FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

# Zadání bakalářské práce

| | |
|---|---|
| **Název:** | Obchodní platforma zasílající upozornění |
| **Student:** | Peter Kolde |
| **Vedoucí:** | Ing. Jan Trdlička, Ph.D. |
| **Studijní program:** | Informatika |
| **Obor / specializace:** | Webové a softwarové inženýrství, zaměření Softwarové inženýrství |
| **Katedra:** | Katedra softwarového inženýrství |
| **Platnost zadání:** | do konce letního semestru 2022/2023 |

## Pokyny pro vypracování

1) Zjistěte jakým způsobem lze získat data ze světových burz (informace o akciích, opcích, kryptu, ...) a porovnejte jednotlivé možnosti (např. z hlediska API, ceny za data, kvality dat, ...).
2) Porovnejte různé způsoby zasílání upozornění na mobilní/desktopová zařízení jednotlivých uživatelů (např. z hlediska API, rychlosti, spolehlivosti, ...).
3) Zjistěte, kteří brokeři umožňují zadávat obchodní příkazy z uživatelské aplikace, a porovnejte jejich řešení.
4) Zvolte vhodný zdroj dat a brokera. Navrhněte pro ně aplikaci, která bude
   a) analyzovat získaná obchodní data o daném instrumentu,
   b) při splnění zadaných pravidel (např. průrazu S/R úrovně, ...) zašle uživateli upozornění,
   c) připraví vhodný obchodní příkaz podle předchozího nastavení, který bude uživatel moci modifikovat/odeslat k příslušnému brokerovi.
5) Naimplementujte a otestujte aplikaci.

---

*Elektronicky schválil/a Ing. Michal Valenta, Ph.D. dne 8. října 2021 v Praze.*

Bakalárska práca

# OBCHODNÁ PLATFORMA POSIELAJÚCA UPOZORNENIA

**Peter Kolde**

Fakulta informačních technologií
Katedra teoretické informatiky
Vedúci: Ing.Jan Trdlička,Ph.D.
5. januára 2023

# Obsah

# Zoznam obrázkov



# Zoznam tabuliek

# Zoznam výpisov kódu

*Rád by som sa poďakoval pánovi Ing. Jan Trdlička, Ph.D. za pomoc pri vedení baklárskej práce.*

# Vyhlásenie

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 5. januára 2023 .....

# Abstrakt

Táto bakalárska práca si kladie za cieľ analyzovať dostupné elementy potrebné pre implementáciu obchodnej aplikácie zasielajúcej upozornenia. Práca sa skladá z dvoch častí. Prvú časť tvorí analýza dostupných prostriedkov na zber trhových dát, porovnanie možných spôsobov zasielania upozornení a prieskum brokerov, ktorí umožnujú zadávať platobné príkazy prostredníctvom API. V rámci tejto analytickej časti sú dostupné riešenia porovnávané. Druhú časť práce tvorí praktická časť, kde je následne za pomoci poznatkov získaných v analytickej časti práce implementovaná výsledná webová aplikácia a taktiež zachycuje popis technológií zvolených pri implementácii aplikácie.

**Klíčová slova** kapitálové trhy, webová aplikácia, trhové upozornenia, kvantitatívna analýza trhových dát, Alpaca, Alpha Vantage, C#, React

# Abstract

This bachelors thesis sets its goal the analysis of available elements required for implementation of a trading platform sending alerts to its users. Thesis is composed of two part. First part is made up of analysis of available resources for collection of market data, comparison of available methods for sending alerts and research of brokers, that enable sending market orders via API. Included in analytical part are also comparisons of available solutions. Second part is composed of practical part, where results gained from analytical part are transformed into implementation of resulting web application and also description of technologies used during implemenation of resulting application.

**Keywords** capital markets, web aplikation, market alerts, market data quantitative analysis, Alpaca, Alpha Vantage, C#, React

# Zoznam skratiek

| | |
|---|---|
| API | Application Programming Interface |
| REST | Representational State Transfer |
| SMTP | Simple Mail Transfer Protocol |
| IM | Instant Messaging |
| AWS | Amazon Web Services |
| OOP | Objektovo orientované programovanie |
| SDK | Software development kit |
| IoT | Intenet of things |
| MVC | Model-view-contoller |
| DB | Databáza |
| DTO | Data Transfer Object |

# Úvod

Obchodovanie na kapitálových trhoch je v súčasnosti pre bežného človeka prístupnejšie ako kedykoľvek v minulosti. V nedávnom období sa relatívne fundamentálne zmenili podmienky prístupu k obchodovaniu na kapitálových trhoch. Táto zmena nastala v závilosti na viacerých faktoroch, ktorými sú napríklad vznik a popularizácia rôznych platforiem určených pre neinštitucionálnych obchodníkov, znižovanie poplatkov spojených s vykonávaním transakcií prostredníctvom brokerov, atď.

Spomínané zmeny síce sprístupnili obchodovanie na kapitálových trhoch väčšiemu množstvu záujemcov, neuľahčili však naplnenie primárneho cieľa pre väčšinu obchodníkov, ktorým je zaujatie ziskových pozícií. V literatúre, ktorá sa zaoberá problematikou obchodovania na kapitálových trhoch sa zväčša uvádza podieľ 90% nových obchodníkov, ktorí obchodovaním stratia časť alebo všetky svoje investície. Tento vysoký podieľ prerábajúcich nováčikov je založený na viacerých komplexných faktoroch, medzi ktoré patria napríklad neopatrné používanie finančných pák, nakupovanie na základe émocií miesto skutočností, alebo neznalosť rozhodovacích procesov založených na vývoji jednotlivých finančných inštrumentov.

Práve na posledné dva body sa práca zameriava vo svojom obsahu, t. j. uľahčiť prístup pre nových obchodníkov prostredníctvom eliminácie časti rozhodovacieho procesu pri obchodovaní. Užívateľ si bude prostredníctvom aplikácie schopný navoliť metódy sledovania vývoja zvolených inštrumentov, na základe ktorých mu budú zasielané upozornenia s navrhovanou pozíciou. Tieto upozornenia budú vyhodnocované samotným užívateľom, ktorý sa môže rozhodnúť prijať, upraviť alebo odmietnuť navrhovanú pozíciu.

Vo výsledku sa výrazne zníži miera obchodov založených na emóciach, miera obchodov neopodstatnených reálnym vývojom jednotlivých inštrumentov a užívateľ sa taktiež odbremení od pracného monitorovania vývoja kapitálových trhov. Uvedené uľahčenia obchodovania nezaručujú úspech, napomáhajú však vo výraznej miere eliminovať jedny z najdôležitejších nepriaznivých faktorov pre nováčikov a v prípade správneho použitia by mali predísť strate všetkých investícií. Prínos práce v podobe ušetreného času je však možný aj u skúsených obchodníkov, ktorí sa vo svojom rozhodovaní opierajú o istú formu kvantitatívnej analýzy totožnou s výsledkom práce.

Osobnou motiváciou autora práce je dlhodobý záujem o finančný sektor.

Prvá časť práce sa zaorebá analýzou. Obsahuje prieskum existujúcich riešení a analýzu stavebných prvkov aplikácie, ktoré bude potreba pre finálnu realizáciu implementácie. Týmito stavebnými prvkami sú zber trhových dát, zasielanie upozornení, dostupní brokeri a spracovanie získaných dát.

Druhá časť pozostáva z návrhu jednotlivých častí, implementácie založenej na analýze a návrhu a v závere testovaním výslednej aplikácie.

# Cieľ práce

Práca sa vo svojom obsahu snaží o dosiahnutie nasledujúcich cieľov:

1. Preskúmať, akým spôsobom možno získať dáta zo svetových búrz v podobe informácií o akciách, opciách, kryptomenách a iných finančných inštrumentoch. Porovnať dostupné riešenia na základe relevatných ukazovateľov, primárne z hľadiska kvality, použiteľnosťi, dokumentácie, ceny za dáta a kvality získavaných dát.

2. Provonať rôzne spôsoby zasielania upozonení na mobilné/desktopové zariadenia jednotlivých užívateľov. Porovnavané sú primárne dva atribúty a tými sú rýchlosť doručenia upozornenia a spolahľivosť doručenia upozornenia.

3. Zmapovať, ktorí brokeri umožňujú zadávať obchodné príkazy z uživatelskej aplikácie a porovnať ich riešenia. Provnávané sú primárne kvalita, spoľahlivosť a cena za využívanie služieb.

4. Na základe poznatkov z predchádzajúcich bodov zvoliť vhodný zdroj dát, brokera a spôsob zasielania správ. S využitím vybraných elementov navrhnúť užívateľskú aplikáciu, ktorá bude vykonávať nasledovné:

   a. Analyzovať získané obchodné dáta o vybranom inštrumente.

   b. Pri splnení zadaných pravidiel (napríklad prerazení S/R úrovne,...) zašle užívateľovi upozornenie.

   c. Pripraví vhodný obchodný príkaz podľa vybraných nastavení, ktorý bude užívateľ môcť modifikovať, odoslať alebo odmietnuť k príslušnému brokerovi.

# Kapitola 1
# Analýza

*Kapitola sa zaoberá analýzou existujúcich riešení a stanovených cieľov práce. V jednotlivých podkapitolách sú skúmané existujúce riešenia, ktoré umožňujú zasielanie upozornení na základe vývoja na trhu, možnosti získavania trhových dát, brokeri poskytujúci zadávanie obchodných príkazov prostredníctvom aplikačného rozhrania, možnosti zasielania upozornení a spôsoby spracovania získaných dát za účeľom vytvorenia ponuky nastavení. Na záver kapitoly sú získané poznatky zhrnuté do funkčných požiadaviek, nefunkčných požiadaviek a prípadov použitia.*

## 1.1 Existujúce riešenia

Na trhu existuje mnoho rôznych platforiem, ktoré sprostredkovávajú obchod na kapitálových trhoch. Tieto platformy obvykle svojím užívateľom poskytujú istú formu rudimentárnych upozornení. V tejto podkapitole je zrealizovaný náhľad na poskytované riešenia jednotlivých platforiem, ktoré sú rozdelené do kategórií.

### 1.1.1 Najznámejšie platformy

Prvá kategória existujúcich riešení zobrazuje najznámejšie obchodné platformy:

- **TD Ameritrade** je obchodná platforma odporúčaná portálom Forbes ako najvhodnejšia pre začiatočníkov a zároveň ako platforma s najkvalitnejšou mobilnou aplikáciou.[1] Platforma je dostupná prostredníctvom desktopovej a mobilnej aplikácie. V oboch formách je možné vytrvoriť upozornenia, ktoré sú následne zasielané prostredníctvom emailov, alebo doručované prostredníctvom push notifikací na mobilné zariadenie. Upozornenia je možné vytvoriť v nasledujúcich kategóriach:[2] [3]
  - Cena vybraného finančného inštrumentu dosiahne vybranú hladinu.
  - Typ upozornení špecifický pre danú platformu, ktorý užívateľa upozorní o vybranej rekurentnej udalosti spojenej so sledovaným finančným inštrumentom (napr. publikácia ziskov).
- **Interactive Brokers** je obchodná platforma odporúčaná pre profesionálnych obchodníkov.[1] Poskytuje desktopovú aplikáciu TWS (Trader Workstation) a mobilnú aplikáciu IBKR Mobile, prostredníctvom ktorých sa dajú vytvoriť upozornenia. Na základe vytvorených upozornení TWS buď vykoná predvolený príkaz a/alebo zašle užívateľovi sms a/alebo email. Upozornenia sú poskytované v nasledujúcich kategóriach:
  - Cena vybraného finančného inštrumentu dosiahne vybranú hladinu.

- U vybraného finančného inštrumentu nastane vybraná percentuálna zmena.
- Nastane obchod vybraného finančného inštrumentu vo vybranom trhu.
- Obchodované množstvo vybraného finančného inštrumentu dosiahne vybranú kvantitu.
- IB poskytuje ešte niekoľko kategórií upozornení, pre túto prácu sú však irelevantné. Zahŕňajú kategórie ako čas (obdoba budíka) alebo P&L (Profit and Loss, metrika viazaná k účtu užívateľa).

- **Fidelity** je odporúčaná online obchodná platforma.[1] Poskytuje desktopovú a mobilnú aplikáciu, v ktorých sa dajú nastaviť upozornenia zasielané prostredníctvom SMS, email-u a push notifikácií na mobilné zariadenie. Upozornenia sú poskytované v nasledujúcich kategóriach:[4]
  - Cena vybraného finančného inštrumentu dosiahne vybranú hladinu.
  - U vybraného finančného inštrumentu nastane vybraná percentuálna zmena.
  - Exponenciálny kĺzavý priemer[1] dosiahne vybranej úrovne.
  - V prípade dosiahnutia 52 týždenného maxima/minima vybraného finančného inštrumentu.
- **Tastyworks** je odporúčaná platforma pre obchodovanie opcií.[1] Poskytuje zasielanie upozornení prostredníctvom emailu a push notifikácií v mobilnej aplikácii. Upozornenia sú vytvárané v nasledujúcej kategórii.[5]
  - Cena vybraného finančného inštrumentu dosiahne vybranú hladinu.

### 1.1.2 Primárne mobilné platformy

V nasledujúcej časti sú spracované platformy, ktoré sú primárne zamerané na distribúciu prostrednícvtom mobilnej aplikácie.

- **Etoro** je obchodná platforma distribuovaná na mobilné zariadenia, ktorej upozornenia sú zasielané prostrednícvtvom push notifikácií na mobilnom zariadení. Upozornenia sú poskytované v nasledujúcich kategóriach:[6]
  - Cena vybraného finančného inštrumentu dosiahne vybranú hladinu.
  - U vybraného finančného inštrumentu nastane vybraná percentuálna zmena.
- **Plus500** je mobilná obchodná platforma, ktorá poskytuje upozornenia doručované prostredníctvom push notifikácií. Upozornenia sú poskytované v nasledujúcich kategóriach:[7]
  - Cena vybraného finančného inštrumentu dosiahne vybranú hladinu.
  - U vybraného finančného inštrumentu nastane vybraná percentuálna zmena.
  - Špecifické kritéria pre danú platformu, ktoré sleduje sentiment užívateľov platformy Plus500 a posiela upozrnenia v prípade zaznamenania zvýšeného/zníženého percentuálneho pomeru nákupov v rámci transakcií u vybraného finančného inštrumentu.
- **Robinhood** je mobilná platforma dostupná aj v desktopovej distribúcii. Upozornenia sú doručované prostredníctvom push notifikácií na mobilné zariadenie a dostupné sú nasledujúce kategórie:[8]
  - Cena vybraného finančného inštrumentu dosiahne vybranú hladinu.
  - U vybraného finančného inštrumentu nastane vybraná percentuálna zmena.
  - V prípade dosiahnutia 52 týždenného maxima/minima vybraného finančného inštrumentu.

[1] Vážený priemer, ktorý dáva väčšiu váhu najnovším hodnotám.

### 1.1.3 Samostatné riešenia

Na trhu existujú samostatné riešenia zasielania upozornení, ktoré posielajú upozornenia na základe sofistikovanejších monitorovacích systémov. Nejedná sa, ako v predchádzajúcich príkladoch o upozornenia vyhodnocujúce elementárne ukazovateľe, ale častokrát sa jedná o hlbšiu analýzu vykonávanú, či už automaticky prostredníctvom algoritmov, alebo manuálne na základe istej formy analýzy trhového prostredia. Tieto riešenia sú zväčša paušálne spoplatnené a upozornenia majú slúžiť ako príručka pre investorov.

- **Seeking Alpha** je webová aplikácia, ktorá využíva kvantitatívnu analýzu na analýzu vývoja finačných inštrumentov. Výsledkom tejto analýzy sú rôzne odporúčania vo forme článkov, alebo zaradenia do doporučovanej kategórie. Tieto kategórie sú napríklad:
  - Strong Buy
  - Buy

  Seeking Alpha poskytuje doručovanie upozornení prostredníctvom sms, e-mailov, alebo Viberu. Upozornenia je možné vytvárať na základe sektora, tématiky, trhovej kapitalizácie[2], autora alebo tickeru.[9]
- V tejto kategórii existuje mnoho ďalších riešení, ktoré sľubujú zaručený zisk za jednorazový, poprípade paušálny poplatok. Nie vždy však sú tieto údaje kredibilné ako príklad je uvedená platforma 1000pipbuilder.

Na záver podkapitoly je vhodné spomenúť skutočnosť, že mnoho brokerov poskytuje podobné riešenia ako spoemnutí brokeri, ale tieto riešenia sú skryté za dotatočné poplatky. Taktiež je vhodné pododknúť, že prierez trhom sa môže zdať náhodný a z časti aj je, ale dôležité je uvedomiť si, že porovnávanie platforiem na základe objemu obchodov alebo počtu užívateľov nemusí byť v tomto prípade relevantné. Platformy zamerané na bežnych užívateľov majú väčší počet užívateľov, zaiaľ čo platformy zamerané na profesionálnych obchodníkov majú väčší objem obchodov.

[2]Celková hodnota všetkých vydaných akcií/mincí v obehu.

## 1.2 Zber trhových dát

Po prieskume dostupných riešení zameraných na poskytovanie trhových dát možno konštatovať, že existuje mnoho riešení od rôznych poskytovateľov s rôznymi výhodami a nevýhodami. Každé riešenie je špecifické svojím zameraním, implementáciou, cenovým plánom, a inými parametrami. Práca v nasledujúcej časti opisuje najrelevatnejšie poskytované rozhrania, okrajovo sú spomenuté aj riešenia, ktoré nespĺňajú požiadavky práce, ale sú v kontexte významné.

Na úvod je dôležité načrtnúť význam vybraných typov dát, ktoré sa dajú z finančných trhov získavať:

- **Intradenné dáta** sa viažu k finančným inštrumentom, ktoré sú obchodované počas bežného pracovného dňa. Tieto dáta zachycujú minimá a maximá, ktoré počas dňa jednotlivé inštrumenty dosahujú.[10]
- **Historické dáta** sú dáta, ktoré zachycujú hodnoty vybraného finačného inštrumentu vo vybranom časovom období. Toto obdobie môžu byť napríklad dni, mesiace, alebo roky.
- **Dáta v reálnom čase** sú dáta, ktoré poskytujú informácie o vybranom finančnom inštrumente získavané tesne pred zadaním obchodného príkazu. Uvádzajú napríklad najvyššiu ponúkanú cenu, zobchodovateľnú kvantitu na tejto cenovej úrovni a trh, na ktorom je možné transakciu zrealizovať.[11]
- **Dáta z konca dňa** sú dáta, ktoré zachycujú stav na konci pracovného dňa pre finančné inštrumenty obchodované počas bežných pracovných hodín a po určitej dobe pre inštrumenty obchodované bez prestávky, ako napríklad Forex.

V kontexte práce je taktiež vhodné zmieniť niektoré základné kategórie finančných inštrumentov uvádzaných v tejto časti práce:

- **Akcie** sú cenné papiere, ktoré vyjadrujú podieľ na vlastníctve spoločnosti, ktorou sú vystavované.
- **Forex**[3] je trh s menami, kde je možné vymeniť jednu menu za druhú.
- **Trh s kryptomenamy** je obdoba Forexu, ale miesto fiatových mien sa na trhu obchoduje s kryptomenami.

Nasleduje výpis vybraných poskytovateľov trhových dát:

- **Alpha Vantage** je poskytovateľom trhových dát, ktoré sú prístupné v neupravenej a upravenej podobe. V upravenej podobe sa podrobujú štandardu CRSP[4]. Mimo nespracovanú formu dát o stave finančných prostriekov Alpha Vantage poskytuje aj alternatívne dáta obsahujúce mnoho rôznych ekonomických ukazovateľov, napr. index spotrebiteľských cien alebo medziročnú mieru inflácie.[5]

  Dokumentácia API je štruktúrovaná do logických celkov, poskytuje príklady v niekoľkých vybraných jazykoch, ktorými sú Python, NodeJS, PHP a C#, je prehľadná a kvalitne spracovaná. Mimo oficiálnu distribúciu API existuje aj open-source komunita, ktorá vytvára rôzne balíčky na dopĺňanie funkcionality a podporu ďalších programovacích jazykov. V súčasnosti existuje viac než 600 takýchto balíčkov podporujúcich viac ako 20 dodatočných jazykov. V implementácii práce je táto informácia irelevantná, ale je dôležité ju zmieniť, pretože živá komunita vývojárov dokáže pomôcť v prípade rastu/rozširovania aplikácie.[13]

[3] Z anglického Foreign Exchange.

[4] Formát dát vytvorený inštitútom Center For Research in Security Prices, využívaný v obore finančnými inštitúciami a akadémiami.[12]

[5] Oba ukazovateľe sa vzťahujú na Spojené štáty americké a USD.

Väčšina funkcionality API je dostupná bez poplatkov. V nespoplatnenom režime je možné dotazovať API päťstokrát denne s maximálnou frekvenciou 5 žiadostí za minútu. Spoplatnenie začína na 49,99 USD mesačne, čo umožňuje užívateľom zadávať 75 žiadostí každú minútu a odstraňuje limit na maximálne množstvo denných dotazov. Alpha Vantage taktiež poskytuje akademický prístup k svojim službám, ktorý je flexibilný na základe potrieb a nemá denné obmedzenie dotazov.

Alpha Vantage poskytuje dáta v nasledujúcich kategóriach:

- Dáta v reálnom čase.
- Historické dáta.
- Dáta z konca dňa.
- Intradenné dáta.

Tieto dáta sú dostupné v nadväznosti na nasledujúce nástroje:

- Akcie.
- Forex.
- Kryptomeny.

- **Yahoo finance API** je rozhranie poskytované prostredníctvom platformy Rapid API. Rapid Api je najväčšia platforma poskytujúca rôznorodé aplikačné rozhrania v kategóriach ako cestovanie, hudba, financie,... Rozhrania z finančného sektora nie sú primárne zamerané na zber trhových dát, patria sem aj rozhrania, ktoré napríklad sprostredkovávajú prístup k bankovým účtom a podobne. Vzhľadom na veľkú ponuku rozhraní je však možné medzi viac než 400 rozhraniami finančného charakteru nájsť aj tie zamerané na zber trhových dát. Rapid API poskytuje 3 vrstvy aplikačných rozhraní na základe modelu poplatkov:
  - Plná funkcionalita rozhrania je dostupná bez poplatkov.
  - Rozhranie je dostupné bez poplatkov, ale istá časť funkcionality je spoplatnená.
  - Akákoľvek funkcionalita rozhrania je skrytá za paušálny, alebo špecifický poplatok.

  Prehľad poskytovaných rozhraní a model poplatkov, do ktorého spadajú je intuitívny a dá sa v ňom jednoducho zorientovať. Rapid API mimo iné poskytuje aj prístup k rozhraniu Alpha Vantage, najdôležitejšie rozhranie ku ktorému poskytuje prístup je však spomínané Yahoo Finance API distribuované tvorcom rozhraní API Dojo. API Dojo vytvorili a prevádzkujú niekoľko rôznorodých rozhraní a v súvislosti s indikátormi spoľahlivosti, ktoré sú na stránkach Rapid API dostupné a živou diskusiou k problémom s využívaním API je možné konštatovať, že táto verzia Yahoo Finance API poskytuje korektné dáta, je udržovaná a podporovaná.[14] Uvedené skutočnosti sú dôležité vzhľadom na to, že oficiálna verzia Yahoo Finance API, od roku 2017 nie je podporovaná. Na stránkach rozhrania prislúchajúcih Yahoo Finance API v rámci portálu Rapid API existuje istá forma testovacieho prostredia, v ktorom sa dajú zadať parametre vstupov a výstupom je hotový výstrižok kódu, ktorý zrealizuje volanie rozhrania v nadväznosti na zvolené parametre. Tento kód stačí skopírovať, čo do značnej miery uľahčuje vývoj a implementáciu. Yahoo Finance spadá do druhej skupiny rozhraní poskytovaných portálom Rapid API a tým je model v ktorom je funkcionalita rozhrania bez poplatkov do 250 volaní za mesiac a pri prekročení tohto objemu je nutné platiť za každé volanie na základe stanovených poplatkov. Komunikácia s rozhraním priebeha prostredníctvom HTTP.

  V prípade využívania Yahoo Finance API v inom ako základnom modeli nastáva komplikácia s implementáciou a tou je možnosť prekročenia limitu požiadavky na rozhranie v priebehu vybraného mesiaca. Tento údaj je nutné sledovať prostredníctvom nástroja poskytovaným platformou Rapid API a uistiť sa o jeho neprekročení, v opačnom prípade je nutné hradiť poplatky za využívanie mimo vymedzenú mieru. Každé volanie rozhrania mimo pôvodných

250 je spoplatnené 0,01 USD za volanie. Nevýhodou tohto riešenia je aj spôsob získavania dát rozhraním samotným, ktoré z časti získavá dáta z portálu `www.yahoofinance.com`. [15]

- **Quandl** agreguje ekonomické, finančné a alternatívne databázy poskytujúce prístup k obrovskému množstvu informácií. Obsahuje 20 miliónov databáz od viac ako 500 dodávateľov, z čoho vyplýva, že platforma poskytuje prístup ku rôznym typom dát finančného charakteru. Niektoré typy dát sú prístupné bez poplatkov. Quandl operuje v modeli rozdelenom do dvoch skupín prístupu k dátam:
  - Voľné dáta:
    * Väčšinou pochádzajú z kredibilných inštitúcií, ako centrálne banky alebo inštitúcie vlád.
    * Neexistuje garancia pravidelnej aktualizácie, dáta sú aktualitozované príležitostne.
  - Prémium:
    * Pochádzajú od poskytovateľov dát ako AlgoSeek, Applied Research, Barchart,...
    * Sú aktualizované na pravidelnej báze.

  Kvalita získaných dát priamo závisí na skupine, do ktorej poskytovatelia spadajú. Dôležité je však uviesť, že každá databáza na platforme Quandl obsahuje vzorové dáta, ktoré je možné získať bez poplatkov. Problémom v kontexte práce je, že dáta ktoré sú dostupné bez poplatkov sú otázneho charakteru v kontexte potencionálneho využitia. Patria sem napríklad dáta z honkonskej buzy. Ďalším evidentným problémom je cena prémiových dát, ktorá závisí od vybranej databázy, z ktorej užívateľ čerpá. V porovnaní s konkurenciou tento model nedokáže súťažiť, vzhľadom na to, že niektoré databázy začínajú na cenovej hladine 10 000 USD ročne. Vo výsledku Quandl poskytuje vysokú kvalitu dát, avšak služby tejto platformy sú primárne určené väčším spoločnostiam, ktoré si náklady spojené so získavaním týchto dát dokážu dovoliť. Komunikácia s API prebieha prostredníctvom RESTového rozhrania.

  Alternatívne dáta vo finančnom kontexte zachycujú širokú škálu ukazovateľov, ktoré nepriamo súvisia s aktivitou vybraných ukazovateľov. Najčastejšie sú spájané s rozhodovacím procesom hedžových fondov, a ako príklad je možné uviesť premávku na stránke vybranej spoločnosti, geolokáciu, transakcie kreditných kariet,... Z týchto dát je následne možné idenfitikovať sentiment kupujúcich čo do značnej miery ovplyvní hodnotu akcií spoločnosti.[16]

  Quandl poskytuje dáta vo všetkých uvedených kategóriach:

  - Dáta v reálnom čase.
  - Historické dáta.
  - Dáta z konca dňa.
  - Intradenné dáta.

- **Iex Cloud** je platforma sprostredkovajúca finačné dáta. Primárne sa zameriava na poskytovanie dát vo formáte časových radov.[6] Poskytuje 500 000 jadrových volaní na mesiac v skúšobnej verzii, čo je na prvý pohľad veľký objem dát, ale každé volanie rozhrania je počítané na základe rôznych faktorov a odpočítava sa z mesačných kreditov. Treba preto monitorovať spotrebu volaní, o čo sa stará automatický sledovač. Spotrebu kreditov je možné vypočítať aj na základe dokumentácie. Hodnota spotreby závisí mimo iného na množstve dát v požiadavke, ako často sa dáta obnovujú, koľko reálne stojí zaobstarať dané dáta, koľko stojí distribuovať dané dáta,... Samotná dokumentácia je rozsiahla, ale nie je intuitívna. Komunikácia je realizovaná prostredníctvom RESTového rozhrania. Iex Cloud poskytuje dáta v nasledujúcich kategóriach:[17]
  - Historické dáta.

[6] Dáta sú zoradené v závislosti na čase.

  - Intradenné dáta.
  - Dáta v reálnom čase.[7]

- **Google Finance API** je rozhranie, ktoré v roku 2012 stratilo oficiálnu podporu.[18] Podobne ako Yahoo Finance API, je však stále relevantným prostriedkom na získavanie trhových dát. Na rozdiel, od Yahoo Finance API však neexistuje komunitou sprostredkované a udržované rozhranie. Pri komunikácii je preto využívané oficiálne rozhranie, ktorého dokumentácia je slabá, niekedy neexistujúca. Oficiálna dokumnetácia je v niektorých prípadoch nedohľadateľná, a preto sa treba spoliehať na dokumentáciu tretích strán. Komunikácia prebieha prostredníctvom RESTového rozhrania. Vzhľadom na komerčné použitie je Google Finance API irelevantné, existuje však jeden spôsob využitia, ktorý je stále relevantný. Tento spôsob zakladá na skutočnosti, že Google Finance API je priamo integrované s prostedím Google Sheets a je tak možné dáta získávať priamo v Exceli.

- Platformy, ktoré poskytujú API na zadávanie obchodných príkazov poskytujú istú formu trhových dát. Tieto dáta sú však často skryté za dodatočnými poplatkami, pochádzajú z jedného alebo viacerých uvedených zdrojov, alebo sú limitované svojim obsahom. Vzhľadom na tieto skutočnosti preto nebudú v práci ďalej skúmané.

Na zákalade uvedených skutočnosti je voľba rozhrania Alpha Vantage najvhodnejšia pre vypracovanie práce. Alpha Vantage je najvhodnejšie riešenie na základe dostupnosti, kvality a množstva poskytovaných dát v bezplatnom modeli. Taktiež má bohatú dokumentáciu a živé vývojárské prostredie.

[7] Dáta v reálnom čase nie sú poskytované v originálnom kontexte, ale v kontexte, že IEX cloud je schopný dané dáta doručiť akonáhľe sú dostupné, nie okmažite.

## 1.3 Brokeri poskytujúci verejne dostupné aplikačné rozhranie

Existuje niekoľko kategórií, do ktorých sa dajú zoskupiť poskytované riešenia. Tieto kategórie sú individuálne skumané v nasledujúcich podkapitolách.

### 1.3.1 Brokeri s vlastnou platformou

V tejto sekcii sú uvedení brokeri, ktorí sa primárne orientujú na klientelu, ktorá využíva nimi poskytované aplikácie, ale taktiež poskytujú API v istej forme. Vybrané riešenia sú nasledovné:

- **Interactive Brokers** poskytujú tri rôzne API:[19] [20]
  - **Client Portal API** je zamerané na správu účtu a operácie s ňou spojené. Umožňuje však taktiež zadávanie obchodných príkazov. Služby rozhrania sú poskytované prostredníctvom RESTového rozhrania, z čoho vyplýva, že napojiť sa naň dokáže široké spektrum programovacích jazykov. Umožňuje realizovať príkazy bez závislosti na distribuovanú platformu WTS. Využívanie API nie je spoplatnené mimo poplatkov využívania služieb platformy Interactive Brokers. Poskytuje elementárnu formu získavania trhových dát.
  - **Trader Workstation API** je API zamerané na obchodnú platformu IB TWS. Podporuje zadávanie obchodných príkazov, monitorovanie stavu na účte a dotazovanie trhových dát. Služby rozhrania sú poskytované v podobe balíčkov pre nasledujúce programovacie jazyky: C++, C#, Java, Python, ActiveX, RTE, DDE. Táto alternatíva je zameraná na vytváranie addonov pre TWS a pre správne fungovanie je nutné mať spustenú instanciu obchodnej platformy TWS. Tým pádom nie je pre prácu relevantná, pretože stanovené podmienky obmedzujú nezávislý beh aplikácie. Využívanie nie je spoplatnené extra poplatkami v porovnaní s využívaním platformy TWS. Spoplatnené sú však služby spojené so získavaním trhových dát, sú rozdelené do balíčkov ako napr. správy, udalosti, apod., v niektorých z nich existujú služby zdarma. Väčšina kvalitných dát je skrytá za platobnú bránu.
  - **Fix API** poskytuje rozhranie pre inštitúcie s pokročilou výpočetnou technikou, čím je im umožnené využívať vysokorýchlostné zadávanie obchodov. Tento model je adekvátne spoplatnený na úrovni 1000 USD/mesiac.

  Spoplatnenie využívania API je komplikované. Založenie účtu je bez poplatkov, na založenie účtu, ktorý využíva páky je potreba 2000 USD a obchodovanie s jednotlivými inštrumentmi môže, ale nemusí byť spoplatnené. Napr. obchodovanie akcií nie je spoplatnené, opcie a pod. sú spoplatnené rôzne, napr. 0,15-0,65USD za opciu, v závislosti na obchodovanom množstve. IB je kvalitná platforma, ale je zameraná na pokročilejších obchodníkov.

- **Etoro**[21] poskytuje API pre developerov v nasledujúcich kategóriach:
  - Discovery je rozhranie spojené s funkcionalitou platformy, umožňuje vyhladávať iných užívateľov.
  - Metadata poskytuje prístup k metadátam na ktorých je založená platforma, napr. dostupné typy finančných inštrumentov a ich zoskupenia.
  - Systém obsahuje informácie o systéme ako napr, trhové udalosti.
  - User poskytuje rozhranie pre získavanie informácií o účte užívateľa aplikácie/zorhania.

  Aj napriek tomu, že Etoro má dostupné API nedovoľuje cezeň obchodovať a teda je pre prácu irelevantné.

- **TD Ameritrade** poskytuje API v 11 kategóriach, jednotlivé kategórie sú zamerané na správu účtu, získavanie trhových dát, a iné. Jednou z kategórií je zadávanie obchodov. Výhody tohto API sú renomé sprostredkovateľa, dobrá dokumentácia a široký výber produktov na obchodovanie. Nevýhodou je komplexnosť prostredia ThinkOrSwim pre nováčikov a vysoké maržové sadzby. [8]

### 1.3.2 Brokeri zameraní na poskytovanie API

Táto sekcia obsahuje brokerov, ktorí sa primárne zameriavajú na realizáciu obchodov prostredníctvom poskytovaného API.

- **Alpaca** je broker zameraný primárne na obchodovanie prostredníctvom svojho rozhrania. Táto skutočnosť je evidentná už na prvý pohľad, pretože Alpaca neposkytuje desktopovú, ani mobilnú aplikáciu. Poskytuje však výbornú kvalitu dokumentácie svojho rozhrania, ktorá je zároveň veľmi prehľadná. Alpaca API taktiež nevyberá poplatky od bežných klientov za zrealizované transakcie a v prípade, že sa nejedná o transakcie spadajúce pod maržovú sadzbu. Spoplatnená je prémiová funkcionalita, napr. prístup k špecifickým trhovým dátam. Istou nevýhodou je, že Alpaca poskytuje len finančné inštrumenty ochodované na trhoch v SŠA. Trhové dáta, ktoré sú v rámci Alpaca API dostupné, nie sú vlastné trhové dáta, ale len sprostredkované trhové dáta od poskytovateľov ako Alpha Vantage. Oficiálne podporuje nasledujúce progromavacie jazyky: Python, C#, Node, Go, ale existuje aj vela ďalších jazykov podporovaných komunitou avizovaných na ich stránkach. Poskytuje prístup k obchodom s akciami, kryptu a EFT.[22][9]

- **Tradier**: Poskytuje API na získavanie trhových dát a API na zadávanie obchodných príkazov. Poskytuje obchodovanie s akciami, opciami,... neposkytuje však obchodovanie s kryptomenami. Model poplatkov je bez poplatkov za akcie, obchodovanie opcií je spoplatnené 0.35USD/opciu. Model je ďalej spoplatnený paušálnym poplatkom 30 USD/mesiac. Maržové sadzby sú na úrovni 5.25% ročne. Obsahuje kvalitnú dokumentáciu a API je dotazovaným prostredníctvom RESTového rozhrania.[23]

Existujú aj ďalšie riešenia primárne zamerané na obchodovanie prostredníctvom API ako napr. Cunningham Trading Systems[24], ale tieto nebudú rozobrané, pretože neprinášajú nič odlišné od vyššie spomenutých nie sú atraktívne svojou kvalitou, alebo nekonkurujú cenovým modelom.

Alpaca API je pre potreby práce ideálnym riešením. Má skvelú dokumentáciu, podporu pre širokú škálu programovacích jazykov a relatívne zhovievavé spoplatnenie.

[8] Maržová sadzba je poplatok, ktorý vzniká pri využívaní finančných prostriedkov obchodníka ako napr. obchodovanie s pákou.

[9] Burzovo obchodovatelné fondy ako napr. NASDAQ

## 1.4 Zasielanie upozornení

Cieľom upozornení v kontexte práce je predať informáciu o možnosti zrealizovania operácie na finančnom trhu na základe predvolených nastavení. Na základe predaných informácií bude užívateľ schopný rozhodovať o odmietnutí, modifikovaní poprípade prijatí ponúkanej pozície. Preto treba pri zasielaní upozornení uvažovať v kontexte celého tohto procesu. Z uvedených skutočností vyplýva, že je potrebné jednak implementovať prenos a zrozumiteľné zobrazenie infomácií a jednak možnosť interakcie s poskytnutými informáciami na prijatie, modifikáciu alebo zamietnutie ponúkanéj pozície. Ponúka sa niekoľko možností, ako realizovať výmenu informácií medzi aplikáciou a jednotlivými užívateľmi a následne spracovať užívateľskú odozvu:

- Prvým spôsobom je implementácia samostatnej mobilnej/desktopovej aplikácie, čo ale do značnej miery limituje potencionálných užívateľov a prináša do procesu implementácie značné komplikácie. Vývoj aplikácie pre jednotlivé prostredia nemusí byť priamočiary a voľba cieľovej platformy limituje zariadenia operujúce na ostatných platformách.
- Druhým potencionálnym spôsobom, ako upozornenia zasielať, je vytvorenie webovej aplikácie, ktorá bude zodpovedná za interakciu s užívateľom a jednotlivé upozornenia budú zasielané prostredníctvom existujúcich komunikačných kanálov, ako napríklad email alebo aplikácie zamerané na okamžité zasielanie správ. Toto riešenie prináša vysokú mieru flexiblity, pri správnej optimalizácii je možné prispôsopbiť webovú aplikáciu širokej škále zariadení a preto bude práca ďalej skúmať riešenia súvisiace so zobrazovaním informácií prostredníctvom tejto varianty.

### 1.4.1 Existujúce komunikačné kanály

Prvým analyzovaným riešením je zasielanie správ prostredníctvom e-mailu. Výhodou využívania e-mailov na zasielanie správ je rozšírenie smartfónov v populácii, čo umožňuje každému vlastníkovi ľahký prístup k elektronickej pošte. V roku 2020 tvoril podiel vlastníkov smartfónov na celkovej populácii Českej republiky 72,6%.[25] Evidentnou nevýhodou je riziko vyhodnotenia prichádzajúcich e-mailov ako spam, čomu sa však dá do značnej miery vyhnúť správnou optimalizáciou na strane odosielateľa, poprípade upozornením užívateľa na nutnú zmenu nastavení. Posielanie správ prostredníctvom e-mailu je možné realizovať nasledujúcimi spôsobmi:

- Na prvý pohľad jednoduchý spôsob ako posielať email je využiť balíček vybraného programovacieho jazyka[10], zvoliť správne nastavenia a poslať e-mail. V jazyku C# by tento proces využíval namespace System.Net.Mail a vyžadoval by založenie vlastného SMTP serveru. Založiť vlastný SMTP server je možné troma spôsobmi:
  - Vytvorenie samostatnej aplikácie. Táto možnosť je pomerne jednoducho realizovateľná[26], ale udržovanie serveru a zabezpečenie správneho chodu spadá mimo rozsah tejto práce, preto sa ňou ďalej práca nebude zaoberať.
  - Využiť existujúce open-source riešenie a vytvoriť aplikáciu, ktorá bude prevádzkovať SMTP server týmto spôsobom. Realizácia tejto varianty je znovu obsiahla a nespadá do rozsahu práce.
  - Využiť existujúce služby platforiem prevádzkujúcich komerčné SMTP servery.
- Ako je v predchádzajúcom texte uvedené vytvoriť a prevádzkovať SMTP server nie je v kontexte práce ideálne riešenie zasielania e-mailov, a preto je vhodnejšie použiť existujúce riešenie. Existujú komerční poskytovatelia SMTP serverov, ktorí umožňujú využívať svoje služby a odpadá tým nutnosť prevádzkovať SMTP server na vlastné náklady. Na trhu je veľké množstvo poskytovateľov spomínaných služieb, nižšie sú uvedené niektoré z nich:

[10] V rámci stručného prieskumu nebol nájdený programovací jazyk, ktorý danú funkcionalitu nepodporuje.

- **Google** poskytuje SMTP server prostredníctvom svojej e-mailovej služby Gmail. Využívanie služby je v prípade nekomerčného využitia bez poplatkov. V prípade komerčného využívania poskytuje Google službu Google Workspace, ktorú je možné využívať bez poplatkov prvých 14 dní a následne je spoplatnená na minimálnej úrovni 9 USD mesačne. Kvantita odoslaných e-mailov je v prípade Gmailu limitovaná na 500[11] e-mailov denne[12]. Služba Google Workspace má limit na úrovni 2000 e-mailov denne. Uvádzanou nevýhodou využívania SMTP prostredníctvom Gmailu je zníženie bezpečnosti e-mailovej schránky odosielateľa vzhľadom na to, že je nutné povoliť prístup externým aplikáciam.[27]
- **SendGrid** poskytuje SMTP server, ktorý umožňuje zasielať 100 e-mailov denne bez poplatkov. Využívanie služieb je realizované prostredníctvom REST API, čo umožnuje napojenie z veľkého množstva programovacích jazykov. Využívanie nespoplatnenej verzie služby nie je časovo nijak obmedzené, ale doručovanie emailov pre neplatiacich užívateľov môže v istých prípadoch trvať 5—15 minút. Najlacnejšia spoplatnená verzia začína na 15 USD mesačne. Doručovanie e-mailov v tomto prípade trvá 0—0.603 sekúnd. SendGrid taktiež uvádza skutočnosť, že pri vysokých mierach spoplatnenia[13] má priemernú doručitelnosť e-mailov 97%.[28]
- **Mailgun** poskytuje SMTP službu, prostredníctvom ktorej je možné zasielať 5000 e-mailov mesačne v najnižšom cenovom pláne. Tento plán je prvé tri mesiace bez poplatkov, následne je spoplatnený začínajúc na 35 USD. MailGun uvádza, že 99% emailov je doručených do 5 minút. [29]
- **AWS Simple Email Service** poskytuje SMTP server v rámci svojich webových služieb. AWS SES je možné využívať v režime bez poplatkov pod podmienkou, že aplikácia, ktorá SMTP server využíva je nasadená prostredníctvom programu AWS EC2[14]. AWS EC2 je dostupný bez poplatkov, ale v nespoplatnenom režime má iba 750 hodín výpočetného času mesačne. V nespoplatnenom režime poskytuje AWS SES odosielanie 62000 e-mailov mesačne. Po prekročení tejto hranice, alebo v prípade, že aplikácia nie je nasadená v ekosystéme AWS, je každých 1000 e-mailov spoplatnených sumou 0.10 USD. Integrácia AWS SES je možná niekoľkými spôsobmi:[30]
  - Prostredníctvom AWS SDK.
  - Prostredníctvom Amazon SES Console - webové rozhranie pre využívanie služieb AWS SES.
  - Prostredníctvom SMTP.
  - Prostredníctvom SES API.
- Existuje mnoho ďalších poskytovateľov SMTP serverov, ako napríklad Mailkit, SocketLabs, WPOven, Sendinblue, Elastic Email,... Tieto služby nie sú analyzované vzhľadom na skutočnosť, že model ich využívania spadá do jedného z predchádzajúcich, alebo model spoplatnenia nie je relevantný v kontexte implementácie práce.

Zasielanie e-mailov nie je okamžité zo strany príjemcu, vo väčšine prípadov sú e-maily doručené v priebehu niekoľkých sekúnd, ale v špecifických prípadoch sa trvanie tohto procesu môže natiahnuť aj na niekoľko minút. Ďalšou možnosťou je preto zasielanie správ prostredníctvom existujúcej aplikácie zameranej na okamžité doručovanie správ. V tejto kategórii sa ponúka niekoľko možností:

- **Telegram** poskytuje vytvorenie bota, ktorý je schopný prostredníctvom API doručovať správy jednotlivým užívateľom. Jedinou prekážkou vo využívaní tejto služby je nutnosť užívateľa spárovať svoj účet s existujúcim botom, ktorý mu následne bude môcť zasielať správy. Komunikácia s API je založená na HTTP protokole a nie je spoplatnená.[31]

[11] V prípade skupinových emailov sa jedná 500 recipientov denne.
[12] V každom 24 hodinovom období.
[13] Plán Pro 89.95 USD — 750.00 USD.
[14] Amazon Elastic Compute Cloud

- **Facebook Messenger** poskytuje API na zasielanie správ. Podobne ako predchádzajúci Telegram, však potrebuje prvý krok od užívateľa, ktorý musí zahájiť komunikáciu s účtom napojeným na aplikáciu. Využívanie tohto API je bez poplatkov.[32]
- **Viber** poskytuje bezplatné RESTové API prostredníctvom, ktorého možno komunikovať s jednotlivými užívateľmi. Podmienkou pre zahájenie komunikácie je vytvorenie bota, ktorý následne môže komunikovať s užtívateľmi.
- **Signal** poskytuje svojim užívateľom vysokú mieru zabezpečenia, ale neposkytuje žiadne verejne dostupné API prostredníctvom ktorého by bolo možné zasielať správy z aplikácie.
- Existuje mnoho ďalších aplikácií sprostredkujúcich okamžité doručovanie správ. Z nich **WhatsApp** poskytuje API pre stredne veľké a veľké spoločnosti, **WeChat** je primárne zameraný na čínsky trh a **Snapchat** sa zameriava na zasielanie fotografií a videa.

## 1.5 Spracovanie trhových dát

Spracovanie dát musí pre užívateľov aplikácie prebiehať v relevantnom čase. Väčšina produktov je obchodovaných v rámci obchodných hodín búrz, na ktorých s nimi je možné obchodovať. Existujú možnosti, ako obchodovať s produktmi aj mimo obchodných hodín, napr. prostredníctvom tzv. ECNs,[15][33] alebo burzy s kryptomenami operujú nepretržite. Objem takto obchodovaných produktov však nie je tak významný ako počas obchodných hodín. Broker Alpaca poskytuje na výber zo 7 búrz, z ktorých 6 má obchodné hodiny 9:30 — 16:30 ET[16] a jedna 9:30 — 17:00 ET. Najväčšie pohyby na kapitálovych trhoch sú však zaznamenávané krátko po otvorení a tesne pred záverom obchodných hodín.[34] Aplikácia musí brať tieto skutočnosti v úvahu a preto budú jednolivé upozornenia vyhodnocované 15 minút po otvorení burzy a 15 minút pred zatvorením burzy. Vzhľadom na skutočnosť že väčšina búrz, obchodovateľná prostredníctvom brokera Alpaca obchoduje v časovom rozmedzí 9:30 - 16:30 ET, budú upozornenia vyhodnocované a zasielané v časoch 9:45 a 15:45 ET. Z jedného z cieľov práce vyplýva príprava pravidiel spracovanie dát, ktoré budú užívatelia voliť pri vytváraní upozornení. Prvotným cieľom práce bolo vytvoriť niekoľko rôznych pravidiel spracovania trhových dát, ktoré budú ponúkané užívateľom, a na základe ktorých budú môcť dostávať upozornenia. Analýza týchto modelov je však časovo náročná a bolo preto nutné obmedziť vybrané riešenia na tri jednoduché kategórie:

- **Absoútna zmena** sleduje absolútnu zmenu vybraného produktu v porovnaní so základnou cenou.
- **Percentuálna zmena** sleduje percentuálnu zmenu vybraného produktu v porovnaní so základnou cenou.
- **Trendová hranica** sleduje prerazenie podpory/odporu prvej úrovne počítanej z dostupných dát z predošlého dňa.

Výpočet absolútnej a percentuálnej zmeny sledovanej hodnoty je triviálny. Prvá úroveň podpory a odporu sa počíta na základe pivotov, pre výpočet boli zvolené klasické výpočty podpory a odporu. Pre ilustráciu je vhodné uviesť, že existuje viacero metód na výpočet podpory a odporu napr. Fibonacciho.

Aplikácia počíta trendové hranice následovne:[35]

$$P = (H + L + C)/3$$

Kde H je najvyššia hodnota predchádzajúceho sledovaného obdobia, L je najnižšia hodnota predchádzajúceho sledovaného obdobia, C je finálna hodnota predchádzajúceho sledovaného obdobia a P je pivot. Po získaní pivotu je možné vypočítať prvú hladinu podpory:[35]

$$S = 2 * P - H$$

A prvú hladinu odporu:[35]

$$R = 2 * P - L$$

Predchádzajúce sledované obdbobie budú v rámci aplikácie dostupné dáta pre predchádzajúci deň. Realistické použitie je nastavenie upozornenia a v prípade, že nebude prerazená trendová línia v priebehu niekoľkých dní a následne sa tak učiní, je možné uvažovať o existujúcom trende a vytvorení pozície.

[15] Electronic communication networks.
[16] Eastern Time.

## 1.6 Funkčné požiadavky

Nasledujúca časť práce obsahuje funkčné požiadavky aplikácie vyplývajúce z predošlej analýzy a cieľov práce.

### 1.6.1 Funkčné požiadavky užívateľských účtov

- **F1.1 Registrácia** - užívateľský účet bude možné registrovať za pomoci emailovej adresy a zvoleného hesla. Emailová adresa bude špecifická pre každý užívateľský účet.
- **F1.2 Prihlásenie** - užívateľský účet bude možné prihlásiť použitím zvolenej kombinácie emailovej adresy a hesla.
- **F1.3 Odhlásenie** - užívateľský účet bude možné po prihlásení odhlásiť.
- **F1.4 Obnova hesla** - užívateľskému účtu bude možné priradiť nové heslo.
- **F1.5 Nastavenie účtu** - po prihlásení bude užívateľskému účtu umožnené vyplniť potrebné nastavenia pre správne fungovanie aplikácie. Bez dodatočného vyplnenia týchto údajov nebude sprístupnený zvyšok aplikácie. Tieto nastavenia tvoria:
  - Alpaca API kľúč - povinný údaj.
  - Alpaca tajný kľúč - povinný údaj.
  - Užívateľské meno v aplikácií Telegram - voliteľný údaj.
  - Schválenie zasielania správ prostredníctvom aplikácie telegram - voliteľný údaj.
- **F1.6 Zmena nastavení účtu** - uživeľský účet bude môcť meniť nastavenia účtu spomínané vo funkčnej požiadavke F1.5. Táto požiadavoka nie je triviálna, vzhľadom na to, že Alpaca na požiadanie vždy generuje novú kombináciu tajného a API kľúča.

### 1.6.2 Funkčné požiadavky upozornení

- **F2.1 Vytvoriť nové nastavenie upozornenia** - nastavenia upozornení bude možné vytvárať niekoľko typov, všetky typy budú potrebovať nasledujúce informácie:
  - Pomenovanie upozornenia
  - Burzu, na ktorej sa obchoduje daný produkt - v nadväznosti na zvoleného brokera sa nám ponúka sedem možností:
    * Nysearca.
    * Nyse.
    * Nasdaq.
    * Bats.
    * Amex.
    * Arca.
    * Otc.
  - Názov produktu - názov produktu bude vždy závisieť na zvolenej burze, užívateľovi bude poskynutý výber všetkých produktov obchodovateľných na vybranej burze.
  - Dátum expirácie - stanovuje dátum do ktorého(vrátane) bude upozornenie vyhodnocované.
  - Typ upozornenia - podporované typy budú:
    * Absolútna zmena.

    - Percentuálna zmena.
    - Trendové ohraničenia.
- Súčasná cena vybraného produktu - táto hodnota nebude modifikovateľná, bude získaná od poskytovateľa trhových dát. Jednotlivé typy budú vyžadovať špecifické dodatočné údaje, preto existujú nasledujúce požiadavky viažuce sa k vytvoreniu nového nastavenia upozornenia:
    - **F2.1.1 Vytvoriť nové upozornenie sledujúce absolútnu zmenu ceny sledovaného produktu** - potrebné dodatočné údaje sú:
        - Smer sledovanej zmeny - prírastok/pokles hodnoty.
        - Hodnota predpokladanej zmeny.
    - **F2.1.1 Vytvoriť nové upozornenie sledujúce percentuálnu zmenu ceny sledovaného produktu** - potrebné dodatočné údaje sú:
        - Smer sledovanej zmeny - Prírastok/pokles hodnoty.
        - Hodnota predpokladanej zmeny.
    - **F2.1.1 Vytvoriť nové upozornenie sledujúce prerazenie trendovej hranice** - úotrebné dodatočné údaje sú:
        - Trendová hranica - Podpora/odpor.
- **F2.2 Uloženie upozornenia** - Upozornenie bude uložené v databáze aplikácie.
- **F2.3 Vyhodnotenie upozornenia** - Všetky uložené upozornenia, ktoré nie sú expirované a naplnené budú vyhodnocované 9:45 a 15:45 ET.
- **F2.4 Zasielanie upozornení** - V prípade úspešného naplnenia podmienok nastavenia upozornenia, bude užívateľovi zaslaná správa na vybrané komunikačné kanály. Email bude odoslaný vždy a telegram v prípade, že si to užívateľ navolil.
- **F2.5 Náhľad na existujúce nastavenie upozornenia** - Zobrazí vybrané upozornenie a všetky navolené hodnoty.
- **F2.6 Prehľad existujúcich upozornení** - Zobrazenie všetkých existujúcich nastavení upozornení v tabuľke s označím naplnených a expirovaných nastavení.
- **F2.7 Odstránenie nastavenia upozornení** - Umožnenie odstránenia akéhokoľvek existujúceho nastavenia upozornení, jednak priamo z náhľadu na všetky upozornenia, alebo z náhľadu na jednotlivé nsatavenie upozornenia.
- **F2.8 Zobrazenie grafu** - Vyobrazí paličkový graf všetkých dostupných EOD hodnôt získanych od poskytovateľa trhových dát. Tento graf sa bude zobrazovať v náhľade na jednotlivé upozornenia a taktiež po zvolení sledovaného produktu v novom nastavení upozornení.

### 1.6.3 Funkčné požiadavky navrhovaných pozícií

- **F3.1 Vytvorenie navrhovanej pozície** - v prípade naplnenia nastavenia bude vytvorená navrhovaná pozícia s nasledujúcimi údajmi:
    - Meno - zhodné s menom nastavenia upozornenia.
    - Produkt - zhodné s produktom nastavenia upozornenia.
    - Základná cena - cena ktorá bola aktuálne dostupná pri vytvorení nastavenia upozornenia.
- **F3.2 Prehľad navrhovaných príkazov** - zobrazí všetky dostupné navrhované príkazy prihláseného užívateľského účtu.

- **F3.3 Náhľad na navrhovaný príkaz** - náhľad bude obsahovať dodatočné informácie:
  - Súčasnú hodnotu sledovaného produktu.
  - Paličkový graf so všetkými dostupnými trhovými dátami sledovaného produktu.

  Ďalej bude obsahovať nasledujúce voliteľné inforácie:

  - Typ obchodovaného množstva
    * Akcie.
    * Absolútna honota.
  - Množstvo zvoleného typu.
  - Možnosť použitia papierového účtu.
  - Možnosť použitia príkazu stop loss - v prípade využitia sa zobrazí okno pre vyplnenie hodnoty, pre ktorú aplikovať stop loss v USD.
  - Možnosť použitia príkazu take profit - v prípade využitia sa zobrazí okno pre vyplnenie hodnoty, pre ktorú aplikovať take profit v USD.
- **F3.4 Odstránenie navrhovanej pozície** - odstráni navrhovanú pozíciu z prehľadu pozícií alebo z náhľadu na pozíciu.
- **F3.5 Odoslanie príkazu na základe vyplnenej pozície** - odošle príkaz brokerovi v prípade, že je možné pozíciu zrealizovať.

## 1.7 Nefunkčné požiadavky

Táto časť práce obsahuje nefunkčné požiadavky vyplývajúce z predošlej analýzy a cieľov práce.

- **N1 Webová aplikácia** - Aplikácia bude implementovaná formou webovej aplikácie.
- **N2 Podpora Alpha vantage** - Python, NodeJs, Php, alebo C#, komunikácia prebieha prostrednictvom HTTP, ale tieto jazyky sú oficiálne zdokumentované a podporované.
- **N3 Podpora Alpaca API** - Python, C#, NodeJs, Go

## 1.8 Prípady použitia

Nasledujúca časť práce popisuje prípady použitia aplikácie z užívateľskej perspektívy, tieto prípady použitia vychádzajú z funkčných požiadaviek. Prípady použitia nie sú rozoberané dopodrobna, pretože funkčné požiadavky dopodrobna určujú všetky nutné náležitosti na ilustráciu týchto prípadov použitia.

### 1.8.1 UC1 Prípady použitia aplikačného účtu

- **UC1.1 Prihlásenie sa.**
- **UC1.2 Vytvorenie nového účtu.**
- **UC1.3 Zmena hesla.**
- **UC1.4 Odhlásenie sa.**
- **UC1.5 Vyplnenie nastavení účtu.**
- **UC1.6 Zmena nastavení účtu.**

### 1.8.2 UC2 Prípady použitia nastavení upozornení

- **UC2.1 Vytvorenie nového nastavenia upozornenia.**
- **UC2.2 Vymazanie existujúceho nastavenia upozornenia.**
- **UC2.3 Prehľad existujúcich nastavení upozornení.**
- **UC2.4 Náhľad na existujúce nastavenie upozornena.**
- **UC2.5 Obdržanie upozornenia na základe naplnenia nastavenia.**

### 1.8.3 UC3 Prípady použitia navrhovaných pozícií

- **UC3.1 Náhľad na navrhovanú pozíciu.**
- **UC3.2 Odstránenie navrhovanej pozície.**
- **UC3.3 Vyplnenie navrhovanej pozície a následné potrvrdenie príkazu.**

# Kapitola 2
# Praktická časť

*Nasledujúca kapitola sa zaoberá návrhom aplikácie, výberom vhodných technológií, následnou implementáciou a na záver testovaním výslednej aplikácie.*

## 2.1 Použité technológie

Nasledujúca časť práce sa zaoberá výberom a popisom zvolených návrhových vzorov a databázovéj schémy.

### 2.1.1 Klient-server architektúra

Vzhľadom na skutočnosť, že výsledným produktom implementácie je webová aplikácia odpadá možnosť vytvorenia distribuovanej aplikácie a najvhodneším riešením implementácie je preto využiť model klient-server. V kontexte práce tvorí server backend aplikácie, na ktorom sa nachádza výpočetná logika a klient tvorí frontendovú časť aplikácie, ktorá má za úlohu prezentovať dáta z backendu užívateľom.[36]

Samotný klient-server je model, na ktorom je založená väčšina webových služieb. Poskytuje minimalizáciu času stráveného vývojom aplikácie vzhľadom na skutočnosť, že rozdeľuje funkcionalitu na dva celky. Klient je v modeli časťou, ktorá vysiela požiadavky a server tieto požiadavky napĺňa. Vo väčšine aplikácií využívajúcich model klient-server prebieha spracovanie dát na strane servera a výsledky sú zasielané klientovi.[36]

Niektoré spôsoby, ktorými medzi sebou klient a server komunikujú sú napríklad:[36]

- **FTP** - File Transfer Protocol
- **SMTP** - Simple Mail Transfer Protocol
- **HTTP** - Hypertext Transfer Protocol

Klient-server model sa dá rozdeliť do nasledujúcich kategórii:[36]

- **2-vrstový**, predstavujúci architektúru, ktorá pozostáva len z DB servera a klientského zariadenia. Užívateľ spúšta aplikáciu na svojom zariadení a prostredníctvom siete sa pripája k DB.
- **3-vrstvový**, obsahuje mimo klientského zariadenia a DB servera taktiež aplikačný server.
- **N-vrstvový**, je forma 3-vrstového, ale obsahuje mnoho aplikačných serverov.

Výhody klient-server modelu:[36]

- Rozdeľuje výpočetné nároky aplikácií na viacero zariadení.
- Umožňuje ľahšiu výmenu prostriedkov od klienta na server.
- Znižuje duplikáciu dát pomocou ukladania dát na strane serveu miesto klienta.

Primárnou nevýhodou klient-server modelu je bezpečnosť. Klientský operačný systém je ľahko prístupný zo strany servera a táto skutočnosť vystavuje systém klienta mnohým problémom. Výmena správ medzi klientom a serverom je taktiež vystavená mnohým bezpečnostným výzvam ako napr. phishing prihlasovacích údajov, alebo MTM[1]. Server môže taktiež byť cieľom DOS[2] útoku.[36]

Výsledná aplikácia v kontexte klient-server modelu vyžíva HTTP na prenos informácií medzi klientom a servom. Realizuje 3 vrstvový klient-server model, v ktorom existuje databázový server, aplikačný server a klient v podobe frontendu.

### 2.1.2 Komunikácia

Ako bolo uvedené v popise klient-server architektúry, HTTP je jedným z možných nástrojov komunikácie medzi klientom a serverom. Využitie tohto protokolu prebieha na základe žiadostí[3] a odpovedí[4]. Je to primárny spôsob prenosu informácií na webe.

Štruktúra HTTP žiadostí a odpovedí má danú stavbu. Je rozdelená na hlavičku[5] a telo[6]. Hlavička obsahuje technické náležitosti ohľadne predávanej informácie. Telo samotné môže, ale nemusí obsahovať rôznorodé informácie. Dajú sa posielať v rôznych formách napr.:

- **HTML** je jazyk určený na vytváranie webových stránok.
- **JSON** je formát určený na ukladanie a prenos dát.
- **CSS** je rozšírenie HTML určené na štylizáciu dokumentov.

Komunikácia prostredníctvom HTTP prebieha v nasledujúcich krokoch:

1. Klient zašle HTTP žiadosť na web.
2. Webový server prijme žiadosť.
3. Server spracuje žiadosť.
4. Server vráti HTTP odpoveď.
5. Klient príjme odpoveď.

Existuje viacero typov HTTP žiadostí, najpoužívanejšie sú ekvivalenty CRUD[7] operácií a tými sú:

- **GET**
- **POST**
- **PUT**
- **PATCH**

[1] Man in the middle.
[2] Denial of service.
[3] Requests.
[4] Responses.
[5] Header.
[6] Body.
[7] Create, read, update and delete.

- **DELETE**

**REST**, Representational state transfer, je nadstavba nad HTTP[37]. Aplikácia komunikuje medzi serverom a klientom výhradne prostredníctvom RESTového rozhrania. Dáta sú prenášané v tele HTTP žiadostí a odpovedí vo formáte JSON. Základom RESTu je statelessness, čo znamená, že server nemá potrebu vedieť nič o stave klienta. Dokáže spracovávať požiadavky aj bez potreby poznať predchádzajúce príchodzie požiadavky. Veľkou výhodou pri vývoji aplikácií je rozdelenie potrieb klienta a servera. Kód klienta a kód servera sa môžu meniť bez potrieb meniť ten druhý a bez ovplyvnenia funkcionality opačnej strany.

### 2.1.3 Backend-frontend architektúra

Aplikácia je rozdelená na dve samostatné aplikácie:

- **Frontend** tvorí časť aplikácie s ktorou priamo pracuje užívateľ. Tvorí ju všetko čo sa užívateľovi zobrazuje a s čím užívateľ môže interagovať. Označuje sa aj ako klientská časť aplikácie. Dostupné technológie na vývoj frontendu sú napríklad:
  - **HTML**
  - **CSS**
  - **JavaScript** je skriptovací programovací jazyk používaný na implementáciu komplexných funkcií na webových stránkach.
  - **TypeScript** je silne otypovaná nadstavba JavaScriptu, t. j. obohacuje JavaScript o OOP paradigmy.
- **Backend** potom naopak od frontendu sa tvorí časť aplikácie, ktorú užívateľ "nevidí". Vo svojej podstate, všetko čo sa deje predtým než sa užívateľovi zobrazí obsah frontendu, je backend. Jedná sa o rôznorodnú výpočentnú logiku a podobne. Technológie na vývoj backendu sú napríklad:
  - **JavaScript**
  - **Java**
  - **C++**
  - **C#**

### 2.1.4 Viacvrstvové aplikácie

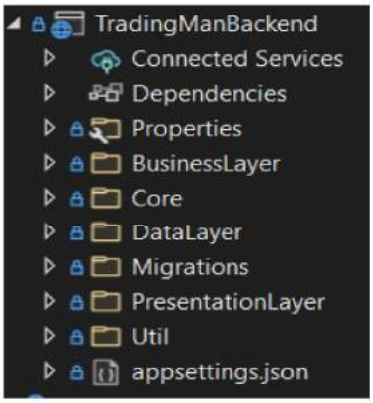
Viacvrstvová architektúra je typ architektúry, v ktorej sú jednotlivé logické časti aplikácie rozdelené do individuálnych vrstiev. Najčastejším príkladom tejto architektúry je trojvrstvová architekúra, ktorá aplikáciu rozdeľuje na nasledujúce vrstvy:

- **Prezentačná vrstva** predstavuje komunikačné rozhranie aplikácie.
- **Biznis vrstva** predstavuje výpočetnú logiku aplikácie.
- **Dátová vrstva** predstavuje modely a perzistenciu aplikácie.

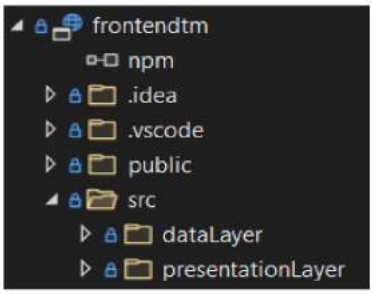
Ako už bolo uvedené, vzhľadom na využitie modelu klient-server sú frontend a backend samostatné aplikácie, a tým pádom implementujú rôzne modely viacvrstvových aplikácií na základe potrieb jednotlivých častí.

Backend podlieha relaxovanej trojvrstvovej architektúre 2.1. Jednotlivé časti sú rozdelné do už predstavených troch vrstiev:

**Obr. 2.1** Architektúra backendu




**Obr. 2.2** Architektúra frontendu




- **Prezentačná vrstva** je vo výslednej aplikácii tvorená controllermi/endpointami ktoré sú volateľné prostredníctvom RESTových volaní. Konkrétne sa v implementácií jedná o HTTP POST a GET žiadosti, ktoré spĺňajú požiadavky frontendu. Triedy tejto vrstvy komunikujú s triedami biznis vrstvy, vo výnimočných prípadoch, kedy nie je potreba využívať biznis vrstvu komunikujú priamo s triedami dátovej vrstvy.
- **Biznis vrsta** obsahuje aplikačnú logiku, potrebnú pre spracovanie požiadaviek.
- **Dátová vrstva** obsahuje modely, prístup k databáze, prístup k externým zdrojom dát, t. j. volania externých API a modely na prenos dát(DTO).

Skutočnosť, že na frontende nie je potrebné realizovať výpočty vyplýva z využitia modelu klient-server. Frontendová aplikácia, preto len komunikuje s backendom a intrepretuje dáta, ktoré jej backend predáva. Nie je potreba vytvávať trojvrstvovú štruktúru a aplikácia podlieha dvojvrtsvovej architekúre v podobe:

- **Dátová vrstva** obsahuje API volania na backenmd a modely na prenos dát.
- **Prezentačná vrstva** obsahuje všetky komponenty, ktoré sa zobrazujú užívateľovi.

### 2.1.5 Technológie frontendu

Nasledujúca časť pojednáva o technológiách použitých výhradne na vývoj frontendu.

Na výber špecifických technológií implementovaných vo frontendovej časti aplikácie nebola žiaden priama požiadavka plynúca z predchádzajúcej analýzy.

#### 2.1.5.1 Typescript

TypeScript je silne otypovaná nadstavba JavaScriptu, ktorá pridáva syntax pre typy, čím do JavaScriptu vnáša koncepty OOP.[38] Primárnymi dôvodmi využitia TypeScriptu sú rozšírenosť JavaScriptu v aplikáciách realizujúcich užívateľské rozhranie, komplikácie pri implementovaní čisto JavaScriptového frontendu a predošlé skúsenosti s TypeScriptom.

#### 2.1.5.2 Užívateľské rozhranie

Na tvorbu užívateľského rozhrania sú používané nasledujúce technológie:

- **React** je knižnica určená pre JavaScript zameraná na tvorbu užívateľského rozhrania. Tvorba aplikácií v Reacte je založená na vytváraní opakovane použiteľných komponent. Komponenty sú triedy alebo funkcie, ktoré vracajú HTML kód. Komponenty v podobe funkcií sú prirovnatelné k funkciám v JavaScripte a komponenty v podobe tried sú základným špecifikom aplikácií napísaných v Reacte. Prístup k dátam je realizovaný prostrednícvtvom props, ktoré predávajú dáta z nadradených komponent a state, ktoré zahŕňajú dáta ukladané v rámci daného komponentu.[39]
- **Tailwind CSS** je rámec uľahčujúci prácu s CSS.[40] V aplikácii je používaný na štylizovanie jednotlivých zobrazovaných prvkov.
- **Ignite UI** je knižnica obsahujúca nástroje na vyobrazenie užívateľského rozhrania.[41] V aplikácii je využívaná primárne na zobrazovanie grafov.

### 2.1.6 Technológie backendu

Nasledujúca časť pojednáva o technológiách použitých výhradne na vývoj backendu.

Výber špecifických technológií implementovaných v backendovej časti aplikácie podlieha nefuknčným požiadavkám pre komunikáciu s brokerom a zdrojom dát. Prierez podporovaných technológií, v ktorých je backend možné vyvíjať je preto NodeJs, Python a C#. Vzhľadom na preferencie a predošlé skúsenosti bola zvolená implementácia v programovacom jazyku C#.

#### 2.1.6.1 ASP.NET Core 6.0

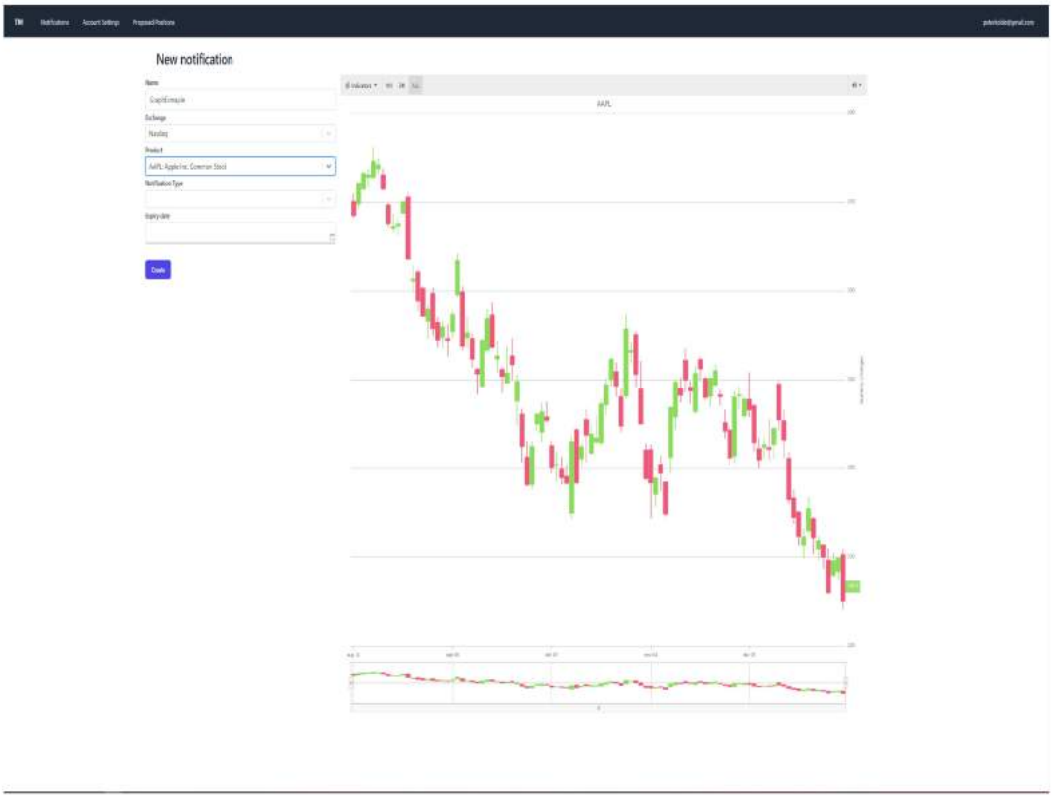
Backendová aplikácia je cielená na použitie najnovšieho rámca .NET v dobe vývoja a tou je verzia 6.0,[8] čo je open-source rámec pre vývoj softvéra vyvinutý spoločnosťou Microsoft. Poskytuje smernice ako vytvárať veľké množstvo rôznorodých aplikácií v rôznych kategóriach ako web, hry, mobilné zariadenia a IoT. Tento rámec obsahuje dva základné komponenty a to:

- **Framework Class Library** je sada knižníc, ktoré poskytujú implementáciu mnohých obecných ale aj aplikačne špecifických typov, algoritmov a úžitkovej funkcionality. Napríklad je možné uviesť datové štruktúry ako DateTime, Uri, Array, alebo rozhranie HttpClientAPI pre výmenu HTTP požiadaviek.[42]
- **Common Language Runtime** je prostredie ktoré spúšťa kód a poskytuje služby pre zjednodušenie procesu vývoja. Mimo iné poskytuje napríklad možnosť instanciám tried písanom v jednom programovacom jazyku volať metódy tried písaných v iných jazykoch.[43]

ASP.NET Core je open-source verzia ASP.NET, ktorá je spustiteľná v prostredí Windows, Linux, macOs a Docker. Pomocou tohto rámca je možné vytvárať webové aplikácie, IoT aplikácie a backendové aplikácie pre mobilné zariadenia. Je to zároveň súčasť rámca .NET určená pre tvorbu webových aplikácií a implmentuje vzor MVC.[44] Backendová časť výslednej aplikácie

[8] 8.11.2022 vyšla verzia 7.0, aplikácia však bola navrhnutá začiatkom roku 2022.

**Obr. 2.3** Zobrazenie grafu



**Výpis kódu 2.1** Ilustrácia registrovania služby do kontajneru.

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddScoped<NotificationsLogic>();
var app = builder.Build();
```

**Výpis kódu 2.2** Ilustrácia DI v controllery

```
public NotificationsController(NotificationsLogic notificationsLogic,
                               ILogger<NotificationsController> logger)
{
        _notificationsLogic = notificationsLogic;
        _logger = logger;
}
```

je vytvorená za pomoci využitia ASP.NET Core Web API a upraveného vzoru MVC. Aplikácia obsahuje modely v pôvodnom kontexte, ale contollery sú zodpovedné iba za komunikáciu s klientom, ktorému sú spätne predávané dáta prostredníctvom Restových HTTP odpovedí, vo formáte JSON. Klient si následne tieto dáta interpretuje podľa potreby. Všetka výpočetná logika je presunutá z contollerov do business vrstvy backendovej aplikácie.

#### 2.1.6.2 Dependency injection

Vkladanie závislostí, je návrhový vzor, ktorý obsahuje techniku pre dosiahnutie IoC[9] medzi jednotlivými triedami a ich závislosťami. Výsledok je tiež kód, ktorý sa dá ľahšie spravovať a testovať.[45] V rámci aplikácie je využívaná primárne technika vkladania závislostí do konštruktorov jednotlivých tried za použitia služieb registrovaných v kontajnery zastrešujúcom služby, ktorým je v ASP.NET Core 6.0 IServiceProvider.[46]

Na základe životnosti je možné registrovať následujúce typy závislostí:[47]

- **Transient** sú vytvárané vždy keď sú potrebné
- **Scoped** sú vytvárané vo webových aplikáciách raz za požiadavku klienta a sú zničené na konci žiadosti.
- **Singleton** sú vytvárané prvýkrát keď sú požadované, alebo priamo vývojárom pri poskytovaní implementácie priamo do kontajneru.

#### 2.1.6.3 Entity Framework

EF je primárny nástroj microsoftu ako komunikovať medzi .NET aplikáciami a relačnými databázami. EF je open-source rámec a využíva objektovo relačné mapovanie(ORM), čo je nástroj, ktorý uľahčuje mapovať medzi objektovým a relačným databázovým modelom.[48]

Najväčšou výhodou ORM je práve uľahčenie práce s databázovým modelom, pretože vývojár pracuje hľavne s objektovým modelom a odpadá mu nutnosť zaoberať sa s prácou s databázou, vytváraním dotazov a pod.

### 2.1.7 Microsoft Sql Server

Microsoft Sql Server je RDBMS, čo je databázový server, ktorý spravuje databázy, komunikáciu s klientmi, vstupy a výstupy dát a ich integritu(RDBMS). Je založený na relačnom modely

[9] Inversion of Control

databázy, relačné modely sú intuitívne a jasné spôsoby reprezentácie dát v tabulkách. V relačných databázách sú v každom riadku tabulky záznamy s jedinečným identifikátorom, kľúčom. Stĺpce tabuliek reprezentujú atribúty dát a každý záznam zväčša obsahuje hodnotu pre každý atribút, čo uľahčuje vytvárať vzťahy medzi jednotlivými dátami.[49] Microsoft Sql Server Podporuje ANSI SQL. V práci je použitý na zabezpečenie perzistencie.

### 2.1.8 Ostatné

Backend využíva nasledovné spomenutiahodné balíčky:

- **Telegram.Bot** je .Netový klient pre Telegram Bot API. Telegram Bot API je rozhranie založené na HTPP. Použitý klient toto rozhranie zaobaluje a poskytuje tak jednoduhšiu implementáciu funkcionality spojenej s využívaním služiem Telegram Bota.
- **Alpca.Markets** je .NET SDK pre Alpaca Markets API,
- **Quartz pre .NET** je open-source systém pre rozvrhovanie úloh. V implementácii je použitý pri vytváraní cron jobov, ktoré sa spúšťajú aby vyhodnotili podmienky upozornení.

## 2.2 Testovanie

Testovanie aplikácie pozostáva z dvoch častí. Backend bol testovaný osobitne za pomoci jednotkových a integračných testov. Frontend bol testovaný zároveň s backendom prostredníctvom systémových testov. Výsledky testov boli využité k odhaleniu chýb, ktoré počas vývoja vznikali. Pomohli odhaliť napíklad nesprávne porovnávanie hodnôt vo vyhodnocovaní podmienok upozornení, automatické odstraňovanie nastavení upozornení v prípade, že boli naplnené, alebo problémy spojené s obnovením okna v prehliadači.

### 2.2.1 Testovanie backendu

V rámci testov backendu sú využívané nasledujúce technológie:

- **Microsoft.EntityFramework.InMemory** je poskytovateľ databáze EF v pamäti a umožňuje tak testovať databázové modely, bez nutnosti vytvárania testovaciej databáze.
- **Moq** slúži na mockovanie služieb.
- **Microsoft.VisualStudio.TestTools.UnitTesting** poskytuje infraštuktúru pre zjednodušenie behu jednotkových testov.

#### 2.2.1.1 Jednotkové testy

Jednotkové testy v rámci aplikácie testujú súčasti biznis vrstvy, ktoré sú relatívne ľahko testovatelné. Realizované testy sú zobrazené na obrázku 2.4.

Pred spustením každého testu je volaná metóda Init z obrázku. Táto metóda má za úlohu pripraviť prostredie pre beh testov.

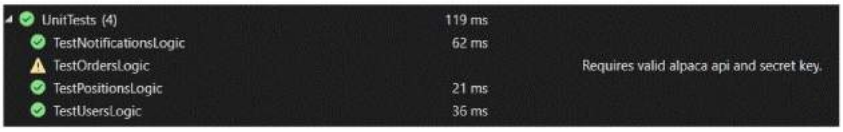
Jednotlivé jednotkové testy následne testujú všetky verejne dostupné metódy vybraných testovaných súčastí biznis vrstvy. Časti, ktoré testované nie sú, sú testované v rámci integračných testov.

#### 2.2.1.2 Integračné testy

Integračné testy zobrazené na obrázku 2.5 testujú kombináciu modulov na backende a priechod celou aplikácou s výnimkou controllerov. Podobne ako jednotkové testy využívajú modifikovanú metódu init, ktorá je ale tentokrát označená atribútom ClassInitialize, čím sa zaručí inicializácia iba raz na začiatku testovania. Testy sú rozdelené na dve časti

- **IntegrationTestStockDataLogic**, ktorá testuje obstarávanie trhových dát, nekontroluje však správnosť obstaraných dát.
- **IntegrationTests**, ktorá vytvára scenáre na základe rôznych druhov nastavení a testuje vyhodnocovanie nastavení.

**Obr. 2.4** Jednotkové testy



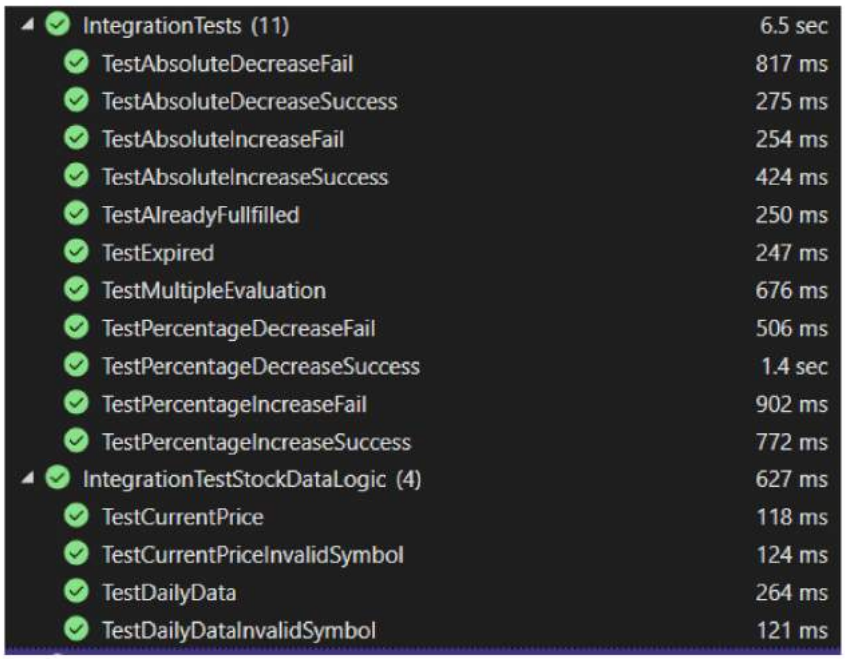
**Výpis kódu 2.3** Metóda init.

```
// Sets up the class - is ran before each test
[TestInitialize]
public void Init()
{
  // Create mock instance of DB
  var options = new DbContextOptionsBuilder<DatabaseContext>()
  .UseInMemoryDatabase(databaseName: "testDb").Options;
  _context = new DatabaseContext(options);

  // Prepare loggers
  var uMocklLogger = new Mock<ILogger<UsersLogic>>();
  var sMockLogger = new Mock<ILogger<StockDataLogic>>();
  var pMockLogger = new Mock<ILogger<PositionLogic>>();
  var nMockLogger = new Mock<ILogger<NotificationsLogic>>();

  // Initialize services
  _usersLogic = new UsersLogic(_context, uMocklLogger.Object);
  _stockDataLogic = new StockDataLogic(sMockLogger.Object);
  _positionLogic = new PositionLogic(_context, pMockLogger.Object);
  _notificationsLogic = new NotificationsLogic(_context,
                                               nMockLogger.Object);
}
```

**Obr. 2.5** Integračné testy



### 2.2.2 Manuálne testy

Manuálne testy boli realizované neustále počas vývoja a taktiež po dokončení vývoja v početných iteráciách. Boli realizované v nasledujúcich scenároch:

- **Registrácia užívateľského účtu.**
- **Prihlásenie užívateľského účtu.**
- **Odhlásenie užívateľkshého účtu.**
- **Zmena hesla užívateľského účtu.**
- **Vyplnenie nastavení užívateľského účtu.**
- **Zmena nastavení užívateľského účtu.**
- **Vytvorenie nového nastavenia upozornenia** vo všetkých kombináciách s rôznymi expiračnými dobami.
- **Obdržanie upozornenia v prípade naplnenia nastavení.**
- **Expirácia upozornenia.**
- **Náhľad na nastavenie upozornenia.**
- **Prehľad nastavení upozornení.**
- **Odstránenie nastavenia upozornenia.**
- **Prehľad navrhovaných pozícií.**
- **Náhľad na navrhovanú pozíciu.**
- **Vyplnenie a prijatie navrhovanej pozície** s využitím všetkých dostupných nástrojov(stop loss, take profit) a papierového účta.
- **Odmietnutie navrhovanej pozície.**

# Kapitola 3
# Záver

Primárnym cieľom tejto bakalárskej práce bolo implementovať obchodnú platformu zasielajúcu upozornenia a vytvoriť analýzu elementov, ktoré boli pri implementácii použité.

## Analytická časť

V rámci analytickej časti prebehol prieskum existujúcich riešení, prieskum a porovnanie riešení poskytujúcich trhové dáta, prieskum a porovnanie možností zasielania upozornení, prieskum a porovnanie brokerov poskytujúcich zasielanie príkazov z užívateľskej aplikácie a následne bol vypracovný model spracovania trhových dát. Na základe týchto poznatkov bolo možné vybrať jednotlivé elementy potrebné k implementácii výslednej platfotmi v druhej časti práce. Boli sformulované funkčné požiadavky, nefukčné požiadavky a prípady použitia na základe, ktorých bola platforma následne vyvinutá. Vrámci analýzi taktiež vyplynula potreba implementovať platformu ako webovú aplikáciu.

## Praktická časť

Praktická časť obsahuje popis použitých technológii pri implementácii výslednej aplikácie. Popisuje rozdelenie aplikácie na dva celky, korými sú frondend a backend, implementovanú architektúru na jednotlivých častiach, technológie použité na komunikáciu medzi nimi a následne technológie špecifické pre jednotlivé časti. V závere tejto časti je popis testovania jednotlivých častí výslednej aplikácie prostredníctvom jednotkových, integračných a manuálnych testov.

## Naplnenie cieľov práce

Mimo primárneho cieľu práca obsahuje aj niekoľko čiastočných cieľov, Práca si kládla za cieľ preskúmať, akým spôsobom možno získať trhové dáta zo svetových búrz v podobe informácií o akciách, opciách, kryptomenách a iných finančných inštrumentoch. Následne mala zá úlohu porovnať dostupné riešenia na základe relevantných ukazovaľov a taktiež mala za cieľ porovnať na základe rýchlosti a spoľahlivosti rôzne spôsoby zasielania upozornení na mobilné/desktopové zariadenia užívateľov. V neposlednom rade mala práca za cieľ preskúmať, ktorí brokeri umožňujú zadávať obchodné príkazy z užívateľskej aplikácie a porovnať ich riešenia primárne na základe kvality, spoľahlivosti a ceny za využívanie služieb. Tieto ciele boli naplnené v rámci analytickej časti práce, jednotlivé položky boli porovnávané na základe atribútov uvedených v cieľoch práce, ale aj iných dostupných a relevantných atribútov. Ďalším stanoveným cieľom práce bolo na základe predchádzajúcich poznatkov zvoliť vhodný zdroj dát, brokera a spôsob zasielania

správ. Ako zdroj dát bol zvolený Alpha Vantage na základe kvality a ceny poskytovaných dát. Ako broker bola zvolená platfoma Alpaca, vzhľadom na kvalitu rozhrania, stav dokumentácie, ceny za poskytované služby a skutočnosť, že sa jedná o brokera zameraného primárne na poskytovanie obchodov prostredníctvom API. Spôsob zasielania správ bol vybraný prostredníctvom e-mailu a aplikácie Telegram zároveň. Tento prístup kombinuje spoľahlivosť doručovania správ prostredníctvom e-mailu a rýchlosť doručovania správ prostredníctvom aplikácie zameranej na IM. Záverečnými cieľmi bolo na základe vybraných elementov navrhnúť užívateľskú aplikáciu, ktorá bude analyzovať dáta o vybranom inštrumente, pri splnení zadaných pravidiel zašle užívateľovi upozornenie a pripraví vhodný obchodný príkaz podľa vybraných nastavení, ktorý bude užívateľ môcť prijať, modifikovať, alebo odstrániť. Tento cieľ bol taktiež naplnený v praktickej časti práce, kde sa za pomoci vybraných elementov podarilo implementovať webovú aplikáciu spĺňajúcu dané požiadavky.

# Dodatok A

# Nějaká příloha

Sem přijde to, co nepatří do hlavní části.

# Bibliografia


1. BLYSTONE, Dan. *The Best Online Brokers Of April 2022* [Forbes Advisor] [online]. 2021-01-01. [cit. 2022-04-08]. Dostupné z : https://www.forbes.com/advisor/investing/best-online-brokers/. Section: Investing.
2. *Online Stock Trading Platform — TD Ameritrade* [online]. [cit. 2022-04-11]. Dostupné z : https://www.tdameritrade.com/tools-and-platforms/investing-stock-trading-platforms/online-stock-trading-features.html.
3. *Mobile Stock Trading App — TD Ameritrade* [online]. [cit. 2022-04-11]. Dostupné z : https://www.tdameritrade.com/tools-and-platforms/investing-stock-trading-platforms/td-ameritrade-mobile-stock-trading-app.html.
4. *4 ways to use alerts when investing — Fidelity* [online]. [cit. 2022-04-12]. Dostupné z : https://www.fidelity.com/viewpoints/active-investor/4-ways-to-use-alerts.
5. *Options Trading, Futures & Stock Trading Brokerage* [online]. [cit. 2022-04-11]. Dostupné z : https://tastyworks.com/.
6. *Crypto, Stocks & Beyond! The power of social investing — eToro* [online]. [cit. 2022-04-17]. Dostupné z : https://www.etoro.com/.
7. *Real Time Alerts — Trading Alerts — Plus500* [online]. [cit. 2022-04-12]. Dostupné z : https://www.plus500.com/en-CZ/Help/Alerts.
8. *Stock Price Alerts* [Robinhood] [online]. [cit. 2022-04-12]. Dostupné z : https://robinhood.com/support/articles/4ODhlAR95LjZZODXIlD2VY/stock-price-alerts/.
9. *Seeking Alpha — Stock Market Analysis & Tools for Investors* [online]. [cit. 2022-04-17]. Dostupné z : https://seekingalpha.com.
10. *The Ins and Outs of Intraday Trading* [Investopedia] [online]. [cit. 2022-04-18]. Dostupné z : https://www.investopedia.com/terms/i/intraday.asp.
11. *Using Market Data* [Robinhood] [online]. [cit. 2022-04-18]. Dostupné z : https://robinhood.com/support/articles/360025308191/using-market-data/.
12. *CRSP - The Center for Research in Security Prices* — [online]. [cit. 2022-04-14]. Dostupné z : https://www.crsp.org/.
13. *API Documentation — Alpha Vantage* [online]. [cit. 2022-04-14]. Dostupné z : https://www.alphavantage.co/documentation/.
14. *API Hub - Free Public & Open Rest APIs* [RapidAPI] [online]. [cit. 2022-04-14]. Dostupné z : https://rapidapi.com/hub.
15. *YahooFinance Stocks API Documentation (integraatio)* [RapidAPI] [online]. [cit. 2022-04-14]. Dostupné z : https://rapidapi.com/integraatio/api/yahoofinance-stocks1/.

16. *Quandl* [online]. [cit. 2022-04-14]. Dostupné z : https://www.quandl.com.
17. *IEX Cloud API — IEX Cloud* [online]. [cit. 2022-04-14]. Dostupné z : https://iexcloud.io/docs/api/.
18. *Google Finance APIs and Tools - Skupiny Google* [online]. [cit. 2022-04-19]. Dostupné z : https://groups.google.com/g/google-finance-apis.
19. *IB API — Interactive Brokers Central Europe* [online]. [cit. 2022-04-15]. Dostupné z : https://www.interactivebrokers.hu/en/index.php?f=40022&gclid=Cj0KCQjwr-SSBhC9ARIsANhzu1569o9W-xOKx_lnSsw0od-ulsn49QGviuMGg_ujp19L7HtgmaggucEaAsTpEALw_wcB.
20. *TWS API v9.72+: Trader Workstation API* [online]. [cit. 2022-04-15]. Dostupné z : https://interactivebrokers.github.io/tws-api/.
21. *¡eToro¿ developer portal* [online]. [cit. 2022-04-15]. Dostupné z : https://api-portal.etoro.com/docs/services.
22. *Alpaca - Developer-First API for Crypto and Stocks* [Alpaca] [online]. [cit. 2022-04-15]. Dostupné z : https://alpaca.markets.
23. *Tradier API Documentation — Tradier* [online]. [cit. 2022-04-15]. Dostupné z : https://documentation.tradier.com/.
24. *Futures and Options Trading, Charting, Analytics, and Risk* [CTS] [online]. [cit. 2022-04-15]. Dostupné z : https://www.ctsfutures.com/.
25. *3. Používání mobilního telefonu a internetu na mobilním telefonu* [3. Používání mobilního telefonu a internetu na mobilním telefonu] [online]. [cit. 2022-04-17]. Dostupné z : https://www.czso.cz/csu/czso/3-pouzivani-internetu-jednotlivci.
26. MANDIOHLINGER. *Appendix D: Create the SMTP Server - BizTalk Server* [online]. [cit. 2022-04-17]. Dostupné z : https://docs.microsoft.com/en-us/biztalk/install-and-config-guides/appendix-d-create-the-smtp-server.
27. *Send email from a printer, scanner, or app - Google Workspace Admin Help* [online]. [cit. 2022-04-16]. Dostupné z : https://support.google.com/a/answer/176600?hl=en.
28. *Email Delivery, API, Marketing Service* [SendGrid] [online]. [cit. 2022-04-16]. Dostupné z : https://sendgrid.com/.
29. *Mailgun Technologies* [online]. [cit. 2022-04-16]. Dostupné z : https://www.mailgun.com.
30. *Amazon Simple Email Service — Cloud Email Service — Amazon Web Services* [Amazon Web Services, Inc.] [online]. [cit. 2022-04-17]. Dostupné z : https://aws.amazon.com/ses/.
31. *Send and Receive Messages with the Telegram API - Will Kelly* [Send and Receive Messages with the Telegram API - Will Kelly] [online]. [cit. 2022-04-15]. Dostupné z : https://wk0.dev/posts/send-and-receive-messages-with-the-telegram-api.
32. *Send API - Messenger Platform - Documentation* [Facebook for Developers] [online]. [cit. 2022-04-17]. Dostupné z : https://developers.facebook.com/docs/messenger-platform/reference/send-api/.
33. *What Is After-Hours Trading, and Can You Trade at This Time?* [Investopedia] [online]. [cit. 2022-07-16]. Dostupné z : https://www.investopedia.com/ask/answers/after-hours-trading-am-i-able-to-trade-at-this-time/.
34. *Best Time(s) of Day, Week, and Month to Trade Stocks* [Investopedia] [online]. [cit. 2022-07-16]. Dostupné z : https://www.investopedia.com/day-trading/best-time-day-week-month-trade-stocks/.
35. GOPALAKRISHNAN, Jayanthi. Stocks &Commodities V.18:2 (16-22):Pivot Points by Jayanthi Gopalakrishnan. [N.d.], vol. 18.

36. SULYMAN, Shakirat. Client-Server Model. *IOSR Journal of Computer Engineering*. 2014, roč. 16, s. 57–71. Dostupné z DOI: 10.9790/0661-16195771.
37. *What is REST?* [Codecademy] [online]. [cit. 2022-06-19]. Dostupné z : https://www.codecademy.com/article/what-is-rest.
38. *JavaScript With Syntax For Types.* [online]. [cit. 2022-06-20]. Dostupné z : https://www.typescriptlang.org/.
39. *React – A JavaScript library for building user interfaces* [online]. [cit. 2022-06-19]. Dostupné z : https://reactjs.org/.
40. *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.* [online]. [cit. 2022-06-20]. Dostupné z : https://tailwindcss.com/.
41. *JavaScript UI Design Software & User Interface Components — Infragistics* [online]. [cit. 2022-06-20]. Dostupné z : https://www.infragistics.com/products/ignite-ui.
42. RICHLANDER. *Framework Libraries* [online]. [cit. 2022-06-19]. Dostupné z : https://learn.microsoft.com/en-us/dotnet/standard/framework-libraries.
43. GEWARREN. *Common Language Runtime (CLR) overview - .NET* [online]. [cit. 2022-06-19]. Dostupné z : https://learn.microsoft.com/en-us/dotnet/standard/clr.
44. *ASP.NET — Open-source web framework for .NET* [Microsoft] [online]. [cit. 2022-06-19]. Dostupné z : https://dotnet.microsoft.com/en-us/apps/aspnet.
45. RICK-ANDERSON. *Dependency injection in ASP.NET Core* [online]. [cit. 2022-06-19]. Dostupné z : https://learn.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection.
46. ARDALIS. *Dependency injection into controllers in ASP.NET Core* [online]. [cit. 2022-06-19]. Dostupné z : https://learn.microsoft.com/en-us/aspnet/core/mvc/controllers/dependency-injection.
47. IEVANGELIST. *Dependency injection - .NET* [online]. [cit. 2022-11-17]. Dostupné z : https://learn.microsoft.com/en-us/dotnet/core/extensions/dependency-injection.
48. WADEPICKETT. *Entity Framework documentation* [online]. [cit. 2022-11-17]. Dostupné z : https://learn.microsoft.com/en-us/ef/.
49. *What is a relational database?* [online]. [cit. 2023-12-04]. Dostupné z : https://www.oracle.com/database/what-is-a-relational-database/.

# Obsah přiloženého média

- `readme.txt` ..... stručný popis obsahu média
- `src`
  - `impl` ..... zdrojové kódy implementácie
  - `thesis` ..... zdrojová forma práce vo formáte LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce vo formáte PDF