

**Obchodní akademie, Vyšší odborná škola a Jazyková škola  
s právem státní jazykové zkoušky Uherské Hradiště**



## **MATURITNÍ PROJEKT**

# **MOBILNÍ APLIKACE PRO SIMULOVANÉ OBCHODOVÁNÍ S AKCIEMI**



Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní jazykové  
zkoušky Uherské Hradiště

Vnitřní předpis OA, VOŠ a JŠ, čj. 025/ORG/2020

### ZADÁNÍ MATURITNÍ PRÁCE INFORMAČNÍ TECHNOLOGIE

Jméno žáka:	Dominik Borek
Téma maturitní práce:	Mobilní aplikace pro obchodování s akcemi
Vedoucí práce:	Mgr. Jaroslav Hodl
Oponent práce:	Ing. Bc. Martin Šimůnek
Způsob zpracování:	<p>Žák vytvoří mobilní aplikaci pro simulaci burzovních operací (nákup a prodej akcií) za virtuální měnu, kterou bude mít na svém portfoliu po registraci.</p> <p>Žák v rámci svého projektu prozkoumá možnosti propojení své aplikace s reálnými daty (např. Stock API Alpha Vantage), popíše ji srozumitelně v dokumentaci projektu, a poté implementuje do svého projektu.</p> <p><b>Mezi základní funkce aplikace bude patřit:</b></p> <ul style="list-style-type: none"><li>• Vytvoření fiktivního účtu v aplikaci spolu se startovní částkou</li><li>• Využití API pro pravidelnou aktualizaci cen akcií</li><li>• Vyhledávání firem, zobrazení ceny jejich akcií, grafické zobrazení historie cen akcií</li><li>• Správa svého portfolia - nákup, prodej akcií, zobrazení své finanční situace, historie operací</li><li>• Přidání zajímavé firmy na watchlist, správa watchlistu</li></ul> <p>Ke zpracování projektu bude využit framework (např. Flutter, spolu s databází Firebase).</p> <p>Součástí dokumentace bude i zpráva z uživatelského testování - žák nechá výslednou aplikaci otestovat minimálně 2 uživateli.</p> <p>Projekt bude po celou dobu vývoje přístupná ve veřejném repozitáři na GitHubu</p>



Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní jazykové  
zkoušky Uherské Hradiště

Pokyny k odevzdání:	<p><b>Žák odevzdá práci v tištěné a elektronické podobě</b></p> <ul style="list-style-type: none"><li>• Tištěná podoba práce obsahuje uživatelskou a technickou dokumentaci.</li></ul> <p>Tištěnou podobu (v kroužkové vazbě) žák odevzdá v jedné verzi na studijní oddělení školy, místnost 311.</p> <p><b>Elektronická podoba práce obsahuje</b></p> <ul style="list-style-type: none"><li>• Dokumentaci ve formátu PDF/A</li><li>• Resumé ve formátu PDF/A</li><li>• Výsledný projekt, zdrojové soubory a potřebné knihovny pro spuštění projektu</li><li>• Prezentaci projektu</li></ul> <p><b>Elektronická podoba</b> práce se nahrává do IS školy dle pokynů vedoucího práce nebo vedení školy.</p> <p>V případě, že se jedná o projekt, na kterém pracovalo více žáků, je povinnou součástí dokumentace podrobné rozdělení činností při práci na projektu.</p>
Kritéria hodnocení:	Hodnocení se skládá z celkové kvality zpracování práce, dokumentace, z kvality prezentace při obhajobě práce, diskuse a z průběžného hodnocení žáka v rámci kontrolních dnů.
Obhajoba projektu	Obhajoba projektu se skládá ze dvou částí - prezentace projektu (včetně podpůrné elektronické prezentace) a diskuse nad řešením. Celková délka obhajoby je 20 minut, délka prezentace projektu by neměla překročit 10 minut.

14. 10. 2021

Datum

Podpis ředitele školy

Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní  
jazykové zkoušky Uherské Hradiště  
Nádražní 22, 686 01 Uherské Hradiště  
IČO: 60371731, tel.: 572 433 011

**Prohlášení:**

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího maturitní práce a ředitele školy. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že jsem na celé maturitní práci pracoval samostatně a veškeré použité zdroje jsem citoval.

V Uherském Hradišti, dne 28.3.2022

.....

podpis absolventa

## **RESUMÉ**

Tento projekt je pro uživatele, které zajímá akciový trh a měli by zájem se naučit obchodovat na něm. Uživatelé budou moci pomocí virtuálních peněz nakupovat či prodávat veřejné akcie firem bez rizik.

Největší výhodou je jednoduchost a přehlednost aplikace a rychlé se zorientování na finančním trhu.

## **RESUME**

This project is for users who are interested in the stock market and would be interested in learning to trade on it. Users will be able to use virtual money to buy or sell public shares of companies without risks.

The biggest advantage is the simplicity and clarity of the application and quick orientation in the financial market.

# OBSAH

<b>ÚVOD.....</b>	<b>8</b>
<b>1 TECHNICKÁ DOKUMENTACE.....</b>	<b>9</b>
1.1 Spuštění .....	10
1.1.1 Spuštění v Android studiu (pro vývojáře) .....	10
1.1.2 Spuštění na mobilním zařízení s operačním systémem Android.....	10
1.1.3 Spuštění pro vývojáře na platformě iOS .....	11
1.2 Použité vývojové prostředí a SDK .....	12
1.2.1 Android Studio .....	12
1.2.2 Knihovny .....	12
1.2.3 Flutter SDK .....	12
1.2.4 Dart SDK .....	12
1.3 Funkce aplikace .....	13
1.4 Popis knihoven .....	13
1.4.1 Hexcolor .....	14
1.4.2 syncfusion_flutter_charts .....	14
1.4.3 http .....	15
1.4.4 Sqflite .....	16
1.5 Technické zpracování .....	18
1.5.1 Návrh aplikace .....	18
1.5.2 Teoretické vytvoření aplikace .....	19
1.6 Flutter.....	21
1.7 Alpha Vantage .....	22
1.8 SQLite.....	23
1.8.1 Práce s SQFlite .....	24
1.9 Modelové třídy databáze.....	24
1.10 Důležité třídy .....	25

1.10.1 DatabaseHelper	25
1.10.2 APIManager	28
1.10.3 MyApp	28
1.11 Nejvíce využívané widgety.....	29
<b>2 UŽIVATELSKÁ DOKUMENTACE .....</b>	<b>30</b>
2.1 Spuštění .....	30
2.1.1 Spuštění na mobilním zařízení s operačním systémem Android.....	30
2.1.2 Spuštění na platformě iOS	30
2.2 Ovládání.....	31
2.3 Funkce .....	32
2.3.1 Portfolio	32
2.3.2 Stocks	32
2.3.3 Orders	33
2.3.4 Setting	34
2.3.5 Stock	35
2.4 Uživatelské testování.....	36
<b>ZÁVĚR.....</b>	<b>41</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>43</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>45</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>46</b>
<b>SEZNAM TABULEK .....</b>	<b>CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.</b>

## ÚVOD

Cílem maturitní práce bude vytvořit simulátor pro obchodování na veřejném akciovém trhu, kde bude možnost obchodovat s akcemi firem za virtuální peníze, tedy bez žádného rizika.

Tento projekt bude určen pro veřejnost, a to převážně pro osoby, které se budou chtít naučit pohybovat na finančním trhu, přesněji na trhu s akcemi. V aplikaci se budou moct naučit fungování akciovému trhu a orientování se na něm.

Uživatel po stažení a spuštění aplikace bude přesměrován přímo na hlavní stránku, na které uvidí, že pro začátek svého obchodování bude mít předem danou částku. Také zde bude mít watchlist jeho oblíbených akcií, které si zde sám přidá. V jednotlivých kartách uvidí vždy název akcie i s jeho obchodní zkratkou. Po rozkliknutí této karty se uživateli zobrazí graf akcie s možností nákupu, či jejího prodeje a základní informace o dané akci. Dále v aplikaci bude možnost si najít akcii, kterou bude uživatel chtít. Bude si zde moct zobrazovat otevřené obchody, ale také historii již provedených a ukončených obchodů.

Aplikace bude mít několik výhod oproti konkurenci, a to převážně v její jednoduchosti. Aplikace bude navržena tak, aby se uživatel bez problémů zorientoval a pochopil hlavní fungování aplikace. Konkurenční produkty bývají většinou nepřehledné, bývá tam hodně nedůležitých faktorů a nadbytečné informace, které mohou uživatele, kteří jsou na trhu poprvé, rychle zmást, nebo dokonce odradit. Jako konkurenční aplikace mohou být považovány <https://play.google.com/store/apps/details?id=de.elfsoft.bestbrokers> nebo <https://play.google.com/store/apps/details?id=com.stocktrainer>

Hlavní technologií pro vytvoření mobilní aplikace bude framework Flutter, který je psaný pomocí programovacího jazyka Dart. Dále bude využívat nejmodernější API pro akciový trh, tedy Alphavantage a databázi, ve které budou uloženy veškeré informace ohledně obchodů, jako provedené obchody, vstupní ceny akcií při nákupu, oblíbené akcie, ale také peníze.

V následujících stránkách technické dokumentace si budete moct prohlédnout informace o samotné aplikaci, použitém frameworku, API a databázi. Dále pak v uživatelské dokumentaci budou informace pro uživatele, jako spuštění a popis stránek. Na konci bude část věnována testování aplikace jednotlivými respondenty.



# 1 TECHNICKÁ DOKUMENTACE

V části technické dokumentace popíši veškeré důležité informace o projektu. Od jeho spuštění ve vývojovém prostředí, či zařízení, přes funkce aplikace, využívané knihovny až po technické zpracování aplikace. V této části budu také popisovat používaný framework, tedy Flutter. API Alpha Vantage a využívanou SQLite databázi.

## 1.1 Spuštění

### 1.1.1 Spuštění v Android studiu (pro vývojáře)

1. Projekt si stáhnete z Githubu: [https://github.com/Hellman666/stock\\_app](https://github.com/Hellman666/stock_app)
2. Otevřete si Android studio
  - a. Pokud ho nemáte stažené, tak si ho zdarma můžete stáhnout a nainstalovat z odkazu zde: <https://developer.android.com/studio>
3. Po otevření Android Studia budete potřebovat:
  - a. Pluginy pro Android Studio Flutter a Dart
    - Stáhnete si je přímo v Android Studiu v levém horním rohu zmáčknete na File -> Settings -> Plugins -> vyhledáte Dart a Flutter a dáte install
  - b. Stáhněte si SDK soubory frameworku Flutter a programovacího jazyka Dart
    - SDK pro Flutter najdete zde: <https://docs.flutter.dev/get-started/install>
    - SDK pro Dart najdete zde: <https://dart.dev/get-dart>
  - c. Nastavíte cestu k SDK souborům ve File -> Settings-> Languages & Frameworks -> zde nastavíte cestu k SDK souborům pro Flutter i Dart
  - d. Stáhněte si v Android Studiu emulátor
    - Kliknete na ikonku mobilního telefonku s androidem (Android Virtual Device Manager) -> Create Virtual Device -> Vyberete si mobilní telefon -> Next -> Vyberete Api level (19 a výš) -> Next -> Pojmenujete a zmáčknete na tlačítko Finish
4. Projekt si rozbalíte ze zipu, otevřete v Android Studiu a spustíte

### 1.1.2 Spuštění na mobilním zařízení s operačním systémem Android

Pro instalaci a spuštění aplikace postupujte následovně:

1. Přejděte na odkaz: [https://github.com/Hellman666/stock\\_app](https://github.com/Hellman666/stock_app)
2. Stáhněte si soubor Stock\_sim.apk
3. Povolení z cizích zdrojů:

- a. Pokud nemáte povolené ve Vašem zařízení instalaci z cizích zdrojů, tak by se Vám mělo zobrazit vyskakovací okno s příslušným tlačítkem pro povolení, které Vás přesměruje do nastavení
  - b. Pokud si nejste jisti, že máte povolenou instalaci z cizích zdrojů, ale žádné vyskakovací okno se Vám nezobrazí, tak přejděte do Nastavení-> Zabezpečení-> Neznámé zdroje/Instalovat neznámé aplikace a povolte instalaci např. z prohlížeče Google, nebo z jiného, který používáte
4. Po povolení instalace z cizích zdrojů přejděte zpátky do složky se staženým souborem a soubor už pouze nainstalujte
  5. Aplikaci nyní můžete spustit a bude plně funkční

### **1.1.3 Spuštění pro vývojáře na platformě iOS**

Pro spuštění na platformě iOS využijte tento video návod:

<https://youtu.be/SqggQlv9rx0>

## 1.2 Použité vývojové prostředí a SDK

### 1.2.1 Android Studio

- Verze: 2021.1.1 (Bumblebee)
- Vývojář: Google
- Ke stažení: <https://developer.android.com/studio>
- Licence: Apache Licence 2.0

### 1.2.2 Knihovny

- hexcolor: ^2.0.4
- syncfusion\_flutter\_charts: ^19.3.55
- http: ^0.13.4
- sqflite: ^2.0.2
- flutter\_launcher\_icons: ^0.9.2
- url\_launcher: ^6.0.20

### 1.2.3 Flutter SDK

- Verze 2.5.3
- Vývojář: Google
- Ke stažení: <https://docs.flutter.dev/get-started/install>
- Licence: BSD licence

### 1.2.4 Dart SDK

- Verze: 2.14.4
- Vývojář: Google
- Ke stažení: <https://dart.dev/get-dart>
- Licence: BSD licence

### 1.3 Funkce aplikace

Mobilní aplikace slouží primárně pro simulované obchodování na finančním trhu s akciami. Do aplikace je možné přistupovat odkudkoli s přístupem k internetu. Z důvodu, že se k API přistupuje online. V aplikaci je možné si zobrazovat akcie firem, které si můžeme vyhledat ve vyhledávacím poli a různě s nimi obchodovat. Tedy nakupovat je a prodávat v určitou chvíli v reálném čase. Díky přehlednému svíčkovému grafu můžeme pozorovat vývoj akcie v minulosti a předpovídat její následující vývin. Také si v aplikaci můžeme označit jednotlivé akcie jako oblíbené, procházet otevřené obchody, které jsme provedli a stále neuzavřeli, nebo také historii již provedených obchodů.

### 1.4 Popis knihoven

Všechny následující využívané knihovny jsou volně dostupné zde: <https://pub.dev/>. Tato stránka je veřejná knihovna pro veškeré balíčky, které Vám usnadní práci při programování. Ovšem v některých případech, jakmile se využívá více těchto balíčků současně, může se stát, že se různé balíčky budou překrývat funkcemi a nemusí Vám fungovat, nebo budou fungovat špatně. Proto se doporučuje využívat ve velkých projektech pouze jejich minimální počet. Ovšem tento projekt není příliš rozsáhlý, ale i přes to jsem se pokoušel využívat minimální počet knihoven, aby se případně zbytečně nepřekrývaly.

```
dependencies:
  flutter:
    sdk: flutter

  hexcolor: ^2.0.4
  syncfusion_flutter_charts: ^19.3.55
  http: ^0.13.4
  sqflite: ^2.0.2
  flutter_launcher_icons: ^0.9.2
  url_launcher: ^6.0.20
```

Obrázek 1 Příklady knihoven využívaných v projektu

Zdroj: vlastní

### 1.4.1 Hexcolor

- Verze: 2.0.4
- Tento balíček umožňuje přidávat hexadecimální barevné kódy do Flutter projektu
- Ukázka z kódu:

```
boxShadow: [
  BoxShadow(
    color: price > 100 ? HexColor("#00FF00") : Colors.red,
    //color: Colors.grey.withOpacity(0.7),
    spreadRadius: 2,
    blurRadius: 8,
    offset: const Offset(4, 5)
  ), // BoxShadow
]
```

Obrázek 2 Ukázka HexColor balíčku

Zdroj: vlastní

### 1.4.2 syncfusion\_flutter\_charts

- Verze: 19.3.55
- Tento balíček je pro zobrazování veškerých grafů
- V aplikaci se využívá převážně pro zobrazování takzvaného svíčkového grafu, ve kterém se vykresluje hodnota akcií



Obrázek 3 Ukázka svíčkového grafu

Zdroj: vlastní

- Ukázka z kódu:

```

— return SfCartesianChart(
    title: ChartTitle(text: widget.title),
    trackballBehavior: _trackballBehavior,
    series: [
        CandleSeries<ChartSampleData, DateTime>(
            dataSource: _chartData,
            xValueMapper: (ChartSampleData sales, _) => sales.x,
            lowValueMapper: (ChartSampleData sales, _) => sales.low,
            highValueMapper: (ChartSampleData sales, _) => sales.high,
            openValueMapper: (ChartSampleData sales, _) => sales.open,
            closeValueMapper: (ChartSampleData sales, _) => sales.close,
            enableSolidCandles: true,

            ), // CandleSeries
    ],
    primaryXAxis: DateTimeAxis(
        dateFormat: DateFormat.MMM(),
        majorGridLines: MajorGridLines(width: 0)), // DateTimeAxis
    primaryYAxis: NumericAxis(
        minimum: minimum - 0.2,
        maximum: maximum + 0.2,
        interval: 0.1,
        numberFormat: NumberFormat.simpleCurrency(decimalDigits: 0)), // NumericAxis
); // SfCartesianChart
}
else{
— return Center(child: CircularProgressIndicator());
}

```

Obrázek 4 Ukázka Syncfusion balíčku - svíčkový graf

Zdroj: vlastní

### 1.4.3 http

- Verze: 0.13.4
- Tento balíček obsahuje sadu funkcí a tříd, které usnadní využívání zdrojů http
- Ukázka z kódu:

```

Future<List> getTime(String symbol) async {
  var client = http.Client();
  var timeSeriesDaily;
  List timeSeriesList = [];

  try{
    var response = await client.get(Uri.parse(
      'https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=$symbol&apikey=$apikey'));
    if (response.statusCode == 200) {
      var jsonString = response.body;
      var jsonMap = json.decode(jsonString);
      var timeSeries = jsonMap["Time Series (Daily)"];
      timeSeries.entries.forEach((e){
        timeSeriesList.add(e);
      });
      return timeSeriesList;
    }
  }
  catch(e) {
    print('error $e');
    return timeSeriesDaily;
  }
  return timeSeriesDaily;
}

```

**Obrázek 5 Ukázka http balíčku**

**Zdroj: vlastní**

#### 1.4.4 Sqflite

- Verze: 2.0.2
- SQLite plugin pro Flutter pomáhající vkládání/dotazy/aktualizace/mazání a jiné operace v databázi. Podrobnější popis najdete v dokumentaci níže

#### 1.4.5 Flutter\_launcher\_icons

- Verze: 0.9.2
- Tento plugin je pro vytvoření ikony aplikace, která se bude zobrazovat na displeji mobilního telefonu.



### 1.4.6 url\_launcher

- Verze: 6.0.20
- Tento plugin pomáhá k připojení odkazů na webové stránky. V aplikaci se využívá ve třídě setting. Díky tomuto pluginu je možnost po kliknutí na kartu přesměrování na webovou stránku.
- Ukázka použití:

```
onTap: () async {  
  const url = 'https://github.com/Hellman666/stock_app';  
  if (!await launch(url)) throw 'Could not launch $url';  
},
```

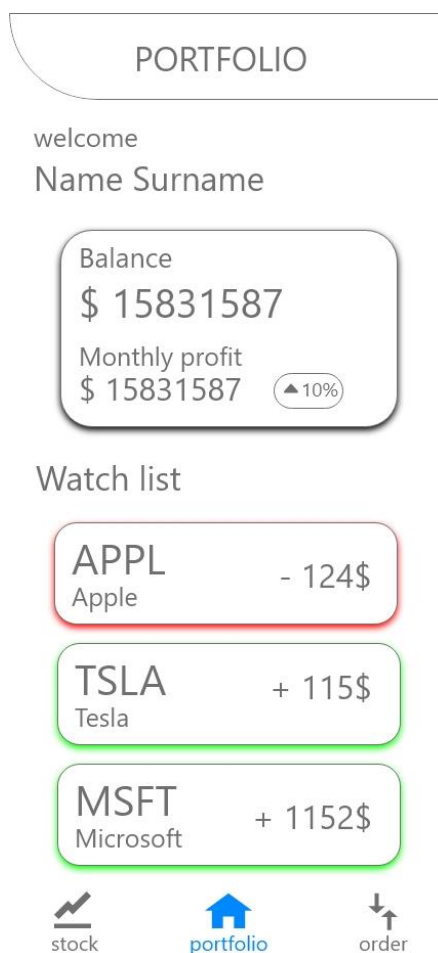
Obrázek 6 Ukázka url\_launcher balíčku

Zdroj: vlastní

## 1.5 Technické zpracování

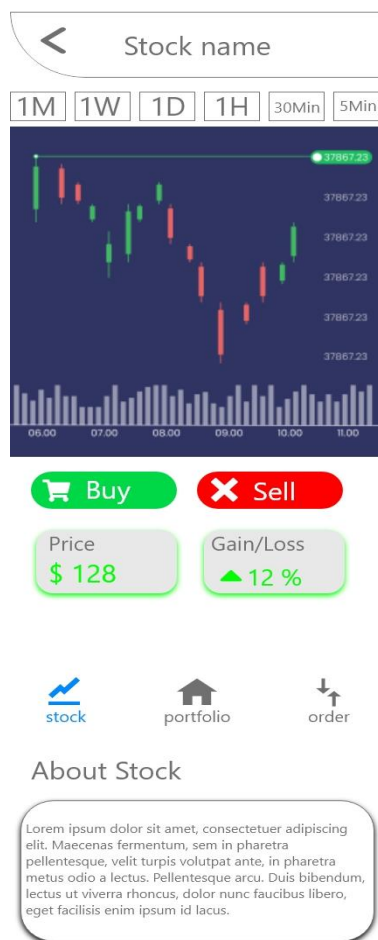
### 1.5.1 Návrh aplikace

Návrh aplikace je vytvořený v programu Adobe XD, ve kterém jsem si rozvrhl základní vzhled stránek, jejich fungování a vzájemné propojení. Původní nákresy aplikace jsem si načrtnul na papíru a poté díky jednoduchosti XD jsem rychle předělal do programu. Tento program jsem si vybral, jelikož je jednoduchý a přehledný. Vytvoření UI přes tento program mi bylo umožněné díky zkušební licenci. Všechny návrhy UI jsou k dispozici v projektu ve složce doc/UI



Obrázek 7 Ukázka UI pro portfolio

Zdroj: vlastní



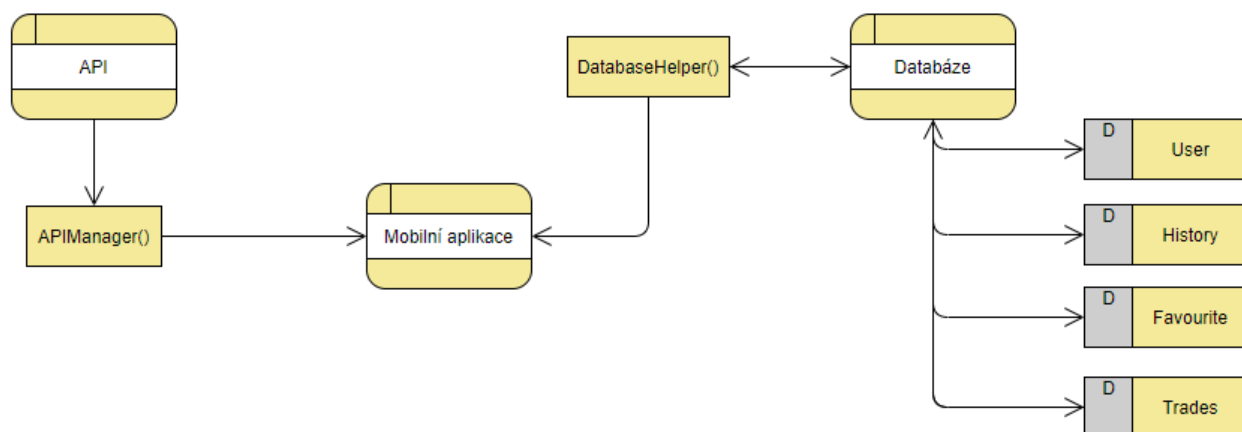
Obrázek 8 Ukázka UI pro stránku se zvolenou akcií

Zdroj: vlastní

### 1.5.2 Teoretické vytvoření aplikace

Pro vytvoření aplikace jsem zvolil již původně zmiňovaný framework Flutter, který je podrobněji rozebraný níže v dokumentaci. Dále jsem si vybral API Alpha Vantage, které je také podrobněji rozepsané níže v dokumentaci. V poslední řadě bylo potřeba si zvolit nějakou databázi pro ukládání dat, které jsou potřeba v aplikaci pro fungování. Původně jsem chtěl zvolit databázi Firebase od Googlu, která podporovala jak storage pro data, tak i zabudovanou Authentifikaci pro přihlašování uživatele, které také původně mělo být v tomto projektu. Ovšem po podrobnější konzultaci s vedoucím projektu, jsme přišli k závěru, že není třeba přihlašování v aplikaci, která je pouze jako simulátor. Z toho důvodu jsem se rozhodl předělat projekt a využít lokální databázi SQLite, která je pro tento projekt ideální.

V následujícím diagramu můžeme vidět, jak funguje celá aplikace.



**Obrázek 9 Data flow diagram**

**Zdroj: vlastní**

## 1.6 Flutter

První verze frameworku Flutter byla vytvořena v roce 2017, to znamená že je to poměrně nový framework. Aktuální verze je 2.10.3, která vyšla letos 2. března. Jeho hlavní výhodou je open-source rozhraní, které je vytvořené společností Google a samotnou komunitou vývojářů. Vývojáři své doplňky projektů sdílejí na serveru [www.pub.dev](https://www.pub.dev). Hlavní výhodou Flutteru je multiplatformost. To znamená, že můžete vytvářet zároveň aplikace pro Android, iOS, Windows, MacOS, Linux, ale také web. Veškerý kód je psaný v programovacím jazyce Dart, vytvořeného také společností Google.

Základní knihovna, napsaná v jazyce Dart, poskytuje základní třídy a funkce, které se používají k vytváření aplikací pomocí Flutteru. Samotný Flutter poté funguje na tzv. widgetech. Tyto widgety můžeme mezi sebou sestavovat a vytvořit z nich určitý strom widgetů. Přitom samotný strom, těchto widgetů je samotný widget. Pro pochopení například Text je widget, do kterého vypíšeme to, co chceme, aby se zobrazilo. Ale také TextStyle je widget, přitom tady definujeme velikost, barvu, font atd. Určité widgety představují objekty, jiné charakteristiky. Hlavní metodou Flutteru, přes kterou se spouští samotná aplikace je main. Pro ukázkou „Hello World!“ ve Flutteru vypadá následovně:

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),
        body: const Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```

Obrázek 10 Ukázka Hello World ve Flutteru

Zdroj: <https://docs.flutter.dev/get-started/codelab>

## 1.7 Alpha Vantage

Společnost Alpha Vantage Inc. je podporována prestižním Y Combinator a složená z úzce propojené komunity výzkumníků, inženýrů a obchodních profesionálů. Uzavřela poradenství s významnými burzami a institucemi po celém světě, aby se stala přednostním poskytovatelem API pro akcie, forex, ale také kryptoměny. Filozofie designu API na burze se soustředí na co nejefektivnější přístup k finančním údajům. Alpha Vantage nabízí několik balíčků včetně balíčku zdarma, který je sice určitým způsobem omezen, ale pro nekomerční účely vystačující. Stačí si pouze vyzvednout API klíč a můžete neomezeně pracovat s většinou knihoven API. Přitom ostatní balíčky se určují podle žádostí na server.

Choose a monthly plan:

1200 API requests per minute: \$249.99/month
600 API requests per minute: \$199.99/month
300 API requests per minute: \$149.99/month
150 API requests per minute: \$99.99/month
75 API requests per minute: \$49.99/month

Obrázek 11 Možnosti upgradu Alpha Vantage API

Zdroj: <https://www.alphavantage.co/premium/>

I přes poměrně vysoké ceny je Alpha Vantage poměrně oblíbené API pro akciový trh. Mezi hlavní partnery patří mimo jiné také London Stock Exchange, Morgan Stanley, Google, Harvard business school a již zmiňovaný Y Combinator.

Alpha Vantage je také jediné kontaktní místo pro data z globálního trhu v reálném čase a historická data o globálním trhu poskytovaná prostřednictvím RESTful. To znamená, že je systém flexibilní pro různé případy použití. Škálovatelný tak, aby vyhovoval velkému počtu uživatelů, komponent a přizpůsobitelný v reálném čase.

Alpha Vantage také nabízí tzv. Alpha Academy. Je to jistá učebnice pro alphavantage, nejen pro investory, ale také softwarové vývojáře, vědce, studenty či pedagogy. Vzdělávací obsah je

vyvinut odborníky z oborů kvantitativního investování, strojového učení, vývoje softwaru a blockchainových technologií a dalších.

Odkaz pro Alpha Academy je zde: <https://www.alphavantage.co/academy/>

Odkaz na dokumentaci Alpha Vantage API: <https://www.alphavantage.co/documentation/>

```
{
  "Global Quote": {
    "01. symbol": "IBM",
    "02. open": "129.5000",
    "03. high": "131.4000",
    "04. low": "129.3100",
    "05. price": "131.3500",
    "06. volume": "3516923",
    "07. latest trading day": "2022-03-25",
    "08. previous close": "129.2500",
    "09. change": "2.1000",
    "10. change percent": "1.6248%"
  }
}
```

Obrázek 12 Ukázka Alpha Vantage API

Zdroj: [https://www.alphavantage.co/query?function=GLOBAL\\_QUOTE&symbol=IBM&apikey=demo](https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=IBM&apikey=demo)

## 1.8 SQLite

Je to relační databázový systém obsažený v poměrně malé knihovně napsaný v jazyce C. Je šířen pod licencí Public domain. První vydání SQLite se datuje k roku 2000.

Na rozdíl od databází založených na principu klient-server, kde je databázový server spuštěn jako samotný proces, je SQLite pouze nevelká knihovna, která po přilinkování k aplikaci je k dispozici pomocí jednoduchého rozhraní. Každá databáze SQLite je uložena zvlášť v souboru v zařízení. Charakteristické prvky pro systém jsou absence databázového systému ve formě grafického prostředí, absence konfigurace, absence serveru. Databáze v jednom souboru nezávislé na platformě. To s sebou nese výhody, jako je migrace dat, ale také nevýhody v podobě fragmentace.

Využití pro SQLite je široké. Používá se hlavně v programovacích jazycích C, C++, C#, PHP, Java atd., ale existují knihovny téměř pro všechny programovací jazyky, včetně Dartu.

### 1.8.1 Práce s SQFlite

V tomto případě je v projektu využíván plugin SQFlite (= SQLite pro Flutter). Práce s SQFlite je poměrně jednoduchá a rychlá. Stačí přistoupit k databázi s využitím metod pracovat s databází. Databázi nejdříve vytváříme, případně přistupujeme. Následovně už pouze píšeme metody pro přístupy do databáze a její správu. V projektu využívám sqflite v souboru sqlite\_db.dart ve třídě DatabaseHelper. Ve které spravujeme veškerou práci s databází. Podrobnější popis třídy je v oddíle Důležité třídy.

## 1.9 Modelové třídy databáze

Modelové třídy aplikace můžete najít ve složce lib/models. Jejich účelem je reprezentace informací dat, se kterými pracujeme. V aplikaci využívám model pro data z API v api\_model.dart a sqflite v sqlite\_model.dart. API model je vytvořený online generátorem <https://app.quicktype.io/>, který automaticky vygeneroval třídy s proměnnými a veškerými důležitými informacemi, které jsou potřebné pro toto API. Model pro práci s databází je podstatně jednodušší než s API. Jsou zde čtyři hlavní třídy s proměnnými. Tyto třídy jsou User, History, Favourite a Trades. Každá třída je zvlášť pro tabulku dat v databázi.



### 1.9.1.1 Ukázka User modelu

```
class User{
    int id;
    int balance;
    int profit;

    User( this.id, this.balance, this.profit);

    User.fromMap(Map<String, dynamic> res)
    : id = res["id"],
      balance = res["balance"],
      profit = res["profit"];

    Map<String, Object?> toMap(){
        return {'id': id, 'balance': balance, 'profit': profit};
    }
}
```

Obrázek 13 User model

Zdroj: vlastní

## 1.10 Důležité třídy

### 1.10.1 DatabaseHelper

Třída DatabaseHelper komunikuje již se zmíněným SQLite. Zprostředkovává komunikaci mezi SQLite databází a aplikací pomocí vlastních funkcí. V této třídě vytváříme databázi, pokud není vytvořená, přistupujeme k ní a spravujeme. V aplikaci poté přistupujeme pouze ke třídě a jednotlivým podtřídám, jako je například getBalance a hodnotu uložíme do proměnné se kterou následně pracujeme v samotném kódu aplikace.

```
void initState() {
    DatabaseHelper.getBalance().then((value) {
        User _userRow = User.fromMap(value[0]);
        setState(() {
            _balance = _userRow.balance;
        });
    });
    super.initState();
}
```

Obrázek 14 Ukázka vypsání hodnoty balance

Zdroj: vlastní

Tyto funkce zapisujeme do metody `initState()`, což je metoda, která se zavolá ihned po spuštění aplikace. To nám umožní, že jakmile spustíme aplikaci, tak se nám načtou metody z databáze a zobrazí se nám. Největší problém v tomto případě byl zajistit to, aby při akci na jiné stránce, jako je nákup, nebo prodej akcií se obnovila cena v aplikaci. Z důvodu, že aplikace funguje na principu, že vezme hodnotu z databáze, k té přičte, případně odečte hodnotu akcie a uloží zpátky do databáze byl problém, že se nezobrazovala aktuální hodnota v portfoliu uživatele. Toto jsem však vyřešil tak, že jsem znovu zavolal metodu `setState()`, což mělo efekt obnovení dat.

```
navigateToStockCardPf(title){
  Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => Stock(stockSymbol: title)),
  ).then((value) => setState(() {print('obnoveno');}));
}
```

Obrázek 15 Ukázka zavolání metody `setState()` při přesměrování

Zdroj: vlastní

V `DatabaseHelperu`, jak jsem zmiňoval, tak také spravuji databázi. Tím je myšlena práce s databází v podobě vytváření, vkládání, získávání atd.

#### 1.10.1.1 Otevření souboru:

```
_initDatabase() async {
  String path = join(await getDatabasesPath(), _databaseName);
  return await openDatabase(
    path,
    version: _databaseVersion,
    onCreate: _onCreate
  );
}
```

Obrázek 16 Ukázka otevření souboru

Zdroj: vlastní

### 1.10.1.2 Vytvoření tabulky:

```
await db.execute(
    '''
    CREATE TABLE IF NOT EXISTS $table(
        $columnId INTEGER PRIMARY KEY AUTOINCREMENT,
        $columnBalance INTEGER NOT NULL,
        $columnProfit INTEGER NOT NULL
    )
    '''
);
```

Obrázek 17 Ukázka vytváření tabulky

Zdroj: vlastní

### 1.10.1.3 Vložení dat:

```
static Future<void> insertUserRow() async {
    User _user = User(1, 50000, 0);
    final db = await instance.database;
    await db!.insert(
        table,
        _user.toMap(),
        conflictAlgorithm: ConflictAlgorithm.replace,
    );
}
```

Obrázek 18 Ukázka vkládání dat do tabulky

Zdroj: vlastní

### 1.10.1.4 Výpis dat:

```
static Future<List<Map<String, dynamic>>> getBalance() async{
    Database? db = await instance.database;
    List<Map<String, dynamic>> res = await db!.query(table);
    print(res);
    //print(res.runtimeType);
    return res;
}
```

Obrázek 19 Ukázka vypisování dat

Zdroj: vlastní

### 1.10.2 APIManager

Třída APIManager komunikuje se zmiňovaným Alpha Vantage API. Tvoří komunikaci mezi API a samotnou aplikací pomocí funkcí. Pro každou činnost je zde zvlášť funkce, kterou provádí. Výpis funkcí:

#### 1.10.2.1 *getTime*

Přistupuje k datům, které jsou potřebné pro vygenerování grafu. Vrací se zde hodnoty open, high, low, close a volume. Jsou to hodnoty peněz v dolarech, podle kterých se pohybují svíčky grafu.

#### 1.10.2.2 *getStock*

Přistupuje k základním datům o akcii. Jako je její cena, poslední tradovací den, cena za minulý den, procentuální změna akcie atd. Tyto hodnoty využívám ve widgetech, kde jsou tyto informace dostačující.

#### 1.10.2.3 *searchStock*

Je to funkce, která přistupuje ke speciální třídě API, která vrací nejlépe odpovídající symbol a informace o trhu na základě klíčového slova, které se zapíše do vyhledávání. Toto umožní najít přesně tu akcii, kterou uživatel vyžaduje bez větších obtíží.

### 1.10.3 MyApp

Zařizuje spuštění aplikace. Můžeme zde určit také základní používané barvy, fonty písem, velikosti, jak budou vypadat nadpisy atd. Také zde určujeme název aplikace a určujeme home page. V tomto případě to je Portfolio. Z důvodu, že nevyužívám žádné přihlášení, tak je odkaz přímo na hlavní stránku. Pokud by v aplikaci bylo přihlášení, tak by se právě v této třídě určilo, jestli aplikace má přesměrovat uživatele do přihlášení, nebo jestli je již přihlášen.

Ukázka třídy MyApp:

```
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'STOCK-SIM',
      theme: ThemeData(
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ), // ThemeData
      home: Portfolio(),
    ); // MaterialApp
  }
}
```

Obrázek 20 Ukázka třídy MyApp v projektu

Zdroj: vlastní

## 1.11 Nejvíce využívané widgety

Ve složce lib/widgets můžete najít několik souborů, které určují nejvíce používané widgety v aplikaci. Mezi tyto widgety patří například tlačítka, karty, nebo také samotný graf.

## 2 UŽIVATELSKÁ DOKUMENTACE

V této kapitole popíši zprovoznění aplikace na Vašem mobilním zařízení, jak se v aplikaci orientovat, popíšu obrazovky a také zde můžete vidět výsledky uživatelského testování, případně scénáře použití.

### 2.1 Spuštění

#### 2.1.1 Spuštění na mobilním zařízení s operačním systémem Android

Pro instalaci a spuštění aplikace postupujte následovně:

1. Přejděte na odkaz: [https://github.com/Hellman666/stock\\_app](https://github.com/Hellman666/stock_app)
2. Stáhněte si soubor stock\_sim.apk, který najdete ve složce spustitelný soubor
3. Povolení z cizích zdrojů:
4. Pokud nemáte povolené ve Vašem zařízení instalaci z cizích zdrojů, tak by se Vám mělo zobrazit vyskakovací okno s příslušným tlačítkem pro povolení, které Vás přesměruje do nastavení
5. Pokud si nejste jisti, že máte povolenou instalaci z cizích zdrojů, ale žádné vyskakovací okno se Vám nezobrazí, tak přejděte do Nastavení-> Zabezpečení-> Neznámé zdroje/Instalovat neznámé aplikace a povolte instalaci např. z prohlížeče Google, nebo z jiného, který používáte
6. Po povolení instalace z cizích zdrojů přejděte zpátky do složky se staženým souborem a soubor už pouze nainstalujte
7. Aplikaci nyní můžete spustit a bude plně funkční

#### 2.1.2 Spuštění na platformě iOS

Pro spuštění na platformě iOS jsem používal tento video návod:

<https://youtu.be/SqgqQlv9rx0>

## 2.2 Ovládání

Aplikaci spustíte po instalaci kliknutím na ikonu ve vašem mobilním zařízení. Po spuštění aplikace budete přesměrováni na stránku portfolio. Při prvním spuštění aplikace se vám zobrazí okno, ve kterém máte dotaz, jestli jste připraveni na obchodování a po kliknutí na „Yes“ se vám načtou data z databáze a můžete využívat aplikaci naplno.

Aplikace se ovládá pomocí navigačního pole na spodní části obrazovky. Zde vás to přesměruje na stránku, kterou si vyberete.



Obrázek 21 Navigační pole

Pokud si budete chtít na stránce stocks najít akcii, která vás zrovna zajímá, tak stačí do vyhledávacího pole napsat nějaké klíčové slovo poté zmáčknou ikonu lupy pro vyhledání. Akcie se vypisují v kartách a po zmáčknutí na jednotlivou kartu budete přesměrováni na akcii, kterou jste si vybrali. Vybranou akcii si můžete označit v pravém horním rohu hvězdičkou do oblíbených, případně šipkou zpět se dostanete na předchozí stránku



Obrázek 22 Navbar stránky akcie

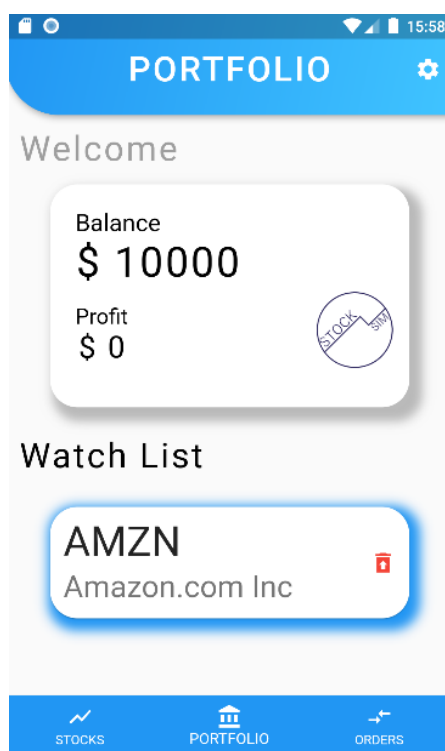
Pro nákup, případně prodej akcie stačí zmáčknout tlačítko buy/sell. V tu chvíli se Vám zobrazí okno, jestli tuto akci chcete opravdu provést a v tu chvíli můžete vybrat akcii koupit či prodat. Po vypnutí aplikace se uživatel nemusí bát ztráty dat, díky jejich uložení v databázi.

## 2.3 Funkce

V aplikaci je celkem 5 obrazovek, mezi které patří portfolio, stocks, orders, setting a stock. To je obrazovka pro jednotlivou akcii.

### 2.3.1 Portfolio

Stránka portfolio se Vám zobrazí po spuštění aplikace. Můžeme zde najít základní informace o Vašem profilu, jako je hodnota a profit. Také si můžete prohlédnout Váš watchlist, což je seznam oblíbených akcií. Případně na této stránce můžete přejít do nastavení a restartovat celou aplikaci.



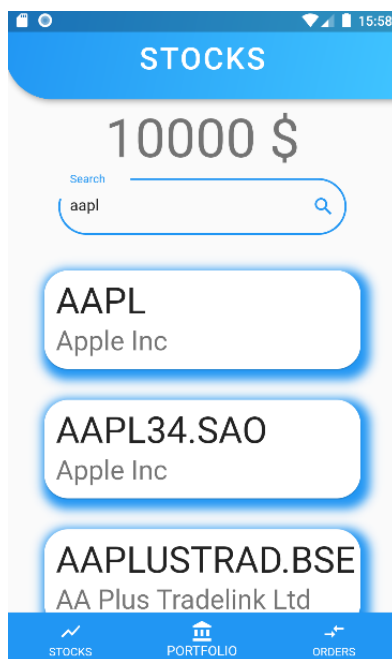
**Obrázek 23 Stránka Portfolio**

**Zdroj: vlastní**

### 2.3.2 Stocks

Na stránce stocks můžete najít ve vyhledávacím poli vaši oblíbenou akcii. Také zde můžete vidět aktuální hodnotu účtu. Po vyhledání akcie ve vyhledávacím poli se Vám vypíše seznam akcií s podobným klíčovým slovem a můžete si vybrat právě tu akcii, kterou potřebujete.



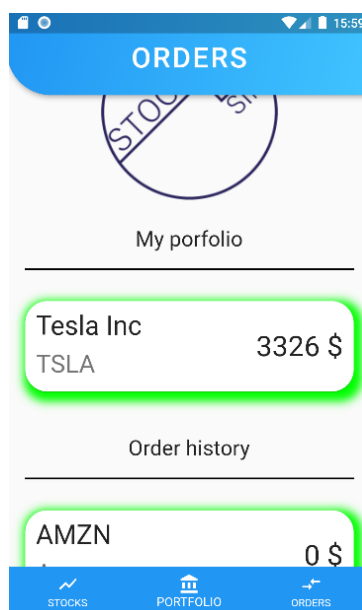


Obrázek 24 Stránka Stocks

Zdroj: vlastní

### 2.3.3 Orders

Na této stránce se vám zobrazují otevřené obchody, u kterých můžete vidět cenu, za kterou jste akcii koupili a název akcie. Níže můžete vidět již zavřené obchody, u kterých vidíte název akcie, se kterou jste obchodovali a váš profit, nebo ztrátu na akcii.

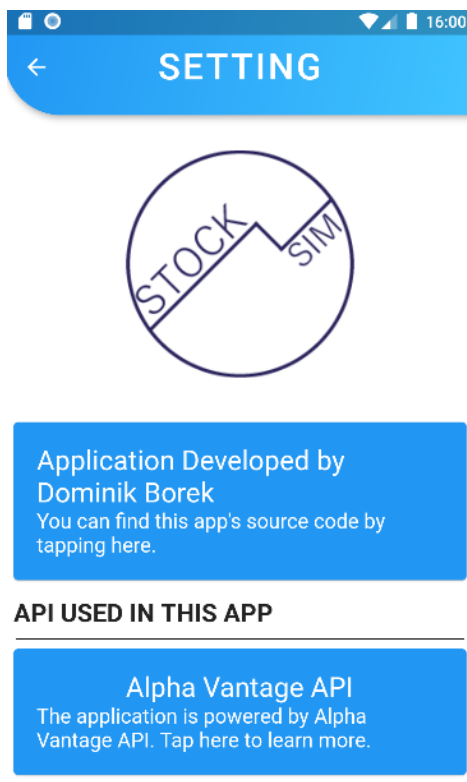


Obrázek 25 Stránka Orders

Zdroj: vlastní

### 2.3.4 Setting

Stránka setting je v této aplikaci hlavně z důvodu pro restart samotné aplikace. Dále se zde uživatelé mohou dozvědět o využívaném API, nebo technologii. Kdyby měli případně zájem, tak se mohou podívat na veřejný github, na kterém je tato aplikace nahraná.

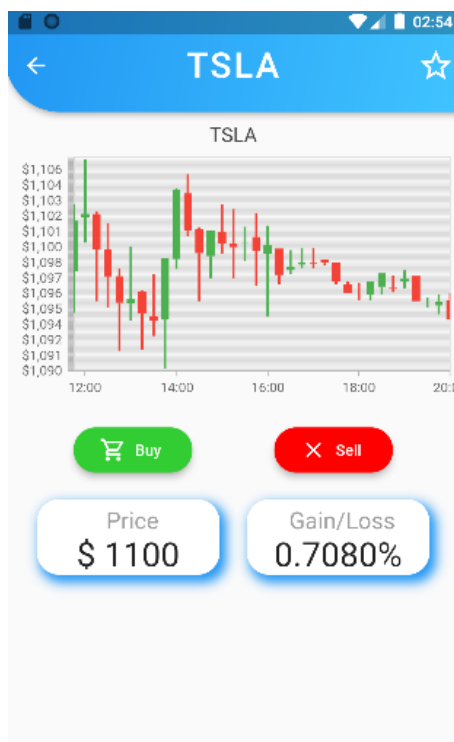


**Obrázek 26** Stránka setting

**Zdroj:** vlastní

### 2.3.5 Stock

Asi nejdůležitější aktivitou v celé aplikaci je stránka stock. Na této stránce můžete vidět graf zvolené akcie, kterou jste si chtěl otevřít z předešlé stránky. Tuto akcii si také můžete vložit do svých oblíbených, které se Vám budou zobrazovat na stránce portfolio. Dále právě na této stránce můžete provádět obchody, přesněji nákup a prodej akcie. Pod tímto vším se Vám zobrazují základní informace o akcii, jako je její cena, aktuální procentuální hodnota akcie a informace o samotné firmě akcii poskytující.

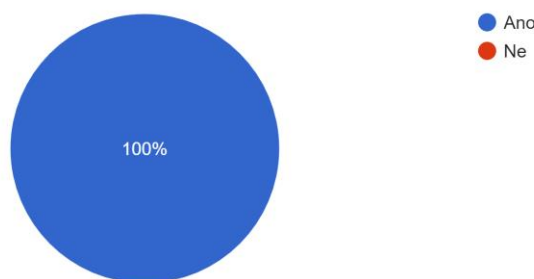


Obrázek 27 Stránka Stock

## 2.4 Uživatelské testování

Aplikace je otestována celkem 4 respondenty ve věku od 17 do 29 let. Jejich závěry jsem získal díky formuláři <https://forms.gle/aLUenGu4fszsvDMw6>, ve kterém jsem se respondentů tázal na několik otázek ohledně aplikace. V dotazníku jsem využíval k odpovědím výběr z možností, lineární stupnici od 1 do 5 (1 = nejlepší, 5 = nejhorší) a volné odpovědi. Zde můžete vidět jejich závěry:

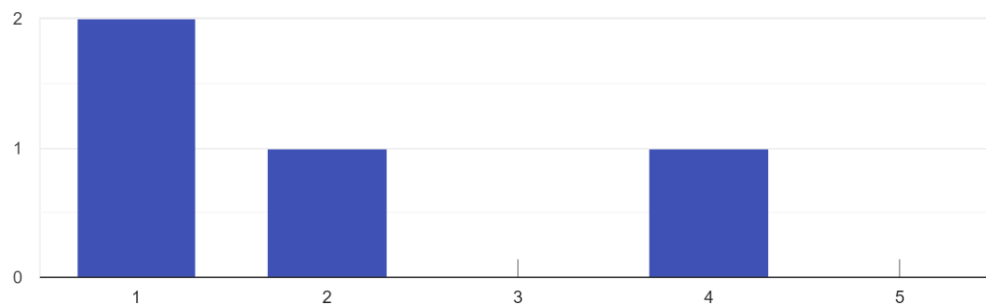
Zaujala vás aplikace?  
4 odpovědi



**Graf 1 Odpovědi respondentů 1**

**Zdroj: vlastní**

Jak se vám aplikace líbila?  
4 odpovědi

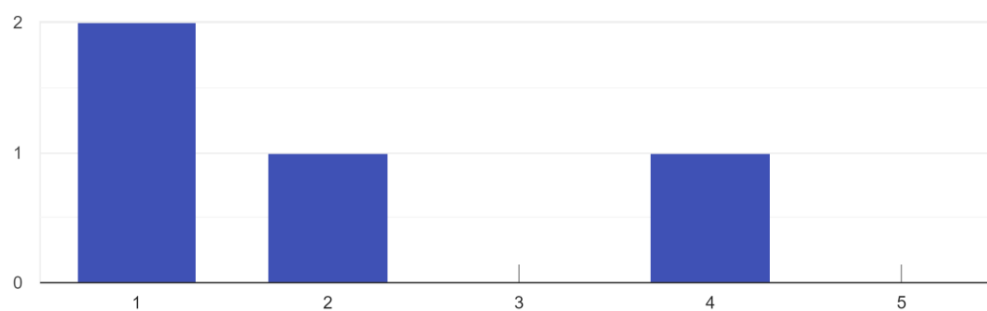


**Graf 2 Odpovědi respondentů 2**

**Zdroj: vlastní**

Jak se vám líbil vzhled aplikace?

4 odpovědi

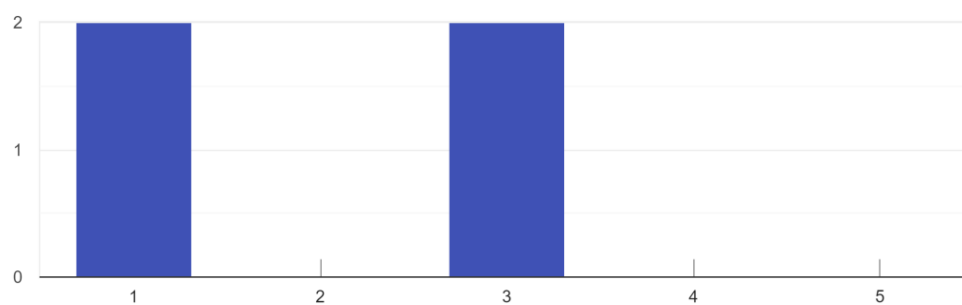


**Graf 3 Odpovědi respondentů 3**

Zdroj: vlastní

Jak by jste ohodnotili funkčnost aplikace?

4 odpovědi

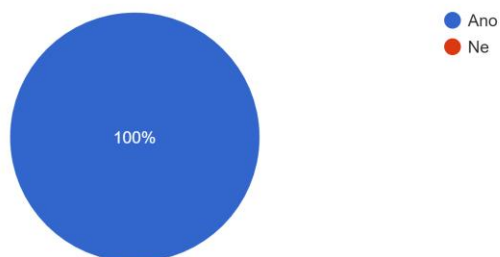


**Graf 4 Odpovědi respondentů 4**

Zdroj: vlastní

Byla pro Vás aplikace zajímavá?

4 odpovědi

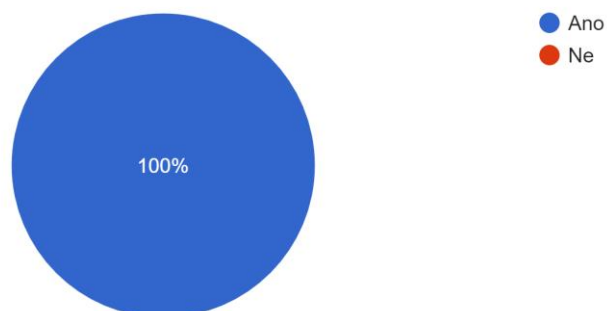


**Graf 5 Odpovědi respondentů 5**

Zdroj: vlastní

Zaujmul vás akciový trh?

4 odpovědi

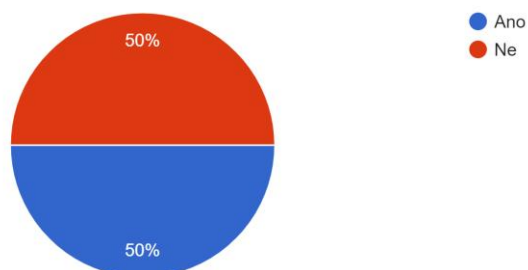


**Graf 6 Odpovědi respondentů 6**

**Zdroj: vlastní**

Máte předchozí zkušenosti s akciemi?

4 odpovědi



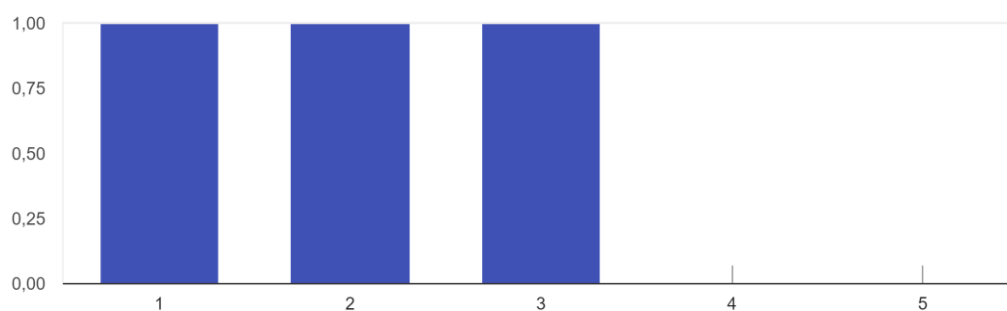
**Graf 7 Odpovědi respondentů 7**

**Zdroj: vlastní**

V další části odpovídali již pouze 3 respondenti z důvodu, že 4 respondent z důvodu, že jiné konkurenční aplikace nezná, tak nemůže na tento dotaz odpovídat.

Líbila se vám aplikace v porovnání s konkurencí?

3 odpovědi

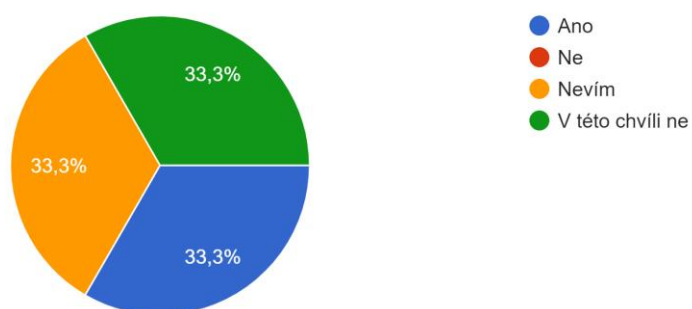


**Graf 8 Odpovědi respondentů 8**

Zdroj: vlastní

Myslíte si, že je aplikace konkurence schopná?

3 odpovědi



**Graf 9 Odpovědi respondentů 9**

Zdroj: vlastní

Jaký program jste využíval pro akciový trh dříve?

3 odpovědi

Trading 212

StockBroker, Plus500

Interactive brokers, Trade station

**Obrázek 28 Volné odpovědi respondentů 1**

Zdroj: vlastní

Jaký je váš názor na aplikaci?

3 odpovědi

Líbí se mi, chtělo by to přidat pár funkcí.

Myslím, že pokud se autor bude ještě aplikaci věnovat, tak ji v budoucnu bude moct využívat také jako reálný program pro nákup akcií a ne jen jako simulátor. V tuto chvíli si myslím že v budoucnosti by to mohl být velmi užitečný simulátor pro lidi, kteří se v akciích nevyznají.

Přehledná, vhodná k intradennímu obchodování, umožňuje rychlý nákup, prodej a uzavření akcie

**Obrázek 29 Volné odpovědi respondentů 2**

**Zdroj: vlastní**



## ZÁVĚR

Mým cílem projektu bylo vytvoření aplikace, pro obchodování s akcemi, která bude uživatelsky jednoduchá, nebude uživatele mást a po grafické stránce bude moderní. Ke splnění těchto cílů jsem využil jednu z nejmodernějších technologií pro tvorbu mobilních aplikací, což je framework Flutter. Dále jsem chtěl, aby aplikaci přijímala pouze reálná data v reálných časech a z API takového, že uživatel bude mít jistotu, že je opravdu na trhu. K tomuto jsem využil Alpha Vantage API, které je jedno z nejmodernějších a nejlepších API veřejně dostupných.

Celý význam aplikace spočívá hlavně v jednoduchosti projektu. Tím, že se zde obchoduje pouze za virtuální peníze, se uživatel nemusí bát žádných obchodů a trénovat se na reálný trh. Nevýhodou toho je, že pokud uživatel vloží peníze do akcie, tak to trh nijak neovlivní, ovšem v takovém případě nelze toto nijak obejít z důvodu, že se zde využívá API, které je s trhem přímo propojené. Pro uživatele je to vlastně pouze aplikace, ve které se naučí základní možnosti na trhu, jak celý trh funguje atd.

Během programování této aplikace, bylo ze začátku pro mě obtížné pochopit fungování Flutteru s API a následně s databází. Z důvodu, že před tímto projektem jsem neměl žádné zkušenosti s prací mezi API, databází a projektem dohromady, tak mi to poměrně komplikovalo celou práci. Ovšem časem jsem poměrně pochopil, jak to funguje a vymyslel způsob, jak všechno zrealizovat. Díky předchozímu projektu, kdy jsem pracoval hlavně na UI aplikace, tak mi nedělalo žádné obtíže vytvořit celé UI během několika dní přehledně a tak, aby bylo bezproblémové měnit některé metody a proměnné.

Během programování jsem zjistil, že API nezobrazuje reálná data v reálném čase, jak uvádí, ale pouze data z minulosti. To znamená, že graf, který se vám zobrazí v konkrétní akci není v ten čas, ve který si ho zobrazujete, ale je zobrazen z dat předchozího dne. Další velký problém, který jsem měl během programování a který je i pro uživatele poměrně velkou nevýhodou je daný počet requestů za minutu na API. V případě, kdy je API poskytované zdarma, tak je to pouhých 5 requestů. To má za důsledek to, že některé data se nezobrazují, nebo zobrazí špatně.

V mobilní aplikaci se již od začátku vyskytovalo několik problému. Hned tím prvním, na který jsem narazil bylo propojení s původně vybranou databází Firebase. Je to databáze přímo od Googlu, která je dělaná právě pro Flutter. Ovšem z důvodu, že je zde několik různých funkcí databáze, jako je Authentication, Realtime database nebo Storage, tak se mi nepovedlo propojit všechno tak, aby

to mezi sebou fungovalo. Při konzultaci s vedoucím projektu jsme dospěli k závěru, že vlastně celé přihlášení, které mělo aplikace původně obsahovat, je vlastně zbytečné, jelikož je to pouze simulátor a uživatel tedy nepotřebuje mít zaheslované žádné data. V tom případě jsem se rozhodl databázi Firebase nevyužívat a místo toho jsem použil SQLite databázi. Je to lokální databáze typu SQL. Tím, že jsem měl ze školy již zkušenosti s databázemi SQL typu, bylo pro mě jednoduché vytvořit nějaký model této databáze tak, aby fungoval. Ovšem během propojování se mi vyskytl další zádrhel v podobě zobrazení updatovaných dat. Z důvodu, že jsem měl v aplikaci vypsaná data přímo z databáze, které se zobrazily hned po spuštění aplikace, musel jsem nějak vyřešit problém s obnovením těchto dat. To jsem ale vyřešil docela rychle, kdy jsem si uvědomil, že po vrácení se na jinou stránku stačí pouze setState metodu, která načte data v initState znovu. Čímž docílím updatu vypsaných dat. Další poměrně velký problém byl pro mě práce s API. Jelikož jsem nikdy předtím neměl ve Flutteru zkušenosti s prací z API, přesněji s takovým obsáhlým, jako je zrovna Alpha Vantage, tak jsem měl problémy získat data. Ale po určité době studování API a propojení API s aplikací jsem došel k tomu, jak to celé funguje a napsal konečně funkci, která fungovala a poté už vytvářel jen další, podobné funkce, pro další widgety aplikace.

V celkovém projektu se mi podařilo splnit téměř všechny cíle projektu až na jeden, kterým je registrace uživatele. Jak jsem zmiňoval již dříve, tak po konzultaci s vedoucím projektu jsem se rozhodl tuto možnost vypustit, jelikož se jedná pouze o simulátor, takže by celé toto přihlášení zde bylo zbytečné. Jinak je projekt funkční a vypadá přesně tak, jak jsem si původně představoval.

Časem bych chtěl doplnit například možnosti nákupu více akcií současně, rozšířit projekt o možnosti nákupu krypta, nebo forexu a případně převést aplikaci na reálný trh tak, že by se dalo zde obchodovat s reálnými akciemi, které by uživatel měl následně doopravdy ve vlastnictví, a ne pouze v simulátoru. Z časových důvodů jsem ovšem nemohl rozšiřovat projekt tak, jak bych jednoho dne chtěl, aby reálně vypadal a fungoval. Je ale možné se k projektu vrátit a pokračovat v něm v budoucnu, jelikož si myslím, že to bude užitečné pro hodně lidí, kteří se chystají vstoupit na burzovní trh a nemají s ním žádné předešlé zkušenosti.

## SEZNAM POUŽITÉ LITERATURY

1. *SQLite* [online]. [cit. 2022-03-22]. Dostupné z: <https://blog.logrocket.com/flutter-sqlite-how-to-persist-data/>
2. *SQLite* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.tutorialkart.com/flutter/flutter-sqlite-tutorial/>
3. *SQLite* [online]. [cit. 2022-03-22]. Dostupné z: [https://github.com/tekartik/sqlite/blob/master/sqlite/doc/how\\_to.md](https://github.com/tekartik/sqlite/blob/master/sqlite/doc/how_to.md)
4. *SQLite* [online]. [cit. 2022-03-22]. Dostupné z: <https://medium.com/flutter-community/using-sqlite-in-flutter-187c1a82e8b>
5. *SQLite data recovery* [online]. [cit. 2022-03-22]. Dostupné z: <https://stackoverflow.com/questions/57307050/flutter-run-a-function-every-time-the-page-changes>
6. *SQLite data recovery* [online]. [cit. 2022-03-22]. Dostupné z: <https://stackoverflow.com/questions/49830553/how-to-go-back-and-refresh-the-previous-page-in-flutter>
7. *SQLite multiple tables* [online]. [cit. 2022-03-22]. Dostupné z: <https://stackoverflow.com/questions/63103065/flutter-sqlite-databaseexceptionno-such-table-project>
8. *Diagram creator* [online]. [cit. 2022-03-22]. Dostupné z: <https://online.visual-paradigm.com/drive/#diagramlist:proj=0&dashboard>
9. *Alpha Vantage* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.alphavantage.co/>
10. *SQLite* [online]. [cit. 2022-03-22]. Dostupné z: <https://cs.wikipedia.org/wiki/SQLite>
11. *SQLite* [online]. [cit. 2022-03-22]. Dostupné z: <https://pub.dev/packages/sqlite>
12. *Flutter* [online]. [cit. 2022-03-22]. Dostupné z: [https://cs.wikipedia.org/wiki/Flutter\\_\(software\)](https://cs.wikipedia.org/wiki/Flutter_(software))
13. *Flutter* [online]. [cit. 2022-03-22]. Dostupné z: <https://flutter.dev/>
14. *Flutter cookbook* [online]. [cit. 2022-03-22]. Dostupné z: <https://docs.flutter.dev/cookbook>

15. *Alpha Vantage connect* [online]. [cit. 2022-03-22]. Dostupné z: <https://mobikul.com/http-api-calling-in-flutter/>
16. *Alpha Vantage connect* [online]. [cit. 2022-03-22]. Dostupné z: <https://medium.com/flutter-community/working-with-apis-in-flutter-8745968103e9>
17. *Alpha Vantage connect* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.geeksforgeeks.org/implementing-rest-api-in-flutter/>
18. *Work with API* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.youtube.com/watch?v=PytQROAncxg>
19. *Use Alpha Vantage in graph* [online]. [cit. 2022-03-22]. Dostupné z: [https://www.youtube.com/watch?v=WJ2t\\_LYb\\_0&t=330s](https://www.youtube.com/watch?v=WJ2t_LYb_0&t=330s)
20. *HTTP request* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.youtube.com/watch?v=hgPTvi0OM4A>
21. *HTTP request* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.youtube.com/watch?v=jymr1hEvfkM&t=146s>
22. *Work with local data* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.youtube.com/watch?v=Hqqz2BaPUis&t=300s>
23. *Animation in flutter* [online]. [cit. 2022-03-22]. Dostupné z: <https://www.youtube.com/watch?v=6vPF2IqCJ9Q&t=532s>
24. *Google form* [online]. [cit. 2022-03-30]. Dostupné z: <https://docs.google.com/forms/>
25. *SQLite viewer* [online]. [cit. 2022-03-30]. Dostupné z: <https://inloop.github.io/sqlite-viewer/>

## **SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

API – Application programming interface (rozhraní pro programování aplikací)

SQL – Structure query language (standardizovaný strukturovaný jazyk)

UI – User interface (uživatelské rozhraní)

SDK – Software development kit

## SEZNAM OBRÁZKŮ

Obrázek 1 Příklady knihoven využívaných v projektu Zdroj: vlastní .....	13
Obrázek 2 Ukázka HexColor balíčku Zdroj: vlastní.....	14
Obrázek 3 Ukázka svíčkového grafu Zdroj: vlastní.....	14
Obrázek 4 Ukázka Syncfusion balíčku - svíčkový graf Zdroj: vlastní .....	15
Obrázek 5 Ukázka http balíčku Zdroj: vlastní .....	16
Obrázek 6 Ukázka url_launcher balíčku Zdroj: vlastní .....	17
Obrázek 7 Ukázka UI pro portfolio Zdroj: vlastní.....	18
Obrázek 8 Ukázka UI pro stránku se zvolenou akcií Zdroj: vlastní .....	19
Obrázek 9 Data flow diagram Zdroj: vlastní .....	20
Obrázek 10 Ukázka Hello World ve Flutteru Zdroj: <a href="https://docs.flutter.dev/get-started/codelab">https://docs.flutter.dev/get-started/codelab</a> .....	21
Obrázek 11 Možnosti upgradu Alpha Vantage API Zdroj: <a href="https://www.alphavantage.co/premium/">https://www.alphavantage.co/premium/</a> .....	22
Obrázek 12 Ukázka Alpha Vantage API Zdroj: <a href="https://www.alphavantage.co/query?function=GLOBAL_QUOTE&amp;symbol=IBM&amp;apikey=demo">https://www.alphavantage.co/query?function=GLOBAL_QUOTE&amp;symbol=IBM&amp;apikey=demo</a> .....	23
Obrázek 13 User model Zdroj: vlastní.....	25
Obrázek 14 Ukázka vypsání hodnoty balance Zdroj: vlastní .....	25
Obrázek 15 Ukázka zavolání metody setState() při přesměrování Zdroj: vlastní .....	26
Obrázek 16 Ukázka otevření souboru Zdroj: vlastní .....	26
Obrázek 17 Ukázka vytváření tabulky Zdroj: vlastní .....	27
Obrázek 18 Ukázka vkládání dat do tabulky Zdroj: vlastní.....	27
Obrázek 19 Ukázka vypisování dat Zdroj: vlastní.....	27
Obrázek 20 Ukázka třídy MyApp v projektu Zdroj: vlastní.....	29
Obrázek 21 Navigační pole.....	31
Obrázek 22 Navbar stránky akcie .....	31

Obrázek 23 Stránka Portfolio Zdroj: vlastní .....	32
Obrázek 24 Stránka Stocks Zdroj: vlastní.....	33
Obrázek 25 Stránka Orders Zdroj: vlastní .....	33
Obrázek 26 Stránka setting Zdroj: vlastní.....	34
Obrázek 27 Stránka Stock.....	35
Obrázek 28 Volné odpovědi respondentů 1 Zdroj: vlastní .....	39
Obrázek 29 Volné odpovědi respondentů 2 Zdroj: vlastní .....	40

## SEZNAM GRAFŮ

Graf 1 Odpovědi respondentů 1 Zdroj: vlastní .....	36
Graf 2 Odpovědi respondentů 2 Zdroj: vlastní .....	36
Graf 3 Odpovědi respondentů 3 Zdroj: vlastní .....	37
Graf 4 Odpovědi respondentů 4 Zdroj: vlastní .....	37
Graf 5 Odpovědi respondentů 5 Zdroj: vlastní .....	37
Graf 6 Odpovědi respondentů 6 Zdroj: vlastní .....	38
Graf 7 Odpovědi respondentů 7 Zdroj: vlastní .....	38
Graf 8 Odpovědi respondentů 8 Zdroj: vlastní .....	39
Graf 9 Odpovědi respondentů 9 Zdroj: vlastní .....	39