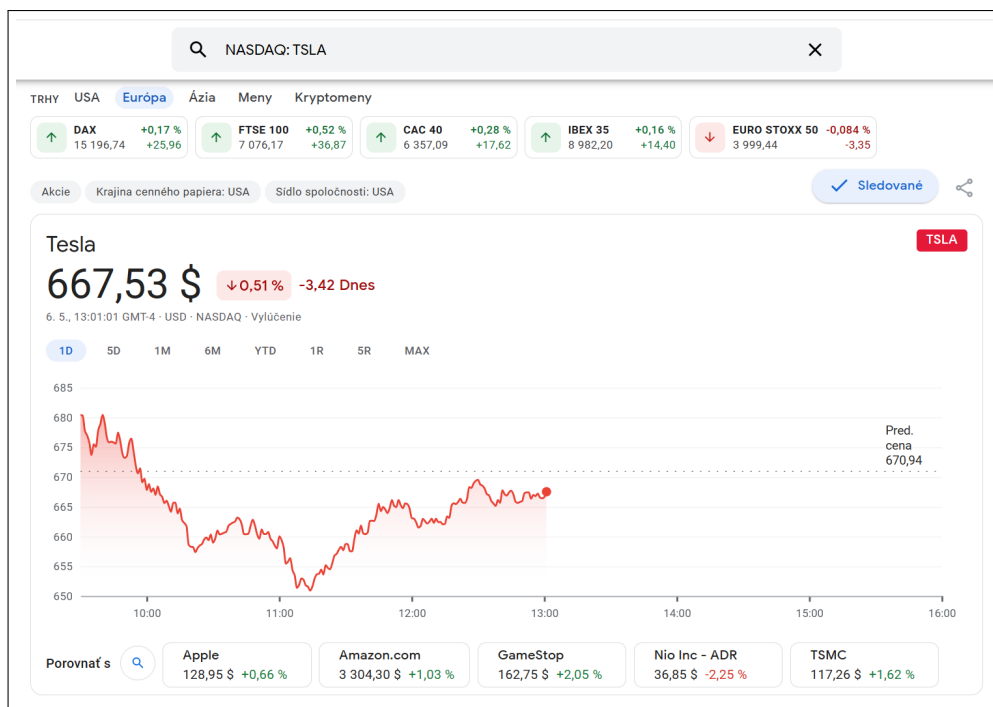
Je ďalšia služba, ktorá slúži investorom pre prehľad a manažovanie svojho virtuálneho portfólia. Oproti Yahoo! Finance pôsobí celá služba jednoduchšie pričom vzbudzuje dojem, že je určená skôr pre jednoduchý prehľad než hlbšiu analýzu a prehľad trhu.

Služba je koncipovaná modernejšie, neuvrhne na používateľa naraz veľa informácií, čo je možné považovať podľa preferencií za pozitívum/negatívum. Pri zobrazení konkrétneho investičného nástroja sa zobrazí jednoduchý čiarový graf, ktorý interaktívne zobrazuje hodnotu v danom časovom bode. Typ hodnoty sa mení podľa veľkosti časovej osi (môže ísť o hodnotu akcie v daný okamih v dni alebo o hodnotu akú mal investičný nástroj v momente uzavretia trhu).

Pre ďalšie zobrazenie informácií je nutné pohyb stránky smerom nadol, pričom sa zobrazujú jednoduché tabuľky so štatistikami pre daný investičný nástroj prípadne ponuka nových súvisiacich investičných nástrojov pre porovnanie.

Grafy je možné porovnať maximálne dva, nie je možné ďalších úprav preferencií grafov pri porovnávaní. Nie je možné zmeniť Y os grafu. Je možné meniť jedine časový interval za ktorý chceme investičné nástroje porovnávať avšak na výber je len veľmi limitovaný počet časových intervalov, nie je možná žiadna osobná preferencia.

Google Finance ponúka pri prehľadávaní zobrazenie správ, ktoré sú avšak skôr všeobecné než cielene zamerané na práve prehľadávané investičné nástroje ako je to možné vidieť pri Yahoo! Finance. Vie zobraziť detail spoločnosti, ktorej sa akcia týka a kvartálový súhrn štatistík o jej finančnom stave za dané obdobie.



Obrázek 3.2: Ukážka služby Google Finance. Zdroj: Výrez z webovej stránky Google Finance. Zdroj: google.com³

³URL: <https://www.google.com/finance/quote/TSLA:NASDAQ>

Kapitola 4

Návrh webovej aplikácie

Kapitola návrhu webovej aplikácie sa venuje dekompozícií požiadavkov webovej aplikácií na menšie problémy, popisuje prvotnú architektúru aplikácie a rozširuje špecifikáciu o ďalšie požiadavky na aplikáciu. Návrh webovej aplikácie prebehol po teoretickom zoznámení sa s technickými problémami aplikácie a ich následným vyhodnotením.

Cieľom kapitoly je:

- Dekompozícia aplikácie na menšie problémy, ktoré budú v návrhu zapuzdrené do modulov aplikácie. Návrh architektúry aplikácie.
- Navrhnutie hierarchie modulov a procesy ako medzi sebou moduly komunikovať. Popis jednotlivých modulov.
- Načrtnutie základnej vizualizácie a funkcionality grafického rozhrania aplikácie. Popis predpokladanej interakcie medzi užívateľom a aplikáciou.
- Navrhnutie logiky programu a získavanie, spracovanie a vizualizácia finančných dát.
- Návrh ako budú uložené dáta aplikácie a vytvorenie ER diagramu databázy.

4.1 Návrh architektúry webovej aplikácie

Architektúra webovej aplikácie je postavená na základoch konceptu Model-View-Template (MVT), ktorá je inšpirovaná architektoým vzorom Model-View-Controller (MVC). MVT deleguje procesy webovej aplikácie do viacerých komponentov:

- URLs - Spravuje všetky URL v rámci svojho komponentu. URL môžu byť zadané viacerými spôsobmi, napríklad reťazcom alebo regulárnym výrazom. V rámci celej aplikácie prebieha vyhodnotenie zadanej URL na dvoch úrovniach. Koreňový modul aplikácie obsahuje koreňový komponent URLs, ktorý predstavuje hlavnú úroveň a na ktorú sú naviazané ďalšie moduly na sekundárnej úrovni. Pri vyhľadávaní vzoru URL v rámci aplikácie je v prvom rade prehľadávaný primárny komponent. Primárny komponent môže priamo odkazovať na zobrazenie, avšak v rámci tejto aplikácie funguje primárny komponent ako radič, ktorý selektuje jednotlivé vzory URL do modulov na sekundárnej úrovni, podľa platnej cesty URL. Dôvodom takéhoto návrhu je jeho jednoduchá rozširiteľnosť o ďalšie vzory URL alebo úprava existujúcich. Takéto oddelenie na úrovni modulov zároveň znižuje závislosť. Prehľadávanie prebieha sekvenčne a pri

prvom vyhovujúcom vyhodnotený sa vykoná a ďalej nepokračuje. Pod vykonaním rozumíme, že príslušný vzor URL má k sebe pridelené zobrazenie, kde ďalej pokračuje tok programu.

- View - Funkcia pre zobrazenie stránky je unikátna v rámci svojho komponentu. Jej úlohou je spracovať HTTP požiadavok a odoslať HTTP odpoveď. Ak je to výhodné, potom môže byť volaná s parametrami. Pomocou parametrov je možné funkcie zobrazenia využívať dynamicky a definovať tak obsah zobrazenia stránky podľa vstupných parametrov. V aplikáciách sú využité dva dynamické spôsoby zobrazovania stránok, definovanie obsahu stránky pomocou parametrov funkcie zobrazenia alebo zmena obsahu stránky pomocou technológie AJAX. Technológia AJAX umožňuje dynamicky upravovať obsah stránky pomocou opätovného volania funkcie zobrazenia, avšak parametre zobrazenia sú uložené ako premenné POST požiadavku (a nie sú prístupné ako parametre funkcie). Pre splnenie požiadavku, funkcia zobrazenia často komunikuje s logikou aplikácie, aby získala potrebné dáta. Ak je požiadavok jednoduchého charakteru, časť logiky môže obsahovať samotné telo funkcie zobrazenia. Pre zložitejšie operácie sú volané funkcie vo vrstve logiky. K prístupu do databázy sa využíva ďalšia komponenta technológie Django, modely.
- Model - Modely sú objekty jazyka Python, ktoré definujú štruktúru aplikačných dát a sprostredkovávajú komunikáciu s databázou. Modely sú súčasťou technológie Django Object-Relational Mapper (ORM). Táto technológia patrí medzi hlavné prednosti technológie Django a vytvára medzivrstvu medzi aplikáciou a databázovým systémom, ktorá medzi nimi sprostredkováva komunikáciu. Umožňuje tak vývojárovi komunikovať s databázou len pomocou príkazov v jazyku Python. Medzi nepostrádateľne výhody patrí aj schopnosť komunikovať s viacerými typmi databáz pričom kód v jazyku Python nie je nutné upravovať. Nastavenie typu databázy prebieha v koreňovom module v súbore `settings.py` nastavením premennej `DATABASE`.
- Template - Šablóny sú textové súbory, ktoré slúžia na definovanie rozloženia objektov na webovej stránke. V tejto aplikácii sa v šablónach definuje rozloženie, dizajn a vzhľad s pomocou značkovacieho jazyka HTML s využitím knižnice pre kaskádové štýly Bootstrap 4. Technológia Django umožňuje šablóny generovať dynamicky s využitím šablónového jazyka Django, pomocou jednoduchých programových konštrukcií ako napríklad cyklus či podmienka. V kombinácii dát zo zobrazenia je možné vsadiť hodnoty premenných do šablóny a generovať tak dynamické webové stránky. Šablóny môžu byť rozšírené o vývojárom vlastné filtre, ktorými je možné dodatočne upravovať/formátovať hodnotu premennej pred tým než je vsadená do šablóny. Filtre prijímajú na vstup hodnotu z premennej a idea fungovania je podobná ako rúry v operačnom systéme UNIX.

Okrem hlavných MVT komponentov sa v Django moduloch môžu nachádzať aj ďalšie. V aplikáciách sú využité tiež:

- Form - Implementuje rozšírenú funkcionálnosť formulárov značkovacieho jazyka HTML. Pri definovaní je možné využiť automatické operácie nad databázovým systémom. Celková funkcionálnosť nie je triviálna. Pre potreby tejto práce sa predpokladá využitie základných formulárov na úrovni jednoduchých vstupných kolóniek, kde užívateľ bude môcť zadať hodnoty, ktoré potvrdí tlačidlom. Hodnoty budú spracované na úrovni funkcie zobrazenia, ktorá bude spracovávať príslušnú stránku.

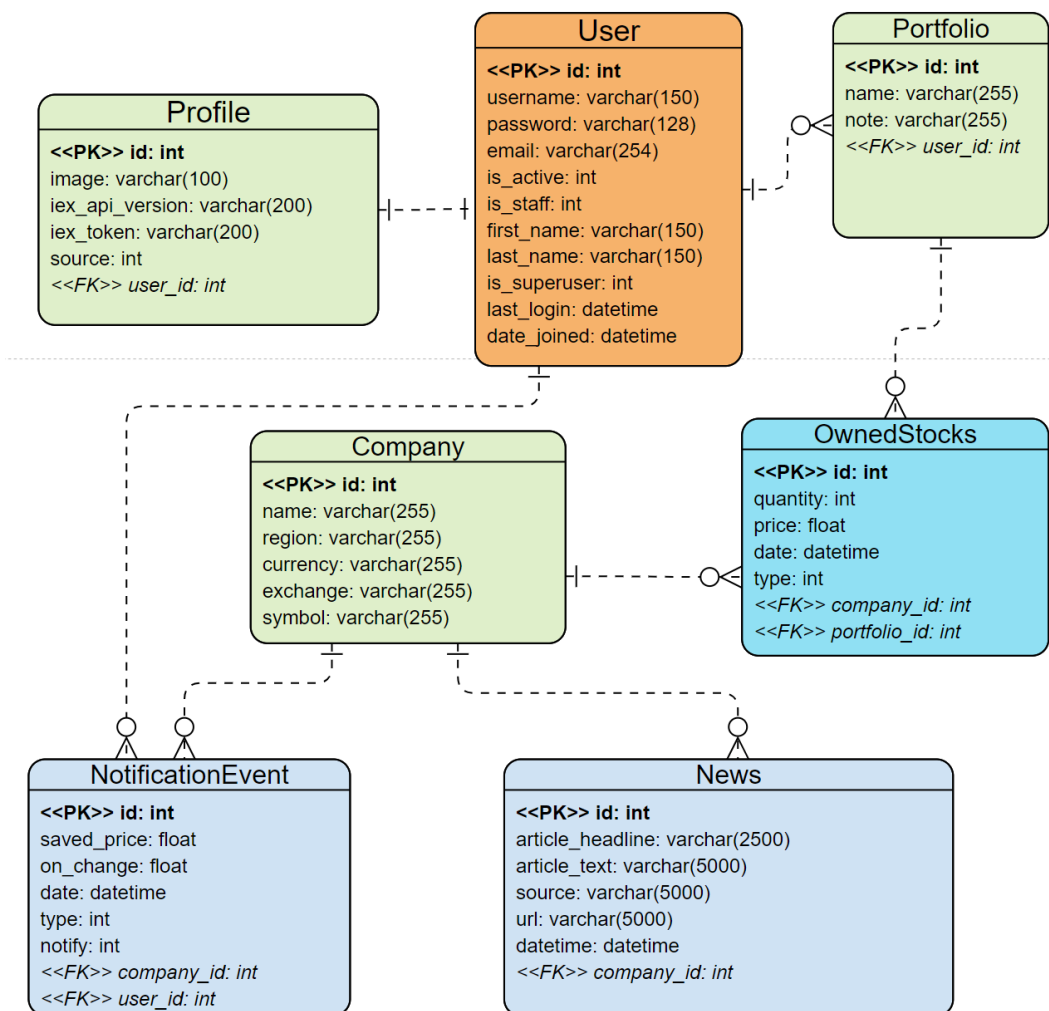
- Signal - Prepájajú objekt, ktorý správu vysiela s objektom/objektami, ktoré správe načúvajú. Komunikácia prebieha na báze notifikácií. Ak nastane udalosť v objekte, ktorá je spojená s vyslaním správy (notifikácie), potom ju dostanú objekty ktoré načúvajú na zadanom kanály. Objekty, ktoré správu dostanú, potom môžu ďalej reagovať.

Popri zmienených Django špecifických moduloch sa v návrhu aplikácie nachádza druhá, podobne dôležitá časť a tou je logika. Z pohľadu architektúry programu je logika zapuzdrená vo špecifickej vrstve, ktorá bude vytvárať rozhranie pre volanie funkcionalít logiky z modulov technológie Django. Parametre volania budú určovať aká funkcionalita bude vykonaná vo vnútri logiky a aká odpoveď bude vrátená.

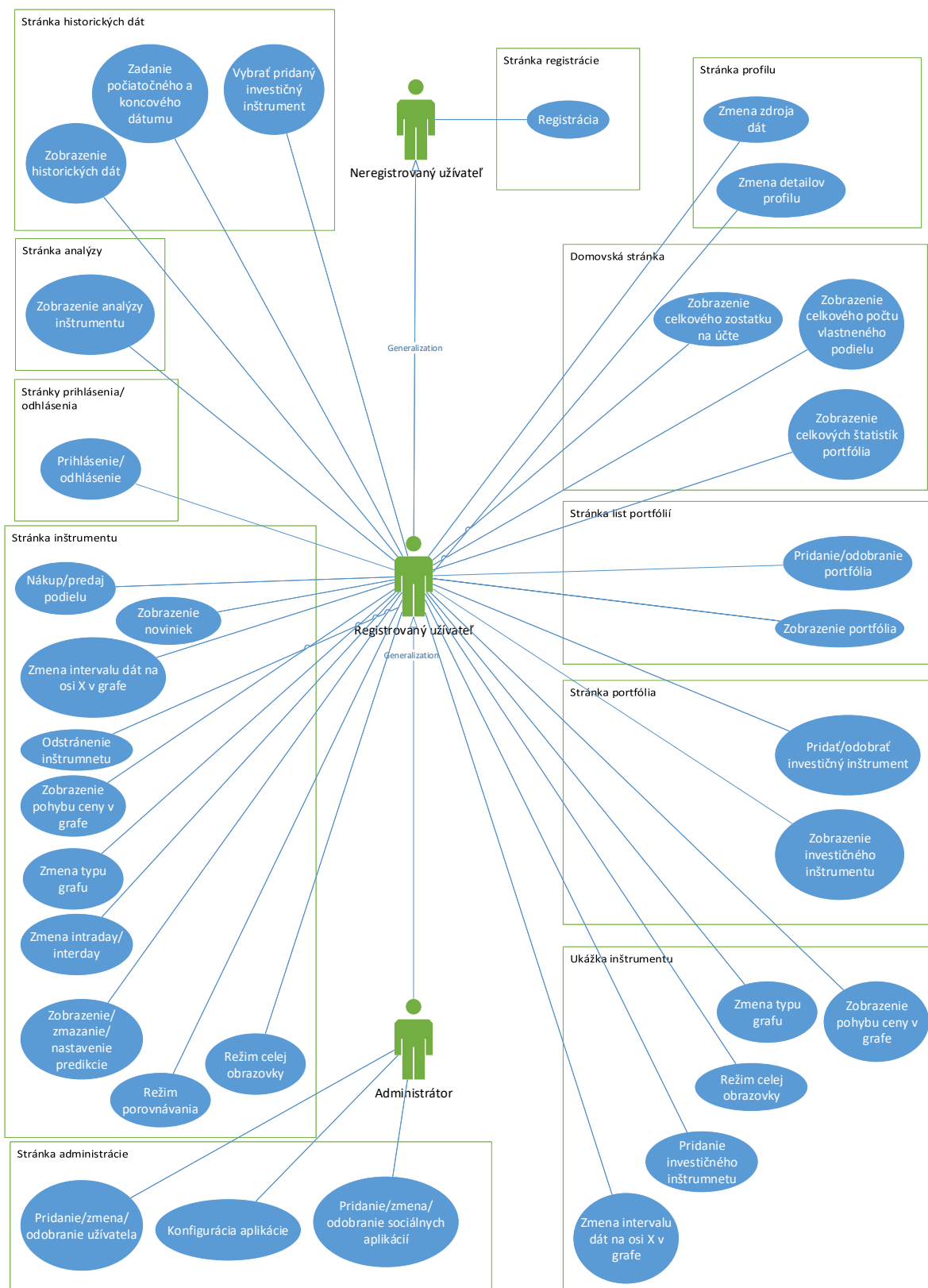
Konkrétnejší návrh modulov Django a logiky sa nachádza v ďalších sekciách kapitoly.

4.2 Entitno-relačný diagram a diagram prípadov použitia

Pred detailnejším popisom návrhu aplikácie, prikladám entitno-relačný diagram a diagram použitia, ktoré sa vzťahujú na návrh a budú v ďalších sekciách bližšie popísané.



Obrázek 4.1: Entitno-relačný model databázy.



Obrázek 4.2: Diagram prípadov použitia aplikácie.

4.3 Návrh modulov technológie Django

Modularita v návrhu je zameraná na dekompozíciu zložitejších problémov na menšie celky. Predpokladom je jasné ohraničenie funkcionalít a ich rozhrania. Ďalším cieľom je minimalizácia previazanosti a závislosti modulov medzi sebou. Tento cieľ sa avšak nie vždy darí splniť a to z dôvodu, že moduly tvoria v aplikácií širšie celky. Ak by bolo cieľom nulová previazanosť medzi všetkými modulmi, potom by to viedlo k zlúčeniu mnoho modulov do jedného, čo by drasticky zhoršilo prehľad v programovom kóde a zároveň komplikovalo úpravy, rozšíriteľnosť či prípadnú detekciu chýb pri testovaní a ich opravu.

4.3.1 Modul Autentifikácia

Tento modul je prvým s ktorým užívateľ príde do kontaktu. Má na starosti autentifikačnú časť aplikácie, pričom každá časť zahŕňa vlastnú HTML stránku a potrebnú logiku pre vykonávanie svojej funkcie. Jednotlivé časti možno stručne popísať ako:

- Proces registrácie - Spravuje možnosť registrácie užívateľa do aplikácie. Po úspešnej registrácii sa vykoná zápis údajov do databázy a presmerovanie užívateľa na prihlasovaciu stránku.
- Proces prihlásenia - Spravuje prihlásenie používateľa a vstup do systému. Prihlasovacia stránka sa skladá z formuláru, ktorý pre úspešné prihlásenie vyžaduje platné prihlasovacie meno a heslo. Ak je prihlásenie neúspešné, užívateľ je o tom informovaný, ak je úspešné, potom je presmerovaný na domovskú stránku aplikácie.
- Proces odhlásenia - Spravuje odhlásenie z aplikácie. Po úspešnom odhlásení je relácia užívateľa v aplikácii ukončená a obnovená až po opätovnom prihlásení.
- Profil užívateľa - Po úspešnej registrácii užívateľa je vytvorený a uložený záznam do databázy. Obsahuje prihlasovacie meno, heslo, email a ikonu profilu. Všetky údaje sú aj po registrácii zmeniteľné. Zároveň tiež obsahuje osobné nastavenia aplikácie kde je možný napríklad výber zdroja dát odkiaľ budú preberané informácie.

V skratke zhrnuté, autentifikačný modul sa stará aby užívateľ mohol pristúpiť do aplikácie pod svojim kontom a boli mu zobrazené len dáta, ktoré patria k jeho kontu. Zároveň tak chráni jeho súkromné dáta pred ostatnými užívateľmi.

4.3.2 Modul Portfólio

Obsahuje hlavnú funkcionalitu aplikácie. Spravuje zoznam užívateľom vytvorených portfólií a v nich obsiahnuté investičné inštrumenty a operácie nad nimi.

- List portfólií - Predstavuje zoznam užívateľom vytvorených portfólií. Voliteľné parametre každého portfólia sú názov portfólia a jeho poznámka. Užívateľ môže vytvoriť ľubovoľný počet portfólií, avšak portfólio musí mať v rámci jedného užívateľa unikátny názov. Portfólio je možné okrem vytvorenia aj vymazať. Pri vymazaní dochádza ku kontrole obsahu portfólia, ktorej výsledok môže skončiť troma scenármi. Ak je portfólio prázdne, to jest, neobsahuje žiadne inštrumenty, potom bude portfólio úspešne zmazané a užívateľ o tom bude graficky informovaný. Ak portfólio nebude prázdne a bude obsahovať investičné inštrumenty, pričom však užívateľ ani v jednom z nich nebude vlastniť žiaden podiel, užívateľ bude graficky vyzvaný k reakcií. Aplikácia umožní

výber reakcie, buď zrušením požiadavku odstránenia portfólia (nebudú vykonané žiadne zmeny) alebo potvrdením vymazania portfólia kedy bude z aplikácie odstránené portfólio a aj všetky inštrumenty s väzbou na toto portfólio. Posledná možnosť je ak užívateľ bude chcieť vymazať portfólio v ktorom existuje aspoň jeden inštrument s nakúpeným podielom. V tomto prípade aplikácia vyzve užívateľa k manuálnemu predaju podielu a zároveň oznámi, že neboli vykonané žiadne zmeny. Zakaždým, keď bude užívateľ vyzvaný k reakcii, bude informovaný, v ktorých inštrumentoch v danom portfóliu má podiel a v ktorých nie. Táto informácia má užívateľovi pomôcť k lepšej orientácii v rámci portfólia a rýchlejšiemu vyriešeniu problému s nakúpeným podielom.

- Portfólio - Obsahuje zoznam užívateľom pridaných investičných inštrumentov v danom portfóliu. Inštrumenty sú v rámci portfólia unikátne. Okrem pridávania je ich možné aj vymazať. Pri vymazaní aplikácia vykoná kontrolu podielu, ak užívateľ v danom inštrumente daného portfólia nemá žiaden podiel, potom je investičný inštrument z portfólia úspešne vymazaný a užívateľ graficky informovaný. Ak v ňom má podiel, potom je akcia zmazania prerušená a užívateľ je vyzvaný k predaniu svojho podielu a k opakovaniu akcie.
- Inštrument - Zahŕňa všetky operácie, ktoré vyžaduje aplikácia k zobrazeniu/nákupu/-predaju inštrumentu. Ak je inštrument súčasťou portfólia z ktorého vzniká požiadavka na jeho zobrazenie, potom je vykonávaná táto časť funkcionality, inak je užívateľ automaticky presmerovaný do nasledujúcej časti modulu - ukážka inštrumentu. Dáta určené k zobrazeniu majú dynamický charakter, čím sa myslí časový interval za aký sú dáta zobrazené. Základný časový interval je automaticky generovaný práve v momente načítania. Časový interval obsahuje dva hraničné body, začiatkový dátum a konečný dátum. Začiatkový dátum je z pohľadu časového plynutia vždy najstarší bod intervalu. Konečný dátum je naopak najskorší bod intervalu. Z toho vyplýva, že začiatkový dátum nemôže byť na časovej osi aktuálnejší než konečný bod a opačne, konečný dátum nemôže byť starší než začiatkový dátum. V špeciálnom prípade sa môžu začiatkový a konečný dátum rovnať, čo znamená, že je požiadavka získať dáta len pre daný konkrétny jeden dátum. Generovanie prebieha tak, že aplikácia zistí aktuálny lokálny čas na zariadení, na ktorom je aplikácia spustená. Zistený aktuálny lokálny čas je prehlásený za konečný dátum a z neho je odvodený začiatkový dátum, presne o mesiac skôr. Preto ak je užívateľ automaticky presmerovaný z portfólia, napríklad pre požiadavku zobrazit inštrument z daného portfólia, potom je inštrument zobrazený s najaktuálnejším mesačným časovým intervalom. Užívateľ potom môže v rámci grafického rozhrania požiadať aplikáciu o zobrazenie iného časového intervalu. Zároveň sú na výber dva typy grafov, kde sú dáta možné zobraziť - líniový a sviečkový graf. Tieto grafy môže užívateľ zväčšiť na plochu celej stránky. V prípade nákupu, užívateľ musí zvoliť parametre nákupu a to dátum v ktorý bola akcia kúpená, množstvo ktoré predstavuje počet akcií a posledným parametrom je cena akcie za jeden kus. V prípade predaja akcie sú parametre totožné ako pri nákupe s dvoma rozdielmi a to, že množstvo akcií, ktoré užívateľ chce predat, nesmie presiahnuť, množstvo akcií ktoré už vlastní a rovnako tiež nemôže akcie predat skôr ako ich nakúpil. Kontrolu integrity týchto dát ma na starosti aplikácia a pri neplatnej operácii je požiadavka stornovaná a užívateľ oboznámený o chybe. Pre obe operácie, nákup a predaj taktiež platí, že dátum nemôže byť z budúcnosti. Počet akcií, ktoré užívateľ z daného inštrumentu

v danom portfóliu vlastní je zobrazený na webovej stránke portfólia a rovnako ďalšie ukazovatele plynúce z nákupu a predaja podielov akcie.

- Ukážka inštrumentu - V tejto časti je návrh funkcionality takmer totožný ako pri prechádzajúcej časti - inštrument. Rozdielom je že ukážka inštrumentu je ochudobnená o možnosť nákupu/predaja podielu a z toho plynúcich ukazovateľov. Dôvodom je, že užívateľ je presmerovaný na ukážku inštrumentu iba v prípade, ak daný inštrument nie je súčasťou portfólia z ktorého požiadavka prišla a v ktorom sa užívateľ pred tým nachádzal.

4.3.3 Modul Historických dát

Poskytuje užívateľovi zobrazenie historických dát inštrumentu v tabuľkovej forme. Tabuľka zobrazuje OHLC¹ dáta investičného inštrumentu. Zobrazenie prebieha na samostatnej stránke na ktorú je možné sa dostať v dvoch režimoch:

- Zobrazenie historických dát inštrumentu je možné po zobrazení stránky z navigačného panelu. V tomto prípade, je nutné manuálne vybrať inštrument zo zoznamu a zároveň vybrať začiatočný a koncový bod časového intervalu za ktorý bude história inštrumentu zobrazená v tabuľke.
- V prípade, ak sa užívateľ nachádza na stránke inštrumentu, ktorý má pridaný vo svojom portfóliu, potom je možné automatické presmerovanie na stránku historických dát inštrumentu na žiadosť užívateľa. Aplikácia presmeruje užívateľa na stránku historických dát a automaticky vyplní inštrument a časový interval a to presne podľa rovnakých parametrov aké mal nastavené užívateľ na stránke inštrumentu. V podstate ide o zobrazenie dát z grafu na stránke inštrumentu v surovej podobe na stránke historických dát.

4.4 Návrh modulov logiky

Rieši všetky problémy aplikácie, ktoré je možné a zároveň praktické odčleniť od modulov technológie Django. Modul logiky predstavuje vrstvu aplikácie s ktorou komunikujú všetky Django moduly, ktorým poskytuje zdroj dát a spracovanie dát podľa nastavených parametrov pri volaní rozhrania vrstvy logiky. Jednotlivé návrhy súčasti nad ktorými vrstva logiky vytvára abstrakciu sú popísané detailnejšie v pokračovaní tejto sekcie.

Počas prieskumu existujúcich zdrojov dát som zistil, že finančné dáta sú extrémne cenné. Najst spoľahlivý neplatený zdroj bez akýchkoľvek obmedzení pri používaní je veľmi zložitý. Zároveň, existuje riziko, že neplatené zdroje sa časom stanú platenými alebo že ich spoľahlivosť nemusí byť do budúcnosti zaručená. Príkladom môže byť vývoj Yahoo! Finance API, ktorý podľa môjho prieskumu bola obľúbený zdroj finančných dát a dal sa využiť bez poplatku. Spoločnosť Yahoo! vývoj tohoto rozhrania pre verejnosť ukončila v roku 2017, pričom vykonala v ňom nezanedbatelné zmeny, ktoré narušili zdroj dát vo viacerých aplikáciách. Avšak v súčasnosti existujú možnosti a open-source knižnice, ktoré sa stále snažia získavať dáta z Yahoo! Finance API.

Pre históriu akou prešlo Yahoo! Finance API som sa rozhodol zakomponovať do návrhu dva zdroje finančných dát - IEX Cloud a Yahoo! Finance. Dôvod je aby si užívateľ mohol

¹OHLC (Open, High, Low, Close) = Otváracia, najvyššia, najnižšia, zatváracia cena obchodovaného dňa

vybrať zdroj finančných dát. Pri riešení IEX Cloud je predpoklad spoľahlivosti a stabilného vývoja v budúcnosti, nakoľko sa jedná hlavne o platenú službu. V prípade Yahoo! Finance naopak má prístup k dátam užívateľ zdarma, avšak vychádzajúc z historických udalostí je možné konštatovať, že budúcnosť môže byť neistá. Dôvod prečo nebol vybraný iný zdarma zdroj namiesto Yahoo! Finance je práve ich značná nedostupnosť. Napriek histórii Yahoo! Finance API som dospel k výsledku, že je to najspoľahlivejší a najviac rozšírený zdroj dát, ktorý je zdarma. Z toho zároveň pramenia ďalšie výhody, napríklad v prípade výpadku alebo zmeny v samotnom Yahoo! Finance bude náhrada/oprava teoreticky jednoduchšia a rýchlejšia, nakoľko to zasiahne mnoho projektov než pri menej známej službe.

Návrh aplikácie počíta aj s možnosťou rozšírenia o ďalšie zdroje dát a v prípade potreby ďalšieho riešenia by úpravy mali zasahovať najvyššiu vrstvu logiky.

4.4.1 IEX Finance Manager

IEX (Investors Exchange) je burza cenných papierov sídliaca v Spojených štátoch amerických. Modul IEX Finance Manager sprístupňuje aplikácií finančné dáta zo služby IEX Cloud. Služba IEX Cloud je platforma zaoberajúca sa finančnými dátami so širokým portfóliom možností a poskytuje rozhranie prístupné pomocou architektúry rozhrania REST. Vlastníkom burzy IEX a služby IEX Cloud je spoločnosť IEX Group, Inc.

IEX Cloud poskytuje širokú škálu dát a funkcionalít. Návrh počíta hlavne s dátami, ktoré by sa dali zhrnúť do niekoľkých bodov:

- Získanie zoznamu symbolov akcií, ktoré služba podporuje. Z týchto symbolov si užívateľ bude môcť vybrať vo vyhľadávacom zozname dostupných akcií a pridať si ich do portfólia.
- Detailné dáta o konkrétnej spoločnosti (Názov spoločnosti, primárne obchodované miesto, mena v ktorej je spoločnosť obchodovaná) a získanie dát vývoja cien za určité časové obdobie.
- Správy a novinky týkajúce sa spoločnosti.

V prípade využívania tohoto modulu je potrebné si zriadiť účet v službe IEX Cloud a získať autentifikačné údaje pre prístup k dátam v službe. Princíp fungovania služby je na kreditovom systéme, kedy užívateľ vlastní istý počet kreditov z ktorých sa mu postupne odoberá pri požiadavkách na službu. Služba rozlišuje dáta, ktoré vyžadujú istý počet kreditov a ktoré sú dostupné bez kreditov. To aká výška kreditu bude užívateľovi strhnutá pri získaní kreditovaných dát je závislé od počtu požiadavok a typu dát, pretože kreditové ocenenie je závislé na type dát. V jednoduchosti povedané, užívateľ vlastní počet kreditov, ktoré slúžia ako platidlo v rámci služby IEX Cloud a s ktorými si platí za dáta pričom existuje mnoho rôznych druhov dát a každý má inú kreditovú cenu.

Na výber je niekoľko možností ako užívateľ môže získať kredity:

- Zdarma účet - Je to základný účet bez poplatkov s veľmi nízkym počtom kreditov, ktorý sa po určitom čase (jeden mesiac) obnoví na pôvodnú hodnotu.
- Platený účet - Účet pre platiacich užívateľov služby. Užívateľ si kúpi balíček vo forme určitého počtu kreditov, ktorý je možné pri vyčerpaní navýšiť.
- Pieskovisko - Tento typ prihlasovacích údajov obsahuje aj zdarma aj platený účet. Predstavuje sekundárny zdroj a užívateľ v ňom má neobmedzené množstvo kreditov.

Negatívom je, že dáta z tohto zdroja sú nepresné/fiktívne, respektívne nemajú reálny charakter. Preto je vhodný pre účely vývoja a návrh aplikácie počíta primárne s týmto zdrojom, nakoľko sa predpokladá, že funkcionality dosiahnutá pomocou pieskoviska bude fungovať aj pri reálnych kreditovaných dátach a poskytuje pri vývoji testovanie, hoci na fiktívnych dátach.

Počas riešenia bakalárskej práce IEX Cloud neraz ponúkal výhodnejšie ponuky či rôzne zmeny, preto spôsob ako užívateľ môže nadobudnúť kredity sa môže časom zmeniť, čo zároveň nie je neočakávané v rámci akejkoľvek služby. Nemalým pozitívom, ktorý pomôže pri implementácii tejto aplikácie je podrobná a prehľadná dokumentácia IEX Cloud.

4.4.2 Yahoo! Finance Manager

Spracováva dáta zo služby Yahoo! Finance a sprostredkováva ich cez logické rozhranie zvyšku aplikácie. Ponúka menšie množstvo dát oproti IEX Cloud a návrh predpokladá prístupovanie skrz tretiu stranu, napríklad knižnicu jazyka Python. Výhodou je, že tieto dáta sú plne zadarmo a ich jediným možným negatívom je otázna funkčnosť v budúcnosti. Vďaka veľkej popularite však Yahoo! Finance ponúka dostatočne široké portfólio finančných inštrumentov na to aby požiadavkom tejto aplikácie vyhovel. Hoci samotné Yahoo! Finance nemá k svojmu rozhraniu zverejnenú oficiálnu dokumentáciu, tak vďaka svojej popularite existuje viacero knižníc/služieb tretích strán, ktoré zastrešujú určitú časť rozhrania a poskytujú aspoň obmedzenú dokumentáciu. Považujem to za prijateľný kompromis medzi možnosťou pracovať s finančnými dátami zdarma a možnými komplikáciami pri implementácii.

4.4.3 Predikcia

Umožňuje zobrazit predpokladaný vývoj ceny inštrumentu. Užívateľ má na výber z viacero možností, podľa ktorých sa určí predikcia. Na výber sú jednoduché matematické metódy, ktorým môže užívateľ meniť parametre - aký základ sa pri výpočte použije a počet dní/hodín, ktoré budú predikované. Zobrazená môže byť vždy len jedna predikcia (to jest jedna matematická metóda). Výpočet je zobrazený v grafe a je ho možné zmazať pre účely zobrazenia inej matematickej metódy alebo pre odstránenie predikcie celkovo. Hodnoty, ktoré sú v grafe zobrazené ako predikcia sa farebne líšia od pôvodných reálnych dát.

4.4.4 Iné funkcie programu

Zahrňuje malé špecifické funkcionality aplikácie pre ktoré by bol ale vytváranie zvlášť modulu zbytočné. Každá funkcionality bude zvlášť umiestnená do jedného súboru s príponou pre programovací jazyk Python. Konkrétnejší popis funkcionality bude viac rozpísaný v implementácii, nakoľko až vtedy bude zrejmejší obraz o aké funkcionality pôjde a aká bude ich vhodná dekompozícia a začlenenie v rámci aplikácie.

4.4.5 Návrh systému upozornení

Užívateľ si môže v rámci stránky investičného inštrumentu zadať upozornenie. Systém upozornení vyžaduje aby mal vyplnenú v profile emailovú adresu, inak bude vyzvaný aby túto informáciu doplnil.

Systém upozornení podporuje poslanie upozornenia pre prekročenie hodnoty ceny, ktorú zadá užívateľ. Ďalšou možnosťou je zadanie percentuálneho nárastu ceny, ktorý je počítaný

z ceny kedy bolo upozornenie zadané voči cene kedy bude upozornenie vyhodnocované. Pri výbere môže špecifikovať či ide o upozornenie v rámci prebehu dňa alebo v rámci priebehu viacerých dní.

Upozornenie príde vo forme emailu s presným vzorom predmetu a správy, ktorý bude viac popísaný v implementácií.

4.5 Návrh užívateľského prostredia

Užívateľské prostredie je grafické rozhranie prostredníctvom ktorého užívateľ komunikuje s aplikáciou. Cieľom návrhu je vytvoriť jednoduché prostredie v ktorom sa užívateľ nebude mať problém orientovať a poskytne mu všetky funkcionality aplikácie v prehľadnej podobe. Každý Django komponent spomenutý vyššie obsahuje HTML stránku štylizovanú kaskádovými štýlmi. Funkcionalitu tejto stránky dopĺňa programový kód v jazyku Javascript s ktorým je možné dosiahnuť vytváranie reakcie stránky na akcie vykonané užívateľom. Nezanedbateľne je využívaná technológia Bootstrap verzie štyri, ktorá pomáha pri vytváraní rozmermi prispôsobivých a dobre štruktúrovaných stránok a zároveň dopĺňa stránku o predom vytvorené komponenty.

V nasledujúcich pod-sekciách bude detailnejšie rozobraný návrh každého Django komponentu. Návrhy dizajnu sú robené pomocou služby Moqups². Podľa aktuálnych podmienok služby Moqups je povolené využívať výstupy zo služby v rámci študentských prác³.

4.5.1 Navigačná lišta

Zobrazuje sa na každej stránke v jej hornej časti. Obsahuje popis a presmerovanie na iné stránky. Stránka na ktorej sa užívateľ práve nachádza je voči ostatným popisom zvýraznená. Ak užívateľ klikne na ľubovoľný popis, potom ho aplikácia presmeruje na danú stránku. Z pohľadu návrhu aplikácie, každý popis de facto predstavuje niektorí z Django modulov.



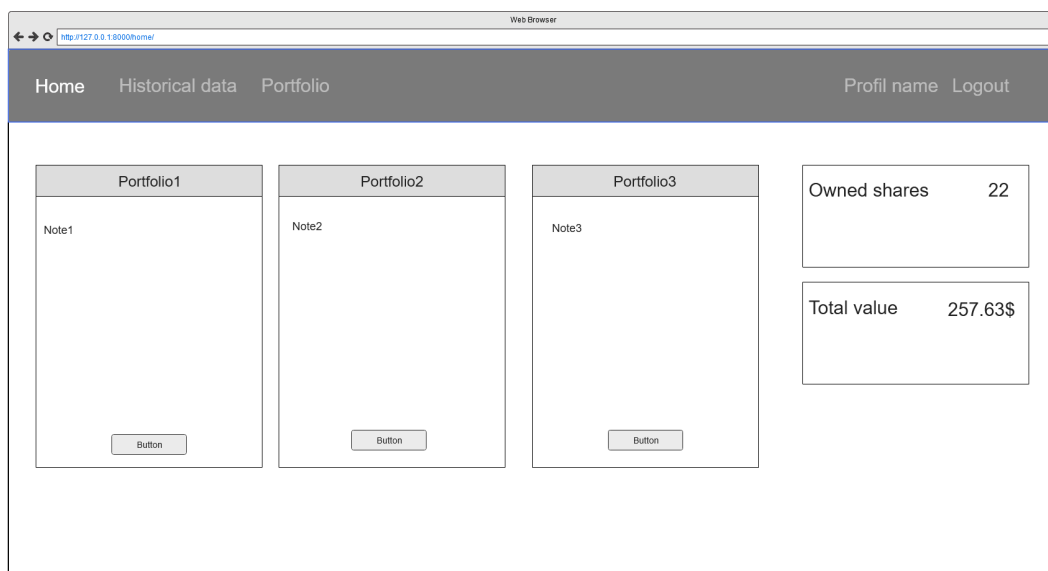
Obrázek 4.3: Návrh navigačnej lišty

²<https://moqups.com/>

³<https://moqups.com/faq/>

4.5.2 Domovská stránka

Uživateľ by tu mal vidieť celkový prehľad svojho účtu. Po pravej strane je zobrazený počet aktuálne vlastnených akcií naprieč všetkými investičnými inštrumentami vo všetkých portfóliách a ich celkovú cenu. V centre stránky sú umiestnené nanejvýš tri karty. Každá karta má vo svojej hlavičke názov portfólia a v päte karty tlačidlo. Každá karta tak predstavuje konkrétne portfólio, ktoré si užívateľ vytvoril. Obsah portfólia tvorí poznámka, ktorú užívateľ vyplnil pri jeho založení. Pri kliknutí na tlačidlo sa ostatné karty portfólií skryjú (ak existujú) a nahradí ich karta o výsledkoch portfólia. Užívateľ zistí tri aktuálne najvýkonnejšie inštrumenty daného portfólia a zároveň tri inštrumenty, ktoré boli v danom momente najmenej výkonné. Výkonnosť inštrumentu sa v tomto prípade myslí jeho percentuálna zmena dvoch najskorších dní v ktorých už obchodovanie bolo ukončené a to z dôvodu, že sa využíva práve uzatváracia cena inštrumentu za daný deň. Ak si užívateľ vytvoril viac než tri portfólia, potom sú na domovskej stránke dostupné po stlačení tlačidiel nad kartami, ktoré umožňujú zobrazit ďalšie portfólia.

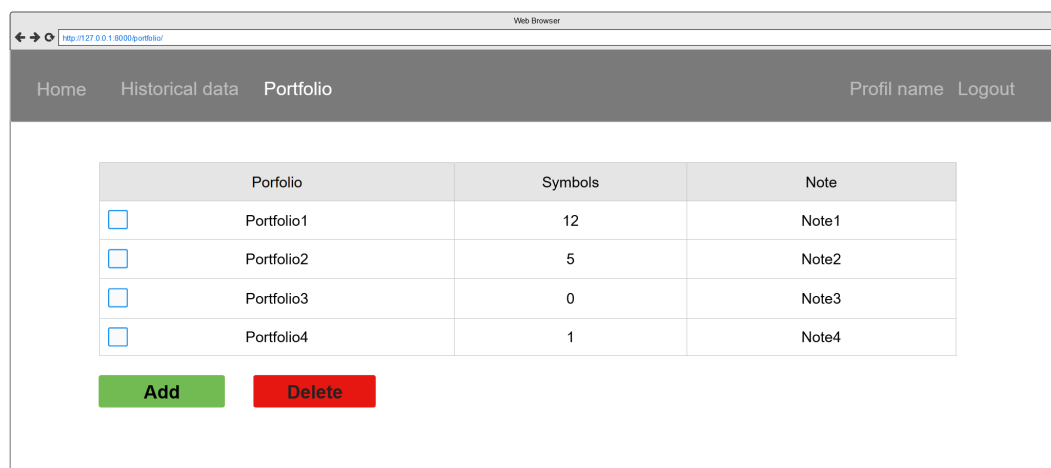


Obrázek 4.4: Návrh domovskej stránky

4.5.3 Stránka listu portfólií

Zobrazuje do tabuľky všetky portfólia, ktoré užívateľ vytvoril. Umožňuje vykonávanie operácií s portfóliami, pridať a odstrániť, pomocou tlačidiel pod tabuľkou. Pri odstránení je nutné v tabuľke zvoliť, ktoré portfólia majú byť odstránené. Portfólií môže byť naraz odstránených viac a to systémom, ktorý bol popísaný v module portfólia v časti 4.2.2. Tabuľku je možné zotriediť pomocou ľubovlného stĺpca. Kliknutím na názov portfólia bude užívateľ presmerovaný na správu daného portfólia. V tabuľke sú po stĺpcoch zobrazené detaily portfólia:

- Názov - Užívateľom zvolený názov portfólia, ktorý je zadaný pri vytváraní portfólia a je unikátny v rámci daného užívateľa.
- Symboly - Počet symbolov (investičných inštrumentov), ktoré v danom momente portfólio obsahuje.
- Poznámka - Užívateľom zadaná poznámka pri vytváraní portfólia.



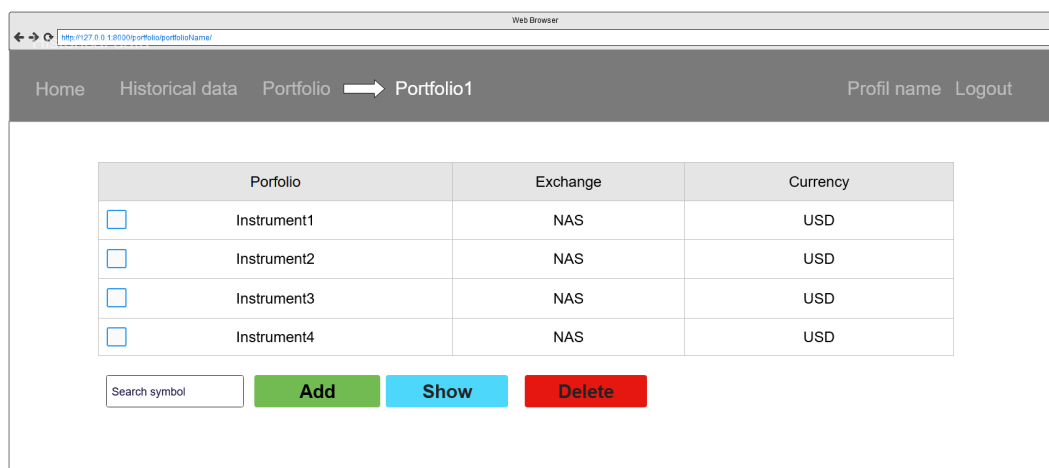
Obrázek 4.5: Návrh stránky pre výber portfólia

4.5.4 Stránka portfólia

Zobrazuje investičné inštrumenty konkrétneho portfólia, ktoré si užívateľ pridal. Investičné inštrumenty sú zobrazené v tabuľke po riadkoch. Stĺpce tabuľky udávajú nasledovné parametre jednotlivých investičných inštrumentov:

- Symbol - Burzový symbol, ktorý jednoznačne identifikuje konkrétny investičný inštrument na konkrétnom trhu.
- Názov - Plný názov spoločnosti.
- Burza - Skratka burzy, na ktorej je investičný inštrument obchodovaný.
- Mena - Skratka meny, ktorá je všeobecne zaužívaná skratka konkrétnej menovej jednotky.
- Region - Geografická skratka alebo popis regiónu v ktorom sa inštrument obchoduje.

Operácie nad investičnými inštrumentami v tabuľke sú sprostredkované cez tri tlačidlá pod tabuľkou a políčkou.



Obrázek 4.6: Návrh stránky pre správu portfólia

Políčko slúži ako vyhľadávač dostupných investičných inštrumentov. Pri zadaní viacerých znakov zobrazí ponuku najviac piatich symbolov investičných inštrumentov. Tie sú vyberané z listu všetkých dostupných symbolov na základe zhodného prefixu.

Tlačidlá umožňujú operácie podľa popisu - pridať, zobraziť, vymazať.

Ak chce užívateľ pridať nový investičný inštrument do portfólia, musí ho najskôr vyhľadať vo vyhľadávači a potom kliknúť na tlačidlo pridať. Aplikácia následne získa/spracuje údaje a uloží potrebné informácie do databázy. Stránka po ukončení operácií automaticky aktualizuje tabuľku s investičnými nástrojmi. Pri celom priebehu pridávania sa zobrazí na tlačítku symbol načítavania (otáčajúce sa kolečko) a užívateľ je tak informovaný graficky, že operácia sa vykonáva. Čas za ktorý sa operácie dokončia sú závislé na výkone servera a rýchlosti internetového spojenia, keďže potrebné údaje musia byť stiahnuté zo serverov vybranej služby.

V prípade, že užívateľ chce investičný inštrument z portfólia vymazať, môže tak urobiť označením riadku s daným investičným inštrumentom a kliknutím na červené tlačidlo zmazať. Operácia zmazania prebehne úspešne ak užívateľ v danom investičnom inštrumente

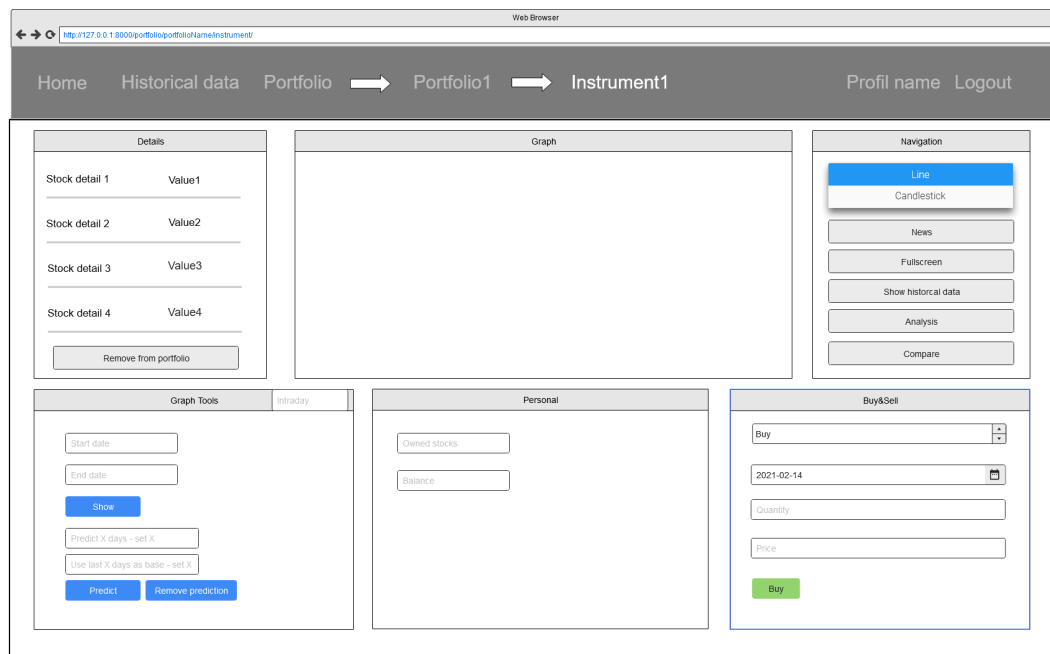
nevlastní žiaden podiel. Ak vlastní, potom je operácia neúspešná. Užívateľ je v oboch prípadoch graficky informovaný o výsledku. Zároveň je možné vymazať aj viac investičných inštrumentov pomocou označenia viacerých riadkov naraz a stlačenia tlačidla vymazať. V tomto prípade, všetky inštrumenty, ktoré budú môcť byť vymazané, budú vymazané a naopak. O všetkých výsledkoch bude užívateľ opäť graficky informovaný.

Posledným tlačidlom ponuky je tlačidlo zobrazenia. Toto tlačidlo má účel v prípade, ak užívateľ chce zobraziť detailnejšie dáta o investičnom inštrumente bez toho aby si ho pridal do portfólia. Rovnako ako v prípade pridania, tak v prípade zobrazenia je podmienkou zadania investičného symbolu do políčka, potom môže byť stlačené tlačidlo zobrazenia a vykonané ďalšie operácie. V jednoduchosti, zobrazenie investičného inštrumentu pomocou tlačidla zobrazenia presmeruje užívateľa na podobnú stránku ako pri zobrazení investičného inštrumentu, ktorý je v portfóliu pridaný, s tým rozdielom, že v tomto prípade stránka nedovoľuje vykonávať operácie, ktoré sú výhradne iba pre pridané investičné inštrumenty - nákup/predaj, zobrazenie histórie, analýza investičného inštrumentu a iné.

Pre účel zobrazenia investičného inštrumentu z tabuľky je možné kliknúť na položky v stĺpci symbolu. Užívateľ bude následne presmerovaný na stránku investičného inštrumentu.

4.5.5 Stránka investičného inštrumentu

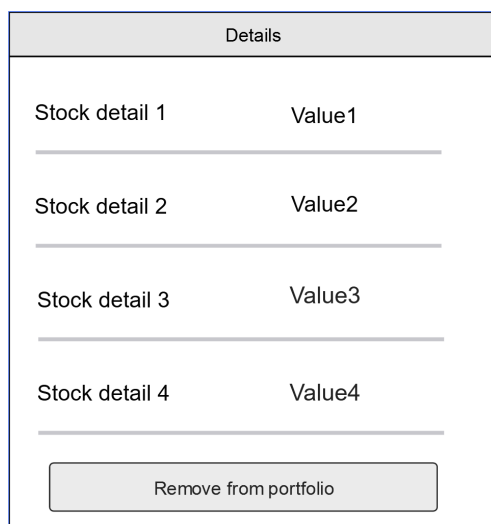
Táto stránka má za úlohu poskytovať užívateľovi detailnejšie dáta o investičnom inštrumente a vykonávať ďalšie operácie s inštrumentom. Ponúka tiež viacero spôsobom akým sa tieto dáta dajú zobraziť. v počítačom strave je stránka rozdelená do šesť malých okien, ktoré každé má špecifickú funkcionality. Každú z nich teraz rozoberiem v samostatnom odseku.



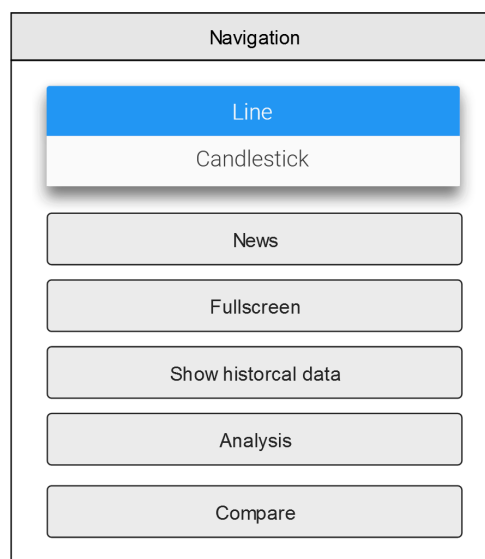
Obrázek 4.7: Návrh pre zobrazenie a správu investičného inštrumentu

Okno s detailami sa nachádza vľavo hornej strane stránky a slúži na zobrazenie detailnejších informácií o investičnom nástroji. Informácie sú vsadené pod seba idúcich buniek, ktoré sú pre lepšiu prehľadnosť oddelené úsečkou. Ľavá strana bunky obsahuje názov informácie a pravá strana bunky obsahuje jej hodnotu. Súbor zobrazených informácií je vybraný tak, aby bol dostupný z oboch zdrojov dát (IEX Cloud, Yahoo! Finance). Na spodnej strane okna sa nachádza tlačidlo pre odobratie nástroja z portfólia. Pre úspešné odobratie musí byť predaný všetok nakúpený podiel užívateľa v investičnom nástroji.

Navigačné okno sa nachádza vpravo hornej strane stránky a sú v ňom umiestnené všetky prvky navigácie stránky. Prvky teraz opíšem prechádzaním z vrchu okna smerom nižšie. Na úplnom vrchu okna sa nachádza dvojica kariet, ktoré ovládajú zobrazenie v okne grafu (ktoré je detailnejšie popísané nižšie). Táto dvojica kariet je na obrazovke prítomná vždy, avšak môžu sa nachádzať na inom mieste v prípade, že okná sú preskupené z dôvodu vyvolania iného stavu zobrazenia stránky (režim celej obrazovky, režim porovnávania). Karty svojou farebnosťou informujú užívateľa o tom, ktorá karta je práve aktívna. Kartu je možné zvoliť kliknutím na ňu. Nižšie od kariet sa nachádzajú tlačidlá navigácie. Tlačidlo noviniek slúži na zobrazenie nadpisov a v nich ukrytých článkov. Užívateľ je tak informovaný o novinkách týkajúcich sa daného investičného nástroja. Každý záznam o novinke obsahuje zdroj, odkiaľ bol článok čerpaný.

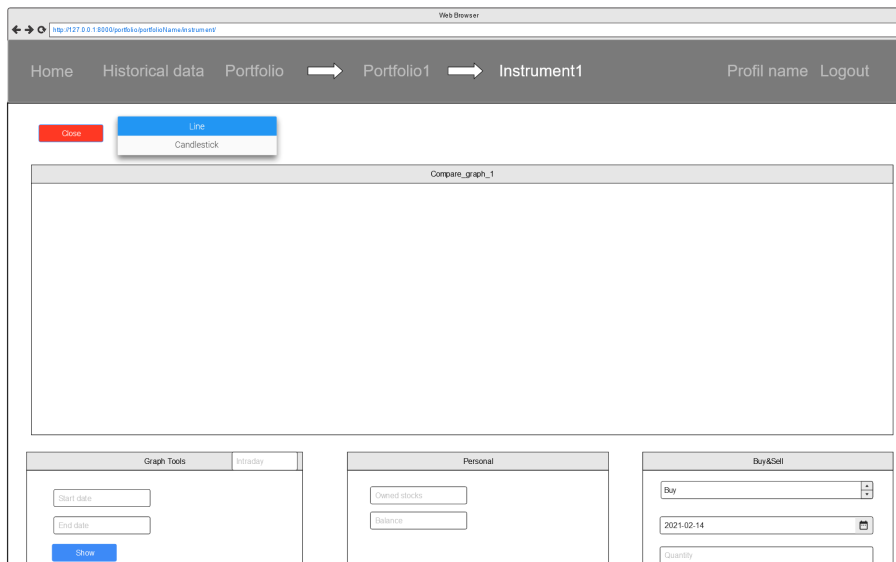


Obrázek 4.8: Návrh okna detailov



Obrázek 4.9: Návrh okna navigácie

Pod tlačidlom novinek na nachádza tlačidlo pre aktivovane režimu celej obrazovky. Tento režim rozšíri rozmery okna grafu na väčšinu časti obrazovky. Dovolí tak užívateľovi využiť celú obrazovku na prehládanie dát zobrazených v grafe. Ovládanie okna grafu sa presunie nad okno grafu. Okno detailov a navigácie v tomto režime nie sú dostupne.

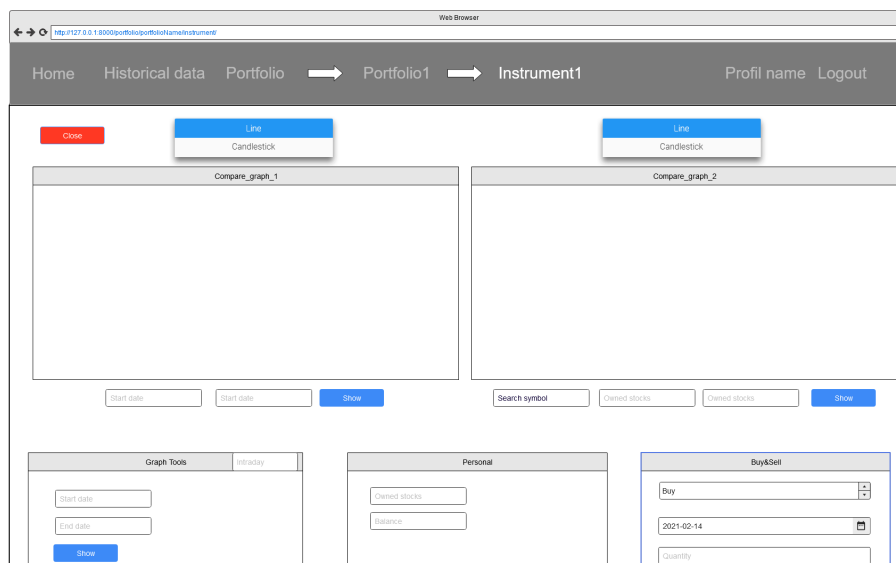


Obrázek 4.10: Návrh pre zobrazenie a správu investičného inštrumentu - režim celej obrazovky

Ďalším tlačidlom je zobrazenie historických dát. Pri kliknutí bude užívateľ presmerovaný na stránku historických dát, kde bude vidieť surové dáta (dátum, otváracia cena, najvyššia cena, najnižšia cena, uzatváracia cena, obchodovaný objem) daného investičného inštrumentu v tabuľke. Tejto stránke sa bude venovať samostatná kapitola, nakoľko nie je súčasťou stránky investičného inštrumentu.

Rovnako ako tlačidlo pre zobrazenie historických dát tak aj tlačidlo pre analýzu odkazuje na zvlášť stránku, ktorá sa celá venuje analýze inštrumentu. Táto stránka bude taktiež detailne popísaná v zvlášť sekcii.

Posledným tlačidlom v navigačnom okne je pre zobrazenie režimu porovnania. V tomto režime sú okna detailov a navigácie opäť skryté. Doteraz existujúce okno grafu sa presunie na ľavú časť stránky a zároveň na pravej strane stránky sa objaví nové, nezávislé okno grafu. Oba grafy majú svoje samostatné ovládanie. Nad grafmi je ponuka pre výber typu grafu. Pod grafmi sú polia kde je možné vybrať počiatočný a koncový dátum pre zmenu časového intervalu zobrazených dát v grafe. Nový graf pre porovnávanie (graf na pravej strane) má navyše pole pre zadanie iného investičného inštrumentu aby mohol užívateľ porovnávať vývoj aj dvoch rôznych investičných inštrumentov.



Obrázek 4.11: Návrh pre zobrazenie a správu investičného inštrumentu - režim porovnávania

V okne grafu je vždy zobrazený jeden z dvoch typov ponúkaných grafov (grafy implementované v aplikácii sú popísané v časti 2.8). Grafy sú interaktívne, umožňujú užívateľovi zobraziť hodnotu investičného inštrumentu v konkrétnom bode. V počiatočnom stave (rozumie sa stav, ktorý nastane ihneď po vykreslení prijatých dát) sú v grafoch zobrazené všetky získané dáta za zvolený časový interval (os X). Grafy tiež podporujú zobrazenie len určitého úseku z časového intervalu. To je dôležité v prípade ak užívateľ požiada aplikáciu o veľký časový interval dát. Aplikácia zvolený rozsah dát získa a grafy ho v počiatočnom stave vykreslia celý. Avšak z dôvodu zobrazenia veľkého rozsahu je krátkodobá analýza vývoja cien obtiažnejšia. Práve v tomto prípade si môže užívateľ zobrazený časový úsek zúžiť na konkrétny časový pod-interval. Nastavenia a pohyb v tomto zobrazení naprieč celým časovým intervalom sa vykonáva priamo v grafoch. Ak je časový interval dostatočne veľký, graf ponúkne užívateľovi prednastavené pod-intervaly, ktoré je možné aktivovať tlačidlami. Zobrazenie rozsahu grafov na osi Y reflektuje extrémny (minimum a maximum) hodnôt v získaných dátach aby prípadne zmeny vývoja cien boli jasne viditeľné. Na požiadavok užívateľa graf vykreslí dáta určené na predikciu. Dáta predikcie sú vykreslené inou farbou ako pôvodné dáta aby ich užívateľ od seba vedel rozlíšiť. Predikcia je zobrazená len v čiarovom grafe. Dôvodom je, že sviečkový graf obsahuje hodnoty, ktoré vo výpočte predikcie nie sú zahrnuté a teda predikciu neovplyvňujú, preto nie sú ani predikované.

Okno nástrojov grafu sa nachádza vľavo dolnej časti stránky. Jeho úlohou je ponúkať režimy okna grafu (v priebehu dňa⁴ a medzi dňami⁵) a umožniť užívateľovi meniť zobrazené dáta v okne grafu. Pri návrhu je zohľadnené, že užívateľ ako investor sa môže zaujímať o dlhodobý alebo aj krátkodobý vývoj cien investičného inštrumentu. Vychádza to z cieľov a úmyslov konkrétneho investora, ktorý bol popísaný v sekcii, ktorá sa venovala investovi v časti 2.2. Tieto režimy sú na stránke od seba navzájom oddelené, to znamená, že zmena dát (načítanie nových dát v inom časovom intervale) napríklad v zobrazení dňa sa neprejaví v režime zobrazenia viacerých dní. Je to z dôvodu rôznej charakteristiky dát medzi týmito dvoma režimami. Zatiaľ čo v zobrazení dňa sa nachádzajú dáta s minútovou

⁴V priebehu dňa = intraday (anglický výraz)

⁵Medzi dňami = interday (anglický výraz)

aktualizáciou, v prípade zobrazenia viacerých dní sa dáta zameriavajú na dennú aktualizáciu. Ďalej sa tu nachádzajú dve textové polia pre zmenu časového intervalu zobrazených dát v grafe. V zobrazení dňa sa jedná o jedno textové pole, pri zobrazení viacerých dní sa jedná o dve textové polia - začiatkový a konečný dátum. Začiatkový dátum sa myslí ako začiatok uzavretého intervalu a konečný dátum ako koniec uzavretého intervalu, tak ako bolo popísané v detailnejšom popise inštrumentu v časti 4.2.2. Potvrdenie pre vykonanie zmeny je prostredníctvom tlačidla *zobraziť*. K obsahu okna nástrojov grafu patrí tiež nastavenie a vytvorenie predikcie. Užívateľ môže zobraziť práve jednu predikciu. Ak chce parametre pre predikciu zmeniť, musí predikciu odstrániť, zmeniť parametre a znova vytvoriť predikciu. Na vytvorenie a zmazanie predikcie slúžia tlačidlá pod textovými poliami. Parametre predikcie je možné zadať do dvoch textových polí. Horné textové pole patriace predikcií slúži pre definovanie počtu dní, ktoré budú predpovedané. Druhé textové pole slúži na zadanie počtu dní, z ktorých bude predikcia vykonaná. Ak toto pole nie je zadane, výpočet je automaticky zo všetkých dní, ktoré sú zahrnuté v časovom intervale zobrazeného grafu v okne grafu. Obe textové polia predpokladajú ako vstup prirodzeného čísla s výnimkou nuly.

Obrázek 4.12: Návrh okna nástrojov grafu - stránka investičného inštrumentu.

Okno napravo od okna nástrojov grafu sa špecializuje na osobný status užívateľa v rámci investičného inštrumentu v danom portfóliu. Obsahuje textové polia určené len pre čítanie. Tieto textové polia zobrazujú aký podiel užívateľ vlastní a aká je jeho hodnota v čase keď bola stránka načítaná, prípadne keď bol zmenený počet vlastneného podielu.

Na koniec zostáva okno umiestnené v pravej dolejši časti stránky, okno nákupu a predaja. Toto okno sprostredkováva možnosť nákupu/predaja pre daný investičný inštrument v danom portfóliu. V okne je možné vybrať aký úkon chce užívateľ vykonať - nákup alebo predaj. Pri nákupe je návrh jednoduchý, užívateľ zadá konkrétny dátum nákupu, cenu za jeden podiel v investičnom inštrumente a veľkosť (počet) podielu. Pri predaji je nutné aby aplikácia nedovolila predaj väčšieho množstva podielu než aký užívateľ v danom momente vlastní. Rovnako aplikácia musí zaistiť aby užívateľ nemohol predať podiel skôr než ho nakúpi. Aplikácie totižto dovoľuje pre potreby analýzy a testovania nákup akcií aj do minulosti. Avšak detailnejšiemu algoritmu, ako budú tieto operácie riešené sa bude venovať sekcia implementácie aplikácie.

4.5.6 Stránka historických dát

Umožňuje náhľad na surové dáta všetkých investičných inštrumentov, ktoré má užívateľ pridané naprieč všetkými portfóliami. Dáta sú zobrazené za zvolený časový interval, podobne ako na stránke investičného inštrumentu. Zobrazenie dát je prostredníctvom tabuľky, kde dni sú členené po riadkoch. Prvý stĺpec tabuľky definuje, o aký deň sa v danom riadku jedná. Ďalšie stĺpce definujú ďalšie hodnoty z dňa. Každý stĺpec definuje konkrétnu hodnotu, ktoré sú:

- Dátum - Definuje konkrétny dátum riadku. Je to deň v ktorý sa na burze obchodovalo.
- Otváracia cena - Definuje počiatočnú cenu pri začatí obchodovania na burze v daný deň.
- Maximum - Definuje maximálnu cenu, ktorú investičný inštrument dosiahol v priebehu daného dňa.
- Minimum - Definuje minimálnu cenu, ktorú investičný inštrument dosiahol v priebehu daného dňa.
- Uzatváracia cena - Definuje koncovú cenu pri ukončení obchodovania na burze v daný deň.
- Obchodovaný objem - Definuje počet (objem) obchodovaný podielov za celý daný deň.

4.5.7 Stránka analýzy inštrumentu

Analýza inštrumentu je dôležitá v prípade, že užívateľ chce vidieť detaily svojho pôsobenia na trhu pri konkrétnom finančnom inštrumente. Obchodovanie na burze môže byť dlhodobá činnosť a zobrazenie priebehu obchodovania je tým pádom očakávaná vlastnosť aplikácie.

Na túto stránku sa môže užívateľ dostať z pridaného inštrumentu v portfóliu. Zobrazenie analýzy potom prebieha práve s týmito parametrami - pred daný inštrument v danom portfóliu. Zároveň táto stránka sa zobrazí len v prípade, že užívateľ vlastní aspoň jeden podiel investičného inštrumentu.

Užívateľovi je zobrazená tabuľka v ktorej sú zobrazené finančné pohyby užívateľa. Dátum, cena, množstvo a celková suma každého nákupu a predaja, ktorý užívateľ historicky vykonal. Nákup a predaj sú pre lepší prehľad farebne odlišené.

Kapitola 5

Použité technológie

V tejto kapitole sú predstavené použité technológie vo webovej aplikácii. Bol kladený dôraz na čo najväčšiu efektivitu pri písaní kódu z dostupných možností ako aj jeho preložitelnosť naprieč viacerými operačnými systémami.

5.1 Programovací jazyk Python

Je to skriptovací programovací jazyk. Zaraduje sa medzi vysoko-úrovňové programovacie jazyky. Podporuje objektovo orientovaný prístup programovania, vďaka ktorému je možné aplikáciu navrhnuť ako viacero modulov (objektov), ktoré medzi sebou komunikujú a sú na sebe nezávislé. Jednotlivé moduly je potom možno zoradiť do akýchsi pomyselných vrstiev aplikácie. Každá vrstva aplikácie môže mať svoju jasne danú úlohu. Pri plnení danej úlohy môže využívať rozhranie nižších vrstiev, pričom využívanie nižších vrstiev je transparentné. Podobný princíp je uplatnený aj v implementácii aplikácie tejto bakalárskej práce.[22]

Výhody programovacieho jazyka Python sú v jeho jednoduchej a dobre čitateľnej syntaxy, vysokej abstrakcii, kompatibilite s rôznymi druhmi operačných systémov a prehľadnej dokumentácii¹. Dokopy všetky tieto výhody umožňujú vytvoriť prehľadný a robustný systém. Jazyk Python navyše disponuje možnosťou zrýchlenia pomocou knižníc C++ prípadne iných optimalizačných knižníc². Jedná sa teda o perspektívny jazyk do budúcnosti so širokou škálou využitia.

Ďalší dôvod k výberu jazyka Python pre moju bakalársku prácu je jeho vysoká popularita³. Tá priamo súvisí väčšími možnosťami pri návrhu a implementácii aplikácie. Aplikácia využíva pre získanie dát (o burzách, o aktuálnych hodnotách investičných nástrojov a podobne) aplikácie tretích strán, ktoré majú svoje aplikačné rozhranie. Väčšina z nich je (alebo má alternatívu) v jazyku Python⁴. To umožňuje aby aplikácia podporovala získavanie dát z viacerých zdrojov. Taktiež je predpoklad pre ľahkú rozšíriteľnosť o ďalšie zdroje dát v prípade potreby.

¹<https://docs.python.org/3/>

²<https://docs.python.org/3/extending/extending.html>

³<https://insights.stackoverflow.com/survey/2020>

⁴<https://financetrain.com/best-python-librariespackages-finance-financial-data-scientists/>

5.2 Django

Je to slobodne šíriteľná nadstavba jazyka Python určená pre vývoj webových aplikácií postavená na návrhovom vzore dátového modelu aplikácie, riadiacej logiky a užívateľského prostredia.

Dôvod prečo som si vybral Django ako nadstavbu na ktorej budem stavať webovú aplikáciu je okrem iného jeho pestrá a dobre spracovaná dokumentácia⁵. Rovnako dáva vývojárovi slobodu v tom ako bude projekt navrhnutý, keďže obsahuje mnoho funkcionalít ako sa dá k projektu pristupovať. Django je vyvíjané s dôrazom na vysokú abstraktnosť kódu, vďaka ktorej nie je nutné pre väčšiu funkcionalitu písať veľké bloky kódu, čím sa stáva kód prehľadnejším a umožňuje výraznejšiu jednoduchšiu údržbu kódu do budúcnosti ako aj možnosť ľahšej rozšíriteľnosti o ďalšiu funkcionalitu. [7]

Ďalšou výhodou je už vopred vybudované rozhranie pre administráciu aplikácie. Administrátor má možnosť spravovať účty užívateľov, vytvárať skupiny s rôznymi priradenými právami do ktorých je možné pridať užívateľov na ktorých sa potom budú dané práve skupiny vzťahovať. Je možné pridať aj ďalšie funkcionality, ktoré administrátor vyžaduje pre lepšiu správu aplikácie.[6]

Django tiež obsahuje aj vysokú abstrakciu po stránke bezpečnosti aplikácie. Má implementované ochranné prvky, ktoré majú zabrániť základným útokom na webové aplikácie. Všetky tieto zabezpečenia má Django implementované priamo v sebe a nie je nutné využívať knižnic alebo rozšírení tretích strán. Ich aplikácia je podmienená zavolaním metód v kóde. Spomeniem niekoľko typov najbežnejších útokov proti ktorým je Django pripravené, napríklad: [8]

- *Cross-site request forgery (CSRF)* - Django má vstavaný modul pre ošetrenie útoku CSRF, čo je technika, ktorou môže útočník vykonať nelegitímne operácie s využitím prihlasovacích údajov práve prihláseného užívateľa s cieľom, aby o tom obeť nemala tušenia. Ochrana na úrovni vstavaného modulu Django je v tomto prípade obmedzená, avšak pokrýva väčšinu variácií ako je možné tento typ útoku vykonať.
- *SQL injection* - Je to typ útoku kedy môže útočník spustiť na databáze, ktorú využíva aplikácia, ľubovlný kód v jazyku SQL. Nebezpečenstvo spočíva k úniku alebo trvalej strate údajov.
- *Clickjacking* - Nebezpečenstvo spočíva v nevedomosti, že užívateľ môže kliknúť na stránke na objekt, ktorý je typicky priehľadný (aby užívateľ nevedel o jeho existencii) a tým spôsobiť rôzne škody. Django obsahuje vstavaný modul, ktorý zabraňuje vykresleniu v rámci. Nevýhodou je, že tento prístup ochrany musí byť podporovaný webovým prehliadačom, čo nemusí byť samozrejmosťou.

Každý projekt obsahuje koreňový adresár. Je to v podstate aplikácia vytvorená a konfigurovaná automaticky pri vytvorení projektu. Táto aplikácia obsahuje všetky dôležité nastavenia a konfiguračné súbory projektu. Medzi najhlavnejšie súbory koreňového adresára patria:

- `__init__.py` - Tento súbor je prázdny a jeho prítomnosť hovorí, že tento adresár je programový balík.

⁵<https://docs.djangoproject.com/en/3.2/>

- `setting.py` - Je to miesto kde sa prepájajú jednotlivé aplikácie projektu a podľa ktorého Django vie aké aplikácie existujú. Rovnako sa sem umiestňujú všetky nastavenia a parametre aplikácie.
- `urls.py` - Je to súbor v ktorom sú umiestnené všetky URL adresy aplikácie. Prezentuje všetky koncové body aplikácie.
- `manage.py` - Obsahuje funkcionality pre príkazy v príkazovom riadku vďaka ktorým je možné napríklad:
 - `runserver` - Príkaz slúžiaci k spusteniu webového servera, ktorý je nutný pre fungovanie webovej aplikácie.
 - `makemigration` - Príkaz vykoná vygenerovanie príkazov SQL pre všetky aplikácie, ktoré sú zaregistrované v projekte.
 - `migration` - Príkaz ktorý vykonáva vygenerované príkazy SQL a tým fyzicky vykonáva zmeny v databáze aplikácie.
- `wsgi.py` - Súbor je generovaný automaticky a obsahuje konfiguráciu pre server WSGI, ktorý slúži na nasadenie webovej aplikácie na server. WSGI špecifikuje ako webová aplikácia komunikuje so servermi.

Ďalej sú v projekte vedľa koreňového adresára aj ďalšie adresáre, takzvané aplikácie. Pomocou aplikácií Django implementuje princípy modulov. Projekt je tak rozdelený medzi viaceré moduly, kde vždy je prítomný hlavný modul (koreňový adresár), ktorý môže byť doplnený o vedľajšie moduly (aplikácie). Koreňový modul taktiež si môžeme predstaviť ako špeciálny typ aplikácie. Princíp modularity je výhodný v prípade, že vieme rozdeliť jednotlivé funkcionality aplikácie do viacerých pomyselných častí, ktoré majú svoje rozhranie, ktoré využívajú na komunikáciu so svojím okolím. Tieto časti môžeme potom pridávať a odstraňovať, prípadne nahradzovať novými s rovnakým rozhraním. Môžeme napríklad v jednej z častí programu vylepšiť logiku, ktorá značne zrýchli výpočet, avšak jej rozhranie sa nezmení a pre ostatné časti aplikácie to nebude predstavovať žiadnu potrebnú zmenu v ich implementáciách, aby program fungoval rovnako korektne ako pred zmenou. Princíp modularity preto zjednodušuje údržbu aplikácie a rovnako pridávanie/odstraňovanie funkcionality či vylepšenie jednotlivých modulov bez toho aby zmeny zasahovali aj do iných častí programu mimo daný upravovaný modul. Ako už bolo spomenuté, dané pomyselné časti programu sa v technológii Django volajú aplikácie. Základnými súbormi, ktoré je možné nájsť v Django moduloch (aplikáciách) sú:

- `__init__.py` - Tento súbor je prázdny a jeho prítomnosť hovorí, že tento adresár je programový balík.
- `admin.py` - Súbor slúži pre registráciu modelov, ktoré majú rozšíriť základnú funkcionality rozhrania pre administráciu aplikácie.
- `apps.py` - V súbore sa nachádza konfigurácia pre konkrétnu aplikáciu v ktorej je súbor umiestnený. Je automaticky vygenerovaný a obsahuje predvolenú konfiguráciu, ktorú je možné podľa potrieb meniť.
- `models.py` - Súbor slúži pre definovanie modelov. V Djangu pod pojmom *model* rozumieme hlavný zdroj o dátach. Každý model sa mapuje do práve jednej databázovej

tabuľky. Preto každý model obsahuje atribúty zvané polia, ktoré majú definovaný svoj typ. Modely vytvárajú abstrakciu, vďaka ktorej je možné vytvárať databázu priamo v Pythone. Obsahuje všetky potrebné funkcionality na vytvorenie databázy. Ďalšia výhoda abstrakcie modelov ako medzivrstva medzi programovým kódom aplikácie a databázy je možnosť migrácie na rôzne relačné databázy bez nutnosti meniť kód aplikácie nakoľko vytvorenie už špecifického kódu SQL pre konkrétny typ relačnej databázy je obsiahnutý v rámci vrstvy a jej použitie je pre vývojára transparentné.

- **views.py** - Obsahuje všetky grafické rozhrania, cez ktoré môže používateľ komunikovať s webovou aplikáciou. Všeobecne povedané, grafické rozhrania predstavujú všetky prvky, ktoré používateľ má možnosť vidieť, interpretujú dáta do vizuálnej formy. Môžu obsahovať aj interaktívne prvky, ktoré slúžia na to aby grafické rozhranie bolo dynamické a aby užívateľ mohol pomocou nich ovládať aplikáciu.
- **urls.py** - V tomto súbore sú umiestnené všetky URL adresy aplikácie. Na konkrétne URL adresy sú pripojené grafické rozhrania zo súboru *views.py* z rovnakej aplikácie, aby bolo možné určiť pri načítaní konkrétnej URL aké grafické rozhranie sa má zobraziť. Štruktúra súboru je rovnaká ako v prípade súboru *urls.py* v koreňovom adresári aplikácie.
- **tests.py** - Je to základný súbor určený pre programový kód, ktorý testuje správnosť funkcionality aplikácie. V prípade, že sa počet testov značne rozšíri a testujú sa viaceré moduly aplikácie, potom je odporúčané pre prehľadnosť tieto testy od seba oddeliť a rozdeliť ich do viacerých súborov podľa časti aplikácie, ktorý testujú.

5.3 Pandas

Pandas je voľne šíriteľná knižnica určená pre programovací jazyk Python, ktorá sa využíva pre spracovanie a analýzu dát. Jej využitie môžeme nájsť aj v rozhraní, ktoré ponúkajú služby pre získavanie dát akciových trhov, ktoré webová aplikácia využíva. Zhrniem viaceré výhody Pandasu a tým pádom aj dôvodový pre jeho využitie v aplikácií: [20]

- Optimalizovaný pomocou volaní knižníc programovacieho jazyka C.
- **Pandas.DataFrame** - Je to dátová štruktúra knižnice Pandas. Poskytuje vyššiu abstrakciu pri práci s dátami. Uchováva dáta v dvojrozmernom poli, pričom bunka má premenlivú veľkosť a všetky údaje v tabuľke sú heterogénne.

5.4 Bootstrap 4

Je to sada nástrojov s otvoreným kódom a voľným použitím⁶. Obsahuje šablóny v jazyku HTML a CSS pre typografiu, tabuľky, tlačidlá, formuláre a iné. Výrazne zjednodušuje tvorbu dynamických webových stránok a zjednocuje postup ako tvoriť takéto stránky. To znamená, že spôsoby akým vytvára obsah by mal byť podporovaný vo väčšine moderných webových prehliadačoch. Ide o populárnu sadu nástrojov, ktorá má rozsiahlu dokumentáciu⁷. Pri niektorých komponentoch využíva aj skriptovací jazyk Javascript.

⁶<https://getbootstrap.com/docs/4.0/about/license/>

⁷<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

V čase práce na tejto bakalárskej práci bola vydaný aktuálnejšia verzia - Bootstrap 5. Napriek mnohým vylepšeniam som sa rozhodol v práci použiť starší Bootstrap 4, keďže táto verzia už obsahuje viacero ďalších čiastkových verzií, ktoré opravovali mnohé chyby a preto predpokladám, že bude stabilnejšia⁸.

5.5 Datatables

Datatables je jQuery plugin s otvoreným kódom dostupný pod licenciou MIT⁹.

Tento plugin rozširuje funkcionality tabuliek značkovacieho jazyka HTML. Je plne kompatibilný s modernými webovými prehliadačmi. Popri základných HTML tabuliek ponúka ďalšiu funkcionality vo forme ich filtrovania, vyhľadávania v tabuľke, implementuje stránkovanie tabuľky a spracovanie dát na strane servera (to znamená jednoduchú dynamickú aktualizáciu tabuľky v kombinácii s jazykom Javascript a technológiou AJAX).

Projekt Datatables je aktívne vyvíjaný a obsahuje obsiahlu dokumentáciu¹⁰.

5.6 Datetimepicker

Je jQuery plugin, ktorý poskytuje možnosť grafického výberu dátumu a času. Tento plugin je požitý pri výbere dátumu a času v textových poliach. Zobrazuje sa pri kliknutí počítačovou myšou na dané textové pole.

Má otvorený zdrojový kód a je ho možné použiť pod licenciou MIT¹¹.

5.7 Nativefier

Je to nástroj pre príkazový riadok, ktorý pomáha zapuzdiť webovú aplikáciu a vytvorí z natívnu aplikáciu pre systém Windows. Tento nástroj pomôže splniť jeden cieľ zadania - vytvorenie Electron JS aplikácie, ktorá bude môcť byť spustiteľná u klienta.

Má otvorený zdrojový kód a je ho možné použiť pod licenciou MIT¹².

5.8 Highcharts

Je to knižnica napísaná výhradne v programovacom jazyku JavaScript, ktorá sa zameriava na grafickú vizualizáciu dát v podobe grafov. Pre nekomerčné/študijné účely je prístupná bez poplatkov. Ďalšie dôvody pre voľbu knižnice Highcharts:

- Knižnica je veľmi dobre zdokumentovaná. Má oficiálnu dokumentáciu rozhrania, ktorá obsahuje aj netriviálne ukážky použitia. Všeobecne je dobre štylizovaná a prehľadná¹³.
- Pre nekomerčné/študijné účely možné knižnicu využiť pod licenciou Creative Commons (CC) Attribution-Non-Commercial license¹⁴¹⁵.

⁸<https://blog.getbootstrap.com/2021/01/19/bootstrap-4.6.0/>

⁹<https://datatables.net/purchase/index>

¹⁰<https://www.datatables.net/manual/>

¹¹<https://github.com/xdan/datetimepicker>

¹²<https://github.com/nativefier/nativefier>

¹³<https://www.highcharts.com/docs/index>

¹⁴<https://shop.highsoft.com/faq/>

¹⁵<https://shop.highsoft.com/highsoft/form/noncommercialform>

- Je aktívne vyvíjaná, pridávanie nových funkcií a oprava chýb (v dobe písania tejto bakalárskej práce) je na mesačnej báze¹⁶.
- Disponuje veľkou škálou rôznych typov grafov. Produkt Highcharts je rozdelený do viacerých častí, rovnako tak jeho rozhranie. Aplikácia využíva časť knižnice Highcharts zaoberajúca sa finančnou oblasťou, ktorá má názov Highcharts Stock¹⁷.
- Highcharts Stock - Úzko špecializované rozhranie knižnice Highcharts poskytujúca podporu pre viaceré požiadavky aplikácie v oblasti spracovania a vizualizácie údajov primárne z finančného sektoru¹⁸.
- Grafy sú vstavanú možnosť interakcie s používateľom. Grafy sú dynamické, viaceré interaktívne prvky sú súčasťou knižnice a nie je nutné ich zložiť programovať, napríklad:
 - Možnosť graf priblížiť/oddialiť.
 - Možnosť zmeniť časový interval na horizontálnej osi a zobrazí tak dáta v inom časovom období. Časový interval je možné zadať manuálne (od, do) alebo zvoliť na panely základných intervalov (deň, mesiac, polrok, rok a podobne).
 - Graf reaguje na počítačovú myš, zobrazuje aktuálnu hodnotu a ďalšie podrobnosti o danom bode, kde sa počítačová myš nachádza.



Obrázek 5.1: Ukážka grafu vykresleného pomocou knižnice Highcharts. Zdroj: highcharts.com²⁰

¹⁶<https://www.highcharts.com/blog/changelog/>

¹⁷<https://www.highcharts.com/demo>

¹⁸<https://www.highcharts.com/blog/products/stock/>

²⁰<https://www.highcharts.com/demo/images/samples/stock/demo/compare/thumbnail.png>

- Vstavaný adaptívny dizajn. Graf prispôsobuje svoje elementy (nadpisy, štítky, legendy) podľa toho v akých rozmeroch je stránka, na ktorej je graf umiestnený, zobrazená. Má implementované inteligentné adaptovanie (elementy umiestni na nové najoptimálnejšie nové miesto v grafe).
- Umožňuje prídanie/upravenie vizuálneho formátovania pomocou kaskádových štýlov, pretože v implementácii knižnice Highcharts je funkcionality je oddelená od grafického dizajnu.

5.9 SQL

Je to vyhľadávací jazyk využívaný pre prácu s dátami v relačných databázach. Obsahuje niekoľko typov príkazov, taktiež záleží nad konkrétnym druhom jazyka SQL, avšak základné príkazy napríklad sú:

- Príkazy určené pre manipuláciu s dátami (SELECT, INSERT, UPDATE, DELETE).
- Príkazy určené k vykonaniu transakcií (START TRANSACTION, SAVE TRANSACTION, COMMIT, ROLLBACK).
- Príkazy určené k definíciám dát v databáze (CREATE, ALTER, DROP).

Aplikácia bude využívať dva typy databáz, PostgreSQL (produkčný server) a SQLite (lokálny server).

5.9.1 PostgreSQL

PostgreSQL je objektovo relačný databázový systém, aktívne vyvíjaný PostgreSQL Global Development Group²¹. Je možné ho slobodne využívať (pre komerčné i nekomerčné účely) a má otvorený zdrojový kód. Využíva model klient-server, preto pre fungovanie PostgreSQL je nutné aby bol nasadený na servery so sieťovým pripojením. Nasadenie servera nie je viazané na konkrétnu platformu, je multiplatformový, rozšírený na väčšine rozšírených platformách.

Medzi výhody je možné považovať jeho spoľahlivosť, plne implementuje ACID vlastnosti databázy. Je vhodný pre nasadenie veľkých databáz, ktoré sa môžu v budúcnosti zväčšovať. Obsahuje viacero mechanizmov zabezpečenie databázy. Plnohodnotne implementuje jazyk SQL, pričom sa snaží o rešpektovanie štandardu ANSI SQL²². Jazyk SQL zároveň vo svojej implementácii aj rozširuje. Je flexibilný čo v prípade rôznych dátových typov. Obsahuje pokročilý systém pre manažment databázy. Veľké možnosti poskytuje aj konfigurácia, ktorú je možné upraviť na mieru pre konkrétnu aplikáciu. Pre veľké rozšírenie a využívanie existuje veľa študijnej literatúry o PostgreSQL.

Rovnako, veľké možnosti pri konfigurácii patria aj k nevýhodám, keďže pre korektné počiatkové nastavenie databázy sú nutnejšie pokročilejšie vedomosti.[21]

5.9.2 SQLite

SQLite je relačný databázový systém distribuovaný ako knižnica v programovacom jazyku C. Knižnicu je možné voľne použiť, pretože je licencovaná ako voľné dielo. Má otvorený zdrojový kód a je stále v aktívnom vývoji (v dobe písania tejto bakalárskej práce), pričom

²¹<https://www.postgresql.org/community/contributors/>

²²<https://postgres.cz/wiki/PostgreSQL>

nové verzie sú spätne kompatibilné so staršími verziami. Podpora je rozšírená pre veľké množstvo programovacích jazykov, vrátane Pythonu. Využitie knižnice je možné na viacero platformách. SQLite patrí medzi významne používané relačné databázové systémy, najmä v oblasti mobilných telefónov alebo je častokrát súčasťou aplikácií pre počítače. [19]

SQLite má mnohé výhody. Celková veľkosť knižnice je niečo menej než 1 megabajt. SQLite beží ako súčasť aplikácie a nepotrebuje mať k dispozícii server. Databázu nie je nutné pracne konfigurovať, konfigurácia je jednoduchá pre základné použitie. Celá databáza je obsiahnutá v jednom súbore, preto je databáza ľahko prenositeľná.

Zároveň má pre svoju netypickú implementáciu databázy aj svoje nevýhody. Pre svoje jednoduché používanie a prakticky minimálne možnosti konfigurácie, má obmedzené vlastnosti v prípade prispôbitelnosti. Rovnako databázu nie je ľahké dodatočne zväčšovať. Niektoré príkazy SQL nie sú implementované prípadne ich implementácia je značne obmedzená oproti PostgreSQL. Absentuje vstavaný autentifikačný systém či iné bezpečnostné prvky databázy. Nie je vhodná ak návrh počíta s veľkou databázou, rovnako nie je možné väčších optimalizácií²³. [19]

²³<https://tableplus.com/blog/2018/08/sqlite-vs-postgresql-which-database-to-use-and-why.html>

Kapitola 6

Implementácia webovej aplikácie

Táto kapitola sa venuje implementácií aplikácie. Slovné popisuje mechanizmy, algoritmy a postupy, ktoré boli vykonané v rámci implementácie funkcionality. Implementácia čerpá hlavne z návrhu aplikácie. V niektorých prípadoch sa vyskytli problémy s ktorými návrh prvotne nepočítal, respektíve nekonkretizoval presný postup ako bude funkcionality fungovať. Tieto problémy sú v sekciách tejto kapitoly vysvetlené a ukazujú postup ako boli vyriešené, prípadne aké obmedzenia pre aplikáciu so sebou prinášajú a dôvody prečo som sa pre takéto riešenie rozhodol.

6.1 Štruktúra programu

Pred detailnejším popisom implementácie programu prikladám vizualizáciu štruktúry aplikácie, ktorá bude táto kapitola ďalej popisovať.

Obrázok 6.1. popisuje adresárovú štruktúru aplikácie. Obsah aplikácie sa nachádza v koreňovom adresári s názvom BPPProject, v ktorom sú umiestnené ďalšie adresáre.

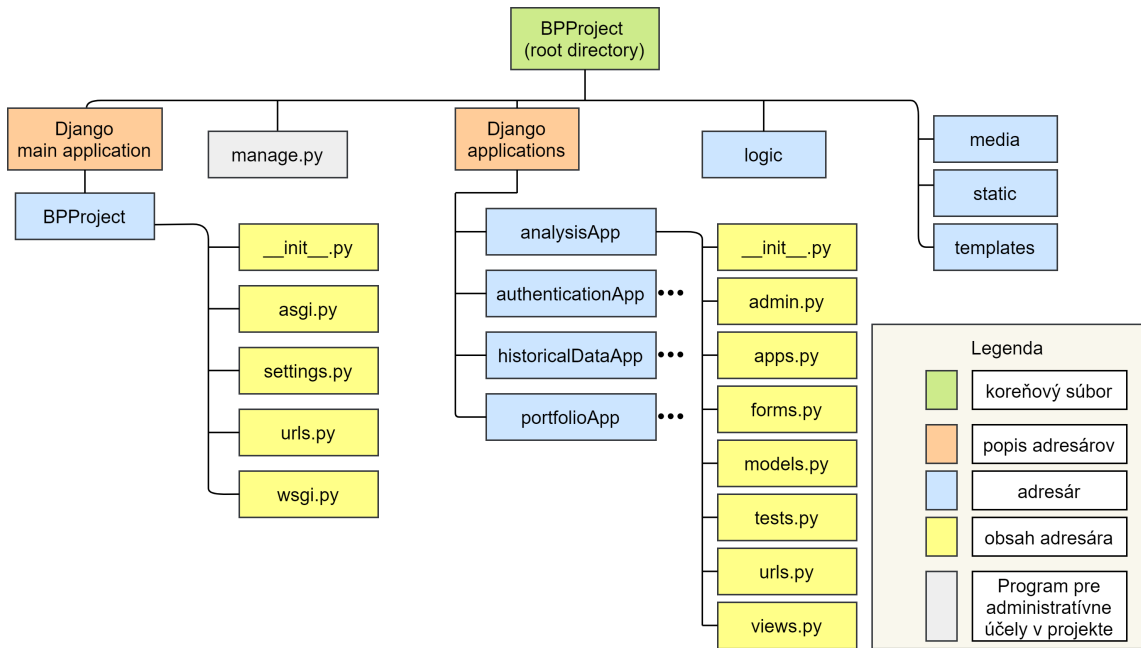
V obsahu koreňového adresára sa nachádza adresár, ktorý má rovnaký názov ako koreňový adresár. Predstavuje hlavnú Django aplikáciu projektu.

Ďalšou skupinou adresárov sú Django aplikácie a adresár logiky. Django aplikácie predstavujú hlavnú časť aplikácie a prepája vrstvu logiky s užívateľským prostredím. Adresár logiky obsahuje ďalšie adresáre, v ktorých sú umiestnené skripty jazyka Python pre špecifické funkcionality a operácie.

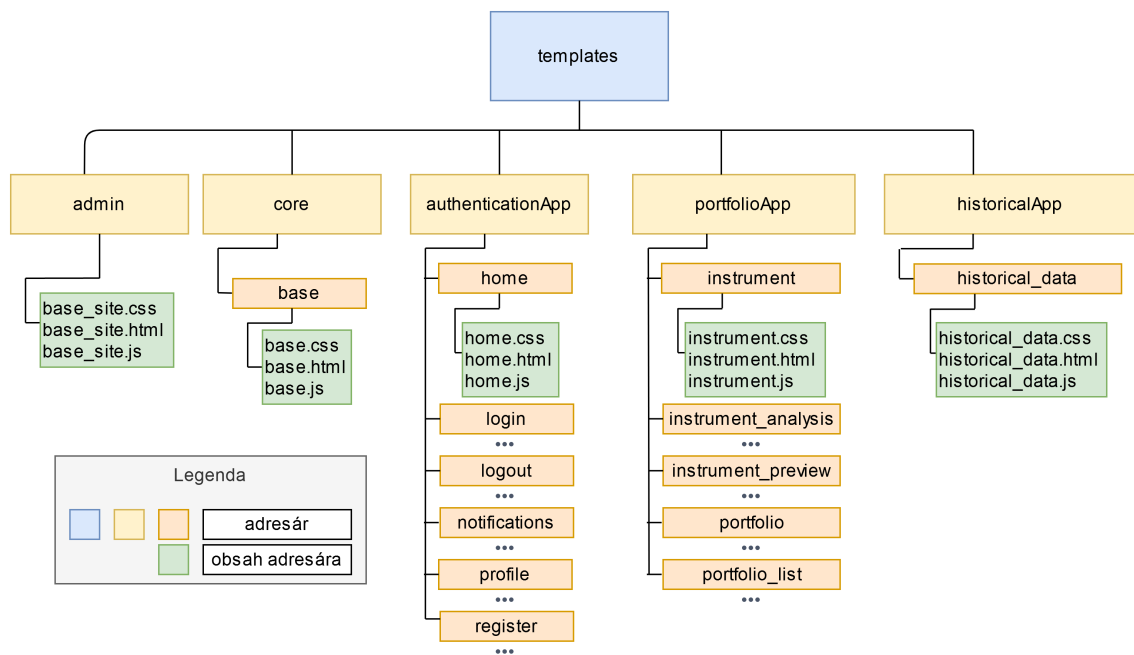
Súbor `manage.py` je skript určený k administratívnym účelom v projekte ako napríklad vytvorenie administrátora aplikácie.

Adresár `static` obsahuje statické súbory aplikácie a adresár `media` obsahuje profilové obrázky užívateľov.

Adresár šablón detailnejšie zobrazuje obrázok 6.2. Obsahuje hierarchicky uložené súbory pre jednotlivé stránky vytvorené v skriptovacom jazyku Javascript a značkovacom jazyku HTML doplnené o kaskádové štýly.



Obrázek 6.1: Vizualizácia štruktúry programu



Obrázek 6.2: Vizualizácia štruktúry adresára šablón

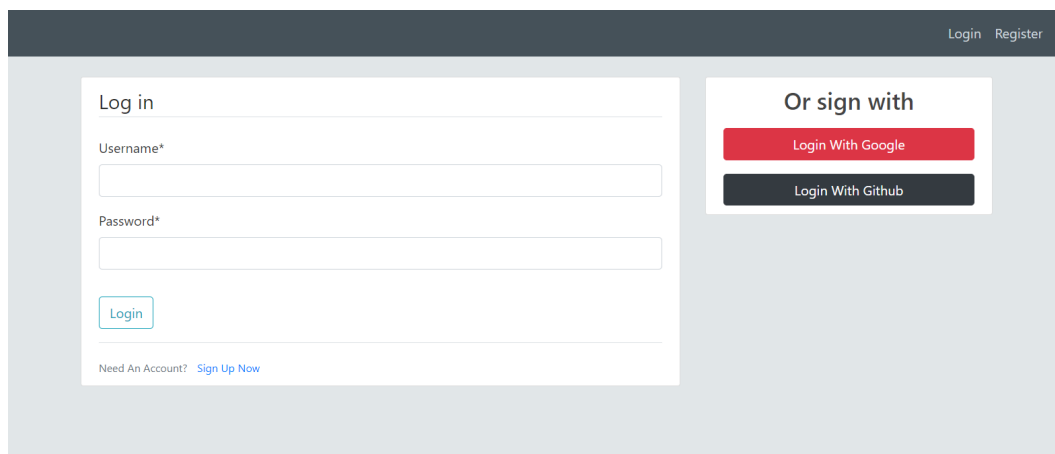
6.2 Implementácia užívateľského prostredia

Táto sekcia sa zameriava na implementáciu užívateľského prostredia. Popisuje rozdiely medzi návrhom a skutočnou implementáciou. Taktiež popisuje postupy ako užívateľské prostredie vykonáva svoju funkcionálnosť a spôsob ako užívateľské prostredie komunikuje so zvyškom aplikácie, ktorého princípy návrh neriešil.

6.2.1 Stránka prihlásenia/odhlásenia/registrácie

Stránky prihlásenia/odhlásenia a registrácie predstavujú pomyslenú bránu do aplikácie. Všetky z menovaných stránok obsahujú podobnú implementáciu. Základom stránok je formulár, v ktorom je možné vyplniť informácie. Pri registrácii sú jednoduché kontroly, ako napríklad dĺžka reťazcov, je vykonávaná na úrovni formulárov. Ak sú všetky potrebné údaje vyplnené a validácia formulára prebehla úspešne, potom je z údajov vytvorený záznam v databáze. Stránka prihlásenia poskytuje formulár pre prihlásenie. Ak sú zadané údaje zhodné s niektorým záznamom v databáze, potom je užívateľ overený a presmerovaný do tohoto profilu. Spôsob prihlásenia bol na základe odozvy jedného z užívateľov priebežného testovania počas implementácie rozšírený o autorizáciu pomocou protokolu OAuth 2.0.

Protokol OAuth 2.0 umožňuje užívateľovi prihlásenie sa do aplikácie pomocou prihlásenia sa do služby tretej strany. Táto služba služba figuruje ako medzivrstva medzi užívateľom a aplikáciou počas prihlásenia. Služba vygeneruje pre aplikáciu prihlasovací token, ktorý je použitý pre prihlásenie do aplikácie. Aplikácia tak nezískava prihlasovacie údaje užívateľa, ktorými sa prihlasuje do služby tretej strany¹. Takýto postup prihlásenia sa do aplikácie je výhodný pre užívateľa. Môže využívať jeden účet v rámci viacerých služieb a podobne. Do aplikácie boli zakomponované dve služby, ktoré podporujú prihlásenie pomocou OAuth 2.0 - Google² a Github³. K tomuto bola využitá knižnica django-allauth⁴. Dôvodom pre výber tejto knižnice je široká kompatibilita so službami tretích strán, ktoré podporujú prihlásenie cez OAuth 2.0⁵.



Obrázek 6.3: Implementácia stránky prihlásenia

¹<https://tools.ietf.org/html/rfc6749>

²<https://developers.google.com/identity/protocols/oauth2>

³<https://docs.github.com/en/developers/apps/authorizing-oauth-apps>

⁴<https://django-allauth.readthedocs.io/en/latest/installation.html>

⁵<https://django-allauth.readthedocs.io/en/latest/providers.html>

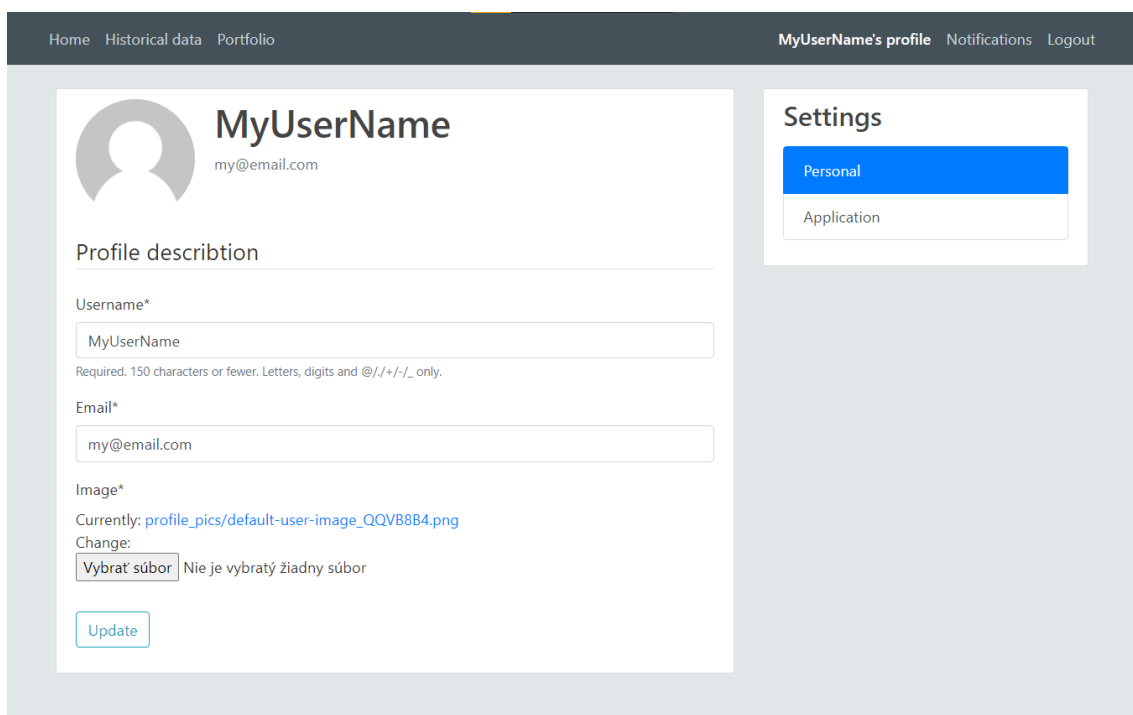
Stránka odhlásenia informuje užívateľa o úspešnom odhlásení. Aplikácia v tomto momente ukončí jeho reláciu.

6.2.2 Stránka profilu

Stránka profilu je implementovaná do dvoch kariet - nastavenia užívateľa a aplikácie. Obe karty obsahujú formulár, cez ktorý je možné zmeniť/doplniť údaje a poslať aplikácií na spracovanie/uloženie.

V karte užívateľa je možné zmeniť/doplniť potrebné údaje, napríklad v prípade ak užívateľ nemá vyplnenú emailovú adresu (prihlásenie pomocou protokolu OAuth 2.0) a chce využívať funkciu notifikácií.

V karte aplikácie si môže užívateľ vybrať, z ktorého zdroja bude aplikácie získavať údaje. Ak služba vyžaduje prihlasovacie meno a heslo pre prístup k údajom, musí ich zadať, inak ich bude aplikácia požadovať pri každej stránke, ktorá dané údaje vyžaduje.



Obrázek 6.4: Implementácia stránky profilu užívateľa

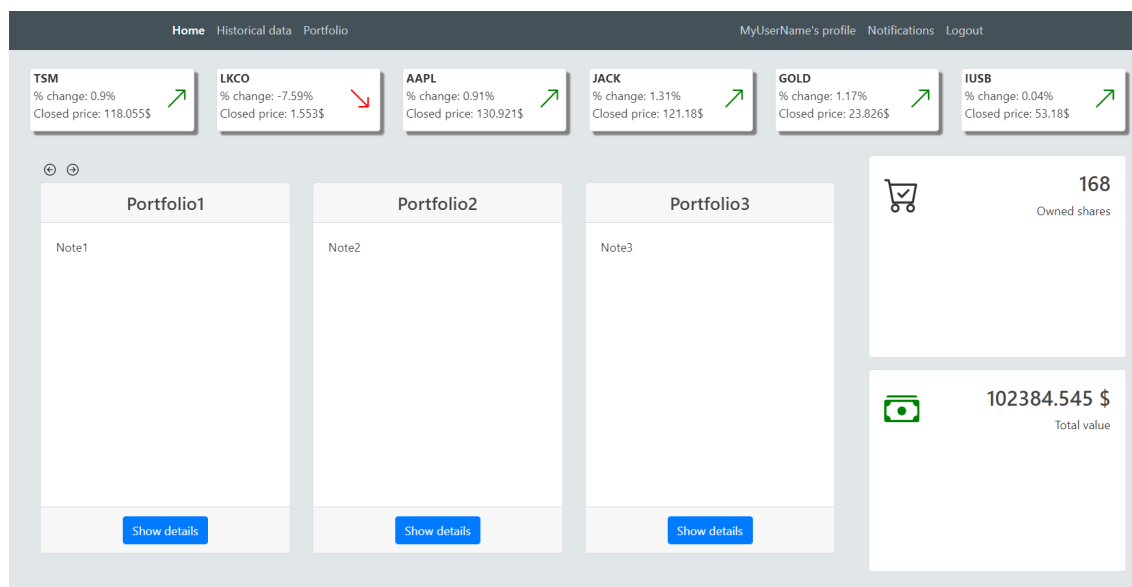
6.2.3 Domovská stránka

Návrh domovskej obrazovky slúžil ako základ pri jej implementácii. Tvoril jednoduchý pohľad na to ako poskytnúť užívateľovi prehľad o stave svojho účtu a svojich portfólií.

Návrh bol rozšírený o implementáciu čiarového grafu pri zobrazení detailu portfólia. Detail portfólia tak zobrazuje nielen najvýkonnejšie a najmenej výkonné investičné inštrumenty pre dané portfólio (pre aktuálny deň) ale zobrazuje tiež prostredníctvom grafu celkovú hodnotu portfólia. Užívateľ tak môže sledovať hodnotu portfólia od dátumu kedy bola kúpená prvá akcia v portfóliu.

Zároveň pribudol aktívny panel, ktorý je umiestnený na vrchole stránky (pod navigačnou lištou). Panel zobrazuje všetky inštrumenty (ktoré má užívateľ pridané v rámci celej

aplikácie) v tvare kariet. Každá karta zobrazuje symbol investičného inštrumentu a jeho nárast/pokles odvodený z uzatváracej ceny. Hodnota je udávaná v percentách. Vstupnými hodnotami výpočtu je uzatváracia cena posledných dvoch najbližších dní. Výsledkom výpočtu je ako sa percentuálne zmenila hodnota investičného inštrumentu minulého dňa voči predminulému. Keďže je výpočet generovaný z uzatváracej ceny, výsledok je stále aktuálny pre aktuálny deň (aktuálny deň vychádza z uzatváracej ceny predošlého dňa). Karty v aktívnom paneli sa aktualizujú v momente načítavania stránky. Výpočet tak neprebíha aktívne a nezávisle. Zvolil som takýto algoritmus kvôli tomu, že existuje iba jedna uzatváracia cena pre nejaký konkrétny deň a táto cena vznikne iba raz - pri uzatváraní burzy.



Obrázek 6.5: Implementácia domovskej stránky

6.2.4 Stránka listu portfólií a stránka portfólia

Obe stránky sa zaobišli bez výrazných zmien. Pribudli prvky, ktoré informujú užívateľa, že akcia, ktorú zvolil (napríklad pridanie inštrumentu) sa vykonáva. Je to z dôvodu, že pri implementácii sa ukázalo načítavanie dát investičných inštrumentov zo zdrojov a ich spracovanie ako viac sekundový proces. Zmena bola teda v doplnení informácie pre užívateľa, že na pozadí sa niečo vykonáva.

6.2.5 Stránka investičného inštrumentu

Táto stránka patrí medzi najkomplexnejšie v rámci aplikácie. Všetky popísané podmienky návrhu boli dosiahnuté dynamickým pohybom/reakciou komponentov na stránke.

Pred zobrazením stránky, server uloží do dočasnej pamäte dáta, ktoré boli zobrazené. Aplikácia tak optimalizuje získavanie/spracovanie dát využitím dočasnej pamäte. Z tohoto úložiska sú potom dáta využité pri predikcii vývoja ceny alebo pri presmerovaní na stránku historických dát.

Pri možnosti nákupu/predaja podielu v investičnom inštrumente som sa snažil implementovať dvojitú kontrolu integrity dát. Prvá je na úrovni užívateľského prostredia, ktoré je kontrolované jazykom Javascript.

Akýkoľvek dátum nemôže byť z budúcnosti a možnosť nákupu/predaja/zobrazenia cez víkend je rovnako znemožnená, nakoľko vtedy burzy nebývajú otvorené. Ak nákup/predaj je vykonaný v dni, kedy je sviatok, táto zmena sa prejaví až v najbližší deň v ktorý sa bude obchodovať. Z prieskumu pred tvorením návrhu som nezistil, ako získať presné a zadarmo dáta pre rôzne burzy tak, aby som vedel vyhodnotiť, počas ktorých dátumov sa obchoduje a počas ktorých nie (napríklad zoznam sviatkov, voľných dní okrem víkendu). Preto som použil logiku transakcií aká je v banke, ak užívateľ zadá platbu cez internetové bankovníctvo cez víkend, platba sa spracuje až v najbližší pracovný deň. V tomto prípade by aplikácia mala mať poverenie o fungujúcej burze vtedy, ak existuje v službe odkiaľ získava dáta pre daný deň záznam vývoja ceny. Ak daný záznam neexistuje (rozumie sa, že v získaných dátach daný dátum nie je prítomný), potom aplikácia predpokladá, že v danom dátume sa s investičným inštrumentom neobchodovalo. Rovnako to rieši problém v prípade, že investičný inštrument má pozastavené obchodovanie na burze.

Na úrovni programovacieho jazyka Python je riešená integrita pre nákup/predaj nasledovne. Táto vrstva vykonáva opäť kontrolu všetkých vstupných parametrov a prebieha rovnaká kontrola ako v jazyku Javascript pred odoslaním dát aplikácií prostredníctvom technológie AJAX. Následne sa skontrolujú či potrebné záznamy (názov portfólia/spoločnosti) existujú v tabuľkách databázy. Ak pred tým nebola z nejakého dôvodu narušená integrita databázy, potom by tieto kontroly mali dopadnúť vždy dobre. Ak ide o požiadavku nákupu, týmto je záznam zapísaný a operácia ukončená. Ak ide o požiadavku predaja, vykonáva sa ďalšia kontrola integrity. Nakoľko aplikácia podporuje predaj v minulosti (pre účely simulácie), je nutné z časového hľadiska zaistiť aby podiel nebol predaný skôr než bol nakúpený. Algoritmus je v tomto prípade nasledovný. Funkcia pre predaj podielu prijme požiadavku. Z požiadavky zistí v akom dátume chce užívateľ podiel predať. Následne zistí z databázy, počet (objem) podielu na účte v danom dni. Ak je počet podielu rovný alebo väčší než požiadavka na predaj, ďalším požiadavkom požiada databázu aby mu poskytla všetky záznamy od tohoto dátumu smerom do súčasnosti. Hoci zistený zostatok v danom dni môže byť rovný/väčší ako chce užívateľ predať, aplikácia musí zistiť či najbližší predaj smerom do súčasnosti tento zostatok neznižuje pod hranicu množstva, ktoré chce užívateľ predať. Aplikácia tak prechádza záznamy od bodu kedy chcel užívateľ predať svoj podiel smerom do súčasnosti. Ak ide o záznamy nákupu, pripočítava ich k počtu podielu. Ak narazí k prvému záznamu predaja, rozhodne či je možné požiadavku úspešne spracovať. Ak aj po nájdení prvého záznamu predaja a odčítania od počtu podielu je počet podielu rovný/väčší ako požadovaný podiel nového predaja, je požiadavka užívateľa úspešne spracovaná. Ak nie je, potom prechádzanie záznamov do súčasnosti pokračuje rovnakým spôsobom ďalej a algoritmus hľadá dátum v ktorom by už predaj mohol byť možný aby ho mohol oznámiť užívateľovi spolu s informáciou, že prvotná požiadavka predaja nie je uskutočniteľná. Limitom tohoto algoritmu je, že pri špecifickej sekvencii nákupov/predajov môže byť ponúknutý dátum pre predaj (myslí sa tým ponúknutý dátum v prípade neúspešného prvotného požiadavku od užívateľa) časovo neskôr. Je teda šanca, že predaj dopadne úspešne ak aj užívateľ bude ignorovať dátum ponúknutý aplikáciou a zvolí neskorší dátum, nakoľko algoritmus nehľadá úplne prvý možný dátum kedy by bol predaj možný.

6.2.6 Stránka analýzy inštrumentu

Spočiatku bola táto stránka implementovaná podľa návrhu. Počas testovania implementácie a využívania aplikácie sa ukázalo, že by bolo vhodné návrh rozšíriť.

Pod tabuľku pribudla séria grafov, ktorých os X je navzájom prepojená a teda spoločná. Grafy zobrazujú v rámci rovnakého časového intervalu osi X rôzne dáta. Názvy grafov a ich účel teraz popíšem.

Prvým grafom je základný čiarový graf zobrazujúci vývoj ceny investičného inštrumentu. Dôvodom jeho prítomnosti je aby užívateľ vedel porovnať a interpretovať si ostatné grafy s vývojom ceny investičného inštrumentu.

Druhým grafom je čiarový graf zobrazujúci veľkosť (počet) podielu, aký užívateľ vlastnil za daný časový interval. Užívateľ tak môže historicky sledovať ako nakupoval/predával svoj podiel a porovnať si ho napríklad s vývojom ceny investičného inštrumentu. Môže tak získať prehľad a ohodnotiť či boli jeho kroky správne.

Tretí graf je odvodený od prvého grafu a zobrazuje percentuálny nárast/pokles ceny inštrumentu voči predchádzajúcemu dňu. Užívateľ tak môže sledovať aké veľké boli percentuálne výkyvy v cene investičného inštrumentu. Môže to byť pre užívateľa vhodná pomôcka pri analýze a zisťovaní trendu.

Štvrtý graf zobrazuje hodnotu finančných prostriedkov, ktoré užívateľ do investičného inštrumentu pri kúpe vložil a koľko pri predaji získal. Získava tak prehľad o tom, či sa mu už investícia vrátila alebo nie, prípadne akú veľkú sumu je v mínuse/pluse. Pri porovnaní s druhým grafom, napríklad zistiť či je v mínuse kvôli tomu, že do inštrumentu práve investoval finančné prostriedky (druhý graf ukazuje, že vlastní nakúpený podiel) alebo ide o čistú stratu (druhý graf ukazuje, že nevlastní žiadny podiel).

Piaty graf je odvodený od prvého a druhého grafu a zobrazuje tak hodnotu nakúpeného podielu v danom časovom bode.

6.2.7 Stránka upozornení

S touto stránkou návrh aplikácie spočiatku nepočítal. Stránka upozornení bola pridaná ako reakcia na odozvu z priebežného testovania užívateľmi. Užívateľ môže na tejto stránke vidieť všetky upozornenia, ktoré vo svojom profile vytvoril vo forme tabuľky. Zároveň bola pridaná možnosť, aby upozornenia bolo možné zmazať.

6.3 Implementácia získavania a spracovania dát

Získavanie a spracovanie dát opisuje spôsob ako boli návrh získavania dát prevedený do implementácie aplikácie. Samotný prieskum počas návrhu naznačoval, že finančné dáta získané zdarma sú veľmi vzácne. Pri implementácii sa tieto zistenia značne prehĺbili a vytvorili tak jeden z hlavných problémov implementácie, ktorého riešenie teraz popíšem.

Implementácia splnila požiadavku návrhu na získavanie dát z viacerých zdrojov, pričom užívateľ si môže zdroj vybrať. V implementácii bol požiadavok poňatý globálne, čo znamená, že pri zvolení konkrétneho zdroja dát bude toto nastavenie platiť pre celú aplikáciu. Nakoľko skoro všetky stránky pristupujú k zdroju dát a získavajú dáta pre vykonanie svojich úloh, javila sa možnosť globálneho nastavenia zdroja ako jednoduchšia varianta oproti nastaveniu zdroja pre každú špecifickú úlohu.

Implementácia rozlišuje viac typov volaní pri získavaní dát. Typy volania sú rozlišované podľa vstupných parametrov funkcie, ktorá získavanie dát riadi. Výsledkom je možnosť získavať dáta pre rôzne časové intervaly, pri rôznom type dát (pre jeden deň alebo pre viac dní), so získaním noviniek a ostatných informácií o investičnom inštrumente alebo bez nich. V rámci každého volania je vykonaná špecifická úprava nad dátami tak, aby výstupné dáta bolo možné zobraziť užívateľovi bez dodatočného spracovania mimo vrstvu logiky.

6.3.1 Knižnica iexfinance

Pre implementáciu dátového zdroja služby IEX Cloud bola použitá knižnica *iexfinance*⁶ pre programovací jazyk Python. Poskytuje rozhranie cez ktoré je možné komunikovať so službou IEX Cloud. Služba IEX Cloud poskytuje všetky potrebné dáta okrem denných dát⁷. Tieto dáta a prístup k rozhraniu v tomto bode je limitovaný len pre platiace účty.

Pri implementácií a testovaní tohto zdroju dát bol využitý zdarma účet, ktorý služba IEX Cloud ponúka. Nakoľko počet kreditov, ktoré základný bezplatný účet ponúka je pri dlhšom testovaní nízky, bol tak vývoj a testovanie primárne na testovacom zdroji (pieskovisko / sandbox), ktorý je súčasťou bezplatného účtu. Negatívom tohto prístupu je, že dáta ktoré testovací zdroj poskytuje nie sú reálne. Zároveň pri každom novom zavolaní sú dáta odlišné, hoci parametre volania sú rovnaké. Túto nevýhodu testovacieho zdroja rieši použitie plateného účtu alebo bezplatného účtu s nízkym kreditovým objemom. Vzhľadom na tento problém, bola hlavná implementácia a testovanie aplikácie uskutočnené na druhom zdroji dát.

IEX Cloud poskytuje viacero možností ako k dátam v ich rozhraní pristupovať. Pre platiace účty poskytuje aj metódy, ktoré sú optimalizované pre časté požiadavky. Pri využívaní zdarma účtu teda nebolo možné dosiahnuť najoptimálnejšie získavanie dát zo služby. Aktuálna implementácia aplikácie získava dáta kombináciou viacerých volaní, čo nie je pri kreditovom systéme akým IEX Cloud funguje optimálne, avšak v rámci možností funkčne.

6.3.2 Knižnica yfinance

Implementácia zdroja dát Yahoo! Finance je riešená prostredníctvom knižnice *yfinance*⁸ pre programovací jazyk Python. Ponúka rozhranie cez ktoré je možné získanie dát zo služby Yahoo! Finance. Získané dáta sú uložené v štruktúre *Pandas DataFrame*.

Všetky potrebné dáta pre túto aplikáciu sú k dispozícii v tejto knižnici okrem novinek. Novinky preto zastrešuje druhá knižnica, ktorá je popísaná v nasledujúcej sekcii o knižnici *stocknews* v časti 6.2.3.

6.3.3 Knižnica stocknews

Aplikácia využíva knižnicu *stocknews*⁹ kvôli prístupu k novinkám pre investičné inštrumenty. Je využívaná len z časti a to pretože získanie dát novinek neprebíha priamo cez túto knižnicu. Knižnica na základe vstupného parametru, ktorým je symbol investičného inštrumentu dokáže poskytnúť údaj, na akom vzore URL adresy sa novinky nachádzajú. Tento vzor URL adresy je spoločný pre viaceré investičné inštrumenty a namiesto priamo zadaného symbolu sa v ňom nachádza reťazec *%s*, ktorý je nahradený symbolom investičného inštrumentu a dáta sú následne získané prostredníctvom tejto adresy s využitím knižnice jazyka Python - *requests*. Dôvod pre tento spôsob získania dát je, že knižnica je napojená na službu, ktorá funguje na podobnom princípe kreditov ako IEX Cloud. Dokumentácia knižnice ale popisuje, že je možné využiť práve takýto spôsob, ktorý je zdarma ak sú požadované len základné dáta o novinkách investičných inštrumentov.

⁶<https://pypi.org/project/iexfinance/>

⁷<https://iexcloud.io/docs/api/>

⁸<https://pypi.org/project/yfinance/>

⁹<https://pypi.org/project/stocknews/>

6.3.4 Stránka historických dát

Návrh je implementovaný pomocou pluginu Datatables. Tabuľka je aktualizovaná pomocou skriptovacieho jazyka Javascript (ďalej len jazyk Javascript).

Pristúp na stránku je implementovaný dvoma spôsobmi. Ak užívateľ pristúpi na stránku skrz navigačný panel, všetky parametre zobrazenia si musí vybrať manuálne. Ak pristúpi zo stránky investičného inštrumentu, potom sú tieto parametre vyplnené automaticky.

Automatické vyplnenie parametrov je implementované kombináciou GET parametrov požiadavky a jazyka Javascript. GET parametre k adrese URL vytvorí a priradí aplikácia pri spracovaní požiadavky. Zároveň server posiela dáta, ktoré sú uložené v dočasnej pamäti. Aplikácia tak optimalizuje čas a prostriedky aby nezískavala a nespracovávala dáta duplicitne.

6.4 Implementácia upozornení

Upozornenia sú v aplikácií implementované s využitím vytvorenia nového vlákna. Dôvod pre takýto spôsob implementácie je aby spracovanie/vyhodnocovanie upozornení bolo nezávislé na hlavnom vlákne aplikácie.

Upozornenia je nutné pri prvotnom spustení aplikácie manuálne zapnúť. Je to z dôvodu, že ak má byť aplikácia nasadená na produkčnom servery, môže vytváranie takto nového vlákna spôsobovať problémy (zistenie pri testovaní aplikácie). Prístup k spusteniu vlákna pre upozornenie má iba správca aplikácie na administrátorskej stránke. Po úspešnom prihlásení sa správcovi zobrazí v ľavom hornom rohu pod nadpisom stránky tlačidlo, ktorým je možné vlákno vytvoriť. Samotné vlákno je možné spustiť iba raz. Ak vlákno ešte nebolo vytvorené, potom je po stlačení tlačidla zobrazená informácia, že vlákno bolo spustené (zmení sa popis tlačidla). Ak už vlákno bolo spustené a tlačidlo bude stlačené opäť, zobrazí sa informácia že vlákno už existuje a nevykoná sa žiadna akcia.

Notifikácie sú vyhodnocované každú minútu. Algoritmus získa z databázy všetky nastavené notifikácie. Z databázových dát zistí, na ktoré investičné inštrumenty sa notifikácie vzťahujú. Následne získa pre každý investičný inštrument najnovšie finančné dáta aby mohol vyhodnotiť či má dôjsť k notifikácii alebo nie. V prípade, že má dôjsť k notifikácii užívateľa, potom algoritmus vygenerujú text, ktorý sa následne pošle užívateľovi emailom. Po odoslaní emailu je daný záznam pre notifikáciu z databázy odstránený.

Email notifikácie v predmete obsahuje symbol investičného inštrumentu, ktorého sa notifikácia týka s popisom, že nastala zmena ceny. Telo emailu informuje o práve aktuálnej cene investičného inštrumentu, o cene na ktorú bola notifikácia nastavená a predošlej cene, ktorú mal investičný inštrument v čase vytvorenia notifikácie užívateľom.

6.5 Implementácia predikcie

Spôsob riešenia problému predikcie burzových cien a ich implementácia boli inšpirované diplomovou prácou pána Jan Ivar Larsen-a¹⁰. Vďaka preštudovaniu práce som získal prehľad, ako k problému predikcie pristupovať.

Aplikácia podporuje dva spôsoby výpočtu predikcie - pomocou jednoduchšej lineárnej regresie a modelu ARIMA. Implementácia spôsobu používania predikcií reflektuje návrh predikcie v aplikácií v časti 4.4.3.

¹⁰<https://core.ac.uk/download/pdf/52104888.pdf>

6.5.1 Jednoduchá lineárna regresia

Jednoduchá lineárna regresia, je štatistická metóda, ktorá umožňuje študovať vzťahy medzi dvoma spojitémi premennými. [23]

K výberu tejto metódy som pristúpil, pretože investori a obchodníci, ktorí používajú grafy, rozoznávajú vzostupy a poklesy cien v závislosti od časového rámca. Pomocou lineárnej regresie tak môže obchodník identifikovať kľúčové cenové body. [15]

Implementácia tejto metódy v aplikácií je s využitím knižnice scikit-learn¹¹. Scikit-learn knižnica jazyka Python zameraná na strojové učenie¹².

6.5.2 Model Auto ARIMA

Použitiu modelu ARIMA pri predikcii vývoja cien na burzách sa zaoberali v článku¹³ a publikácií [1], odkiaľ som aj čerpal ďalšie informácie.

Pri správne nakonfigurovaných parametroch môže byť výstup z modelu ARIMA použiteľný v rámci predikcie burzových cien. Nakoľko tieto parametre boli konfigurované často manuálne podľa analýzy priebehu dát, rozhodol som sa využiť implementáciu Auto ARIMA z knižnice pmdarima¹⁴. Knižnica pmdarima je štatistická knižnica pre jazyk Python, ktorá implementuje funkcionality modelu Auto ARIMA z programovacieho jazyka R. Výhodou implementácie Auto ARIMA voči modelu ARIMA je, že algoritmus sa snaží sám posúdiť charakteristiku dát a automaticky ich doplní vhodnými parametrami.

6.5.3 Porovnanie výstupov

Oba spôsoby predikcie boli po implementácii porovnané. Cieľom bolo zistiť, ktorý spôsob výpočtu je vhodnejší a ako predikciu celkovo využívať.

Pre účel porovnania bola zvolená jednoduchá metodika. Bolo zvolených 10 časových intervalov v investičnom nástroji spoločnosti Tesla Inc. (TSLA). Dôvodom pre testovanie na investičnom nástroji Tesla Inc. (TSLA) je jej zo začiatku stabilný a potom dynamický historický vývoj ceny. Časové intervaly na ktorých sa predikcia testovala boli mnou vybrané tak, aby mali od seba odlišný priebeh vývoja cien a trendu. To zahŕňa priebehy cien ako veľmi strmý nárast/pokles ceny, stabilizovaná cena a ich rôzne kombinácie. Dĺžka časových intervalov bola rozmanitá a pokusy zahŕňali aj zúženie/rozšírenie časového intervalu pričom bolo pozorované aký má táto zmena dopad na predikciu. Všetky časové intervaly boli vybrané z minulosti a to preto, aby bolo možné zobrazit vývoj ceny bližšie k súčasnosti a overiť tak výstup predikcie na tom aký vyvoj ceny skutočne nastal.

Interpretácia výsledkov porovnania je zložitá. Pri porovnávaní sa oba postupy mnohokrát zhodli na podobnej predikcii. Úplne presná predikcia ceny nikdy nenastala, avšak výstupy predikcií dokázali určiť budúci trend (stúpanie ceny/klesanie ceny) vo väčšine prípadov. Ideálnym spôsobom ako používať predikciu v rámci aplikácie sa javí spôsob vykonať oba výpočty a porovnať výstupy. Ak sa obe predikcie zhodujú, potom je pravdepodobné, že cena sa tak bude aj v skutočnosti vyvíjať. Rovnako je odporúčané vykonať predikciu na rôzne veľkých časových intervaloch a porovnať ich výstupy predikcie. Ak sa jednotlivé výsledky veľmi líšia, potom pravdepodobnosť reálneho vývoja ceny je viac neistá. Avšak

¹¹https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

¹²<https://github.com/scikit-learn/scikit-learn>

¹³<https://ijssst.info/Vol-15/No-4/data/4923a105.pdf>

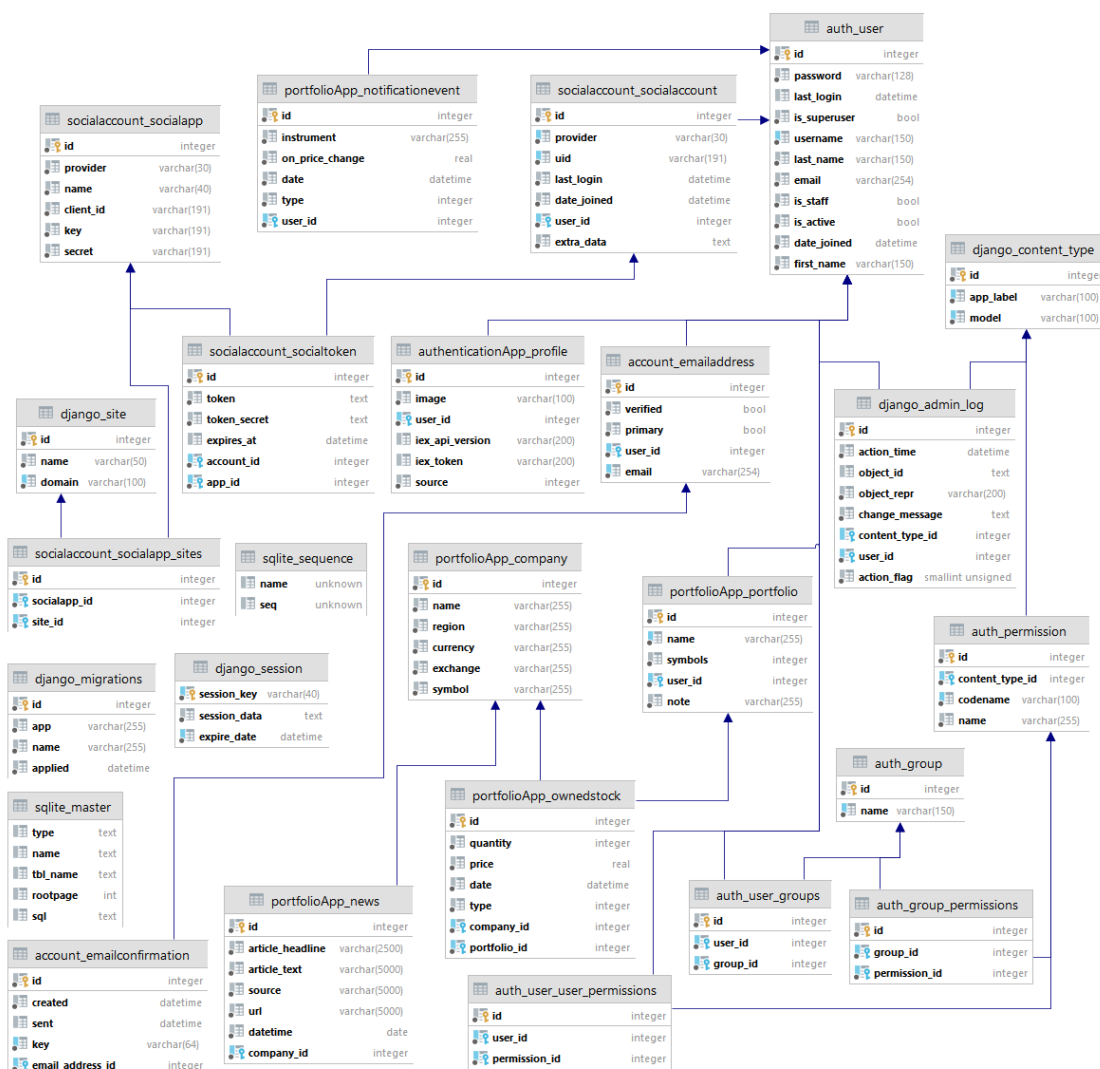
¹⁴<https://pypi.org/project/pmdarima/>

nakolko na cenu investičného inštrumentu má vplyv veľké množstvo faktorov, potom je takáto predikcia vývoja ceny iba orientačná.

6.6 Databázová štruktúra aplikácie

Pre komunikáciu s databázou aplikácie bola využitá technológia Django modelov. Každý model spravidla reflektuje jednu databázovú tabuľku. Model predstavuje triedu jazyka Python, ktorá je podtriedou `django.db.models.Model`. Každý atribút modelu predstavuje databázové pole. Technológia Django poskytuje automaticky generované API pre prístup do databázy¹⁵.

Základný návrh databázy je zhodný s entitno-relačným diagramom v návrhu sekcií 4.2 a implementovaný v súboroch `models.py` v jednotlivých aplikáciách. Vplyvom doplnenia ďalších funkcionalít s využitím knižníc sa zväčšila aj celková databázová štruktúra aplikácie, ktorá je na obrázku 6.6.



Obrázek 6.6: Úplná databázová štruktúra aplikácie.

¹⁵<https://docs.djangoproject.com/en/3.2/topics/db/models/>

Kapitola 7

Testovanie webovej aplikácie

Táto kapitola popisuje spôsoby a prístupy k testovaniu aplikácie. Úlohami pri testovaní boli nájsť chyby v aplikácii a vytváranie spätnej odozvy o aplikácii ako takej z pohľadu užívateľa. Cieľom testovania teda bolo nie len oprava chýb aplikácie ale aj rozšírenie/úprava návrhu o ďalšie vlastnosti či komponenty.

7.1 Typy testovaní

Testovanie aplikácie prebiehalo vo viacerých úrovniach.

Základné testovanie funkcionality aplikácie prebiehalo v rámci procesu implementácie. Výsledkom bolo otestovanie komponent, ktorých funkcionality je oddelená od iných komponent aplikácie a je možné testovať izolovane.

Postupom času ako začala aplikácia tvoriť väčší celok, boli vykonávané testovania na úrovni týchto celkov. Výsledkom bolo odhalenie komplexnejších chýb či nedostatkov implementácie a ich následné riešenie. V tejto súvislosti prichádzali prvé pripomienky od testovacích užívateľov aplikácie, ktoré boli po diskusiách s nimi zakomponované do riešenia.

Je nutné poznamenať, že oba predchádzajúce spôsoby testovania sa vykonávali pri aktívnej implementácii a teda v čase keď boli pridávané do aplikácie nové funkcionality. Ako boli časom funkcionality riešené a občas upravované, vznikali rôzne chyby, ktoré boli výsledkom prepojenia viacerých modulov do väčších celkov. Preto boli chyby často odhalené aj pri dobre fungujúcich celkoch, nakoľko sa mohla prejaviť chyba z iného celku.

Posledná fáza testovania prebiehala po dokončení implementačnej fázy aplikácie. Do projektu neboli pridávané už žiadne nové funkcionality a zväčšil sa priestor pre testovanie, ktoré malo za cieľ už len odhaliť chyby.

7.2 Proces testovania

Základné testovanie bolo vykonávané na lokálnom serveri. Približne v strede implementačnej fázy, keď aplikácia začala tvoriť väčší celok, bola nasadená na produkčný server.

V kombinácii s nasadením na produkčný server bola odhalená rada problémov, ktoré sa pri lokálnom testovaní neobjavili. Nasadenie na produkčný server pomohlo zvýšiť celkové testovanie. Na testovaní sa v tejto fáze začali zúčastňovať v rámci časových možností ďalší dvaja užívatelia, ktorí informovali o chybách no hlavne vytvárali spätnú odozvu pri práci s aplikáciou. Postupy testovania sa menili postupom implementácie, nakoľko testovanie overovalo správnosť výsledkov implementácie.

Testovanie prebiehalo vykonávaním štandardných úkonov v rámci aplikácie. Užívatelia mali za úlohu vytvoriť si profil, vytvoriť si portfólia a vykonávať operácie s inštrumentami. Zároveň sa snažili nájsť logické chyby v analýzach, ktoré aplikácia vykonáva. Takto bolo odhalených mnoho chýb kedy algoritmus obsahoval zlý výpočet a tým pádom bola interpretácia dát chybná.

Ďalším postupom bolo otestovať čo najviac kombinácií rôznych úkonov užívateľa. Mnohokrát implementácia nepočítala s niektorými scenármi, napríklad so zadaním URL s existujúcim symbolom investičného inštrumentu ale neexistujúcim portfóliom. Boli preto do aplikácie implementované rôzne validácie aby bolo podobným úkonom zabránené.

Kapitola 8

Záver

Výsledkom bakalárskej práce je webová aplikácia, ktorej úlohou je umožniť užívateľovi spravovať svoje virtuálne portfólia a sprostredkovať výstupy analýz, ktoré popisujú historické správanie užívateľa na burzovom trhu s využitím reálnych burzových dát. Počas vypracovania bakalárskej práce som sa zoznámil na jednoduchej úrovni s finančným trhom a princípmi obchodovania na burze. Vykonal som prieskum existujúcich riešení s podobným zámerom ako má táto bakalárska práca. Vybral som si niekoľko existujúcich riešení, ktoré som si vyskúšal a skúsenosti s nimi si poznamenal. Následne som skúsenosti spätne porovnal a analyzoval, čoho výstupom je kapitola venujúca sa existujúcim riešeniam, ktorá stručne popisuje hlavné zistenia. Poznatky z analýzy boli následne využité pri tvorbe návrhu aplikácie. Pri návrhu aplikácie som dopredu predpokladal, že sa môžu vyskytnúť problémy, nakoľko som sa v čase vytvárania návrhu ešte len zoznamoval s princípmi vývoja webových aplikácií. Preto hlavným cieľom návrhu bolo definovať základné funkcionality aplikácie a vytvoriť tak pomyselné základne piliere, ktoré boli postupom času a získavania znalostí rozširované o ďalšie funkcionality. Výsledkom je kapitola návrhu aplikácie a Use-Case diagram.

Popri tom som si viac prehlboval znalosti programovacieho jazyka Python a zoznamoval sa s technológiou Django. Následne som analyzoval možné spôsoby ako získať reálne dáta z burzy. Potom som sa zoznámil s štruktúrou dát a hľadal spôsoby ako tieto dáta využiť pri vizualizácii a analýze správania užívateľa na burzovom trhu. Po zoznámení sa s dátami, bol v rámci návrhu doplnený entitno-relačný diagram, ktorý popisoval štruktúru databázy aplikácie.

Pred implementáciou boli presne zadefinované použité technológie. Výberu technológií prechádzal prieskum a menšie praktické zoznámenie sa s možnosťami. Následne som vybral konkrétne technológie, ktoré som využíval v implementácii aplikácie.

Všetky spomenuté body v závere boli pre mňa prínosom. Mal som možnosť naučiť sa nové technológie a vytvoriť webovú aplikáciu, ktorá je prepojená s dátami z reálneho sveta. Získal som tak nové skúsenosti pri spracovávaní väčších dát z reálneho sveta a pri používaní nových technológií. Veľmi kladne tiež hodnotím získane skúsenosti pri využití informačných technológií v inom odbore. Odbor financií a obchodovania na burze bol pre mňa zvlášť zaujímavým prínosom.

V priebehu vypracovania bakalárskej práce som prišiel na niekoľko možností, akými by mohol ďalší vývoj pokračovať. V prípade väčšieho využívania platených služieb pre získanie burzových dát, vidím priestor pre väčšiu optimalizáciu aplikácie. Rozšírenie funkcionality ukladania dát o permanentné ukládanie často vyhľadávaných investičných inštrumentov a to z dôvodu šetrenia poplatkov za získavanie nových dát. Ďalším bodom by mohla byť optimalizácia získavania dát cez metódy, ktoré sú dostupné len pre platené účty a prípadná

správa a kontrola už minútých kreditov v rámci služieb. Funkcionalita predikcie budúceho vývoja ceny investičného inštrumentu môže byť tiež zdokonalená. Rovnako pridanie automatickej analýzy, ktorá by vedela užívateľa informovať či nakúpiť/predať podiel s pomocou analýzy výstupu z existujúcich riešení pre rozhodovanie sa na burzovom trhu.

Literatura

- [1] ADEBAYO, F., R, S. a SHANGODOYIN, K. Forecasting Stock Market Series with ARIMA Model. *Statistical and Econometric Methods*. Srpen 2014, sv. 3, s. 65 – 77.
- [2] CHEN, J. *Instrument Definition* [online]. Investopedia, srpen 2019 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/i/instrument.asp>.
- [3] CHEN, J. *Risk* [online]. Investopedia, říjen 2020 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/r/risk.asp>.
- [4] CHEN, J. *Index* [online]. Investopedia, březen 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/i/index.asp>.
- [5] CHEN, J. *Understanding Investors* [online]. Investopedia, únor 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/i/investor.asp>.
- [6] DJANGOPROJECT. *The Django admin site* [online]. Django Software Foundation, 2021 [cit. 2021-05-02]. Dostupné z: <https://docs.djangoproject.com/en/3.2/ref/contrib/admin/>.
- [7] DJANGOPROJECT. *Documentation* [online]. Django Software Foundation, 2021 [cit. 2021-05-02]. Dostupné z: <https://docs.djangoproject.com/en/3.2/misc/design-philosophies/>.
- [8] DJANGOPROJECT. *Security in Django* [online]. Django Software Foundation, 2021 [cit. 2021-05-02]. Dostupné z: <https://docs.djangoproject.com/en/3.2/topics/security/>.
- [9] DOLAN, B. *Investment Time Horizon Definition* [online]. Investopedia, únor 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/t/timehorizon.asp>.
- [10] HAYES, A. *Financial Markets* [online]. Investopedia, únor 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/f/financial-market.asp#types-of-financial-markets>.
- [11] HAYES, A. *Investment Definition* [online]. Investopedia, únor 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/i/investment.asp>.
- [12] HAYES, A. *Stock* [online]. Investopedia, duben 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/s/stock.asp>.
- [13] HAYES, A. *Technical Analysis* [online]. Investopedia, duben 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/t/technicalanalysis.asp>.

- [14] KENTON, W. *The Ins and Outs of Financial Instruments* [online]. Investopedia, březen 2020 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/f/financialinstrument.asp>.
- [15] NORRIS, E. *The Linear Regression of Time and Price* [online]. Leden 2020 [cit. 2021-05-08]. Dostupné z: <https://www.investopedia.com/articles/trading/09/linear-regression-time-price.asp>.
- [16] SEGAL, T. *Fundamental Analysis* [online]. Investopedia, březen 2021 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/terms/f/fundamentalanalysis.asp>.
- [17] STAFF, I. *Active vs. Passive Investing: Whats Best for You?* [online]. Investopedia, září 2020 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/news/active-vs-passive-investing/>.
- [18] TAYLOR, B. *Investment Strategies To Learn Before Trading* [online]. Investopedia, leden 2020 [cit. 2021-05-02]. Dostupné z: <https://www.investopedia.com/investing/investing-strategies/>.
- [19] TEAM, S. D. *About SQLite* [online]. 2021 [cit. 2021-05-02]. Dostupné z: <https://www.sqlite.org/about.html>.
- [20] TEAM, T. pandas development. *About pandas* [online]. Zenodo, duben 2021 [cit. 2021-05-02]. Dostupné z: <https://pandas.pydata.org/about/index.html>.
- [21] TEAM, T. P. development. *About PostgreSQL* [online]. The PostgreSQL Global Development Group, 2021 [cit. 2021-05-02]. Dostupné z: <https://www.postgresql.org/about/>.
- [22] TEAM, T. P. development. *What is Python? Executive Summary.* [online]. Python Software Foundation, 2021 [cit. 2021-05-02]. Dostupné z: <https://www.python.org/doc/essays/blurb/>.
- [23] UNIVERSITY, T. P. S. *2.1 - What is Simple Linear Regression?* [online]. 2018 [cit. 2021-05-08]. Dostupné z: <https://online.stat.psu.edu/stat462/node/91/>.
- [24] WEM.SK. *7 otázok, ktoré si položte predtým, než investujete do čohokoľvek* [online]. Říjen 2020 [cit. 2021-05-02]. Dostupné z: <https://wem.sk/7-otazok-ktore-si-polozte-predtym-nez-investujete-do-cohokolvek/>.

Příloha A

Obsah priloženého paměťového média

- BPPProject - Zdrojové kódy aplikace bakalářské práce.
- BPPProject-requirements.zip - Zoznam závislosti potrebných pre spustenie BPPProject.
- BPPProject-offline-packages.zip - Balíčky závislosti pre inštaláciu bez internetového pripojenia.
- BPPProject-requirements-manual.pdf - Návod na inštaláciu balíčkov.
- BPPProject-thesis.zip - Zdrojové kódy textu bakalářské práce.
- BPPProject.pdf - Text bakalářské práce v súbrovom formáte PDF.