



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**MOBILNÍ APLIKACE PRO SPRÁVU TRANSAKČÍ
S KRYPTOMĚNAMI**

MOBILE APP FOR MANAGEMENT OF TRANSACTIONS WITH CRYPTOCURRENCIES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL BOBČÍK

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Bobčík Pavel**
Program: Informační technologie
Název: **Mobilní aplikace pro správu transakcí s kryptoměny**
Mobile App for Management of Transactions with Cryptocurrencies
Kategorie: Uživatelská rozhraní

Zadání:

1. Seznamte se s problematikou obchodování s kryptoměny, zaměřte se na problematiku vykazování, danění apod.
2. Analyzujte existující nástroje pro evidenci transakcí s kryptoměny.
3. Navrhněte a testujte s uživateli prvky uživatelského rozhraní mobilní aplikace pro evidenci transakcí s kryptoměny.
4. Implementujte řešenou aplikaci, iterativně ji testujte a vylepšujte uživatelskou zkušenost s aplikací.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN: 978-0321965516
- Steve Krug: Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability, ISBN: 978-0321657299
- Joel Marsh: UX for Beginners: A Crash Course in 100 Short Lessons, O'Reilly 2016
- Jennifer Bailey, Aldo Olivares Domingues a Dean Djemarović. Saving data on Android. McGaheysville: Razeware, 2019. ISBN 978-1-942878-89-6.
- Android Developers: <https://developer.android.com/index.html>

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodu 3.

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

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 1. listopadu 2021

Abstrakt

Cílem této bakalářské práce je vytvoření mobilní aplikace, jež má sloužit uživateli jako pomocný nástroj při obchodování s kryptoměny. Aplikace tedy cílí na uživatele, kteří se zapojují do zmíněných obchodů. Uživatel má možnost evidovat své transakce typu nákup, prodej nebo směna. Aplikace pracuje se základními informacemi jako je množství, cena a název kryptoměny a datum porízení. Na základě těchto informací provede patřičné výpočty pro správné uspořádání dle FIFO fronty. Transakce mohou být nadále upravovány. Získané informace jsou následně zpracovány. Uživatel má možnost vygenerovat PDF soubor, který slouží jako přehled jeho obchodování za zvolené daňové období. Rovněž si může zobrazit přehled vlastněných, respektive evidovaných, kryptoměn. Vzhled aplikace byl konzultován a upravován s potenciálními uživateli s cílem dosažení maximálního uživatelského prožitku a jednoduché orientace. Některé postřehy slouží jako možnosti dalšího vývoje aplikace.

Abstract

The main goal of this bachelor thesis is to create a mobile application which would serve as auxiliary tool in cryptocurrency trading. The application is therefore targeting group of people already engaged in this market. The user can store and keep track of information about his transactions, such as purchase, sale and exchange. The application works with basic information, such as quantity, price, cryptocurrency name and date of proceeding. Based on the information it makes necessary calculations for correct organization in FIFO queue. Transactions can be further modified. Newly gained information are then processed. The user has option to generate PDF file, including all the information about own transactions in selected tax period. Also the overview of the currently registered cryptocurrencies can be displayed in the application. The design and entire looks of the application was consulted with potential users with its aim for maximizing user experience and also simplify orientation. Some observations serve as opportunities for further development.

Klíčová slova

mobilní aplikace, kryptoměny, správa transakcí, Android, Android Studio, Firebase, ROOM, PDF

Keywords

mobile application, cryptocurrencies, management of transactions, Android, Android Studio, Firebase, ROOM, PDF

Citace

BOBČÍK, Pavel. *Mobilní aplikace pro správu transakcí s kryptoměny*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Mobilní aplikace pro správu transakcí s kryptoměny

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Ing. Adama Herouta Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Pavel Bobčík
8. května 2022

Poděkování

Na úvod bych rád poděkoval vedoucímu práce panu prof. Adamovi Heroutovi, který přinesl do mé práce nový úhel pohledu v podobě rad a zajímavých názorů. Dále bych chtěl poděkovat všem, kteří mi pomohli formou testování aplikace.

Obsah

1	Úvod	3
2	Přínos práce v porovnání s existujícími aplikacemi	4
2.1	Cíl práce	4
2.2	Delta	4
2.3	FTX	8
2.4	EveryTrade.io	10
2.5	Zhodnocení existujících aplikací v porovnání s přínosy práce	12
3	Co tvoří mobilní aplikaci pro Android a technologie k tomu použité	14
3.1	Z čeho se skládá mobilní aplikace na zařízení Android	14
3.2	Vývojové prostředí	15
3.3	Kritéria výběru verze SDK a následky, kterým musíme čelit	15
3.4	Material Design	17
3.5	Databáze	17
3.6	PDFBox	19
3.7	Shape Shifter	19
4	Návrh řešení	20
4.1	Problematika správy kryptoměn	20
4.2	Případy užití	21
4.3	Návrh databáze	23
4.4	Uživatelské rozhraní	25
5	Implementace	34
5.1	Správa transakcí	34
5.2	Operace FIFO nad transakcemi	36
5.3	Získání a uchování dat	37
5.4	Generování PDF dokumentu	39
5.5	Zpracování informací pro přehledný výpis	40
6	Testování a vyhodnocení	41
6.1	Testování kompatibility	41
6.2	Testování funkcionality s využitím uživatelů	42
6.3	Objevené nedostatky	43
6.4	Budoucí rozšíření aplikace	43
7	Závěr	45

Literatura	46
A Plakát	48

Kapitola 1

Úvod

Dalo by se říct, že každá generace svou dobu může označit za moderní období. Stejně tak, jako se 60. léta dají označit dobou vesmírného dobrodružství, by se naše doba mohla označit digitálním věkem. Technologie 21. století nás posouvají do zcela nových končin. Člověk má informace takřka na dosah ruky. Konkrétně na vzdálenost od palce.

Existuje nespočet mobilních aplikací, nás nicméně budou zajímat aplikace zabývající se kryptoměnami. Kryptoměnou rozumíme typ digitální měny, či nehmotnou movitou věc, s kterou lze obchodovat. Aplikace poskytující tento typ transakce mají často i svou mobilní verzi, jež se těší rostoucímu počtu uživatelů. Nelze se tedy nechat překvapit, že roste i počet uživatelů, kteří tento druh obchodu vykonávají pomocí svého mobilního telefonu. Zároveň s touto skutečností roste i poptávka po aplikacích pro správu těchto transakcí.

Jak bude uvedeno v kapitole 2, existuje již spousta aplikací zaměřených na dodatečnou správu kryptoměn. Nicméně se převážně jedná o správu portfolia. Cílem této bakalářské práce je správa již uskutečněných transakcí typu nákup, prodej a směna. Poté, co uživatel zaeviduje své transakce, může vytvořit PDF soubor za určené daňové období. Tento soubor bude obsahovat výpis s uskutečněnými transakcemi uspořádanými dle FIFO pravidla (kapitola 5.2) v daném daňovém období. Včetně všech důležitých informací. Uživatel může soubor nadále předat osobě vypracovávající daňové přiznání jako jednotný formát výpisu provedených transakcí.

Aplikace je vyvíjena pro mobilní telefony s operačním systémem Android v jazyce Java. Bylo využito vývojového prostředí Android Studio. Data jsou získávána pomocí realtime databáze (v českém znění databáze v reálném čase, dále označované pojmem realtime databáze) Firebase a API spravované společností CoinGecko. Pro ukládání dat je využita interní databáze SQLite. Podrobnější popis technologií je uveden v kapitole 3. V následujících kapitolách se seznámíme s návrhem aplikace (kapitola 4), implementací (kapitola 5) a samozřejmě testováním v kapitole 6.

Kapitola 2

Přínos práce v porovnání s existujícími aplikacemi

I přesto, že je trh s kryptoměnami relativně mladý, tak se může pyšnit rychlostí svého růstu. Jak již název napovídá, nejedná se o skutečnou měnu, nýbrž o digitální. Tyto faktory samozřejmě poskytují dobré zázemí pro velké množství aplikací zabývajících se kryptoměnami.

V následujících podkapitolách budou popsány cíle práce a zkoumány některé z existujících aplikací. Jelikož je má bakalářská práce zaměřena na mobilní zařízení, tak se bude jednat o výběr převážně mobilních aplikací. A to konkrétně aplikací Delta¹ a FTX². Nicméně zde bude uvedena i jedna webová aplikace, jež se funkcionalitou nejvíce podobá vypracované práci.

Na závěr kapitoly budou vyzdvíženy přínosy, ale i nedostatky v porovnání s vybranými aplikacemi. A samozřejmě popsány rozdíly i podobnosti, které mohou být již u uživatelů podobných aplikací vžitě.

2.1 Cíl práce

Cílem mé bakalářské práce je vytvoření mobilní aplikace pro správu transakcí s kryptoměnami. Tímto názvem je myšleno vytvoření aplikace, ve které si uživatel ukládá provedené nákupy, prodeje a směny s kryptoměnami. Při vytváření transakcí je zapotřebí uvést základní informace jako jsou koupené/prodané množství, celková cena, poplatek, datum, čas a další. Následně se uložená data zpracují metodou FIFO. Stejně tak dojde k aktualizaci FIFO fronty při vytvoření nové nebo úpravě stávající transakce. Uživatel má dále možnost generovat PDF výpis za určité daňové období. Tento výpis obsahuje seznam nákupů, prodejů a směn za dané období včetně celkového zisku či ztráty. Uživatel má rovněž přehled o vlastněných kryptoměnách.

2.2 Delta

Delta je mobilní aplikace pro operační systém Android a iOS. Již na úvodní obrazovce (obr. 2.1.a) si lze povšimnout, že se aplikace nezaměřuje výhradně na správu kryptoměn. Rovněž nabízí správu indexů, akcií, fondů, forexu a NTFs. Význam jednotlivých investičních možností nebude dále popisován, jelikož se nejedná o náplň bakalářské práce.

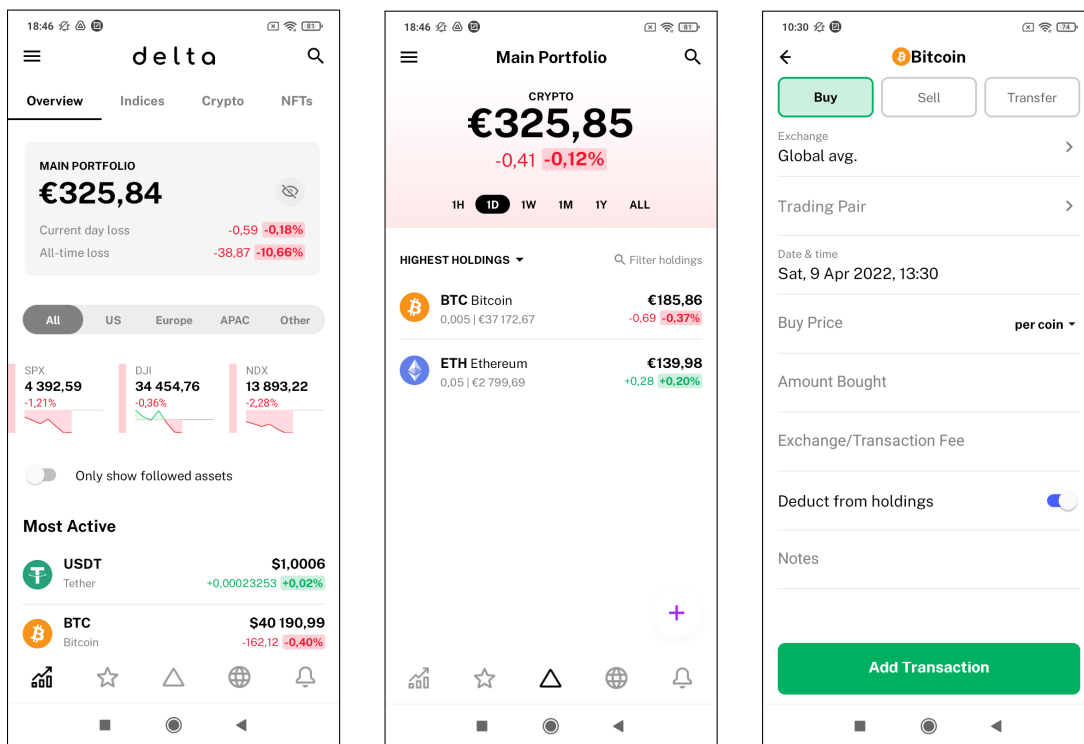
¹<https://delta.app/en>

²<https://ftx.com/intl/apps>

Aplikace je rozdělena do pěti základních fragmentů. Úvodní obrazovka se skládá z prvotního fragmentu pro přehled (obr. 2.1.a) a zbylých podfragmentů pro jednotlivé typy investic. Přehled poskytuje uživateli základní informace. Uživatel zde vidí stav svého portfolia, tedy jeho hodnotu, denní a celkovou ztrátu nebo zisk. Dále se zde zobrazují stručné přehledy různých typů. Mezi tyto typy patří například přehled největšího nárůstu nebo naopak poklesu hodnoty. Dále i přehled pro jednotlivé druhy investic, které aplikace nabízí, seřazeny podle market ranku.

Další fragment umožňuje sledování zvolených investic. Poskytuje rozsáhlý filtr uspořádání dle různých kritérií a vypočtený za časový interval. Základní přehled se skládá ze seznamu vyobrazujícího název kryptoměny, akcie nebo jiného produktu včetně hodnoty a její změny za dané období.

Přechod na další fragment, vyobrazený na obrázku 2.1.b, poskytne uživateli hlavní funkci aplikace, kterou je správa portfolia v aktuálním čase. Zobrazuje vlastněné kryptoměny, akcie a další produkty včetně množství, ceny, celkové hodnoty investice a podobně ve zvoleném časovém intervalu. Rovněž uživatele informuje o celkové hodnotě portfolia a jak se danému portfoliu daří. Tedy zdali je ziskové nebo naopak ztrátové. Portfolio je tvořeno na základě přidanych transakcí (obr. 2.1.c). Operace lze uvádět v i českých korunách.



(a) Úvodní obrazovka.

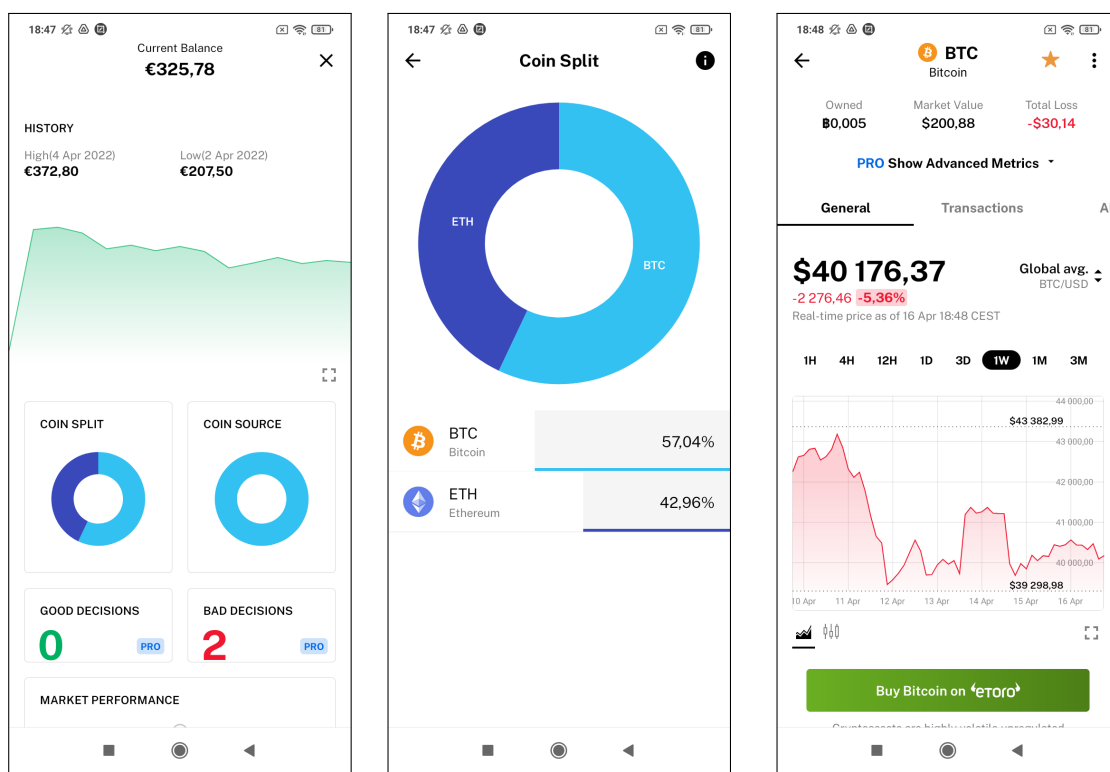
(b) Obrazovka portfolia.

(c) Přidání transakce.

Obrázek 2.1: Na prvním obrázku je vyobrazena úvodní obrazovka aplikace s informací o portfoliu, volbou investičních možností a různými typy uspořádání investičních produktů. Následující obrázek znázorňuje přehled portfolia. Poslední obrázek vyobrazuje okno pro přidání transakcí, které dané portfolio utváří.

Mimo popsané fragmenty se zde ještě vyskytují fragmenty s ekonomickými novinkami ze světa investic a historií notifikací. Aplikace dále poskytuje i grafické zobrazení portfolia na obrázku 2.2.a, včetně grafu znázorňujícího rozložení portfolia mezi investicemi (obr. 2.2.b).

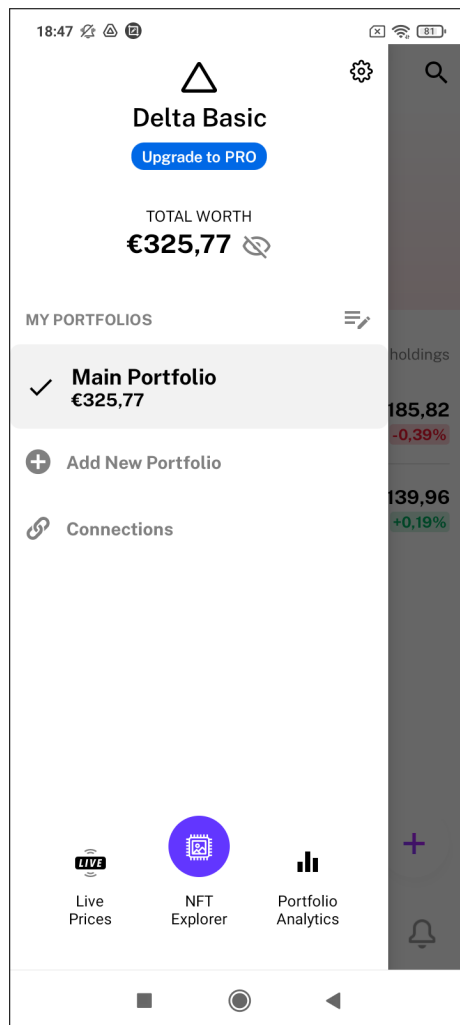
Dále jsou k dispozici informace k jednotlivým investičním produktům. Tyto informace obsahují graf znázorňující vývoj ceny v určitém intervalu, tržní hodnotu i zprávy ze světa, jak si lze povšimnout na obrázku 2.2.c.



(a) Grafický náhled portfolia. (b) Graf rozdělení portfolia. (c) Výpis měny Bitcoin.

Obrázek 2.2: První obrázek zobrazuje snímek aplikace s grafickým znázorněním portfolia. K vidění je zde graf vývoje hodnoty portfolia v čase, včetně nejvyšší a nejnižší hodnoty. Dále graf druhého obrázku znázorňující rozložení portfolia mezi všechny investice. Poslední obrázek je ukázka obrazovky zobrazující aktuální informace o daném investičním produktu.

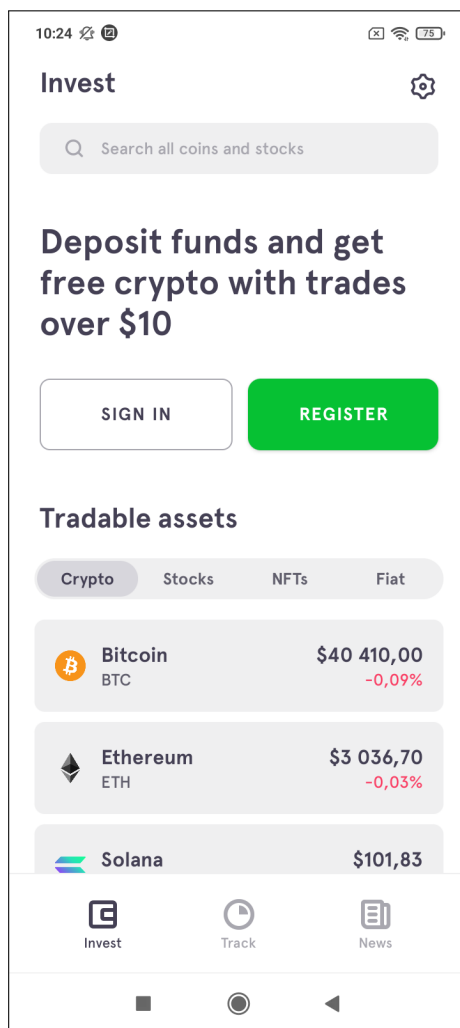
Z průzkumu vyplývá, že se jedná o skutečně rozsáhlou aplikaci nabízející širokou škálu funkcí. Uživatelé zajisté nezklame ani uživatelský prožitek (často označován anglickým názvem user experience - UX). A to i přesto, že například grafické znázornění portfolia není přístupné přes fragment portfolia, ale v rozšířených možnostech (obr. 2.3). V celkovém hodnocení aplikace obstála velmi dobře a nejsou zde důvody vytvářet práci na stejné téma. Nicméně, dojde-li k porovnání funkcí zkoumané aplikace s cíli mé bakalářské práce, zjistíme že se liší. Zkoumaná aplikace poskytuje správu portfolia v aktuálním čase, včetně zpráv ze světa a cen jednotlivých kryptoměn. Kdežto cílem mé práce je evidence provedených transakcí a výpis těchto operací za dané období.



Obrázek 2.3: Přístup ke grafickému zobrazení portfolia přes rozšířené možnosti.

2.3 FTX

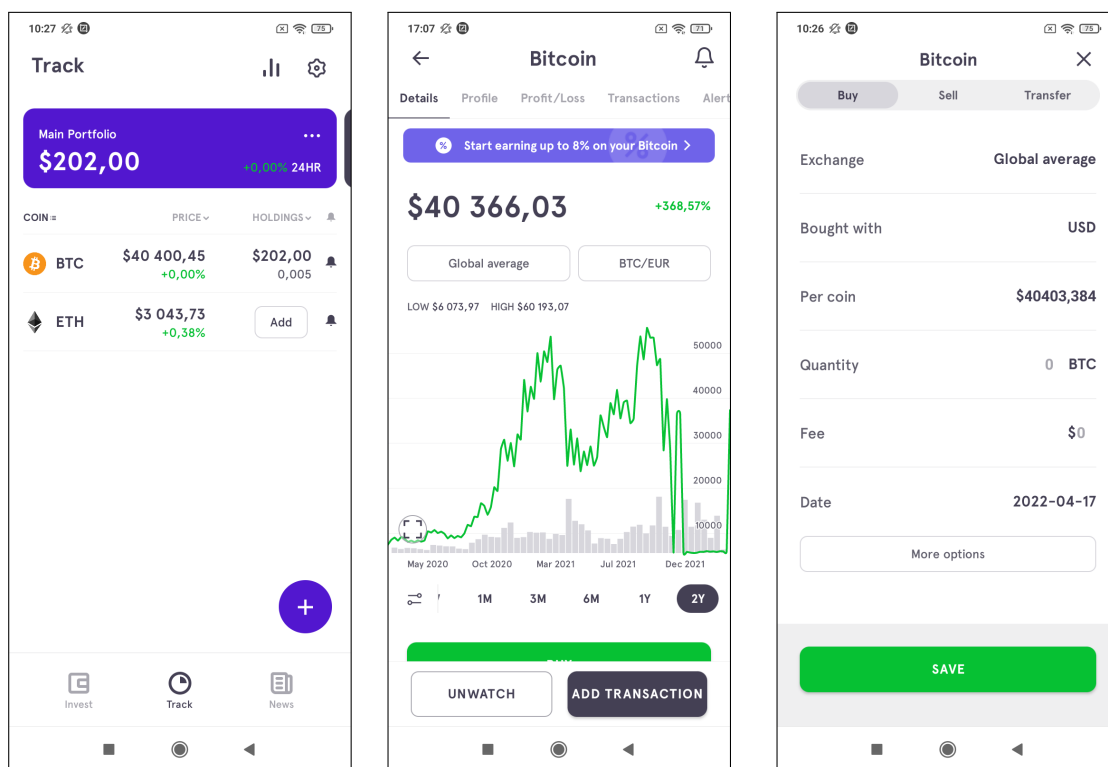
FTX je další oblíbenou mobilní aplikací pro správu kryptoměn, dříve známou pod názvem Blockfolio. Je rovněž rozdělena do fragmentů jako aplikace Delta. I tato aplikace poskytuje přehled portfolia, více druhů investičních možností, a dokonce i novinky ze světa investic. Na rozdíl od aplikace Delta nabízí i možnost nákupu. Tato možnost se v předchozí aplikaci vyskytuje jen za využití aplikace třetí strany. V aplikaci FTX se stačí pouze registrovat, jak si lze všimnout na obrázku 2.4.



Obrázek 2.4: Úvodní okno aplikace FTX s výzvou k přihlášení pro zpřístupnění možnosti nákupu.

Dále je zde přehled portfolia (obr. 2.5.a). V porovnání s aplikací Delta zobrazuje méně informací. Vývoj hodnoty portfolia, tedy zdali je v zisku či ztrátě, je zde pouze v ohledu jednoho dne. Dále obsahuje pouze graf historického vývoje hodnoty, nikoliv již graf zobrazující procentuální rozložení investic v rámci portfolia. Na druhou stranu aplikace poskytuje nastavení notifikací ohledně významné změny cen a dalších informací pro jednotlivé investiční produkty. Nicméně, uživatelský prožitek může být narušen skutečností, že jsou již na počátku nastaveny určité investiční produkty jako sledované. Ty se následně zobrazují v pře-

hledu portfolia, i když nejsou vedeny jako vlastněné, a tedy mohou na první pohled působit matoucím dojmem. Navíc odstranění těchto produktů ze seznamu nemusí být intuitivní, jelikož se jednotlivé produkty odebírají v přehledu daného produktu, jak je znázorněno na obrázku 2.5.b. Zbylé možnosti jsou si s předchozí aplikací podobné, a to včetně přidávání transakcí (obr. 2.5.c), které rovněž podporuje i českou měnu.



(a) Obrazovka s portfoliem.

(b) Popis kryptoměny.

(c) Přidání transakce.

Obrázek 2.5: První obrázek obsahuje přehled portfolia včetně sledovaných, v případě vytvoření transakce i vlastněných, investičních produktů. Druhý obrázek zobrazuje popis investičního produktu včetně možnosti započítání nebo ukončení sledování daného produktu. Na posledním obrázku je obrazovka pro přidání transakce.

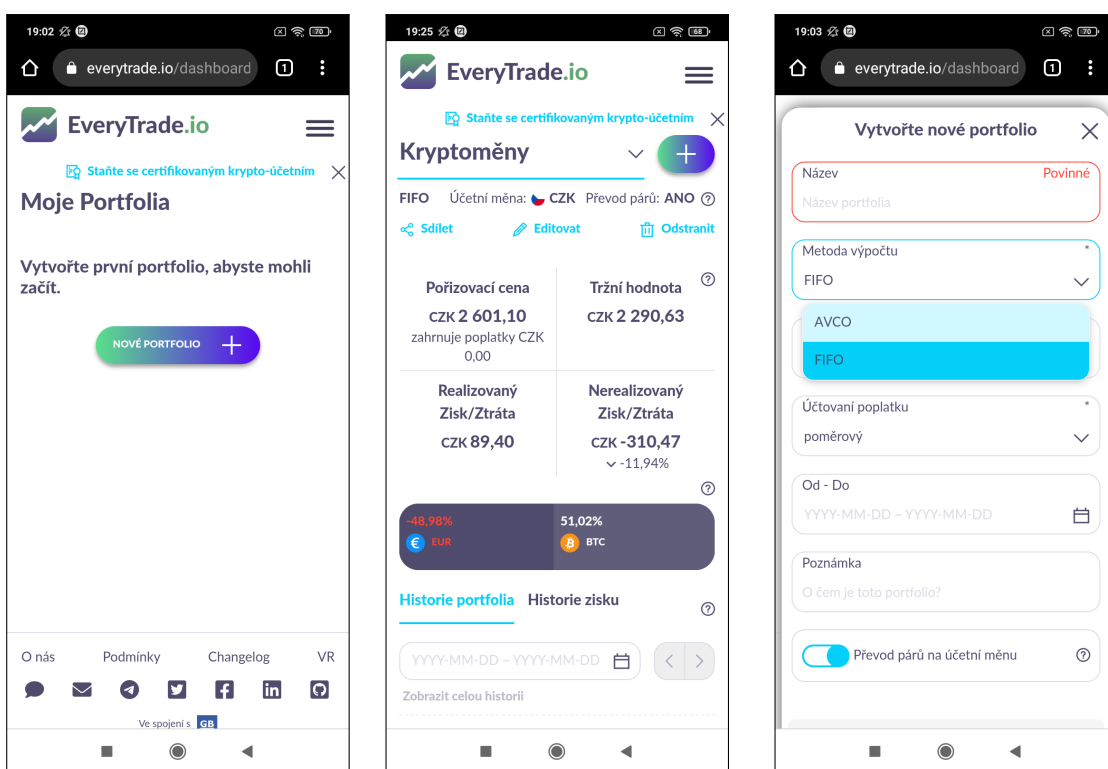
Další podobností je výpis novinek a informací o jednotlivých investičních produktech (obr. 2.5.b). Na rozdíl od předchozí aplikace nabízí zkoumaná aplikace uživateli způsob vykreslení grafu vývoje ceny.

Z finálního hodnocení vyplývá, že se aplikace od předchozí liší pouze v maličkostech. Přesto ji v určitém hledisku doplňuje a v jiném naopak zaostává. Aplikace FTX, na rozdíl od aplikace Delta, poskytuje i nákup kryptoměny. To každopádně není hlavním cílem hledané aplikace, kterým je evidence transakcí. Naopak neposkytuje grafické rozložení portfolia. Nicméně, stejně jako předchozí aplikace, i tato se zaměřuje na aktuální hodnotu portfolia a kryptoměn, nikoliv na evidenci již provedených transakcí a jejich výpis za určité časové období. Opět se tedy nepodařilo najít aplikaci, která by splňovala cíle mé bakalářské práce.

2.4 EveryTrade.io

Jelikož stále nedošlo k nalezení aplikace splňující dané cíle, hledání pokračovalo. Průzkum byl tedy rozšířen i mimo mobilní platformu. V tomto okamžiku se již podařilo najít aplikaci splňující stanovené cíle. Aplikace EveryTrade.io³ je webovou aplikací, jenž splňuje funkčnost mé bakalářské práce, jak bude v následující kapitole popsáno.

Jedná se o placenou službu, která má rovněž omezenou verzi zdarma, jež poskytuje až deset tisíc záznamů transakcí. Po bezplatné registraci a přihlášení je uživateli naskytuta možnost vytvořit nové portfolio (obr. 2.6.a), nemá-li žádné. V opačném případě dojde k zobrazení již vytvořeného (obr. 2.6.b). Při vytváření portfolio má uživatel na výběr metodu výpočtu AVCO nebo FIFO (obr. 2.6.c). Pro naše účely byla zvolena metoda FIFO. Dále je zde možnost časového ohraničení portfolio. V takovém případě se do portfolio nedostane transakce spadající pod jiný časový interval.



(a) Portfolio není vytvořeno.

(b) Přehled portfolio.

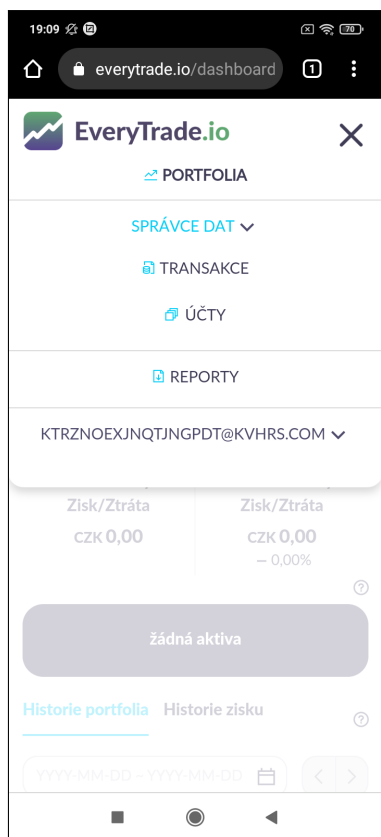
(c) Volba metody výpočtu.

Obrázek 2.6: První obrázek znázorňuje okno portfolio v případě, že žádné zatím nebylo vytvořeno. Na druhém obrázku má již uživatel portfolio vytvořeno, včetně evidovaných transakcí. Lze si všimnout informací jako jsou pořizovací cena nebo skutečný zisk. Poslední obrázek ukazuje možnost výběru metody výpočtu při vytváření portfolio.

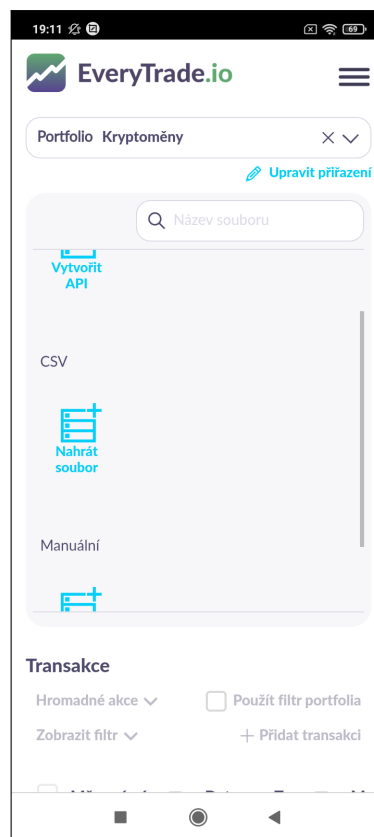
Poté, co je portfolio vytvořeno, může začít přidávání transakcí. Přidání první transakce může zabrat o něco delší čas. Toto je způsobeno skutečností, že se tato možnost nevyskytuje v přehledu portfolio, ale v rozšířených možnostech a tam následně pod sekci **Správa dat** (obr. 2.7.a). V této sekci jsou na výběr tři možnosti kontejnerů, do kterých budou určitým způsobem nahrány transakce (obr. 2.7.b). Prvním způsobem je API, jež získá data

³<https://everytrade.io/>

z jedné z mnoha podporovaných burz. Dále může uživatel nahrát CSV soubor obsahující data transakce. A nakonec manuální přidání. V případě vytvoření manuálního kontejneru je potřeba jednotlivé transakce přidat. Na výběr je typ obchodu: nákup, prodej, poplatek, sleva, vklad a výběr. Dále je vyžadován datum a čas nákupu, množství a celková částka.



(a) Přechod na správce dat a reporty.

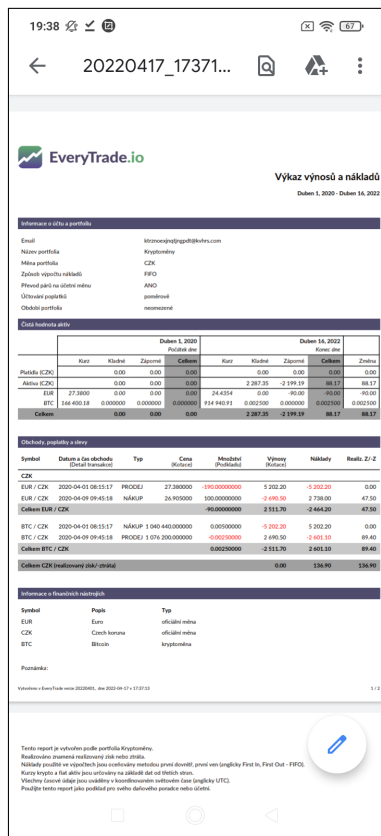


(b) Typy kontejnerů dat.

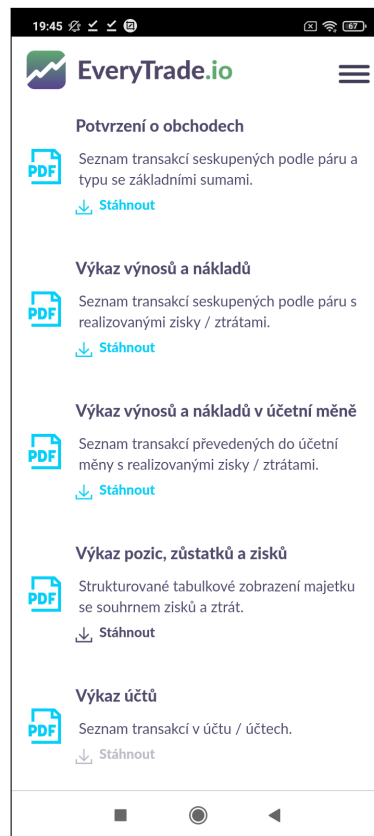
Obrázek 2.7: První obrázek zobrazuje rozšiřující výběr, též známý jako hamburger menu, pro přechod na správce dat nebo reporty. Na druhém obrázku jsou vyobrazeny typy kontejnerů dat pro vytváření transakcí.

Jakmile jsou transakce vytvořeny a uživatel se navrátí zpět do portfolia, uvidí zde výpis informací jako jsou pořizovací cena prodaného množství nebo realizovaný zisk/ztráta (obr. 2.6.b). Dále je zde graf historie portfolia zobrazující vývoj nákupů a prodejů uživatele. Uživatelem tedy vidí, zdali prodeje uskutečnil ve správný moment, nebo zdali by byl jeho zisk vyšší, pokud by vyčkal do určitého data.

Nechybí ani generování PDF přehledu za dané období, který si lze prohlédnout na obrázku 2.8.a. Ten je vygenerován automaticky a stačí jej pouze stáhnout v sekci Reporty pod názvem Výkaz výnosů a nákladů v účetní měně. Kromě daného přehledu poskytuje služba i potvrzení o obchodech, výkaz výnosů a nákladů a další, jak lze vidět na obrázku 2.8.b.



(a) PDF výpis.



(b) Typy PDF výpisů.

Obrázek 2.8: První obrázek zobrazuje PDF soubor s výkazem výnosů v účetní měně, jenž byl vygenerován aplikací. Na druhém obrázku jsou znázorněny typy PDF výpisů ke stažení.

2.5 Zhodnocení existujících aplikací v porovnání s přínosy práce

Jak již bylo zmíněno, existuje spousta mobilních aplikací pro správu kryptoměn. Nicméně se jedná o aplikace spravující aktuální hodnotu kryptoměn a informující uživatele o aktuálních cenách a dění ve světě investic. Aplikace, i přes své rozsáhlé služby, nesplňují cíle bakalářské práce a budou tedy nadále porovnávány pouze prvky využívající podobné principy.

Kupříkladu vytváření portfolia je založeno na přidávání transakcí, stejně jako v mé práci. Obě zmíněné aplikace, tedy Delta a FTX, požadují od uživatele stejné informace, jako má práce. Jedná se o základní údaje pro správu transakcí, jako jsou množství, cena, poplatek, měna a čas. Uživatelé, kteří již používali minimálně některou ze zkoumaných aplikací, budou tedy obeznámeni s údaji, které mají vyplnit. Dále došlo na základě zkoumání k inspiraci funkcionality. Jedná se o využití uložených dat pro vytvoření přehledu vlastněných kryptoměn. Podobnost lze nalézt i u základního uspořádání kryptoměn podle jejich market ranku nebo přidání kryptoměn do seznamu oblíbených.

Třetí ze zkoumaných aplikací je aplikace EveryTrade.io. V tomto případě se jedná o webovou aplikaci, která nemá svou mobilní verzi. Aplikace svými službami poskytuje

stejnou funkci jako vypracovaná bakalářská práce. Jedná se tedy o vynikající příklad k porovnání aplikací a zjištění přínosů a nedostatků mé práce.

Jedním z přínosů mé práce je skutečnost, že se jedná přímo o mobilní aplikaci. Aplikace je tedy uzpůsobena pro mobilní telefony a není omezoována funkcemi různých webových prohlížečů na mobilní zařízení. Uvedeme-li si některé nevýhody této závislosti, tak se může jednat kupříkladu o špatné zobrazení okna. Například v prohlížeči Safari nebylo dosažitelné tlačítko k uložení transakce. V prohlížeči Google Chrome byl problém s filtrováním výběru kryptoměn, pokud již byla provedena volba měny. V tomto okamžiku nebylo možné daný text upravit. Toto je pouze ukázka pár problémů, které mohou vzniknout použitím různých webových prohlížečů na mobilním telefonu.

Jako další přínos lze označit orientaci v aplikaci. Práce byla již od počátku navrhována s potenciálními uživateli a byla zde snaha o vytvoření intuitivního rozhraní. Naopak od toho bylo první použití aplikace EveryTrade.io mírně zmatečné. Vytvoření transakce u portfolia není možno provést přímo z přehledu portfolia. Tato možnost se nachází v sekci **Správa dat**, která je umístěna v tzv. hamburger menu, tedy v nabídce schované běžně pod třemi čarami. Nicméně po prvním použití uživatel již ví, kde má danou funkci hledat.

Samozřejmě jsou místa, na kterých má aplikaci ztrácí. Prvním příkladem může být přidávání transakcí. Má práce v aktuálním stavu poskytuje pouze manuální přidávání, na rozdíl od zkoumaných aplikací, které jsou propojeny s různými burzami pomocí API. EveryTrade.io navíc nabízí přidání transakcí i pomocí CSV souboru. Zkoumané aplikace rovněž nabízejí i doplňkové funkce jako jsou různé grafy zkoumající prodej nebo znázorňující rozdělení portfolia.

Jelikož má aplikace EveryTrade.io stejný účel, tak zde dochází k určitým podobnostem. Opět se zde nacházejí stejné údaje při vytváření transakce. Rovněž obě aplikace poskytují generování PDF výpisu. Další shodou je výpočet skutečného zisku/ztráty. Jedná se o očekávanou skutečnost, jelikož jsou popisovány funkce, které uživatel poptává od daného typu aplikace. Je tedy potřeba se zaměřit na mezery druhé aplikace a ty se snažit vyplnit.

Kapitola 3

Co tvoří mobilní aplikaci pro Android a technologie k tomu použité

V této kapitole budou popsány základní stavební kameny mobilní aplikace na Android a technologie k tomu použité. Budou upřesněny funkce vývojového prostředí, včetně způsobu sestavení balíčku aplikace a nedávná změna, kterou společnost Google zavedla. Dále bude vysvětlena problematika výběru verze SDK a následky, které tato volba obnáší. Rovněž budou popsány využití druhy databází. Na závěr dojde k popisu nástroje pro tvorbu PDF dokumentu a animovaných ikon.

3.1 Z čeho se skládá mobilní aplikace na zařízení Android

Základním stavebním kamenem aplikace na zařízení s operačním systémem Android jsou aktivity a fragmenty. Aktivita umožňuje uživateli interagovat s aplikací, jelikož je zodpovědná za vykreslení okna grafického rozhraní. Reaguje na podstatné události, které mohou při používání aplikace nastat, jako jsou `onClick`, `onCreate`, `onResume`, `onPause` a další. Jedná se tedy o události životního cyklu aplikace, které sledují, zdali byla vytvořena, navrácena na popředí a podobně. V module MVC (Model View Controller) by se dala třída `Activity` označit jako část spadající pod `View`. Tedy část, jež se stará o interakci s uživatelem a požaduje vykonání funkce po `Controlleru`. Uvedený model je využit v mé práci popsaným způsobem.

Další možností je `fragment`, jenž tvoří část uživatelského rozhraní v aktivitě. Umožňuje rozdělit jednu aktivitu do více oken, mezi kterými je typicky přepínáno pomocí panelu akcí nebo navigace. `Fragment` má svůj vlastní životní cyklus. Ten je nicméně závislý na nadřazené aktivitě, která jej spravuje a bez které nemůže vzniknout. Tato aktivita obsahuje komponentu `FrameLayout` do které je `fragment` vykreslován [13].

Dalším typickým prvkem reprezentující aplikace pro operační systém Android jsou ikony. Na rozdíl od aplikací pro iOS, které využívají spíše textového popisu. Rovněž se od iOS aplikací liší i využitím hardwarového vybavení u zařízení s operačním systémem Android. Mezi tato vybavení spadá i tlačítko `Zpět`, které není třeba explicitně v aplikaci implementovat.

3.2 Vývojové prostředí

Android Studio je integrované vývojové prostředí (též známé pod anglickou zkratkou IDE, neboli Integrated Development Environment) k vývoji aplikací pro operační systém Android. Prostředí bylo představeno v roce 2013 společností Google, která jej nadále spravuje. Je založeno na vývojovém prostředí IntelliJ IDEA od společnosti JetBrains [4]. Mimo jiné poskytuje nástroj pro automatizované sestavení zvaný Gradle, jenž obsahuje informace jako jsou závislosti (anglicky známé pod pojmem dependencies) či verze SDK [7]. Dále svým uživatelům poskytuje možnost využít emulátor zařízení s operačním systémem Android, který lze stáhnout přímo ve vývojovém prostředí. Testování aplikací může tedy probíhat na virtuálním stroji bez využití služeb třetích stran. Mezi další výhody patří propojení se službami společnosti Google. Například služby Firebase lze propojit s projektem pár snadnými kroky. Rovněž lze využít služby Material Design poskytující mimo jiné knihovnu ikon. Android Studio nabízí i prostředí pro vývoj uživatelského rozhraní.

Projekt je rozdělen do složek `manifest`, jež obsahuje soubor `AndroidManifest.xml` s metadaty aplikace, `java`, obsahující zdrojové soubory aplikace, a `res`, obsahující všechny nekódové soubory. Mezi nekódové soubory patří typicky soubory formátu XML, jako například okna grafického rozhraní, barvy, texty a další [4].

Sestavená aplikace je do zařízení nainstalována z balíčku APK (Android Packages) obsahující zkompilovanou verzi aplikace. Pro publikování aplikace se využívá balíček AAB (Android App Bundle). Dříve Google na své platformě Google Play podporoval i balíček APK, nicméně od srpna 2021 musí být nové aplikace publikovány právě pomocí balíčku AAB. Jedná se o formát publikování, který obsahuje zkompilovaný kód a zdroje. Samotné generování a poskytování optimalizovaných souborů APK probíhá až na platformě Google Play, vizuálně je tento postup znázorněn na obrázku 3.1. Uživatel obdrží pouze kód a zdroje potřebné pro konkrétní zařízení [16][4].



Obrázek 3.1: Obrázek znázorňující cestu sestaveného balíčku aplikace od vývojáře k uživateli. Vývojář vytvoří AAB balíček, který je umístěn na platformu Google Play. Z té uživatel stáhne APK balíček obsahující zkompilovaný kód a zdroje přizpůsobené na dané zařízení. Obrázek byl převzat z [14].

3.3 Kritéria výběru verze SDK a následky, kterým musíme čelit

Android SDK (v anglickém znění Software Development Kit) neboli sada pro vývoj softwaru obsahuje prostředky jako jsou knihovny, ladící nástroje, emulátory a další [15]. Jedná se o důležitou součást projektu, jejíž výběr následně ovlivňuje celou aplikaci. Při výběru

minimální podporované verze SDK je zapotřebí brát v potaz dva hlavní faktory. Prvním faktorem je procentuální zastoupení zařízení běžících na odpovídající či vyšší verzi Androidu. Úroveň API (neboli verze SDK) je s verzí Androidu úzce spjatá, jelikož odpovídá určité verzi Androidu. Uvedené informace jsou zobrazeny na obrázku 3.2. V mé práci je například využita minimální úroveň 23, tedy minimální podporovaná verze operačního systému Android je Marshmallow, značená rovněž jako Android 6. Nabízí se zde samozřejmě možnost použití takové úrovně, která pokrývá nejvyšší procento dostupných zařízení. Nicméně je třeba myslet i na druhé kritérium, kterým je míra podpory různých knihoven a funkcí. Čím starší verze Androidu, tedy čím nižší úroveň API, tím menší podporu bude poskytovat. Výběr správné verze SDK je tedy určitým způsobem balancování mezi počtem zařízení, která budou schopna používat výslednou aplikaci a možnostmi, jež nám zvolená verze poskytne.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,8%
4.3 Jelly Bean	18	99,5%
4.4 KitKat	19	99,4%
5.0 Lollipop	21	98,0%
5.1 Lollipop	22	97,3%
6.0 Marshmallow	23	94,1%
7.0 Nougat	24	89,0%
7.1 Nougat	25	85,6%
8.0 Oreo	26	82,7%
8.1 Oreo	27	78,7%
9.0 Pie	28	69,0%
10. Q	29	50,8%
11. R	30	24,3%

Obrázek 3.2: Obrázek znázorňuje graf jednotlivých verzí SDK. Z grafu lze vyčíst verzi a název Androidu, odpovídající úroveň API a procentuální zastoupení zařízení, která jsou schopna spustit aplikaci od dané verze. Jelikož je má bakalářská práce vyvíjena pro zařízení s minimální verzí Android 6, je i tato verze v grafu znázorněna. Graf lze vygenerovat ve vývojovém prostředí Android Studio při vytváření nového projektu.

V případě, že narazíme na problém druhého kritéria, tedy výskyt funkce, jenž není podporována zvolenou verzí SDK, můžeme postupovat třemi způsoby. Prvním a rovněž nejméně obtížným způsobem je danou funkci nevyužít. Nejedná se nicméně o dobrý postup, usoudíme-li, že existují i lepší způsoby. Navíc může nastat situace, kdy je daná funkce nezbytná. Druhým způsobem je změna verze SDK. Není-li v takovém případě funkce podporována, je zapotřebí změnit minimální verzi SDK alespoň na úroveň od níž je podporována. Toto může být nicméně drastické řešení, které zasahuje do již zmíněného balancování při výběru mezi počtem zařízení, které mohou aplikaci spustit, a funkcemi, které API poskytuje. A proto se zde vyskytuje i třetí možnost. Skutečnost že není nějaká funkce podporována nemusí znamenat, že na nižší verzi neexistuje. Často došlo pouze k její úpravě, ať už z důvodu opravy chyby či změny technologie. Toto řešení tedy počítá s podporou dané funkcionality i na zvolené verzi SDK, pouze pod jinou funkcí. Podaří-li se takovou funkci najít, je potřeba aplikaci určit, kdy ji má použít. K tomuto slouží podmínka uvedená v následující ukázce kódu zobrazeného ve výpise 3.1 [11].

```
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.N){
    // Vykonání funkce u zařízení s verzí
    // Android N (API úrovně 24) a vyšší
} else{
    // Jinak dojde k vykonání funkce uvedené zde
}
```

Výpis 3.1: Vykonání akce na základě SDK verze zařízení.

3.4 Material Design

Material Design je takzvaným designovým jazykem vytvořeným společností Google v roce 2014. Je inspirován objekty fyzického světa jako jsou papír a inkoust. Způsobem, jakým odráží světlo či vrhají stíny. I přesto, že byl původně zamýšlen pro aplikace na Android, tak se tento styl rozšířil i za jeho hranice. Knihovny komponent jsou dostupné pro Android, iOS, Flutter a web. Specifikuje podmínky jako jsou typografie, mřížky rozložení, měřítko, barva a další. Tímto způsobem pomáhá vývojářům, jenž se rozhodli dodržet dané principy, k docílení vzhledu aplikace působícího téměř jako nativní aplikace společnosti Google [2][1].

Mezi dále poskytované služby patří ikony typické pro aplikace na zařízení Android. Rovněž poskytuje i různá grafická témata, jako jsou například dialogová okna, okno pro výběr data či času a další komponenty, jež jsou popsány na stránkách Material Design¹.

3.5 Databáze

Databáze je způsob uchování dat. Ta jsou uchována v perzistentní podobě. Data tedy přetrvávají během programu, jinak řečeno jsou zachována i po jeho vypnutí [17][slide 3]. Častým typem databáze jsou takzvané relační databáze. U tohoto typu jsou data reprezentována v tabulkách [12]. Nad těmito tabulkami provádíme dotazy, díky kterým můžeme data přidávat, editovat, mazat, vyhledávat a podobně. Mají ovšem i svá omezení, která je třeba dodržet. Jedná se o integritu a konzistenci dat. Integritní omezení zajišťují, že skutečná fyzická data uložená v databázi budou správná, například unikátní rodné číslo [18][slide 3].

¹<https://material.io/components?platform=android>

Konzistence dat znamená, že data nebudou v rozporu. Například *převod částky z účtu A na účet B, kdy je součet na obou účtech jiný než před převodem* [17][slide 10-11].

Firestore

Firestore je další službou spravovanou společností Google, která ji převzala v roce 2014, poskytující služby pro Android, iOS, web a Unity [5]. Využívá konceptu NoSQL, též označovaného jako nerelační databáze. Tento koncept se liší od předchozího popisu databáze v tom, že nevyužívá tabulkové schéma [10]. V případě Firestore se pro uchování dat využívá formát JSON.

Firestore poskytuje řadu služeb ulehčující práci vývojáře. Mezi tyto služby patří analýza uživatelské aktivity, autentizace, realtime databáze, cloudové úložiště pro velké soubory jako obrázky, audio nebo video, notifikace a mnoho dalších² [5].

ROOM

Jednou z možností, jak ukládat data na zařízení s operačním systémem Android, je využití relační databáze SQLite. Jak již bylo zmíněno, relační databáze pracuje s tabulkovým schématem a s daty můžeme pracovat pomocí SQL dotazů. Práci s daty nadále usnadňuje knihovna ROOM, které poskytuje abstraktní vrstvu nad SQLite. Dokumentace pro Android, spravovaná společností Google, dokonce přímo doporučuje využití této knihovny namísto přímých SQL dotazů.

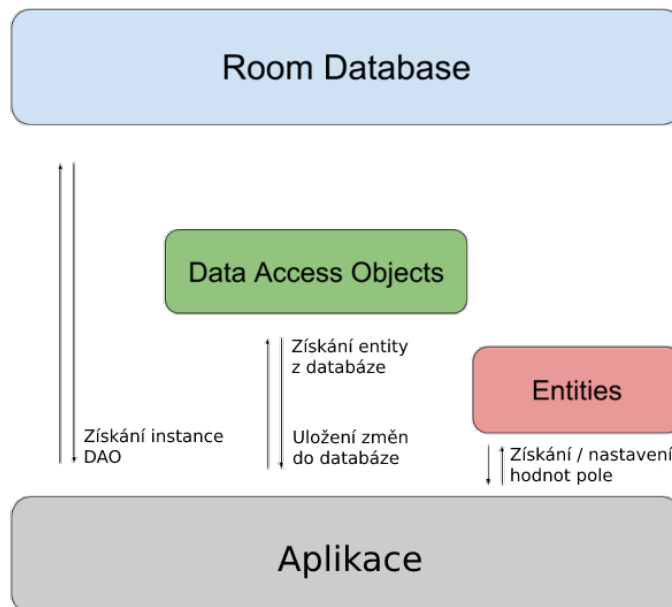
Knihovna ROOM vychází z principu ORM (objektově relačního mapování), tedy možnosti psát SQL dotazy pomocí objektově orientovaného jazyka. Snižuje množství standardního kódu náchylného k chybám a ověřuje SQL dotazy při kompilaci. ROOM se skládá ze tří hlavních částí. První je abstraktní třída `Database` definující seznam entit a objektů pro přístup k datům v databázi. Další částí jsou třídy jednotlivých `Entities` reprezentující tabulky databáze. Poslední částí je rozhraní DAO nebo `Data Access Objects` obsahující metody pro přístup k datům.

Třída `Database` poskytuje aplikaci instanci rozhraní DAO. Tato instance je nadále využívána aplikací ke správě dat v databázi. S daty pracujeme jako s objekty za využití jednotlivých tříd `Entity`. Schéma na obrázku 3.3 znázorňuje architekturu knihovny ROOM. Kód na výpisu 3.2 zobrazuje abstrakci nad SQL dotazem, zapsanou v rozhraní DAO, pro vložení záznamu tabulky do databáze [3].

```
@Insert
void insertToEntity(EntityName myEntity);
```

Výpis 3.2: Ukázka abstrakce SQL dotazu insert.

²<https://firebase.google.com/products-build>



Obrázek 3.3: Schéma architektury ROOM znázorňující získání instance DAO přidružené databáze. Dále aplikaci využívající této instance pro získání entit, jež reprezentují tabulky databáze. Rovněž se využívá pro úpravu či smazání tabulky. A entit, ke kterým se přistupuje jako k objektům. Obrázek byl převzat z [3].

3.6 PDFBox

PDFBox je open-source knihovna napsána v programovacím jazyce Java pro práci s PDF soubory. Pro účely práce byla využita možnost vytváření nových PDF dokumentů. Dále poskytuje manipulaci s již existujícími dokumenty, přidávání záložek či extrahování obsahu [6].

Původní knihovna PDFBox není podporována aplikacemi pro mobilních zařízení s operačním systémem Android. Existuje nicméně knihovna, které vychází z původní knihovny a je přístupná pod licencí Apache 2.0. Tato knihovna podporuje stejné funkce a příkazy jako originální knihovna. Jedná se pouze o určité rozšíření knihovny na Android. Knihovna je dostupná v GitHub repozitáři uživatele TomRoush³.

3.7 Shape Shifter

*Shape Shifter*⁴ je webová aplikace, jež zjednodušuje vytváření animací ikon pro Android, iOS a web [9]. Aplikace poskytuje nejrůznější nastavení jako jsou změna měřítka, polohy, přiblížení a neposledně i rotace. Dále nastavení časového intervalu animace i jednotlivých částí. Výslednou animaci lze exportovat do různých formátů, včetně `animated vector drawable` využívaného operačním systémem Android.

³<https://github.com/TomRoush/PdfBox-Android>

⁴<https://shapeshifter.design/>

Kapitola 4

Návrh řešení

Před samotnou implementací projektu bylo potřeba nejdříve vypracovat návrh aplikace. Bylo zapotřebí zjistit postup evidování transakcí s kryptoměny, včetně postupu výpočtu zisku či ztráty za daňové období. Poté co byly zjištěny informace nutné ke správné evidenci, došlo k vypracování případu užití. Tedy jaké údaje a jakým způsobem budou od uživatele požadovány. Dále jaké funkce uživatel požaduje a užitečné funkce, které jsou již využívány existujícími aplikacemi.

Poté co byl vytvořen seznam poskytovaných služeb a zjištěny informace, které budou od uživatele požadovány, bylo potřeba vyřešit způsob uchování dat. A samozřejmě návrh uživatelského rozhraní přizpůsobeného zpětné vazbě od potenciálních uživatelů.

4.1 Problematika správy kryptoměn

Správa transakcí je založena na třech základních operacích. Nákup, prodej a směna. Aplikace podporuje tři fiat měny, jimiž jsou česká koruna, euro a americký dolar. Euro a dolar jsou nejčastěji uváděné fiat měny u hodnoty kryptoměn a samozřejmě je zde uvedena i česká koruna, jakožto národní měna. Rovněž jsou do ní později přepočteny všechny transakce.

Pro evidenci nákupu je zapotřebí: nakoupená kryptoměna, nakoupené množství, celková cena nebo cena bez poplatku, poplatek, datum a čas provedení a měna, ve které je transakce evidována. Poplatek je důležité uvést, aby bylo možné dopočítat cenu bez poplatku nebo naopak celkovou cenu, není-li od uživatele vyžadováno vyplnění obou údajů.

Pro evidenci prodeje jsou zapotřebí stejné informace. Pouhým rozdílem je typ transakce. Místo nakoupené kryptoměny jsou tedy zapotřebí informace o prodané kryptoměně a prodaném množství.

Směna vyžaduje: nabytou kryptoměnu (tedy nakoupená kryptoměna), směněnou kryptoměnu (tedy prodaná kryptoměna), nakoupené a prodané množství, poplatek, měnu, ve které je poplatek evidován, datum a čas.

Evidované kryptoměny je zapotřebí uchovávat v pořadí určeném FIFO frontou dle času provedení transakce. Dojde-li k prodeji kryptoměny, je zapotřebí vypočítat rozdíl mezi nákupem a prodejem z nejstaršího dostupného nákupu. Uvedeme-li si příklad: uživatel nakoupil dva Bitcoinu 31. prosince 2019 a jeden Bitcoin 2. února 2020. Poté se 3. února rozhodl pro prodej jednoho Bitcoinu. Od 31. prosince 2019 do 3. února 2020 neprovedl žádný prodej. Zisk z prodaného Bitcoinu je tedy vypočten jako rozdíl pořizovací ceny jednoho Bitcoinu z 31. prosince 2019 a prodejní ceny Bitcoinu z 3. února 2020.

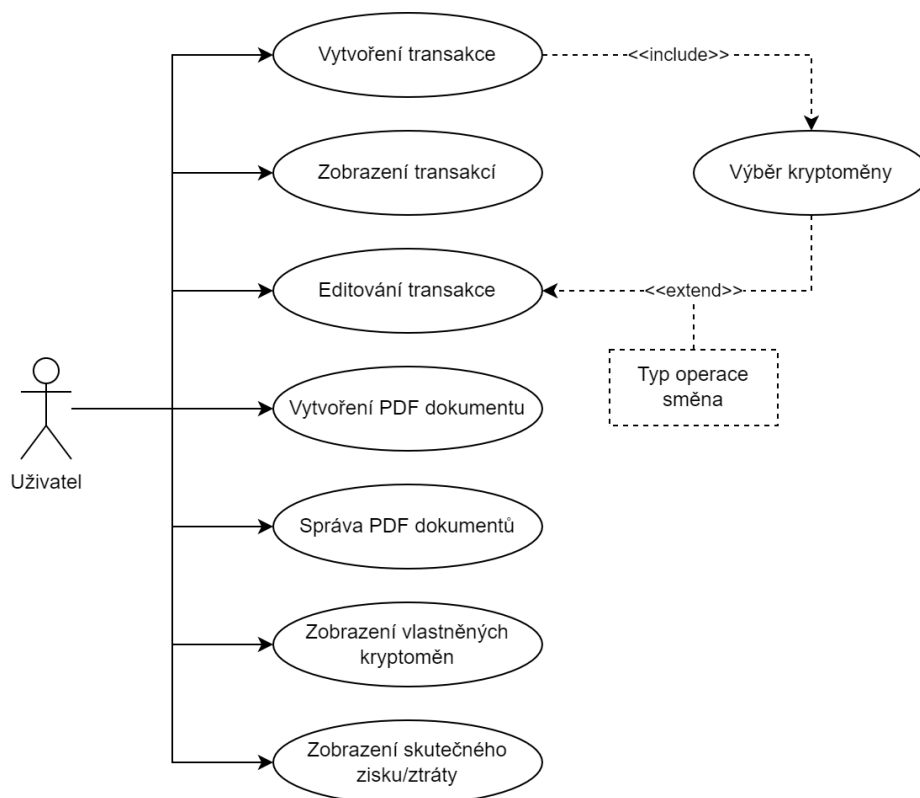
Z předchozího příkladu tedy vyplývá, že při generování výsledku za určité daňové období může aplikace počítat i s nákupy provedenými před daným obdobím. Jinak řečeno se vždy začíná tam, kde se minulý rok skončilo. Nákupy se tedy přesouvají. Tato skutečnost samozřejmě platí pouze u nákupu, prodej je uvažován pouze v rozmezí daného období. Z tohoto důvodu je tedy důležité správné uspořádání podle již zmíněné FIFO fronty.

Skutečný zisk nebo ztráta je tedy rozdíl všech nákupů (i těch, které mohly proběhnout před daným obdobím) odpovídajícím všem prodejem, včetně směny, za dané daňové období. Není-li nákup využit celý, je zapotřebí počítat s cenou využitého množství, nikoliv s celkovou cenou.

Cena nákupu je brána včetně poplatku. Cena prodeje je brána bez poplatku. Ceny v cizích měnách jsou přepočteny podle jednotného kurzu GFŘ za dané období. Není-li v aplikaci tento kurz za určité období dostupný, je využit nejbližší kurz z předchozích let a uživatel je o této skutečnosti informován.

4.2 Případy užití

Po získání informací potřebných pro správu kryptoměn bylo zapotřebí zjistit případy užití. Minimální potřebné operace jsou vytvoření transakce, zobrazení transakcí, editace transakce, generování PDF dokumentu a správa PDF dokumentů. Dalšími užitečnými funkcemi jsou zobrazení celkového zisku či ztráty a zobrazení vlastních kryptoměn. Grafické znázornění diagramu případu užití je na obrázku 4.1.



Obrázek 4.1: Diagram případu užití.

Vytvoření transakce

Vytvoření transakce se skládá ze tří typů operace, jimiž jsou nákup, prodej a směna. Jednotlivé typy vyžadují od uživatele stejné či rozdílné informace. Mezi stejné informace patří:

- poplatek,
- měna ve které je vedena cena a poplatek,
- datum a čas provedení transakce.

Poplatek není povinné pole, v případě jeho neuvedení je výchozí hodnotou nula.

Dále jsou zde informace pro jednotlivé typy. Pro nákup jsou vyžadovány:

- název nakoupené kryptoměny,
- nakoupené množství,
- celková cena včetně poplatku.

Prodej vyžaduje:

- název prodané kryptoměny,
- prodané množství,
- celková cena včetně poplatku.

A nakonec směna:

- název nakoupené kryptoměny,
- nakoupené množství,
- název prodané kryptoměny,
- prodané množství,
- cena směny včetně poplatku.

Zobrazení a editace transakce

Vytvořené transakce je potřeba zobrazit. Pro tento účel jsou navrženy dva typy zobrazení. První typ počítá s přehledem všech transakcí a možností filtrování. Dále je rozšířen o stručný přehled základních informací. Druhý případ užití je zaměřen na kompletní přehled informací o dané transakci. Rovněž je zapotřebí počítat s možností vzniku chyby. Je tedy potřeba implementovat možnost editace transakce. V případě editace lze upravit všechny údaje výjimkou typu spravované kryptoměny a operace. Dále je přidána možnost smazání transakce.

Vytvoření a správa PDF dokumentu

Dalším případem užití je správa PDF dokumentů jednotlivých výpisů. Je zapotřebí uživateli poskytnout možnost vytvoření výpisu. Uživatel by měl mít možnost volby, pro jaké daňové období chce výpis generovat. Dále by měl mít přehled o všech vytvořených dokumentech. Uživatel může vytvořit více dokumentů pro jeden rok, mělo by tedy dojít k určité indikaci nebo uspořádání daných dokumentů. Rovněž je zapotřebí implementovat možnost smazání dokumentu. A samozřejmě implementace otevíření.

Hlavní přehled

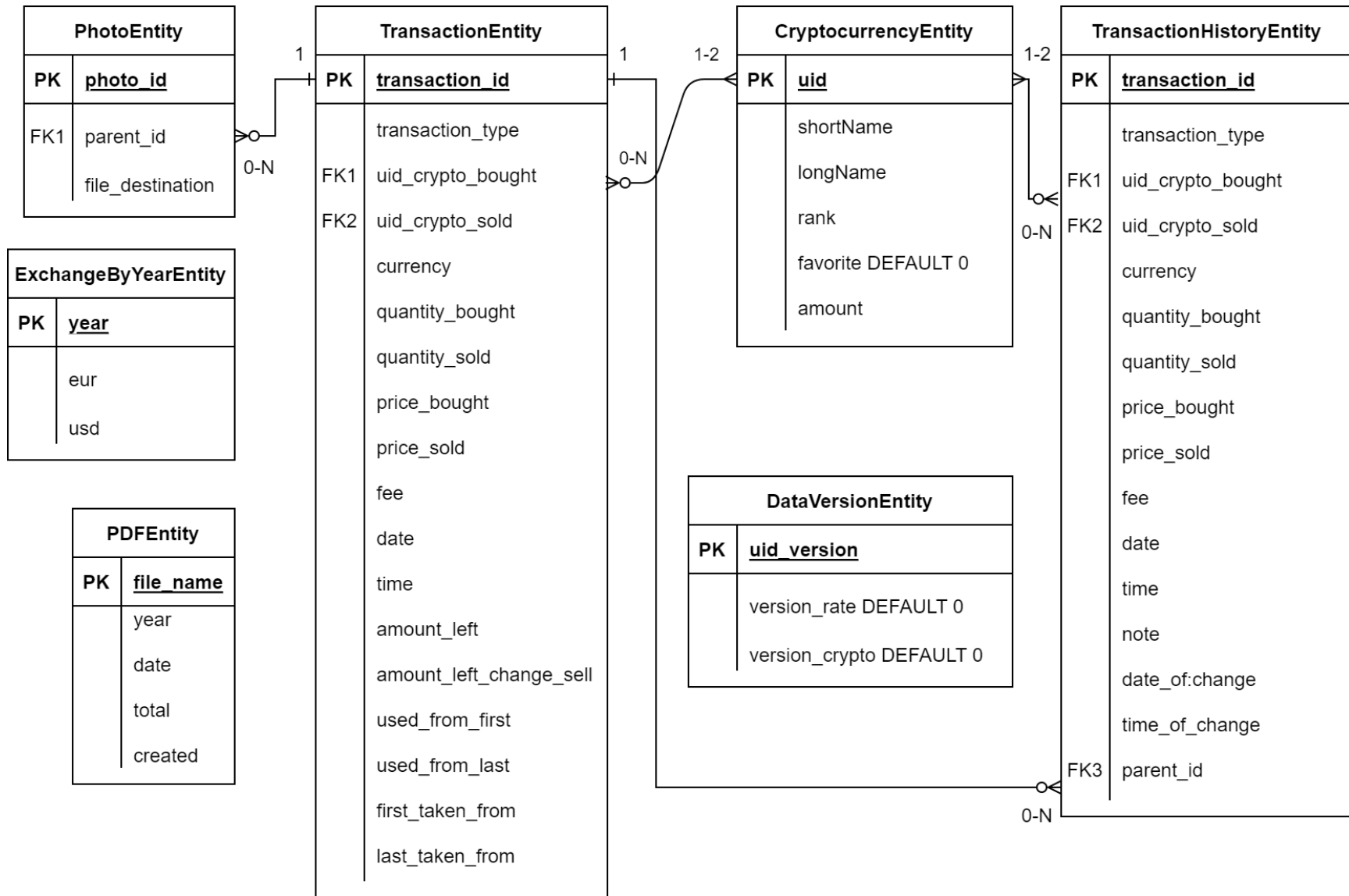
Cílem hlavního přehledu je nabídnout uživateli hodnoty skutečného zisku či ztráty za jednotlivá vygenerovaná období. V případě více výpisů pro dané daňové období je brán nejnovější záznam. Uživatel pro zobrazení výsledků svého obchodování tedy nemusí otvírat jednotlivé PDF dokumenty. Později došlo k rozšíření návrhu o výpis vlastněných, respektive evidovaných, kryptoměn utvářející přehled o portfoliu uživatele.

4.3 Návrh databáze

Jelikož aplikace pracuje s daty, které je třeba uchovat, bylo zapotřebí navrhnout způsob jejich uložení. A nejedná se pouze o data uživatele. Aplikace ke svému fungování potřebuje data získaná z internetu. Jedná se o seznam kryptoměn a hodnoty kurzů pro jednotlivá daňová období. V následující podkapitole si popíšeme ER diagram návrhu databáze na obrázku 4.2.

- **TransactionEntity** – Entita ukládá informace o jednotlivých transakcích včetně informací FIFO fronty.
- **TransactionHistoryEntity** – Entita slouží k uložení historie změn transakce. Jedna entita TransactionEntity může odkazovat na více entit TransactionHistoryEntity, podle počtu provedených změn.
- **PhotoEntity** – Tato entita slouží k uložení cest ke snímkům transakce. Entita TransactionEntity opět může obsahovat více těchto entit na základě počtu uložených snímků.
- **CryptocurrencyEntity** – Entita slouží k uložení stažených dat z API CoinGecko¹. Obsahuje seznam kryptoměn. Dále obsahuje vlastněné množství jednotlivých kryptoměn, hodnotu market ranku a údaj, zdali má uživatel danou kryptoměnu uloženou jako oblíbenou.
- **ExchangeByYearEntity** – Entita ukládá jednotné kurzy za jednotlivé roky stažené z realtime databáze Firebase.
- **PDFEntity** – Tato entita slouží k uložení cesty k PDF souboru. Dále obsahuje informace o datu vytvoření, roku daňového období a celkového zisku/ztráty.
- **DataVersionEntity** – Jedná se o entitu uchovávající pomocné informace potřebné ke kontrole aktuálnosti stažených dat.

¹<https://www.coingecko.com/en/api>



Obrázek 4.2: ER digram interní databáze.

4.4 Uživatelské rozhraní

Uživatelské rozhraní je důležitou částí aplikace. Jedná se o první věc, kterou uživatel spatří. V případě špatně navrženého rozhraní se může zároveň jednat i o poslední. Aplikace může splňovat všechno, co od ní uživatel očekává, nicméně nepřehledné uživatelské rozhraní může uživatele odradit od jejího využívání.

Existuje spousta knih a doporučení, jak vytvořit přívětivé uživatelské rozhraní. Příkladem může být již zmíněná služba Material Design poskytovaná společností Google. Dále kniha *Don't Make Me Think*², jež se těší velké oblibě a v níž jsem hledal ponaučení i rady, jak navrhnout přívětivé uživatelské prostředí.

Základní rozložení aplikace

Na počátku návrhu uživatelského rozhraní bylo rozhodnuto, jak bude aplikace strukturována. Aplikace je rozdělena mezi tři fragmenty (obr. 4.3.a). Každý fragment reprezentuje jednu ze základních funkcí. Jedná se o fragmenty: úvodní obrazovka, přehled transakcí a přehled PDF dokumentů. Tento typ rozdělení je běžně využíván mobilními aplikacemi.

Přehled transakcí se sestává ze seznamu všech transakcí. Seznam lze filtrovat podle určitého časového intervalu. K přidání nové transakce slouží plovoucí tlačítko (anglicky *floating button*). Rozložení okna pro přehled transakcí je vyobrazeno na obrázku 4.3.b. Od úvodního okna k vytvoření nové transakce uživateli stačí pouhé dvě kliknutí. Jedná se o jedno z doporučení převzatých z knihy *Don't Make Me Think*, která uvádí, že by nic důležitého nemělo být dále než dvě kliknutí [8]. Na podobném principu zobrazení funguje i seznam PDF dokumentů, včetně plovoucího tlačítka pro generování výpisu.

Poslední dva návrhy slouží pro vytvoření transakce a její podrobný výpis. Oba návrhy můžeme vidět na obrázku 4.4. Vytvoření transakce (obr. 4.4.a) se skládá z řady vstupních polí, do kterých uživatel zadá údaje o provedené transakci. Mezi jednotlivými operacemi se lze přepínat pomocí menu v horní části obrazovky. Podrobný výpis transakce má obdobné uspořádání jako vytvoření transakce. Na rozdíl od menu pro výběr operace obsahuje nadpis s typem operace a tlačítkem pro otevření okna k editování. Vstupní pole nahradila textová pole. Popisovaný návrh si lze prohlédnout na obrázku 4.4.b. Okno editování se zakládá na oknu vytvoření transakce a není tedy vytvořen jeho návrh.

²<https://sensible.com/dont-make-me-think/>



(a) Návrh rozdělení aplikace do fragmentů.

(b) Návrh výpisu transakcí do seznamu.

Obrázek 4.3: První obrázek zobrazuje návrh rozdělení aplikace do tří fragmentů. Každý fragment zastupuje jednu z hlavních funkcí. První (levý) fragment slouží pro přehled transakcí. Druhý fragment zastupuje úvodní obrazovku a je otevřen při zapnutí aplikace. A poslední fragment slouží pro seznam PDF výpisů. Druhý obrázek znázorňuje okno fragmentu pro přehled transakcí. Fragment obsahuje seznam transakcí, filtr podle časového intervalu a plovoucí tlačítko pro přidání transakce.



(a) Návrh okna pro přidání transakce.

(b) Návrh okna podrobného přehledu transakce.

Obrázek 4.4: Na prvním obrázku je zobrazeno okno pro vytvoření transakce. Obsahuje vstupní pole pro vložení údajů a menu v horní části obrazovky pro výběr typu operace. Dle vybraného typu operace budou uzpůsobeny vstupní pole. Na druhém obrázku je okno s přehledem transakce. Údaje jsou vypsány v textovém poli. V horní části obrazovky je nadpis s typem operace a tlačítko k přechodu na obrazovku pro editování transakce.

Práce s uživatelem

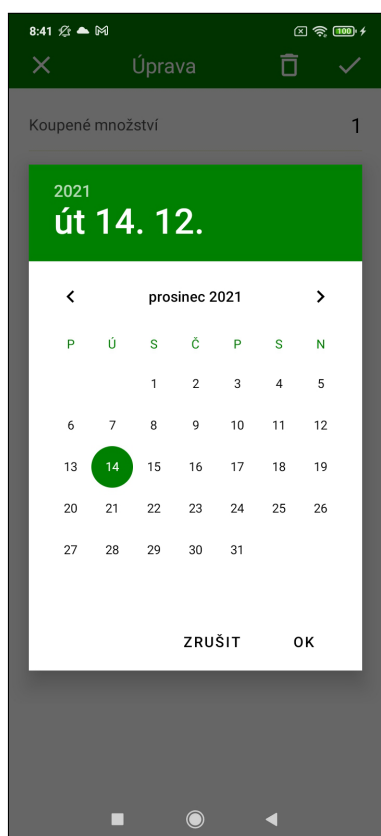
Při následovném vytváření uživatelského rozhraní byla snaha o co nejužší spolupráci s uživateli. Tímto postupem byl do práce vnesen nový úhel pohledu, který se tak nezastavil pouze na mé představě o tom, co by mohlo, ale rovněž nemuselo, být dobré. Po načrtnutí návrhu rozložení byly postupně pro různé sekce vypracovány dva až tři koncepty, jež vycházely z návrhu, ale lišily se po grafické stránce. Na základě těchto konceptů probíhal výběr a ladění s uživateli. V této části zkoumání byla hlavním cílem přehlednost a uspořádání prvků. Druhotným cílem byla estetika, nicméně se jednalo stále o návrh.

V případě vytvoření transakce se jednalo o uspořádání vstupních polí a jejich popisku. Jednotlivé reakce se lišily s tím, jak se lišil názor uživatelů. Volba tedy nebyla jednoznačná. Došlo k vyhodnocení slabých a silných stránek jednotlivých konceptů a vypracování nového, jenž přebíral jejich nejsilnější body. Nově vypracovaný koncept se setkal již s lepším ohlasem a byl tedy využit jako stavební kámen aktivity. V následujícím vývoji byla prováděna vylepšení zaměřená převážně na drobnější detaily, jako například nepatrné vizuální oddělení jednotlivých vstupních polí nebo varování na povinná pole.

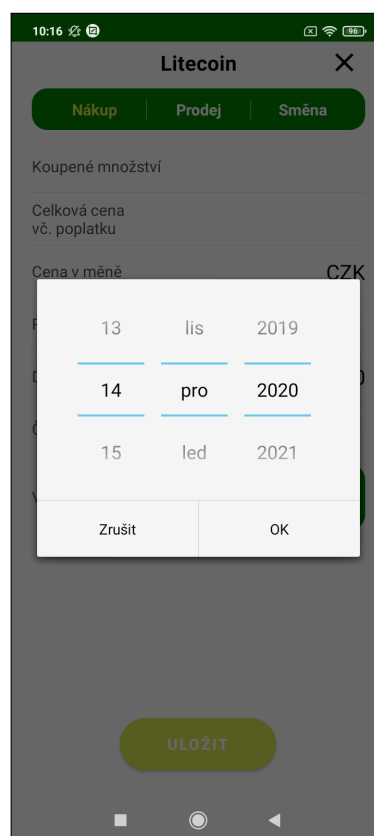
Dalším důkladněji zkoumaným prvkem byl výběr data a času. Došlo k porovnání způsobu použití a uživatelského prožitku dvou typů výběru. V případě data se jednalo a okna pro výběr ve stylu kalendáře a posuvného výběru, obě možnosti jsou znázorněny na obrázku 4.5. Výsledkem zkoumání byly dvě cesty, kterými se lze vydat. V případě výběru pomocí kalendáře byla silná stránka na straně estetického dojmu a vyšší přehlednosti. Výběr data působil přirozenějším dojmem, kdy uživatel viděl jednotlivé dny v měsíci. Nicméně jeho slabá stránka byla nakonec důvodem, proč nebyl zmíněný výběr použit. I přesto, že mohl uživatelský prožitek působit přívětivěji, nejednalo se o hlavní kritérium výběru. Vezmeme-li v potaz případ užití, kdy uživatel své transakce uvádí do vzdálenější minulosti, tak vychází výběr pomocí posunu jako rychlejší volba zpracování. Rozdíl času potřebného pro volbu data roste se vzdáleností do minulosti. Na základě tohoto zjištění bylo vybráno okno pro výběr data pomocí posunu.

Pro další úpravu na základě zpětné vazby od uživatele můžeme zůstat u výběru data. V případě nezvoleného data či času je automaticky nastaven aktuální čas. V opačném případě je nastaven již zvolený čas, jelikož chyba provedená uživatelem při výběru vznikala převážně zbrklostí a uživatel upravoval zvolenou hodnotu v rozsahu jednoho posunu.

Na podobném principu probíhal i zbytek tvorby. Některé části aplikace byly navrženy podle již zaběhnutých konceptů a nemusely být tedy důkladněji zkoumány. Jiné se mohly od původních konceptů naprosto lišit. Jedná se například o úvodní obrazovku, které si z prvotních konceptů ponechala pouze hlavní cíl. Na základě testování došlo k celkovému přeuspořádání včetně nového grafického návrhu a přidání přehledu vlastněných kryptoměn. Rovněž byl na základě zkoumání vytvoř i světlý mód aplikace, jelikož by jej někteří uživatelé uvítali.



(a) Výběr data pomocí kalendáře.

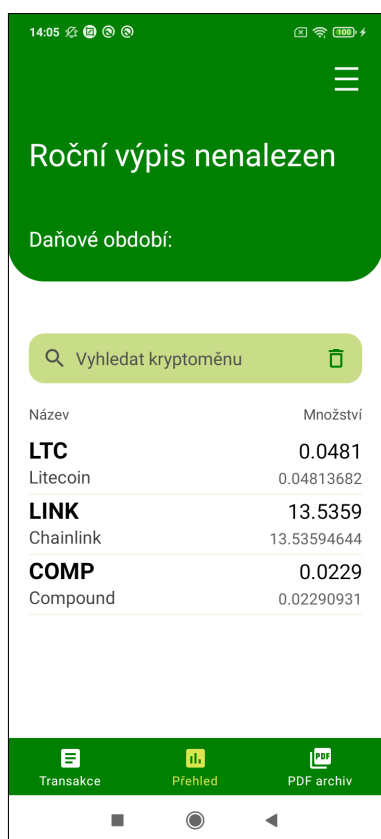


(b) Výběr data pomocí posunu.

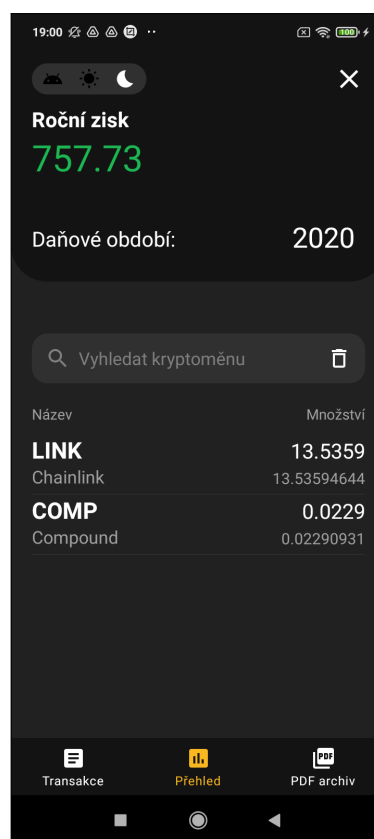
Obrázek 4.5: Na obrázcích jsou znázorněny dva způsoby výběru data, z kterých bylo na základě zkoumání zvoleno okno výběru. Na prvním obrázku je zobrazeno okno pro výběr data pomocí kalendáře. I přesto, že působí přívětivěji, nebylo nakonec vybráno pro svou časovou náročnost při volbě data ve vzdálenější minulosti. Druhý obrázek znázorňuje okno pro výběr data pomocí posunu, které bylo nakonec použito.

Úvodní obrazovka

Úvodní obrazovka (obr. 4.6) je tvořena jedním ze tří základních fragmentů hlavní části aplikace. Její uspořádání je rozděleno na dvě části poskytující odlišné informace. Hlavní část umístěnou v horní polovině obrazovky tvoří přehled ročního zisku/ztráty za daňové období, je-li vygenerován výpis. V opačném případě je uživatel informován o nenalezení ročního výpisu. V případě, že existuje více výpisů pro jeden rok, je uvažován ten, jenž byl vygenerován jako poslední. Uživatel tedy vidí své výsledky za jednotlivá daňová období, aniž by musel otvírat PDF výpisy. Dále je zde nastavení barevného módu aplikace. Možnosti výběru jsou: systémové nastavení, světlý mód a tmavý mód. Ukázka tmavého módu včetně výběru je na obrázku 4.6.b. Druhou část obrazovky tvoří přehled vlastněných kryptoměn. Množství je uvedeno v zaokrouhlené podobě na čtyři desetinná místa včetně skutečné hodnoty, která je uvedena jako méně výrazná.



(a) Úvodní obrazovka.

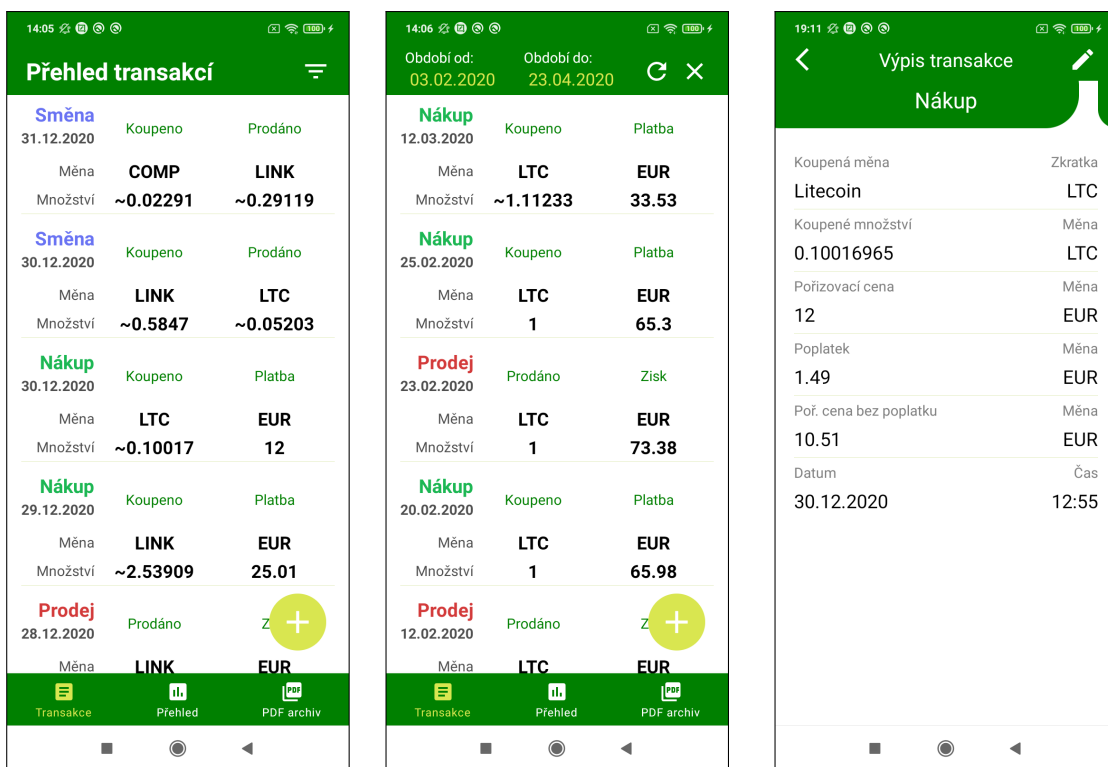


(b) Úvodní obrazovka – tmavý mód.

Obrázek 4.6: Na obrázcích je znázorněno úvodní okno aplikace ve světlém a tmavém módu. Horní část okna tvoří výpis ročního zisku za jednotlivá daňová období, případně oznámení, že výpis nebyl nalezen. Dále se zde vyskytuje nastavení barevného módu ve třech volbách (systém, světlý a tmavý). Spodní část obrazovky slouží k výpisu vlastněných kryptoměn.

Přehled transakcí

Transakce lze zobrazit ve dvou formátech. Prvním formátem, který rovněž slouží pro přístup k druhému, je výpis seznamu transakcí se základními informacemi, znázorněný na obrázku 4.7.a. Výpis obsahuje informace o typu transakce, datu provedení a měnách včetně cen. Seznam transakcí lze filtrovat pomocí třech nastavení data. Lze nastavit období od data A do současnosti, od data A do data B (obr. 4.7.b) a od první transakce do data B. Při rozkliknutí transakce ze seznamu dojde k otevření podrobného výpisu, jenž obsahuje kompletní informace (obr. 4.7.c). Rovněž poskytuje možnost úpravy. Pro zlepšení uživatelského prožitku byla využita funkce, s kterou se můžeme setkat například v galerii snímků. Tedy pohyb mezi jednotlivými transakcemi posunem vlevo a vpravo. Uživatel může navazující transakce procházet rychleji, jelikož se nemusí vracet na seznam transakcí. Ze seznamu transakcí se dále dostaneme k vytvoření nové transakce pomocí plovoucího tlačítka.



(a) Seznam všech transakcí. (b) Seznam transakcí OD – DO. (c) Podrobný výpis transakce.

Obrázek 4.7: První dva obrázky znázorňují přehled všech transakcí (případně v rozmezí zvoleného filtru) se stručnými informacemi jako jsou typ operace, typ měn a ceny. Dále plovoucí tlačítko pro přidání transakce. Na posledním obrázku je vyobrazeno okno s kompletním výpisem transakce.

Vytvoření a editace transakce

Vytvoření transakce (obr. 4.8.a) je rozděleno do tří fragmentů. Toto rozdělení se vhodně nabízelo, jelikož má uživatel na výběr tři typy transakcí, které se od sebe více či méně liší. Rozdělení do fragmentu nám poskytuje možnost měnit pouze grafický obsah aktivity (neboli zobrazovaného okna), nikoliv celou aktivitu.

Grafické zpracování editace transakce (obr. 4.8.b) vychází z grafického rozhraní vytvoření transakce. Uživatel má možnost změnit všechny údaje výjimkou typu evidované kryptoměny a typu transakce. Tyto změny jsou nadále uloženy do historie změn, která je vyobrazena společně s celkovým přehledem transakce. Rovněž je zde možnost transakci smazat. V obou případech uživatel musí akci odsouhlasit v dialogovém okně.

17:25 100% 100% 100% 100%

Litecoin X

Nákup | Prodej | Směna

Koupené množství

Celková cena vč. poplatku

Cena v měně CZK

Poplatek

Datum provedení

Čas provedení

Vložit snímek

ULOŽIT

(a) Vytvoření transakce.

17:25 100% 100% 100% 100%

X Úprava X ✓

Koupené množství 0.10016965

Pořizovací cena 12

Cena v měně EUR

Poplatek 1.49

Datum provedení 30.12.2020

Čas provedení 12:55

Vložit snímek

Poznámka k úpravě

(b) Editování transakce.

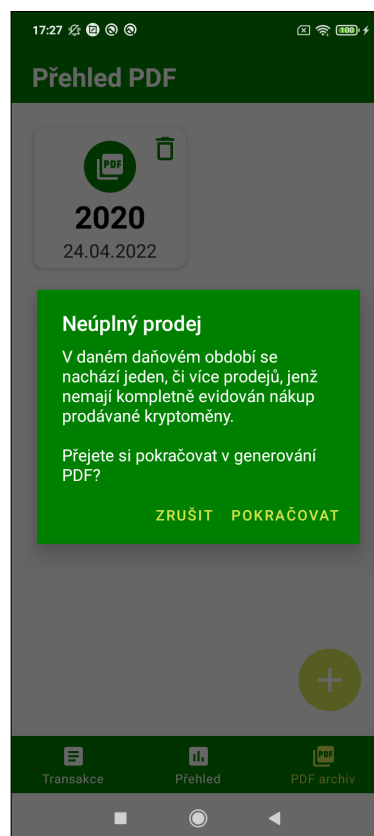
Obrázek 4.8: Na prvním obrázku je vyobrazeno okno vytvoření transakce. Skládá se z aktivity, která je rozdělena do tří fragmentů pro různé typy transakcí. Na pravém obrázku je znázorněno okno pro editaci transakce. Lze si povšimnout, že základní rozložení jsou stejná, aby nedocházelo ke zmatení uživatele. U editace lze uvést poznámku k úpravě či smazat transakci.

PDF archiv

PDF archiv je poslední ze tří hlavních částí aplikace. Obsahuje seznam vytvořených PDF výpisů (obr. 4.9.a) včetně možnosti vytvoření nového výpisu pro zvolený rok. Pokud se v generovaném roce nachází prodej, jenž nemá správně evidováno nabytí prodávané kryptoměny, je uživatel o této skutečnosti informován dialogovým oknem znázorněným na obrázku 4.9.b.



(a) Přehled PDF výpisů.



(b) Upozornění na nekompletní nákup.

Obrázek 4.9: První obrázek znázorňuje seznam vygenerovaných výpisů v podobě PDF dokumentů a plovoucí tlačítko pro vygenerování výpisu. Na druhém obrázku je vyobrazeno dialogové okno upozorňující uživatele, že pro vybrané daňové období není evidováno nabytí všech prodávaných kryptoměn.

Kapitola 5

Implementace

V následující kapitole budou popsány způsoby implementace jednotlivých částí aplikace. Bude popsán způsob správy transakce, tedy vytvoření, zobrazení a editování transakce. Dále bude vysvětlen způsob evidování transakcí podle FIFO fronty a způsob získání a uložení dat. Na závěr kapitoly budou popsány způsoby generování PDF souboru a zpracování informací pro přehledný výpis.

5.1 Správa transakcí

Správu transakcí lze rozdělit na tři typy, kterými jsou vytvoření, zobrazení a editování transakce. Následující podkapitola popíše implementaci jednotlivých typů a případné problémy, jež bylo třeba řešit.

Vytvoření transakce

Pro implementaci vytvoření transakce bylo zapotřebí postupně splnit způsob výběru typu operace, získání dat od uživatele a jejich uložení, které je popsáno v sekci 5.3. Prvním krokem bylo vytvoření volby operace. Tedy nákup, prodej nebo směna. Tento výběr byl zpracován prostřednictvím fragmentu, jež umožnil měnit obsah okna v jedné aktivitě. S využitím této skutečnosti nebylo zapotřebí předávat potřebné údaje opakovaně při změně typu operace.

Druhým krokem bylo vytvoření způsobu získání a zpracování dat od uživatele. Způsoby získání se liší podle formátu dat. Na počátku bylo zapotřebí vypracovat výběr kryptoměny. Na rozdíl od informací jako jsou cena nebo množství, nebylo možné kryptoměny řešit pomocí vstupního pole (v Android Studiu označovaném pojmem EditText). Kryptoměna nepracuje pouze s názvem. Pro správnou funkci aplikace je zapotřebí znát i dodatečné informace jako jsou symbol, vlastněné množství a samozřejmě identifikátor. Výběr byl tedy zpracován ve vlastní aktivitě, kde je uživateli zobrazen seznam kryptoměn získaných z internetu. Způsob získání bude popsán v navazující podkapitole 5.3.

Uvedené řešení přineslo další problém, jež bylo zapotřebí vyřešit. Seznam kryptoměn obsahuje více než třináct tisíc položek, výběr je tedy značně zkomplikován. Bylo zapotřebí vytvořit způsob, který uživateli výběr usnadní. Jelikož se jedná o často opakující se úkon, došlo k implementaci třech řešení, napomáhající rychlejší volbě kryptoměn. Seznam je seřazen podle dvou kritérií, kterými jsou hodnota market ranku prvních pěti set kryptoměn a uživatelem zvolené kryptoměny jako oblíbené. Hodnota market ranku zajistí, že na začátku seznamu budou nejvíce obchodované kryptoměny, u nichž je vyšší pravděpodobnost,

že právě tyto kryptoměny uživatel plánuje spravovat. Před těmito kryptoměnami budou nadále uživatelem zvolené kryptoměny jako oblíbené. Samozřejmě nechybí možnost vyhledávání podle názvu nebo symbolu.

Zpracování množství, ceny a poplatku bylo provedeno pomocí vstupního pole omezeného na kladná desetinná čísla. Pro výběr data a času bylo využito okna výběru za pomoci posunu. V případě výběru první hodnoty odpovídá výchozí datum a čas, zobrazený ve výběru, aktuálnímu dni. V opačné případě je nastavena již vybraná hodnota, jelikož při změně docházelo převážně k posunu o jednu hodnotu. Pro přiřazení snímků k transakci bylo využito funkce zaznamenání zpětného volání výsledku `registerForActivityResult()` poskytované Activity Result API.

Pro případ, že nebyly uvedeny všechny povinné údaje, nebo byl zadán čas, jenž je vůči aktuálnímu času v budoucnu, bylo implementováno vizuální upozornění na chybné vyplnění. Transakci nelze uložit, pokud nejsou všechny údaje správně vyplněny.

Zobrazení transakcí

Zobrazení transakce bylo rozděleno na dvě části. První část byla vytvořena jako seznam transakcí vyobrazený pomocí komponenty `RecyclerView`. Jednotlivé transakce jsou utvářeny položkami, které jsou vykresleny v uvedené komponentě. Zmíněné zobrazení bylo vypracováno jako rychlý přehled všech transakcí. Uživatel může rovněž požadovat zobrazení transakcí za určité období, a proto byl implementován filtr pro nastavení data od – do zvoleného období. Bylo zapotřebí vzít v potaz tři možné způsoby použití, jimiž jsou zobrazení od data A do současnosti, od data A do data B a od počátku do data B. Rovněž bylo potřeba zamezit nastavení data B před datum A. Informace o množství a ceně byly s ohledem na omezený prostor upraveny do zaokrouhlených hodnot. Z důvodu zachování nejvyšší možné přesnosti je zaokrouhlení implementováno na počet míst. Míra zaokrouhlení desetinného čísla, případně zdali bude zobrazeno, vychází z velikosti celého čísla. Menší čísla jsou tedy přesnější. Z popisové aktivity může uživatel přejít na vytvoření transakce nebo na podrobný výpis zvolené transakce.

Druhou část zobrazení transakce představuje podrobný výpis. Aktivita zobrazuje všechny vyplněné údaje, včetně poplatku, jenž není povinný. V případě nevyplnění nabyde hodnoty nula. Z případu užití lze očekávat, že uživatel může při vytváření transakce provést chybu. Bylo tedy přidáno tlačítko na přechod k editaci transakce, jež bude popsána v navazující části. S editací transakce byla implementována i historie změn. Původní hodnoty změněných částí jsou zobrazeny v přehledu transakce, včetně času provedení úpravy.

Editace transakce

Poslední částí správy transakce je možnost úpravy a mazání. V případě mazání bylo zapotřebí zpracovat smazání a úpravu všech dat spjatých s transakcí. Při mazání je zapotřebí nejdříve odstranit historii změn, přiřazené snímky a cesty k nim vedoucí, upravit množství vlastněných kryptoměn a přepočítat FIFO frontu. Až poté je možné transakci odstranit.

Implementace úpravy transakce byla do značné míry podobná vytvoření transakce. Získání dat od uživatele, včetně jejich uložení, bylo převzato z vytvoření transakce. Jako počáteční údaje byly nastaveny již evidované údaje. Dále bylo zapotřebí vytvořit dialogové okno pro potvrzení úpravy. Pro uložení změn byla implementována funkce porovnávající ukládaná data s původními. Funkce zajistí, že jsou data v databázi upravena pouze v případě, že se alespoň jeden údaj neshoduje. V takovém případě jsou původní hodnoty uloženy do nové tabulky historie transakce.

5.2 Operace FIFO nad transakcemi

Správné uspořádání transakcí podle FIFO fronty lze považovat za nejdůležitější a nejkritičtější část aplikace. Rovněž se jedná o hlavní funkci aplikace, jelikož bez správného uspořádání nelze generovat výpis transakcí za určité daňové období. V následující podkapitole budou popsány způsoby implementace FIFO fronty. Rovněž budou uvedeny případy, kdy je potřeba aktualizovat danou frontu. A samozřejmě bude popsán způsob uložení dat potřebných pro uspořádání transakcí.

Uložení dat FIFO fronty

K uložení dat FIFO fronty byla využita existující tabulka databáze pro uchování dat transakce. Data byla takto implementována, jelikož se jedná o informaci uchovávanou každou transakcí. V případě nákupu bylo zapotřebí uchovat informace o zatím neprodaném množství kryptoměny (`amountLeft`) z daného nákupu. Pro prodej bylo zapotřebí uchovat informace o zbývajícím množství (`amountLeft`), které nemá dostupný záznam o nákupu. Dále transakci prvního (`firstTakenFrom`) a posledního (`lastTakenFrom`) využitého nákupu včetně množství (`usedFromFirst` a `usedFromLast`), které bylo z odpovídajících transakcí odebráno. Pro směnu bylo zapotřebí evidovat stejné informace jako u nákupu i prodeje, jelikož směnu tvoří nákup jedné kryptoměny a prodej druhé. U směny bylo tedy zapotřebí uchovat záznam o zbývajícím množství dvakrát, a proto byl přidán sloupec `amountLeftChangeSell`.

Implementace fronty

Implementování přepočtu fronty bylo zapotřebí u většiny operací prováděných s transakcemi. Jedná se o vytvoření, smazání a úpravu transakce. V případě úpravy transakce je aktualizace fronty provedena pouze u změny množství nebo času provedení transakce. U směny bylo potřeba přidat aktualizaci i při změně prodané kryptoměny.

- **Vytvoření nákupu** – Metoda implementovaná pro vytvoření nákupu musí provést obnovení všech nákupů pro danou kryptoměnu, jenž následují po novém nákupu. Dále musí obnovit všechny prodeje, které jsou spjaté s těmito nákupy. Poté je do fronty přidán nový nákup. Po přidání jsou nalezeny prodeje, kterým můžou být přiřazeny dostupné nákupy a nově vypočtená FIFO fronta je aktualizována v databázi.
- **Vytvoření prodeje** – Prací metody je obnovení všech prodejů následovaných po novém prodeji a nákupů s nimi spjatými. Je-li nákup využit i neobnovenými prodeji, je z něj obnoveno pouze množství odpovídající obnovené hodnotě prodeje. V opačném případě je využitý nákup zcela obnoven. Následně je nový prodej přidán a je vypočteno nové uspořádání fronty.
- **Vytvoření směny** – Směna se skládá z nákupu jedné kryptoměny za druhou. K její implementaci jsou tedy využity metody pro vytvoření nákupu i prodeje.
- **Smazání nákupu** – Metoda funguje na podobném principu jako vytvoření nákupu. Je zapotřebí obnovit všechny prodeje od prvního nákupu po mazané transakci. Prodeje mazané transakce je potřeba zpracovat zvlášť, jelikož daná transakce může zpracovat pouze část z prvního prodeje. V takovém případě nemůže dojít k obnovení celého prodeje, ale pouze hodnoty zpracované danou transakcí. Následně dojde k obnovení navazujících nákupů, odstranění transakce a přepočítání fronty.

- **Smazání prodeje** – Implementace smazání prodeje musí zajistit, že z prvního nákupu dojde k obnovení pouze množství, které bylo využito prodeji následujícími od mazaného prodeje včetně. Následující nákupy jsou kompletně obnoveny. Dále jsou obnoveny všechny prodeje navazující na mazanou transakci a transakce je smazána. Nakonec dochází k přepočtu fronty.
- **Smazání směny** – Implementace využívá metod pro smazání nákupu a prodeje.
- **Úprava koupeného množství** – Opět je zapotřebí zpracovat první prodej zvlášť. První prodej si musí zanechat využitou hodnotu z nákupů, jenž se nachází před upravenou transakcí. Následující prodeje jsou obnoveny. Rovněž jsou obnoveny všechny nákupy po editované transakci. Transakci je nastavena nová hodnota nakoupeného množství. Poté je přepočtena fronta.
- **Úprava prodaného množství** – Stejně, jako u smazání prodeje, i zde je zapotřebí zpracovat první nákup odděleně. Zbylé nákupy a všechny prodeje od měněné transakce jsou následně obnoveny. Poté je nastavena nová hodnota prodeji a je přepočtena fronta.
- **Úprava množství při směně** – Pro nákup je opět využita metoda úpravy koupeného množství. V případě, že prodaná kryptoměna nebyla změněna, je využito metody pro úpravu prodaného množství. V opačném případě dojde k smazání původního prodeje za využití metody pro smazání prodeje a následnému vytvoření prodeje nového pro nově zvolenou kryptoměnu metodou vytvoření prodeje.
- **Úprava času nákupu** – V případě úpravy času bylo zapotřebí implementovat dvě potenciální události. Posun transakce v čase zpět, a naopak blíže k přítomnosti. Při posunu před původní hodnotu jsou provedeny stejné úkony, které vykonává metoda pro úpravu nakoupeného množství. Nicméně je zapotřebí dané úkony provést od nového data, jelikož dojde k ovlivnění navazujících transakcí. Tyto úkony jsou provedeny i v případě posunu transakce blíže k současnosti. V tomto případě jsou vykonány od původního data nákupu, jelikož nedochází k ovlivnění předchozích transakcí.
- **Úprava času prodeje** – Implementace se svým principem neliší od úpravy času nákupu. Rovněž zde musí dojít ke kontrole typu posunu a na základě daného typu jsou data upravována od nové nebo původní pozice.
- **Úprava času směny** – Metoda funguje na stejném principu jako úprava množství při směně, liší se pouze ve využití metod pro úpravu času, nikoliv množství.

5.3 Získání a uchování dat

Aplikace ke svému fungování potřebuje pracovat se získanými daty, které je třeba uchovávat. Data jsou získána dvěma způsoby. Prvním způsobem jsou data generovaná uživatelem. Tedy záznamy jednotlivých transakcí. Druhý způsob získávání dat je stažení potřebných dat z internetu. Tato data jsou stažena ve formátu JSON, který je nadále potřeba upravit.

Jak jsou data získána

Získání dat generovaných uživatelem bylo popsáno v sekci 5.1. Následující sekce bude popisovat získání dat stažených z internetu, jež jsou potřebná pro správnou funkci aplikace.

Aplikace pro vygenerování výpisu potřebuje jednotný kurz eura i amerického dolaru za jednotlivá daňová období. Ty se samozřejmě každým rokem mění a je tedy potřeba aktualizovat seznam po jejich vydání. Bohužel nebyla nalezena žádná API poskytující tyto informace. Došlo tedy k vytvoření databáze pomocí služby Firebase, do které jsou data zanesena po jejich zveřejnění. Databáze spravovaná službou Firebase je NoSQL, její formát tedy není tabulkový, nýbrž využívá formátu JSON. Obsahuje dva JSON objekty, které jsou znázorněny v ukázce 5.1.

```
{
  "exchange": {
    "version": 2,
    "year": {
      "2009": {
        "eur": "26.45",
        "usd": "19.06"
      },
      "2010": {
        "eur": "25.29",
        "usd": "19.11"
      }
    }
  }
}
```

Výpis 5.1: Ukázka struktury uchování dat pomocí JSON v databázi Firebase.

První slouží k uchování verze dat, druhý obsahuje JSON pole s jednotnými kurzy za dané roky. Při spuštění aplikace s dostupným internetovým připojením je stažena hodnota prvního objektu obsahujícího verzi. Získaná verze se porovná s verzí dat uložených v interní databázi aplikace. Jsou-li verze odlišné, dojde ke stažení aktuálních hodnot. Výhodou tohoto řešení je menší objem dat stažených při kontrole aktuálnosti dat.

Dalšími potřebnými daty získanými z internetu jsou údaje o kryptoměněch. Ty jsou zprostředkovány pomocí API poskytované společností CoinGecko. Prostřednictvím zmíněné API jsou získány jména a symboly více než třinácti tisíc kryptoměn. Cílem aplikace není uchování aktuálních informací o kryptoměněch. Potřebné informace jsou tedy uloženy do zařízení pouze jednou, jelikož se jedná o velké množství kryptoměn, a tedy i výkonnostně náročnou operaci. Pravděpodobnost, že by se uživatelem vyžadovaná kryptoměna nenacházela ve staženém seznamu je zásluhou množství stažených kryptoměn malá. I přesto může tato skutečnost nastat, a proto bylo navrženo řešení, které bude uvedeno jako možnost dalšího rozvoje aplikace. Dále je získána hodnota market ranku pro prvních pět set kryptoměn. Důvodem získání této hodnoty pouze pro uvedený počet je způsob, jakým jsou data získána. Využitá API neposkytuje zmíněný údaj společně s již získanými údaji. Pro získání je tedy zapotřebí vykonat další dotaz. Ten je nicméně omezen na maximální množství dvě stě padesáti kryptoměn. Získání market ranku pro všechny kryptoměny by tedy vyžadovalo

$$\frac{N}{250} + 0.5 \text{ kde } N \text{ je celkový počet stažených kryptoměn}$$

stažení, zaokrouhлено na celé číslo. S rostoucím počtem kryptoměn roste i časová náročnost získání daného údaje. Pět set kryptoměn tvoří dostatečné množství za předpokladu, že uživatel nebude procházet celý list, ale využije možnost vyhledávání.

Při prvním spuštění aplikace je vyžadováno připojení k internetu, aby mohlo dojít k stažení uvedených dat, jež jsou nezbytná pro fungování aplikace. Stahování probíhá v načítacím okně aplikace. V tomto okně aplikace setrvává do doby, než jsou data stažena. Při každém zapnutí aplikace přes načítací okno je provedena kontrola aktualizací v databázi Firebase, má-li zařízení přístup k internetu.

Jak jsou data uchována

Aplikace využívá k uchování dat interní databázi SQLite. Jsou zde uchovány údaje o jednotlivých transakcích, včetně potřebných údajů FIFO fronty. Dále je zde uloženo množství vlastněných kryptoměn či seznam oblíbených kryptoměn uživatele. Rovněž jsou zde uchovány i stažené informace, tedy seznam kryptoměn a jednotlivých kurzů. Databáze též obsahuje cesty k souborům generovaným aplikací. Schéma databáze je vyobrazeno v ER diagramu na obrázku 4.2.

Stažené informace jsou získány ve formátu JSON, je tedy zapotřebí jejich zpracování do tabulkového formátu relační databáze SQLite. Převod usnadňuje využití knihovny ROOM. Data stačí pouze extrahovat z formátu JSON a nadále s nimi pracovat za pomoci objektů `Entities` reprezentujících jednotlivé tabulky databáze.

V případě přiřazení snímku k transakci nebo vygenerování PDF souboru jsou tyto soubory uloženy ve složkách aplikace a cesty k nim jsou uloženy v databázi.

5.4 Generování PDF dokumentu

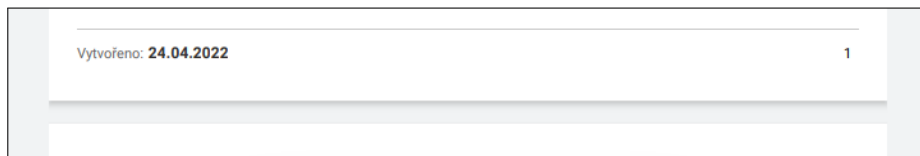
K implementaci generování PDF dokumentu byla využita knihovna PDFBox. Tato knihovna poskytuje řadu nástrojů pro tvorbu dokumentu. Pro vytvoření dokumentu bylo zapotřebí provést jeho inicializaci pomocí `PDDocument()`. Dále bylo potřeba inicializovat použité fonty. Knihovna poskytuje pár základních fontů, ty nicméně nepodporují českou znakovou sadu. Bylo tedy zapotřebí stáhnout a inicializovat vlastní fonty. V práci je využit font Roboto¹ pod licencí Apache 2.0. Následně byl vytvořen nový listu formátu A4 pomocí `PDPPage(PDRectangle.A4)`. Dále bylo potřeba inicializovat třídu `content stream` spravující daný list.

Pozicování textu na listu bylo prováděno pomocí souřadnic určujících horizontální a vertikální polohu. Došlo tedy k vytvoření pomocných proměnných uchovávající aktuální šířku a výšku pomyslného kurzoru. Pro vkládání textu bylo potřeba započít tuto událost pomocí třídy `content stream` voláním metody `beginText()`. Při ukončení práce s textem nebo vkládání netextového obsahu bylo zapotřebí ukončit danou operaci pomocí `endText()`. Text je možné vkládat dvěma způsoby. Prvním způsobem je vložení textu na nový řádek pomocí `newLine()`. Horizontální poloha bude odpovídat umístění kurzoru. Posun o vertikální polohu je nastaven pomocí `setLeading(hodnota_posunu)`. Druhou možností je `newLineAtOffset(tx, ty)`. Tato metoda poskytuje možnost posunu kurzoru o hodnoty `tx` a `ty`. Po vložení textu bylo zapotřebí aktualizovat hodnoty proměnných. Vytvoření nového listu bylo rozšířeno o přidání záhlaví (obr. 5.1). K ukončení práce s listem bylo přidáno vytvoření zápatí (obr. 5.2).

¹<https://fonts.google.com/specimen/Roboto>



Obrázek 5.1: Záhlaví PDF dokumentu.



Obrázek 5.2: Zápatí PDF dokumentu.

Uložení bylo implementováno do složky aplikace, přesněji ve složce `Documents`. Název byl nastaven jako rok vybraného daňového období a čas vytvoření. Uložení cesty k dokumentu bylo vyřešeno pomocí databáze.

5.5 Zpracování informací pro přehledný výpis

Po vytvoření výpisu bylo dalším cílem implementovat přehled o skutečném zisku/ztrátě a vlastněných kryptoměnách. Pro přehled o zisku/ztrátě za jednotlivá daňová období bylo zapotřebí vymyslet způsob, jak tuto hodnotu získat z vygenerovaného výpisu. Došlo k rozšíření tabulky `PDFEntity` o sloupec `total`, jenž obsahuje výslednou hodnotu. Dále bylo zapotřebí vytvořit možnost listování mezi výsledky za jednotlivá daňová období. Výsledky byly uloženy do pole a na základě jeho velikosti byla vytvořena tlačítka pro posun mezi jednotlivými roky uloženými v poli. Rovněž bylo zapotřebí eliminovat duplicitní výskyt daňových období, pokud bylo vygenerováno více výpisů pro jeden rok. V takovém případě je uvažován nejnovější výpis. Pro danou implementaci bylo zapotřebí opět rozšířit tabulku `PDFEntity` o sloupec `created` obsahující čas vytvoření výpisu.

Pro vytvoření seznamu vlastněných kryptoměn bylo využito uchovaného množství kryptoměn v tabulce `Cryptocurrency`. Tato hodnota je přepočítána každým přidáním a smazáním transakce. Rovněž i úpravou porízeného nebo prodaného množství.

Kapitola 6

Testování a vyhodnocení

Testování je důležitou částí jakéhokoliv vývoje. Správné testování vývojáři pomáhá odhalit chyby a slabiny jeho práce, na které je potřeba se zaměřit. A proto bylo při tvorbě mé bakalářské práce využíváno již od počátku tvorby návrhu. V následující kapitole si uvedeme co bylo testováno, jaké byly vyvozeny výsledky a jak dané testy ovlivnili aktuální verzi aplikace. Dále budou popsány objevené nedostatky, které se nestihly vyřešit a možnosti budoucího rozvoje aplikace.

6.1 Testování kompatibility

Testování kompatibility mobilních aplikací je důležitou částí vývoje, jelikož je zapotřebí otestovat chování aplikace na odlišných velikostech displeje a různých verzích operačního systému. Uživatelské rozhraní je tedy zapotřebí vytvořit dostatečně responzivní a je zapotřebí zajistit funkci všech prvků v rozsahu všech podporovaných verzí.

Testování probíhalo převážně pomocí emulátoru dostupného ve vývojovém prostředí Android Studio. Výhodou emulátoru je dostupnost různých velikostí displeje a všech verzí operačního systému Android. Rovněž byla snaha o co nejvyšší testování na fyzických zařízeních. Cílem bylo získání dat ze skutečného použití aplikace, kdy je uživatelem ovládána obvyklým způsobem, nikoliv myší. Tyto informace jsou důležité pro získání přehledu o dostupnosti jednotlivých tlačítek.

Responzivita byla testována nastavením různých velikostí displeje. Převážně se jednalo o hraniční hodnoty. Tedy nejmenší a největší možnou šířku nebo výšku. Bylo sledováno jak jsou data zobrazována a zdali nedojde ke schování některých informací. Největší problém tvořila data generována uživatelem, jelikož mohou nabývat různých délek. Na základě testování bylo vytvořeno řešení problematiky těchto dat v různých situacích. V případě rychlého přehledu nemusí být data přesné a jsou tedy zaokrouhlena na počet míst. Naopak u podrobného výpisu transakce je zapotřebí data zobrazit v celé jejich hodnotě. Při testování různých šířek bylo nicméně zjištěno, že opravdu dlouhé hodnoty můžou překročit hranici displeje a nejsou tedy dostupné. Proto bylo využito nastavení `marquee`, jenž zajistí, že se text zobrazovaný ve výstupním poli (v Android Studiu komponenta `TextView`) bude pohybovat zprava doleva v určitém intervalu.

Nadále bylo zapotřebí otestovat funkčnost všech knihoven a funkcí pro všechny podporované verze. Na základě testování byly objeveny případy, kdy určité funkce nebyly podporovány na nižších verzích operačního systému. Bylo zapotřebí nalézt alternativní řešení

dané funkce pro tyto verze a využít podmínky určující, kterou funkci vykonat na základě úrovně API, jež je uvedena v ukázce 3.1.

6.2 Testování funkcionality s využitím uživatelů

Testování funkcionality probíhalo v průběhu celé implementace. Každou implementovanou část bylo zapotřebí důkladně ověřit, než bylo přikročeno k dalším částem. Jednotlivé části jsou na sebe navázány a chybné fungování jedné části tedy ovlivní i zbytek aplikace. Testování probíhalo na základě různých případů a často se zaměřovalo na chování aplikace při zadání extrémních hodnot. Pro testování vytvořených transakcí, včetně operací s nimi provedenými, bylo zapotřebí sledovat změnu dat v databázi. Zdali data odpovídají očekávanému výsledku. Pro zobrazení dat byl využit program DB Browser for SQLite¹.

Popsané testování s náhledem do databáze samozřejmě probíhalo při vývoji a nebyli do něj začleněni uživatelé. Nicméně bylo zapotřebí provést velké množství testů s jejich začleněním. Jakožto autor aplikace očekávám, jak se bude aplikace chovat při různých vstupech a znám postup použití, testy tedy nemají z tohoto pohledu velkou vypovídající hodnotu. Jsou užitečné pouze k rychlému nalezení nedostatků a chyb. Pro získání dat simulujících skutečné použití aplikace je zapotřebí využít osob, jež se nepodílí na vývoji. Takto lze získat lepší přehled o skutečném uživatelském prožitku při používání aplikace a o podrobnějších nedostatcích či problémech.

Způsobů, jak provést testování s uživatelem, je mnoho. V mé práci probíhaly tyto typy testování:

- **Testování návrhu a uživatelského rozhraní** – Tento typ testování byl proveden při vytváření návrh a následného uživatelského rozhraní aplikace. Na základě výsledků testování bylo vyvíjeno uživatelské prostředí aplikace. Dále sloužilo k výběru použité komponenty nebo typu zobrazení v případě více možných řešení.
- **Testování nad snímkem obrazovky** – Jedná se o velice podobné testování jako při návrhu uživatelského rozhraní. Rozdíl je tvořen způsobem a cílem testování. Při testování uživatelského rozhraní často docházelo k porovnání dvou a více návrhů a hledání silných a slabých stránek daného návrh a uživatelského rozhraní. Toto testování zkoumalo již vytvořené rozhraní. Uživatel viděl snímek obrazovky a byl dotázán, kde a jak vykoná určitou akci. Cílem bylo zkoumání orientace na daném okně.
- **Testování na provedení zadané akce** – Testování bylo zaměřeno na zkoumání specifických akcí. Uživateli byly poskytnuty základní informace o fungování aplikace a cíl, kterého má dosáhnout. Na základě sledování uživatele byla provedena analýza postupu k dosažení daného cíle. Zdali uživatel rozpoznal, kam patří zadaná údaje nebo věděl, jak danou akci vykonat.
- **Testování bez vysvětlení** – Na rozdíl od předchozího testování uživateli nebyly poskytnuty informace o aplikaci. Uživatel pouze věděl, co je cílem aplikace a bylo mu zadáno jakou část má testovat. Cílem tohoto testu bylo zkoumání, zdali uživatel pochopil fungování dané části jako celku. Dále nalezení chyb a nedostatků zkoumané části, jelikož zde nebyl uživatel omezen zadanými příklady.

¹<https://sqlitebrowser.org/>

- **Kompletní testování** – V případě kompletního testování měl uživatel seznam transakcí, které má evidovat. Dále typ operací, které s nimi provede. A nakonec měl vygenerovat výpis za dané období. Při tomto testování byl sledován postup, jakým uživatel informace eviduje a upravuje. Dále byla sledována míra orientace v aplikaci a správnost výsledku. Uživatel byl dále dotázán k nalezení různých informací.

Uživatelé rovněž mohli aplikaci používat bez určených cílů. Výsledkem těchto testů bylo často nalezení drobných chyb, jež nemusely vyjít najevo z testovacích případů.

Na základě provedených testů došlo k úpravě popisu určitých částí, uživatelského prostředí i funkcionality aplikace. Dále byly objeveny nedostatky, které významně neovlivní chod aplikace a na jejichž opravu již nebyl čas. Tyto nedostatky budou uvedeny v následující sekci. Nakonec byla navrhována i možná budoucí rozšíření.

6.3 Objevené nedostatky

Na základě testování byly objeveny dva nedostatky, které výrazně neovlivňují chod aplikace, pouze snižují uživatelský prožitek při jejich výskytu. Oba nedostatky jsou spjaté s načítacím oknem aplikace, které má za úkol stáhnout a aktualizovat data z internetu.

První nedostatek souvisí přímo se stažením. Při prvním spuštění aplikace je zapotřebí připojení k internetu. Není-li uživatel připojen, je o tom informován. Nicméně po připojení k internetu není tato událost detekována a je zapotřebí aplikaci opakovaně zapnout. V budoucím rozšíření je jedním z hlavních cílů přidat detekování změny stavu připojení a vytvoření reakce, jež započne stahování dat.

Druhým nedostatkem je otevření načítacího okna po návratu do aplikace z pozadí. I přesto, že tato událost neovlivní chod aplikace, může ovlivnit uživatelský prožitek. Do budoucna bych rád eliminoval otevření tohoto okna pouze při prvním načtení aplikace.

6.4 Budoucí rozšíření aplikace

V následující podkapitole budou popsána rozšíření, která jsou plánována pro budoucí rozvoj aplikace. Jednotlivá rozšíření vychází z výsledků testování aplikace, zkoumání již existujících aplikací i z opravy nedostatků.

Prvním cílem budoucího vývoje bude oprava nedostatků zmíněných v předešlé podkapitole. Plánovaný způsob jejich opravy byl již popsán u jednotlivých nedostatků.

Další rozšíření vychází z průzkumu již existujících aplikací. Jedná o rozšíření způsobu evidence transakcí. Současná verze aplikace podporuje pouze manuální přidávání. V průběhu vývoje byla zkoumána myšlenka vyplnění údajů pomocí přiloženého snímku transakce. Bylo provedeno i počáteční testování technologie. Nicméně od ní bylo nakonec upuštěno. V budoucím rozšíření bych rád implementoval možnost propojení aplikace s různými burzami, poskytujícími obchod s kryptoměny, za využití API. Na základě tohoto propojení by došlo k získání provedených transakcí.

Dále je plánováno přidání tlačítka **Prodat vše** u vytvoření transakce. Toto tlačítko automaticky nastaví prodané množství na celkovou hodnotu vlastněné kryptoměny. Uživatel v případě celkového prodeje tedy nemusí vypisovat částku manuálně. Opět se jedná o funkci inspirovanou již existujícími aplikacemi.

Posledním plánovaným rozšířením je implementování možnosti pro přidání vlastní kryptoměny. Toto rozšíření vychází z poznatků při testování, kdy byl nadnesen dotaz, zdali je možné tento úkon vykonat. Jak již bylo zmíněno, seznam je s ohledem na počet kryptoměn

stažen pouze jednou. Může tedy nastat situace, kdy uživatel obchoduje s novou kryptoměnou, která v době stažení dat ještě nebyla součástí API. Rovněž zde nemusí být evidovány všechny kryptoměny. Uživatel by touto funkcí dostal příležitost tento nedostatek eliminovat.

Kapitola 7

Závěr

Obchod s kryptoměny je stejně, jako i jiné obchody, zapotřebí při překročení určitého obratu přiznat. Nicméně evidence těchto operací nemusí být snadná a může zabrat spoustu času. Cílem mé bakalářské práce bylo zaměřit se na danou problematiku a její usnadnění pro uživatele. Tedy na správu transakcí s kryptoměny.

Nejdříve bylo zapotřebí prostudovat postupy evidence transakcí s kryptoměny. Poté došlo k nalezení a prozkoumání existujících řešení. Zjistil jsem, že v oblasti mobilních aplikací daná problematika vytváří takzvanou díru na trhu. Pouze jediná zkoumaná aplikace splňovala dané cíle, jednalo se nicméně o webovou aplikaci, nikoliv mobilní. I přesto, že zbylé aplikace nesplňovaly přesný cíl mé práce, obsahovaly zajímavá řešení, která byla více či méně implementována při vývoji aplikace.

Dalším krokem bylo vytvoření návrhu a vypracování uživatelského rozhraní. Již od počátku byla aplikace konzultována se skupinou lidí, kteří ji testovali. Uživatelské rozhraní bylo upravováno na základě zpětné vazby ve snaze dosažení jednoduché a přehledné aplikace, jež splňuje dané cíle.

Implementace splnila všechny cíle práce. Uživateli stačí pouze přidat své transakce a následně si vygenerovat již propočtený přehled za zvolené daňové období. Je tedy zcela oproštěn od časově náročných výpočtů správy transakcí. Tento přehled je navíc vygenerován jako PDF dokument, který uživatel může snadno přeposlat například svému účetnímu. V rámci zadání bakalářské práce byl rovněž vytvořen plakát práce a video¹ popisující aplikaci.

Jelikož se aplikace zabývá neustále rostoucím odvětvím, a vyplňuje určitou díru na trhu, tak bych ji chtěl v budoucnu nadále rozvíjet. Mezi hlavní cíle patří propojení aplikace s kryptoměnovými burzami. Uživatel by nemusel nadále evidovat transakce manuálně, ale mohl by je přímo nahrát. Dalším plánovaným vylepšením je eliminace objevených nedostatků, které sice neovlivňují chod aplikace, ale mohou ovlivnit uživatelský prožitek.

I přesto, že se nejedná o mou první aplikaci pro mobilní zařízení, se zajisté jedná o jednu z nejrozsáhlejších. To platí i pro počet uživatelů, jenž sloužili jako testeři. Míra zapojení těchto uživatelů do návrhu aplikace a vývoje uživatelského rozhraní pro mne byla zcela novou zkušeností, ze které si odnáším spoustu poznatků.

¹https://youtu.be/CJr1R35m_L4

Literatura

- [1] CHAPMAN, C. *Why Use Material Design? Weighing the Pros and Cons* [online]. [cit. 2022-04-18]. Dostupné z: <https://www.toptal.com/designers/ui/why-use-material-design>.
- [2] GOOGLE INC.. *Introduction* [online]. [cit. 2022-04-18]. Dostupné z: <https://material.io/design/introduction>.
- [3] GOOGLE INC.. *Save data in a local database using Room* [online]. [cit. 2022-04-19]. Dostupné z: <https://developer.android.com/training/data-storage/room>.
- [4] GOOGLE INC.. *Meet Android Studio* [online]. Google Developers, duben 2022 [cit. 2022-04-18]. Dostupné z: <https://developer.android.com/studio/intro>.
- [5] JAGDALE, D. *Firebase – Introduction* [online]. GeeksforGeeks, červenec 2021 [cit. 2022-04-19]. Dostupné z: <https://www.geeksforgeeks.org/firebase-introduction/>.
- [6] JAVATPOINT. *PDFBox Tutorial* [online]. [cit. 2022-04-19]. Dostupné z: <https://www.javatpoint.com/pdfbox-tutorial>.
- [7] KOTAČKA, V. *Gradle, moderní nástroj na automatizaci* [online]. Devel.cz Lab s.r.o., červen 2013 [cit. 2022-04-18]. Dostupné z: <https://zdrojak.cz/clanky/gradle-moderni-nastroj-na-automatizaci/>.
- [8] KRUG, S. *Don't Make Me Think, Revisited*. 3. vyd. New Riders, 2014. ISBN 978-0-321-96551.
- [9] LOCKWOOD, A. *ShapeShifter* [online]. GitHub, 2018. Dostupné z: <https://github.com/alexjlockwood/ShapeShifter>.
- [10] MICROSOFT CORPORATION. *Co jsou databáze NoSQL?* [online]. [cit. 2022-04-19]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/nosql-database/>.
- [11] OL_V_ER a SIARHEI. *Retrieving Android API version programmatically* [online]. Stackoverflow.com, říjen 2010 [cit. 2022-04-18]. Dostupné z: <https://stackoverflow.com/a/3940823>.
- [12] ORACLE CORPORATION. *What is a Relational Database (RDBMS)?* [online]. [cit. 2022-04-19]. Dostupné z: <https://www.oracle.com/database/what-is-a-relational-database/>.
- [13] SERVIDONI, A. *Android Basics* [online]. Avenue Code, únor 2017 [cit. 2022-04-20]. Dostupné z: <https://blog.avenucode.com/android-basics-activities-fragments>.

- [14] SHA, A. *APK vs AAB (Android App Bundles): Everything You Need to Know!* [online]. Beebom Media, červenec 2021 [cit. 2022-04-18]. Dostupné z: <https://beebom.com/apk-vs-aab/>.
- [15] SINGH, J. *Android SDK and it's Components* [online]. GeeksforGeeks, únor 2021 [cit. 2022-04-20]. Dostupné z: <https://www.geeksforgeeks.org/android-sdk-and-its-components/>.
- [16] WALKER, J. *APK vs App Bundle: Why Is Google Changing Android's App Format?* [online]. LifeSavvy Media, červenec 2021 [cit. 2022-04-18]. Dostupné z: <https://www.cloudsavvyit.com/12544/apk-vs-app-bundle-why-is-google-changing-androids-app-format/>.
- [17] ZENDULKA, J. a BARTÍK, V. *1. Úvod, základní pojmy databázových technologií* [online]. FIT VUT v Brně [cit. 2022-04-18]. Dostupné z: https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIDS-IT%2Flectures%2Fcz%2F1_uvod.pdf&cid=13974.
- [18] ŠROTÝŘ, M. *Datová integrita, relační algebra* [online]. FD CVUT v Praze [cit. 2022-04-18]. Dostupné z: https://www.fd.cvut.cz/personal/srotymar/db/03-datova_integrita_relacni_agebra.pdf.

Příloha A

Plakát

Součástí mé práce bylo vytvoření propagačního plakátu, který si můžeme prohlédnout v následující příloze. Plakát je rovněž přiložen ve své původní velikosti na paměťovém médiu.

Mobilní aplikace
pro správu transakcí
s kryptoměny

Bakalářská práce 2022
Pavel Bobčík
prof. Ing. Adam Herout, Ph.D.

Úvodní obrazovka
Skutečný roční obrat
Vlastněné kryptoměny
Barevný mód

Vytvoření transakce
Nákup
Prodej
Směna

Seznam transakcí
Seznam všech transakcí
Stručný rychlý přehled
Možnost filtrování

Výpis transakce
Podrobný výpis transakce
Možnost úpravy transakce
Výpis historie změn

Přehledný výpis
Automatický výpočet
Výpis v PDF dokumentu
Otevření v PDF čtečce

Krypto Evidence

Obrázek A.1: Ukázka vytvořeného plakátu pro prezentaci aplikace Krypto Evidence.