



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Multiplatformní aplikace pro sledování osobních výdaj
Student:	Bc. Jaroslav Šev ík
Vedoucí:	Ing. Jan Václavík
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Cílem práce je návrh a implementace mobilní aplikace pro správu osobních výdaj s možností importu dat nebo napojení na API banky.

Aplikace umožní automaticky t ídit transakce podle nastavených pravidel do jednotlivých kategorií, bude pro n vykreslovat grafy, po ítat p íjmy, výdaje a další zajímavé ekonomické ukazatele. Bude také možné v aplikaci propojit více ú t a to i ve více m nách.

Nastavení aplikace a pravidla pro t ídn ní transakcí se ukládají na server, ale samotné transakce se na server z d vod soukromí neposílají.

K realizaci aplikace bude použita technologie React Native. Výsledná aplikace bude dostupná pro platformy iOS a Android a její zdrojové kódy budou zve ejn ny na Githubu.

Práce bude obsahovat také řešerši existujících ešení a podrobnou analýzu možností napojení na API bank a nebo formát pro import dat.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 12. prosince 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Multiplatformní aplikace pro sledování osobních výdajů

Bc. Jaroslav Ševčík

Vedoucí práce: Ing. Jan Václavík

4. května 2017

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mé práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 4. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Jaroslav Ševčík. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Ševčík, Jaroslav. *Multiplatformní aplikace pro sledování osobních výdajů*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato práce se zabývá návrhem a implementací multiplatformní mobilní aplikace pro správu osobních výdajů. Implementace je provedena v jazyce JavaScript s použitím nástrojů React Native a Expo. Klíčovými funkcemi jsou: import transakcí z banky, automatická kategorizace transakcí, možnost vést více účtů v různých měnách a vykreslování přehledových grafů. Součástí práce je také analýza možností napojení na API bank, zejména v České republice.

Klíčová slova Multiplatformní aplikace, Mobilní aplikace, Správa osobních výdajů, Android, iOS, Bankovní API, Expo, React Native

Abstract

The thesis deals with the design and the implementation of a multiplatform mobile application for personal expenses management. The application is implemented in JavaScript with the use of React Native and Expo. The key functions are: importing of transactions from a bank, an automatic categorization of the transactions, a possibility to keep many accounts in various currencies, and rendering of overview charts. Another part of the thesis is an analysis of the possibilities of connecting with banking APIs, especially in the Czech Republic.

Keywords Multiplatform application, Smartphone application, Personal expenses management, Android, iOS, Banking API, Expo, React Native

Obsah

Úvod	1
1 Cíl práce a metodika	3
2 Rešerše existujících řešení	5
2.1 Rozbor jednotlivých aplikací	7
2.2 Závěr	19
3 Analýza možností napojení na API bank	23
3.1 Možnosti komunikace programu s bankou	24
3.2 Rozbor bankovních API v ČR	26
3.3 Závěr	36
4 Analýza a návrh	39
4.1 Analýza požadavků	39
4.2 Případy užití	41
4.3 Architektura a design	45
5 Realizace	67
5.1 Použité technologie	67
5.2 Uživatelské rozhraní	71
5.3 Ukázky kódu	71
Závěr	81
Literatura	83
Použité programy	99
A Seznam použitých zkratk	101

Seznam obrázků

4.1	Model případů užití	42
4.2	Výdaje podle kategorie	46
4.3	Výdaje v čase	46
4.4	Transakce	47
4.5	Účty	47
4.6	Rozpočty	48
4.7	Nová transakce	48
4.8	Výběr kategorie	49
4.9	Nový převod	49
4.10	Nový účet	50
4.11	Nový rozpočet	50
4.12	Navigace — přehledy výdajů	51
4.13	Navigace — účty	51
4.14	Navigace — rozpočty	52
4.15	Kontextový model	53
4.16	Model funkční struktury	54
4.17	Entity-relationship model — aplikace	56
4.18	Entity-relationship model — server	57
4.19	Návrhový model tříd — ExpensesView	58
4.20	Návrhový model tříd — přehled výdajů	59
4.21	Návrhový model tříd — přehled výdajů v čase	59
4.22	Návrhový model tříd — transakce	60
4.23	Návrhový model tříd — účty	60
4.24	Návrhový model tříd — rozpočet	61
4.25	Model procesu — přidání účtu	62
4.26	Model procesu — import transakcí	63
4.27	Model procesu — přidání kategorizace	64
4.28	Model procesu — kategorizace transakcí	64
4.29	Model nasazení	65

5.1	Účty	72
5.2	Výdaje podle kategorie	72
5.3	Výdaje v čase	73
5.4	Transakce	73
5.5	Nový výdaj	74
5.6	Autentizace u banky	74
5.7	Příklad komponenty <i>BalanceBar</i>	75
5.8	Hlavní navigace	77
5.9	Navigační lišta	77

Seznam tabulek

2.1	Cílové platformy a lokalizace	20
2.2	Způsoby importu transakcí	21
2.3	Funkce	22
3.1	Stav vývoje otevřeného API v bankách působících na území České republiky	37

Úvod

Bankovníctví prochází v posledních letech masivními změnami. Příslušníci takzvaných generací Y a Z žijí v digitálním světě a chtějí v něm mít vše na dosah ruky, včetně bankovních služeb. Internetové bankovníctví už nestačí. Zejména generace Z, která nyní přichází do zaměstnání a stává se tak ekonomickou silou, tráví nejvíce času na svých chytrých telefonech. Změnil se také přístup k poskytovatelům bankovních služeb. Pro zákazníky už není takový problém přejít od jednoho poskytovatele k jinému, který nabízí lepší služby. Kvůli těmto skutečnostem jsou banky pod velkým tlakem na stálou inovaci svých produktů a služeb.[1, 2, 3]

Nastalou situaci využívají nové technologicky založené firmy, které nabízejí finanční služby netradičním, pro uživatele pohodlným, způsobem. Tyto firmy však často potřebují pro své produkty podporu finančních ústavů a pokud ji vyloženě nevyžadují, tak jim stále může výrazně usnadnit práci. Banky jsou této spolupráci stále více nakloněny právě kvůli tlaku na neustálou inovaci, kterou nemusí sami stíhat. Otevření bankovníctví třetím stranám je někde (např. v EU) podpořeno dokonce legislativně.[4, 5, 3, 2]

Z tohoto trendu těží i aplikace nazvaná Expenses Manager, jejíž vývoj je stěžejní náplní této práce. Ta nabízí neustálý přehled nad výdaji a jejich rozložením (kdy a za co). Výdaj financí je totiž dnes tak snadný a leckdy zcela „neviditelný“ (např. bezkontaktní platba kartou, automatické trvalé platby), že může být těžké nad nimi udržet kontrolu. Pro poskytnutí vysokého uživatelského komfortu, zejména v podobě automatizace klíčových procesů, se využívá právě programového napojení na jednu z našich bank, která to v současnosti umožňuje. Jedná se navíc o aplikaci mobilní, tedy cílenou na hlavní prostředek komunikace současné generace.

Jedním z příkoří vývoje mobilní aplikace jsou rozdíly mezi platformami. Vývojář se může rozhodnout pouze pro jednu platformu, tím ale ztratí mnoho potenciálních zákazníků. Nebo může implementovat více aplikací, každou pro jinou platformu. Tím se zase výrazně navýší nároky na prostředky a znalosti nutné k vývoji. Dále lze obejít platformní rozdíly vytvořením webové apli-

kace přizpůsobené pro mobilní prostředí. Tím se zase vývojář vzdá některých funkcionalit dostupných pouze nativním aplikacím, maximálního výkonu a toho správného „*look and feel*“, na které jsou uživatelé těchto zařízení zvyklí. Poslední možností je využít některého „cross-platform“ frameworku, který umožňuje vyvíjet nativní aplikace v jednom jazyce a do jisté míry také využít jednoho kódu pro více platforem.[6, 7]

Zhruba od konce roku 2012 se neustále zvětšuje podíl užívání JavaScriptu oproti ostatním programovacím jazykům.[8] Nemalý podíl na tom jistě bude mít vznik mnoha frameworků umožňujících snadný vývoj i pro mobilní zařízení.[9, 10] Většina takových frameworků využívá webview, formu integrovaného prohlížeče. Jde vlastně o webové aplikace, kterým je ale díky nástrojům jako Cordova nebo PhoneGap umožněno do jisté míry využívat nativních možností zařízení. Grafické rozhraní je však vykreslováno stejně jako u webové aplikace s použitím HTML a CSS. Druhou skupinou jsou frameworky, které plně využívají nativní komponenty platformy. Výsledné produkty jsou pak svým chováním, vzhledem a výkonem blíže skutečně nativním aplikacím.[6, 7, 11]

Jedním z těchto frameworků, který v posledních dvou letech vzbudil rozruch ve světě mobilního vývoje, je React Native od firmy Facebook. React Native navazuje na úspěch webového frameworku React, extrahuje jeho základní postupy a přenáší je na mobilní platformy Android a iOS. Jako své hlavní přednosti uvádí možnost vystavět bohaté uživatelské prostředí z deklarativních komponent, skutečnost, že výsledkem je skutečně nativní aplikace, vývoj bez nutnosti rekompilace a další. Jde stále o velmi mladou technologii, která ještě nedosáhla první stabilní verze. Přesto je již používána v produkci u aplikací jako Facebook nebo Instagram.[12] Zdá se, že jde o velmi slibnou technologii, která má potenciál zahýbat mobilním vývojem. I proto byla vybrána pro implementaci zadané aplikace.

Cíl práce a metodika

Cílem této práce je návrh a implementace mobilní aplikace pro správu osobních výdajů s možností napojení na API banky. Aplikace bude umožňovat importovat transakce přes internet, přiřazovat jim automaticky kategorie a následně vykreslovat grafy a další užitečné přehledy. Bude také možné v aplikaci propojit více účtů a to i ve více měnách. Cílové platformy jsou Android a iOS. Aplikace bude implementována v jazyce JavaScript s použitím frameworku React Native.

Nejprve bude provedena rešerše existujících řešení za účelem zjištění situace na trhu a zda má vůbec vývoj takovéto aplikace smysl. Následně budou analyzovány možnosti programového napojení na banky a to zejména na ty české prostřednictvím webového API. Tím se zjistí, zda je implementace takovéto aplikace proveditelná. V případě, že možnost takového napojení nebude našimi finančními ústavy podporována, bude aplikace data o transakcích importovat ze souborů.

Po provedení nezbytných analýz se přistoupí k návrhu aplikace. Podklady pro návrh budou získány od vedoucího práce, který tím přebírá úlohu zákazníka. Návrhy se budou týkat zejména uživatelského rozhraní, struktury aplikace, komunikace aplikace s externími systémy, datové vrstvy a klíčových procesů. Po vytvoření návrhů bude provedena samotná implementace, což bude demonstrováno jednak fungující aplikací, jejíž kód bude zveřejněn, a jednak vybranými okomentovanými ukázkami kódu. Tyto ukázky spolu s analýzou použitých technologií budou dále demonstrovat pochopení látky, která souvisí s náplní této práce.

Rešerše existujících řešení

Nápad vytvořit mobilní aplikaci pro správu výdajů není zcela originální. Je tomu spíše naopak. Po chvíli hledání v obchodech s aplikacemi majoritních poskytovatelů lze dohledat stovky existujících řešení. V drtivé většině jde ovšem o řešení umožňující pouze ruční zadávání transakcí a všech údajů o nich. Řešení, která kromě importu transakcí z banky kombinují také ostatní klíčové funkce mé aplikace, jako je automatická kategorizace nebo podpora účtů ve více měnách, je již výrazně méně. České prostředí včetně našich finančních ústavů potom podporuje jen hrstka aplikací, u kterých je třeba za tyto služby platit relativně vyšší poplatky.

Detailně rozebrat všechna konkurenční řešení bylo vzhledem k jejich množství a rozsahu této práce nereálné. Objem zkoumaných aplikací jsem proto zmenšil stanovením následujících kritérií:

- Musí to být samostatné mobilní aplikace, nikoliv PC či webové aplikace nebo mobilní aplikace, které slouží pouze jako doplněk k verzi pro stolní počítač.
- Musí to být aplikace pro jednotlivce a rodiny, nikoliv aplikace pro firmy.

Takovýchto řešení je stále velmi mnoho. Jako reprezentativní vzorek jsem vybral vždy prvních deset aplikací z obchodů dvou majoritních poskytovatelů, Google Play a iTunes Store, které mi byly zobrazeny po zadání následujících termínů do vyhledávače těchto obchodů:

- personal finance
- osobní finance
- expenses manager
- expense manager
- správa financí

2. REŠERŠE EXISTUJÍCÍCH ŘEŠENÍ

- spending
- accounting
- účetnictví
- personal accounting
- budget
- rozpočet

Pokud to bylo možné (podpora našeho regionu vydavatelem, operační systém Android), všechny aplikace pro první čtyři vyhledávací termíny jsem si nainstaloval a důkladně prošel. U ostatních jsem čerpal z internetových zdrojů, zejména z jejich oficiálních stránek, popisu v obchodu a případně i recenzí třetích stran. Dále jsem prošel také aplikace zahrnuté v seznamech doporučených aplikací tohoto typu od několika webových médií. Volil jsem pouze články mladší jednoho roku. Jde o následující články:

- 10 best Android budget apps for money management[13]
- TOP3 mobilní aplikace pro správu osobních financí[14]
- 13 Best Budget Apps for 2017[15]
- Mějte svůj (rodinný) rozpočet pod kontrolou s těmito aplikacemi[16]
- 7 cool personal finance apps for iPhone[17]

U vybraných aplikací jsem sledoval zejména následující charakteristiky, které přímo konkurují mému programu.

- Cílové platformy
- Možnost kategorizace transakcí
- Možnost importu transakcí přímo z finančního ústavu
- Možnost importu transakcí ze souboru exportovaného z internetového bankovníctví
- Automatická kategorizace transakcí
- Nabízené přehledy a grafy
- Možnost stanovení rozpočtů
- Nabízené měny

- Možnost vést více účtů v různých měnách
- Podporované jazyky
- Cena

Nyní rozeberu jednotlivě vybrané aplikace. Jde o aplikace, které jsem prozkoumal nejdůkladněji (první čtyři vyhledávací termíny) a které nejvíce odpovídají mojí práci. Mimo nich jsem vybral ještě další, které se lišily od ostatních v nějaké z klíčových funkcionalit či jinak přinášely něco nového. Pro lepší celkový přehled pak poskytuji také tabulky 2.12.22.3 zobrazující dostupnost sledovaných funkcí u těchto aplikací.

2.1 Rozbor jednotlivých aplikací

2.1.1 Expense Manager od Bishinews

První zkoumanou aplikací je Expense Manager. Tato aplikace je součástí bohatého balíčku finančních funkcionalit propojených do jednoho rozhraní aplikací Personal Finance. Lze ji však nainstalovat i samostatně.

Tato aplikace je dostupná pro operační systém Android, verze s názvem EZ Expense Manager je dostupná také pro iOS. Transakce je možné zadávat pouze manuálně. Kategorie se transakcím přidělují také jen ručně.

Expense Manager nabízí velmi širokou škálu grafů a přehledů ve srovnání s konkurencí. Jsou to seznamy transakcí filtrovatelné přes nejrůznější kritéria — datum, kategorie, podkategorie, plátce/příjemce, vlastní štítky a další. Dále sloupcový nebo spojnicový graf s příjmy a výdaji rozloženými v čase, koláčový nebo sloupcový graf s výdaji podle kategorie či podkategorie, kalendář se součty příjmů, výdajů a bilancí v každém dni, spojnicový graf srovnávající vývoj celkových příjmů či výdajů v různých měsících, sloupcový graf s příjmy, výdaji a bilancí po měsících a další. Je zde i možnost vytvořit si vlastní graf sledující vybrané statistiky. Takovéto možnosti jsou skutečně nadprůměrné.

V aplikaci je možnost nastavit si rozpočet pro vybrané kategorie a to na libovolné období. Uživatel pak může sledovat plnění rozpočtu a je upozorněn, pokud se výdaje přiblíží stanovené částce nebo ji překročí. Účty je možno vést v různých měnách, ale částky nejsou přepočítány do jedné měny, takže celkové přehledy nejsou relevantní. Aplikace nemá českou lokalizaci.

Celkově je Expense Manager jednou z nejkompexnějších aplikací, co se nabízí přehledů týče. Tomu odpovídá i větší složitost uživatelského rozhraní a složitější navigace. Produkt je zcela zdarma (obsahuje reklamy).[18, 19, 20]

2.1.2 Monefy od MonefyApp

Monefy je v kontrastu s Expense Manager velmi jednoduché. Je vyvíjené pro platformy Android, iOS a Windows Phone, kde nabízí pouze několik málo

funkcionalit. Přidávat a kategorizovat transakce lze opět pouze manuálně. Nabízen je jeden hlavní přehled a to výdaje podle kategorie ve formě koláčového grafu nebo seznamu. V seznamu lze také zobrazit transakce ve zvolené kategorii. Viditelná je ještě aktuální bilance a součty výdajů a příjmů.

V Monefy je možné nastavit jeden celkový rozpočet na měsíc. Místo součtu příjmů je pak zobrazen právě limit rozpočtu. Účty mohou být vedené v různých měnách. Aplikace opět není přeložena do češtiny.

Monefy vsází zejména na líbivé intuitivní rozhraní. Základní verze je zdarma. V placené verzi (70 Kč) se odemkne správa kategorií, možnost vést účty v různých měnách a online synchronizace s ostatními zařízeními.[21, 22, 23]

2.1.3 Mobills od MobillsLabs

Aplikace Mobills je vyvíjena pro širokou škálu platforem: Android, iOS, Windows Phone a web. Podporuje také řadu jazyků, včetně češtiny.

Transakce je možné zadat a kategorizovat manuálně nebo je lze naimportovat z SMS od banky. Nepodařilo se mi však dohledat, jaké banky jsou podporovány. Mobills nabízí poměrně pestrou nabídku grafů a přehledů: zůstatek na účtech, součty příjmů a výdajů, výdaje podle kategorie za zvolené období (koláčový nebo sloupcový graf), vývoj výdajů ve vybrané kategorii v čase (spojnicový graf), kalendář s příjmy a výdaji den po dni, bilance za různá období, seznam transakcí s možností filtrování podle různých kritérií, srovnání výdajů v různých obdobích a statistiky pro vybranou kategorii — největší měsíční výdaj, průměrné denní/roční výdaje a celkové výdaje.

V aplikaci je možno nastavit rozpočty podle kategorie a následně sledovat jejich plnění. Uživatel je také upozorněn na blížící se k limitu výdajů a aplikace sama doporučuje rozpočet a predikuje zůstatek podle výdajů v minulosti.

Mobills je rozsáhlejší aplikace s různými „chytrými“ funkcemi jako predikce zůstatku nebo upozornění na výdaje výrazně nad průměrem. Ve verzi zdarma je mnoho funkcí zamknutých. Plná verze stojí \$25 ročně.[24, 25, 26]

2.1.4 Money Lover od Finsify

Money Lover podobně jako Mobills cílí na mnoho platforem: Android, iOS, Windows Phone a web. Také nabízí mnoho jazykových lokalizací včetně české a možnost vést více účtů v různých měnách.

Transakce lze importovat přímo z finančního ústavu. Podporovány jsou ale pouze banky z Vietnamu a okolních států, Citibank a PayPal. Transakce lze také zadat přečtením bankovní SMS zprávy. Podporované jsou opět vietnamské banky. V beta verzi je navíc dostupná funkce rozpoznání transakcí z paragonu.

Mezi nabízenými přehledy jsou tyto: výdaje podle kategorie (koláčový graf, seznam), celkové příjmy/výdaje v čase (sloupcový graf), bilance v čase (spojnicový graf), seznam transakcí, součet výdajů, součet příjmů, bilance a zůstatky

na účtech. Podobně jako u Mobills je možné zadat rozpočty podle kategorií, sledovat jejich plnění a být upozorněn na blížení se limitu.

Verze zdarma obsahuje reklamy a je omezena počtem rozpočtů a účtů a pár vedlejších funkcí je zamknutých. Premium verze stojí \$5 a pro skenování účtenek je třeba dokupovat kredit (100 kreditů stojí \$9,75).[27, 28, 29]

2.1.5 ClevMoney od Clevenl Inc.

ClevMoney lze nainstalovat na zařízení se systémem Android. Účty lze vést jen v jedné měně. Do češtiny jsou přeloženy pouze názvy měsíců.

Transakce je možno zadávat několika způsoby. Kromě ručního zadání to je čtení SMS zpráv z banky a také čtení notifikací od bankovních aplikací. Pro tyto služby jsou však podporovány zejména korejské banky, ne české. Aplikace nabízí i automatickou kategorizaci na základně předchozích dat a klíčových slov zadaných k transakci.

Nabízené přehledy jsou následující: výdaje/příjmy podle kategorie, platební metody, dne v týdnu, hodiny nebo názvu transakce a to za libovolné období (koláčový graf, seznam seskupený po dnech), výdaje/příjmy v čase (sloupcový graf, seznam), seznam transakcí, ve kterém lze vyhledávat podle názvu a poznámky, kalendář s příjmy a výdaji ve dnech, zůstatek na účtech a součty příjmů a výdajů za měsíc. Aplikace umí také pracovat s rozpočty podle kategorie nastavené na různá období, jejichž plnění lze v průběhu sledovat.

ClevMoney je zdarma s veškerou funkcionalitou. Pro odstranění reklam je třeba zaplatit 140 Kč.[30, 31]

2.1.6 AndroMoney od AndroMoney

AndroMoney je vyvíjeno pro Android, iOS a web v několika jazycích, ale bez češtiny. Účty lze vést v různých měnách najednou.

Transakce se musí zadávat a kategorizovat manuálně. Všechny přehledy lze zobrazit za libovolné období a filtrovat podle mnoha různých kritérií. Nabízeny jsou tyto: výdaje podle kategorie (koláčový graf), výdaje podle kategorie a podkategorie (sloupcový graf), výdaje v čase po dnech, týdnech nebo měsících (spojnicový graf), zůstatky na účtech, transakce, kalendář s příjmy a výdaji ve dnech. Je umožněno nastavit rozpočty podle kategorie na různá období a uživatel bude upozorněn při překročení padesáti procent z limitu.

AndroMoney je zdarma s reklamou. Lze zakoupit i Pro verzi bez reklam a s novým vzhledem za 102 Kč.[32, 33, 34]

2.1.7 Money Manager od Realbyte

Money Manager je aplikace pro systém Android. Všechny účty musí být vedeny v jedné vybrané měně. Jediným podporovaným jazykem je angličtina.

Jediným způsobem zadání nových transakcí a jejich kategorizace je ruční vyplnění formuláře. Množství nabízených přehledů je podobné jako v předchozím případě. Jsou to: seznam transakcí po měsících, součet příjmů, výdajů po dnech, týdnech nebo měsících, bilance v aktuálním měsíci, kalendář se součty příjmů, výdajů a bilancí ve dnech, srovnání výdajů s předchozím měsícem, vývoj zůstatku na účtu (spojnicový nebo sloupcový graf), výdaje/příjmy podle kategorie za libovolné období (koláčový graf), výdaje v čase podle kategorie (spojnicový graf). Přehledy lze filtrovat podle účtu a typu transakce. V transakcích lze navíc vyhledávat podle období a textu. Jako u většiny konkurence i zde může být veden rozpočet podle kategorií a sledováno jeho plnění.

Základní verze je zdarma, pro odstranění reklam, limitu na počet účtů a odemčení pár okrajových funkcionalit je potřeba zaplatit 100 Kč.[35, 36, 37]

2.1.8 Alzex Personal Finance od Anna Shirokova

Cílové platformy aplikace jsou Android, iOS a Windows. Účty je možno vést v mnoha měnách, ale pouze mezi osmi z nich se provádí přepočty podle kurzů. Pro ostatní měny, mezi nimiž je i česká koruna, tedy celkové přehledy neukazují správné hodnoty. Čeština je podporována jen částečně. Např. názvy kategorií jsou stále anglicky.

Transakce je třeba zadávat a kategorizovat vždy ručně. Nabízené přehledy jsou následující: zůstatek na účtech, vývoj zůstatku (spojnicový graf), příjmy/-výdaje za libovolné období podle kategorie, osoby, plátce/příjemce a projektu (koláčový nebo sloupcový graf), příjmy/výdaje v čase za libovolné období podle stejných parametrů jako v předchozím případě (sloupcový nebo spojnícový graf), seznam transakcí za měsíc.

Rozpočty je možno nastavit podle mnoha kritérií — období, osoba, kategorie, příjemce a pod. Jejich plnění lze snadno sledovat, ale uživatel není nikdy upozorněn.

Aplikace nabízí rozsáhlé možnosti kategorizace — kategorie, podkategorie, pod-podkategorie, různé vlastní štítky. Základní verze bez většiny grafů, rozpočtů a dalších funkcí je zdarma, premium pak stojí 140 Kč.[38, 39, 40]

2.1.9 HandWallet Expense Manager od HandWallet Expense Manager

HandWallet Expense Manager lze nainstalovat na zařízení se systémem Android a iOS. Aplikace je přeložena do velmi mnoha jazyků včetně češtiny. Účty mohou být vedeny v různých měnách najednou.

Po stáhnutí doplňku je možné aplikaci naučit číst zprávy z banky a importovat z nich transakce. Transakce je možné importovat také ze souborů ve formátu QIF, QFX, OFX, ELS, XLSX, CSV, XML, IIF a dalších. Aplikace nabízí tyto přehledy, všechny za libovolné období: seznam transakcí filtrovatelný a seřaditelný podle mnoha kritérií — datum, účty, kategorie, osoby, projekty,

způsob platby, popis a další, zůstatky na účtech, vývoj zůstatku (sloupcový nebo spojnicový graf), výdaje podle kategorie (koláčový nebo sloupcový graf, seznam), srovnání příjmů a výdajů podle kategorie (dva koláčové nebo sloupcové grafy), výdaje/příjmy v čase (sloupcový nebo kruhový graf, seznam), srovnání výdajů a rozpočtu, plnění rozpočtů (koláčový nebo sloupcový graf).

Rozpočty podle kategorie, osoby nebo projektu lze zadat jako procento hlavního rozpočtu, fixní částku, vyrovnat rozpočet jiné kategorie a další. Na jejich překročení uživatel není upozorněn.

Jedná se o velmi mocnou aplikaci, co se nabízených přehledů a funkcí týče. Ovládání je ovšem výrazně složitější než u konkurence a v navigaci jsem se často ztrácel. Základní aplikace je zdarma s omezením na maximálně pět účtů vedených současně. Pro verze za \$35 navíc přidává export do souboru, projekty, vlastní pole u transakce a další funkce.[41, 42, 43]

2.1.10 Expense IQ Money Manager od Handy Apps

Aplikace je vyvíjena pro systém Android. Účty lze vést v různých měnách, ale najednou jde zobrazit jen přehledy nad účty ve stejné měně, stejně tak je to s rozpočty. Česká lokalizace není v nabídce.

Transakce mohou být importovány ze souboru ve formátu QIF nebo CSV. Transakce se automaticky kategorizují. Čím déle se aplikace používá, tím je kategorizace přesnější. Pro usnadnění zadávání jsou předvyplněné desítky předmětů transakce a prodejců.

Dostupné jsou tyto přehledy: výdaje podle kategorií (seznam), zůstatek na účtech, transakce podle účtu, srovnání příjmů a výdajů (koláčový graf), vývoj zůstatku v čase (spojnicový graf), výdaje v čase podle kategorie (sloupcový graf), příjmy/výdaje podle projektu a kategorie (koláčový graf). V aplikaci lze stanovit měsíční rozpočet podle kategorie a účtu. Zobrazuje se pak jeho plnění a v případě překročení limitu se zobrazí upozornění.

Základní aplikace je zdarma, premium verze bez reklam, omezení počtu účtů, rozpočtů, projektů a s dalšími funkcemi stojí 290 Kč ročně nebo 740 Kč jednorázově.[44, 45, 46]

2.1.11 Spendee od CLEEVIO

Spendee je dostupné pro platformy Android a iOS. Účtů může být vedeno více i v různých měnách. Všechny přehledy jsou ale vždy pouze přes jeden účet. Aplikace je dostupná i v českém jazyce.

Transakce je možné stahovat přímo z banky a to i z mnoha českých. Transakce jsou zároveň automaticky kategorizovány. Všechny vyjmenované přehledy jsou pouze pro jeden vybraný účet: seznam transakcí, zůstatek na účtu, bilance za sledované období, výdaje/příjmy podle kategorie (koláčový graf, seznam), bilance v čase (sloupcový graf).

Spendee nabízí méně přehledů a funkcí než většina předchozích aplikací, ale zato poskytuje službu importu z českých bank a automatickou kategorizaci. Základní aplikace je zdarma, verze plus se sdílením a neomezeným počtem účtů, rozpočtů stojí \$2 měsíčně, plná verze s napojením na banku pak vyjde na \$3 měsíčně.[47, 48, 49]

2.1.12 Wallet od BudgetBakers

Wallet cílí na platformy Android, iOS a web. Podporováno je mnoho jazykových variant včetně české. Účty mohou být vedeny i v různých měnách najednou.

Transakce lze importovat přímo z bank, včetně mnoha českých. Tyto jsou automaticky kategorizovány pomocí blíže nespécifikovaného machine learning algoritmu. Nabídka přehledů je širší než u většiny. Jsou to: zůstatky na účtech, vývoj zůstatku v čase (sloupcový nebo spojnicový graf), seznam transakcí za libovolné období seskupené podle kategorie, příjmy/výdaje podle kategorie (koláčový graf, seznam - lze rozkliknout do podkategorií), kumulativní výdaje (spojnicový), místa provedení transakce na mapě, cashflow trend — spojení vývoje zůstatku (spojnicový) a výdajů/příjmů v čase (sloupcový), cashflow tabulka — počty příjmů a výdajů, různé průměry za den, za transakci, zůstatky na začátku a konci období, bilance.

Aplikace umožňuje nastavit rozpočty podle účtů a kategorií (i více najednou) na různá období. Poté lze sledovat jejich plnění a uživatel je upozorněn při přiblížení se k limitu nebo jeho překročení.

Wallet je komplexnější aplikace s mnoha pokročilými funkcemi. Základní aplikace s mnoha chybějícími přehledy je zdarma. Verze Starter se všemi přehledy a napojením až na dva bankovní účty stojí 42,50 Kč měsíčně. Verze Master s neomezeně napojenými účty a sdílením s dalšími osobami stojí 65 Kč měsíčně.[50, 51, 52]

2.1.13 Fast Budget od FerApps - Mobile Solutions

Fast Budget je vyvíjený pro systém Android. Účty je možno vést v různých měnách, ale je třeba zadat ručně kurz pro převod do hlavní měny. Aplikace je dostupná i v češtině.

Transakce lze importovat ze souboru CSV a XLS ve specifikovaném formátu. Nabízené přehledy jsou následující: zůstatky na účtech, součet příjmů a výdajů (koláčový graf), bilance za sledované období, transakce (filtrovatelné podle kategorie, poznámky, doby opakování), výdaje/příjmy podle kategorie (filtrovatelné podle účtu) (koláčový nebo sloupcový graf, seznam), vývoj příjmů/výdajů podle kategorie v čase (sloupcový graf), kalendář s bilancemi ve dnech. Rozpočty lze nastavit na libovolné období a sledovat jejich plnění.

Základní verze je zdarma. Verze Pro s neomezeně účty, neomezeně rozpočty, bez reklam a pod. stojí 100 Kč. Pro verze se synchronizací stojí 340 Kč

ročně pro dvě synchronizovaná zařízení, 500 Kč ročně pro maximálně pět zařízení.[53, 54, 55]

2.1.14 Expense Manager od Markus Hintersteiner

Aplikace je dostupná pro systém Android. Je lokalizovaná do češtiny. Účty mohou být vedené pouze v jedné vybrané měně.

Transakce je nutné zadávat manuálně. Dostupné přehledy jsou tyto: výdaje podle kategorie (seznam, koláčový graf), transakce filtrovatelné podle kategorie, výdaje v čase podle kategorie srovnané s rozpočtem, průměrné výdaje za den v týdnu, měsíc. Lze nastavit měsíční rozpočty podle kategorie a sledovat jejich plnění.

Základní verze bez některých grafů a statistik je zdarma. Plná verze vyjde na 52 Kč.[56, 57, 58]

2.1.15 Expense Manager od Money Tower

Expense Manager od Money Tower lze nainstalovat na zařízení se systémem Android. Je možné pracovat jen s jednou vybranou měnou. Čeština není podporována.

Transakce musí být zadány ručně. Nabízeny jsou tyto přehledy: zůstatky na účtech, součet příjmů, součet výdajů, bilance, příjmy/výdaje po měsících (seznam), příjmy/výdaje podle kategorie za měsíc (seznam), kalendář s bilancí a zůstatkem na vybraném účtu ve dnech, příjmy/výdaje podle kategorie/plátce/příjemce pro vybraný účet (koláčový nebo sloupcový graf), srovnání příjmů a výdajů na vybraném účtu (koláčový nebo sloupcový graf), příjmy/výdaje spolu s bilancí po měsíci (sloupcový graf), výdaje za posledních sedm dní po dnech (sloupcový graf). V aplikaci je možné nastavit rozpočty pro jednotlivé kategorie na různá období a sledovat jejich plnění.

Verze s reklamami je zdarma. Jejich odstranění stojí 44 Kč. Dostupná je také téměř identická aplikace s názvem My Wallet od Balli. Ta má pouze graficky odlišnou domovskou obrazovku a odstranění reklam stojí 30 Kč.[59, 60, 61]

2.1.16 Expense Manager od CASorin

Tato aplikace je opět dostupná pouze pro Android. Pracovat lze najednou pouze s jednou měnou. Čeština není podporována.

Transakce musí být zadávány manuálně. Nabízeny jsou následující měsíční přehledy: příjmy, výdaje, bilance, kalendář s označením dnů s příjmy a výdaji, výdaje podle kategorie (sloupcový graf). Další přehledy jsou: dnešní transakce, součet příjmů/výdajů za libovolné období, výdaje podle kategorie za libovolné období (sloupcový graf), transakce filtrovatelné podle kategorie a období (seznam).

V Expense Manager od CASorin neexistuje koncept účtů. Jde o jednu z nejjednodušších ze zkoumaných aplikací a je dostupná zcela zdarma. Prakticky identický s touto aplikací je Expense Manager od Larisa Maria Berce pro iOS.[62, 63]

2.1.17 Wally od Wally Global

Wally je vyvíjen pro platformy Android (Wally+) a iOS. Najednou se používá pouze jedna měna, není zde systém účtů. Aplikace je dostupná ve třech jazykových variantách, ale bez češtiny.

Transakce se musí zadávat manuálně. Kategorizace je částečně automatická — aplikace se pokusí určit kategorii podle místa, kde byla transakce zadána. Přístup Wally ke správě výdajů je trochu jiný, čemuž odpovídají nabízené přehledy: lidé, se kterými se utrací, rozdělení do skupin jako rodina, práce, a pod. (koláčový graf), srovnání výdajů s ostatními uživateli aplikace, součet výdajů v posledních dnech (seznam), bilance, výdaje podle kategorie (koláčový graf, seznam), výdaje podle místa (seznam). Rozpočet se automaticky nastaví podle příjmů a plánovaných úspor. Na domovské obrazovce je pak zobrazeno, kolik je ještě možno utratit.

Wally se liší od ostatních podobných aplikací důrazem na to kde a s kým se utrací. Aplikace je zcela zdarma a bez reklam.[64, 65, 66, 67, 68]

2.1.18 MoneyCoach od MoneyCoach UG

MoneyCoach je vyvíjen pro systém iOS. V aplikaci lze používat více měn najednou. Čeština není podporována.

Transakce se musí zadávat vždy ručně. Mezi nabízenými přehledy jsou: součet příjmů, součet výdajů, kolik procent z příjmu je již utraceno, zůstatky na účtech, seznam transakcí, výdaje/příjmy podle kategorie (koláčový graf, seznam), srovnání měsíčních příjmů s výdaji, předpokládané měsíční výdaje, kalendář s příjmy a výdaji ve dnech. V aplikaci není možnost stanovení rozpočtu.

Aplikace nabízí méně funkcí než většina. V obchodu je k dostání zdarma.[69, 70]

2.1.19 Pocket Expense od Appxy

Pocket Expense lze nainstalovat na zařízení se systémem iOS. Čeština není podporována.

Transakce je nutné zadávat ručně. Nabízené přehledy jsou následující: zůstatky na účtech, seznam transakcí, výdaje podle kategorie (koláčový graf, seznam), součet příjmů, součet výdajů, kalendář s příjmy a výdaji ve dnech. V aplikaci je možnost nastavit rozpočty podle kategorie a sledovat jejich plnění.

Základní aplikace je zdarma. Pro verze stojí 5 Euro.[71, 72, 73]

2.1.20 CoinKeeper od Disrapp

CoinKeeper je vyvíjen pro Android, iOS a web. Účty lze vést i ve více měnách najednou. Aplikace je bez české lokalizace.

Transakce musí být zadány manuálně. Pro všechny přehledy lze nastavit různá časová období. Jsou to tyto: zůstatek na účtech, součet výdajů a příjmů, součet plánovaných výdajů, bilance po dnech, seznam transakcí řazený podle účtu nebo kategorie (pouze kategorie a cena), příjmy/výdaje podle kategorie filtrovatelné podle různých kritérií (koláčový graf nebo seznam), příjmy a výdaje v čase (spojnicový graf), průměrné a maximální měsíční příjmy a výdaje. Lze nastavit měsíční rozpočty podle kategorie a sledovat jejich plnění.

Základní aplikace je zdarma. Premium verze bez omezení počtu kategorií a štítků, se všemi statistikami a některými bonusovými funkcemi stojí 470 Kč na rok nebo 39 Kč na měsíc.[74, 75, 76, 77]

2.1.21 Spending Tracker od MH Riley

Spending Tracker cílí na platformy Android, iOS a Windows. Lze pracovat pouze s jednou měnou. Čeština v aplikaci není podporována a to ani český font — znaky s háčkem nebo čárkou jsou jiným fontem než ostatní.

Transakce se zadávají pouze ručně. Nabízené přehledy jsou vždy za týden, měsíc nebo rok. Dostupné jsou tyto: součet příjmů, součet výdajů, bilance, seznam transakcí podle účtu a kategorie, výdaje/příjmy podle kategorie (koláčový nebo sloupcový graf, seznam), výdaje a příjmy v čase (spojnicový graf). Lze nastavit celkový rozpočet a zobrazí se, kolik je ještě možno utratit.

Aplikace je zdarma. Pro odstranění reklam se musí zaplatit 67 Kč.[78, 79, 80, 81]

2.1.22 Cost Track od Artezio LLC

Cost Track lze nainstalovat na zařízení se systémem iOS. Lze vést účty ve více měnách najednou. Česká verze aplikace není.

Transakce je opět třeba zadat manuálně. Aplikace nabízí tyto přehledy: výdaje podle kategorie a místa (koláčový graf, seznam), zůstatky na účtech, seznam transakcí filtrovatelný podle kategorie, místa a času. Není možnost stanovit rozpočet.

Základní aplikace je zdarma. Plná verze bez omezení počtu účtů a zadáných transakcí stojí 5 Euro.[82, 83]

2.1.23 MoneyWiz Premium od SilverWiz

MoneyWiz cílí na platformy Android, iOS, Mac a Windows. Účty lze vést ve více měnách najednou. Aplikace je dostupná v mnoha jazycích včetně češtiny.

Transakce mohou být importovány přímo z finanční instituce. Podporováno je velké množství institucí včetně mnoha českých. Transakce lze im-

portovat také ze souborů s příponou QFX, QIF, OFX a CSV. Kategorizace transakcí probíhá automaticky.

Nabízených přehledů je velké množství. Jsou to: zůstatky na účtech, součet výdajů a příjmů, kalendář s plánovanými transakcemi, seznam transakcí s mnoha možnostmi řazení, vyhledávání a filtrování. Dále je to množství upravitelných tabulek, seznamů a grafů zobrazujících hodnoty za libovolná období, filtrovatelné podle různých kritérií. Mezi nimi jsou např.: vývoj zůstatku v čase (spojnicový graf), příjmy a výdaje (sloupcový graf), srovnání rozpočtů, zůstatků vůči rozpočtu, výdaje/příjmy podle kategorie a podkategorie (koláčový nebo sloupcový graf, seznam). Reporty se zpravidla načítají déle než u konkurence.

Rozpočty lze nastavit podle vícero kategorií a na různá období. Jejich plnění lze v průběhu sledovat a uživatel je upozorněn na přiblížení se k / překročení stanoveného limitu.

MoneyWiz je velmi komplexní aplikace s mnoha nastavitelnými prvky, přehledy a „chytrými“ funkcemi. Umožňuje např. také automatické placení plánovaných výdajů nebo vytvoření vlastního reportu. Omezená verze aplikace je zdarma. Plná verze s online bankovníctvím, bez reklam, bez omezení počtu účtů, rozpočtů a plánovaných transakcí stojí 1800 Kč ročně.[84, 85, 86, 87]

2.1.24 Money Pro od iBear LLC

Money Pro je vyvíjeno pro iOS a Mac. V aplikaci lze vést účty ve více různých měnách najednou. Dostupná je i česká verze.

Transakce mohou být importovány ze souboru OFX. Kategorie jsou přidělovány automaticky. Nabízené přehledy jsou následující: zůstatky na účtech, kalendář s označením dnů s příjmy a s výdaji, seznam transakcí s vyhledáváním podle různých kritérií, součet vybraných příjmů a výdajů za libovolné období (sloupcový nebo koláčový graf), vývoj zůstatku na účtech (spojnicový graf), výdaje/příjmy podle kategorie (koláčový nebo sloupcový graf, seznam, ikony s hodnotou). V aplikaci lze stanovit rozpočty podle kategorie na různá období a následně sledovat jejich plnění.

Zajímavou funkcí je předvídání transakcí bez zadání uživatelem. Základní aplikace stojí \$5. Pro synchronizaci s dalšími účty je třeba platit \$12 ročně.[88, 89]

2.1.25 Finance Lite od Ondřeje Brichty

Finance Lite lze nainstalovat na zařízení se systémem iOS. Je možno používat jen jednu vybranou měnu — není zde systém účtů. Aplikace je dostupná v češtině a angličtině.

Transakce musí být zadávány ručně a nejsou zde žádné kategorie. Nabízeny jsou tři druhy přehledů filtrovatelné podle účelu transakce a času: vývoj

zůstatku, příjmů a výdajů v čase (spojnicový graf), příjmy a výdaje (koláčový graf) a seznam transakcí. Aplikace je zdarma.[90, 91]

2.1.26 Expense Manager od Phi Dung

Aplikace je vyvíjena pro systém iOS. Účty je možno vést v jedné měně. Jediným podporovaným jazykem je angličtina.

Transakce je nutné zadávat manuálně. Mezi nabízenými přehledy jsou tyto: příjmy a výdaje v čase (sloupcový graf), příjmy/výdaje podle kategorie (koláčový graf), seznam transakcí. Nastavení rozpočtů není dostupné. Aplikace je zdarma.[92, 93]

2.1.27 Live Expenses od Bonazzi

Live Expenses lze nainstalovat na zařízení se systémem iOS. V aplikaci lze vést více účtů v různých měnách. Česká lokalizace není dostupná.

Transakce je nutné zadávat ručně. Jediným přehledem je seznam transakcí. V aplikaci není možnost nastavení rozpočtu.

Live Expenses slouží pouze k záznamu transakcí a jejich následnému odeslání v CSV nebo HTML. Základní verze je zdarma. Pro odemčení více měn je třeba zaplatit 2 Eura, pro přidání pole s účtem při zadávání transakce je třeba zaplatit 4 Eura.[94, 95]

2.1.28 Toshl Finance od Toshl

Toshl Finance je vyvíjeno pro Android, iOS, Windows a web. Účty je možno vést v různých měnách najednou. Aplikace nepodporuje češtinu.

Transakce je nutné zadávat ručně. Dostupných je několik běžných přehledů a dva neobvyklé grafy, všechny je možné filtrovat přes období a účet. Jsou to: seznam transakcí, součet příjmů, součet výdajů, výdaje/příjmy podle kategorie (koláčový graf, seznam), výdaje/příjmy podle štítků (bublinový graf — čím větší bublina, tím více výdajů/příjmů s daným štítkem), výdaje v čase za měsíc (sloupcový graf), průměrné denní výdaje, místa utrácení na mapě. Dalším specifickým přehledem je „river flow“ graf. Jde o graf zobrazující rozdělení prostředků z příjmů na úspory a rozpočtené výdaje. Rozpočtené výdaje se dále větví na skutečné výdaje a zbylé prostředky. Rozpočty je možno vést i podle vícero kategorií a následně sledovat jejich plnění.

Základní aplikace je zdarma. Pro verze s neomezeně účty, rozpočty a několika bonusovými funkcemi stojí 600 Kč.[96, 97, 98, 99]

2.1.29 Moneyboard od Damien Vieira

Moneyboard je vyvíjeno pro platformu iOS. Účty je možno vést ve vícero měnách najednou. Aplikace není dostupná v češtině.

Transakce musí být zadávány manuálně. Mezi nabízenými přehledy jsou: seznam transakcí s filtrem podle kategorie a vyhledáváním podle názvu, součet příjmů, součet výdajů, bilance, zůstatky na účtech, výdaje/příjmy podle kategorie (koláčový graf). V aplikaci lze nastavit měsíční rozpočet. Základní aplikace je zdarma. Pro verze vyjde na 3 Eura.[100, 101]

2.1.30 Frugi od Iman Ghasempour

Frugi lze nainstalovat na zařízení se systémem iOS. V aplikaci lze vést více účtů v různých měnách. Dostupná je i verze v českém jazyce.

Transakce se musí zadávat ručně a nelze jim přidělovat kategorie, pouze názvy. Nabízené přehledy jsou následující: seznam transakcí filtrovatelný podle názvu, zůstatky na účtech, součet příjmů, součet výdajů, bilance, výdaje podle názvu transakce (seznam), srovnání aktuálního měsíce s předchozím co do příjmů/výdajů s daným názvem, kalendář s označením dnů s transakcemi. Aplikace neumožňuje stanovit rozpočet. Frugi stojí 5 Euro.[102, 103]

2.1.31 Expense Center od Starkode

Expense Center je vyvíjeno pro systém iOS. V aplikaci je možno pracovat s více měnami současně. Česká lokalizaci není dostupná.

Transakce je nutno zadávat ručně. Jediným nabízeným přehledem přímo v aplikaci je seznam transakcí podle projektů. Tabulku s přehledem výdajů lze exportovat do PDF. Nelze žádným způsobem nastavit rozpočet. Základní aplikace je zdarma. Pro odemknutí několika pokročilejších funkcí je třeba zaplatit 5 Euro.[104]

2.1.32 Banktivity od IGG Holdings

Banktivity cílí na platformy iOS a Mac. Lze vést více účtů v různých měnách. Jediným podporovaným jazykem je angličtina.

Transakce lze stahovat přímo z finanční instituce, pokud tato podporuje funkci Direct Access. Žádná z českých bank tuto službu nepodporuje. Další možností je import ze souboru s příponou QIF, QMTF, OFX nebo QFX. Transakce jsou na základě iniciálních uživatelských dat automaticky kategorizovány.

Mezi nabízenými přehledy jsou tyto: výdaje/příjmy podle kategorie (koláčový graf), zůstatky na účtech, seznam transakcí, příjmy a výdaje v čase (sloupcový graf), další přehledy týkající se investic, půjček, dluhů a spoření. Aplikace umožňuje nastavit rozpočty podle kategorie a následně sledovat jejich plnění.

Banktivity pro iPhone stojí \$10. Za funkci automatického stahování transakcí z banky je třeba platit dalších \$40 ročně.[105, 106, 107]

2.2 Závěr

Nabízených aplikací tohoto typu je skutečně mnoho. Většina z nich umožňuje vést vícero účtů jako osobní účet, spořicí účet, peněženka a pod. Jen některé ovšem dovolí vést tyto účty v různých měnách a ještě méně z nich nad nimi zobrazuje smysluplné přehledy v jedné zvolené měně. Všechny aplikace, které lze u nás nainstalovat podporovaly českou korunu. Zhruba jedna třetina zkoumaných aplikací je také lokalizována do češtiny.

Jen několik málo aplikací umožňuje stahovat transakce přímo z finanční instituce a většina z nich nepodporuje české banky. Takové jsem našel pouze tři. Jiné aplikace toto řeší čtením SMS zpráv o provedení transakce nebo čtením notifikací od bankovních aplikací, odkud extrahují potřebné informace. Zhruba jedna sedmina zkoumaných aplikací umožňovala import ze souborů exportovaných z banky. Tyto soubory však musí být ve specifických formátech, které opět české banky většinou nepodporují (pouze Fio a Raiffeisenbank umožňují export ve formátu OFX).

Automatickou kategorizaci podporuje opět jen drtivá menšina zkoumaných aplikací. Mechanismus je založený na datech dříve zadaných uživatelem, na lokaci, kde byla transakce zadána, na předvyplněné databázi prodejců a předmětů transakce a dalších blíže nespecifikovaných učících se algoritmech.

Nejčastějšími přehledy byly seznamy transakcí, aktuální zůstatky na účtech a součty příjmů a výdajů za sledované období. Z grafů to byl zejména koláčový graf výdajů podle kategorie. Ve většině aplikací byla možnost nastavit rozpočet pro vybrané kategorie a následně sledovat jeho plnění.

Nejčastější cenovou strategií byl takzvaný „freemium“ model. Omezená verze aplikace je nabízena zdarma, ať už s limitovaným počtem účtů či reportů nebo bez pokročilejších funkcí, jako je synchronizace mezi zařízeními a online napojení na banku. Uživatel má poté možnost zbavit se těchto omezení zaplacením jednorázového nebo opakujícího se poplatku. Nejdražší funkcionalitou byl online import transakcí z finančního ústavu.

Aplikace které v sobě propojují všechny klíčové funkce mého programu, a zároveň podporují české prostředí jsem našel dvě — Wallet a MoneyWiz Premium / MoneyWiz 2. V obou případech jde o velmi rozsáhlé a nákladné produkty. Potenciál své aplikace zde spatřuji v její jednoduchosti, dostupnosti a časově i finančně nenáročném vývoji, díky využití pouze open source technologií a frameworků umožňujících vývoj pro obě hlavní platformy najednou.

2. REŠERŠE EXISTUJÍCÍCH ŘEŠENÍ

Tabulka 2.1: Cílové platformy a lokalizace

	Platformy	Čeština	Více měn
Expense Manager (Bishinews)	Android		x
Monefy	Android		x
Mobills	Android, iOS, Windows Phone, web	x	x
Money Lover	Android, iOS, Windows Phone, web	x	
ClevMoney	Android		
AndroMoney	Android, iOS, web		
Money Manager	Android		
Alzex Personal Finance	Android, iOS, Windows	x	
HandWallet	Android, iOS	x	x
Expense IQ	Android		x
Spendee	Android, iOS	x	x
Wallet	Android, iOS, web	x	x
Fast Budget	Android	x	x
Expense Manager (Markus Hintersteiner)	Android	x	
Expense Manager (Money Tower)	Android		x
Expense Manager (CASorin)	Android		
Wally	Android, iOS		x
MoneyCoach	iOS		x
Pocket Expense	iOS		x
CoinKeeper	Android, iOS, web		x
Spending Tracker	Android, iOS, Windows		x
Cost Track	iOS		
MoneyWiz Premium	Android, iOS, Mac, Windows		x
Money Pro	iOS, Mac	x	x
Finance Lite	iOS	x	x
Expense Manager (Phi Dung)	iOS		x
Live Expenses	iOS		x
Toshl Finance	Android, iOS, Mac, Windows		x
Moneyboard	iOS		x
Frugi	iOS	x	x
Expense Center	iOS		x
Banktivity	iOS, Mac		x

Tabulka 2.2: Způsoby importu transakcí

	Online	Soubor	SMS, notifikace	České banky
Expense Manager (Bishinews)				
Moneyfy				
Mobills			x	
Money Lover	x	x	x	
ClevMoney			x	
AndroMoney				
Money Manager				
Alzex Personal Finance				
HandWallet		x	x	x
Expense IQ		x		
Spendee	x			x
Wallet	x			x
Fast Budget				
Expense Manager (Markus Hintersteiner)				
Expense Manager (Money Tower)				
Expense Manager (CASorin)				
Wally				
MoneyCoach				
Pocket Expense				
CoinKeeper				
Spending Tracker				
Cost Track				
MoneyWiz Premium	x	x		x
Money Pro		x		
Finance Lite				
Expense Manager (Phi Dung)				
Live Expenses				
Toshl Finance				
Moneyboard				
Frugi				
Expense Center				
Banktivity	x		x	

2. REŠERŠE EXISTUJÍCÍCH ŘEŠENÍ

Tabulka 2.3: Funkce

	Automatická kategorizace	Rozpočty	Cena plné verze
Expense Manager (Bishinews)		x	zdarma
Monify		x	70 Kč
Mobills		x	\$25/rok
Money Lover		x	150 Kč
ClevMoney		x	zdarma, 140 Kč
AndroMoney		x	zdarma, 101,53 Kč
Money Manager		x	100 Kč
Alzex Personal Fin.		x	140 Kč
HandWallet		x	\$35
Expense IQ	x	x	290 Kč/rok, 740 Kč
Spendee	x	x	\$3/měsíc
Wallet	x	x	42,50 Kč/měsíc, 65 Kč/měsíc
Fast Budget		x	100 Kč, 340 Kč/rok, 500 Kč/rok
Expense Manager (Markus Hinter.)		x	52 Kč
Expense Manager (Money Tower)		x	zdarma, 44 Kč
Expense Manager (CASorin)			zdarma
Wally	x	x	zdarma
MoneyCoach			zdarma
Pocket Expense		x	5€
CoinKeeper		x	470 Kč/rok
Spending Tracker		x	zdarma, 67 Kč
Cost Track			5€
MoneyWiz Premium	x	x	1800 Kč/rok
Money Pro		x	\$5, \$12/rok
Finance Lite			zdarma
Expense Manager (Phi Dung)			zdarma
Live Expenses			7€
Toshl Finance		x	600 Kč
Moneyboard		x	3€
Frugi			5€
Expense Center			5€
Banktivity			\$10 + \$40/rok

Analýza možností napojení na API bank

V poslední době hýbe světem finančnictví rozvoj takzvaného fintech průmyslu. Jde o využívání moderních technologií pro nabízení nových finančních služeb zákazníkům. Nejvýraznější dopad mají zejména nové firmy, jejichž produkty nabízejí svým uživatelům služby, které jsou tradičně doménou velkých finančních institucí. Tím jsou tyto instituce nuceny k urychlené modernizaci a adaptaci na toto nové prostředí, ve kterém již nejsou jediným hráči.[108, 3, 109]

Tyto nové finančně-technologické produkty často vyžadují napojení na systémy bank pro získání informací o uživatelských účtech a transakční historii, pro provádění online plateb či převodů a podobně. Proto narůstá tlak na finanční ústavy, aby zpřístupnily část svých systémů třetím stranám a to co nejpohodlnějším a samozřejmě co nejbezpečnějším způsobem. Tento tlak je v Evropské unii podpořen i legislativně.[4, 110]

Různé možnosti programového napojení na finanční ústavy tu byly již dříve. A to jak interní, pro potřeby internetového bankovníctví a podobných služeb, tak také externí, např. pro komunikaci s účetními systémy. Rozvoj otevřených API přístupných aplikacím třetích stran v téměř neomezeném měřítku je však něco, co v současnosti nabízí jen velmi málo finančních institucí. Je však zcela zřejmé, že právě tento přístup bude trendem v blízké budoucnosti.[5]

V první části této analýzy krátce zmíním obecné možnosti programového napojení na systémy finančních institucí nebo spíše možnosti získání finančních dat z těchto institucí. Ne vždy se totiž jedná o napojení v pravém slova smyslu. Ve druhé části již detailně rozeberu možnosti přímé komunikace s těmito systémy prostřednictvím otevřených API. Procento bank poskytující tuto službu není vysoké, ale ve světovém měřítku může jít stále o stovky i více finančních ústavů, kde každý nabízí potenciálně jinou formu API. V této části se proto budu věnovat pouze bankám fungujícím na území České republiky a to zejména analýzou konkrétních případů pro banky, které tuto službu nabízejí.

3.1 Možnosti komunikace programu s bankou

3.1.1 Homebanking

Zřejmě nejstarší možností napojení programu na účty vedené u banky je takzvaný homebanking. Při spravování účtu prostřednictvím homebankingu je třeba nainstalovat speciální program schopný komunikovat s bankou. Vytvořená aplikace pak využívá možnosti tohoto programu pro zadávání transakcí, stahování transakční historie, stahování kurzovních lístků a podobně. Tyto speciální programy mohou být používány jen jednou finanční institucí (např. Přímý kanál od KB[111]) nebo to mohou být univerzální programy třetích stran, které jsou vybranými bankami podporované (např. MultiCash[112]). Tento způsob spojení s bankou je využíván zejména účetními systémy.[113, 114, 115]

3.1.2 Přímé napojení na API finančních ústavů

Finanční ústavy jistě využívají interně API již dlouhá léta pro komunikaci se svými aplikacemi pro internetové bankovníctví, homebanking či mobilní bankovníctví. S poskytováním přístupu třetím stranám jsou však zdrženlivé. Pro jeho využívání je většinou třeba mít uzavřenou smlouvu s danou institucí a platit příslušné poplatky. Zejména ve Spojených státech amerických bylo vyvinuto několik standardů pro tuto formu komunikace. Nejrozšířenějším otevřeným standardem je Open Financial Exchange (OFX).[116, 117]

OFX je standard pro systém typu klient — server a cloud API pro výměnu finančních dat mezi finančními ústavami a aplikacemi. První verze byla vyvinuta v roce 1997 firmami Intuit, Microsoft a CheckFree. Standard specifikuje zprávy, které si mezi sebou klient a server zasílají (HTTP POST) a formát vyměňovaných dat (XML).[118] Ve světě tento standard implementovalo a výsledný produkt aktivně provozuje přes sedm tisíc finančních ústavů.[116] U bank v České republice však podporován není. Některé banky pouze umožňují stáhnout transakce v příslušném formátu.[119, 120] — Ověřeno také osobní komunikací se zákaznickou podporou bank.

S bouřlivým rozvojem fintech programů v posledních letech, zmíněném na začátku kapitoly, souvisí i rozvoj bankovních API otevřených pro aplikace třetích stran. Nejvýraznějším počinem v této oblasti je směrnice Evropské unie 2015/2366/EU nazývaná také Payment Services Directive 2 (PSD2) vydaná 25. listopadu 2015 (do 13. 1. 2018 má být zapracována do legislativy jednotlivých členských států). V té se mimo jiné stanoví, že banky musí umožnit zákazníkovi, aby byl schopen autorizovat poskytovatele třetích stran k provádění plateb a k přístupu k jejich informacím přímo z jejich účtů.[121, 4, 5] Podobné nařízení bylo vydáno i ve Velké Británii.[5] Jako nejpravděpodobnější forma poskytování finančních dat se uvádí právě otevřené API.[4] V souvislosti s tím se očekává vznik jednotného standardu, který by umožnil hladkou integraci aplikací se systémy finančních ústavů.[5] Například ve Velké Británii

vznikla Open Banking Working Group za účelem vytvoření právě takového standardu.[122]

Posun k poskytování otevřených API ve finančních institucích po celém světě potvrzuje i nedávný průzkum provedený iniciativami Open Bank Project, Bank Innovation a APIdays. Podle něj 39% z dotázaných bank již spustilo program otevřených API a dalších 39% toto plánuje v následujících dvanácti měsících. Průzkum byl proveden v říjnu 2016.[5]

Nezávisle na PSD2 některé banky již otevřené API poskytovaly dříve. U nás to je např. Fio banka.[123] Jiné začaly s vývojem vlastního API krátce před nebo po vydání směrnice. U nás například Česká spořitelna.[124] Těmto produktům se budu detailně věnovat později.

3.1.3 Externí poskytovatelé bankovních API

Jak vyplývá z předchozí části, bank poskytujících otevřené API zatím není mnoho. Přesto existuje řada aplikací, které jsou schopné bankovní data přes internet získávat nebo dokonce manipulovat s finančními prostředky na účtech a přitom nevyužívají bankovních aplikací (viz. Rešerše konkurenčních řešení). Řešením pro tyto aplikace jsou externí poskytovatelé bankovních API (Kontomatik, Plaid, Yodlee a další).[125]

Tyto firmy získávají potřebná data kombinací přímé spolupráce s finančními ústavami a metody známé jako web scraping.[125] Při web scrapingu program prakticky simuluje živého uživatele. Přihlásí se do internetového bankovníctví, lokalizuje hledaná data a ty následně poskytne klientské aplikaci. Někteří poskytovatelé zároveň obohatí stažená data o další údaje, jako je např. kategorie transakce, normalizované jméno prodejce nebo místo uskutečnění nákupu. Služby těchto poskytovatelů jsou většinou dostupné ve formě REST API.[126, 127]

3.1.4 Další metody

Mimo již zmíněných metod existuje ještě několik dalších způsobů, jak mohou aplikace získat bankovní data. Při nich není navázán přímý kontakt se systémem banky a to ani prostřednictvím externích poskytovatelů. Není třeba zadávat ani přihlašovací údaje k internetovému bankovníctví. Při těchto způsobech se využívá různých zpráv nebo upozornění produkovaných bankovními programy.

Jedním z těchto způsobů je čtení e-mailů s výpisy z účtu. Některé banky poskytují službu pravidelného zasílání takových zpráv nebo zaslání e-mailu hned po provedení transakce.[128] Z takových zpráv je možno programově extrahovat potřebná data a využít je v aplikaci. Tento postup je využíván např. při párování plateb s objednávkami v e-shopech. Obdobný postup lze využít i pro SMS na mobilních telefonech.[27] Velmi podobnou metodou je pak čtení notifikací, které produkují některé bankovní aplikace.[30]

3.2 Rozbor bankovních API v ČR

Důsledky směrnice PSD2 se týkají i naší republiky. Do začátku roku 2018 musí být zapracována i do naší legislativy a tím budou banky přinuceny k poskytnutí svých dat třetím stranám. Některé tuto informaci zatím zdánlivě neberou na vědomí, některé čekají na konkrétní zákon a vyjasnění některých problematických oblastí a jiné proaktivně zahájily vlastní vývoj otevřených API, aby v budoucnu nezaostaly za konkurencí.[129]

V současnosti jsou u nás dvě banky, které otevřené API poskytují. První byla Fio banka, která je nabízí již od roku 2012.[130] Druhou z nich je Česká spořitelna. Situace u ostatních našich bank není zcela přehledná. Na svých webových stránkách většinou o API nic neuvádí a na zákaznické podpoře nemusí mít příslušné informace. V tabulce 3.1 shrnuji výsledek svého výzkumu založeného především na telefonické a emailové komunikaci s bankami a na rozhovorech s jejich představiteli dostupných v médiích. V potaz jsou brána pouze API pro přístup k běžným a karetním účtům, API pro firemní klienty nejsou zahrnuta. Seznam bank působících na našem území je převzat ze stránek České národní banky.[129, 131]

Následuje detailní rozbor otevřených API obou českých bank, které jej poskytují. V rozboru se soustředím zejména na:

- autentizaci a autorizaci aplikace a jejího uživatele
- nabízené zdroje a funkce
- komunikaci mezi aplikací a bankou
- formát předávaných dat
- indikaci chybových stavů

3.2.1 API České spořitelny

Informace jsou převzaty z dokumentace API České spořitelny.[132, 133]

3.2.1.1 Autentizace, autorizace

Pro přístup k API České spořitelny musí uživatel mít tyto údaje:

- WebAPI klíč — identifikuje aplikaci, umožňuje přístup k některým zdrojům, neslouží k autorizaci jako takové, ale pro statistické účely banky (zašle banka)
- client_id — identifikuje vývojáře, ať jde o firmu nebo osobu (zašle banka)

- `client_secret` — slouží k některým operacím souvisejícím s autorizací (zašle banka)
- `login page` — webová adresa, na které se může autentizovat uživatel (zašle banka)
- `redirect_uri` — adresa, na kterou je uživatel přesměrován po úspěšné autentizaci (volí si sám autor aplikace)

Při procesu autorizace se pracuje se třemi typy tokenů:

- `code` — dočasný kód sloužící pouze k výměně za ostatní přístupové tokeny
- `access_token` — slouží k provolávání příslušných API, má omezenou životnost
- `refresh_token` — slouží k získání nového `access_token` po vypršení platnosti předešlého

Pokud aplikace přistupuje ke zdrojům, u kterých je nutná autorizace uživatele, musí nejprve získat příslušný `access_token`. Každé volání API pak musí v hlavičce obsahovat *WebAPI klíč* a tento `access_token`. Proces jeho získání probíhá podle protokolu OAuth[134] následovně: Uživatel je přesměrován na adresu uvedenou v `login page` s hodnotou `client_id` a `redirect_uri` v parametru. Zde se přihlásí svými přihlašovacími údaji k internetovému bankovníctví. Po úspěšném přihlášení je uživatel přesměrován na adresu zadanou v `redirect_uri`. V parametru dotazu je předána hodnota `code`. Tento `code` se následně zašle na adresu pro výměnu za `access_token`. V těle tohoto dotazu se mimo `code` uvádí také `client_id`, `client_secret`, `redirect_uri` a `grant_type` s hodnotou „authorization_code“. V odpovědi formátu JSON přijdou hodnoty `access_token` a `refresh_token`. Ve zkráceném procesu lze namísto `code` získat rovnou `access_token` bez `refresh_token`.

Obnova prošlého `access_token` probíhá následovně: Na stejnou adresu jako při získání nových tokenů se zašle dotaz s parametry `client_id`, `client_secret`, `redirect_uri`, `refresh_token` a `grant_type` s hodnotou „refresh_token“. V odpovědi přijde nový `access_token`.

3.2.1.2 Zdroje a funkce

Česká spořitelna dělí své API do šesti skupin. Jsou to Netbanking, Kurzovní lístky, Transparentní účty, Corporate, Pobočky a bankomaty, Hypocalc. API je typu REST, data jsou předávána ve formátu JSON. Základní forma jeho volání vypadá zhruba takto:

3. ANALÝZA MOŽNOSTÍ NAPOJENÍ NA API BANK

`www.csas.cz/api/<verze API>/<API skupina>/cz/my/
<kategorie> (např. accounts)/<id>/<podkategorie> (např.
transactions)?<parametry specifikující stránkování,
řazení, podmnožinu výsledku atp.>`

Pokud je v odpovědi seznam (seznam účtů, transakcí atp.) umožňující stránkování, obsahuje odpověď mimo údajů popsaných dále také číslo stránky, počet stránek, velikost stránky a následující stránku.

Seznam nabízených funkcí je následující:

- Netbanking (vše se vztahuje k přihlášenému uživateli — použitému *access_token*)
 - Informace o profilu — jméno, příjmení, pozdrav, customer id, institute id, pohlaví, datum posledního přihlášení, zda uživatel od-souhlasil §107 telecommunication act
 - Seznam účtů (běžné účty, spořicí účty, půjčky) — id, čísla účtu, popis (v současnosti jméno majitele účtu), zůstatek, měna, alias, kód produktu, podúčty, úspory, nastavení účtu (např. zda umožňuje zahraniční platby), půjčka a její specifikace a pod.
 - Detail jednoho účtu (viz. výše)
 - Zjednodušený seznam účtů — obsahuje seznam čísel účtů
 - Zůstatek na jednom účtu — zůstatek, disponibilní zůstatek, pře-čerpání
 - Historie transakcí na jednom účtu za vybranou dobu — informace o protistraně, částka, datum, měna, zpráva pro příjemce/plátce, popis, symboly a pod.
 - Změna poznámky u transakce — umožňuje také označit ji jako oblíbenou (Je to jediná funkce umožňující změnu údajů.)
 - Seznam karet — číslo, validita, majitel, značka, typ, limit, zůstatek, měna atp.
 - Detail jedné karty (viz. výše)
 - Seznam stavebních spoření — podobné jako u účtů
 - Detail jednoho stavebního spoření — podobné jako u účtů
 - Seznam penzijních připojištění — majitel, validita, číslo smlouvy, částka, strategie, starobní/invalidní atp.
 - Detail penzijních připojištění (viz. výše)
 - Seznam pojištění — podobné penzijnímu připojištění
 - Detail pojištění — podobné penzijnímu připojištění
 - Seznam účtů s cennými papíry — id, číslo, alias, popis, zůstatek, cenné papíry — id, název, ISIN, tržní hodnota, poslední cena, typ atp.

- Detail účtu s cennými papíry (viz. výše)
- Kurzovní lístky
 - Seznam aktuálních kurzů pro 21 měn — kód měny, datum validity, název měny, země, na jaké množství se kurz vztahuje, kurzy pro prodej / pro nákup atp.
 - Seznam kurzů měn ke specifickému datu (viz. výše)
- Transparentní účty ČS
 - Seznam transparentních účtů České spořitelny — zjednodušené oproti účtům v Netbanking
 - Detail
 - Historie transakcí na vybraném účtu — odpovídá historii transakcí v Netbanking
- Corporate
 - Seznam firem asociovaných s přihlášeným uživatelem, detail firmy, marketingové kampaně, velikosti firem (kategorie podle počtu zaměstnanců), různá bankovní data o firmě, informace o bankovních poradcích atp.
- Pobočky a bankomaty
 - Detail pobočky — adresa, služby, otevírací doba, email atp.
 - Detail bankomatu — adresa, služby, otevírací doba atp.
 - Seznam poboček a bankomatů — je umožněno vyhledávat podle různých parametrů — stát, město, souřadnice, typ (pobočka/bankomat), specifické příznaky (např. pro slepce)
 - Seznam poboček a bankomatů v obdélníku — vyhledávání v obdélníku specifikovaném geografickými souřadnicemi pro dva protilehlé rohy
 - Seznam poboček (viz. Detail pobočky)
 - Poslední/nadcházející výpadek pobočky — od kdy, do kdy, popis atp.
 - Seznam poboček v obdélníku (viz. Seznam poboček a bankomatů v obdélníku)
 - Seznam bankomatů (viz. Detail bankomatu)
 - Seznam bankomatů v obdélníku (viz. Seznam poboček a bankomatů v obdélníku)
- HypoCalc

3. ANALÝZA MOŽNOSTÍ NAPOJENÍ NA API BANK

- Základní kalkulace hypotéky podle zadaných parametrů – cena, půjčka, fixace, datum splatnosti

3.2.1.3 Příklad volání

Pro ilustraci jsem zvolil volání API pro získání zůstatku na účtu s ID D2CE52.

Příslušné URI vypadá následovně:

```
https://www.csas.cz/api/v3/netbanking/my/accounts/D2CE52/  
balance
```

Odpověď serveru:

```
{  
  "balance": {  
    "value": 45865,  
    "currency": "CZK",  
    "precision": 2  
  },  
  "disposable": {  
    "value": 45865,  
    "precision": 2,  
    "currency": "CZK"  
  },  
  "overdraft": {  
    "value": 20000,  
    "precision": 2,  
    "currency": "CZK"  
  }  
}
```

3.2.1.4 Indikace chyb

Pro sdělování chyb volajícímu je využíváno standardních HTTP návratových kódů. Pokud je volání bez chyby, je vrácen návratový kód 200 nebo 204. Pro chyby při parsování zprávy, autorizaci nebo hledání požadovaného zdroje jsou vyhrazeny kódy 400, 403 respektive 404. V těle zprávy je navíc zaslán chybový objekt blíže specifikující, o jakou chybu se jedná. Další kódy jsou 500 pro interní serverovou chybu a 415 pro nepodporovaný typ média (Accept/Accept-Language unsupported).

Ukázka chybového objektu:

```
"errors": [  
  {  
    // error code  
    "error": "SOME_ERROR_CODE",  
  },  
  {  
    // error code
```

```

        "error": "ANOTHER_ERROR_CODE",
        // optional attribute indicating JSON parameter
        // name that caused the error.
        "scope": "account.amount.currency"
    },
    {
        // error code
        "error": "OTHER_ERROR_CODE",
        // list of named parameters for current message
        "parameters": {
            "AMOUNT_ENTERED": 10000,
            "CURRENCY": "EUR",
            "LIMIT": 500,
        }
        // scope can use array index
        // for identification of particular field
        "scope": "orders [3].amount.value"
    }
],
// call identification for tracing purposes
"cz-transactionId": "160216825156501"

```

Příklady chybových kódů:

- `TOKEN_EXPIRED` — vypršela platnost *access_token*
- `TOKEN_NOT_ALLOWED` — poskytnutý *access_token* nedovoluje toto volání
- `RESOURCE_NOSORT` — zdroj nepodporuje řazení

3.2.2 API Fio banky

Informace jsou převzaty z dokumentace API Fio banky.[135]

3.2.2.1 Autentizace, autorizace

Proces získání autorizačního tokenu je zde zcela jiný než v předchozím případě. Na první pohled je jednodušší, ale je k němu potřeba větší účast uživatele. Ten se musí nejprve přihlásit do svého internetového bankovníctví. Zde si v sekci „Nastavení“ nechá token automaticky vygenerovat. K dispozici jsou dva typy. Jeden slouží pouze ke sledování účtu (stahování transakcí). Druhý slouží jak pro sledování účtu, tak pro zadávání platebních a inkasních příkazů. Token lze vždy použít k přístupu pouze k jednomu účtu. Pokud jich má uživatel více, musí si pro každý nechat vygenerovat jiný. Tento token pak musí být předán do aplikace, která s ním dále pracuje.

3.2.2.2 Zdroje a funkce

Fio banka poskytuje API pro export pohybů na účtu a výpisů z banky a pro import platebních příkazů. API je typu REST (GET pro získávání dat, POST pro zadávání platebních příkazů). Data lze získávat v mnoha formátech. Pro pohyby na účtu jsou to: CSV, GPC, HTML, JSON, OFX, XML (Fio). Pro výpisy z účtů jsou to: CSV, GPC, HTML, JSON, OFX, XML (Fio), PDF, MT940, ČBA XML (CAMT.053), SBA XML (CAMT.053). Data pro import platebních příkazů pak mohou být ve formátech ABO, XML (Fio) nebo SEPA (plain).

Struktura volání pro získání dat je následující:

```
www.fio.cz/ib_api/rest/<způsob_vybírání_transakcí>  
(např. by-id)/<token>/<parametry_oddělené_lomítkem>/  
transactions.<formát>
```

Volání pro import je pak následující:

```
www.fio.cz/ib_api/rest/import/
```

Při importu se autorizační token zasílá v HTTP hlavičce v parametru „token“. Samotný soubor s daty k importu je taktéž specifikován v hlavičce v parametru „file“, jeho formát v parametru „type“ a jazyk odpovědi serveru v parametru „lng“ (cz, sk nebo en).

Seznam nabízených funkcí je následující:

- Export pohybů a výpisů z banky
 - Stažení pohybů na účtu za dané období
 - Stažení výpisů z účtu – podle roku a čísla výpisu
 - Stažení pohybů na účtu od posledního stažení – při úspěšném stažení pohybů na účtu se nastaví zarážka (id pohybu) na poslední stažený pohyb
 - Nastavení zarážky na id pohybu
 - Nastavení zarážky na datum
- Import platebních příkazů do bank
 - Platba do ČR
 - Europlatba
 - Zahraniční platba
 - Inkaso a SEPA inkaso

Pro všechna stažení pohybů a výpisů je stejná odpověď. Ta obsahuje různé údaje v závislosti na zvoleném formátu. Soubory ve formátech GPC, OFX, MT940, ČBA XML (CAMT.053), SBA XML (CAMT.053) odpovídají svým

příslušným definicím. Soubory ve formátech HTML, JSON a XML (Fio) obsahují tyto údaje o účtu: ID účtu, měna, IBAN, BIC, zůstatek na účtu na začátku a na konci vybraného období, ID prvního a posledního pohybu v daném výběru, počátek a konec zvoleného období, rok a ID výpisu, ID posledního úspěšně staženého pohybu a tyto údaje o transakcích: ID pohybu, datum, částka, měna, informace o protiúčtu (číslo, název, kód banky, název banky), symboly, uživatelská identifikace pohybu, zpráva pro příjemce atp.

Import platebních příkazů musí být dodatečně autorizován uživatelem (SMS, fio podpis). Soubory ve formátech ABO a SEPA odpovídají příslušným definicím. Soubory ve formátech XML (Fio) musí obsahovat tyto údaje: číslo účtu příkazce, měna, částka, číslo účtu příjemce/inkasovaného, kód banky příjemce, datum a volitelně symboly, zpráva pro příjemce, vlastní označení, platební titul a typ platby (standardní, zrychlená, prioritní, příkaz k inkasu). V případě zahraniční platby zde musí být ještě BIC, jméno a adresa příjemce, poplatky (vše plátce, vše příjemce, každý sám své). Odpověď serveru je vždy ve formátu XML a obsahuje tyto údaje: chybový kód, ID dávky, status, suma debetních položek v dávce, suma kreditních položek v dávce.

3.2.3 Příklad volání

Pro ilustraci exportu jsem zvolil volání API pro získání transakcí za období od 1. 7. 2012 do 31. 7. 2012 ve formátu XML (Fio).

Příslušné URI vypadá následovně (autorizační token je aBC123):

```
https://www.fio.cz/ib_api/rest/periods/aBC123/2012-08-25/2012-08-31/transactions.xml
```

Odpověď serveru:

```
<AccountStatement>
  <Info>
    <accountId>2111111111</accountId>
    <bankId>2010</bankId>
    <currency>CZK</currency>
    <iban>CZ7920100000002111111111</iban>
    <bic>FIOBCZPPXXX</bic>
    <openingBalance>7356.22</openingBalance>
    <closingBalance>7321.22</closingBalance>
    <dateStart>2012-07-01+02:00</dateStart>
    <dateEnd>2012-07-31+02:00</dateEnd>
    <idFrom>1147608196</idFrom>
    <idTo>1147608197</idTo>
  </Info>
  <TransactionList>
    <Transaction>
      <column_22 id="22" name="ID pohybu">
        1147608196
      </column_22>
    </Transaction>
  </TransactionList>
</AccountStatement>
```

3. ANALÝZA MOŽNOSTÍ NAPOJENÍ NA API BANK

```
<column_0 id="0" name="Datum">
  2012-07-27+02:00
</column_0>
<column_1 id="1" name="Objem">
  -15.00
</column_1>
<column_14 id="14" name="Měna">
  CZK
</column_14>
<column_2 id="2" name="Protiúčet">
  222233333
</column_2>
<column_3 id="3" name="Kód banky">
  2010
</column_3>
<column_12 id="12" name="Název banky">
  Fio banka, a.s.
</column_12>
<column_7 id="7" name="Uživatelská identifikace">
</column_7>
<column_8 id="8" name="Typ">
  Platba převodem uvnitř banky
</column_8>
<column_9 id="9" name="Provedl">
  Novák, Jan
</column_9>
<column_25 id="25" name="Komentář">
  Můj test
</column_25>
<column_17 id="17" name="ID pokynu">
  2102392862
</column_17>
</Transaction>
</TransactionList>
</AccountStatement>
```

URI pro import vypadá následovně:

```
https://www.fio.cz/ib_api/rest/import/
```

Soubor s importovanými daty ve formátu XML (Fio):

```
<?xml version="1.0" encoding="UTF-8"?>
<Import xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:noNamespaceSchemaLocation=
"http://www.fio.cz/schema/importIB.xsd">
<Orders>
<DomesticTransaction>
  <accountFrom>1234562</accountFrom>
  <currency>CZK</currency>
  <amount>100.00</amount>
```

```

<accountTo>2212-2000000699</accountTo>
<bankCode>0300</bankCode>
<ks>0558</ks>
<vs>1234567890</vs>
<ss>1234567890</ss>
<date>2013-04-25</date>
<messageForRecipient>
  Hračky pro děti
</messageForRecipient>
<comment></comment>
<paymentType>431001</paymentType>
</DomesticTransaction>
</Orders>
</Import>

```

3.2.3.1 Indikace chyb

V dokumentaci je vyjmenováno pět známých chybových stavů, které mohou být vráceny.

- The server encountered an internal error () that prevented it from fulfilling this request. — Soubor s daty pro import je odeslán jako tělo POST zprávy a nikoli jako příloha.
- Status Code:404 Not Found — Špatně zaslaný dotaz na který server nemůže řádně odpovědět.
- Status Code:409 Conflict — Není dodržen minimální interval 30 sekund mezi stažením dat z banky / uploadem dat do banky u konkrétního tokenu. Konkrétní token lze použít pouze 1x pro čtení nebo zápis během 30 sekund
- Status Code:500 Internal Server Error — Chyba indikuje neexistující nebo neaktivní token.
- SSL certificate problem: unable to get local issuer certificate — Kontrola certifikátu certifikační autority selhala. Je nutné získat nový používaný certifikát.

Po importu dat do banky je navíc vrácen XML objekt specifikující status importu a případný chybový kód. Statusy mohou být čtyři:

- ok — příkaz přijat
- warning — varování, některý z údajů je nesprávně vyplněn (např. datum)
- error — chyba v příkazu

- fatal — chyba na straně bankovního systému banky

Možné chybové kódy vrácené v případě chyby jsou tyto:

- 0 — příkaz byl přijat
- 1 — nalezené chyby při kontrole příkazů
- 2 — varování kontrol — chybně vyplněné hodnoty
- 11 — syntaktická chyba
- 12 — prázdný import, v souboru nejsou žádné příkazy
- 13 — příliš dlouhý soubor, soubor je delší než 2 MB
- 14 — prázdný soubor, soubor neobsahuje příkazy

3.3 Závěr

Různé možnosti programové komunikace s finančními ústavy tu jsou již dlouho, ale skutečně otevřené API poskytující, po schválení od vlastníka, téměř neomezený přístup k účtům je věc velmi aktuální. Díky rozmachu fitech průmyslu, tlaku na digitální inovace a podpoře ze strany EU však lze očekávat velmi rychlý nástup a převládnutí právě této formy programové komunikace s bankou.

U nás tuto službu nabízejí zatím pouze dvě finanční instituce, Česká spořitelna a Fio banka. Jejich přístup je při tom značně odlišný. Fio banka se soustředí především na navazující stahování informací o pohybech na účtu v mnoha formátech, jak standardních, tak vlastních, a platby. Zatímco Česká spořitelna vytváří rozsáhlé, víceúčelové API, schopné vyhovět nejrůznějším požadavkům vývojářů od stahování informací o stavu na účtu po vyhledávání bankomatů podle aktuální pozice. Zatím je však zcela bez podpory zadávání nových transakcí.

U ostatních bank působících na našem území je pravděpodobné, že se také připravují na příchod nového zákona. Získat bližší informace o aktuálním stavu vývoje/plánování je však téměř nemožné. Představitelé těchto institucí na přímé otázky odpovídají velmi neurčitě a jejich zástupci na pozicích zákaznické podpory často ani nevědí, že se něco takového blíží. Nabízenou tabulku 3.1 je tak třeba brát s rezervou spíše jako indikaci toho, kde je vývoj na tolik daleko, že jsou řídicí pracovníci ochotni předat tuto informaci směrem k zákazníkům.

Tabulka 3.1: Stav vývoje otevřeného API v bankách působících na území České republiky

	Stav API
Moneta	plánuje se na rok 2018
ČSOB	zkoumá možnosti
Modrá pyramida	zatím se neplánuje
Komerční banka	plánuje se
UniCredit bank	zatím se neplánuje
Raiffeisenbank	zkoumá možnosti
J&T banka	plánuje se
PPF bank	zatím se neplánuje
Air bank	zkoumá možnosti
Banka Creditas	zatím se neplánuje
Česká exportní banka	?
Equa bank	zkoumá možnosti
Expobank CZ	plánuje se
Sberbank CZ	zatím se neplánuje
Wüstenrot hypoteční banka / stavební spořitelna	zatím se neplánuje
Českomoravská stavební spořitelna	plánuje se
Cetelem	zatím se neplánuje
mBank	zatím se neplánuje
Oberbank	zatím se neplánuje
PRIVAT BANK	?
Raiffeisenbank im Stiftland eG	zatím se neplánuje
Waldviertler Sparkasse Bank	plánuje se
ZUNO	zatím se neplánuje

Analýza a návrh

4.1 Analýza požadavků

V této části se věnuji specifikaci požadavků kladených na zadanou aplikaci. Zdrojem požadavků bylo samotné zadání diplomové práce a předcházející i následná komunikace se zadavatelem / vedoucím práce. Funkční požadavky se týkají zejména způsobu zadávání transakcí, jejich automatické kategorizace a zobrazovaných přehledů. Obecné požadavky pak zahrnují hlavně použité technologie, cílové platformy a bezpečnost.

Funkční požadavky jsou ohodnoceny prioritou podle metody MoSCoW.[136] (M – Must have, S – Should have, C – Could have)

4.1.1 Funkční požadavky

1. Účty

- **M** V aplikaci bude možno vést dva typy účtů: bankovní účet a hotovost.
- **M** Možnost přidat, odebrat, upravit účet
- **S** Účty je možno vést v různých měnách (min. CZK, EUR)

2. **M** Aplikace bude pracovat se třemi typy transakcí – příjem, výdej, převod.

3. Zadání transakcí

- **M** Aplikace bude importovat údaje o transakcích z banky.
 - Manuální import souborů exportovaných z internetového bankovníctví
 - Nebo automatický import transakcí přes web API banky
- **M** Do aplikace bude možno přidávat manuálně hotovostní transakce.

4. ANALÝZA A NÁVRH

- **C** Do aplikace bude možno přidávat hotovostní transakce vyfočením paragonu.
4. Transakce se budou dělit do kategorií.
 - **M** Transakci bude automaticky přiřazena kategorie.
 - **M** Transakci bude manuálně přiřazena kategorie.
 5. Zobrazení přehledů o účtech a transakcích
 - **M** Koláčový graf s výdaji podle kategorií za zvolené období
 - **S** Možnost zobrazit i jako seznam (i s příjmy)
 - **M** Součet všech výdajů a příjmů za zvolené období
 - **M** Aktuální zůstatek na účtech
 - **M** Rozložení výdajů v čase za zvolené období — sloupcový graf
 - **M** Možnost zvolit jednu kategorii
 - **M** Seznam všech transakcí za zvolené období
 - **M** Možnost zvolit jednu kategorii
 6. Kontrola výdajů
 - **M** Nastavení měsíčního rozpočtu pro vybranou kategorii
 - **M** Při přiblížení se a překročení stanovené částky bude uživatel upozorněn.
 - **M** Zobrazení seznamu rozpočtů pro různé kategorie

4.1.2 Obecné požadavky

1. Mobilní aplikace
 - iOS
 - Android
2. Technologie
 - React Native
 - NodeJS
3. Zveřejnění zdrojových kódů na GitHub
4. Transakce nebudou přímo ukládány na server.
5. Čísla účtů nebudou přímo ukládána na server.

4.2 Případy užití

V nadcházející kapitole ukazují výčet případů užití pokrývající množinu funkčních požadavků. Každý případ je doplněn bližší specifikací či krátkým scénářem, popisujícím kroky uživatele při plnění případu. V modelu se počítá s jedinou uživatelskou rolí pro všechny případy užití.

UC1: Zobrazit přehled výdajů podle kategorie — koláčový graf

- Hodnoty v grafu jsou součty výdajů v jednotlivých kategoriích za zvolené období.
- Zobrazí se také součet všech výdajů a příjmů ve zvoleném období.
- Uživatel může změnit sledované období. (UC8)

UC2: Zobrazit přehled výdajů v čase — sloupcový graf

- Hodnoty v grafu jsou součty celkových výdajů nebo výdajů ve zvolené kategorii rozdělené do kratších časových úseků v rámci zvoleného období.
- Zobrazí se také součet všech výdajů a příjmů ve zvoleném období.
- Uživatel může změnit sledované období (UC8)

UC3: Zobrazit seznam transakcí

- Zobrazí se kategorie, datum, částka a případně i poznámka pro všechny příjmy a výdaje ve zvoleném období.
- Zobrazí se také součet všech výdajů a příjmů ve zvoleném období.
- Uživatel může změnit sledované období (UC8)

UC4: Zobrazit seznam účtů

- Zobrazí se název a aktuální zůstatek v měně účtu pro všechny uložené účty.
- Zobrazí se také součet všech zůstatků v korunách.

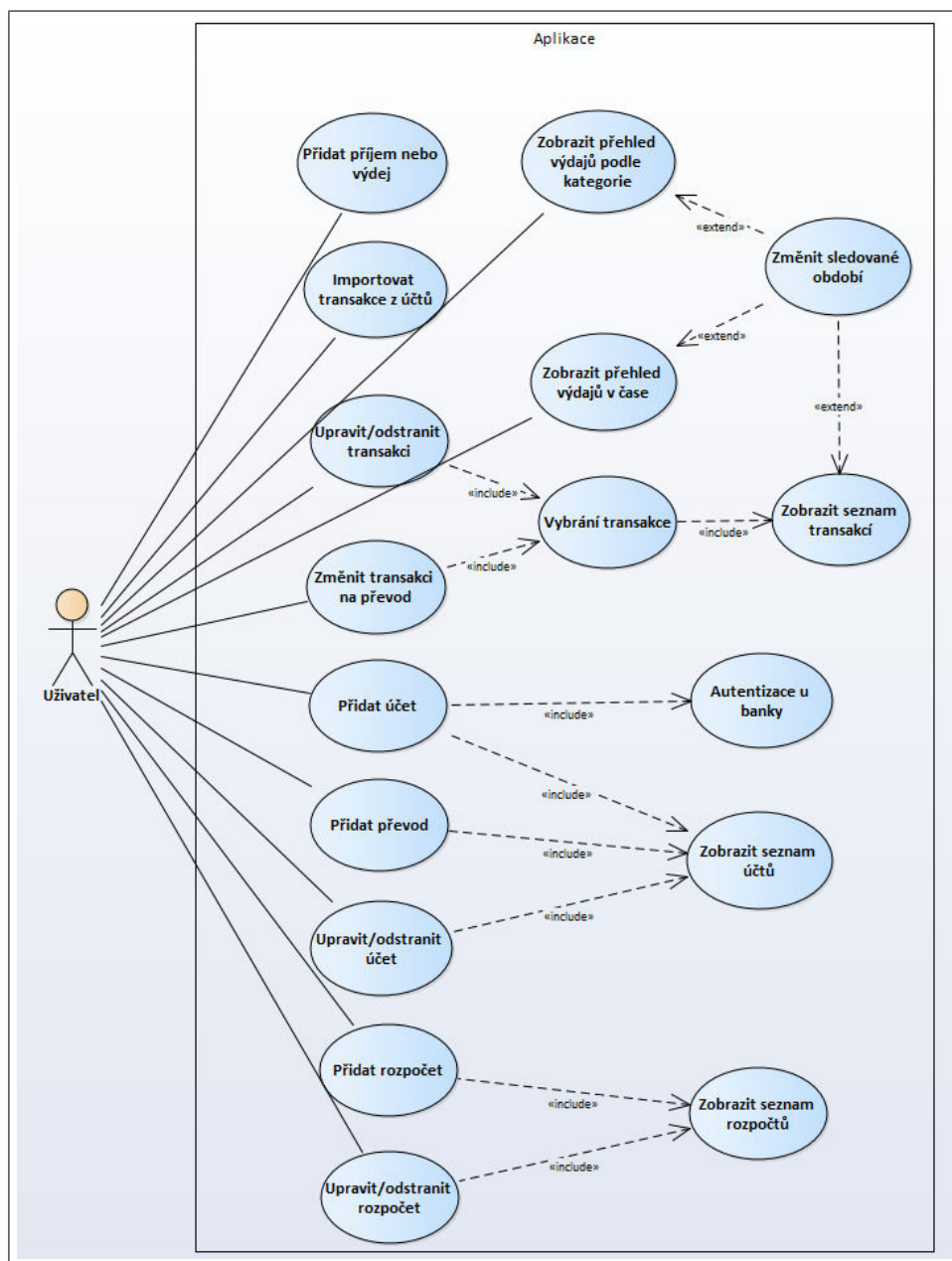
UC5: Zobrazit seznam rozpočtů

- Zobrazí se výše stanoveného rozpočtu pro jednotlivé kategorie.

UC6: Přidat příjem nebo výdej

- Podmínka: Uživatel se nachází na jedné z přehledových obrazovek. (UC1, UC2, UC3)

4. ANALÝZA A NÁVRH



Obrázek 4.1: Model případů užití

- Zde zvolí, zda přidat výdaj nebo příjem.
- Ve formuláři vyplní účet, datum a částku transakce.
- Z nabídky zvolí kategorii transakce.

UC7: Importovat transakce z účtů

- Podmínka: Uživatel se nachází na jedné z přehledových obrazovek. (UC1, UC2, UC3)
- Zde zvolí možnost importování transakcí.

UC8: Změnit sledované období

- Uživatel zvolí, zda se posunout o jedno sledované období vpřed či zpět.

UC9: Autentizace u banky

- Uživatel zvolí možnost přihlásit se.
- Uživatel po přesměrování na autentizační portál banky vyplní své přihlašovací údaje.

UC10: Přidat účet

- Hlavní scénář: Účet u podporované banky
 1. Uživatel si zobrazí seznam účtů. (UC4)
 2. Zde zvolí možnost přidat účet.
 3. Ve formuláři zvolí název a typ účtu.
 4. Dále vyplní číslo účtu.
 5. Autentizuje se u banky. (UC9)
- Alternativní scénář: Účet u nepodporované banky
 1. Uživatel pokračuje od kroku 4 hlavního scénáře.
 2. Pokud byl zvolen bankovní účet u nepodporované banky, uživatel vyplní kromě čísla účtu také měnu a zůstatek na účtu.
- Alternativní scénář: Hotovostní účet
 1. Uživatel pokračuje od kroku 3 hlavního scénáře.
 2. Pokud byl zvolen hotovostní účet, uživatel vyplní měnu a zůstatek na účtu.

UC11: Přidat rozpočet

- Uživatel si zobrazí seznam rozpočtů. (UC5)
- Zde zvolí možnost přidat rozpočet.

4. ANALÝZA A NÁVRH

- Ve formuláři zvolí kategorii, částku a procento z této částky, při kterém má být upozorněn.

UC12: Přidat převod

- Uživatel si zobrazí seznam účtů. (UC4)
- Zde zvolí možnost přidat převod.
- Ve formuláři zvolí účty, částku (dvě částky v případě, že jsou účty v různých měnách) a datum převodu.

UC13: Upravit/odstranit účet

- Uživatel si zobrazí seznam účtů. (UC4)
- Zde vybere účet.
- Ve formuláři upraví zadané informace či odstraní účet.

UC14: Vybrání transakce

- Uživatel si zobrazí seznam transakcí. (UC3)
- Zde vybere transakci.

UC15: Upravit/odstranit transakci

- Uživatel vybere transakci. (UC14)
- Ve formuláři upraví zadané informace, či odstraní transakci.

UC16: Změnit transakci na převod

- Uživatel vybere transakci. (UC14)
- Zvolí možnost změnit transakci na převod.
- Přidá převod (UC12) s předvyplněnými údaji — jeden účet, částka, datum.

UC17: Upravit/odstranit rozpočet

- Uživatel si zobrazí seznam rozpočtů. (UC5)
- Zde vybere rozpočet pro zvolenou kategorii.
- Ve formuláři upraví zadané hodnoty či jej odstraní.

4.3 Architektura a design

V této části předkládám návrhy a popisy vyvíjené aplikace. Jednotlivé modely vždy popisují aplikaci z jiného pohledu či jinou její část. Vybíral jsem takové modely, které jsem shledal relevantní vzhledem ke struktuře a rozsahu programu. Cílem je poskytnout ucelený přehled o aplikaci, konceptech využívaných pro její tvorbu a provázanosti s okolními systémy.

Mezi hlavní faktory ovlivňující architekturu systému určitě patří cílové platformy — mobilní zařízení se systémem Android, iOS. Klíčová je dále nutnost komunikace s externími systémy prostřednictvím síťových rozhraní, zadaná technologie (React Native) a způsob kategorizace transakcí s využitím databáze společné pro všechny uživatele. Omezující by mohl být menší úložný prostor na mobilních zařízeních, nicméně ukládaná data jsou tak malá, že toto nemuselo být řešeno. Největší výzvou bylo pochopení a přizpůsobení se programovací filosofii React Native a Expo, což pro mě byly zcela nové technologie.

4.3.1 Návrh uživatelského rozhraní

Jednoduchý návrh uživatelského rozhraní je v případě uživatelských aplikací dobré udělat hned v prvních fázích projektu. Je velmi užitečný pro komunikaci se zadavatelem a upřesnění požadavků na vytvářenou aplikaci.[137]

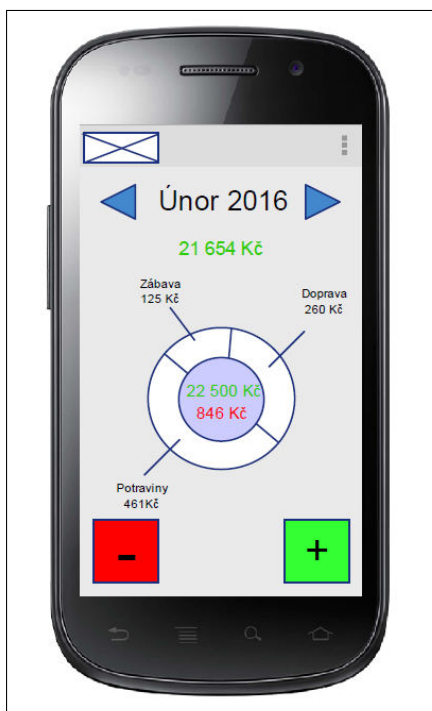
Následující návrh se skládá z takzvaných wireframes. Jde o zjednodušené návrhy jednotlivých obrazovek výsledné aplikace. Jejich účelem není zobrazit výsledný vzhled aplikace, ale spíše rozložení funkčních komponent tak, aby splňovaly požadavky zadavatele. Návrh byl vytvořen pro potřeby počáteční komunikace se zadavatelem a od výsledné aplikace se na několika místech odlišuje. (Vystaveny jsou zde jen návrhy, které nakonec byly použité pro vývoj. Zbylé rozšiřující návrhy včetně verze pro webovou aplikaci jsou poskytnuty k nahlédnutí na přiloženém DVD.)

4.3.2 Navigace

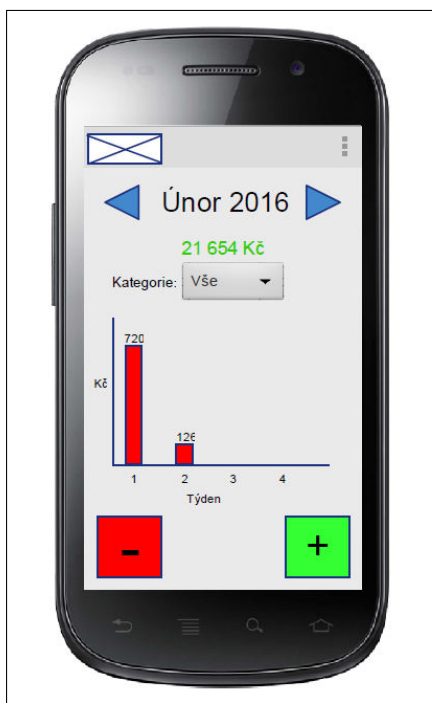
Důležitým prvkem ovlivňujícím výstavbu mobilní aplikace je způsob navigace mezi jednotlivými obrazovkami. Při jejím návrhu jsem se soustředil zejména na snadnou a rychlou dostupnost obrazovek, na které bude uživatel pravděpodobně přistupovat nejčastěji a intuitivní návaznost přechodů při vykonávání operací jako je přidání nové transakce či účtu.

Za hlavní navigační element jsem zvolil takzvaný „navigation drawer“. Jde o menu, které lze „vytáhnout“ ze strany obrazovky. Toto menu je dostupné ze všech obrazovek aplikace a umožňuje okamžitě přejít na jednu z pěti výchozích obrazovek - Výdaje, Výdaje v čase, Transakce, Účty a Rozpočet. Na diagramech není pro přehlednost tato navigace znázorněna.

4. ANALÝZA A NÁVRH



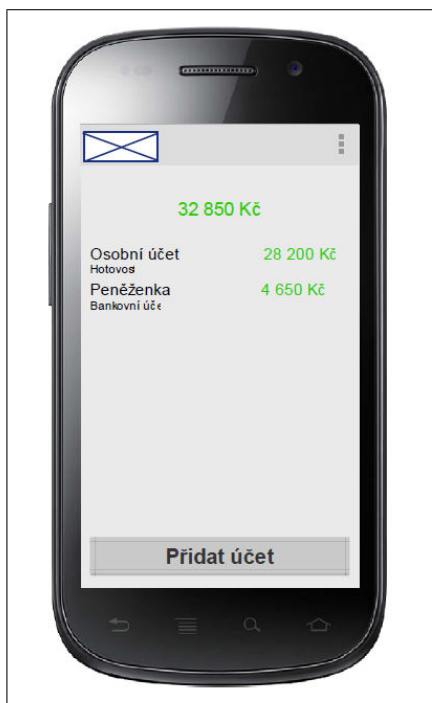
Obrázek 4.2: Výdaje podle kategorie



Obrázek 4.3: Výdaje v čase

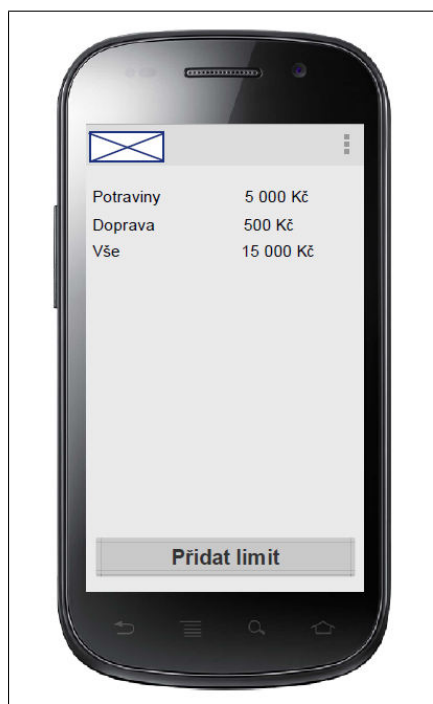


Obrázek 4.4: Transakce

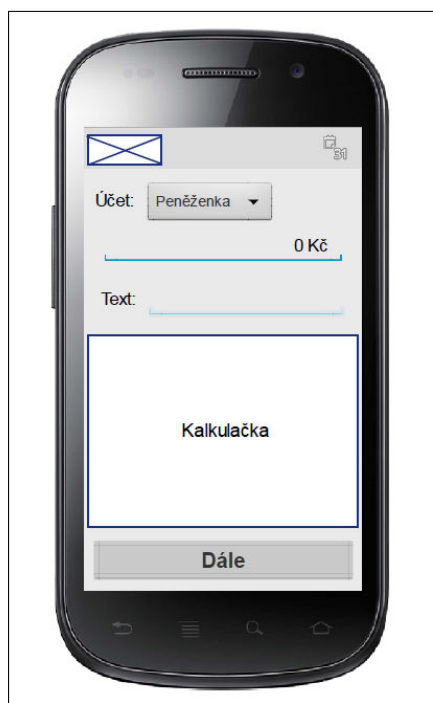


Obrázek 4.5: Účty

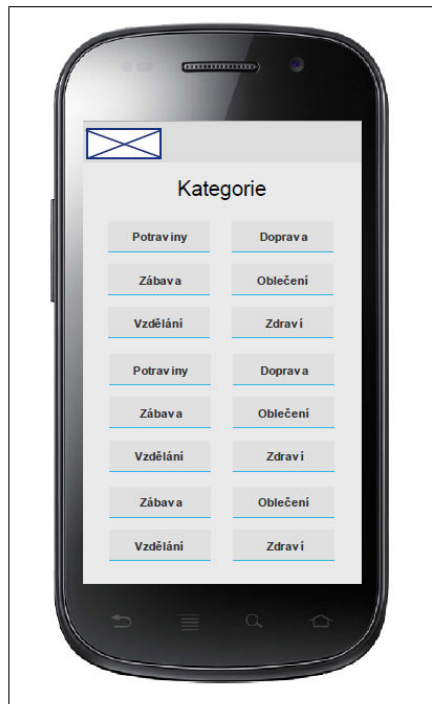
4. ANALÝZA A NÁVRH



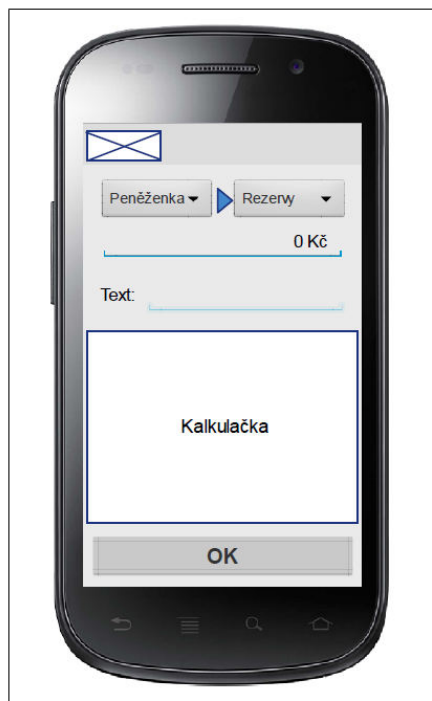
Obrázek 4.6: Rozpočty



Obrázek 4.7: Nová transakce

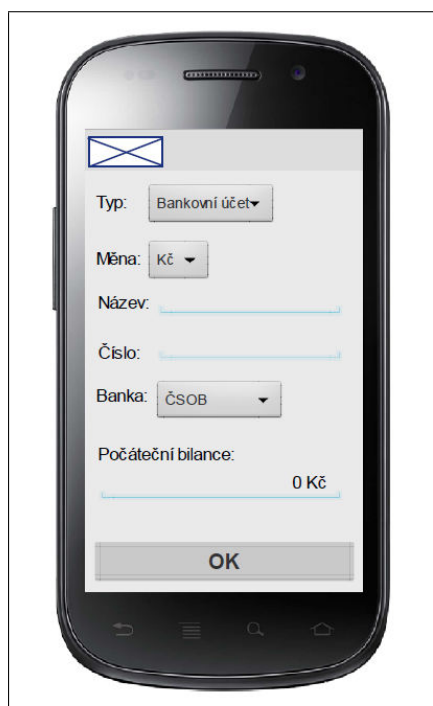


Obrázek 4.8: Výběr kategorie

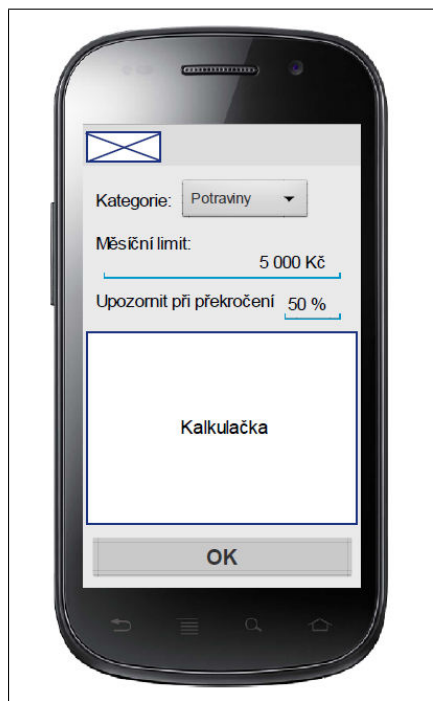


Obrázek 4.9: Nový převod

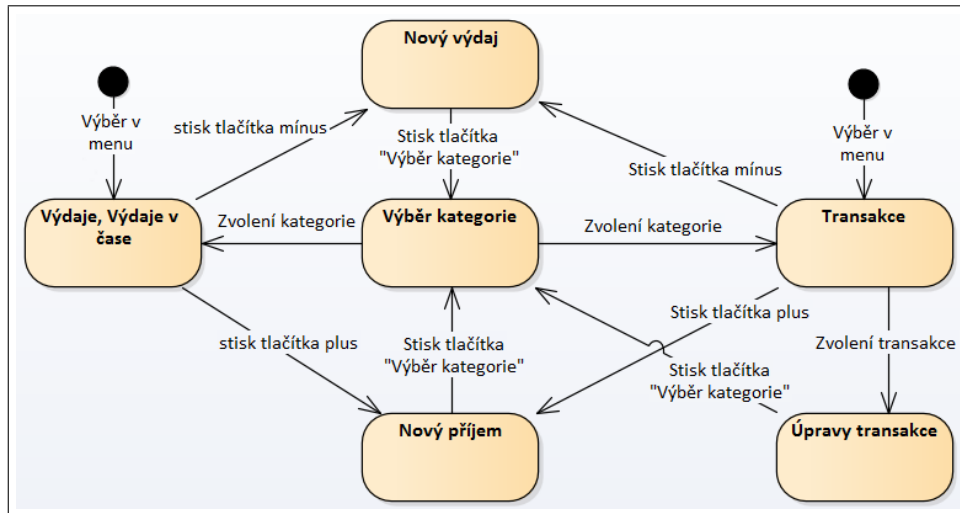
4. ANALÝZA A NÁVRH



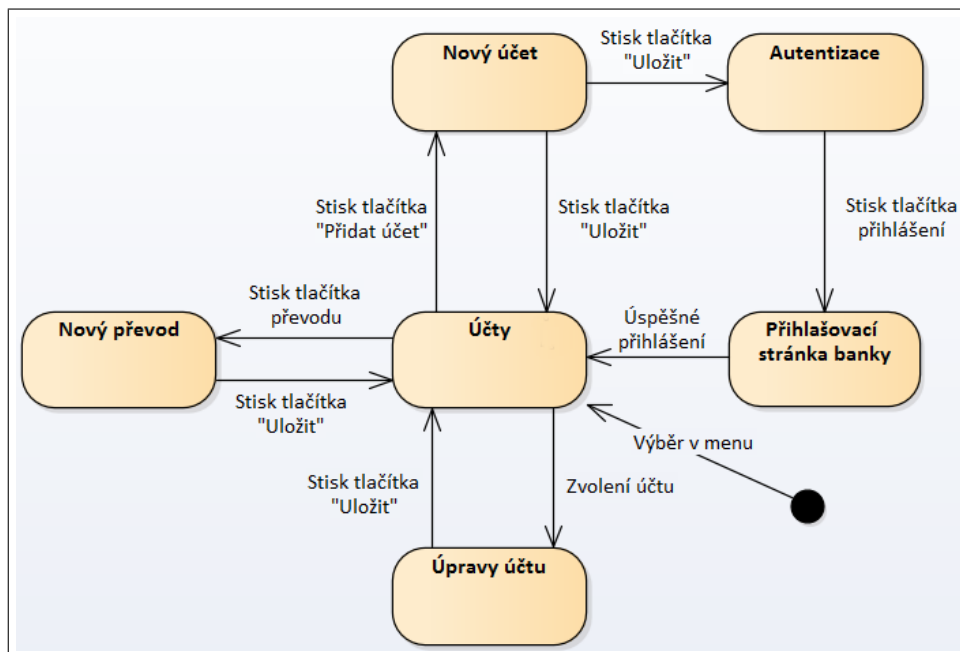
Obrázek 4.10: Nový účet



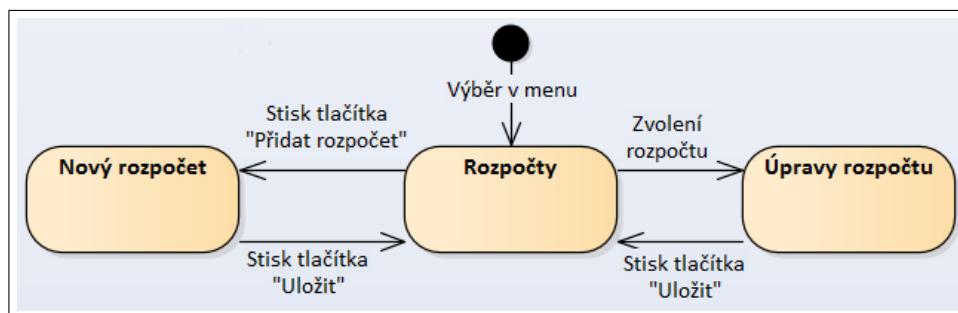
Obrázek 4.11: Nový rozpočet



Obrázek 4.12: Navigace — přehledy výdajů



Obrázek 4.13: Navigace — účty



Obrázek 4.14: Navigace — rozpočty

Ostatní navigace probíhá po kliknutí na příslušná tlačítka. Navazující řetězec přechodů je vždy zakončen návratem na jednu ze zmíněných pěti výchozích obrazovek. Snaha byla docílit co nejmenšího počtu nutných kliknutí pro vykonání příslušné operace (např. Přidání transakce nevyžaduje potvrzení volby kategorie kliknutím na tlačítko uložit nebo podobným mechanismem, ale již samotná volba kategorie ukončí celý proces.) a zároveň nezahltit uživatele příliš mnoha položkami na jedné obrazovce (např. Volba kategorie je přesunuta na vlastní obrazovku, kde se zároveň umožní rychlejší výběr zobrazením celého nebo téměř celého seznamu najednou.).

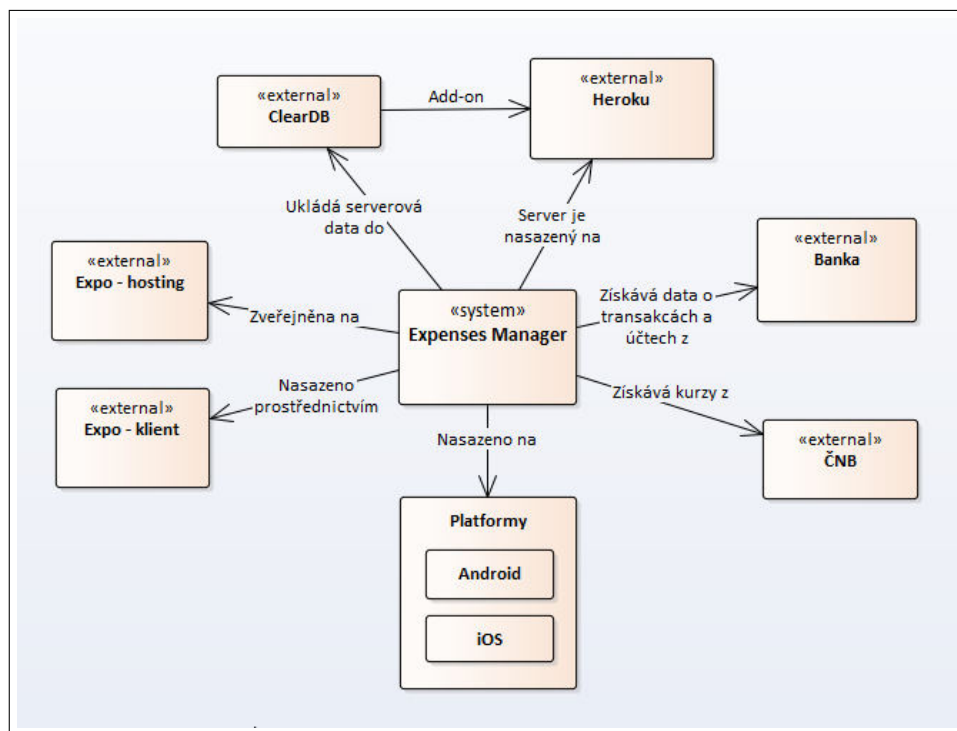
4.3.3 Kontextový model

Kontextový model slouží k pochopení systému/aplikace a jejího zasazení do prostředí ostatních relevantních entit. Model definuje vztahy, závislosti a interakce mezi aplikací a okolním prostředím, ať už jde o osoby, systémy či jiné entity. Účelem tohoto modelu je znázornění a vysvětlení celkového pohledu na aplikaci a jejích interakcí s okolním světem. Sumarizuje také role a zodpovědnosti účastníků těchto interakcí.[138]

Jedinou relevantní rolí v mém případě je uživatel aplikace. Existence této role je zřejmá a nemá dopad na architekturu aplikace, proto ji v modelu blíže nerozebírám.

Entity v okolním prostředí vyvíjené aplikace se dají rozdělit do tří skupin. Jsou to systémy sloužící pro nasazení a ukládání dat serverové části, systémy související s nasazením aplikace a systémy, ze kterých aplikace získává data nezbytná pro svůj provoz.

Serverová část aplikace je nasazena na cloudové aplikační platformě Heroku. Tato platforma zaručuje neustálou dostupnost serveru spolu s možností škálovat prostředky podle potřeby. Kromě běhového prostředí poskytuje Heroku také analytické nástroje pro sledování běhu a zatížení nasazené aplikace. Součástí této platformy je i datové úložiště ClearDB. Jedná se o poskytovatele MySQL databáze, který své služby nabízí jako Heroku add-on. ClearDB

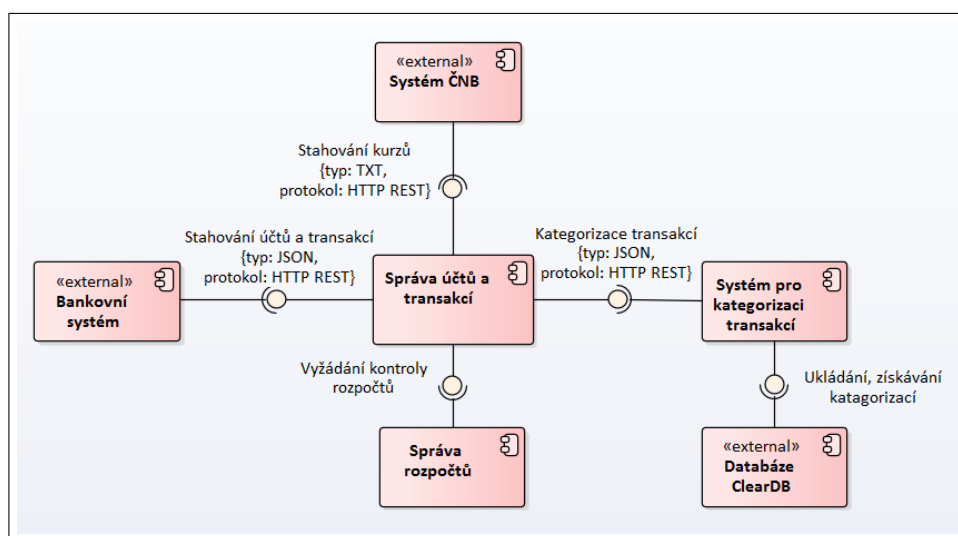


Obrázek 4.15: Kontextový model

zaručuje persistenci dat, jejich zabezpečení a neustálou dostupnost. Veškerá komunikace se servery Heroku je zabezpečena šifrováním.

U samotné aplikace se pro vývoj i nasazení využívá sada nástrojů Expo. Při publikování se aplikace nahraje na jejich servery, čímž se stane dostupnou pro všechny uživatele, kteří mají na svém mobilním telefonu či tabletu staženou aplikaci Expo. Expo tedy zajišťuje dostupnost aplikace jejím uživatelům. Pro bezproblémový běh je důležité sledovat vývoj nových verzí těchto nástrojů a udržovat kompatibilitu s aplikací. Expo zároveň zajišťuje provoz aplikace na dvou majoritních mobilních platformách — Android a iOS.

Poslední skupinou systémů, na kterých je Expenses Manager závislý, jsou systémy organizací, ze kterých se získávají data nezbytná pro správné fungování. Jde zejména o systémy bank, ze kterých se stahují informace o účtech a transakcích uživatelů. Tyto systémy mohou vyžadovat různé způsoby autentizace a autorizace, komunikovat pomocí různých protokolů a poskytovat data v různých formátech. Je proto třeba věnovat se každému zvláště a přizpůsobit se jim. V současné době připadají v úvahu dvě banky (viz. Analýza možností napojení na API bank), přičemž obě vystavují své aplikační programovací rozhraní v souladu s protokolem REST a data poskytují ve formátu JSON. Bližší specifikace těchto rozhraní může být dohledána v příslušných dokumentacích. Česká spořitelna dále využívá autorizační framework OAuth



Obrázek 4.16: Model funkční struktury

2.0.[133, 132, 134, 135]

Druhým z této poslední skupiny je systém České národní banky, který poskytuje mé aplikaci aktuální kurzy několika desítek světových měn. Seznam kurzů je poskytován v textovém formátu, kde jsou jednotlivé měny oddělené novými řádky a položky jako název, kód či kurz svislými čarami. Rozhraní je v souladu s protokolem REST. Seznam měn je poměrně stabilní, ale může se stát, že některá z měn bude vyřazena či přidána.[139]

Komunikace s žádným ze zmíněných systémů nebude příliš objemná, snad s výjimkou serverové části v případě velkého zájmu o aplikaci. Důraz je kladen spíše na jejich dostupnost než na vysokou propustnost nebo krátkou odezvu. Musí se však počítat s tím, že mobilní zařízení nebývají vždy připojena k internetu a aplikace si s tímto stavem musí poradit.

4.3.4 Model funkční struktury

Model funkční struktury zobrazuje funkční celky systému, jak se jeví za jeho běhu. Popisuje jejich zodpovědnosti, rozhraní a hlavní interakce. Účelem modelu je poskytnout přehled vnitřních funkčních celků, rozhraní, která vystavují, a vzájemné interakci mezi nimi a externími systémy.[138]

Interní části vyvíjené aplikace lze rozdělit do tří funkčních celků — Správa účtů a transakcí, Správa rozpočtů a Systém pro kategorizaci transakcí. Tyto celky pak prostřednictvím různých rozhraní komunikují mezi sebou a s externími systémy, jako jsou systémy bank, systém České národní banky a externí úložiště dat.

Jádrem celé aplikace je *Správa účtů a transakcí*. Tento element zobrazuje uživateli všechny přehledy výdajů, příjmů a zůstatku na účtech. Také

umožňuje přidávat, mazat a upravovat účty a transakce. S ostatními částmi systému komunikuje za účelem získání kategorií pro transakce či iniciování kontroly rozpočtů. Od externích systémů pak získává zejména údaje o nových transakcích nebo účtech a dále také aktuální kurzy měn, pro správné zobrazení souhrnných přehledů.

Správa rozpočtů poskytuje funkcionalitu zobrazení rozpočtů, jejich přidání, úprav či mazání a v neposlední řadě jejich kontroly. Ostatním částem systému poskytuje možnost iniciovat kontrolu rozpočtů prostřednictvím prostého volání funkcí. Tuto funkcionalitu využívá Správa účtů a transakcí.

Systém pro kategorizaci transakcí je oddělená část přiřazující kategorie k transakcím. Poskytuje HTTP REST rozhraní pro kategorizaci zasláných transakcí a pro přidání nové kategorizace. Data nutná pro kategorizaci ukládá do externí *Databáze ClearDB*, ke které přistupuje prostřednictvím MySQL konektoru (knihovna).

Z externích systémů je klíčový *Bankovní systém*. Ten poskytuje prostřednictvím HTTP REST rozhraní veškeré potřebné údaje o uživatelských účtech a transakcích, které na nich proběhnou. Dále také poskytuje rozhraní pro autentizaci uživatele nutnou pro přístup ke zmíněným údajům.[132, 133]

Posledním z využívaných externích systémů je *systém ČNB*. Od něj je vyžadována jediná funkce a sice zaslání aktuálního kurzovního lístku na vyžádání. Tato funkce je poskytnuta taktéž prostřednictvím HTTP REST rozhraní.[139]

4.3.5 Datový model

V tomto modelu jsou zobrazeny datové entity, se kterými se v programu pracuje a jsou ukládány do databáze. V kódu programu jsou dostupné ve formě JavaScriptových objektů.

4.3.5.1 Klientská část

1. accounts — všechny informace o účtech získané jak od uživatele, tak z banky
2. transactions — všechny informace o příjmech a výdajích získané jak od uživatele, tak z banky
3. transfers — převody mezi uloženými účty
4. budgets — uživatelem nastavené rozpočty — kolik finančních prostředků hodlá maximálně utratit v dané kategorii
5. rates — kurzy různých měn
6. ratesLastFetch — datum poslední aktualizace kurzů

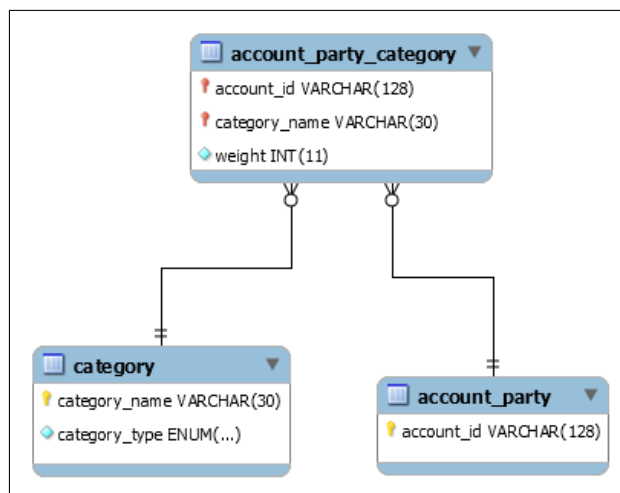
4. ANALÝZA A NÁVRH



Obrázek 4.17: Entity-relationship model — aplikace

U *accounts* se zhruba polovina údajů týká informací zobrazených uživateli v přehledu (např.: název, zůstatek, měna), zbytek je důležitý pro účty napojené na banku a slouží k jednoznačné identifikaci u banky (accountId, číslo účtu, IBAN, kód banky), přístupu k bankovním údajům uživatele (access token, refresh token) a správnému importu transakcí (datum posledního stažení). U *transactions* opět zhruba polovina údajů slouží pro zobrazované přehledy (např.: kategorie, částka, datum), ostatní údaje se týkají protistrany transakce a slouží k automatické kategorizaci nebo identifikaci převodu na jiný vlastní účet. *Transfers* ukládají kolik přibylo/ubylo na jakých účtech, *budgets* ukládají nastavené rozpočty pro vybrané kategorie. *Rates* a *ratesLastFetch* slouží jako cache pro kurzy měn stahované z ČNB.

Od implementace datového úložiště je zde vývojář kompletně odstíněn. S prezentovanými entitami pracuje pouze jako s JavaScriptovými objekty. Pří-



Obrázek 4.18: Entity-relationship model — server

stup k úložišti zajišťuje AsyncStorage (key — value úložiště) od React Native. AsyncStorage volí konkrétní úložiště podle platformy a dostupných databázových systému. U iOS je to serializovaný slovník nebo samostatné soubory pro větší hodnoty, u systému Android se využívá podle dostupnosti RockDB nebo SQLite.[140]

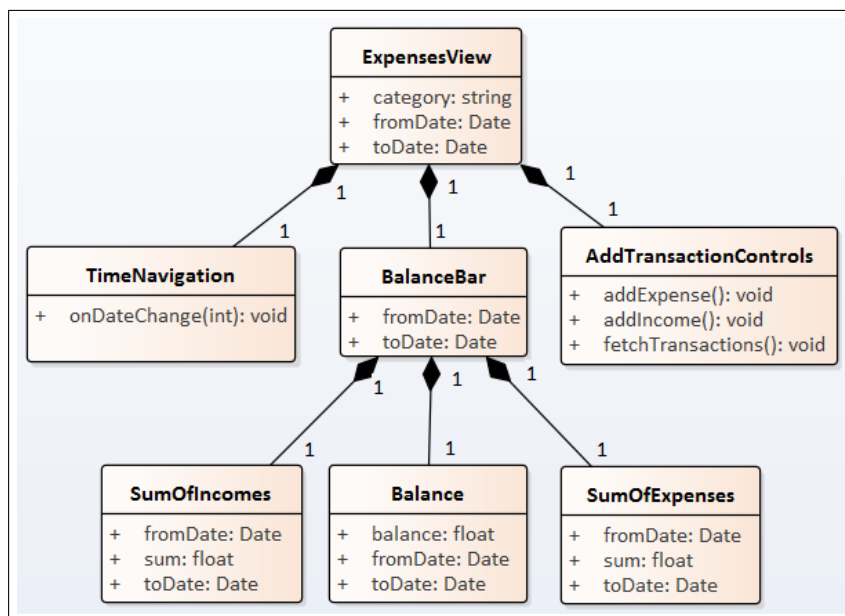
4.3.5.2 Serverová část

1. `account_party` — protistrana transakce identifikovaná hashí čísla účtu a dalších unikátních informací
2. `category` — kategorie transakce — název a zda se jedná o příjem či výdaj
3. `account_party_category` — propojení protistrany transakce s kategorií spolu s vahou tohoto propojení

Na serverové části jsou ukládána data sloužící ke kategorizaci příchozích transakcí. `Account_party` obsahuje hashované údaje identifikující účet protistrany transakce. `Category` ukládá názvy a typ (výdaj, příjem) kategorií. `Account_party_category` potom propojuje předchozí dvě entity a vytváří tak souvislost mezi účtem protistrany a kategorií transakce. Údaj s vyšší vahou má přednost při kategorizaci.

4.3.6 Návrhový model tříd

V následujícím modelu dokumentuji strukturu tříd několika zásadních částí systému. V React Native jsou třídy dědicí od `React.Component` úzce spjaté s vizuální reprezentací. Jejich struktura odpovídá rozložení prvků v jednotlivých obrazovkách aplikace. Tyto komponenty jsou vkládány do sebe, čímž

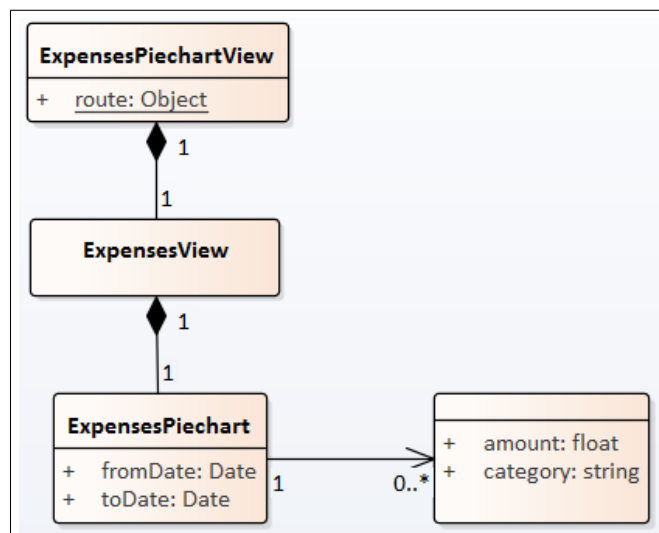
Obrázek 4.19: Návrhový model tříd — `ExpensesView`

vzniká hierarchie elementů podobná struktuře HTML stránky. Pro vytváření takovéto kompozice byl dokonce vyvinut vlastní jazyk, JSX, který skutečně připomíná HTML.

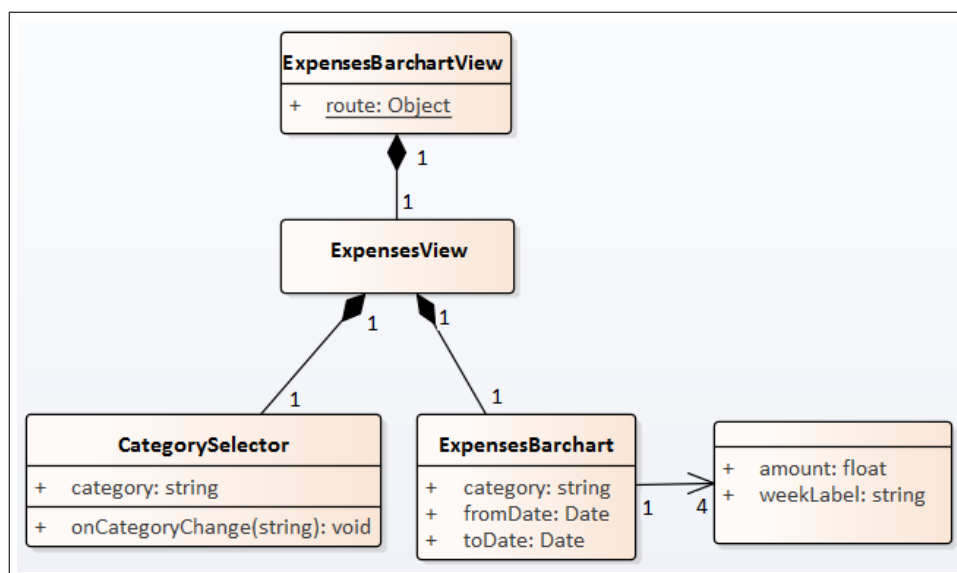
Na prvním diagramu je k vidění složení komponenty *ExpensesView* 4.19. Ta slouží jako základní layout pro tři přehledové obrazovky — Přehled výdajů, Přehled výdajů v čase a Transakce. Obsahuje prvky shodné pro všechny tyto přehledy. Je to navigace v čase, sloužící k zobrazení přehledu za jiné období, řádek s bilancí ve sledovaném období a lišta s ovládacími prvky pro přidání nové transakce nebo import transakcí z banky. Důležitými atributy jsou zejména časové hranice sledovaného období.

Další tři diagramy odpovídají zmíněným přehledům 4.204.214.22. V prvních dvou případech jsou do `ExpensesView` přidány příslušné grafy. V případě Transakcí je to pak seznam (`ListView`) obsahující prvky typu `TransactionsListItem`. U Přehledu výdajů v čase a Transakcí je navíc přidána komponenta pro výběr kategorie.

Diagramy pro Účty a Rozpočet jsou si v mnohém podobné 4.234.24. V obou se opakuje vzor se seznamem prvků, jeden pro každý odpovídající objekt z příslušného *View* (`AccountsListItem` pro *Account*, `BudgetListItem` pro *Budget*). Oba také obsahují objekt reprezentující tlačítko pro přidání nového prvku příslušného typu (`FullWidthButton` spolu s funkcí v `AccountsView`, respektive `BudgetView`). U Účtů je navíc komponenta sloužící pro výpočet a zobrazení celkového zůstatku na účtech a další pro tlačítko, jehož účelem je navigace na obrazovku s přidáním nového převodu.

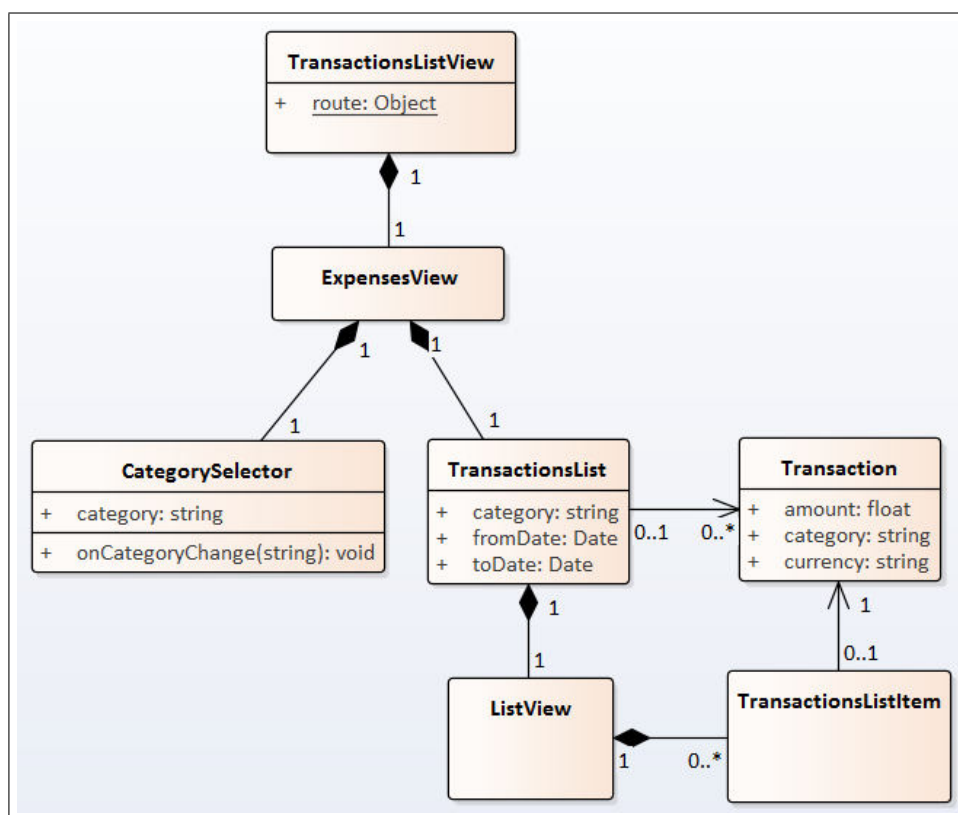


Obrázek 4.20: Návrhový model tříd — přehled výdajů

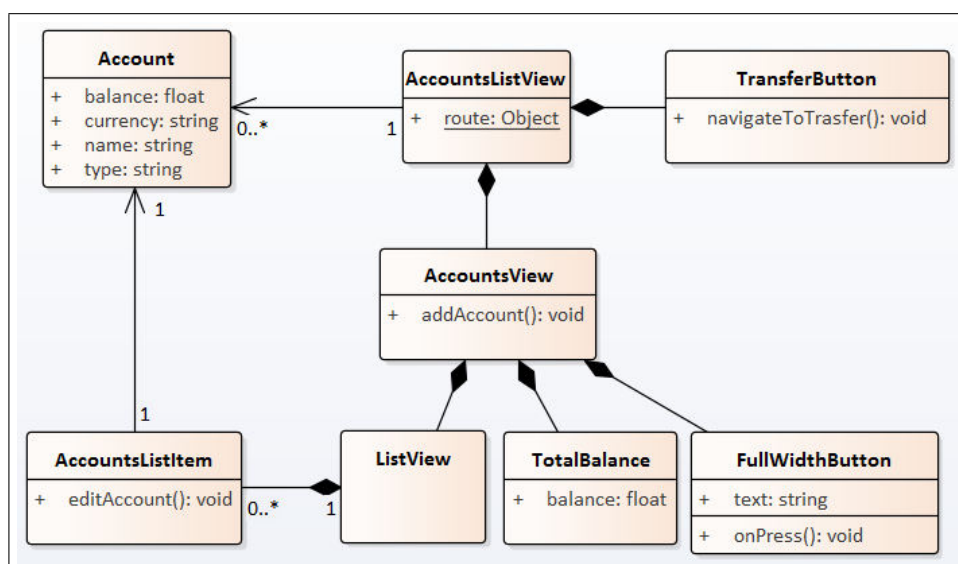


Obrázek 4.21: Návrhový model tříd — přehled výdajů v čase

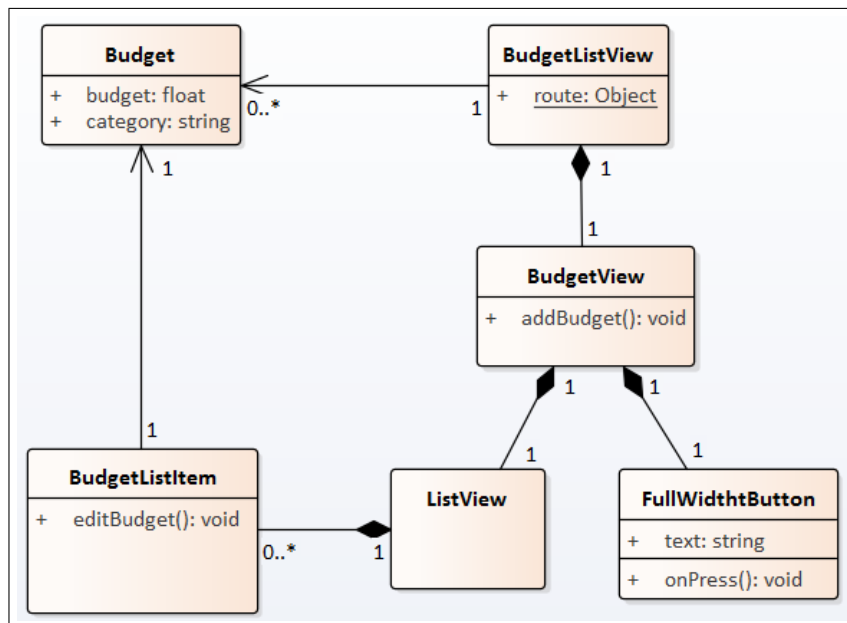
4. ANALÝZA A NÁVRH



Obrázek 4.22: Návrhový model tříd — transakce



Obrázek 4.23: Návrhový model tříd — účty



Obrázek 4.24: Návrhový model tříd — rozpočet

4.3.7 Modely klíčových procesů

Pro lepší pochopení fungování systému zde uvádím modely několika vybraných procesů v podobě sekvenčních diagramů. Součástí všech těchto procesů je komunikace s externím bankovním systémem nebo serverovou částí systému. Tyto jsou z hlediska interakce mezi různými objekty, komponentami, systémy nejsložitější a proto je zde blíže rozebírám.

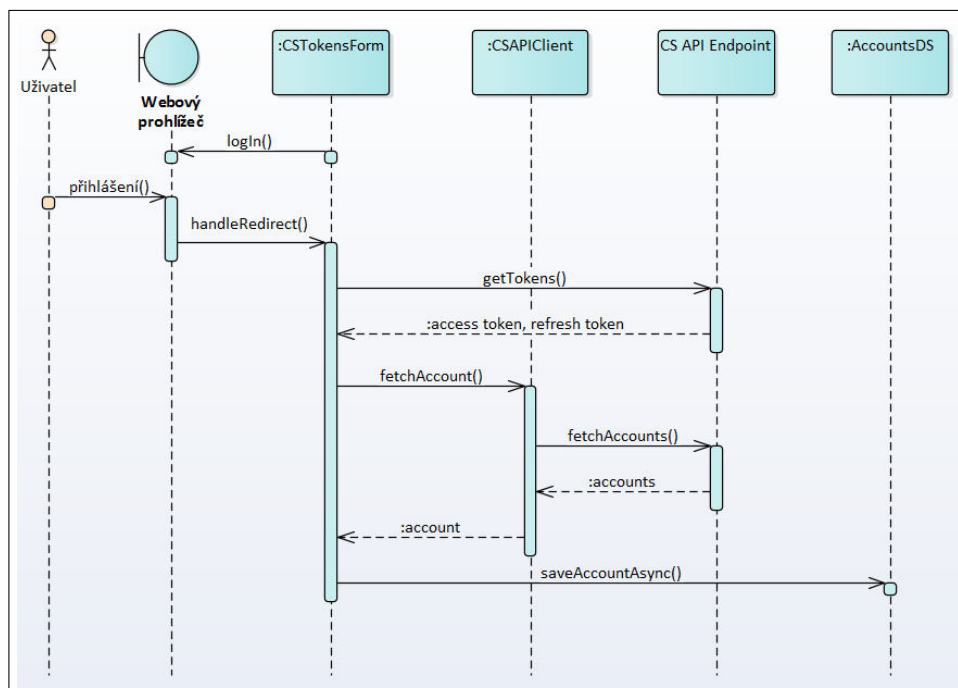
4.3.7.1 Přidání účtu

První vybraný proces je přidání nového účtu.4.25 Jako začátek jsem zde pro přehlednost zvolil až uživatelskou autentizaci u jeho banky, ostatní údaje o účtu považuji již za vyplněné. Proces tedy začíná zvolením možnosti „přihlásit se“ a končí uložením veškerých dostupných údajů o účtu do databáze. V průběhu se pro uživatele otevře prohlížeč na přihlašovací stránce banky. Po vyplnění přihlašovacích údajů dojde k přesměrování zpět do aplikace a následnému stažení přístupových tokenů nutných pro získání informací o účtech uživatele. Tyto informace jsou vzápětí staženy a následně uloženy do databáze pro budoucí použití.

4.3.7.2 Import transakcí

Proces importu nových transakcí je zřejmě ten zcela nejsložitější.4.26 Zahrnuje komunikaci jak s bankovním systémem, tak s vlastním serverem. Začíná se

4. ANALÝZA A NÁVRH



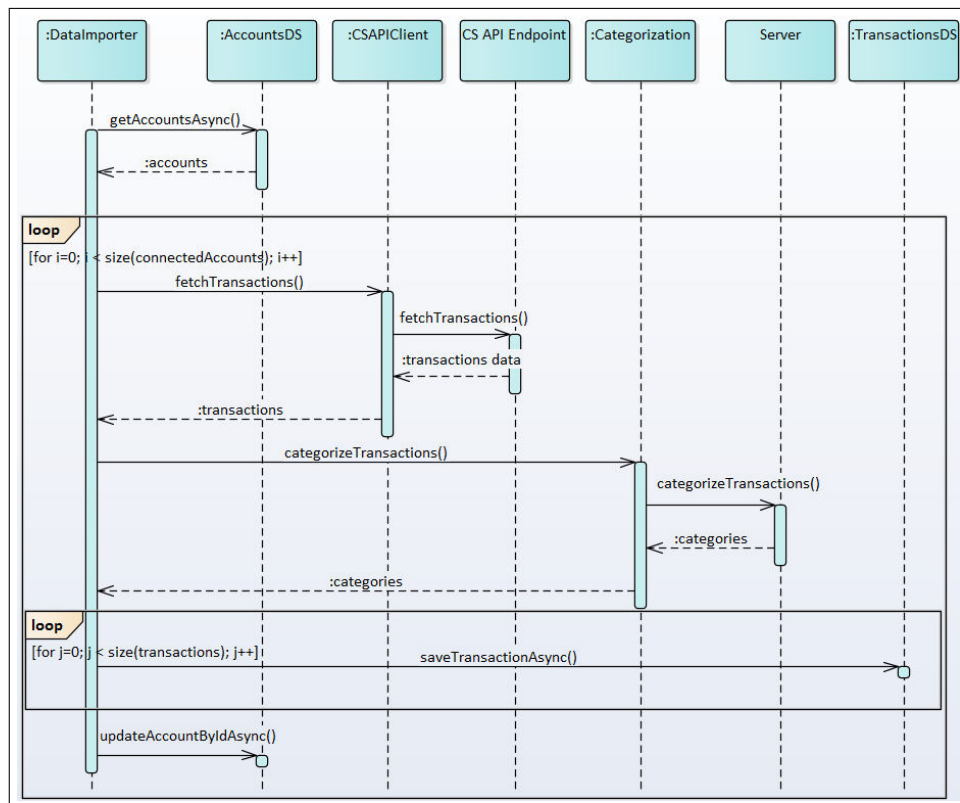
Obrázek 4.25: Model procesu — přidání účtu

získáním všech uložených účtů připojených k podporované bance. Pro každý z těchto účtů jsou z příslušné banky staženy nové transakce. Ty jsou následně zaslány ke kategorizaci na server. Kategorizované transakce jsou uloženy do databáze. Posledním krokem je aktualizování údajů o účtu k němuž transakce náleží.

4.3.7.3 Kategorizace transakcí

Kategorizace transakcí na serveru zahrnuje dva procesy. Jedním je přidání nové kategorizace specifikované uživatelem v aplikaci.4.27 Na server je při něm zaslána dvojice protistrana transakce — kategorie. Informace o protistraně jsou zahashovány a podle výsledné hashy se v databázi dohledá, zda takto již někdo protistranu kategorizoval. Pokud ne, vloží se do databáze nová kategorizace s vahou jedna. V opačném případě se pouze inkrementuje váha u již existující kategorizace.

Druhým z procesů kategorizace je samotné přidělení kategorie k přijatým transakcím.4.28 Podobně jako v předešlém případě se nejdříve zahashuje informace o protistraně transakce. S touto hashí se opět dohledá, zda pro ni již existuje kategorizace. Pokud ano, vybere se kategorie s nejvyšší vahou. Pokud ne, kategorie zůstane nespecifikována. Kategorie pro všechny zaslání transakce se nakonec odešlou zpět do mobilní aplikace.

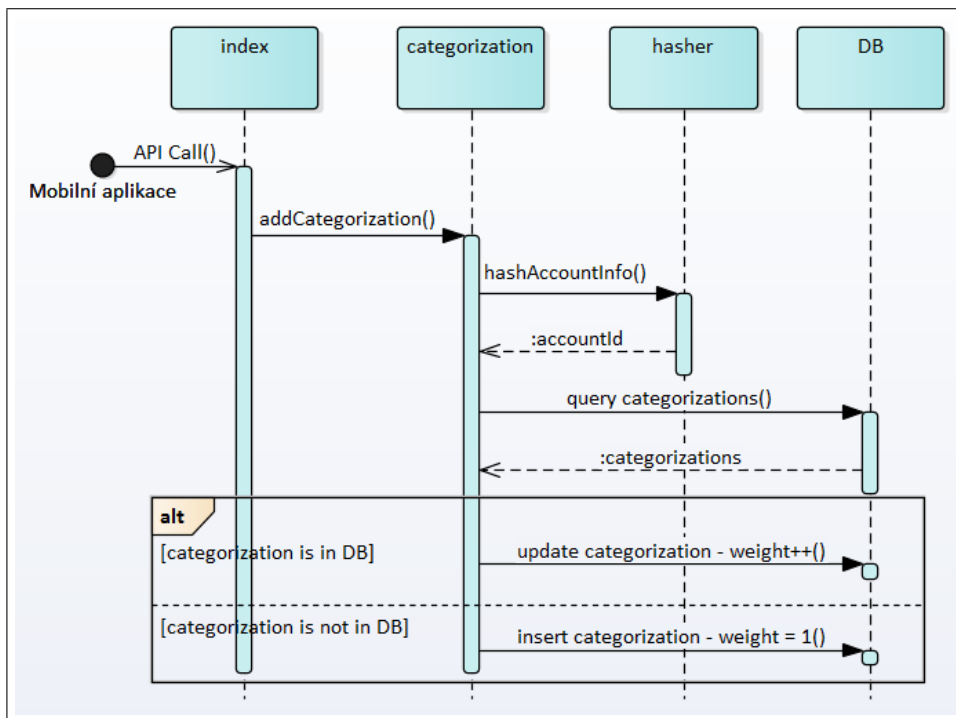


Obrázek 4.26: Model procesu — import transakcí

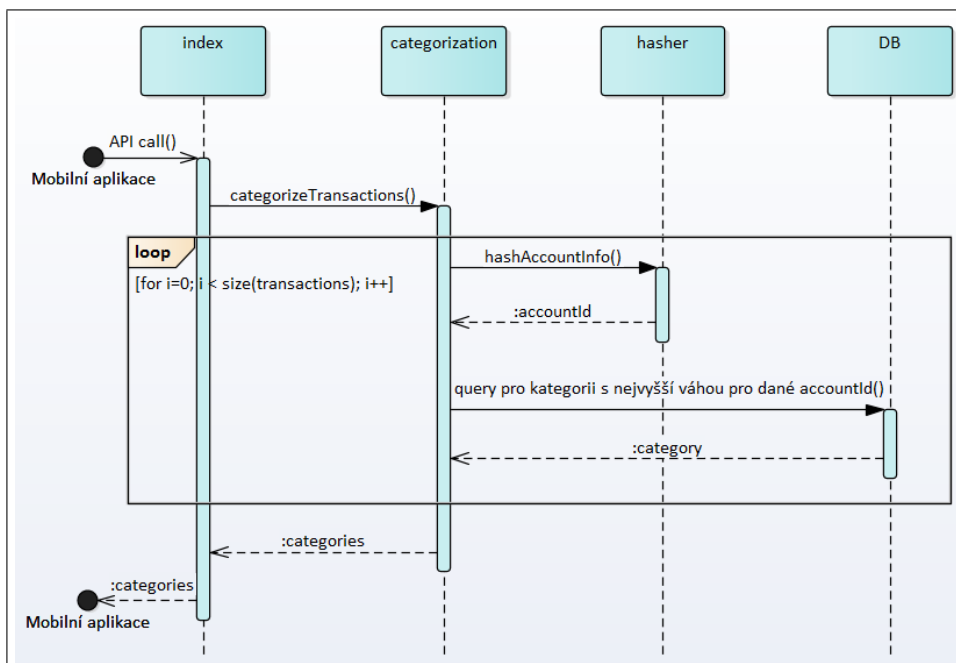
Při vybírání metody automatické kategorizace jsem zvažoval různé možnosti a tato mi vyšla jako nejspolehlivější a nejsnáze proveditelná. Bez účasti uživatele bych byl odkázán pouze na informace o transakci stažené z banky. Zde by v úvahu připadala např. kategorizace na základě názvu účtu, zpráv pro příjemce/odesilatele či popisu transakce. Tato data, mimo názvu účtu, však nejsou povinná a tak nemusí být vždy přítomna. Množství informací v nich je navíc často velmi malé (pouze pár slov) a nikdo nezaručuje, že se bude týkat kategorie transakce. Analýza textu a přirozené řeči navíc nebyla předmětem mého studia. Názvy obchodů v těchto datech také někdy bývají různě zkomoleny a tak by úspěšnost takovéto metody byla velmi sporná. Využít by se pro kategorizaci teoreticky daly také ostatní údaje, jako částka či datum a čas transakce. To by ovšem pravděpodobně vyžadovalo rozsáhlou a časově velmi náročnou analýzu a hluboké znalosti v oboru.

Současný postup by se dal vylepšit například předvyplněním dat v databázi o známé obchody a jiné podniky. Tím by se zlepšila šance na automatickou a správnou kategorizaci transakce bez prvotního vstupu od uživatelů.

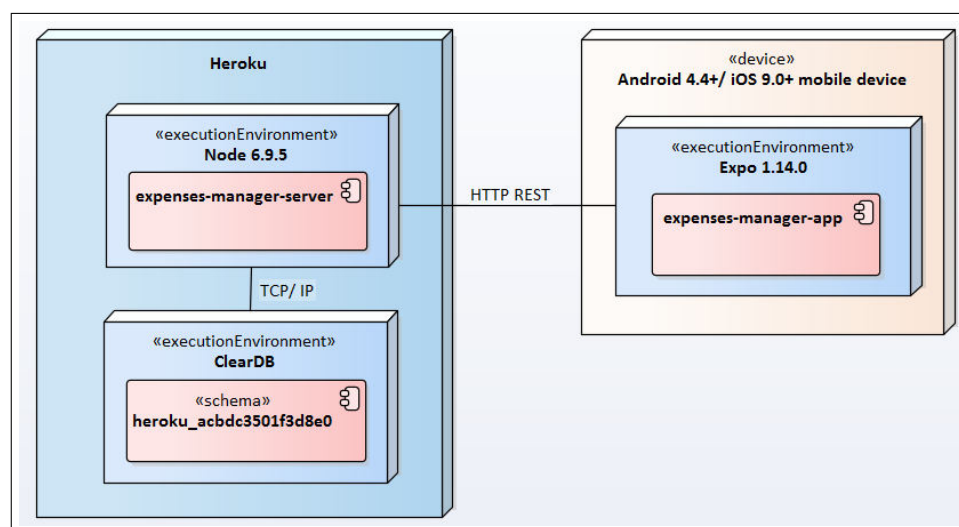
4. ANALÝZA A NÁVRH



Obrázek 4.27: Model procesu — přidání kategorizace



Obrázek 4.28: Model procesu — kategorizace transakcí



Obrázek 4.29: Model nasazení

4.3.8 Model nasazení

Tento model 4.29 znázorňuje, jak jsou jednotlivé softwarové části systému rozloženy na hardwarové komponenty. Dále také ukazuje jaká prostředí musí být na těchto komponentách připravena, aby byl umožněn běh systému.

Technologie nutné pro běh systému:

server — Node.js (ověřená verze: 6.9.5)

aplikace — Expo klient (ověřená verze: 1.14.0), Android (min. verze: 4.4) nebo iOS (min. verze: 9.0)

Realizace

5.1 Použité technologie

5.1.1 React Native

React Native je framework pro vývoj hybridních multiplatformních mobilních aplikací (Android, iOS) v JavaScriptu. Termínem hybridní se zde myslí běh programu jak v JavaScriptu, tak v nativním kódu platformy. Hlavním mottem React Native je „*Learn once, write anywhere*“, tedy přístup k programování, kdy se stačí naučit jednu sadu principů, vzorů, jeden postup a ten pak používat při programování pro různé platformy. Jak vyplývá z názvu, klíčovou je také skutečnost, že přestože je kód psaný v JavaScriptu, jsou využívány nativní komponenty a možnosti platformy.[12, 141]

React Native je založený na React, JavaScriptové knihovně pro vývoj webových uživatelských rozhraní. Toto rozhraní se skládá z takzvaných komponent. Jedna komponenta typicky odpovídá jednomu funkčnímu prvku uživatelského rozhraní a obsahuje vše nutné pro jeho chování a vykreslení, ať jsou to funkce měnící jeho stav, styly upravující jeho vzhled nebo další komponenty, ze kterých se skládá. Právě vkládáním těchto komponent do sebe se vytváří výsledný produkt. Konceptuálně je komponenta funkce, která přijímá libovolné parametry (tzv. props — zkráceně properties) a vrací React elementy popisující, co se má vykreslit na obrazovku. Skládání komponent pak může být provedeno dvěma způsoby: Předáváním v parametrech funkce nebo za využití k tomuto účelu vytvořeného značkovacího jazyka JSX. Ten svou strukturou připomíná např HTML. Komponenta může být definována jako klasická funkce nebo jako ECMAScript 6 třída.[12, 142, 141, 143]

V React Native je uživatelské rozhraní programováno deklarativně. Definuje se, jak má aplikace vypadat v závislosti na vstupních datech a React Native provede potřebné změny, když se změní stav. View je vlastně čistým výstupem stavu aplikace. Vývojář se tak nesoustředí na samotné měnění view, ale musí zajistit, aby jeho komponenty zvládly správně reagovat na různé stavy

aplikace.[144, 145, 146]

Jedním z důvodů pro vznik React native bylo umožnění webovým vývojářům programovat i pro mobilní platformy. Pokud vývojář zná JavaScript, HTML a CSS nebo dokonce již pracoval s React, je přechod na React native velmi přímočarý. Na rozdíl od webových aplikací, které lze také spustit na mobilních zařízeních, má tento framework přístup k nativním možnostem platformy, jako jsou sofistikovaná gesta, odladěné standardní animace a podobně. Tím výsledná aplikace získá skutečný „*look and feel*“ cílové platformy, kterého lze u webové aplikace dosáhnout jen stěží, zda-li vůbec. Dále také nevyžaduje inherentně přístup k internetu.[141] Oproti aplikacím psaných v nativním kódu, Java pro Android a Objective C nebo Swift pro iOS, jsou aplikace psané v React Native o něco méně výkoné. Nejpomalejší jsou v místech, kde se přechází z JavaScriptu do nativního kódu a obráceně.[147]

Poněkud rozporuplné reakce vyvolává přístup k členění kódu. Zavedenou praxí je psát kód týkající se vzhledu (např. CSS) do oddělených souborů. V React Native se však upřednostňuje zapisovat definice stylů přímo do souboru s hlavním kódem. Tyto styly jsou definované v JavaScriptu a komponentám se přidávají takzvaně inline. Podobně jako styly i značkovací jazyk JSX, ve kterém se definuje skládání komponent, se zapisuje do stejného souboru, což jde opět proti zavedenému trendu. Vývojáři React Native to zdůvodňují snahou o usnadnění orientace v kódu, kdy všechno, co se týká jedné komponenty, jednoho funkčního prvku aplikace, je na jednom místě. Jde o celek, který lze znovu použít na různých místech aplikace bez dalších úprav, protože vše, co je třeba pro jeho fungování a zobrazení je přímo v něm.[141]

Výsledná aplikace není kompilována do nativního kódu, ale využívá JavaScript engine. JavaScriptový kód běží asynchronně ve vlastním vlákne na pozadí, odkud řídí běh aplikace. Základní komponenty, které nabízí knihovna React Native, jsou vlastně jen obalené nativní komponenty. Výsledné view je tedy poskládáno z nativních částí a ty jsou posílány do hlavního vlákna, kde běží nativní kód.[143] Běh aplikace přímo v JavaScriptu bez kompilování umožňuje rychlý re-loading aplikace a to v mnoha případech i bez ztráty stavu (metoda známá jako hot loading). Tím se značně usnadňuje práce vývojářům. Další výhodou je možnost nahrávat změněný kód ze serveru do již nasazených aplikací a tím urychlit proces aktualizace bez nutnosti projít procesem schvalování u Google Play nebo iTunes Store.[146] Je otázkou, jestli obcházení kontrol těchto obchodů je ta správná cesta.

React Native obsahuje mnoho užitečného, ale jde stále o mladou technologii s omezenými možnostmi oproti nativním jazykům. V případě, že neposkytuje některou vyžadovanou funkci, je ale možno ji doprogramovat v nativním kódu. Vývoj frameworku navíc probíhá velmi rychle. Nová verze vychází každé dva týdny. To je na jednu stranu příjemné, nové funkce přibývají vysokým tempem, ale na druhou stranu tyto změny často způsobí problémy v dříve fungujících aplikacích. Je proto třeba se jim neustále přizpůsobovat. Nativní jazyky také budou vždy lépe připravené na příchod nové verze systému. React

Native není přímo podporován od Google ani Apple a proto vždy chvíli trvá, než se přizpůsobí novým změnám.[145, 148]

V React Native lze psát aplikace pro více platforem. To ale neznamená, že stačí pro obě verze napsat pouze jeden kód. Mnoho z nabízených základních komponent je skutečně společných pro Android i iOS, ale nemalé množství je rozdílných. Celkově je překryv kódu pro obě platformy kolem 80%. [145] To je relativně mnoho oproti nativním jazykům, ale například u webových aplikací stačí skutečně napsat kód jen jeden. Pokud je záměr vývojáře vytvořit jednu multiplatformní aplikaci, existuje v React Native způsob, jak zjistit aktuální platformu. Podle výsledku se pak může určit, která komponenta a s jakými parametry se použije.[144]

5.1.2 Expo

Expo je sada nástrojů, knihoven a služeb vystavěných kolem React Native. Aplikace vytvořené s Expo jsou React Native aplikace obsahující Expo SDK. Hlavní myšlenkou Expo je maximální usnadnění vývoje nové aplikace snížením počtu nastavení, která je nutné udělat na začátku projektu, naprostým odstíněním od nativního kódu a poskytnutím nástrojů pro jednoduché debugování, sdílení a nasazení.[149, 150, 151]

Expo dovádí myšlenky React Native do extrému. Veškerý kód je psaný pouze v JavaScriptu, přístup do nativního kódu není umožněn. Toto je podpořeno sadou vlastních komponent (např. ikony, blur view), API (např. přístup k foťáku, kontaktům, místnímu úložišti) a služeb (např. správa assets, push notifications) rozšiřujících nabídku React Native. V základní instalaci také obsahuje mnoho knihoven třetích stran (např. react-native-maps, Facebook authentication) dále rozšiřujících vývojářské možnosti takzvaně „out of the box“. Díky tomu, že je 100% kódu v JavaScriptu, může vývojář maximálně těžit z nabízených výhod, jako jsou: okamžité nasazení „over-the-air“ bez nutnosti kompilace a posílání APK či IPA souborů po každé změně, odpadnutí nutnosti znát nativní jazyky a jejich vývojářská prostředí u obou platforem, využívání aktualizací a podpory Expo. Pokud je aplikace napsaná pouze v JavaScriptu a využívá pouze multiplatformní součásti (všechny v Expo, většina v React Native) jde zároveň o skutečně multiplatformní aplikaci, kdy stačí napsat jeden kód a použít jej na obou platformách.[152, 149, 153]

Skutečnost, ze které vyplývají hlavní výhody Expo, tedy veškerý kód v JavaScriptu, je zároveň i jeho největší nevýhodou. Pokud některá funkce není dostupná (např. práce s Bluetooth), vývojář nemá přímočarou možnost ji implementovat v nativním kódu, jak je tomu u čistého React Native. Jedinou možností v takovém případě je „detach“ od Expo, tedy z většiny automatická transformace, při které se z Expo projektu přejde na velmi podobný takovému, jaký používá React Native. Tato transformace je zatím v alfa fázi vývoje.[152, 149]

Velké zefektivnění vývoje přináší nástroj XDE (Expo development environment) a jeho verze pro příkazový řádek. Jde o samostatnou desktop aplikaci, která obsahuje všechny závislosti nezbytné pro vývoj. Umožňuje spustit Expo projekt a vygeneruje jeho URI. Na mobilním zařízení je pak třeba nainstalovat klientskou aplikaci. Ta slouží jako prohlížeč pro Expo aplikace. Do té se vloží vygenerované URI, které využije pro stažení a spuštění vyvíjené aplikace. Veškeré změny provedené v kódu aplikace se při tom v rozmezí několika vteřin projeví v nasazené aplikaci. V samotném XDE jsou zobrazovány logy ze zařízení a z React Packager. XDE také umožňuje jedním kliknutím publikovat aplikaci na serveru Expo, čímž se stane dostupnou pro všechny potenciální uživatele s nainstalovaným klientem. Pokud je to žádoucí, lze v XDE vygenerovat APK a IPA archivy, které je možné umístit na Google Play a iTunes Store. Samotné generování těchto souborů se provádí na serverech Expo, takže není třeba instalovat příslušný software lokálně.[149, 150]

Expo je stejně jako React Native velmi mladá technologie procházející neustálým vývojem. Možnosti oproti nativním jazykům jsou ještě omezenější, protože je třeba vybírat z React Native jen ty funkce a komponenty, které nepotřebují konfiguraci v nativních prostředích. Na druhou stranu, jak bylo řečeno výše, Expo přináší sadu vlastních rozšíření. Stejně jako u React Native i zde s sebou překotný vývoj přináší výhodu rychlého přibývání funkcionalit a nevýhodu nutnosti neustálé adaptace na větší změny, jak je patrné z oznámení nových verzí na blogu Expo.[149, 154]

5.1.3 Ostatní

Node.js je běhové prostředí pro JavaScript postavené na V8 JavaScript engine od Chrome. Umožňuje tedy běh JavaScriptových aplikací mimo prohlížeč. Jako jedna z hlavních možností využití Node.js se uvádí implementace REST API.[155, 156] Toho jsem využil, abych nemusel opustit JavaScript ani u serverové části mého systému. Spolu s použitím webového frameworku Express stačilo napsat necelých dvacet řádků kódu pro nastavení HTTP serveru a vystavení dvou endpointů.

Pro hashování údajů o účtu na serveru jsem použil knihovnu node-scrypt implementující kryptograficky silnou funkci scrypt.[157] Serverová část systému je nasazená na cloudové platformě Heroku. Důvodem byla zejména podpora požadovaných technologií a základní provoz zdarma.[158]

Aby bylo možné napsat jeden kód a nasadit aplikaci na obou cílových platformách, bylo nutné často hledat podporu knihoven třetích stran, pro komponenty nedostupné v samotném React Native a Expo. Vzhledem k mládí používaných technologií byl však výběr omezený, knihovny často nejsou perfektně odladěné a chvíli trvá, než se přizpůsobí novým verzím. Autoři však podle mé zkušenosti velmi ochotně komunikují a aktivně řeší nahlášené problémy.

Pro vykreslování grafů se používá knihovna react-native-pathjs-charts[159], podporující oba používané typy — koláčový a sloupcový graf. Dále je s drob-

nými úpravami využíván multiplatformní „picker“ `react-native-picker-xg` pro výběr kategorií, účtů atp. a multiplatformní „picker“ pro výběr data `react-native-datepicker`. Formátování částek v různých měnách pak velmi usnadnila knihovna `react-native-globalize`.

5.2 Uživatelské rozhraní

Tvorba uživatelského rozhraní je hlavní doménou React a tedy i React Native.[142, 12] Jednotlivé prvky rozhraní odpovídají React komponentám a jejich skládáním se dosáhne výsledného produktu. Komponenty mají svou grafickou reprezentaci, která se může lišit na různých platformách. Pro nastavení vzhledu má React svůj vlastní systém příbuzný CSS. Vývojáři je vysoce ceněna zejména možnost používat flexbox algoritmus.[160, 161, 162] Ten umožňuje dosáhnout konzistentního vzhledu na různě velkých obrazovkách díky dynamickému přizpůsobování rozměrů a rozložení komponent.[163, 164]

Pro vytvoření lepší představy o výsledné aplikaci přikládám obrázky několika základních obrazovek. Na obrázku 5.1 je seznam účtů a zůstatků na nich. Za povšimnutí zde stojí zejména částka nad seznamem, poskytující součet všech zůstatků v jedné měně. Následují tři obrazovky poskytující přehledy o výdajích ve formě grafů a seznamu transakcí.5.25.35.4 Nahoře je vždy vidět zvolený měsíc spolu s šipkami umožňujícími přejít na předešlý nebo nadcházející měsíc. Pod ním je součet příjmů, bilance a součet výdajů za vybraný měsíc. U dvou přehledů lze zvolit kategorii výdajů, pro kterou se zobrazí přehled. Na středu je vždy hlavní přehled, ať je to koláčový graf s výdaji podle kategorie, sloupcový graf se součty výdajů po týdnech nebo seznam transakcí. Ve spodní části jsou tlačítka pro přidání nového výdaje nebo příjmu a pro import transakcí z připojených bankovních účtů.

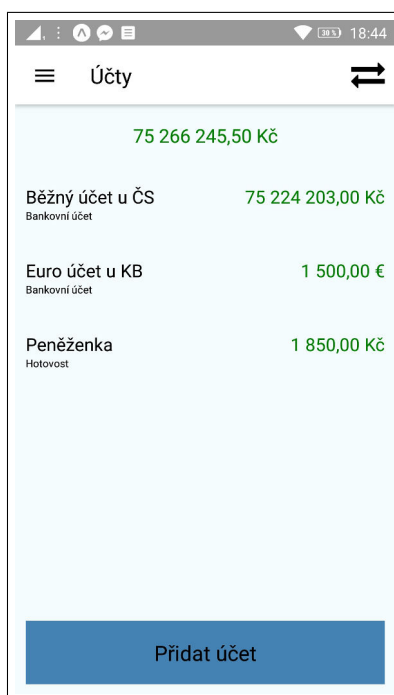
Další obrazovka slouží pro přidání nové transakce.5.5 Pro výpočet částky u složitějších transakcí se k zadávání používá jednoduchá kalkulačka. Poslední obrazovka není přímo součástí aplikace.5.6 Jde o stránku sloužící k autentizaci u České spořitelny, na kterou je uživatel naveden při přidávání nového účtu. Po úspěšném přihlášení dojde k přesměrování zpět do aplikace.

5.3 Ukázky kódu

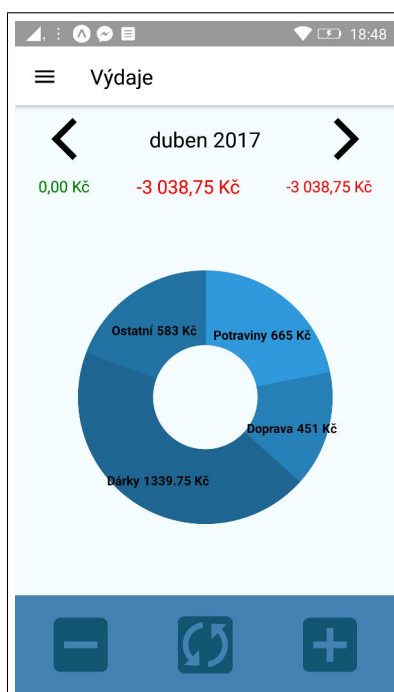
5.3.1 Komponenty, JSX, props

První ukázka je definice jednoduché komponenty *BalanceBar*. Ta je definována jako třída dědící od *React.Component*. Jedinou povinnou metodou je *render()*, která stanoví, jak se má komponenta vykreslit. Návrátová hodnota je specifikována v jazyce JSX. V příkladu je názorně vidět, jak se komponenta *BalanceBar* skládá ze tří dílčích komponent *SumOfIncomes*, *Balance* a *SumO-*

5. REALIZACE

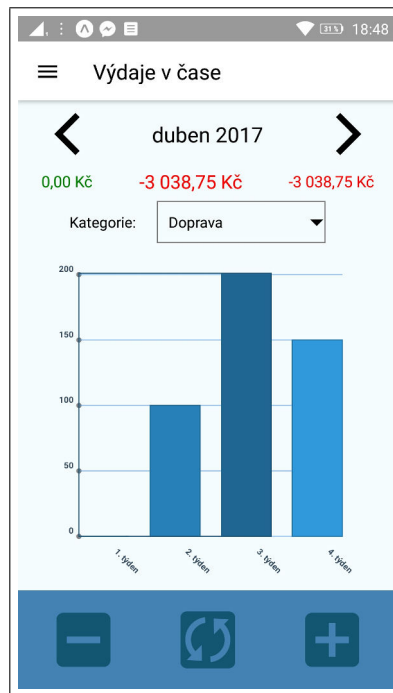


Obrázek 5.1: Účty



Obrázek 5.2: Výdaje podle kategorie

5.3. Ukázky kódu



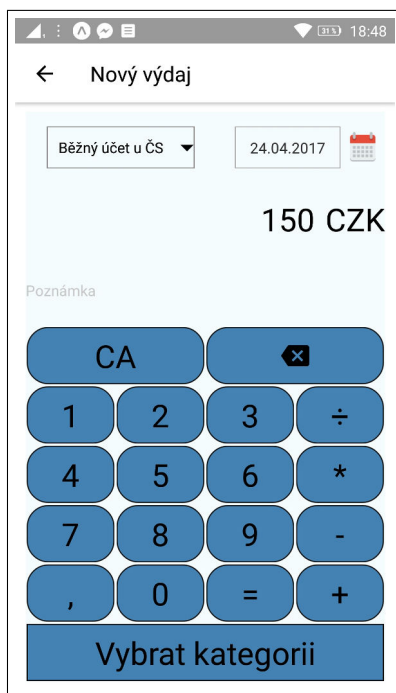
Obrázek 5.3: Výdaje v čase

The screenshot shows a mobile application interface for tracking transactions. The title is 'Transakce'. The date is 'duben 2017'. The total expense is -2 388,75 Kč. The category is 'Vše'. The transactions are:

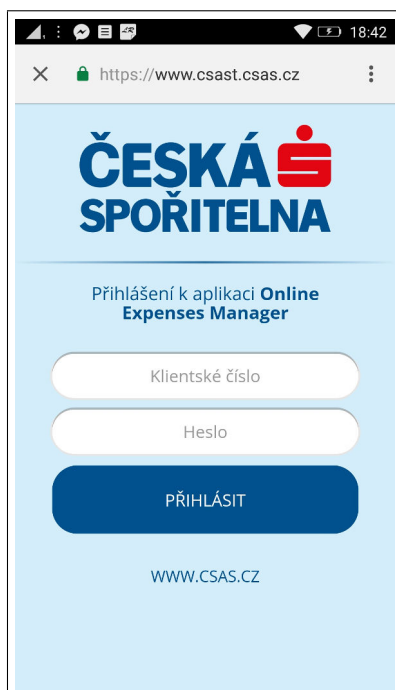
Category	Date	Expense (Kč)
Potraviny	24.4.2017	-165,00 Kč
Dárky	24.4.2017	-50,00 €
Doprava	19.4.2017	-100,00 Kč
Doprava	19.4.2017	-100,00 Kč
Doprava	19.4.2017	-1,00 Kč
Ostatní	12.4.2017	-583,00 Kč
Doprava		-100,00 Kč

Obrázek 5.4: Transakce

5. REALIZACE



Obrázek 5.5: Nový výdaj



Obrázek 5.6: Autentizace u banky

Obrázek 5.7: Příklad komponenty *BalanceBar*

fExpenses. Komponenta *View* slouží jako kontejner pro tyto části. Každá nová komponenta musí obsahovat právě jednu kořenovou komponentu.

BalanceBar používá tři hodnoty — *fromDate*, *toDate* a *style*. Tyto hodnoty jsou jí předány z nadřazené komponenty. Způsob předání je zobrazen ve druhém výstřížku kódu. Zápis je podobný jako u HTML atributů. V metodách *BalanceBar* jsou pak dostupné prostřednictvím objektu *props*.

```

1 export default class BalanceBar extends Component {
2   render() {
3     return (
4       <View style={[styles.balance, this.props.style]}>
5         <SumOfIncomes fromDate={this.props.fromDate} toDate={this
6           .props.toDate} />
7         <Balance fromDate={this.props.fromDate} toDate={this.
8           props.toDate} />
9         <SumOfExpenses fromDate={this.props.fromDate} toDate={
10            this.props.toDate} />
11       </View>
12     );
13   }
14 }

```

```

1 <BalanceBar style={styles.balance} fromDate={this.state.fromDate}
2   toDate={this.state.toDate} />

```

5.3.2 Navigace

Navigace v aplikaci je řešena s pomocí knihovny `@expo/ex-navigation`. Nejprve je definován router, kde se přiřadí obrazovky ke klíčům. Poté se do kořenové komponenty aplikace vloží *NavigationProvider* spolu se specifikací navigační lišty a navigační komponentou *DrawerNavigationPanel*. Tím je umožněno využívat této navigace v celé aplikaci.

```

1 export const Router = createRouter(() => ({
2   pieChart: () => ExpensesPieChartView,
3   barChart: () => ExpensesBarChartView,
4   accounts: () => AccountsListView
5   ...
6 }));
7
8 class App extends Component {
9   render() {
10    return (
11      <FormattedWrapper locale="cs" currency="CZK" cldr={[require
12        ('./localization/currencies-cs-cldr.json')]}>

```


5. REALIZACE

```
12     <NavigationProvider router={Router}>
13       <StatusBar barStyle="light-content" />
14       <DrawerNavigationPanel />
15     </NavigationProvider>
16   </FormattedWrapper>
17 );
18 }
19 }
```

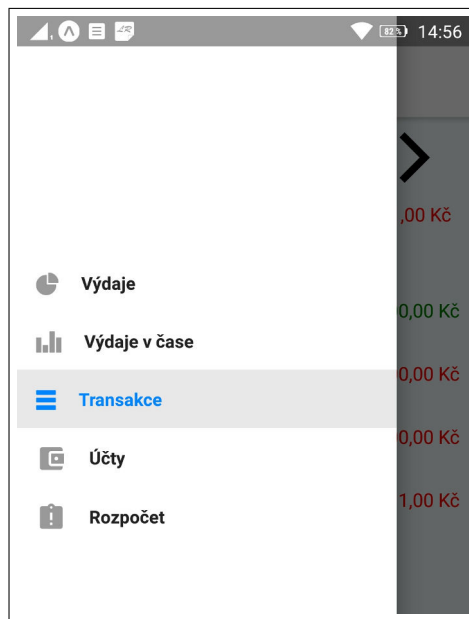
V *DrawerNavigation* je specifikován vzhled a pozice vysouvacího panelu s navigačním menu. Dále pak jednotlivé položky v tomto menu. Pro každou je zvolen název, ikona a nový navigační zásobník, na který se po zvolení dané položky umístí první obrazovka. Vstupní položka aplikace je specifikována v parametru *initialItem*. V jednotlivých navigačních zásobnících je poté možno se pohybovat pomocí šipky zpět. Pro vložení iniciální obrazovky na zásobník je třeba ji nejprve získat z routeru s pomocí příslušného klíče — *Router.getRoute('pieChart')*.

```
1 <DrawerNavigation
2   drawerPosition="left"
3   renderHeader={this._renderHeader}
4   drawerWidth={300}
5   initialItem="pieChartItem">
6   <DrawerNavigationItem
7     id="pieChartItem"
8     selectedStyle={styles.selectedItemStyle}
9     renderTitle={isSelected => this._renderTitle('Výdaje',
10      isSelected)}
11     renderIcon={isSelected => this._renderIcon('graph-pie',
12      isSelected)}>
13     <StackNavigation
14       id="pieChart"
15       defaultRouteConfig={{
16         navigationBar: {
17           tintColor: 'black'
18         },
19       }}
20       initialRoute={Router.getRoute('pieChart')}
21     />
22   </DrawerNavigationItem>
23   ...
```

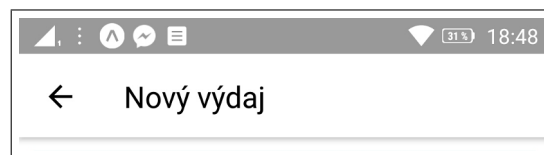
V kořenové komponentě každé obrazovky je možné specifikovat obsah navigační lišty. Zde je to titulek „Nový výdaj“.

```
1 static route = {
2   navigationBar: {
3     title: 'Nový výdaj'
4   },
5 }
```

K další navigaci / vložení nové obrazovky na navigační zásobník se používá objekt *navigator* a jeho metoda *push()*. Pro navigaci zpět / odstranění



Obrázek 5.8: Hlavní navigace



Obrázek 5.9: Navigační lišta

aktuální obrazovky z vrcholu zásobníku se používá metoda `pop()`. Při navigaci na novou obrazovku je možné také předat libovolné parametry. Navigační objekt `navigator` je dostupný pouze v kořenové komponentě každé obrazovky. Pokud jej chceme zpřístupnit i jinde, stačí nad definicí třídy přidat anotaci `@withNavigation`.

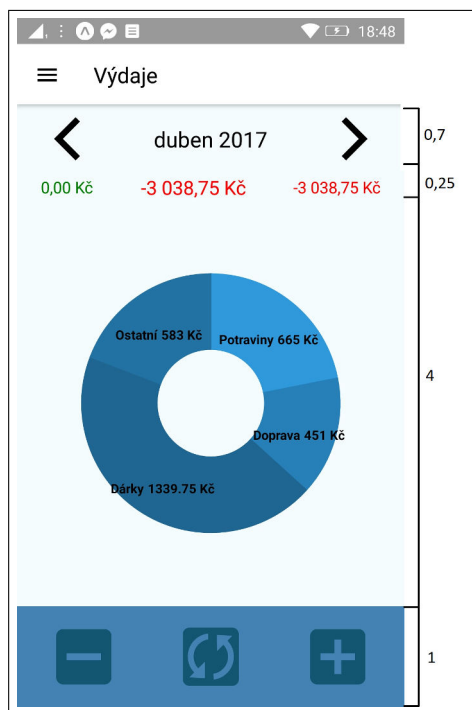
```
1 this.props.navigator.push(Router.getRoute('editAccount', {account
  : this.props.account}));
2 this.props.navigator.pop()
```

5.3.3 Styly

Nastavení vzhledu komponent probíhá přímo v JavaScriptu. V konstantě `styles` se připraví jednotlivé objekty, ekvivalent tříd v CSS. Ty se potom předají komponentám v parametru `style`. Hodnoty `flex` určují relativní velikost vzhledem k ostatním komponentám ve stejném kontejneru.

```
1 const styles = StyleSheet.create({
```

5. REALIZACE



Obrázek 5.10: Relativní velikost komponent

```
2 expensesView: {
3   flex: 1,
4   backgroundColor: '#F5FCFF'
5 },
6 timeNavigation: {
7   flex: 0.7,
8   justifyContent: "center",
9 },
10 content: {
11   flex: 4,
12   justifyContent: "space-around"
13 },
14 addTransactionControls: {
15   flex: 1,
16   backgroundColor: 'steelblue'
17 },
18 balance: {
19   flex: 0.25
20 }
21 };
```

5.3.4 JavaScript a JSX

Do JSX kódu může být s pomocí složených závorek vložen JavaScript. Toho lze využít pro dynamické generování a vkládání komponent.

```

1 <View style={styles.nameAndBalance}>
2   <Text style={styles.name}>{this.props.account.name}</Text>
3   {this.renderBalance(this.props.account.balance, this.props.
      account.currency)}
4 </View>

1 renderBalance(balance, currency){
2   ...
3   return (
4     <FormattedCurrency
5       value={balance}
6       currency={currency}
7       style={balanceStyle} />
8   );
9 }

```

5.3.5 Metody životního cyklu komponenty

V průběhu životního cyklu komponenty je zavoláno několik metod, které umožňují provést daný kód v konkrétním bodě procesu. Toho lze využít např. k načtení dat nebo zaregistrování posluchače pro vybranou událost.

```

1 componentWillMount(){
2   this.loadData();
3 }
4
5 componentDidMount() {
6   this.accountsChangedSubscribtion =
7     ACCOUNTS_DS_EVENT_EMITTER.addListener('accountsChanged', this
      .loadData.bind(this));
8 }
9
10 componentWillUnmount() {
11   this.accountsChangedSubscribtion.remove();
12 }

```

5.3.6 Databáze

V době tvorby aplikace nebyla v Expo podporována pokročilejší datová úložiště. Doporučován byl React Native *AsyncStorage*, u kterého je možné ukládat perzistentně data bez zásahu v nativním kódu. Jde o jednoduché key — value úložiště.[140] Pro snadnější práci s ním jsem využil knihovnu *react-native-db-models*, která umožňuje vytvářet databázový model a pracovat s daty pomocí pokročilejších CRUD operací, jako například vyhledávání podle hodnot. Rozhraní je přímo v jazyce JavaScript. Od verze Expo 16.0.0 lze již využít i *SQLite*.^[165]

5. REALIZACE

```
1 export var DB = {
2   "accounts": new RNDBModel.create_db('accounts'),
3   "transactions": new RNDBModel.create_db('transactions'),
4   "transfers": new RNDBModel.create_db('transfers'),
5   "budgets": new RNDBModel.create_db('budgets'),
6   "rates": new RNDBModel.create_db('rates'),
7   "ratesLastFetch": new RNDBModel.create_db('ratesLastFetch')
8 };

1 DB.accounts.remove_id(account._id, result => {...});

1 DB.transactions.get({category: category}, result => {...});

1 DB.transactions.update({_id: oldTransaction._id}, newTransaction,
   result => {...});
```

5.3.7 HTTP komunikace

Pro získávání dat ze serverů umožňuje React Native použít fetch API.[166] Zde je příklad volání API České spořitelny. V hlavičce HTTP dotazu jsou zaslány WEBApi klíč a autorizační token. Odpověď přijde ve formátu JSON a je posléze načtena do JavaScript objektu. Všechna volání jsou asynchronní.

```
1 response = await fetch(csAccountsURI, {
2   method: 'GET',
3   headers: {
4     'Accept': 'application/json',
5     'WEB-API-key': csApiKey,
6     'Authorization': accessToken
7   }
8 });
9 let responseJson = await response.json();
```

5.3.8 Grafy

Grafy v aplikaci jsou vykresleny s pomocí knihovny react-native-pathjs-charts, která je vystavěna nad knihovnou react-native-svg. Nejprve je třeba nastavit parametry grafu jako jsou rozměry, barva, písmo popisek a pod. Ty se následně spolu s daty předají komponentě pro vybraný graf. Formát dat se liší v závislosti na použitém grafu. Výsledkem je SVG obrázek grafu. Knihovna je teprve v rané fázi vývoje a výsledky se někdy mírně liší na iOS a Android nebo obsahují jiné chyby.[159]

```
1 const options = {
2   margin: {
3     top: 0,
4     left: 0,
5     right: 0,
6     bottom: 0
7   },
```

```
8   width: 300,  
9   height: 300,  
10  color: '#2980B9',  
11  r: 50,  
12  R: 120,  
13  legendPosition: 'topLeft',  
14  label: {  
15    fontFamily: 'Arial',  
16    fontSize: 12,  
17    fontWeight: true,  
18    color: '#000'  
19  }  
20 }  
  
1 <Pie data={this.state.data} options={options} accessorKey="amount  
  " />
```

Závěr

Cílem této práce bylo navrhnout a implementovat mobilní aplikaci pro správu osobních výdajů s možností napojení na API banky. Aplikace by měla umožnit stáhnutí transakčních dat přímo z finančního ústavu a následně je automaticky kategorizovat. Na základě těchto dat by pak měla vykreslovat grafy a jiné přehledy týkající se výdajů a finanční situace uživatele. Aplikace by měla dále umožnit vedení více účtů a to i v různých měnách. Výsledný produkt by měl být spustitelný na zařízeních se systémem Android a iOS.

Věřím, že se mi tohoto cíle podařilo dosáhnout, jak dokládá obsah příloženého DVD. Výsledná aplikace splňuje všechny podmínky vypsane výše. Zároveň jsem se snažil, aby byla uživatelsky co nejpřívětivější a nabízené přehledy byly přínosné. Přestože je aplikace funkční, před jejím ostrým provozem bych jistě provedl řadu úprav a vylepšení. Mezi prvními by bylo zlepšení stability aplikace, která si zatím neumí poradit s některými okrajovými stavy (např. zneplatnění refresh tokenu). Dále by to byla změna perzistentního úložiště na SQLite, které je lépe uzpůsobeno pro práci s používanými daty a v neposlední řadě také vylepšení uživatelského rozhraní. A to jak po vzhledové tak po funkční stránce. Zde by bylo nejlepším řešením nejspíše spolupráce s autory používaných knihoven. Ty jsou stejně jako používaný framework velmi mladé a často nehotové či chybové. Co se delšího výhledu do budoucna týče, vidím největší přínos ve zlepšování stávajících funkcí spíše než v přidávání dalších. Ať je to zlepšení automatické kategorizace (např. rozpoznání výběru z bankomatu) s pomocí učících se a jiných algoritmů, automatická synchronizace s účtem (např. jednou denně) nebo rozšíření o další banky. Dalšími šikovnými rozšířeními by byla záloha dat pro import na nové zařízení, zabezpečení hesle nebo zobrazení přehledů za jiné období než měsíc.

Největším problémem při práci s React Native a Expo pro mě byly neustálé změny, které provází rychlý vývoj těchto technologií a na které je nutné se adaptovat. Spolu s tím jde ruku v ruce i nestabilita přidružených knihoven. Co se samotného programování týká, chvíli mi trvalo, než jsem vstřelil používané vzory a postupy. Před touto prací jsem neměl mnoho zkušeností ani

s JavaScriptem. Nicméně po počátečních prodlevách jsem si zvykl a byl jsem schopen ocenit výhody těchto technologií. Např. vývoj bez nutnosti rekompile po každé změně mě po předchozích zkušenostech s programováním pro Android velmi nadchl. Stejně tak možnost nasazení aplikace na fyzická zařízení během několika desítek vteřin s pomocí klientské aplikace Expo. Díky využívání pouze multiplatformních komponent jsem navíc napsal jen jeden kód a výsledný produkt lze nasadit na Android stejně jako na iOS. Svou zkušenost s React Native a Expo tedy považuji rozhodně za pozitivní.

Součástí práce byla také analýza možností napojení na API bank. U nás tuto službu nabízí zatím pouze dva finanční ústavy a sice Fio banka a Česká spořitelna. Je však zřejmé, že se tato situace brzy změní. Většina, ne-li všechny banky, již minimálně zkoumají možnosti v této oblasti a v roce 2018 by se měla objevit řada nových produktů.

Díky rozšiřování možností programového napojení a podpory ze strany EU bude již v blízké budoucnosti velmi snadné vytvořit aplikaci schopnou komunikovat s vícero bankami. Takové aplikace budou moci kromě zobrazování přehledů jako v mém případě nabízet i mnoho dalších funkcí, ať jsou to online platby, srovnávání produktů od různých bank či větší personalizace bankovních služeb založená na datech, která banka z používané aplikace získá. Možností je mnoho. Takový vývoj dost možná změní přístup zákazníků k bankovním službám a správě vlastních financí a posune bankovníctví velkým krokem dopředu.

Literatura

- [1] KLEINSCHMIT, M.: *Generation Z characteristics: 5 infographics on the Gen Z lifestyle*. *VISIONCRITICAL [online]*, 4. 12. 2015, [cit. 2017-04-30]. Dostupné z: <https://www.visioncritical.com/generation-z-infographics/>
- [2] Wells Fargo: *What millennials and Generation Z expect from their banks and service providers*. *WELLS FARGO [online]*, [cit. 2017-04-30]. Dostupné z: <http://global.wellsfargobank.com/wfinsights-articles-millennials>
- [3] BIRMINGHAM, A.: *Hesitate And Lose. Retail Banks Will Bet On Fintech Startups Says Gartner. Which-50 [online]*, 7. 5. 2016 [cit. 2017-04-30]. Dostupné z: <https://which-50.com/hesitate-lose-retail-banks-will-bet-fintech-startups-says-gartner/>
- [4] PwC: *PSD2 v kostce*. [online], září 2016, [cit. 2017-04-30]. Dostupné z: <http://thesweetsetup.com/apps/favorite-app-managing-personal-finances-budgets/>
- [5] OPENBANKINGPROJECT, bank innovation, API days, Axway: *Banking APIs State of the Market 2016*, 10 2016. Dostupné z: <https://static.openbankproject.com/docs/BANKING-APIS-2016.pdf>
- [6] KORF, M.; OKSMAN, E.: *Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options*. *Salesforce Developers [online]*, 6 2016, [cit. 2017-04-30]. Dostupné z: https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options
- [7] MASSART, F.: *React Native vs Ionic: A Side-by-Side Comparison*. *Codementor Community [online]*, 18. 1. 2017 [cit. 2017-04-30]. Dostupné z: <https://www.codementor.io/fmcorz/react-native-vs-ionic-du1087rsw>

- [8] TIOBE Software: *TIOBE Index for April 2017*. [online], [cit. 2017-04-30]. Dostupné z: <https://www.tiobe.com/tiobe-index/>
- [9] ARORA, S.: *10 Best Hybrid Mobile App UI Frameworks: HTML5, CSS and JS*. Noeticforce [online], 24. 12. 2016 [cit. 2017-04-30]. Dostupné z: <http://noeticforce.com/best-hybrid-mobile-app-ui-frameworks-html5-js-css>
- [10] Priya Phutela: *8 Top JavaScript Frameworks For Mobile App Development in 2017*. Perception Blog [online], 17. 2. 2017 [cit. 2017-04-30]. Dostupné z: <http://blogs.perceptionssystem.com/javascript-frameworks-for-mobile-app-development/>
- [11] Itay HERSKOVITS: *Angular 2 NativeScript Vs. React Native*. Back& [online], 20. 5. 2016 [cit. 2017-04-30]. Dostupné z: <http://blog.backand.com/angular-2-nativescript-vs-react-native/>
- [12] Facebook: *React Native*. [online], [cit. 2017-04-30]. Dostupné z: <https://facebook.github.io/react-native/>
- [13] HINDY, J.: *10 best Android budget apps for money management*. Android Authority [online], únor 2017, [cit. 2017-4-21]. Dostupné z: <http://www.androidauthority.com/best-android-budget-apps-for-money-management-586807/>
- [14] KOMÁRKOVÁ, N.: *TOP3 mobilní aplikace pro správu osobních financí*. poradce.cz [online], září 2016, [cit. 2017-4-21]. Dostupné z: <https://www.poradce.cz/clanky/bankovni-ucty/top3-mobilni-aplikace-pro-spravu-osobnich-financi>
- [15] SMITH, J.: *13 Best Budget Apps for 2017*. GottaBe Mobile [online], únor 2017, [cit. 2017-4-21]. Dostupné z: <http://www.gottabemobile.com/best-budget-apps/>
- [16] TOMANOVA, A.: *Mějte svůj (rodinný) rozpočet pod kontrolou s těmito aplikacemi*. applehelp [online], květen 2016, [cit. 2017-4-21]. Dostupné z: <http://applehelp.cz/2016/05/02/mejte-svuj-rodinny-rozpocet-pod-kontrolou-s-temito-aplikacemi/>
- [17] BERGER, S.: *7 cool personal finance apps for iPhone*. Bankrate [online], ZÁŘÍ 2016, [cit. 2017-4-21]. Dostupné z: <http://www.bankrate.com/finance/personal-finance/best-iphone-personal-finance-apps-1.aspx>
- [18] BISHINEWS: *Expense Manager*. Google Play. [online], [cit. 2017-04-27]. Dostupné z: <https://play.google.com/store/apps/details?id=com.expensemanager>

-
- [19] BISHINEWS: *Expense Manager*. [online], [cit. 2017-04-27]. Dostupné z: <https://sites.google.com/site/expensemgr/>
- [20] BISHINEWS: *Expense Manager*. [program], 2017, verze 3.2. Dostupné z: <https://play.google.com/store/apps/details?id=com.expensemanager>
- [21] MonefyApp: *Monefy - Money Manager*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.monefy.applite>
- [22] MonefyApp: *Monefy*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.monefy.me/>
- [23] MonefyApp: *Monefy*. [program], 2017, verze 1.7.9. Dostupné z: <https://play.google.com/store/apps/details?id=com.monefy.applite>
- [24] Labs, M.: *Mobills: Budget Planner*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=br.com.gerenciadorfinanceiro.controller>
- [25] Labs, M.: *Mobills*. [online], [cit. 2017-04-30]. Dostupné z: <https://www.mobills.com.br/>
- [26] Labs, M.: *Mobills*. [program], 2017, verze 3.0.17.03.19. Dostupné z: <https://play.google.com/store/apps/details?id=br.com.gerenciadorfinanceiro.controller>
- [27] Finsify: *Money Lover - Money Manager*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.bookmark.money>
- [28] Finsify: *Money Lover*. [online], [cit. 2017-04-30]. Dostupné z: <https://moneylover.me/>
- [29] Finsify: *Money Lover*. [program], 2017, verze 3.5.128. Dostupné z: <https://play.google.com/store/apps/details?id=com.bookmark.money>
- [30] Cleveni: *ClevMoney - Personal Finance*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.dencreak.spbook>
- [31] Cleveni: *MClevMoney*. [program], 2017, verze 3.9.9. Dostupné z: <https://play.google.com/store/apps/details?id=com.dencreak.spbook>
- [32] AndroMoney: *AndroMoney (Expense Track)*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.kpmoney.android>

- [33] AndroMoney: *AndroMoney*. [online], [cit. 2017-04-30]. Dostupné z: <https://web.andromoney.com/>
- [34] AndroMoney: *AndroMoney*. [program], 2017, verze 3.0.3. Dostupné z: <https://play.google.com/store/apps/details?id=com.kpmoney.android>
- [35] Realbyte: *Money Manager Expense and Budget*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.realbyteapps.moneymanagerfree>
- [36] Realbyte: *Money Manager Android*. [online], [cit. 2017-04-30]. Dostupné z: <https://moneymanagerandroid.wordpress.com/>
- [37] Realbyte: *Money Manager*. [program], 2017, verze 3.6.23. Dostupné z: <https://play.google.com/store/apps/details?id=com.realbyteapps.moneymanagerfree>
- [38] Shirokova, A.: *Alzex Personal Finance*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.alzex.finance>
- [39] Shirokov, A.: *Alzex Finance*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.alzex.com/>
- [40] Shirokova, A.: *Alzex Personal Finance*. [program], 2017, verze 2.7. Dostupné z: <https://play.google.com/store/apps/details?id=com.alzex.finance>
- [41] Manager, H. E.: *Expense Manager*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=mediavision.handwallet>
- [42] Manager, H. E.: *EXPENSE MANAGER*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.handwallet.com/expensemanager/default.asp>
- [43] Manager, H. E.: *HandWallet*. [program], 2017, verze 5.07. Dostupné z: <https://play.google.com/store/apps/details?id=mediavision.handwallet>
- [44] Apps, H.: *Expense IQ Money Manage*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.handyapps.expenseiq>
- [45] Apps, H.: *Expense IQ MONEY MANAGER*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.expense-iq.com/>

-
- [46] Apps, H.: *Expense IQ*. [program], 2017, verze 2.0.3. Dostupné z: <https://play.google.com/store/apps/details?id=com.handyapps.expenseiq>
- [47] CLEEVIO: *Spendee*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.cleevio.spendee>
- [48] CLEEVIO: *Spendee*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.spendee.com/>
- [49] CLEEVIO: *Spendee*. [program], 2017, verze 3.0.7. Dostupné z: <https://play.google.com/store/apps/details?id=com.cleevio.spendee>
- [50] BudgetBakers: *Wallet - Budget Tracker*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.droid4you.application.wallet>
- [51] BudgetBakers: *Wallet*. [online], [cit. 2017-04-30]. Dostupné z: <http://budgetbakers.com/>
- [52] BudgetBakers: *Wallet*. [program], 2017, verze 4.5.4. Dostupné z: <https://play.google.com/store/apps/details?id=com.droid4you.application.wallet>
- [53] Solutions, F. M.: *Fast Budget - Expense Manager*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.blodhgard.easybudget>
- [54] Solutions, F. M.: *Fast Budget*. [online], [cit. 2017-04-30]. Dostupné z: <https://www.fastbudget.it/>
- [55] Solutions, F. M.: *Fast Budget*. [program], 2017, verze 4.6.4. Dostupné z: <https://play.google.com/store/apps/details?id=com.blodhgard.easybudget>
- [56] Hintersteiner, M.: *Expense Manager*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=at.markushi.expensemanager>
- [57] Hintersteiner, M.: *Expense Manager*. [online], [cit. 2017-04-30]. Dostupné z: <https://markushintersteiner.at/expense-manager.html>
- [58] Hintersteiner, M.: *Expense Manager*. [program], 2015, verze 2.2.4. Dostupné z: <https://play.google.com/store/apps/details?id=at.markushi.expensemanager>

- [59] Tower, M.: *Expense Manager*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.moneytower.expmgr>
- [60] Tower, M.: *Expense Manager*. [program], 2016, verze 1.3.1. Dostupné z: <https://play.google.com/store/apps/details?id=com.moneytower.expmgr>
- [61] Balli: *My Wallet - Expense Manager*. [program], 2016, verze 1.3.1.2. Dostupné z: <https://play.google.com/store/apps/details?id=com.apps.balli.mywallet>
- [62] CASorin: *Expense Manager - Tracker*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.smartexpenditure>
- [63] CASorin: *Expense Manager*. [program], 2017, verze 1.2.4. Dostupné z: <https://play.google.com/store/apps/details?id=com.smartexpenditure>
- [64] Global, W.: *Wally+*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=me.wally.android.plus>
- [65] napkin SAL: *Wally - Smart Personal Finance*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/wally-smart-personal-finance/id610314677>
- [66] Wally: *Wally*. [online], [cit. 2017-04-30]. Dostupné z: <http://wally.me/>
- [67] Global, W.: *Wally+*. [program], 2015, verze 0.9.12.3. Dostupné z: <https://play.google.com/store/apps/details?id=me.wally.android.plus>
- [68] HOLLINS, F.: *MGT12- Wally App Review*. YouTube. [online], 31. 5. 2016 [cit. 2017-04-30]. Dostupné z: <https://www.youtube.com/watch?v=aPZVF2GVf98>
- [69] UG, M.: *MoneyCoach - Personal Finance, Bills and Expenses*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/moneycoach-budget-manage-money/id989642198>
- [70] MoneyCoach: *MoneyCoach App Showcase*. YouTube. [online], 12. 8. 2016 [cit. 2017-04-30]. Kanál uživatele Duuro. Dostupné z: <https://www.youtube.com/watch?v=aPZVF2GVf98>
- [71] Appxy: *Pocket Expense - Personal Finance Assistant*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/pocket-expense-personal-finance-assistant/id424575621>

-
- [72] Appxy: *Pocket Expense*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.appxy.com/pocket-expense/>
- [73] HARTER, A.; WARBER, B.: *Pocket Expense*. YouTube. [online], 8. 11. 2016 [cit. 2017-04-30]. Kanál uživatele Bailee Warber. Dostupné z: <https://www.youtube.com/watch?v=Fy7jtjyagW0>
- [74] Disrapp: *CoinKeeper: spending tracker*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.disrapp.coinkeeper.material>
- [75] Disrapp: *CoinKeeper: spending tracker, money and finance*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/coinkeeper-spending-tracker-money-and-finance/id849747345>
- [76] Disrapp: *CoinKeeper*. [online], [cit. 2017-04-30]. Dostupné z: <https://coinkeeper.me/>
- [77] Disrapp: *CoinKeeper*. [program], 2017, verze 2.2.6. Dostupné z: <https://play.google.com/store/apps/details?id=com.disrapp.coinkeeper.material>
- [78] Riley, M.: *Spending Tracker*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.mhriley.spendingtracker>
- [79] Riley, M.: *Spending Tracker*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/app/spending-tracker/id548615579>
- [80] Riley, M.: *Spending Tracker*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.mhriley.com/spendingtracker/>
- [81] Riley, M.: *Spending Tracker*. [program], 2016, verze 1.6.1. Dostupné z: <https://play.google.com/store/apps/details?id=com.mhriley.spendingtracker>
- [82] Artezio: *Cost Track – personal finance tracker*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/cost-track-personal-finance-tracker/id797459670>
- [83] Artezio: *COST TRACK*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.artezio.com/products/cost-track>
- [84] SilverWiz: *MoneyWiz 2 Personal Finance*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: https://play.google.com/store/apps/details?id=com.moneywiz_2.androidphone_free

- [85] SilverWiz: *MoneyWiz Premium - Personal Finance*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/moneywiz-premium-personal-finance/id1004710719>
- [86] SilverWiz: *MoneyWiz*. [online], [cit. 2017-04-30]. Dostupné z: <http://moneywizapp.com/>
- [87] SilverWiz: *MoneyWiz 2*. [program], 2017, verze 2.5.15. Dostupné z: https://play.google.com/store/apps/details?id=com.moneywiz_2.androidphone_free
- [88] iBear: *Money Pro - Personal Finance, Budget, Bills*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/id918609651>
- [89] iBear: *Money Pro*. [online], [cit. 2017-04-30]. Dostupné z: <http://ibearmoney.com/pro/overview.html>
- [90] BRICHTA, O.: *Finance lite*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/US/app/id633930517>
- [91] AppAdvice: *Finance lite*. AppAdvice. [online], [cit. 2017-04-30]. Dostupné z: <https://appadvice.com/app/finance-lite/633930517>
- [92] DUNG, P.: *Expense Manager - Tracking daily expenses and Saving your money*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/US/app/id895065091>
- [93] AppAdvice: *Expense Manager - Tracking daily expenses and Saving your money*. AppAdvice. [online], [cit. 2017-04-30]. Dostupné z: <https://appadvice.com/app/expense-manager-tracking-daily-expenses-saving-your-money/895065091>
- [94] con socio unico, B. S.: *Live Expenses - Expense tracker - money manager and cash flow*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/app/dovecatalog/id486195690>
- [95] con socio unico, B. S.: *Bonazzi software*. [online], [cit. 2017-04-30]. Dostupné z: http://www.bonazzisoftware.it/liveexpenses_en.aspx
- [96] Toshl: *Toshl Finance Budget and Expense*. Google Play. [online], [cit. 2017-04-30]. Dostupné z: <https://play.google.com/store/apps/details?id=com.thirdframestudios.android.expensoor>
- [97] Toshl: *Toshl Finance - expense tracker and budget manager*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/app/toshl-finance/id921590251>

-
- [98] Toshl: *Toshl Finance*. [online], [cit. 2017-04-30]. Dostupné z: <https://toshl.com/>
- [99] Toshl: *Toshl Finance*. [program], 2017, verze 2.0.16. Dostupné z: <https://play.google.com/store/apps/details?id=com.thirdframestudios.android.expensoor>
- [100] Vieira, D.: *Moneyboard - Personal Finance Manager For Expenses*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/moneyboard-personal-finance-manager-for-expenses/id625887028>
- [101] Vieira, D.: *Moneyboard*. Facebook. [online], [cit. 2017-04-30]. Dostupné z: <https://www.facebook.com/moneyboard/>
- [102] GHASEMPOUR, I.: *Frugi - Personal Finance Manager to Track your Budget, Expenses, Income and Future Reminder*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/frugi-personal-finance-manager-to-track-your-budget/id630382033>
- [103] Appes, S.: *Frugi - Personal finance manager [iPhone] Video review by Stelapps*. YouTube. [online], 29. 4. 2014 [cit. 2017-04-30]. Dostupné z: <https://www.youtube.com/watch?v=L5n8gmBqrtg>
- [104] Starkode: *Expense Center*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/us/app/expense-center/id885316083>
- [105] Holdings, I.: *Banktivity for iPhone - Personal Finance*. iTunes. [online], [cit. 2017-04-30]. Dostupné z: <https://itunes.apple.com/app/apple-store/id919518925>
- [106] Software, I.: *The iPhone's Most Capable Money Management App*. [online], [cit. 2017-04-30]. Dostupné z: https://www.iggsoftware.com/banktivity_for_iphone/
- [107] SCHMITZ, M.: *Banktivity: The best app for managing personal finances and budgets*. THE SWEET SETUP. [online], 18. 10. 2016 [cit. 2017-04-30]. Dostupné z: <http://thesweetsetup.com/apps/favorite-app-managing-personal-finances-budgets/>
- [108] PwC Financial Services Institute: *What is FinTech?* [online], duben 2016, [cit. 2017-04-30]. Dostupné z: <https://www.pwc.com/us/en/financial-services/publications/viewpoints/assets/pwc-fsi-what-is-fintech.pdf>
- [109] Business Insider: *These are the fintech segments most likely to grow in 2017*. Business Insider [online], 14. 3. 2017 [cit. 2017-04-30]. Dostupné z: <http://www.businessinsider.com/these-are-the-fintech-segments-most-likely-to-grow-in-2017-2017-3>

- [110] BOYD, M.: *The Rise of Open Banking Platforms*. *Medium [online]*, 5. 4. 2016 [cit. 2017-04-30]. Dostupné z: <https://medium.com/\spacefactor\@mAPIdays/the-rise-of-open-banking-platforms-a1101915760d>
- [111] FG Forrest: *PRÍMÝ KANÁL*. [online], 2016, [cit. 2017-04-30]. Dostupné z: <https://www.kb.cz/cs/ucty-platby-a-karty/platby/primy-kanal/>
- [112] Česká spořitelna: *MultiCash*. [online], [cit. 2017-04-30]. Dostupné z: http://www.csas.cz/banka/content/inet/internet/cs/PRODUCT_DESCRIPTION_CS_PI01_008983.XML
- [113] ŽŮRKOVÁ, L.: *Elektronické bankovníctví a jeho integrace do účetního programu*. Diplomová práce, Vysoké učení technické v Brně, Fakulta podnikatelská, 2007.
- [114] BULLA, M.: *Elektronické bankovníctví - 2. část. Účetní kavárna [online]*, 2. 4. 2010 [cit. 2017-04-30]. Dostupné z: <http://www.ucetnikavarna.cz/archiv/dokument/doc-d9466v12332-elektronicke-bankovnictvi-2-cast/>
- [115] Měsíc.cz: *Homebanking*. [online], [cit. 2017-04-30]. Dostupné z: <https://www.mesec.cz/bankovni-ucty/prime-bankovnictvi/home-banking/pruvodce/>
- [116] Open Financial Exchange: *About OFX*. [online], [cit. 2017-04-30]. Dostupné z: <http://www.ofx.net/about-ofx.html>
- [117] No Thirst Software: *Direct Connect Banking*. [online], [cit. 2017-04-30]. Dostupné z: <http://nothirst.com/help/moneywell/mac/directConnectBanking.html>
- [118] Intuit, Envestnet: *Open Financial Exchange: Specification 2.2 Public DRAFT #3*. červenec 2016. Dostupné z: <http://www.ofx.net/downloads/OFX%202.2.pdf>
- [119] Raiffeisen Bank: *Nápověda k aplikaci Internetové bankovníctví*. [online], [cit. 2017-04-30]. Dostupné z: https://ibs.rb.cz/IB/help/Help_aplikace.html
- [120] Fio banka: *FIO API BANKOVNICTVÍ*. prosinec 2016, verze 1.5.2. Dostupné z: https://www.fio.cz/docs/cz/API_Bankovnictvi.pdf
- [121] *SMĚRNICE EVROPSKÉHO PARLAMENTU A RADY (EU) 2015/2366 ze dne 25. listopadu 2015 o platebních službách na vnitřním trhu, kterou se mění směrnice 2002/65/ES, 2009/110/ES a 2013/36/EU a nařízení (EU) č. 1093/2010 a zrušuje směrnice*

- 2007/64/ES. 2015. Dostupné z: <http://eur-lex.europa.eu/legal-content/CS/TXT/PDF/?uri=CELEX:32015L2366&from=CS>
- [122] Open Data Institute: *Open Banking Working Group: Terms of Reference*. *Open Data Institute: News [online]*, 2. 10. 2015 [cit. 2017-04-30]. Dostupné z: <https://theodi.org/news/open-banking-working-group-terms-of-reference>
- [123] VEČEŘA, Z.: *Fio banka nabídla API Bankovníctví pro zpracování přijatých plateb*. *Živě [online]*, 7. 11. 2012 [cit. 2017-04-30]. Dostupné z: <http://www.zive.cz/bleskovky/fio-banka-nabidla-api-bankovnictvi-pro-zpracovani-prijatych-plateb/sc-4-a-166202/default.aspx>
- [124] Investujeme.cz: *Česká spořitelna nabídne klientům přístup k otevřeným datům. Elektronickou složenku tak zaplatíte bezpečně a jedním klikem*. *Investujeme.cz [online]*, 30. 10. 2015 [cit. 2017-04-30]. Dostupné z: <http://www.investujeme.cz/tiskove-zpravy/ceska-sporitelna-nabidne-klientum-pristup-k-otevrenym-datam-elektronickou-slozenku-tak-zaplatite-bezpecne-a-jednim-klikem/>
- [125] MESROPYAN, E.: *Top Banking APIs Enabling Access to Aggregated Rich Financial Data. Let's Talk Payments [online]*, 27. 10. 2016 [cit. 2017-04-30]. Dostupné z: <https://letstalkpayments.com/top-banking-apis-enabling-access-to-aggregated-rich-financial-data/>
- [126] Plaid Technologies: *Aggregate transaction data*. [online], [cit. 2017-04-30]. Dostupné z: <https://plaid.com/solutions/transaction-data/>
- [127] Finicity: *Financial Management*. [online], [cit. 2017-04-30]. Dostupné z: <https://www.finicity.com/financial-management/>
- [128] JANÁK, T.: *Automatické párování plateb*. *Superfaktura.cz [online]*, 8. 5. 2015 [cit. 2017-04-30]. Dostupné z: <https://www.superfaktura.cz/blog/automaticke-parovani-plateb/>
- [129] BERÁNEK, J.: *Fintech revoluce pokračuje. Ve jménu PSD2 budou muset banky otevřít svá API*. *Lupa.cz [online]*, 2. 8. 2016 [cit. 2017-04-30]. Dostupné z: <https://www.superfaktura.cz/blog/automaticke-parovani-plateb/>
- [130] Fio banka: *Fio banka nabízí bezpečné automatizované získávání dat z účtů*. [online], 6. 11. 2012 [cit. 2017-04-30]. Dostupné z: <https://www.fio.cz/spolecnost-fio/media/tiskove-zpravy/122142-fio-banka-nabizi-bezpecne-automatizovane-ziskavani-dat-z-uctu>

- [131] Česká národní banka: *Banky a pobočky zahraničních bank dle velikosti (stav ke dni 12.02.2017)*. [online], [cit. 2017-02-12]. Dostupné z: https://apl.cnb.cz/apljerrsdad/JERRS.WEB15.BASIC_LISTINGS_RESPONSE_3?p_lang=cz&p_DATUM=12.02.2017&p_hie=HU&p_rec_per_page=25&p_ses_idx=10
- [132] Česká spořitelna: *API PORTÁL*. [online], [cit. 2017-03-18]. Dostupné z: <https://developers.csas.cz/>
- [133] Česká spořitelna: *Integrace s API v ČR*. Verze 1.0.
- [134] DENNISS, W.; BRADLEY, J.: *OAuth 2.0 for Native Apps: draft-ietf-oauth-native-apps-10*. OAuth Working Group, duben 2017. Dostupné z: <https://tools.ietf.org/html/draft-ietf-oauth-native-apps-10>
- [135] Fio banka: *FIO API BANKOVNICTVÍ*. prosinec 2016, verze 1.5.2. Dostupné z: https://www.fio.cz/docs/cz/API_Bankovnictvi.pdf
- [136] Wikipedia: MoSCoW method — Wikipedia, The Free Encyclopedia. 2017, [Online; accessed 1-May-2017]. Dostupné z: https://en.wikipedia.org/w/index.php?title=MoSCoW_method&oldid=770191060
- [137] MLEJNEK, J.: Analýza a sběr požadavků. Technická zpráva, České vysoké učení technické v Praze, Fakulta informačních technologií, Katedra softwarového inženýrství, Softwarové inženýrství BI-SI1, 2017, [cit. 2017-05-01]. Dostupné z: https://edux.fit.cvut.cz/courses/BI-SI1/_media/lectures/03/03.prednaska.pdf
- [138] ROZANSKI, N.; WOODS, E.: *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Upper Saddle River (New Jersey): Addison-Wesley, druhé vydání, 2011, ISBN 978-0321718334, 678 s.
- [139] Česká národní banka: Kurzy devizového trhu. 2017, [cit. 2017-05-01]. Dostupné z: https://www.cnb.cz/cs/financni_trhy/devizovy_trh/kurzy_devizoveho_trhu/denni_kurz.jsp
- [140] Facebook: AsyncStorage. [cit. 2017-05-01]. Dostupné z: <https://facebook.github.io/react-native/docs/asyncstorage.html>
- [141] OCCHINO, T.: *React.js Conf 2015 Keynote - Introducing React Native*. YouTube. [online], 28. 1.2015 [cit. 2017-04-30]. Kanál uživatele Facebook Developers. Dostupné z: <https://www.youtube.com/watch?v=KVZ-P-ZI6W4>
- [142] Facebook: *React*. [online], [cit. 2017-04-30]. Dostupné z: <https://facebook.github.io/react/>

-
- [143] CHEDEAU, C.: *React.js Conf 2015 Keynote 2 - A Deep Dive into React Native*. YouTube. [online], 29. 1.2015 [cit. 2017-04-30]. Kanál uživatele Facebook Developers. Dostupné z: <https://www.youtube.com/watch?v=7rDsRXj9-cU>
- [144] ANDERSON, C.: *Why You Should Consider React Native For Your Mobile App*. *Smashing Magazine [online]*, 7. 4. 2016 [cit. 2017-04-30]. Dostupné z: <https://www.smashingmagazine.com/2016/04/consider-react-native-mobile-app/>
- [145] Novoda, T.: *React Native: Is it the end of native development? Novoda [online]*, 5. 7. 2016 [cit. 2017-04-30]. Dostupné z: <https://www.novoda.com/blog/is-it-the-end-of-native-development/>
- [146] ELKIN, A.: *Why I'm not a React Native Developer*. *Ariel Elkin's Website [online]*, 15. 2. 2017 [cit. 2017-04-30]. Dostupné z: <https://ariielelkin.github.io/articles/why-im-not-a-react-native-developer.html>
- [147] KOL, T.: *React Native Performance — An Updated Example*. *Hackernoon [online]*, 15. 4. 2017 [cit. 2017-04-30]. Dostupné z: <https://hackernoon.com/react-native-performance-an-updated-example-6516bfde9c5c>
- [148] NARAYAN, A.: *React Native vs Native iOS/Android*. *Course Report [online]*, 25. 1. 2017 [cit. 2017-04-30]. Dostupné z: <https://www.coursereport.com/blog/so-you-want-to-build-a-mobile-app-react-native-vs-native-mobile>
- [149] Expo: *Quick Start*. Verze 16.0.0. Dostupné z: <https://docs.expo.io/versions/v16.0.0/index.html>
- [150] Expo: *Expo Explained in 2 Minutes*. YouTube. [online], 22. 8. 2016 [cit. 2017-04-30]. Kanál uživatele Expo. Dostupné z: <https://www.youtube.com/watch?v=IQI9aU1louMI>
- [151] GRIDER, S.: *Native's Exponent with Charlie Cheever*. *Reactjs News [online]*, 22. 11. 2016 [cit. 2017-04-30]. Dostupné z: <https://reactjsnews.com/react-native-exponent-with-charlie-cheever>
- [152] PERRY, A.: *Introducing Create React Native App*. *React Native Blog [online]*, 13. 3. 2017 [cit. 2017-04-30]. Dostupné z: <https://facebook.github.io/react-native/blog/2017/03/13/introducing-create-react-native-app.html>
- [153] MEREDITH, C.; VATNE, B.; PERRY, A.: *The Future of React Native with Brent Vatne and Adam Perry*. *Software Engineering Daily [online]*, 11. 4. 2017 [cit. 2017-04-30]. Dostupné z:

- <https://softwareengineeringdaily.com/2017/04/11/the-future-of-react-native-with-brent-vatne-and-adam-perry/>
- [154] Expo: *Exposition*. [online]. Dostupné z: <https://blog.expo.io/>
- [155] Node.js Foundation, Joyent: *node.js*. [online]. Dostupné z: <https://nodejs.org/en/>
- [156] MROZEK, J.: *JavaScript na serveru: Patří budoucnost Node.js? Zdroják.cz [online]*, 21. 9. 2012 [cit. 2017-04-30]. Dostupné z: <https://www.zdrojak.cz/clanky/javascript-na-serveru-patri-budoucnost-node-js/>
- [157] PERCIVAL, C.: *STRONGER KEY DERIVATION VIA SEQUENTIAL MEMORY-HARD FUNCTIONS*. [online]. Dostupné z: <http://www.tarsnap.com/scrypt/scrypt.pdf>
- [158] Salesforce: *Heroku*. [online]. Dostupné z: <https://www.heroku.com/>
- [159] MARZOLF, B.: *Android and iOS charts based on react-native-svg and paths-js*. GitHub. [online], [cit. 2017-04-30]. Dostupné z: <https://github.com/capitalone/react-native-pathjs-charts>
- [160] TARADAEV, S.; IVANOVA, T.: *React Native Review: To Use or Not To Use. Waverley [online]*, 7. 12. 2016 [cit. 2017-04-30]. Dostupné z: <https://waverleysoftware.com/blog/react-native-review/>
- [161] CHEN, F.: *Using React Native: One Year Later. Discord [online]*, 7. 6. 2016 [cit. 2017-04-30]. Dostupné z: <https://blog.discordapp.com/using-react-native-one-year-later-91fd5e949933>
- [162] NARUSIS, V.: *Pros and cons of React Native - mobile apps with JavaScript. Devbridge Group [online]*, 28. 4. 2016 [cit. 2017-04-30]. Dostupné z: <https://www.devbridge.com/articles/pros-cons-of-react-native-crash-course/>
- [163] Facebook: *Layout with Flexbox*. [cit. 2017-05-01]. Dostupné z: <https://facebook.github.io/react-native/docs/flexbox.html>
- [164] SKRIPNIK, D.: *Understanding React Native flexbox layout. Medium [online]*, 20. 2. 2016 [cit. 2017-04-30]. Dostupné z: <https://medium.com/the-react-native-log/understanding-react-native-flexbox-layout-7a528200afd4>
- [165] VATNE, B.: *Expo SDK v16.0.0 is now available. Exposition [online]*, 20. 4. 2017 [cit. 2017-04-30]. Dostupné z: <https://blog.expo.io/expo-sdk-v16-0-0-is-now-available-2151d555a580>

- [166] Facebook: Networking. [cit. 2017-05-01]. Dostupné z: <https://facebook.github.io/react-native/docs/network.html>

Použité programy

- [1] Evolus: *Pencil* [software]. [přístup 23. května 2017]. Dostupné z: <http://pencil.evolus.vn/>.
- [2] Expo: *Expo* [software]. [přístup 23. května 2017]. Dostupné z: <https://play.google.com/store/apps/details?id=host.exp.exponent&hl=cs>.
- [3] Expo: *XDE* [software]. [přístup 23. května 2017]. Dostupné z: <https://expo.io/>.
- [4] Facebook: *React Native* [software]. [přístup 23. května 2017]. Dostupné z: <https://facebook.github.io/react-native/>.
- [5] Genymobile: *Genymotion* [software]. [přístup 3. května 2017]. Dostupné z: <https://www.genymotion.com/>.
- [6] GitHub: *Atom* [software]. [přístup 3. května 2017]. Dostupné z: <https://atom.io/>.
- [7] Node.js Foundation: *Node.js* [software]. [přístup 3. května 2017]. Dostupné z: <https://nodejs.org/en/>.
- [8] Oracle: *MySQL Workbench 6.3* [software]. [přístup 23. května 2017]. Dostupné z: <https://dev.mysql.com/downloads/workbench/>.
- [9] Sparx Systems: *Enterprise Architect* [software]. [přístup 23. května 2017]. Dostupné z: <http://www.sparxsystems.com/products/ea/trial/request.html>.

Seznam použitých zkratk

- ABO** Automatizované bankovní operace
- API** Application programming interface
- APK** Android package kit
- BIC** Business identification code
- CAMT** Cash management
- CRUD** Create, read, update, delete
- CSS** Cascading style sheets
- CSV** Comma-separated values
- ČBA** Česká bankovní asociace
- ČNB** Česká národní banka
- ČS** Česká spořitelna
- ECMA** European computer manufacturers association
- EU** Evropská unie
- HTML** HyperText markup language
- HTTP** Hypertext transfer protocol
- IBAN** International bank account number
- ID** Identifikátor
- IIF** Intuit interchange format
- IPA** Iphone application

A. SEZNAM POUŽITÝCH ZKRATEK

- ISIN** International securities identification number
- JSON** JavaScript object notation
- KB** Komerční banka
- MoSCoW** Must have, Should have, Could have, Won't have but would like
- OFX** Open financial exchange
- PC** Personal computer
- PDF** Portable document format
- PSD2** Payment services directive 2
- QFX** Quicken financial exchange
- QIF** Quicken interchange format
- QMTF** Quicken Mac Transfer File
- REST** Representational state transfer
- SBA XML** Slovenská banková asociácia
- SDK** Software development kit
- SEPA** Single euro payments area
- SVG** Scalable vector graphics
- TCP/IP** Transmission control protocol / internet protocol
- URI** Uniform resource identifier
- XDE** Expo development environment
- XLS** Excel spreadsheet
- XML** Extensible markup language

Obsah přiloženého DVD

readme.txt.....	stručný popis obsahu DVD
manual.txt.....	uživatelská příručka
src	
├─ impl.....	zdrojové kódy implementace
│ ├─ expenses-manager-app.....	zdrojové kódy mobilní aplikace
│ └─ expenses-manager-server.....	zdrojové kódy serverové části
└─ thesis.....	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
thesis.pdf.....	text práce ve formátu PDF
wireframes.....	návrhy uživatelského rozhraní