

Aplikace pro správu osobních financí

Diplomová práce

Vedoucí práce:

Ing. Ondřej Popelka, Ph.D.

Bc. Jiří Brada

Brno 2017

Touto cestou bych rád poděkoval vedoucímu práce Ing. Ondřeji Popelkovi, Ph.D. za cenné rady a připomínky. Dále bych chtěl poděkovat svým blízkým, kteří mi svou podporou a pozitivním přístupem vytvořili vhodné prostředí k napsání diplomové práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Aplikace pro správu osobních financí** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 20. května 2017

Abstract

Brada, J. Application for managing personal finances. Bachelor thesis. Brno: Mendel University, 2017.

This Diploma thesis addresses the different problems regarding Czech web applications in relation to managing personal finances. The first part describes the current state of the Czech market with web applications. The next part evaluates synchronization methods of both incomes and costs with internet bankings. These findings are based on executed design and implementation of new web applications which contain an option for synchronization with banking API and better suits the needs of a defined target group. Finally there is a comparison made between monetisation and gamification.

Keywords

Web application, Personal finance, PHP, API, Internet banking

Abstrakt

Brada, J., Aplikace pro správu osobních financí. Diploma thesis. Brno: Mendel University, 2017.

Diplomová práce se věnuje problematice českých webových aplikací ve vztahu ke správě osobních financí. V první části práce je popsán současný stav na českém trhu s webovými aplikacemi. Dále následuje zhodnocení metod synchronizace příjmů i výdajů s internetovým bankovníctvím. Na základě dosažených zjištění je proveden návrh a implementace nové webové aplikace, která obsahuje možnost synchronizace s bankovním API a lépe splňuje požadavky definované cílové skupiny. Na závěr bylo provedeno zhodnocení monetizace a gamifikace.

Klíčová slova

Webová aplikace, Osobní finance, PHP, API, Internetové bankovníctví

Obsah

1	Úvod a cíl práce	15
1.1	Úvod	15
1.2	Cíl práce	15
2	Cílová skupina	16
3	Současný stav	18
3.1	sMoneyBox	18
3.1.1	Funkcionalita	18
3.1.2	Bezpečnost	19
3.1.3	Shrnutí	19
3.2	eÚčty.cz	20
3.2.1	Funkcionalita	20
3.2.2	Bezpečnost	21
3.2.3	Shrnutí	21
3.3	Fakturoid	21
3.3.1	Funkcionalita	21
3.3.2	Bezpečnost	22
3.3.3	Shrnutí	22
3.4	iDoklad	22
3.4.1	Funkcionalita	23
3.4.2	Bezpečnost	24
3.4.3	Shrnutí	24
3.5	Zhodnocení aplikací	24
4	Napojení na internetové bankovníctví	26
4.1	Ruční vložení transakcí do aplikace	26
4.2	Synchronizace pomocí zasílání výpisů z účtu	26
4.3	Synchronizace pomocí bankovního API	27
4.3.1	Neoficiální bankovní API	27

4.3.2	Oficiální bankovní API.....	28
4.3.2.1	Fio Banka	30
4.3.2.2	Česká spořitelna	33
4.4	Zhodnocení možností	36
4.5	Budoucnost bankovního API – PSD2	37
5	Návrh aplikace	39
5.1	Funkční a nefunkční požadavky.....	39
5.1.1	Funkční požadavky.....	39
5.1.2	Nefunkční požadavky	40
5.2	Případy užití	40
5.3	Návrh uživatelského rozhraní	41
5.3.1	Přihlášení uživatele.....	41
5.3.2	Společná kostra rozhraní aplikace – příjmy a výdaje.....	42
5.3.3	Správa závazků a pohledávek.....	43
5.4	Grafický návrh rozhraní	44
5.4.1	Logo.....	44
5.4.2	Typografie	45
5.4.3	Ikony	46
5.4.4	Postup tvorby návrhu	46
5.5	Návrh databáze.....	48
6	Implementace	50
6.1	Použité technologie	50
6.2	Struktura aplikace.....	52
6.3	Přihlášení	53
6.4	Synchronizace s bankou	54
6.5	Kategorizace příjmů a výdajů.....	55
6.6	Správa závazků.....	57
6.6.1	Short Payment Descriptor (SPAYD).....	57
6.6.2	Zadání závazku.....	60
6.6.3	Změna závazku.....	63
6.6.4	Úhrada závazku.....	63

6.7	Správa pohledávek.....	63
6.8	Párování plateb	63
6.9	Šablona HTML.....	63
6.10	Testovací instalace.....	65
7	Možnosti monetizace aplikace	66
7.1	Affiliate marketing	66
7.2	PPC.....	68
7.3	CPM.....	68
7.4	Zpoplatnění aplikace.....	68
7.5	Návrh monetizace aplikace	68
8	Možnosti gamifikace aplikace	70
8.1	Využití gamifikace v existujících aplikacích.....	70
8.1.1	Internetové bankovníctví mBank	70
8.1.2	SaveUP	71
8.1.3	Contango	71
8.1.4	EpicWin	71
8.2	Využití gamifikace v implementované aplikaci	71
9	Závěr	72
10	Literatura	73
11	Seznam obrázků	80
12	Seznam tabulek	81

1 Úvod a cíl práce

1.1 Úvod

Osobní finance je téma, které se nějakým způsobem dotýká každého z nás. Pryč je doba, kdy si šlo vystačit s bankovkami a mincemi. Účet v bance má dnes už více než 4/5 obyvatel České republiky [81]. Není výjimkou, že někteří lidé mají účtů i několik. Lidé si čím dál tím víc zvykají platit platebními kartami v kamenných obchodech, či on-line, kde se objem transakcí za posledních 10 let pětkrát navýšil [20]. Když je do toho třeba myslet na pravidelné měsíční platby za služby jako telefon, internet a sledovat i svoje pohledávky, stává se z tvorby celkového přehledu o financích relativně náročná a zdlouhavá činnost.

Na českém trhu existuje několik aplikací, které se snaží problematiku osobních financí řešit. Bohužel není, dle mého názoru, dostatečně využit funkční i technologický potenciál, který je v současné době dostupný. Trend by měl být v automatizaci, dlouhodobé udržitelnosti a uživatelské přívětivosti. Uživatel by měl mít k dispozici aplikaci, která je přístupná jak z desktopu, tak z mobilního zařízení a která bude sama sledovat příjmy, výdaje, závazky a pohledávky bez nutnosti soustavného ručního zadávání transakcí do systému.

Věřím, že problematiku osobních financí čeká v dohledné době velmi dynamický rozvoj. Důvodem jsou především chystané legislativní změny Evropské unie, které mimo jiné dávají za povinnost bankám umožnit jejich klientům zpřístupnit informace o svých účtech aplikacím třetích stran.

1.2 Cíl práce

Cílem práce je zmapovat a zhodnotit české webové aplikace, které s osobními financemi pracují, a následně navrhnout a implementovat aplikaci vlastní. Aplikace by měla být responzivní a obsahovat možnost získat informace o bankovním účtu uživatele pomocí *API*¹. Aplikace by měla nabízet možnost synchronizovat příjmy a výdaje bez nutnosti explicitního zadávání každé transakce zvlášť a poskytovat funkcionalitu pro sledování závazků a pohledávek. Dále bude provedeno zhodnocení možností monetizace a gamifikace.

¹ API je zkratka anglického sousloví application program interface. Jedná se o rozhraní, které umožňuje programům komunikovat mezi sebou. API definuje postup, jak si vyžádat informace z jiného systému nebo jakým způsobem systému informace podat. Skládá ze dvou na sebe navazujících částí. První část zajišťuje zpracování informací a odpovědi na volání. Druhá část je rozhraní, které poskytuje souhrn metod, pomocí kterých mezi sebou systémy komunikují. [8]

2 Cílová skupina

Pojem cílová skupina je možné specifikovat jako skupinu zákazníků na cílovém trhu, kterou by měla vyvíjená aktivita, marketingová kampaň nebo v tomto případě webová aplikace pro správu osobních financí nejvíce oslovit [15]. Cílová skupina je výsledkem segmentace trhu a lze ji určit dle segmentačních kritérií:

- Demografická kritéria – skupinu rozdělujeme na základě věku, pohlaví, národnosti, vyznání, etnika, velikosti rodiny, příjmu a podobně. [10]
- Geografická kritéria – cílová skupina zákazníků rozdělena na základně svého geografického rozmístění. Jedná se zejména o rozdělení dle kontinentu, státu, města, charakteru krajiny a podobně. [58]
- Psychografická kritéria – skupinu rozdělujeme podle životního stylu, osobních vlastností, nebo příslušnosti ke společenské vrstvě. Kritérium se používá jako doplněk k demografickému nebo geografickému rozdělení a zlepšení porozumění chování cílové skupiny. [29]
- Behaviorální kritéria – jedná se o kritéria, která jsou založená na postoji nebo reakci zákazníků vůči produktu. [49]

Pro účel této práce, zhodnocení současného stavu na českém trhu s webovými aplikacemi pro správu osobních financí, jsem zvolil cílovou skupinu následovně:

- **Geograficky**
 - Stát: Česká republika
- **Demograficky**
 - Věk: 20-34 let

Na základě cílové skupiny nyní přistoupím k vytvoření fiktivních person. Persony si lze představit jako profily konkrétních lidí (uživatelů) – typických představitelů cílové skupiny. V tomto případě se bude jednat o zjednodušené verze persony. [80]

Jaroslav Novák (25 let)

- Dokončuje magisterské studium na vysoké škole s technickým zaměřením. U studia si přivydělává podnikáním jako fyzická osoba.
- Bydlí v Brně.
- Mezi jeho zájmy patří nové technologie, počítače, hudba a dobré jídlo. Pravidelně provozuje sportovní aktivity pro své zdraví.
- Problémy s osobními financemi: Jaroslav má jeden osobní účet a jeden podnikatelský. Dělá mu problém sledovat příjmy a výdaje na obou svých bankovních účtech a kontrolovat splatnosti vydaných faktur zákazníkům v rámci své podnikatelské činnosti.

Eva Šťastná (31 let)

- Pracuje jako asistentka marketingu. Má středoškolské vzdělání s maturitou a nemá žádný další příjem.

- Bydlí v Praze.
- Mezi její zájmy patří zejména móda, sport a dobré jídlo. Eva také ráda sleduje televizní pořady o vaření.
- Problémy s osobními financemi: Eva často zapomíná na splatnost svých účtů za internet a telefon.

Na základě stanovené cílové skupiny a fiktivních person usuzují, že cílová skupina bude potřebovat v oblasti osobních financí řešit následující problémy:

- automatizace evidence příjmů a výdajů z jednoho a více bankovních účtů;
- evidence závazků;
- evidence pohledávek a sledování jejich splatnosti;
- přístup z on-line i z mobilního zařízení.

3 Současný stav

V této kapitole popíši současný stav na českém trhu s webovými aplikacemi, které slouží ke správě osobních financí. Při analýze současného stavu budu brát ohled na předpokládané potřeby definované cílové skupiny z předchozí kapitoly. Aplikace budu hodnotit z pohledu celkové uživatelské přívětivosti, funkcionality a bezpečnosti.

Následující přehled obsahuje dva typy aplikací, které slouží (někdy jen částečně) k organizaci osobních financí:

1. aplikace sloužící k evidenci příjmů a výdajů (sMoneyBox, eÚčty);
2. aplikace sloužící k evidenci pohledávek a závazků (iDoklad, Fakturoid).

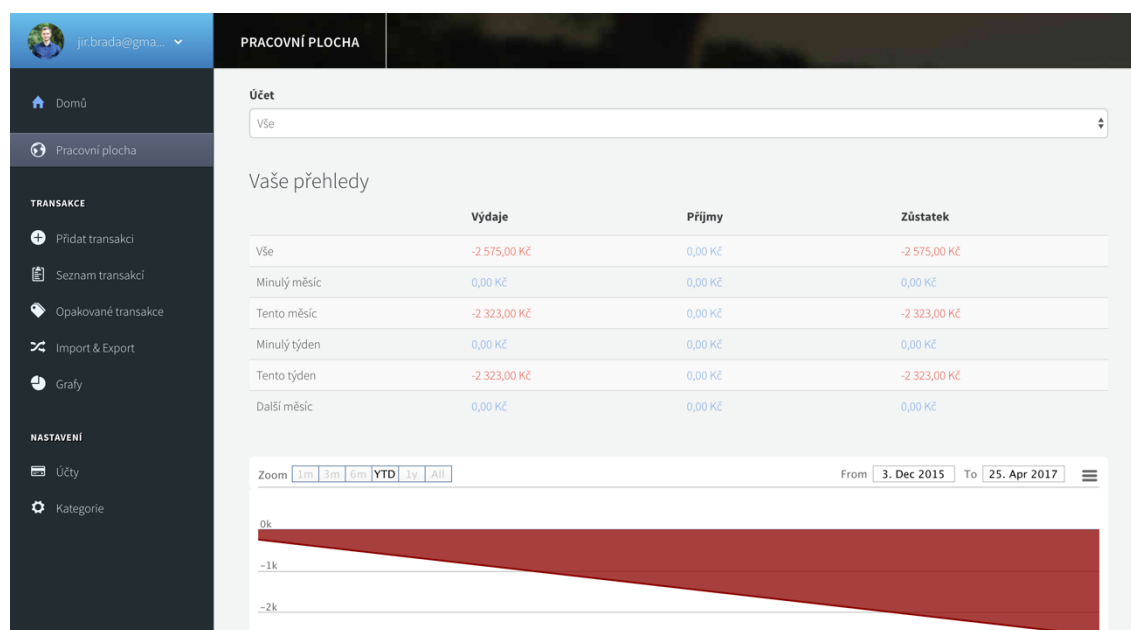
3.1 sMoneyBox

Jedná se o jednoduchou on-line peněženku, která slouží k evidenci příjmů, výdajů a jejich kategorizaci. Aplikace je v současné době dostupná jako webová aplikace a nativní mobilní aplikace pro iOS a Android. Jelikož se v této diplomové práci zaměřuji na webové aplikace, budu v následujícím textu popisovat pouze verzi pro webový prohlížeč.

3.1.1 Funkcionalita

Aplikace sMoneyBox je dostupná na adrese <https://smoneybox.com> a uživatelům je po registraci zdarma přístupná. Pro samotný proces registrace lze využít účtu na sociální síti Facebook, Google+ nebo kombinaci e-mailové adresy a hesla. Během tvorby této diplomové práce bylo možné využít pouze registraci pomocí Google+. Ostatní dvě možnosti nebyly zcela funkční, protože se na konci registrace objevovala neočekávaná chyba.

Po přihlášení do systému je uživateli k dispozici pracovní plocha, která nabízí přehled posledních plateb a tyto informace se zobrazují i v grafu.



Obr. 1 Pracovní plocha aplikace sMoneyBox

Stěžejním prvkem aplikace je účet, do něhož se následně přidávají peněžní transakce. Uživatel má možnost vlastnit neomezený počet účtů a provádět neomezené množství peněžních transakcí, které se dají pomocí funkce „Opakovaná transakce“ automatizovat, pokud se dějí opakovaně.

Vizualizace evidovaných transakcí je možná pomocí volby „Grafy“. K dispozici je zobrazení poměru zůstatků dle kategorií, přehled příjmů a výdajů podle kategorií a vývoj zůstatku v čase. Zajímavou funkcí je hromadný import a export. Podporovány jsou formáty XML a CSV s různým nastavením oddělovacích znaků a kódování.

3.1.2 Bezpečnost

- Webová aplikace využívá k přístupu protokol HTTPS s certifikátem vydaným autoritou Let's Encrypt Authority X3.
- Dvoufaktorová autentizace není možná.
- Provozovatel je registrován u Úřadu pro ochranu osobních údajů.

3.1.3 Shrnutí

Funkcionalita aplikace odpovídá jednoduché on-line peněženke. Na zvážení je trvalá udržitelnost zadávání příjmů a výdajů z toho důvodu, že systém neumožňuje vkládat uskutečněné transakce automaticky. Tento problém částečně řeší možnost hromadného importu, ale stále je nutná akce uživatele, který musí importovaný soubor připravit a manuálně vložit.

3.2 eÚčty.cz

Služba eÚčty.cz je webová aplikace, která se zabývá správou osobních a rodinných financí. Jedná se o volnočasový neziskový projekt, za nímž stojí Jan a Vladimír Lašovi. Služba byla původně vyvíjena pouze pro osobní potřebu tvůrců, ale postupem času se autoři rozhodli ji nabídnout i širší veřejnosti. [59]

3.2.1 Funkcionalita

Aplikace je dostupná na adrese <http://eucty.cz>. Užívání aplikace je zdarma s nutností registrace. Po úspěšném absolvování registračního procesu má uživatel přístup k administračnímu rozhraní, ve kterém provádí všechny operace.

Mezi hlavní funkce aplikace patří přehled účtů, transakce, automatizace, hypotéka a evidence zboží. Obdobně jako u aplikace sMoneyBox je hlavním prvkem účet, do něhož jsou přiřazovány jednotlivé transakce. Každá transakce je přidána buď manuálně, nebo pomocí importů. Služba eÚčty.cz nabízí možnost importovat transakce z celkem 21 zdrojů, mezi něž patří např. Fio Banka, Česká spořitelna, Komerční banka a další. Import neprobíhá automaticky, ale je nutné vygenerovat většinou v internetovém bankovním soubor, který je následně manuálně importovaný do aplikace eÚčty.cz.

Aplikace se snaží pokrýt svými funkcemi další možné problémy osobních financí. Poskytuje jednoduché rozhraní pro evidenci hypoték, nakoupeného zboží (včetně záruční doby), najetých kilometrů a sledování energií.

The screenshot displays the user interface of the eÚčty.cz application. At the top, there is a navigation bar with the logo 'eÚčty.cz' and a menu with options: Přehled, Transakce, Automatizace, Hypotéka, Zboží, Statistiky, Nástroje, and Nastavení. The user is logged in as 'Demo' (Rodina: Nováková).

The main section is titled 'Celkový přehled' (Overall Overview). It features a table of accounts ('Účty') with columns for 'Typ' (Type), 'Částka' (Amount), and 'Měna' (Currency). The table lists several accounts, including 'Hotovost' (Cash) with a balance of -65 978,16 Kč, 'Sporožiro - Petr' (Savings - Petr) with 1 608 525,35 Kč, 'Sporožiro - Lucka' (Savings - Lucka) with -1 033 406,80 Kč, 'ING konto' (ING account) with 230 000,00 Kč, 'Stavební spoření - Petr' (Savings - Petr) with 138 000,00 Kč, 'Eura - Hotovost' (Euro - Cash) with 305,00 €, and 'Dluhy' (Debts) with -45 000,00 Kč.

On the right side, there are several summary cards:

- Celkové součty** (Total Sums): Celkový součet na účtech: 785 312,86 Kč; Kurz Kč/€: 26,795; Součet na odložených účtech: -45 000,00 Kč (není zahrnut v celkovém součtu).
- Podpora vývoje** (Development Support): A message encouraging users to support development by sending financial contributions.
- Auta** (Cars): A card for 'Opel Zafira'.
- Statistiky - rychlý přístup** (Statistics - quick access): A card for 'Celkový stav na účtech' (Overall status on accounts) with a line graph showing trends for 'Tento měsíc' (This month), 'Minulý měsíc' (Last month), 'Tento rok' (This year), and 'Celé období' (Entire period).
- Přehled dle kategorií** (Overview by categories): A card for 'Rozdělení příjmů a výdajů do jednotlivých kategorií' (Distribution of income and expenses into individual categories).

At the bottom left, there is an 'Alarmy' (Alarms) section with a warning icon and the text 'Málo hotovosti' (Low cash), and a 'Upozornění na docházející záruku zboží' (Warning on expiring goods warranty) section with columns for 'Název zboží' (Goods name), 'Datum nákupu' (Purchase date), and 'Záruka do' (Warranty until).

Obr. 2 Rozhraní přihlášeného uživatele aplikace eÚčty.cz

3.2.2 Bezpečnost

Registrace do systému je anonymní a nemusí se povinně uvádět žádný údaj, na jehož základě by bylo možné uživatele identifikovat. Protokol HTTPS je ovšem nedostupný, což považuji za vážný bezpečnostní nedostatek.

3.2.3 Shrnutí

Jan a Vladimír Lašovi vytvořili aplikaci, která evidenci příjmů a výdajů zvládá velmi dobře. Podporuje i hromadný, ale bohužel ne automatizovaný export z banky. Aplikace obsahuje také další zajímavou funkcionalitu, jako je např. sledování počtu ujetých kilometrů nebo sledování energií. Z mého pohledu je zásadním problémem, obdobně jako u výše popisované aplikace sMoneyBox, nutnost ručního zadávání všech hodnot. Možnost evidovat závazky a pohledávky aplikace neposkytuje.

3.3 Fakturoid

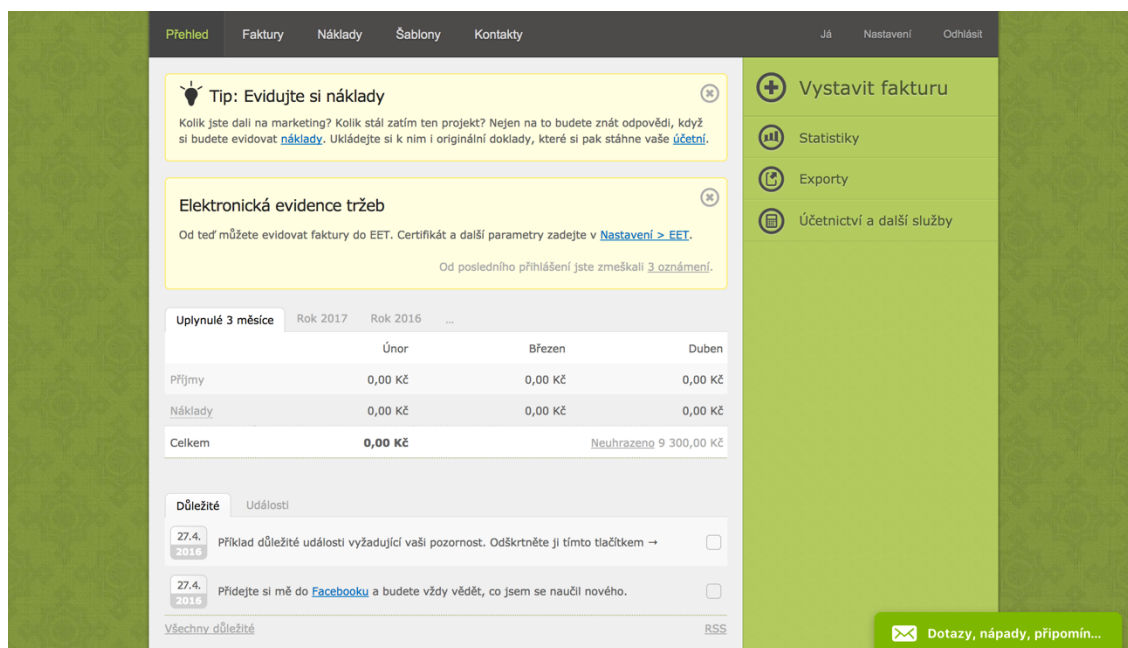
Aplikace Fakturoid, za níž stojí podnikatelé Lukáš Konratovský a Jan Korbel, zprvu sloužila pro vystavování faktur pro drobné živnostníky. Postupně se její funkce rozšiřovaly a dnes se snaží poskytovat ucelený servis pro živnostníky i firmy ve formě vystavování faktur s evidencí jejich úhrady a možnou automatizací souvisejících úkonů.

3.3.1 Funkcionalita

Fakturoid je přístupný na adrese <https://www.fakturoid.cz/> a registrace i základní funkcionalita jsou zdarma. Po provedené registraci získá uživatel přístup do administračního rozhraní <https://app.fakturoid.cz>.

Základní funkcí aplikace je vystavování faktur a jejich evidence. Proces vystavení faktury je intuitivní a rychlý a na vystavenou fakturu jsou navázány další doplňkové funkce. Uživatel může zadat platbu do systému EET nebo díky synchronizaci aplikace s bankovním účtem sledovat splatnost faktur. Dle dostupných informací na webových stránkách umí Fakturoid párovat platby s Komerční bankou, Fio bankou, ČSOB, Reiffeisen bankou, Air bankou a dalšími. Využívá k tomu elektronické bankovní výpisy, které jsou zasílány bankou na speciální e-mailovou adresu aplikace. Jedná se fakticky o **možnost správy pohledávek**. Aplikace také poskytuje možnost nastavení e-mailových upozornění pro příjemce faktury v předem stanovených případech. Jde o vystavení nové faktury, upomínku splatnosti, zaplacení proformy a poděkování za platbu. Podobu těchto zpráv může uživatel specifikovat v sekci „Nastavení e-mailů“.

Druhá část funkcionality se věnuje zadávání nákladů. Jedná se vlastně o **možnost evidence závazků**. Systém zadávání nákladů je obdobný jako u vytváření faktur.



Obr. 3 Rozhraní přihlášeného uživatele aplikace Fakturoid

3.3.2 Bezpečnost

- Aplikace je přístupná po zadání přihlašovacího jména a hesla. Při zapomenutí hesla je možnost jej obnovit pomocí e-mailové adresy zadané při registraci uživatelského účtu.
- Dvoufaktorová autentizace není možná.
- Aplikace podporuje protokol HTTPS.

3.3.3 Shrnutí

Fakturoid se z původně jednoduché aplikace rozrostl do podoby, jejíž velkou část pokrývá administrativa, spojená s vystavováním faktur pro živnostníky a malé firmy. Na rozdíl od předchozích dvou popisovaných aplikací se jedná o komerční projekt a vůbec se nezaměřuje na kategorizaci příjmů a výdajů.

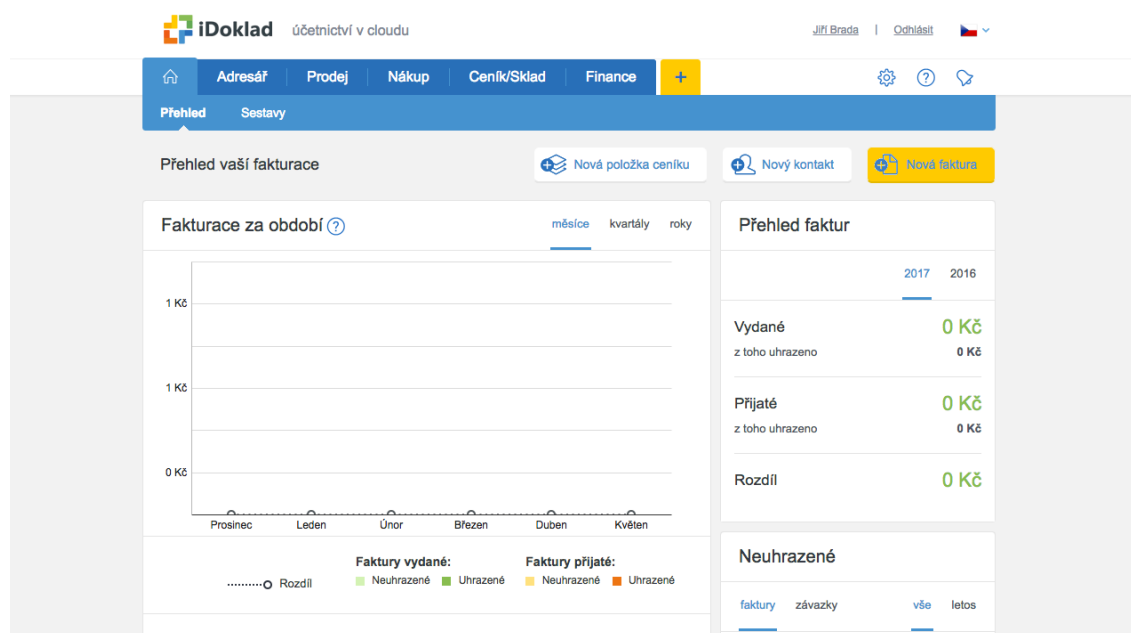
3.4 iDoklad

Za aplikací iDoklad stojí akciová společnost Cíglér Software. iDoklad je funkčně velmi podobná aplikace, jako je Fakturoid. Aplikace se zaměřuje na živnostníky a malé firmy, kterým poskytuje ucelenou funkcionalitu pro správu přijatých a vydaných faktur. Nezapomíná ani na párování úhrad s bankou.

3.4.1 Funkcionalita

Aplikace je dostupná na adrese <https://www.idoklad.cz/>. Registrace je zdarma, stejně jako veškeré funkce. Stěžejní funkcionalitou je tvorba a správa faktur, a to jak vydaných, tak přijatých.

Uživatelské rozhraní je jednoduché a přímočaré. Vytvořit vydanou fakturu je otázkou pár kliknutí. Zhotoví se na základě zadání odběratele, které je možné získat ze serveru Ares², a zanesením položek faktury.



Obr. 4 Rozhraní aplikace iDoklad

Aplikace také umožňuje evidovat přijaté faktury – závazky. Proces tvorby takové faktury je obdobný jako u faktury vydané. U závazků a pohledávek lze sledovat i jejich úhradu. Úhradu faktury lze provést manuálním přidáním v systému nebo synchronizací aplikace s bankou.

Synchronizace plateb s bankou probíhá prostřednictvím bankovních výpisů zasílaných na speciální e-mailovou adresu aplikace. Systém z těchto zpráv rozpozná platby a přidá úhrady přijatých nebo vydaných faktur. iDoklad v současnosti podporuje synchronizaci se šesti českými bankami (ČSOB, Fio, Raiffeisen bank, Komerční banka, Air bank a Sberbank). Dále umí pracovat se čtyřmi bankami ze Slovenska (ČSOB, Fio, Tatra banka, VÚB banka).

Mezi další nabízenou funkcionalitu patří např. ceník a skladové pohyby.

² Aplikace ministerstva financí, která souhrnně zpřístupňuje údaje z informačních systémů veřejné zprávy o ekonomických subjektech. [9]

3.4.2 Bezpečnost

- Přihlášení do systému je možné na základě uživatelského jména a hesla.
- Dvoufaktorová autentizace není podporována.
- Aplikace pracuje pouze na protokolu HTTPS se SSL certifikátem vydaným společností GeoTrust Inc.

3.4.3 Shrnutí

iDoklad poskytuje možnost vytvářet a spravovat faktury vydané i přijaté. Tato funkcionality je doplněna o možnost přidání úhrady, a to buď manuálně, nebo synchronizací s bankovním účtem prostřednictvím bankovních výpisů. Aplikace neumožňuje správu a kategorizaci příjmů a výdajů. Velkou výhodou aplikace je to, že je poskytována zdarma, což při kvalitě aplikace poskytuje značnou výhodou oproti konkurenci.

3.5 Zhodnocení aplikací

Všechny definované **požadavky** cílové skupiny – poskytnout přehled o příjmech, výdajích, závazcích a pohledávkách – **nesplňuje žádná aplikace**. Je možné však narazit na aplikace, které vyhovují podmínkám alespoň částečně.

U správy příjmů a výdajů není současný stav na českém trhu s webovými aplikacemi příliš povzbudivý. Aplikace jako Moře financí, Futu nebo Netina jsou již nefunkční a v době zhotovování této práce byly k dispozici pouze aplikace dvě, a to sMoneyBox a eÚčty. Obě aplikace nabízí velmi podobou funkcionality a ovládání. Hlavním prvkem jsou transakce, které se manuálně přidávají k vybranému účtu. Tímto způsobem lze zapsat provedené příjmy, výdaje a následně zobrazit různé statistiky.

Největším problémem je nutnost trvalého manuálního zadávání transakcí. Částečně tento problém řeší možnost importovat většinou CSV nebo XML soubor s transakcemi. Stále je však nutná explicitní akce uživatele. Základní funkcionality rozšiřuje aplikace eÚčty, která přidává další možnosti např. správu hypoték, ujetých kilometrů nebo vodné a stočné. Z bezpečnostního hlediska má každá aplikace své nedostatky. Zejména absenci protokolu HTTPS u aplikace eÚčty považuji za velkou slabinu.

U správy závazků a pohledávek je situace o poznání lepší. Vzhledem k tomu, že zájem spravovat zejména pohledávky mají už z principu svého fungování hlavně podnikatelé, kteří jsou spíše ochotni do těchto služeb investovat, existuje spousta aplikací, které se tento problém snaží řešit. Společným jmenovatelem těchto aplikací je vytvoření pohledávky ve formě faktury. Na tuto funkci jsou následně navázány funkce další, které se aplikaci od aplikace zlepšují.

Pro potřeby zhodnocení a další tvorbu této práce jsem zvolil dvě aplikace, s nimiž mám vlastní zkušenosti na základě aktivního užívání – Fakturoid a iDoklad.

Z ohledu nabízené funkcionality jsou tyto aplikace více méně totožné. U každé aplikace lze vytvořit pohledávku ve formě faktury i závazek a lze sledovat, zda jsou

položky splaceny synchronizací s bankovním účtem (automatickým zasíláním výpisů z banky). Použitá metoda synchronizace má tu nevýhodu, že neumožňuje aplikaci přístup k bankovnímu účtu v reálném čase, protože je závislá na obdrženém e-mailu z internetového bankovníctví. Může tak docházet k nekonzistenci a zastaralosti údajů o platbách.

Výše zmíněná totožnost funkcionality je relativní s přihlédnutím na cenovou politiku obou zmíněných aplikací. iDoklad je celý zdarma a Fakturoid poskytuje zdarma jen omezenou verzi, která slouží spíše k vyzkoušení možností aplikace. Další tarify se pohybují od 160-640 Kč za měsíc bez DPH. Z bezpečnostního hlediska používají obě aplikace protokol HTTPS, přihlášení prostřednictvím uživatelského jména (e-mailu) a hesla. Dvoufaktorová autentizace není možná.

4 Napojení na internetové bankovníctví

Než přistoupím k návrhu aplikace, považuji za nutné, na základě stanovených požadavků na systém, zhodnotit metody komunikace s internetovým bankovníctvím. Z analýzy současného stavu, která byla provedena v kapitole 3, vyplývá, že jsou v praxi používány dva způsoby, jak napojení provést, a to ruční vložení transakcí do aplikace a přeposlání výpisu z účtu na speciální e-mailovou adresu aplikace.

4.1 Ruční vložení transakcí do aplikace

Metoda ručního vkládání transakcí je použita v aplikacích eÚčty a sMoneyBox. Princip je následující:

1. **Zapsání informace o platební transakci** – Po provedení platební transakce si musí uživatel zapamatovat částku platby a název, z jakého bankovního účtu byla transakce provedena.
2. **Zadání transakce do aplikace** – Jakmile bude mít uživatel přístup k počítači s internetovým připojením, ručně zadá zaznamenané hodnoty transakce do aplikace.

Hlavní problém tohoto principu je především v nutnosti provádět definované úkony kontinuálně, pokud mají být informace v aplikaci aktuální. Vyžadovat součinnost uživatele při každém provedení transakce je nekomfortní a dlouhodobě neudržitelné.

4.2 Synchronizace pomocí zasílání výpisů z účtu

Druhá zmíněná metoda, jak zadat provedené transakce do aplikace, je metoda formou zasílání výpisů z účtu. V současnosti službu nabízí většina největších tuzemských bank. Jedná se např. o banky: ČSOB, Fio, Raiffeisenbank, Komerční banka, Air bank a Sberbank [40]. Postup je z velké části podobný u všech bank a aplikací. Pro konkrétní ukázkou jsem zvolil postup synchronizace aplikace iDoklad a Fio banky.

1. **Vygenerování e-mailové adresy** – Po přihlášení do aplikace iDoklad je třeba vygenerovat unikátní e-mailovou adresu pro zasílání elektronických výpisů. Lze tak učinit v sekci „Nastavení -> Banka“ pod volbou „Automatické párování plateb“.
2. **Nastavení upozornění v internetovém bankovníctví** – Fio banka ve webovém rozhraní svého internetového bankovníctví umožňuje prostřednictvím SMS nebo e-mailové zprávy zasílat upozornění při pohybu na bankovním účtu. Pro synchronizaci s aplikací iDoklad je nutné zvolit e-mail a zadat e-mailovou adresu, která byla vygenerována v předchozím kroku.

Po každé provedené transakci na bankovním účtu odešle internetové bankovníctví informaci na zadanou e-mailovou adresu. Aplikace údaje zpracuje a zadá transakci do své databáze.

Tento přístup eliminuje hlavní nedostatek minulého přístupu, a to nutnost zadávat každou transakci ručně. Nevýhodou může být pasivní role aplikace – musí sama počkat, až přijde e-mail, protože zasílání výpisů má v režii banka. Pokud aplikace nebude z nějakého důvodu schopna e-mail přijmout, dojde tím ke ztrátě informace o transakci a je nutná explicitní akce uživatele k nápravě tohoto stavu.

4.3 Synchronizace pomocí bankovního API

Bankovní *API* je rozhraní, pomocí něhož lze komunikovat s online bankovním systémem. Třetí strana tak získá přístup k údajům klienta bankovní instituce. Typicky se může jednat o přístup k informacím o pohybu na účtu, výpisy z účtu nebo zadávání platebních a inkasních příkazů. Přístup k těmto údajům je možný pouze s výslovným souhlasem klienta, který v rozhraní svého internetového bankovníctví souhlas udělí.[25]

Mezi hlavní výhody bankovního *API* patří hlavně možnost přístupu třetí strany k datům bankovního systému, která byla dříve obtížně přístupná nebo vůbec nepřístupná. [1] Oproti výše zmíněným metodám synchronizace eliminuje nutnost uživatele exportovat a importovat výpisy ručně nebo spoléhat na doručenou notifikaci o provedené transakci e-mailem. Otevírá se tak spousta možností pro vývojáře třetích stran a lze se jen dohadovat, k jakým inovacím tato možnost povede.

V současnosti můžeme narazit na dva typy bankovního *API*. Jedná se o rozhraní neoficiální, které poskytují třetí strany, nebo oficiální, které je poskytováno přímo bankovní institucí, pro niž je určeno.

4.3.1 Neoficiální bankovní API

Jako neoficiální bankovní *API* bude v tomto textu označováno takové rozhraní, které se vyznačuje tím, že není vydáno a podporováno bankovní institucí, ale spravuje ho třetí osoba. Jedním z neoficiálních *API* je *mBank HTTP parser* od vývojáře Martina Zvaríka. Toto *API* poskytuje možnost pracovat s účtem u mBank a provádět základní operace pro práci s účtem [53]:

3. přihlášení k uživatelskému účtu;
4. výpis bankovních účtů a jejich zůstatky;
5. formátovaný výpis transakcí;
6. export transakcí ve formátu CSV, ABO a PDF;
7. výpis seznamu oblíbených příjemců z adresáře;
8. provedení platby.

API je vytvořeno pomocí programovacího jazyku *PHP*. Skript simuluje průchod internetovým bankovníctvím přes web. Následující ukázka kódu ilustruje, jakým způsobem lze získat seznam vedených účtů.

```
<?php
require 'mbank_api.php';

$mb = new mBank("vase_prihlasovaci_cislo", "vase_heslo");
$mb->nastavProfil(1);
$accounts = $mb->ucty();

foreach ($accounts as $account) {
    echo $account;
}
```

V první části ukázkového kódu je vyžádán soubor se zdrojovým kódem k *API*. Následně je vytvořena instance třídy *mBank* s přihlašovacími údaji v konstruktoru. Po úspěšném přihlášení je možno vypisovat seznam vedených bankovních účtů. [53]

Používání jakéhokoli neoficiálního *API* není možné obecně doporučit, zvláště z toho důvodu, že při jakékoli změně struktury internetového bankovníctví nastává možnost omezení funkčnosti *API*. Uživatel tak spoléhá na dodavatele *API*, že pravidelně sleduje změny a implementuje je do systému. Dalším možným problémem může být otázka bezpečnosti – uživatel vkládá své přihlašovací údaje přímo do rozhraní, o němž vlastně neví, jakým způsobem s nimi dodavatel *API* naloží, a tak vyvstává možnost jejich zneužití.

4.3.2 Oficiální bankovní *API*

Jako oficiální bankovní *API* bude označováno takové rozhraní, které je poskytováno přímo bankovní institucí. Bankovní instituce, které otevřely své *API* veřejnosti, se v současnosti vyskytují na českém trhu pouze dvě. Jedná se o *Českou spořitelnu* a *Fio banku*.

REST *API*

Obě dvě v současnosti dostupná *API* fungují na principu REST *API*. Jedná se o zkratku slov Representation State Transfer. Princip REST *API* byl představen v disertační práci Roye Fieldinga – jednoho z vedoucích představitelů specifikace *HTTP*³. [4] Je to architektura, která je orientovaná hlavně na data a na operace s nimi (zejména *CRUD*⁴).

³ *HTTP* je zkratka pro Hypertext Transfer Protokol. Jedná se o protokol pro výměnu hypertextových dokumentů. Protokol je nestavový a funguje na principu dotazu a odpovědi. [33]

⁴ *CRUD* je zkratka anglických slov: vytvořit, získat, změnit a smazat (create, retrieve, update a delete). Odkazuje na čtyři hlavní funkce, které jsou implementovány v databázových aplikacích. [17]

REST API můžeme rozdělit do čtyř navazujících vrstev[73]:

- **Nultá úroveň** – obsahuje samotný přenos dat. I když je nejpoužívanější metodou přenosu protokol HTTP, REST se na ně přímo neváže. Proto je teoreticky možné použít i jinou metodu přenosu.
- **První úroveň (zdroje)** – všechny požadavky se nemusí posílat pouze na jeden centrální bod, ale ke každému zdroji je přiřazen právě jeden koncový bod, kterého lze dosáhnout. Např. pro získání seznamu článků je možno použít `GET /articles` nebo použijeme `GET /articles/1/comments` pro získání seznamu komentářů k jednomu danému článku.
- **Druhá úroveň**
 - **HTTP verbs** – jedná se o seznam akcí, které je možné provést. Nejpoužívanější je pravděpodobně metoda `GET`, která slouží k získávání dat. Níže uvedené metody jsou výčtem nejčastěji užívaných. Použití v jednotlivých API se může lišit, tzn. API může implementovat 0 až N metod.
 - `GET` – získání dat
 - `POST` – vytvoření
 - `PUT` – úpravy
 - `DELETE` – smazání
 - `PATCH` – částečné úpravy
 - **Stavové kódy** – v REST API se používají HTTP stavové kódy. Mezi nejpoužívanější patří:
 - 200 OK – vše proběhlo v pořádku;
 - 201 Created – vytvoření nového záznamu;
 - 204 No Content – vše proběhlo v pořádku, ale odpověď serveru byla prázdná;
 - 304 No Modified – od posledního požadavku nebyl změněn obsah;
 - 400 Bad Request – požadavek je špatný, nečitelný;
 - 401 Unauthorized – klient není autorizován;
 - 403 Forbidden – k vyžádanému obsahu nemá klient přístup;
 - 404 Not Found – nenalezeno;
 - 405 Method Not Allowed – metoda pro zdroj není dostupná (Např.: je možné číst články `GET /articles`, ale není možné je smazat `DELETE /articles`.);
 - 429 Too-Many Requests – klient překročil počet požadavků za časové období.
- **Třetí úroveň (Hypermedia controls)** – každý koncový bod poskytuje informace i o dalších koncových bodech, které s ním souvisí. V praxi stačí znát základní koncový bod a informaci o dalších bodech je možné získat právě od něj. Je tím docílena nezávislost na tvaru adres, které se mohou měnit za běhu aplikace.

4.3.2.1 Fio Banka

Fio banka získala bankovní licenci roku 2010. Jedná se o ryze českou banku, která si zakládá na tom, že běžné bankovní služby by měly být bez poplatku. [7] Zároveň se stala v roce 2012 první českou bankou, která zpřístupnila své API veřejnosti. V té době se jednalo pouze o základní funkcionalitu, prostřednictvím které bylo možno stahovat pohyby účtu za dané období, oficiální výpisy z účtu nebo seznam pohybů na účtu od posledního stažení. [27] V současné době je API nadále rozšiřováno a základní funkcionalitu doplnila např. možnost zadání příkazu k úhradě. [7]

Popis API

API Fio banky funguje na principu REST API. Základní prerekvizitou k používání API je získání 64 znakového autorizačního tokenu. Protože se jedná o nestavovou komunikaci, je token součástí každého požadavku. Každý vygenerovaný token se svou platností vztahuje pouze k účtu, pro který byl vygenerován, a je možno zvolit mezi dvěma různými typy oprávnění:

1. **Sledování účtu** – získaný token umožňuje pouze výpis informací.
2. **Sledování účtu a zadávání platebních příkazů** – token poskytuje oprávnění pro využívání veškeré funkcionality API. Je možné vypisovat informace i zadávat platební příkazy.

Token lze vygenerovat v internetovém bankovníctví, a může tak učinit majitel účtu nebo osoba, která má odpovídající práva. Konkrétně se token vytváří po přihlášení na obrazovce „Nastavení“ v záložce *API*. Požadavek na vytvoření tokenu podléhá standardní autorizaci (SMS kód nebo elektronický klíč) a po 5 minutách je přístup aktivní. *API* nevyžaduje aktivní přihlášení do internetového bankovníctví. Systém provádí autorizaci výhradně na základě tokenu. Komunikace probíhá pomocí protokolu HTTPS s minimálně 128bitovým šifrováním. Z *HTTP verbs* jsou využity metody *GET* a *POST*. [26]

Metoda *GET* slouží k získání dat:

- stažení pohybů na účtu za dané období;
- stažení oficiálních výpisů na účtu;
- stažení pohybu na účtu od posledního stažení.

V následujícím příkladu uvedu, jakým způsobem lze získat pohyby na bankovním účtu za určené období, konkrétně od 1. září 2016 do 20. října 2016.

```
https://www.fio.cz/ib_api/rest/periods/aGEMQB9Idh35fh1g51h3ekkQwyG1Q/  
2016-09-01/2016-10-20/transactions.xml
```

V první části požadavku je uvedena adresa API, k níž je přistupováno. Je uvozena názvem protokolu – HTTPS⁵ (https://www.fio.cz/ib_api/rest/periods).

Další částí je autorizační token (aGEMQB9Idh35fh1g51h3ekkQwyG1Q), následovaný počátečním datem stahovaného výpisu ve formátu rrrr-mm-dd a konečným datem stahovaného výpisu ve stejném formátu rrrr-mm-dd (2016-09-01/2016-10-20). Poslední částí požadavku je specifikováno, jaký formát bude odpověď serveru mít. Konkrétně u pohybů na účtu se může jednat o formáty CSV, GPC, HTML, JSON, OFX, FIO XML.[26]

Metoda **POST** slouží k podání platebních příkazů:

- platba do České republiky;
- europlatba;
- zahraniční platba;
- inkaso.

Zadání platby se provede metodou POST na adresu API, která je uvozena názvem protokolu – HTTPS (https://www.fio.cz/ib_api/rest/import/). U plateb zadáných tímto způsobem je nutno explicitně vyvolat autorizaci požadavku v internetovém bankovníctví uživatelem – konkrétně se jedná o stránku „Příkazy k podpisu“. Po odsouhlasení bankovního příkazu bude platba autorizována např. prostřednictvím SMS zprávy.

V následujícím textu budu popisovat příklad zadání platby na účet v České republice. Pro demonstrační odeslání požadavku metodou POST využiji PHP knihovnu Guzzle⁶. Požadavek se odesílá jako multipart/form-data s parametry token, type, file a lng.

⁵ HTTPS je zkratka slov Hypertext Transfer Protocol Secure. Jedná se o protokol, který používá protokol HTTP a vytváří šifrované spojení. [33]

⁶ Guzzle je HTTP klient v jazyce PHP, který usnadňuje zasílání HTTP požadavků a integraci s webovými službami. [30]

Tab. 1 Souhrn parametrů požadavku při zadávání platebního příkazu u Fio banky

Parametr	Stav	Hodnoty	Popis
token	povinný		token vygenerovaný v internetovém bankovníctví
type	povinný	abo xml pain001_xml pain008_xml	
file	povinný		datový soubor
lng	volitelný	cs sk en	jazyk odpovědi serveru

Zdroj: Dokumentace Fio API [26]

Pokud bude zvolený typ požadavku „xml“, je nutné do parametru „file“ přiložit datový soubor ve formátu XML (file.xml). Struktura souboru je následující:

```
<?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>
```

Po provedení následujícího kódu bude odeslán platební příkaz, který je specifikovaný výše.


```
<?php

$client = new \GuzzleHttp\Client();
$response = $client->request('POST',
    'https://www.fio.cz/ib_api/rest/import/', [
        'multipart' => [
            [
                'name' => 'type',
                'contents' => 'xml'
            ],
            [
                'name' => 'token',
                'contents' => 'aGEMQB9Idh35fh1g51h3ekkQwyGlQ'
            ],
            [
                'name' => 'lng',
                'contents' => 'cs'
            ],
            [
                'name' => 'file',
                'contents' => fopen('file.xml', 'r')
            ],
        ]
    ]);

echo $response->getBody();
```

Více informací o fungování API je dostupné v PDF souboru s dokumentací na adrese http://www.fio.cz/docs/cz/API_Bankovnictvi.pdf.

Práce s dokumentací je jednoduchá, přímočará a účelná. Obsahuje popis veškeré funkcionality API a je doplněna o příklady.

4.3.2.2 Česká spořitelna

V říjnu 2015 zpřístupnila API svého internetového bankovníctví druhá tuzemská banka, Česká spořitelna. [19]

Prostřednictvím svého portálu <https://developers.csas.cz/> banka představuje API pro široké spektrum svých služeb [22]:

- **Netbanking** – poskytuje přístup ke klientským produktům a jejich aktuálnímu stavu;
- **Corporate** – poskytuje přístup k účtům firemních zákazníků;
- **Kurzovní lístky** – poskytuje rozhraní pro aktuální kurzy měn s možností zobrazení historie až 2 roky;
- **Transparentní účty** – poskytuje rozhraní pro přístup k transparentním účtům;
- **Pobočky a bankomaty** – nabízí seznam všech poboček a bankomatů České spořitelny a umožňuje zobrazit i detailní informace, jako je otevírací doba nebo seznam poskytovaných služeb;
- **Hypocalc** – poskytuje rozhraní pro základní hypotéční kalkulačku.

V této práci se budu zabývat převážně částí *API* „Netbanking“ z toho důvodu, že právě přístup ke klientským produktům je potřebný při potřebě synchronizovat transakce u aplikací pro osobní bankovníctví.

Popis API

API České spořitelny funguje podobně, jako tomu je u FIO banky, na principu *REST API*. Autorizace je však prováděna mírně odlišně. Shodnou vlastností je přítomnost autorizačního tokenu při odesílání každého požadavku. Česká spořitelna navíc využívá funkcionalitu *API* klíče, který je unikátní pro každou aplikaci používající *API*. Server ověří, zda má aplikace identifikovaná klíčem přístup k požadovanému zdroji, či nikoli. Tento *API* klíč je přítomný v každém požadavku. Získat ho lze vyplněním formuláře na webové stránce *API* České spořitelny.

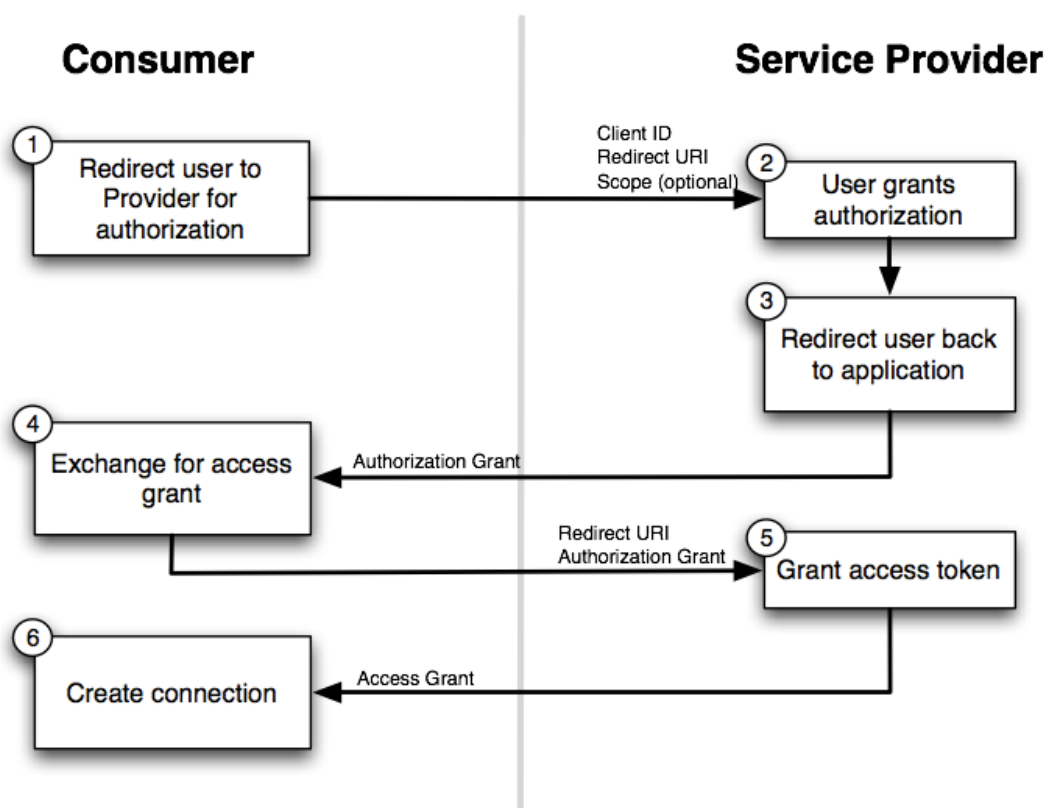
Samotná autorizace, zda je aplikace využívající *API* oprávněna přistupovat ke zdroji klienta banky, se uděluje pomocí frameworku *OAuth2*. [67]

OAuth2

Framework se stal v dnešní době standardem pro autorizaci přístupu ve webových desktopových i mobilních aplikacích. Využívají ho služby jako Facebook, Instagram [12] nebo Github. Framework definuje celkem čtyři typy rolí:

- **Vlastník zdroje** – vlastníkem zdroje je zákazník banky, který poskytuje aplikaci přístup ke svému účtu. Oprávnění aplikace je omezené podle rozsahu, který jí vlastník zdroje přidělí.
- **Konzument** – aplikace žádá přístup k zákaznickému účtu. Přístup autorizuje zákazník a je ověřen poskytovatelem zdroje.
- **Poskytovatel zdroje** – samotné *API*, které odpovídá za autorizaci tokenů a zpracování požadavků.

Následující obrázek ilustruje postup komunikace mezi všemi typy rolí:



Obr. 5 Princip fungování OAuth2 protokolu [60]

Aplikace požádá zákazníka, aby autorizoval aplikaci jako oprávněnou pro přístup k jeho zdrojům. Pokud zákazník požadavek autorizuje, aplikace obdrží autorizační pověření. Toto pověření je následně odesláno aplikací na autorizační server spolu s požadavkem o vydání přístupového tokenu. Server vyhodnotí, zda je pověření platné, a pokud ano, tak pošle aplikaci zpět odpověď s požadovaným přístupovým tokenem. Nyní může aplikace požádat o data ze serveru. Odešle požadavek na server spolu s přístupovým tokenem. Pokud je přístupový token správný, odpoví server vyžádanými daty. [6]

Volání serveru je implementováno čtyřmi metodami:

- **GET** – metoda slouží např. k získání seznamu účtů nebo transakcí a slouží pouze pro čtení. Po jejím provedení nedojde ke změně stavu.
- **POST** – metoda slouží k přidání nového zdroje. Jedná se např. o zadání nové platební transakce. Po provedení metody dochází ke změně stavu.
- **PUT** – metoda mění stav existujícího zdroje. Je možné se s ní už setkat např. u zamykání platební karty.
- **DELETE** – tato metoda odstraní existující zdroj, např. dojde ke smazání zprávy.

Příklad získání přehledu transakcí

Následující příklad ilustruje, jakým způsobem je možné získat přehled provedených transakcí v období od 1. září 2016 do 20. října 2016. Výchozím předpokladem je znalost API Key aplikace a přístupového tokenu zákazníka.

```
https://api.csas.cz/sandbox/webapi/api/v3/netbanking/my/accounts/  
CZ5508000000000379554193/transactions?dateStart=2016-09-  
01T00%3A00%3A00%2B02%3A00&dateEnd=2016-10-20T00%3A00%3A00%2B02%3A00
```

Pokud bude výše uvedený příklad doplněný o pomyslné proměnné, bude vypadat následovně:

```
https://api.csas.cz/sandbox/webapi/api/v3/netbanking/my/accounts/  
{IBAN}/transactions?dateStart={počátečníDatum}&dateEnd={konečnéDatum}
```

Proměnná *IBAN*⁷ obsahuje textový řetězec s *IBAN* bankovního účtu, k němuž se transakce vztahují. Proměnné *počátečníDatum* a *konečnéDatum* ohraničují časový interval, do kterého budou transakce spadat. Formát proměnných je časový údaj ve tvaru *rrrr-mm-ddThh:mm:ss+hh:mm*, v němž časová hodnota za znakem „+“ představuje časovou zónu. Po provedení požadavku a kontrole správnosti všech potřebných údajů odešle API seznam transakcí ve formátu *JSON*⁸. [67]

4.4 Zhodnocení možností

Jako první možnost byla popsána *metoda ručního vkládání transakcí*. Ruční vkládání má výhody v univerzálnosti celého procesu. Bohužel hlavní nevýhodou je právě nutnost ručně vkládat každou provedenou transakci, která činí zmíněnou metodu dlouhodobě neudržitelnou. Proto nebude nad ručním vkládáním transakcí dále uvažováno jako nad možností, která bude využita ve výsledné aplikaci.

Další možností je synchronizace, která je prováděna pomocí *zasílání bankovních výpisů* na speciální e-mailovou adresu aplikace. Výhodou je zejména zautomatizování celého procesu. Tuto možnost synchronizace poskytuje převážná část českých bank. Mezi možné problémy lze počítat závislost na bance na odeslané informaci o transakci. Pokud dojde k technickým potížím a e-mail nebude korektně doručen, dojde tak ke ztrátě informací o transakci. Je nutné transakce tímto způsobem postupně shromažďovat a není možné vynutit zaslání konkrétního výpisu z účtu přímo aplikací.

Poslední popisovanou metodou byla *synchronizace pomocí bankovního API*. Domnívám se, že jde o metodu, která je budoucností pro všechny aplikace jakkoli

⁷ IBAN je formát pro mezinárodní reprezentaci čísla bankovního účtu. Definovaný je mezinárodní normou ISO 13616. [38]

⁸ JSON je zkratka slov JavaScriptová notace (JavaScript object notation). Jedná se o strukturovaná data, která jsou reprezentována jako JavaScriptiové objekty. Používá se pro přenos dat na webových stránkách. [92]

pracující s osobními financemi. Díky možnosti získat libovolný výpis nebo zadat platbu. Aplikace tak není omezena pouze na zasílané výpisy z banky. Vývojáři tímto dostávají do rukou nové možnosti, jak udělat aplikace lepší. Problémem je nízká podpora mezi bankami. V současnosti tuto metodu synchronizace poskytuje pouze Česká spořitelna a Fio banka. S API Fio banky se dobře pracuje, protože je jednoduchá a má přehlednou dokumentaci. Problém vidím v bezpečnosti. Při získání přístupového tokenu může k API, potažmo bankovnímu účtu, přistupovat kdokoli bez omezení. Česká spořitelna poskytuje API pro širší portfolio svých služeb – nejen pro práci s účtem a transakcemi, jako je tomu u Fio banky. Dokumentace je propracovaná a zabezpečení je na vyšší úrovni, zejména díky použití OAuth2 k autorizaci požadavků ve spojení s API klíčem. Paradoxem je, že práce s dokumentací byla i přes technickou propracovanost méně intuitivní než s dokumentací Fio banky, která je dodávaná v obyčejném PDF souboru.

Skutečnost, že své API doposud zpřístupnila pouze Fio banka a Česká spořitelna, v současnosti značně ztěžuje použití API jako jediného způsobu synchronizace – aplikace *API* používají jako doplněk k zasílání bankovního výpisu nebo k možnosti ručního zadání. Předpokladem však je, že budou v letošním roce v trendu zpřístupňování *API* pokračovat i banky další. Značným impulsem k tomu bude nová směrnice evropské unie PSD2.

4.5 Budoucnost bankovního API – PSD2

Jako PSD2 je známá směrnice č. 2015/2366/EU. Hlavním cílem směrnice je snaha o integraci bankovního trhu. Využívá k tomu standardizaci procesů a regulaci nových i stávajících bank. Nová směrnice zavádí velké množství změn. Jedná se např. o tzv. „one-leg“ transakce, při nichž se jedna strana nachází mimo území evropské unie, dále jde o sjednocování bezpečnostních požadavků nebo možnost iniciace plateb pro obchodníky. [63]

Z pohledu této práce je nejzajímavější novinkou nutnost bank otevřít možnost komunikovat se svým bankovníctvím třetím osobám. Konkrétně se bude jednat o možnost získat výpis z účtu nebo zadávat platební příkazy. Banky mohou být v budoucnosti postaveny do role pouhého poskytovatele bankovních služeb – všechny ostatní operace bude teoreticky moci klient provádět prostřednictvím služeb třetích stran. V návaznosti na zpřístupnění operací třetím osobám je též ve směrnici kladen velký důraz na zavedení bezpečnostních standardů, které povedou ke zvýšení bezpečnosti plátců.

Směrnice sama o sobě neurčuje, jakým způsobem mají banky své bankovníctví zpřístupnit. Úkol určit vhodnou technologii připadl autoritě *EBA* (European Banking Authority), která dohlíží nad evropským bankovním sektorem. Mezi finančními institucemi však panuje názor, že by to mělo být právě *API*, prostřednictvím kterého bude přístup umožněn. Jedním z argumentů pro *API* je, že se jedná o široce rozšířenou technologii, kterou již používají velké služby, sociální sítě nebo on-line obchody.

Již zmíněné posílení bezpečnosti je jednou z hlavní části směrnice. Cílem je sjednocení požadavků na autentizaci uživatele, a to ověřením minimálně dvou z kritérií SCA (Strong Customer Authentization)[47]:

- Znalost – něco, co zná pouze uživatel;
 - heslo
 - pin
- Vlastnictví – něco, co vlastní pouze uživatel;
 - certifikát
 - telefon
- Jedinečnost – unikátní poznávací znamení uživatele;
 - otisk prstu
 - oční sítnice

Směrnice nabývá účinnosti 18. ledna 2018. [64]

5 Návrh aplikace

Na základě předchozích kapitol, kde byl podrobně analyzován současný stav aplikací na trhu, a zmapována možnost napojení na internetové bankovníctví nyní přistoupím k návrhu vlastní aplikace. Vzhledem k tomu, že žádná dostupná aplikace neodpovídá požadavkům cílové skupiny, rozhodl jsem se vytvořit aplikaci vlastní, která bude požadavky řešit.

Navržená aplikace se bude jmenovat „Countee“. Název je jednoduchý, unikátní a zapamatovatelný.

5.1 Funkční a nefunkční požadavky

Jedním z prvních kroků procesu systémového inženýrství je stanovení požadavků na systém. Využívá se k tomu metody stanovení funkčních a nefunkčních požadavků.

5.1.1 Funkční požadavky

- **Uživatel má možnost sledovat příjmy a výdaje na bankovních účtech v českých korunách** – jedno místo, kde uživatel získá kompletní přehled o svých příjmech a výdajích. V současnosti, kdy není výjimkou, že uživatel vlastní více než jeden bankovní účet, se může stát, že ztratí přehled o zůstatcích a transakcích na jednotlivých účtech.
- **Uživatel má možnost příjmy a výdaje kategorizovat** – pro získání přehledu o tom, za co uživatel aplikace vydává své prostředky a za co je získává, je vhodné příjmy a výdaje kategorizovat.
- **Uživatel má možnost sledovat závazky** – účet za telefon, zdravotní pojištění, inkaso či internetové připojení – to jsou příklady závazků, u nichž je vhodné sledovat částku a dobu splatnosti. Aplikace bude nabízet uživateli možnost tyto ukazatele sledovat.
- **Uživatel má možnost sledovat pohledávky.**
- **Uživatel se pro práci s aplikací musí přihlásit a mít k dispozici oddělené pracovní prostředí.**
- **Uživatel musí mít možnost odhlásit se ze svého pracovního prostředí.**
- **Aplikace bude synchronizovat transakce na bankovním účtu bez nutnosti explicitního zadávání jednotlivých transakcí uživatelem** – počet bankovních transakcí má neustále vzrůstající tendenci [79]. Vyžadovat po uživateli, aby každou svou transakci jednotlivě zadával do systému ručně, vznáší pochybnosti o dlouhodobé udržitelnosti tohoto stavu. Aplikace musí nějakým způsobem sama synchronizovat příjmy a výdaje na bankovních účtech, aby byla tato činnost zautomatizovaná a nevyžadovala explicitní zadání údajů od uživatele při každé transakci.

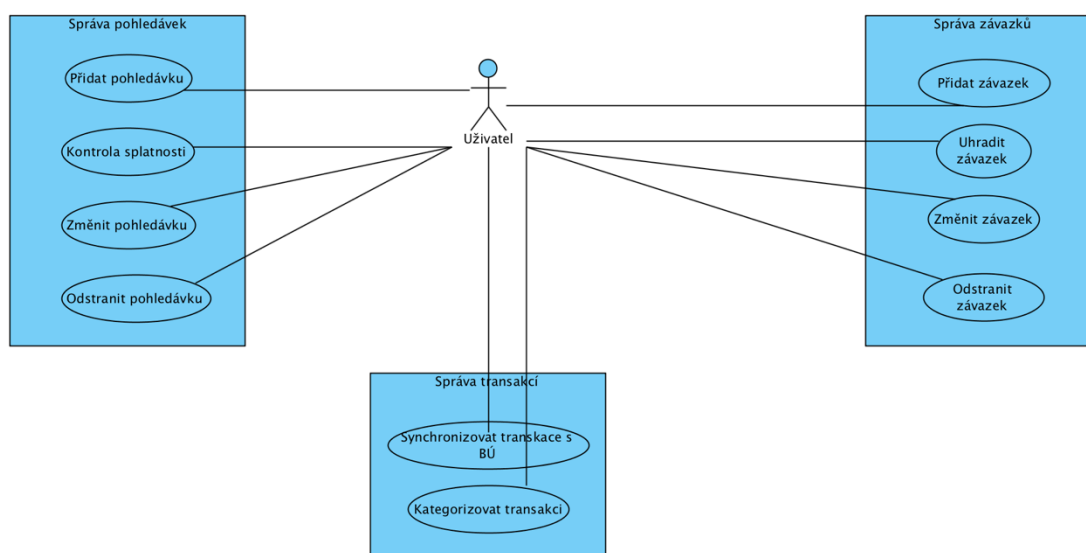
5.1.2 Nefunkční požadavky

- Aplikace bude přístupná ve webovém prohlížeči z desktopu i mobilního zařízení.
- Aplikace bude dostupná v každou denní dobu.
- Aplikace bude implementována v programovacím jazyce PHP s použitím některého z open-source frameworků⁹.

5.2 Případy užití

Na základě požadavků jsou níže zpracovány jednotlivé případy užití. Jediným actorem¹⁰, který bude v této aplikaci přítomen, je uživatel. Uživatel bude mít možnost provádět tři hlavní činnosti, a to:

- spravovat transakce;
- spravovat závazky;
- spravovat pohledávky.



Obr. 6 Use case aplikace

1. Správa transakcí – vlastně se jedná o kategorizaci příjmů a výdajů. Uživatel může synchronizovat účet a zadat kategorii, do níž jednotlivé transakce patří. Všechny transakce budou synchronizovány pouze z bankovního účtu uživatele.

⁹ Kolekce tříd, které jsou programátorovi nápomocny při tvorbě aplikací. [90]

¹⁰ Actor je role, která je vykonávána externí entitou při interakci se systémem. [82]

2. Správa závazků – uživatel bude moci přehledně spravovat své závazky zadáním do systému. Může je měnit i odstraňovat. V neposlední řadě je bude díky napojení na API internetového bankovníctví moci i uhradit.

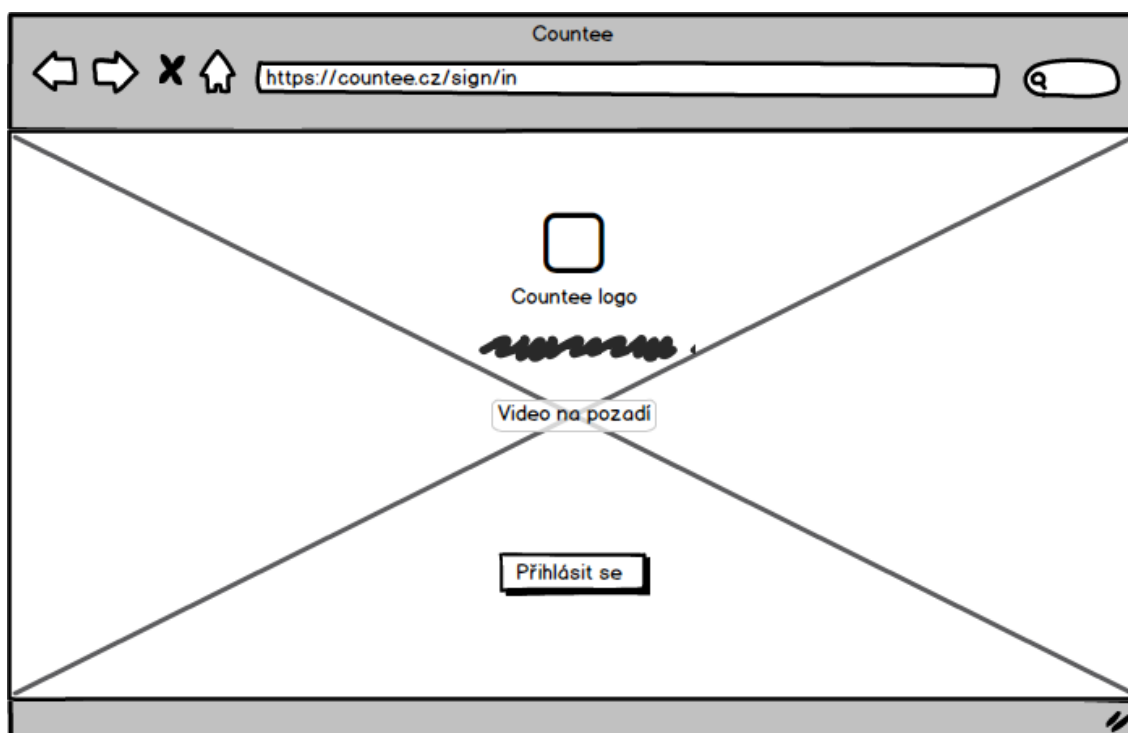
3. Správa pohledávek – uživatel bude moci přidávat a měnit pohledávky. Díky synchronizaci transakcí může sledovat, jestli je pohledávka uhrazená.

5.3 Návrh uživatelského rozhraní

Návrhy uživatelského rozhraní jsou realizovány pomocí drátěných modelů. Jedná se o efektivní způsob, jak navrhnout webovou stránku na strukturální úrovni – jaké budou na klíčových stránkách prvky a jak budou rozmístěny. [3] Samotný drátěný model je pouze vizuální prezentace uživatelského rozhraní, oproštěná od jakéhokoli vizuálního či grafického designu.

5.3.1 Přihlášení uživatele

Jako první byla navržena obrazovka pro přihlášení uživatele. Tato obrazovka je vstupním bodem do celé aplikace a musí jí projít každý uživatel. Na této obrazovce je myšlenka jednoduchá – přímočaře nasměrovat uživatele na přihlašovací tlačítko, které vyvolá akci a ta povede k autorizaci a autentizaci uživatele.

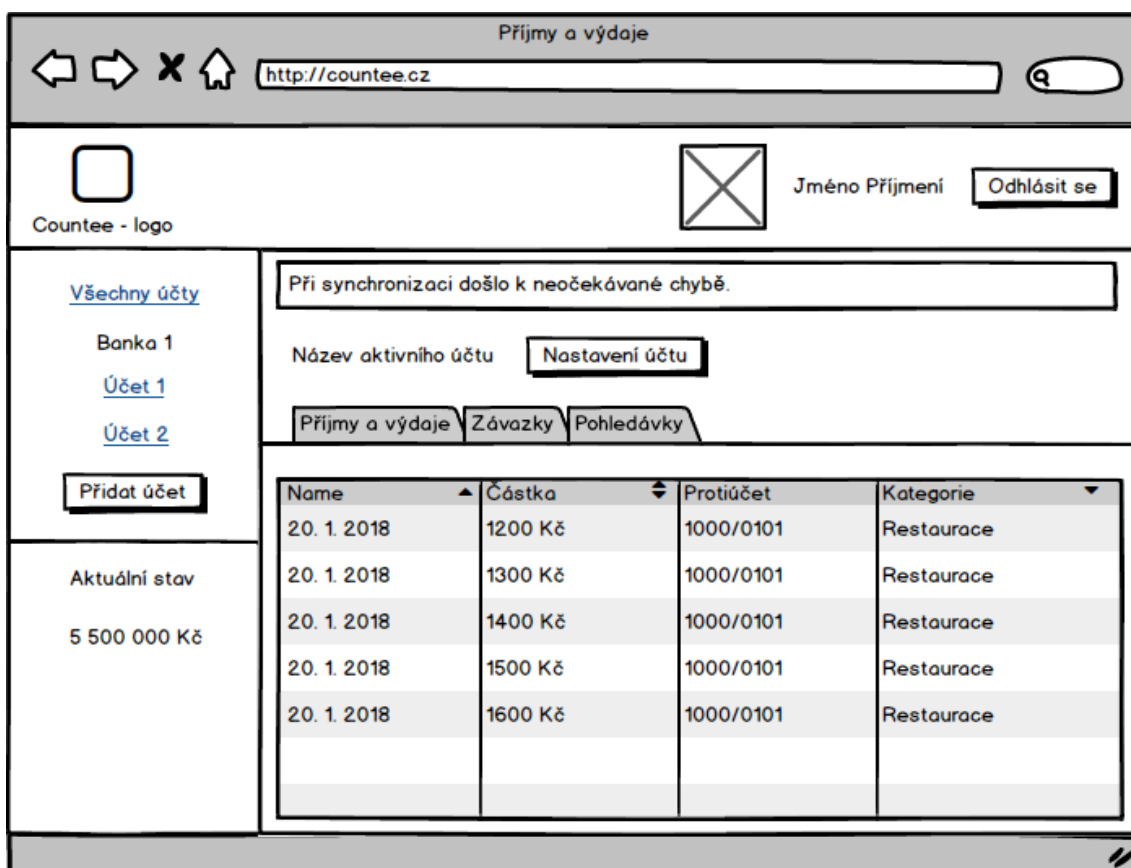


Obr. 7 Drátěný model pro obrazovku přihlášení uživatele

Z výše zmíněného důvodu budou na stránce pouze dva dominantní elementy, a to logo aplikace a přihlašovací tlačítko. Samotné přihlášení bude realizováno pomocí sociální sítě Facebook a jejího uživatelského účtu.

5.3.2 Společná kostra rozhraní aplikace – příjmy a výdaje

Po přihlášení naskočí obrazovka s výpisem příjmu a výdajů. Tato obrazovka je základem pro rozložení všech dalších obrazovek, které aplikace obsahuje. Hlavní části obrazovky jsou: hlavička, boční panel a obsahová část. Právě rozdělení obrazovky do separovaných částí je velmi užitečné z toho důvodu, že uživateli umožňuje rychle se rozhodnout, na jakou část obrazovky má zaměřit svou pozornost. Tento proces bývá často velmi rychlý (stejně jako odpoutání pozornosti od ostatních částí obrazovky – uživatel se zaměřuje jen na pro něj důležité věci). Uvedené poznatky o chování uživatele byly získány z výzkumů metodou sledování lidského oka. [50]



Obr. 8 Drátěný model pro obrazovku s výpisem příjmů a výdajů

V *hlavičce obrazovky* je umístěno logo, *avatar*¹¹, a tlačítko pro odhlášení uživatele z aplikace. Rozdělení hlavičky právě tímto způsobem má své opodstatnění a tím jsou konvence. Uživatel je zvyklý na určité rozložení ovládacích prvků se instinktivně při jejich hledání zaměřuje na ta místa, kde takové ovládací prvky už někdy viděl či použil. Proto je logo umístěno na levé straně hlavičky a bude následně propojeno hypertextovým odkazem na úvodní stranu aplikace. Je to jediný element na stránce, který musí být přítomný na každé obrazovce webové aplikace.[50] *Avatar* a tlačítko s odhlášením je umístěno na pravé straně hlavičky.

Boční panel obrazovky aplikace obsahuje položky menu pro obsluhu přidávaných bankovních účtů. V další části bočního panelu je informační panel s aktuálním stavem účtu. *Hlavní obsahová část obrazovky* se skládá z bloku se zprávami, názvu aktuální stránky spolu s názvem aktivního účtu, hlavního menu a vlastní části pro obsah.

Blok se zprávami je prostor pro vyvolání informace o aktuálním stavu aplikace. Jedná se zejména o zobrazení chybových hlášení, úspěšného zadání nebo informace o průběhu zpracování. Tento blok by měl být umístěn na místě, na němž uživatel taková hlášení očekává. [44] *Hlavní menu* je vlastně globální navigace, která je přítomná na každé obrazovce přihlášeného uživatele. Navigace obsahuje tři položky, které bude uživatel pravděpodobně nejvíce využívat. Obsahuje odkazy pro přepnutí mezi hlavními funkcemi aplikace, což jsou „Příjmy a výdaje“, „Závazky“ a „Pohledávky“. To, že je navigace stále na stejném místě, je pro uživatele velmi důležité – dává mu tak určitý pocit jistoty. Neustále ví, že se nachází ve stejném prostředí, a ví, že navigaci může kdykoli a z jakékoli obrazovky použít. Pokud by na některé z obrazovek nebyla navigace přítomna, může dojít ke zmatení uživatele. [50] Další součástí obsahové části obrazovky je *vlastní obsah*. Ten se mění podle dané obrazovky. Konkrétně u výpisu příjmů a výdajů obsahuje tabulku s přehledem jednotlivých transakcí, které jsou synchronizovány z bankovního účtu.

5.3.3 Správa závazků a pohledávek

Obrazovky pro správu závazků a pohledávek jsou postaveny na stejném principu jako obrazovka pro výpis příjmů a výdajů. Rozdílný je pouze vlastní obsah, který obsahuje výpis závazků, potažmo pohledávek.

¹¹ Avatar je grafická ilustrace, která reprezentuje podobiznu uživatele počítače. Avatary jsou hojně využívané na webových stránkách nebo v počítačových hrách [13]

5.4 Grafický návrh rozhraní

Grafický návrh rozhraní jsem provedl jako další krok v návaznosti na navržené uživatelské rozhraní pomocí drátěných modelů. K návrhu jsem využil programy Adobe Illustrator pro vektorovou grafiku a Adobe Photoshop pro výslednou kompletaci návrhů.

5.4.1 Logo

„Logo je grafické ztvárnění názvu organizace, společnosti, firmy nebo instituce obohacené o piktogram. Tento pojem odpovídá českému výrazu grafická značka.“ [84]

Podle vizuálního motivu se dělí na logo[84]:

- obrazové – má za svůj vizuální prvek piktogram;
- typografické – má individuálně zpracováno písmo a je bez piktogramu;
- kombinované – vzniká spojením loga obrazového a typografického.

Symbol v logu aplikace vznikl na základě spojení následujících motivů:

- oko – symbolizující přehled a kontrolu;
- mince – ztvárňující finance;
- pruhy – znázorňující výpis příjmů i výdajů;
- písmeno „C“ – jako iniciála názvu.

Obrazové logo v kombinaci s názvem aplikace vysázeným verzálkami „Coun-tee“ tvoří výsledné kombinované logo (viz Obrázek č. 9).

Barvy v logotypu jsem zvolil s ohledem na psychologii barev a jejich dopad pro uživatele aplikace. V logu je použita zelená „Caribbean green“ a modrá „Rhino“. Názvy barev jsem vygeneroval pomocí generátoru „Name that Color“[56]. Zelená barva může symbolizovat peníze a bohatství [75] nebo navozovat pocit bezpečí, rovnováhy, klidu a naděje. Působí uklidňujícím dojmem, což může napomáhat k celkové regeneraci a načerpání nové síly [77]. Modrá barva je barvou spolehlivosti, celkově uklidňuje a zlepšuje schopnost se soustředit. Vyvolává v lidech pocity důvěryhodnosti, proto také bývá často užívaná v designu, který se dotýká finančních institucí nebo on-line podnikání. Jedná se o nejpoblárnější firemní barvu[37].

Logo je v základním provedení tvořené zeleným obrazovým logem a modrým typografickým logem na bílém (bezbarvém) pozadí. Při užití loga je možné vybrat i jednu ze dvou dalších nabízených barevných variant. Pokud bude celé logo na modrém pozadí nebo se varianta s modrým podkladem bude více hodit, barva obrazového loga se nezmění a zůstane tedy zelená. Typografická část loga by ovšem zanikla, proto je zvolena původně barva pozadí, tedy bílá (bez barvy). Bílá typografie se použije i v případě zeleného pozadí. Obrazové logo získá barvu typografie ze základního provedení, modrou barvu. Jiná barevná varianta loga není možná.

Zvoleným písmem pro potřeby loga je Alegreya Sans v řezu Bold. Jde o elegantně působící humanistické písmo, jehož autorem je designer Juan Pablo del Peral [5]. V případě použití písma (nejen) pro logotyp je potřeba brát ohledy na to, jaká je jeho licence. V tomto případě se jedná o font z knihovny Google Fonts s licencí *OFL*¹², která je poskytovaná zdarma a umožňuje téměř neomezené využívání fontu a jeho modifikaci.

Návrh loga jsem v první fázi prováděl jako skicu na papíře. Následně jsem vytvořenou skicu přetvořil do digitální podoby pomocí programu Adobe Illustrator CC, který je zaměřený především na práci s vektorovou grafikou. Zvolenými výstupy je logo ve vektorových formátech pdf, svg a v rastrovém formátu png.



Obr. 9 Logo aplikace

5.4.2 Typografie

Pro celou aplikaci jsem zvolil písmo Rubik, které je stejně jako písmo Alegreya Sans z knihovny Google Fonts s *OFL* licencí. Jedná se o bezpatkové písmo s mírně zaoblenými rohy od designerů Huberta & Fisher. Písmo bylo vytvořené jako součást projektu Chrome Cube Lab a v roce 2015 bylo designérem Meir Sadanem doplněné o hebrejské znaky. O rok později se do písma přidaly také cyrilické znaky designérem Alexeiem Vanyashinem [66]. Práci na grafickém návrhu jsem začal v programu Adobe Photoshop CC, v němž jsem si nejprve předdefinoval odstavcové styly, které jsem poté využíval při formátování textu.

¹² OFL je zkratka pro Open Font License. Jedná se o volnou, svobodnou a open-source licenci navrženou speciálně pro písma. [71]

Rubik, Medium (velikost:18 px, prostrkání: 25 px, proklad: 26 px)

Rubik, Light (velikost 18 px, prostrkání: 25 px, proklad 26 px)

Rubik Medium (velikost: 14 px, prostrkání: 25 px, proklad: 22 px)

Rubik Regular (velikost: 14 px, prostrkání: 25 px, proklad: 22 px)

Rubik Medium (velikost: 12 px, prostrkání: 25 px, proklad: 20 px)

Rubik Regular (velikost: 12 px, prostrkání: 25 px, proklad: 20 px)

Rubik Medium (velikost: 10 px, prostrkání 25 px, proklad: 15 px)

Rubik Regular (velikost: 10 px, prostrkání 25 px, proklad: 15 px)

Obr. 10 Předdefinované odstavcové styly v grafickém návrhu

5.4.3 Ikony

Ikony, které jsem použil v grafickém návrhu aplikace, pochází z projektu FontAwesome. Technicky se jedná o font s OFL licencí, který je v základní verzi zdarma a jehož znaky jsou jednotlivé ikony. Celkem obsahuje 675 ikon, u nichž jde jednoduše měnit rozměry, barvu, stínování a další atributy. Díky vektorovému formátu nedochází k problémům při změně velikosti nebo zobrazování na obrazovkách s retina displayem. FontAwesome jsem zvolil kvůli snadné použitelnosti jak v grafickém návrhu, tak v jeho následné implementaci. [76]

Pro potřeby grafického návrhu lze font nainstalovat do počítače a používat jej v programu Adobe Photoshop. Pro výběr ikony jsem použil tzv. cheatsheet, který je dostupný na oficiálních stránkách projektu. Cheatsheet obsahuje seznam všech ikon a jejich unikátní kód. Po zkopírování ikony do textového bloku v programu a nastavení fontu na FontAwesome ikonu zobrazí [35].

5.4.4 Postup tvorby návrhu

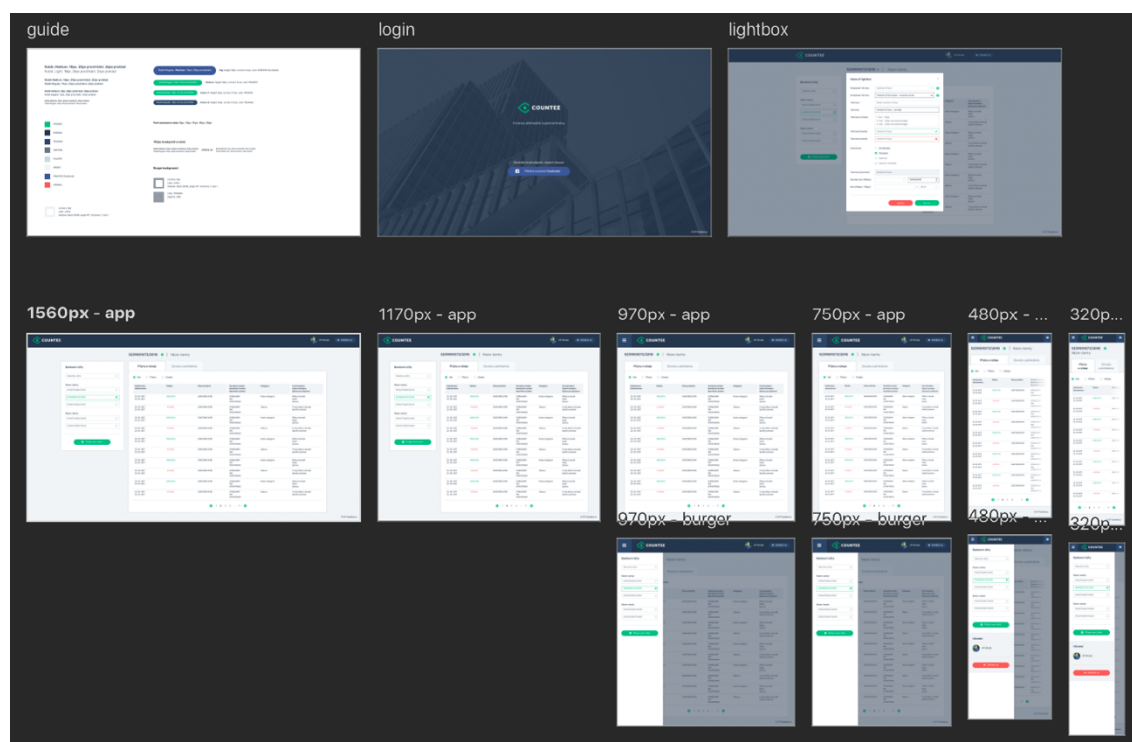
Grafický návrh aplikace jsem vytvářel v programu Adobe Photoshop CC. Tento program ve své aktuální verzi disponuje tzv. návrhovými plochami, které umožňují, aby bylo v jednom souboru umístěno více ploch najednou. Není tak už potřeba pro každou navrhovanou stranu vytvářet nový dokument pouze s jedním plátnem. Návrhové plochy lze samozřejmě pojmenovat, což pomáhá přehlednosti v dokumentu při vyšším počtu ploch, jako je tomu v tomto případě.

Dalším důležitým kritériem pro dobrou orientaci v PSD souboru je přehledné hierarchické uspořádání a věcné pojmenování jednotlivých vrstev a složek v panelu „Vrstvy“. Tato činnost je přínosná i v případě, že bude později potřeba provést úpravy nebo předávat PSD soubor jako podklady pro práci další osobě.

Ve Photoshopu se při zpracování grafického návrhu dá dopracovat téměř obdobného výsledku více způsoby. Jako příklad může být např. tvorba jednobarevné-

ho obdélníku. Pokud bych jej vytvořil pomocí nástroje „obdélníkový výběr“ a nástroje „výplň“, docílím tak zdánlivě stejného výsledku, jako v případě použití nástroje „obdélník“. V prvním případě je ovšem vytvořena rastrová vrstva, která neposkytuje tolik možností, jako nabízí nástroj „obdélník“, vytvářející živý tvar. Živé tvary mají nastavitelné vlastnosti od určení přesných rozměrů (bez nutnosti měnit transformaci pomocí tažení myši za okraje) po nastavení zaoblených okrajů, jejichž velikost se určuje v jednotkách, v tomto případě v pixelech.

Finální grafický návrh této aplikace má množství vrstev a 13 návrhových ploch. Každý obdobně vypadající objekt, text či písmo by bylo časově náročné tvořit od základu znovu. V tom pomáhají funkce „duplikovat vrstvu“, „duplikovat skupinu“ nebo „kopírovat styl vrstvy“ a „kopírovat atributy tvaru“, které jsou dostupné při otevření nabídky pravým klikem myši na požadovanou položku v panelu „vrstvy“.



Obr. 11 Návrh obrazovek aplikace, rozdělený do návrhových ploch programu Adobe Photoshop

5.5 Návrh databáze

Návrh databáze obsahuje osm tabulek – `account`, `account_meta`, `bank`, `transaction`, `receivable`, `category`, `commitment` a `user`.

Tabulka `user` slouží k uchování údajů o uživateli. Je přímo či nepřímo propojena se všemi ostatními tabulkami v databázi.

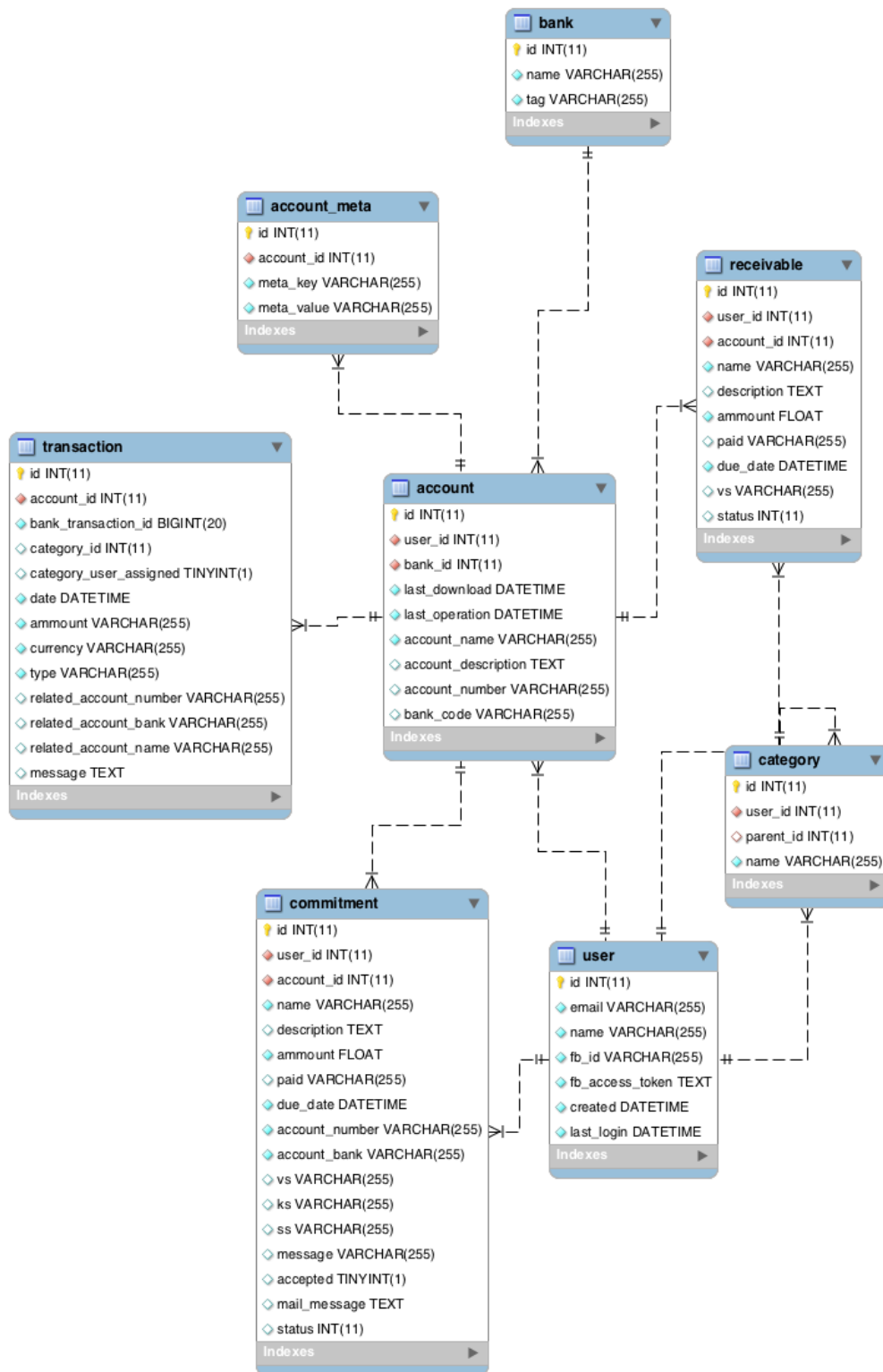
Tabulka `account` slouží k uchování údajů o přidaných bankovních účtech uživatelem. Kvůli nutnosti budoucí rozšiřitelnosti jsem přidal do návrhu databáze tabulku `account_meta`, která poskytuje možnost k uložení atributů, specifických pro jednotlivá bankovní *API*. Konkrétně v případě Fio banky, se může jednat o bankou vygenerovaný autorizační token.

Tabulka `transaction` uchovává informace o synchronizovaných transakcích. Obsahuje nepovinné atributy o protiúčtu, protože např. při karetních transakcích není protiúčet uveden.

Tabulky `receivable` a `commitment` slouží ke správě pohledávek a závazků.

Tabulka `category` slouží pro definici kategorií, které je následně možné přiřadit ke každé transakci.

Schéma databáze je znázorněno na obr. 12.



Obr. 12 Schéma databáze

6 Implementace

Aplikaci jsem implementoval na základě návrhu v předchozí kapitole. V souladu se zadáním diplomové práce budou (mimo jiné) použity technologie PHP a JavaScript. Webová aplikace bude též responzivní.

Tento základní požadavek na použité technologie však musím rozšířit o technologie další, které na ně přímo, či nepřímo navazují. Převážnou část těchto technologií popíšete v první části této kapitoly. Další použité technologie a knihovny popíšete v dalším textu, který se bude jejich použitím přímo týkat.

6.1 Použité technologie

PHP

PHP je široce rozšířený programovací jazyk, který je primárně určen pro vývoj webových aplikací [91]. Jedná se o programovací jazyk, v němž je implementovaná hlavní logika aplikace – konkrétně ve verzi 7.0. PHP jsem ji použil kvůli rozšířenosti a vhodnosti pro vývoj webových aplikací. Verzi 7.0 jsem zvolil oproti starší verzi 5.6. hlavně kvůli novinkám a vylepšením, které verze 7.0 obsahuje. Vyzdvihnul bych lepší práci této verze s pamětí a násobně větší schopnost zpracování požadavků než u předchozí verze. Ze syntaktického hlediska přináší nová verze např. typovou kontrolu pro skalární typy, typy návratových hodnot nebo anonymní třídy. [43]

Nette

Při realizaci rozsáhlejšího projektu se u webových aplikací v PHP lze v dnešní době těžko obejít bez PHP frameworku. I když samotný jazyk obsahuje spoustu užitečných funkcí, časem začne nutně docházet k tomu, že je třeba stále dokola programovat různé obslužné funkce např. na správu databáze, obsluhu nebo zabezpečení formulářů a podobně. K tomu slouží framework, který dokáže ušetřit značnou část času, který se stráví na opakovaných činnostech, a pomoci i při návrhu a přehlednosti aplikace. [83]

Pro účely této diplomové práce, jsem zvolil framework **Nette** ve verzi 2.4. Vybral jsem ho z toho důvodu, že je to řešení převážně od českých vývojářů, s aktivní českou komunitou uživatelů a je dlouhodobě vyvíjený a udržovaný. Mimo jiné se umístil na třetím místě v anketě portálu Sitepoint.com o nejlepší Framework roku 2015. [74]

Nette pracuje v návrhovém vzoru Model – View – Presenter (MVP). Jedná se o odvozený a velmi podobný návrhový vzor jako MVC, lišící se však pouze v detailech. [54]

MariaDB

MariaDB je relační databáze zpětně kompatibilní s MySQL. Zvolil jsem ji zejména kvůli lepší výkonnosti oproti MySQL a také kvůli otevřenosti vývoje. [52]

Phinx

PHP knihovna Phinx nabízí způsob, jakým jednoduše spravovat databázové migrace v PHP aplikaci. Činí tak ze změn databáze jednoduchou a dlouhodobě udržitelnou činností, kterou lze verzovat, tj. uchovávat historii veškerých změn provedených v informacích nebo datech, pomocí verzovacích systémů. V aplikaci Phinx využívám pro vytvoření databázových tabulek, vazeb mezi nimi a následně k naplnění tabulek záznamy. Phinx při spuštění migrací vytvoří vlastní tabulku v databázi, v níž udržuje záznamy o aktuálním stavu. [57]

HTML 5

HTML je značkovací jazyk pro tvorbu webových stránek. Verze 5 je nástupcem dnes již zastaralé verze 4 z roku 1999. Novější verzi HTML jsem pro implementaci zvolil z toho důvodu, že obsahuje novou funkcionalitu, která napomáhá implementaci dnešních požadavků na webové prezentace. Jedná se například o audio a video rozhraní, sémantické elementy (header, footer, section, nav, article, aside), rozhraní pro SVG, geolokaci, úložště a další. [51]

Javascript / JQuery

Použití Javascriptu, potažmo knihovny JQuery¹³, je vyžadováno převážně jako prerekvizita pro fungování dalších technologií, které jsem v aplikaci použil. Užívá se např. pro validaci formulářů, AJAX nebo vyskakovací okna.

Bootstrap 3

Bootstrap je nejpopulárnější framework pro tvorbu frontendu webových stránek. Je responzivní a poskytuje širokou paletu komponent, jako jsou modální okna, ikony, formuláře, mřížku, tabulky a mnoho dalších možností. [74] Pro účely aplikace jsem zvolil Bootstrap ve verzi 3 z toho důvodu, že nová, čtvrtá verze frameworku byla v době zpracování této práce ve verzi alfa. Proto jsem ji zatím shledal nevhodnou k implementaci této aplikace.

Docker

Docker je kontainerová platforma, která napomáhá ke snadnějšímu vývoji a provozování aplikací. Kontainer je izolované prostředí, které poskytuje vše potřebné, co je třeba k funkčnosti provozovaného software. [88] V této implementované aplikaci je Docker použit ke zprovoznění Apache serveru, databáze MariaDB a e-mailového serveru.

Spuštění Docker kontainerů se provádí provedením příkazu `docker-compose` ve složce `/docker`.

¹³ JQuery je JavaScriptová knihovna, která slouží k manipulaci s HTML, pro správu událostí a obecně ke zjednodušení práce s JavaScriptem. [89]

6.2 Struktura aplikace

Základní struktura aplikace vychází ze sandboxu nette/sandbox. Jedná se o jednoduchý, předpřipravený balíček, který lze použít jako výchozí bod pro postavení vlastní aplikace, založené na Nette frameworku. Pro správu balíčků využívá Composer¹⁴, pomocí něhož jsou spravovány všechny potřebné knihovny.

Aplikace je postavena na návrhovém vzoru MVC. Hlavním principem je oddělení logiky aplikace od zobrazení výstupu. Architektura rozděluje aplikaci na tři vrstvy - Model, Controller a View.

Adresářová struktura aplikace

- App – obsahuje vlastní soubory s aplikací:
 - Components – logika a šablony komponent;
 - Config – konfigurační soubory;
 - Console – logika pro příkazy vyvolané z CLI¹⁵;
 - Helpers – pomocné třídy a funkce;
 - Model – modelová vrstva a její třídy;
 - Presenters – třídy presenterů a šablony;
 - Router – třídy routerů;
- Db
 - Migrations – třídy s migracemi databáze;
 - Seeds – třídy pro naplnění databáze daty;
- Docker – soubory potřebné pro spuštění aplikace v prostředí dockeru;
- Frontend – zdrojové soubory frontendu;
- Log – logy, erry;
- Temp – dočasné soubory;
- Vendor – soubory knihoven stažené pomocí composeru;
- www – veřejně přístupný adresář obsahující obrázky, styly a další soubory.

Model

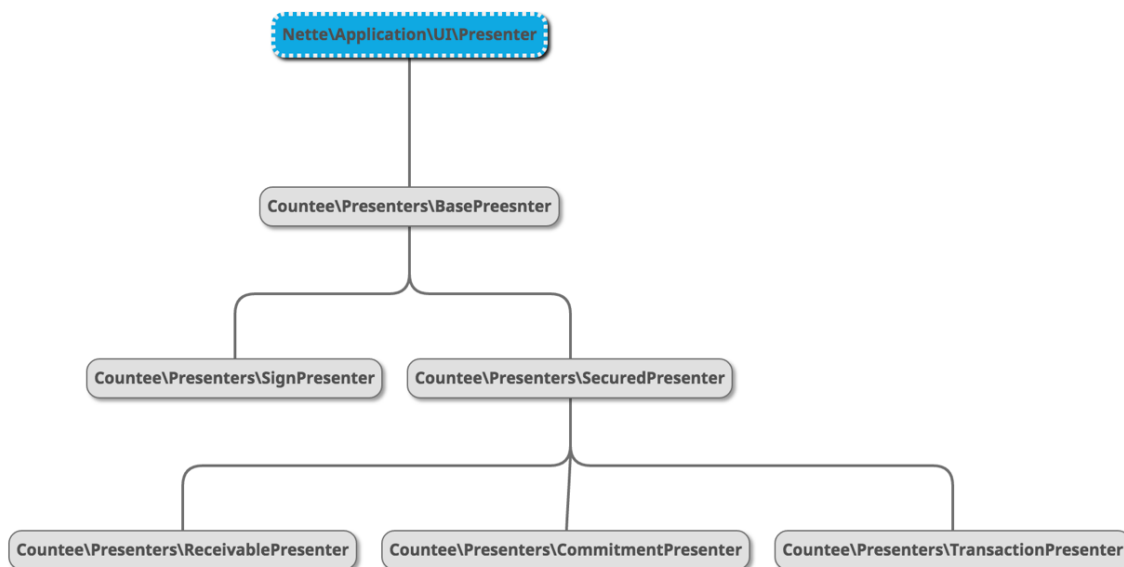
Modelová vrstva reprezentuje data a práci s nimi. V aplikaci jsou třídy s modelem umístěny do jmenového prostoru Countee\Model. Základní abstraktní třídou, která rozšiřuje většinu tříd modelu, je Countee\Model\BaseDatabaseModel. Třída BaseDatabaseModel obsahuje základní funkcionality pro komunikaci s databází a metody pro získání záznamu s účtem, bankou, uživatelem, závazkem a pohledávkou.

¹⁴ Composer je nástroj pro efektivní správu balíčků knihoven a jejich závislostí v PHP. [16]

¹⁵ CLI je zkratka Comain Line Interface. Jedná se o uživatelské rozhraní, které je ovládané textovými příkazy, zadávanými do příkazových řádků. [60]

Presenter

Presentery jsou objekty, které zajišťují správu akcí uživatele a propojují modelovou vrstvou s šablonou (view). Presentery jsou umístěny ve jmenném prostoru `Countee\Presenters`.



Obr. 13 Hierarchie presenterů aplikace

Rodičem všech presenterů je třída `Nette\Application\UI\Presenter`, která je součástí balíčku `nette\application`. Od základního presenteru dědí `BasePresenter`, který poskytuje společnou vlastní funkcionalitu aplikace pro všechny presentery. Přihlášení a registraci uživatelů do aplikace zajišťuje `SignPresenter`. Ověření přihlášení a rodiče pro všechny presentery, které vyžadují přihlášení uživatelé, zajišťuje `SecuredPresenter`. V následujícím příkladu je znázorněn postup ověření přihlášení uživatele. Při neoprávněném přístupu přesměruje presenter uživatele na úvodní stránku s přihlášením do systému.

```
if (!$this->user->isLoggedIn()) {  
    $this->flashMessage('Pro vstup na tyto stránky je nutné přihlášení.', 'error');  
    $this->redirect('Sign:');  
    die('Unauthorized');  
}
```

6.3 Přihlášení

Jedním z funkčních požadavků je poskytnout uživateli možnost přihlášení a s tím související oddělené pracovní prostředí. Jako metodu přihlášení jsem vybral funkcionalitu Facebooku, který umožňuje rychlou, snadnou registraci a přihlášení

do třetích aplikací pomocí účtu na této sociální síti. Integraci přihlašování velmi usnadňuje rozšíření Nette frameworku - Kdyby/Facebook. Jde o řešení postavené na oficiálním *SDK*¹⁶ od Facebooku, ale plně integrované do prostředí Nette.

Proces přihlášení a registrace řeší `SignPresenter` (`Countee\Presenters\SignPresenter`). Uživatel je při pokusu o přihlášení přesměrován na Facebook, který požádá uživatele o udělení přístupu ke konkrétním údajům, vyžadovaných od aplikace. Jedná se o jméno, e-mailovou adresu a profilový obrázek. Pokud uživatel aplikaci udělí přístup, Facebook nasměruje svůj požadavek zpět do aplikace a aplikace se následně rozhodne, zda provede přihlášení, nebo registraci nového uživatele.

Pokud aplikace nenalezne uživatele podle přiděleného ID Facebooku, provede registraci nového uživatele. V případě nutnosti dojde k aktualizaci přístupového tokenu a poté je uživatel do aplikace přihlášen pomocí `Identity` (`Nette\Security\Identity`). Ta je vytvořena na základě údajů poskytnutých Facebookem, která je přidána třídě `User` (`Nette\Security\User`). Následující ukázka kódu znázorňuje popisovanou funkcionalitu při přihlašování.

```
// Získání údajů z Facebook API
$me = $fb->api('/me?fields=id,name,email,picture');
$existing = $this->userManager->findByFacebookId($fb->getUser());

// Ověření existence uživatele v databázi a případná registrace.
if (!$existing) {
    $existing = $this->userManager->registerFromFacebook($me);
}

$this->userManager->updateFacebookAccessToken($fb->getUser(), $fb->
    >getAccessToken());

// Přihlášení uživatele do aplikace
$this->user->login(new \Nette\Security\Identity($existing->id, null,
    $existing));
```

6.4 Synchronizace s bankou

Zcela zásadní funkcionalitou je synchronizace bankovních transakcí s aplikací. Pro účely této práce jsem zvolil možnost synchronizovat transakce s Fio bankou, avšak aplikace je za pomocí interface a abstraktních metod snadno rozšiřitelná o API České spořitelny či dalších API, až budou v budoucnu zpřístupněny veřejnosti.

Funkcionalitu spojenou se synchronizací transakcí vykonávají třídy ve jmenovém prostoru `Countee\Model\Bank`. Konkrétně v případě Fio banky se jedná o třídu `Account` (`Countee\Model\Bank\Fio\Account`), která implementuje rozhra-

¹⁶ SDK je zkratka anglických slov Software Development Kit. Jedná se o sadu nástrojů, které umožňují programátorovi vytvářet aplikace pro specifickou platformu. [68]

ní IAccount a třídu Transaction (Countee\Model\Bank\Fio\Transaction), implementující rozhraní ITransaction.

1. Presenter získá instanci třídy dle banky, u níž je účet veden. Objekt předloží presenteru třída AccountFactory a její metoda getObject.
2. Získaný objekt má implementovanou metodu syncAccount, která má v parametru identifikátor účtu a provede jeho synchronizaci s databází aplikace.

Fio banka obsahuje omezení v počtu požadavků. Konkrétně se jedná o jeden požadavek za 30 sekund pro jeden vygenerovaný token. Pokud je omezení požadavků překročeno, dochází ze strany banky k vyvolání chybového stavu č. 409 (Conflict). V aplikaci se snažím tomuto stavu předcházet pomocí zaznamenání posledního požadavku, který byl proveden. Pokud bude požadavek přesto chtít uživatel aplikace provést, dojde k jeho zablokování.

```
if (($downloadTo->getTimestamp() - $downloadFrom->getTimestamp()) <
self::SYNC_LIMIT) {
    throw new \Countee\Model\Bank\Exception\SyncLimitException;
}
```

Konkrétní příklad se týká nejen synchronizace transakcí, ale podobným způsobem je ošetřena i veškerá komunikace s API Fio banky. Ke komunikaci s bankovním API využívám PHP knihovnu *Guzzle*.

6.5 Kategorizace příjmů a výdajů

Ke každé synchronizované transakci je možné přiřadit kategorii, která reprezentuje její účel:

- Auto a doprava;
- Děti;
- Firemní;
- Každodenní;
- Osobní;
- Ostatní;
- Platby;
- Úspory a investice;
- Zábava.

Přiřazení probíhá dvěma způsoby, a to ručním přiřazením uživatelem a automatickým přiřazením uživatele. Aplikace se učí z každého zadání kategorie uživatelem a následně je schopna přiřadit kategorie i sama.

Proces automatického přiřazení kategorií pracuje s mírou shody bankovního účtu a poznámkami, oproti transakcím, které uživatel sám kategorizoval. Porovnání poznámek se provádí pomocí php metody `similar_text`. Metoda porovnává

procentuální shodu dvou textových řetězců – v tomto případě poznámek u transakcí. Řetězce jsou vyhodnoceny jako odpovídající, pokud přesáhne míra procentní shody 80 %. Výsledkem porovnání obou parametrů jsou dvě pole, z nichž každé obsahuje transakci s největší mírou shody seřazené sestupně.

Konečné vyhodnocení a získání id kategorie, do které by měla transakce patřit, vykoná následující část algoritmu metody assignCategory.

```
if ($messageMatch && $bankAccountMatch) {
    foreach ($messageMatch as $key => $value) {
        if (in_array($key, $bankAccountMatch)) {
            return $this->getTable('transaction')->get($key)->category_id;
        }
    }
} if ($bankAccountMatch) {
    reset($bankAccountMatch);
    return $this->getTable('transaction')->get(key($bankAccountMatch))->category_id;
} if ($messageMatch) {
    reset($messageMatch);
    return $this->getTable('transaction')->get(key($messageMatch))->category_id;
}
```

Transakce je vyhodnocena jako shodná prioritně v tomto pořadí:

1. Je nalezena shoda jak v čísle účtu, tak v poznámce transakce.
2. Je nalezena shoda v čísle účtu.
3. Je nalezena shoda v poznámce transakce.

Pokud není shoda nalezena v žádném z uvedených bodů, není provedeno přiřazení žádné kategorie. Automatické přiřazení je i po nalezení shody prioritně na druhém místě – první je vždy přiřazení kategorie uživatelem.

20. 12. 2016	-190,47 CZK	Domácnost	
		Doplňky a vybavení	čka 34, Brno
		Dům a zahrada	.12.2016,
		Pojištění domácnosti	
		Provoz domácnosti	
		Domácnost - jiné	čka 34, Brno
		Firemní	.12.2016,
		Pronájem obchodních prostor	
		Převod na firemní účet	NO, dne
		Příjmy z podnikání	360.27 CZK
		Firemní - jiné	
		Každodenní	čka 34, Brno
		Alkohol	.12.2016,
		Cigarety	
		Domácí zvířata	
		✓ Potraviny a drogerie	čka 34, Brno
		Restaurace	.12.2016,
		Každodenní - jiné	
		Osobní	NO, dne
		Dárky a dobročinnost	293.30 CZK

Obr. 14 Ruční přiřazení kategorie transakce uživatelem

6.6 Správa závazků

Implementovaná aplikace obsahuje možnost správy závazků – umožňuje závazek přidat, upravit a uhradit. Přidat závazek lze třemi různými způsoby: ručním zadáním informací o závazku, zadáním pomocí faktury s QR kódem a zadáním pomocí SPAYD řetězce. Funkcionalitu spojenou se správou závazků implementuje Commitment model (Countee\Model\Payable\Commitment), Commitment presenter (Countee\Presenters\CommitmentPresenter) a komponenty ve jmenném prostoru Countee\Components\Payable.

6.6.1 Short Payment Descriptor (SPAYD)

SPAYD je formát, prostřednictvím kterého lze usnadnit placení faktur. Fakticky se jedná o řetězec znaků, do něhož jsou vloženy všechny potřebné údaje pro platbu, jako je číslo účtu, kód banky, variabilní symbol nebo částka, kterou je třeba uhradit. Informace o platbě v tomto formátu jsou přenášeny buď v souboru s příponou `.spayd` nebo v QR¹⁷ kódu. [72]

Protože se tento formát používá i pro uložení v QR kódu, je důležité, aby byl SPAYD řetězec sestaven pouze ze znaků [72]:

- čísla 0–9,
- velká písmena anglické abecedy A–Z,
- mezera,
- \$, %, *, +, -, ., /, : .

Začátek řetězce je tvořen hlavičkou SPD*, za níž následuje verze protokolu (1.0*). Za touto úvodní částí následují platební atributy ve tvaru {klíč}:{hodnota}*. Název klíče je sestaven z velkých písmen anglické abecedy (A-Z) a volitelnou pomlčkou. Klíče mohou být dvojího typu: přípustné a proprietární. Přípustné klíče jsou základní sadou atributů, kterou rozšiřují proprietární klíče s lokálním významem. Odlišují se tím, že název atributu začíná „X-“. [72]

Hodnota atributu může nabývat libovolného znaku s výjimkou „*“ a „:“, které ve formátu slouží jako oddělovače. Hodnota může obsahovat i speciální znaky, zakódované pomocí URL kódování – hvězdička lze tím pádem zapsat jako „%2A“, a obejít tak uvedené předchozí omezení. [72]

Formát obsahuje celkem 17 nepovinných a 1 povinný parametr. V následujícím výpisu blíže popíši 9 parametrů, které dokáže zpracovat jakákoli tuzemská banka [72]:

¹⁷ QR kód je zkratka pro Quick Read a jedná se o dvourozměrný čárový kód. Může uchovávat zakódovaná data. Ve většině případů obsahuje odkaz na webovou stránku, ale může obsahovat i jakoukoli jinou informaci. [87]

- ACC – Jediný povinný parametr, který obsahuje číslo účtu protistrany při platebním styku. Skládá se z povinného *IBAN* a nepovinného *BIC*¹⁸.
 - Na následujícím příkladu je uvedeno číslo účtu protistrany a swift kód– `CC:CZ5855000000001265098001+RZBCCZPP*`
- ALT-ACC – Je to nepovinný parametr, uplatňující se jako doplněk k parametru ACC. Slouží ke specifikaci až dvou alternativních účtů pro možnou optimalizaci bankovních poplatků za bankovní převod. Skládá se z maximálně dvou řetězců ve stejném formátu jako ACC, které jsou odděleny znakem „,“.
 - Příklad obsahu parametru se dvěma bankovními účty:
`ALT-ACC:CZ5855000000001265098001+RZBCCZPP,
CZ5855000000001265098001*`
- AM – Nepovinný parametr pro určení výše částky platby. Jedná se o desetinné číslo s maximálně dvěma desetinnými ciframi.
 - Příklad obsahu parametru: `AM:480.55*`
- CC – Nepovinný parametr definuje měnu platby ve formátu ISO 4217. Obsahuje pouze velká písmena.
 - Příklad obsahu parametru: `CC:CZK*`
- DT – Nepovinný parametr definuje datum splatnosti. Obsahuje numerické znaky ve formátu ISI 8601 (RRRRMMDD).
 - Příklad obsahu parametru: `DT:20171231*`
- MSG – Nepovinný parametr definuje zprávu pro příjemce. Může obsahovat libovolný znak z povolené množiny kromě „*“.
 - Příklad obsahu parametru: `MSG:PLATBA ZA ELEKTRINU*`
- X-VS – Nepovinný parametr pro zadání variabilního symbolu. Obsahuje celé číslo.
 - Příklad obsahu parametru: `X-VS:1234567890*`
- X-SS – Nepovinný parametr pro zadání specifického symbolu. Obsahuje celé číslo.
 - Příklad obsahu parametru: `X-SS:1234567890*`
- X-KS – Nepovinný parametr pro zadání konstantního symbolu. Obsahuje celé číslo.
 - Příklad obsahu parametru: `X-KS:1234567890*`

Implementovaná aplikace SPAYD řetězec zpracuje a dále s ním pracuje. Ke zpracování řetězce jsem v rámci této diplomové práce vytvořil open-source knihovnu, která je ve zpracování řetězce nápomocná. Jmenuje se *Spayd parser* a je veřejně

¹⁸ „Kód BIC (Business Identification Code) jednoznačně identifikuje banku v mezinárodním platebním styku (podobně jako kód banky v tuzemském platebním styku). Skládá se z 8 nebo 11 znaků. Je založen na mezinárodní normě ISO 9362. Další informace lze nalézt na webových stránkách www.swift.com.“ [39]

přístupná na serveru Github (<https://github.com/higi90/spayd-parser/>).

Instalaci knihovny je možné realizovat pomocí *composeru* (`composer require higi90/spayd-parser`). Funkcionalitu budu demonstrovat na následujícím příkladu, který je alternativně dostupný i na stránce knihovny na Githubu.

```
<?php

require __DIR__ . '/vendor/autoload.php';

use Higi90\SpaydParser;

$spaydString =
'SPD*1.0*ACC:CZ2806000000000168540115+FIOBCZPP*AM:450.00*CC:CZK*MSG:PL
ATBA ZA ZBOZI*X-VS:1234567890';
$spayd = new SpaydParser\Spayd($spaydString);

echo 'ACC: ' . $spayd->getParam('ACC') . "\n";
echo 'MSG: ' . $spayd->getParam('MSG') . "\n";

$account = $spayd->getAccount();
echo 'IBAN: ' . $account->getIban() . "\n";
echo 'SWIFT: ' . $account->getSwift() . "\n";

if($account instanceof SpaydParser\AccountCzech) {
    echo 'Account number: ' . $account->getAccountNumber() . "\n";
    echo 'Bank code: ' . $account->getBankCode() . "\n";
}
```

Zpracování řetězce počíná vytvořením objektu `higi90\SpaydParser\Spayd` s parametrem konstruktoru, který obsahuje samotný *SPAYD* řetězec. Validace správného formátu řetězce není řešena svépomocí, ale zajišťuje ji server *QR Platba*¹⁹ pomocí *REST API* na adrese <http://api.paylibo.com/paylibo/>.

Po ověření správnosti je řetězec rozdělen na jednotlivé parametry, k nimž lze následně přistupovat pomocí metody `getParam`. Knihovna poskytuje rozšířenou funkcionalitu při práci s bankovním účtem. Metoda `getAccount` vrací potomka abstraktní třídy `higi90\SpaydParser\Account`, dle toho, o jaký účet, v jaké zemi se jedná. Konkrétně v případě tuzemského bankovního účtu vrací instanci třídy `higi90\SpaydParser\AccountCzech`, obsahující metody, které umožňují získání čísla účtu a kódu banky.

Při vložení *SPAYD* řetězce s účtem z jiné země než z České republiky je vrácen objekt třídy `higi90\SpaydParser\AccountDefault`.

¹⁹ QR platba je webová stránka, která poskytuje podporu pro vývojáře, uživatele a výstavce QR plateb. Obsahuje např. generátor QR plateb, grafický manuál implementace a další podpůrné zdroje. [48]

6.6.2 Zadání závazku

Zadání závazku je realizováno prostřednictvím formuláře, umístěného v modálním okně, které lze vyvolat stiskem tlačítka „Přidat závazek“ na stránce „Závazky“.

Modální okno obsahuje horizontální navigaci s různými možnostmi, jakými lze závazek zadat do systému:

1. Ruční zadání závazku;
2. Spayd řetězec;
3. Faktura s QR kódem.

The image displays three variations of the 'Závazek' (Liability) modal form, each with a different active tab in the top navigation bar.

- Top-left screenshot:** The 'Ruční zadání' (Manual entry) tab is active. The form includes fields for 'Název' (Name), 'Popis' (Description), 'Zaplatit z účtu' (Pay from account) with a dropdown menu showing 'firma', 'Částka k uhrazení' (Amount to be paid), 'Datum splatnosti' (Due date), 'Číslo účtu' (Account number) with the value '2001049665', 'Kód banky' (Bank code) with the value '2010', 'Variabilní symbol' (Variable symbol), 'Kontaktní symbol' (Contact symbol), 'Specifický symbol' (Specific symbol), and 'Zpráva pro příjemce' (Message to recipient). A green 'Uložit' (Save) button is at the bottom left, and a 'Zavřít' (Close) button is at the bottom right.
- Top-right screenshot:** The 'Faktura s QR' (Invoice with QR) tab is active. It features 'Název' and 'Popis' fields, 'Zaplatit z účtu' dropdown (showing 'Společný'), and a 'SPAYD řetězec' (SPAYD string) field. A green 'Uložit' button is centered below the string field, and a 'Zavřít' button is at the bottom right.
- Bottom screenshot:** The 'Spayd řetězec' tab is active. It includes 'Název' and 'Popis' fields, 'Zaplatit z účtu' dropdown (showing 'Společný'), and a 'Soubor s fakturou' (Invoice file) field with a 'Vybrat soubor' (Select file) button and the text 'Soubor nevybrán'. A green 'Uložit' button is centered below the file field, and a 'Zavřít' button is at the bottom right.

Obr. 15 Varianty modálního okna pro přidání závazku.

Každá položka menu obsahuje komponentu s formulářem, pomocí něhož může uživatel zadat potřebné údaje. Komponenty jsou umístěny v adresáři `/app/components/payable` a `Countee\Components\Payable-jmenném` prostoru.

První zmíněnou variantou je možnost ručního zadání závazku. Uživatel vyplní všechny náležitosti závazku a určí, z jakého účtu se chystá závazek uhradit. Zadání potvrdí kliknutím na tlačítko uložit. Komponent provede základní kontrolu zada-

ných informací a předá údaje do metody `createCommitment` modelu `Commitment` (`Countee\Model\Payable\Commitment`), který provede uložení závazku do databázové tabulky „`commitment`“.

V případě zadání závazku pomocí *SPAYD* řetězce je situace obdobná. Hlavním viditelným rozdílem pro uživatele je absence informací o samotné platbě (číslo účtu, částka, datum splatnosti atd.), místo kterých je přítomno jedno textové pole, sloužící ke vložení *SPAYD* řetězce. Následné zpracování provádí metoda `createCommitmentSpayd` v modelu `Commitment`. Metoda využívá funkcionality výše popisované knihovny `Spayd parser` k získání informací o náležitostech platby. Výsledek je předán metodě `createCommitment`, která dokončí zadání závazku.

Třetím způsobem je zadání pomocí faktury s QR kódem, u kterého formulář nabízí možnost vložit soubor s fakturou ve formátu PDF. Po odeslání formuláře se přesouvá zpracování opět do modelu `Commitment` prostřednictvím metody `createCommitmentInvoice`. V metodě je provedeno uložení faktury na disk a zahájení jejího dalšího zpracování.

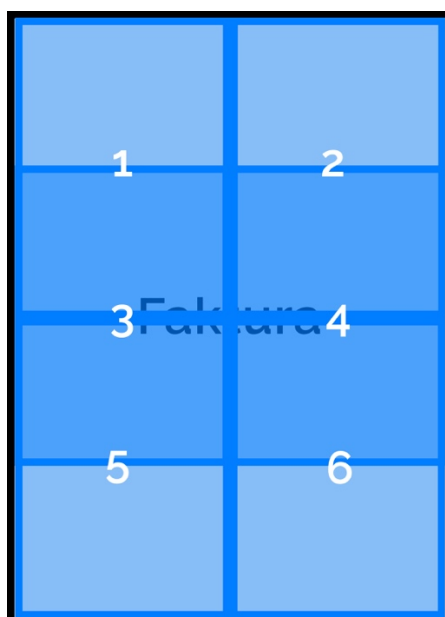
Zjištění, zda faktura obsahuje QR kód a jestli je obsahem QR kódu *SPAYD* řetězec, zajišťuje třída `Parse` (`Countee\Model\Invoice\Parse`). Postup, pomocí kterého je z faktury přečtena hodnota QR kódu je následující:

1. Převod PDF do JPG

Faktura je převedena do rastrového formátu JPG pomocí příkazu `exec("/bin/bash -lc 'convert -density 200 -trim " . $fileLocation . " -quality 100 " . $jpgPath . "'");`. K samotnému převodu je využita funkce `convert`, která je součástí balíku programu `ImageMagick`. Důležitým parametrem funkce `convert` je `density`, který určuje, kolik DPI bude mít výsledný rastrový obraz. Proces čtení QR kódu vykazoval lepší výsledky při hodnotě 200. Pokud byla hodnota DPI nižší, docházelo k chybovosti v určení QR kódu.

2. Rozdělení JPG

Vzhledem k tomu, že je zvolený postup dekodování QR kódu z rastrového obrazu velmi náročný na paměť PHP, rozhodl jsem se při zpracování soubor rozdělit na 6 menších částí (výřezů), které se budou následně dekodovat zvlášť. Každý výsledný obrázek má čtvrtinovou velikost oproti původnímu obrázku – celý obraz by stačilo teoreticky rozdělit pouze na čtyři části. Přidat další dvě stejně velké části jsem se rozhodl z důvodu většího pokrytí střední části obrazu, a tím snížení pravděpodobnosti znehodnocení QR kódu jeho „ustřihnutím“.



Obr. 16 Rozdělní faktury na výřezy

3. Detekce QR kódu a získání SPAYD řetězce

K samotné detekci QR kódu jsem použil PHP knihovnu `khanami-ryan/qrcode-detector-decoder`. Knihovna obsahuje třídu `QrReader`, která v konstruktoru načte cestu k obrázku. Pokud je hledání QR kódu úspěšné, metoda `text` vrací řetězec s jeho obsahem. V opačném případě vrací `false`. Následující ukázka zdrojového kódu implementované aplikace zobrazuje proces hledání QR kódu na faktuře.

```
<?php

foreach ($dividedImages as $dividedImage) {
    $qrcode = new \QrReader($dividedImage);
    $text = $qrcode->text();

    if (!empty($text)) {
        $spayd = $text;
        break;
    }
}
```

4. Úklid

Po provedené detekci je proveden úklid po jejich zpracování již nepotřebných souborů. Jedná se o soubor s fakturou ve formátu PDF, převedenou fakturu do JPG a 6 výřezů. Tyto soubory jsou nyní již nepotřebné, a proto dojde k jejich smazání.

Pokud bylo hledání QR kódu úspěšné, dodá třída `Parse` pomocí metody `getSpayd` modelu `Commitment` *SPAYD* řetězec. Následným zavoláním metody `createCommitmentSpayd` je závazek zadán do systému.

6.6.3 Změna závazku

Zadaný závazek je možné měnit. Využívá k tomu stejnou formulářovou komponentu jako v případě ručního zadání závazku.

6.6.4 Úhrada závazku

U každého zadaného závazku lze též provést jeho úhradu. Správu procesu úhrady zajišťuje model Commitment metodou `pay()`. Metoda získá objekt účtu banky, k níž se vztahuje účet, ze kterého je závazek hrazen. Každý bankovní účet musí implementovat rozhraní IAccount. Rozhraní vyžaduje po třídě implementovat metodu `payment`, která platbu provede. Po provedení příkazu k platbě je závazek postoupen k zpracování na straně banky.

6.7 Správa pohledávek

Aplikace poskytuje jednoduché rozhraní i pro správu pohledávek, kterou zajišťuje Receivable model (*Countee\Model\Payable\Receivable*).

Pohledávku lze vytvořit kliknutím na tlačítko „Přidat pohledávku“ na obrazovce s výpisem pohledávek. Přidání pohledávky provádí metoda `createReceivable` modelu Receivable.

6.8 Párování plateb

Důležitou funkcionalitou je možnost platby párovat. Při každé synchronizaci bankovního účtu aplikace ověří, zda nově zadané transakce nesouvisí s úhradou závazků nebo pohledávek. Pokud tomu tak je, aplikace skutečnost o úhradě v aplikaci zohlední. Pokud není závazek nebo pohledávka uhrazena celá, je zadána pouze částečná úhrada. Uchování stavu úhrady zajišťují tabulky „commitment“ pro závazky a „receivable“ pro pohledávky.

6.9 Šablona HTML

HTML šablonu jsem vytvořil na základě grafického návrhu, který je popsán v předchozí kapitole. Šablona se skládá z HTML kódu, CSS stylů a JavaScriptu. HTML kód je rozmístěn v šablonách (view), které jsou ve složkách `/app/components` a `/app/presenter/templates`. V šablonách využívám šablonovací systém Latte, který usnadňuje provádění opakovaných úkonů a pomáhá se zabezpečením bezpečnostních zranitelností (např.: XSS²⁰).

²⁰ XSS je zkratka pro Cross Site Scripting, což je široce rozšířená zranitelnost, která umožňuje útočníkovi vložit škodlivý kód (převážně v jazyce JavaScript) na webové stránky oběti. Pomocí tohoto kódu pak může s napadenou stránkou manipulovat, např. zjistit heslo zadané do formuláře. [11]

Zdrojové soubory CSS stylů, Javascriptů a SVG grafiky jsou umístěny do adresáře /frontend. V šabloně využívám další knihovny, které spravuji pomocí nástroje *Bower*²¹:

- JQuery – z důvodu kompatibility využita verze 1.12;
- SmoothScroll – JavaScriptová knihovna pro hladký průchod stránkou;
- Bootstrap – verze 3.3.7;
- Selectivizr – JavaScriptová knihovna, která emuluje CSS 3 pseudo třídy na starších prohlížečích;
- HTML5shiv – knihovna pro podporu HTML elementů na starších prohlížečích.

Pro vytvoření CSS využívám preprocesor LESS. Tento preprocesor rozšiřuje základní funkcionalitu CSS o další vylepšení a techniky, které činí CSS lépe udržovatelný a rozšiřitelný.[28] Zdrojové kódy v LESS jsou umístěny v adresáři /frontend/less. Adresář obsahuje soubory:

- Style.less – Style je kořenový soubor. Součástí jsou nejdůležitější definice CSS a zahrnutí dalších separovaných LESS souborů. Zahrnuté soubory obsahují další definice stylů a funkcionalit.
 - Button.less – definice tlačítek;
 - Fonts.less – import fontů a definice proměnných pro jejich používání;
 - Header.less – styly pro hlavičku stránky;
 - Login.less – definice stylů pro stránku s přihlášením;
 - Main.less – většina stylů pro aplikaci (převážně pro hlavní stranu po přihlášení);
 - Mixins.less – vytvořené mixiny;
 - Typography.less – globální definice typografických pravidel, zejména nadpisů a odstavců;
 - Variables.less – soubor obsahuje definice proměnných pro barvy media query nebo rozměry.

JavaScriptový kód je umístěn ve složce frontend/js, v níž je v podsložce script je umístěna vlastní implementovaná funkcionalita. V podsložce library jsou vložené knihovny, které nelze spravovat přes balíčkovací systém *Bower*.

Styly ve formátu LESS a JavaScriptový kód takto rozdělený do složek je nutné přeložit, spojit a následně minifikovat. Přeložením LESS souborů vznikne CSS kód, který je schopen webový prohlížeč zpracovat. Minifikace se provádí z toho důvodu, aby se snížila velikost souborů, které jsou přenášeny k uživateli. Spojením souborů se minimalizuje i počet dotazů na server. Následně tak dojde ke zvýšení rychlosti načtení stránky (hlavně u použitého protokolu HTTP 1.0.). K tomu, aby bylo možné celý proces automatizovat, jsem využil nástroj Grunt. Grunt slouží k automatizaci nejčastějších úkolů, jako výše zmíněné spojení souborů, minifikace nebo i překlad

²¹ Bower je správce balíčků, které obsahují zejména HTML, CSS nebo Javascripty. [86]

LESS do CSS. Definice úkolů je provedena v souboru `/frontend/Gruntfile.js`. Po zpracování souborů jsou výsledné soubory uloženy do složky `/www/assets`.

6.10 Testovací instalace

K testovací instalaci jsem zvolil *VPS*²² s operačním systémem Debian 8 od společnosti Wedos Internet, a. s. Přidělený hardwarový výkon *VPS* je následující:

- 30 GB SSD pevný disk;
- 4096 MB RAM;
- 1 procesor.

Provoz aplikace je řešen pomocí Docker containeru s připojeným lokálním úložištěm serveru, ze kterého čerpá zdrojové soubory aplikace.

Server podporuje přístup pomocí protokolu *HTTPS*. Použitý SSL certifikát je vystavený autoritou Lets Encrypt.

Testovací instalace je přístupná na doméně `countee.cz`.

²² VPS je zkratka anglických slov Virtualized Private Server. Jedná se virtuální server na sdíleném hosingu. [85]

7 Možnosti monetizace aplikace

V kapitole monetizace aplikace se budu zabývat otázkou, jakým způsobem získat prostředky z provozu webových stránek, resp. webových aplikací.

Možností zpeněžení provozu webových stránek existuje celá řada. Jako příklad lze uvést [34]:

- Affiliate Marketing;
- PPC;
- CPM;
- zpoplatnění aplikace.

7.1 Affiliate marketing

Affiliate marketing funguje na principu doporučení produktů návštěvníkům stránek. Tvůrce webových stránek si nejdříve vyhledá produkt, u něhož uzná za vhodné ho návštěvníkům webu doporučit. Následně produkt na svých stránkách propaguje. Pokud si zákazník produkt zakoupí prostřednictvím referenčního odkazu, který byl na webových stránkách umístěn, dostane ze zmíněného nákupu provizi. [34]

Provize z prodeje může být poskytována buď jako procento z kupní ceny za prodané zboží či služby (na českém trhu se pohybuje od 1 % do 70 %), nebo jako konkrétní částka za prodanou položku (provize za prodej půjčky/sjednání půjčky apod.). [70]

Vlastník webových stránek dostane odměnu za provedení nákupu nebo registraci návštěvníka, který přichází z jeho webových stránek. Odměna není tedy závislá na návštěvnosti webových stránek (např. model CPM) ani na počtu prokliků (např. model PPC). [2]

Mezi výhody Affiliate marketingu bych zařadil hlavně nezávislost na počtu návštěvníků nebo počtu kliků na reklamu. Návštěvnost webu nemusí být velká – i přesto může vlastník webových stránek získat plnění (za provedenou akci) v podobě provize z prodeje produktu nebo registrace. Záleží jenom na tom, kolik návštěvníků inzerovaný produkt či služba zaujme a provedou nákup na stránce inzerenta.

Možný problém affiliate marketingu spočívá v závislosti aplikace na nastavení prohlížeče zákazníka. Jedná se o nutnost uživatele webových stránek mít zapnutou možnost ukládání cookies do svého webového prohlížeče. Pokud má uživatel tuto možnost blokovanou, nemusí dojít při nákupu přes referenční odkaz k identifikaci tvůrce webu, a tak provizi z prodeje nezíská. Další nevýhoda se týká toho, že ze samotné propagace výrobku nebo služby neplyne plnění žádné. Uživatel musí přijít pomocí referenčního odkazu na web prodejce a provést požadovanou akci. [55]

V následující tabulce jsou uvedeny příklady českých affiliate programů, které tematicky souvisí s aplikací Countee. Cílem je vytvoření představy o situaci na trhu a výších peněžního plnění.

Název programu	Předmět činnosti	Hodnota	Popis
CreditON.com	Poskytování půjček	250 Kč	Provize je získána za uzavřenou půjčku. Registrační bonus činí 100 Kč
CreditOn.cz	Poskytování půjček	od 500 Kč	Provize je získána za uzavřenou půjčku. Výplata provizí je možná od 1000 Kč.
Heuréka.cz	Srovnávač cen zboží	100 %	Hodnota provize je 100 % z ceny prokliku do obchodů. Výplata získaných provizí je možná od 500 Kč.
HypotékyBezRegistru.cz	Hypotéky	2 %	Výše provize je vypočítána z hodnoty úvěru.
PrávoProPodnikatele.cz	Právní služby	10 %	Provize je získána z každé provedené objednávky. Výplata získaných provizí je možná od 500 Kč.
ZENRON.cz	Offshore, firmy v Anglii, daňová optimalizace	1500–3000 Kč	Provize je získána z provedené objednávky. Výplata provizí je možná po získání prvních 1000 Kč.

Tab. 2 Přehled affiliate programů [70]

Pokud by byl jako možnost monetizace zvolen např. affiliate program Hypoteky-BezRegistru.cz a uživatel aplikace Countee si prostřednictvím affiliate odkazu pořídil úvěr za 500 000 Kč, tak by ze zprostředkovaného prodeje plynula provize ve výši 2 % z výše úvěru, tedy 2 % z 500 000 Kč = 10 000 Kč.

7.2 PPC

Pay per click (PPC) neboli platba za proklik. Prostřednictvím tohoto modelu monetizace získá vlastník webu peníze podle počtu kliků na reklamu umístěnou na jeho stránkách. Jako příklad lze uvést službu Adsence od Google nebo český Sklik od společnosti Seznam.cz a.s. [55]

Fungování modelu je u obou zmíněných služeb velmi podobný. Po integraci služby jsou na webové stránky zobrazovány reklamy, které ve většině případů souvisejí s obsahem daného webu. Pokaždé, když uživatel stránek klikne na zobrazenou reklamu, provozovatel webu obdrží provizi. [34]

Konkrétně u služby Sklik závisí hodnota výdělku na počtu prokliků, jejichž cena je odvozena od oblíbenosti obsahu webu a řádově se pohybuje v jednotkách korun českých za klik. [61]

7.3 CPM

Cost per mile (CPM) neboli cena za tisíc zobrazení. Provozovatel webu obdrží provizi za každých tisíc zobrazení reklamy návštěvníkům webu. Vlastník webu získává provizi bez ohledu na to, zda na reklamu lidé kliknou či ne. Výdělek prostřednictvím tohoto modelu je přímo úměrný návštěvnosti webu – je tedy vhodný pro weby s velkou návštěvností. V případě nízké návštěvnosti je výdělek téměř nulový. Například pokud má stránka návštěvnost 10 000 uživatelů za měsíc a sazba je stanovena na 10 Kč, vlastník webu tak získá za měsíc odměnu 100 Kč. [2]

7.4 Zpoplatnění aplikace

Další možností, jak aplikaci monetizovat, je její zpoplatnění. Následující rozdělení vychází z obchodních modelů pro aplikace na Apple Appstore [36]:

- Freemium – aplikace je poskytována v základní funkcionalitě zdarma. Za poplatek jsou nabízeny další rozšiřující funkce;
- Předplatné – uživatel hradí užívání aplikace pomocí pravidelných plateb;
- Placená aplikace – samotné užívání je zpoplatněno určitou částkou. Jakmile uživatel částku uhradí, tak má k dispozici veškerou funkcionalitu aplikace bez omezení;
- Paymium – spojení placené aplikace s možností přikupovat další rozšiřující funkce aplikace.

7.5 Návrh monetizace aplikace

Na základně výše popsaných možností považuji za nejvhodnější metodu monetizace kombinaci affiliate marketingu a zpoplatnění aplikace pomocí předplatného.

Pokud se uživatelův stav na účtech dostane pod určitou úroveň, je mu možné nabídnout affiliate reklamu úvěrových produktů. Když se uživatel rozhodne navý-

šit svůj stav účtu úvěrem, o který zažádá prostřednictvím referenčního odkazu, aplikace obdrží provizi za poskytnutý úvěr provizi.

Model předplatného by mohl počítat s určitým omezením základní funkcionality aplikace. Pokud by uživatelé bez předplatného mohli synchronizovat pouze jeden bankovní účet, stačí to na vyzkoušení funkcionality aplikace. Uživatel může obstarávat bankovních účtů více. Využije k tomu možnost předplatného, které mu umožní přidávat účty další.

8 Možnosti gamifikace aplikace

Gamifikace je způsob, jakým lze zapojit publikum do řešení problému pomocí využití herních mechanik v neherním prostředí [45]. Princip gamifikace pracuje hlavně s motivací a snaží se o zatraktivnění relativně méně záživných a dlouhodobých činností. V praxi se lze setkat např. s gamifikací, aplikací pro podporu pravidelného cvičení nebo dobrých pracovních návyků [78].

Důležitým činitelem pro pochopení principů motivace a potažmo gamifikace jako prostředku k jejímu vyvolání je látka dopamin. Jedná se o neurotransmitter v mozku [23]. Základní premisou je, že tělo uvolňuje dopamin při pocitu štěstí a slasti. Tyto pocity jsou vyvolány právě tehdy, pokud jedinec dosáhne svého cíle nebo při obdržení odměny. Čím více je dosaženo cílů a získáno odměn, tím více a častěji je dopamin uvolňován. Potom je výrazně snazší zůstat motivovaný a pokračovat v započaté činnosti. [78].

Motivaci vykonávat jakoukoli činnost je možné rozdělit na dvě roviny – externí motivace a interní motivace. Externí motivace je vyvolána externími faktory, které na jedince působí. Patří mezi ně např. peněžní odměna, získání titulu za studium, hrozba kontroly z finančního úřadu a podobně. Naopak interní motivace pracuje na bázi vlastního pocitu jedince nutnosti a prospěšnosti. Jedná se hlavně o jeho vlastní zájem o problematiku nebo v případě, že je prováděná činnost pro jedince zábavná. [41]

8.1 Využití gamifikace v existujících aplikacích

Princip gamifikace v existujících aplikacích, které se zabývají prací s financemi, není v současnosti ojedinělý jev. V následujícím přehledu uvedu několik aplikací, které gamifikaci již využívají.

8.1.1 Internetové bankovníctví mBank

mBank použila gamifikaci jako prostředek k oživení svého internetového bankovníctví. Zaměřila se hlavně na vzdělávání uživatelů ve funkcích bankovníctví a na větší zapojení uživatelů do procesu jeho užívání. Mezi mechaniky, které banka ke gamifikaci zvolila, patří:

- možnost získání bodů;
- možnost získání odznaků;
- žebříčky;
- odměny ve virtuální měně.

Uživatel obdrží za splněné úkoly body, na základě kterých následně získá různé typy odznaků. [62]

8.1.2 SaveUP

Aplikace SaveUp je navržena jako pomocník při placení dluhů a tvorbě úspor. Je poskytována zdarma ve formě webové a mobilní aplikace pro platformu iOS. Gamifikace aplikace se vztahuje k motivaci uživatele naplňovat své finanční cíle. Za každý dolar, který přispěje k naplnění finančního cíle, obdrží uživatel body, které může následně vyměnit za možnost hrát o skutečné výhry. Aplikace nabízí výhry až za 2 000 000 dolarů. [31]

8.1.3 Contango

Hlavním účelem mobilní aplikace Contango je naučit hravou formou své uživatele investovat na burze. Uživatel neinvestuje své reálné prostředky – jedná se jen o hru s fiktivními prostředky v prostředí burzy, v němž jsou ceny i názvy firem reálné. Nemůže tím pádem dojít k reálné finanční ztrátě vlastních prostředků.

Na základě výkonu na burze je uživateli přidělena určitá úroveň. Tato přidělená úroveň není konečná – k navýšení dochází při zvyšování dovedností uživatele. Spolu s navyšováním úrovně aplikace se postupně odemykají skutečné výhry. Jedná se např. o dárkové karty, filmy a cestování. [31]

8.1.4 EpicWin

Mobilní aplikace EpicWin není primárně zaměřena na osobní finance. EpicWin nabízí uživateli možnost zadat úkoly, cíle nebo osobní výzvy [31]. Aplikace je simuluje *RPG*²³ prostředí známé z počítačových her [31]. Za každé splnění vytyčeného úkolu se uživatel posouvá ve hře kupředu[62]. Uživatel postupně zlepšuje úroveň a dovednosti své postavy. [31]

8.2 Využití gamifikace v implementované aplikaci

Rozšíření funkcionality implementované aplikace o gamifikaci by mohlo být realizováno pomocí plnění úkolů, které by podporovaly dobré finanční návyky. Tyto úkoly by mohly být např.:

- včasné uhrazení závazku;
- důsledná kategorizace příjmů a výdajů;
- kladný stav na bankovním účtu.

Za splnění úkolů může uživatel obdržet body zkušeností, na základě kterých dochází k zvýšení úrovně uživatele. Body zkušeností by šly směnit za určité časové období placeného předplatného, které bylo navrženo jako možnost monetizace v kapitole 7.

²³ RPG je zkratka anglických slov Role Playing Game. Jedná se o hru, při níž se hráč vžívá do role herní postavy. [65]

9 Závěr

V první části práce jsem se věnoval stanovení cílové skupiny a jejích požadavků na správu osobních financí. Stanovené požadavky se týkaly možnosti správy příjmů, výdajů, závazků a pohledávek. Se znalostí požadavků jsem mohl posléze provést zhodnocení současného stavu na českém trhu s webovými aplikacemi pro správu osobních financí. Hodnotil jsem celkem 4 aplikace – sMoneyBox, eÚčty, iDoklad a Fakturoid. Všechny stanovené podmínky najednou nesplnila žádná aplikace, a proto jsem se rozhodl navrhnout a implementovat vlastní řešení.

Jednou ze zásadních částí této práce je synchronizace transakcí uživatele s jeho bankovním účtem. Srovnávané webové aplikace nabízejí dvě možnosti: ruční zadání transakcí a zpracování výpisů z banky. Třetí variantou, nad kterou jsem uvažoval, bylo napojení pomocí bankovního *API*. Tento způsob synchronizace poskytuje možnost nejen získat informace o příjmech a výdajích na účtu, ale lze i zadat novou platbu. Důvod, proč stávající aplikace služeb *API* bank zatím nevyužívají, je pravděpodobně v nízké podpoře ze strany samotných bank. V současnosti přístup pomocí *API* podporuje z tuzemských bank pouze Fio banka a Česká spořitelna. Situaci výrazně změní nová směrnice evropské unie – PSD2. Ta udává povinnost bankám zpřístupnit svá bankovníctví aplikacím třetím stran a s největší pravděpodobností bude zpřístupnění provedeno pomocí *API*. Kvůli možnostem a budoucí očekávané implementaci dalšími tuzemskými bankami jsem se rozhodl tento způsob využít v dalších částech práce.

V návaznosti na výběr metody synchronizace s bankou jsem provedl návrh a implementaci nové aplikace. Aplikace řeší problémy správy osobních financí cílové skupiny a využívá *API* Fio banky pro synchronizaci transakcí. Při tvorbě jsem kladl důraz na snadnou rozšiřitelnost aplikace o komunikaci s dalšími *API*, až budou dostupná – aplikace je tak připravena na účinnost směrnice PSD2.

Výslednou aplikaci jsem implementoval pomocí programovacího jazyka PHP a framewroku Nette. Frontend je vytvořen s pomocí technologií HTML 5, Bootstrap 3, CSS 3, LESS a Grunt. Z funkčního hlediska aplikace nabízí možnost synchronizovat příjmy a výdaje s bankovním účtem, spravovat závazky a pohledávky. Zajímavou funkcionalitou je možnost kategorizace příjmů a výdajů nebo možnost zadání závazku ve formě faktury s QR kódem. Testovací instalace aplikace je přístupná na doméně countee.cz

V průběhu implementace jsem se rozhodl vyčlenit část aplikace pro práci se SPAYD řetězcem a publikovat ji jako open-source knihovnu na serveru Github (higié/spayd-parser). Knihovna disponuje rozšířenou funkcionalitou pro práci s českými bankovními účty a je dostupná pomocí composeru.

10 Literatura

- [1] 10 questions about banking API. Kontomatik [online]. 2016 [cit. 2016-12-12]. Dostupné z: <http://kontomatik.com/post/10-questions-about-banking-api>
- [2] 5 Great Website Monetization Ideas That Will Guarantee Profit. Adngin [online]. [cit. 2017-05-20]. Dostupné z: <https://www.adngin.com/blog/adsense-tips/5-great-website-monetization-ideas-that-will/>
- [3] A Beginner's Guide to Wireframing. EdrawSoft [online]. [cit. 2017-05-19]. Dostupné z: <https://www.edrawsoft.com/wireframing-guide.php>
- [4] A multi-tier architecture for building RESTful Web services. DeveloperWorks [online]. [cit. 2017-05-19]. Dostupné z: <https://www.ibm.com/developerworks/library/wa-aj-multitier/>
- [5] Alegreya Sans. Google Fonts [online]. [cit. 2017-05-19]. Dostupné z: <https://fonts.google.com/specimen/Alegreya+Sans>
- [6] An Introduction to OAuth 2. DigitalOcean [online]. [cit. 2017-05-19]. Dostupné z: <https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>
- [7] API Bankovníctví. *Fio Banka* [online]. [cit. 2016-12-12]. Dostupné z: <http://www.fio.cz/bankovni-sluzby/api-bankovnictvi>
- [8] Application program interface (API). Techtarget.com [online]. 2014 [cit. 2016-12-10]. Dostupné z: <http://searchexchange.techtarget.com/definition/application-program-interface>
- [9] ARES - Ekonomické subjekty. Ministerstvo financí [online]. [cit. 2017-05-19]. Dostupné z: http://www.info.mfcr.cz/ares/ares_es.html.cz
- [10] ARMSTRONG, Gary a Philip. KOTLER. Marketing: an introduction. 7th ed. Upper Saddle River, NJ: Pearson/Prentice Hall, c2005. ISBN 0131424106.
- [11] Ask Sucuri: What is an XSS Vulnerability? SucuriBlog [online]. [cit. 2017-05-19]. Dostupné z: <https://blog.sucuri.net/2016/04/what-is-an-xss-vulnerability.html>
- [12] Authentication. Instagram [online]. [cit. 2017-05-19]. Dostupné z: <https://www.instagram.com/developer/authentication/>
- [13] Avatar. Techopedia [online]. [cit. 2017-05-19]. Dostupné z: <https://www.techopedia.com/definition/4624/avatar>

- [14] Bootstrap 3 Tutorial. W3schools [online]. [cit. 2017-05-19]. Dostupné z: <https://www.w3schools.com/bootstrap/>
- [15] CO JE CÍLOVÁ SKUPINA? TRIAD Advertising BLOG [online]. [cit. 2017-05-19]. Dostupné z: <https://blog.triad.cz/marketingovy-slovník/co-je-cielova-skupina/>
- [16] Command line interface (CLI). SearchWindowsServer [online]. [cit. 2017-05-19]. Dostupné z: <http://searchwindowsserver.techtarget.com/definition/command-line-interface-CLI>
- [17] Create, Retrieve, Update and Delete (CRUD). Techopedia [online]. [cit. 2017-05-19]. Dostupné z: <https://www.techopedia.com/definition/25949/create-retrieve-update-and-delete-crud>
- [18] CSS. Bootstrap [online]. [cit. 2017-05-19]. Dostupné z: <http://getbootstrap.com/css/>
- [19] Česká spořitelna nabídne klientům přístup k otevřeným datům. Elektronickou složenku tak zaplatíte bezpečně a jedním klikem. Investujeme [online]. [cit. 2017-05-19]. Dostupné z: <http://www.investujeme.cz/tiskove-zpravy/ceska-sporitelna-nabidne-klientum-pristup-k-otevrenym-datam-elektronickou-slozenku-tak-zaplatite-bezpecne-a-jednim-klikem/>
- [20] Češi platí kartou pětkrát více než před 10 lety, hlavně bezkontaktně. Finance.cz [online]. [cit. 2017-05-19]. Dostupné z: <https://www.finance.cz/473544-platby-kartou-pribyvaji/>
- [21] Defining Breakpoints. Responsivedesign [online]. [cit. 2017-05-19]. Dostupné z: <https://responsivedesign.is/strategy/page-layout/defining-breakpoints/>
- [22] Dej kus banky do své appky!. API Portál - Česká spořitelna [online]. [cit. 2017-05-19]. Dostupné z: <https://developers.csas.cz/>
- [23] Dopamin v mozku – na co je a jak zvýšit jeho hladinu? Rehabilitace [online]. [cit. 2017-05-19]. Dostupné z: <http://www.rehabilitace.info/zdravotni/dopamin-v-mozku-na-co-je-a-jak-zvysit-jeho-hladinu/>
- [24] FAQ. SMoneybox [online]. [cit. 2017-05-19]. Dostupné z: <https://smoneybox.com/cs/stranky/faq>
- [25] FIO API Bankovníctví. FIO Banka [online]. [cit. 2016-12-12]. Dostupné z: http://www.fio.cz/docs/cz/API_Bankovnictvi.pdf
- [26] FIO API Bankovníctví. FIO Banka [online]. [cit. 2016-12-12]. Dostupné z: http://www.fio.cz/docs/cz/API_Bankovnictvi.pdf
- [27] Fio banka nabízí bezpečné automatizované získávání dat z účtů. Fio Ban-

- ka* [online]. 2012 [cit. 2016-12-12]. Dostupné z: <http://www.fio.cz/spolecnost-fio/media/tiskove-zpravy/122142-fio-banka-nabizi-bezpecne-automatizovane-ziskavani-dat-z-uctu>
- [28] Getting started: An overview of Less, how to download and use, examples and more. Less [online]. [cit. 2017-05-19]. Dostupné z: <http://lesscss.org/>
- [29] GUNTER, Barrie. a Adrian. FURNHAM. Consumer profiles: an introduction to psychographics. 0th Edition. New York: Routledge, 1992. ISBN 0415075343.
- [30] Guzzle Documentation. Docs guzzle php [online]. [cit. 2017-05-19]. Dostupné z: <http://docs.guzzlephp.org/en/latest/>
- [31] Have Fun with Finances: Level-Up Your Finances Using Gamification. Ally [online]. [cit. 2017-05-20]. Dostupné z: <https://www.ally.com/do-it-right/money/have-fun-with-finances-level-up-your-finances-using-gamification/>
- [32] Historie: Příběh Fio banky. Fio banka [online]. [cit. 2017-05-19]. Dostupné z: <https://www.fio.cz/o-nas/fio-banka/historie>
- [33] HTTP. Computer Hope [online]. [cit. 2017-05-19]. Dostupné z: <https://www.computerhope.com/jargon/h/http.htm>
- [34] <https://websitesetup.org/33-ways-to-monetize-website/>
- [35] Cheatsheet. Font Awesome [online]. [cit. 2017-05-19]. Dostupné z: <http://fontawesome.io/cheatsheet/>
- [36] Choosing a Business Model. Developer [online]. [cit. 2017-05-20]. Dostupné z: <https://developer.apple.com/app-store/business-models>
- [37] Choosing Great Logo Colors. The logo factory [online]. [cit. 2017-05-19]. Dostupné z: <http://www.thelogofactory.com/choosing-great-logo-colors-help-brand-selection/>
- [38] IBAN – mezinárodní formát čísla účtu. Česká národní banka [online]. [cit. 2017-05-19]. Dostupné z: https://www.cnb.cz/cs/platebni_styk/iban/iban_mezinar_cislo_uctu.html
- [39] IBAN a BIC - využití v platebním styku. Česká národní banka [online]. [cit. 2017-05-19]. Dostupné z: https://www.cnb.cz/cs/platebni_styk/iban/iban_vyuziti.html
- [40] iDoklad. Náповěda iDokladu [online]. [cit. 2017-05-19]. Dostupné z: <https://www.idoklad.cz/pub/Help/index.html>
- [41] Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. Pbworks [online]. [cit. 2017-05-19]. Dostupné z: <http://mmrg.pbworks.com/f/Ryan,+Deci+00.pdf>
- [42] Introduction. Getcomposer [online]. [cit. 2017-05-19]. Dostupné z: <https://getcomposer.org/doc/00-intro.md>

- [43] Jaké novinky přinese PHP 7. Zdroják [online]. [cit. 2017-05-19]. Dostupné z: <https://www.zdrojak.cz/clanky/jake-novinky-prinese-php-7/>
- [44] JOHNSON, Jeff. *Designing with the mind in mind: simple guide to understanding user interface design rules*. Boston: Morgan Kaufmann Publishers/Elsevier, c2010. ISBN 978-0-123-75030-3.
- [45] KAPP, Karl M. and Carolyn CHANDLER. *The gamification of learning and instruction: game-based methods and strategies for training and education*. San Francisco, CA: Pfeiffer, 2012. ISBN 9781118096345
- [46] Kdyby/Facebook. GitHub [online]. [cit. 2017-05-19]. Dostupné z: <https://github.com/Kdyby/Facebook>
- [47] Komunikace mezi poskytovateli – třetími stranami a bankami. Jaký bude dopad na technologie? PwC [online]. [cit. 2017-05-19]. Dostupné z: <https://www.pwc.com/cz/cs/bankovnictvi/assets/psd2-v-kostce-n02-cz.pdf>
- [48] Konec přepisování platebních údajů z faktur. QR Platba [online]. [cit. 2017-05-19]. Dostupné z: <http://qr-platba.cz/>
- [49] KOTLER, Philip. a Kevin Lane KELLER. *Marketing management*. 13th ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2009. ISBN 0136009980.
- [50] KRUG, Steve. *Don't make me think, revisited: a common sense approach to Web usability*. Third edition. ISBN 978-0-321-96551-6.
- [51] LUBBERS, P. -- ALBERS, B. -- SALIM, F. *HTML5: programujeme moderní webové aplikace*. 1. vyd. Brno: Computer Press, 2011. 304 s. ISBN 978-80-251-3539-6.
- [52] MariaDB 10.1 and MySQL 5.7 performance on commodity hardware. MariaDB [online]. [cit. 2017-05-20]. Dostupné z: <https://mariadb.org/maria-10-1-mysql-5-7-commodity-hardware/>
- [53] MBank HTTP API parser pro výpis transakcí – PHP skript. Api.teplaky.net [online]. [cit. 2017-05-19]. Dostupné z: <https://api.teplaky.net/mbank-http-api/>
- [54] Model-View-Presenter (MVP). Nette [online]. [cit. 2017-05-19]. Dostupné z: <https://doc.nette.org/cs/0.9/model-view-presenter>
- [55] Monetizace webu – aneb jak pasivně vydělat peníze. Netzin [online]. [cit. 2017-05-20]. Dostupné z: <http://www.netzin.cz/monetizace-webu-aneb-jak-pasivne-vydelat-penize>
- [56] Name that Color. Chir.ag [online]. [cit. 2017-05-19]. Dostupné z: <http://chir.ag/projects/name-that-color/>
- [57] Phinx Documentation. Docs [online]. [cit. 2017-05-19]. Dostupné z: <http://docs.phinx.org/en/latest/>
- [58] PICKTON, David a Amanda BRODERICK. *Integrated marketing communications*. 2nd ed. Upper Saddle River, N.J.: Prentice Hall Financial Times, 2005. ISBN 0273676458.
- [59] Podpořte vývoj služby eÚčty.cz. eÚčty.cz [online]. [cit. 2017-05-19]. Dostupné z: <http://eucty.cz/SupportUs.aspx>

- [60] Popis MVC architektury. ITnetwork [online]. [cit. 2017-05-19]. Dostupné z: <https://www.itnetwork.cz/php/mvc/objektovy-mvc-redakcni-system-v-php-popis-architektury>
- [61] POROVNÁNÍ SKLIK A ADSENSE. Webcentral [online]. [cit. 2017-05-20]. Dostupné z: <http://webcentral.cz/vyhody-sklik/> <http://webcentral.cz/vyhody-sklik/>
- [62] PŘÍKOPOVÁ, Monika. Gamifikace z pohledu designu a vytvoření online výukového průvodce. Brno, 2016. Diplomová práce. Masarykova univerzita.
- [63] PSD2 v kostce: Detailní pohled na novou evropskou směrnici. PwC [online]. [cit. 2017-05-19]. Dostupné z: <http://www.pwc.com/cz/cs/bankovnictvi/psd2.html>
- [64] PSD2-v-kostce. PwC [online]. [cit. 2017-05-19]. Dostupné z: <https://www.pwc.com/cz/cs/bankovnictvi/assets/psd2-v-kostce-n01-cz.pdf>
- [65] RPG. Urban Dictionary [online]. [cit. 2017-05-20]. Dostupné z: <http://www.urbandictionary.com/define.php?term=RPG>
- [66] Rubik. Google Fonts [online]. [cit. 2017-05-19]. Dostupné z: <https://fonts.google.com/specimen/Rubik>
- [67] Sandbox Netbanking V3. API Portál - Česká spořitelna [online]. [cit. 2017-05-19]. Dostupné z: <https://developers.csas.cz/docs/netbanking-v3#/introduction/basic-mechanisms>
- [68] SDK - software development kit. Webopedia [online]. [cit. 2017-05-19]. Dostupné z: <http://www.webopedia.com/TERM/S/SDK.html>
- [69] Service Providers. SpringSource [online]. [cit. 2017-05-19]. Dostupné z: <http://www.springframework.net/social/refdoc/serviceproviders.html>
- [70] Seznam affiliate programů. Mladý podnikatel [online]. [cit. 2017-05-20]. Dostupné z: <https://mladypodnikatel.cz/seznam-affiliate-programu>
- [71] SIL Open Font License (OFL). NRSI: Computers & Writing Systems [online]. [cit. 2017-05-19]. Dostupné z: http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL
- [72] Specifikace formátu. QR Platba [online]. [cit. 2017-05-19]. Dostupné z: <http://qr-platba.cz/pro-vyvojare/specifikace-formatu/>
- [73] Stopařův průvodce REST API. ITnetwork [online]. [cit. 2017-05-19]. Dostupné z: <https://www.itnetwork.cz/nezarazene/stoparuv-pruvodce-rest-api>
- [74] The Best PHP Framework for 2015: SitePoint Survey Results. SitePoint [online]. [cit. 2017-05-19]. Dostupné z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
- [75] The Color Green. Empower yourself with color psychology [online]. [cit. 2017-05-19]. Dostupné z: <http://www.empower-yourself-with-color-psychology.com/color-green.html>

- [76] The iconic font and CSS toolkit. Font Awesome [online]. [cit. 2017-05-19]. Dostupné z: <http://fontawesome.io/>
- [77] The Psychology and Meaning of Colors. Color Psychology [online]. [cit. 2017-05-19]. Dostupné z: <https://www.colorpsychology.org/>
- [78] The Psychology of Gamification: Can Apps Keep You Motivated? Lifehacker [online]. [cit. 2017-05-19]. Dostupné z: <http://lifehacker.com/the-psychology-of-gamification-can-apps-keep-you-motiv-1521754385>
- [79] Transakce platebními kartami - vedou bankomaty, ale bezkontaktní placení raketově roste. Finparáda [online]. [cit. 2017-05-19]. Dostupné z: <http://www.finparada.cz/1938-Transakce-platebnimi-kartami.aspx>
- [80] Tvoříme persony pro obsahový marketing. Včeliště [online]. [cit. 2017-05-19]. Dostupné z: <https://vceliste.cz/blog/tvorime-persony-pro-obsahovy-marketing/>
- [81] Účet v bance mají už čtyři pětiny Čechů, většina ale nerozezná kreditní a debetní kartu. Aktuálně.cz [online]. [cit. 2017-05-19]. Dostupné z: <https://zpravy.aktualne.cz/ekonomika/ucet-v-bance-maji-uz-ctyri-petiny-cechu-vetsina-ale-nero-zezn/r~d348deea393011e6888a0025900fea04/?redirected=1495221031>
- [82] UML Use Case Diagrams. UML diagrams [online]. [cit. 2017-05-19]. Dostupné z: <http://www.uml-diagrams.org/use-case-diagrams.html>
- [83] Úvod do Nette frameworku pro PHP. ITnetwork [online]. [cit. 2017-05-19]. Dostupné z: <https://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette/>
- [84] VLASTNOSTI DOBRÉHO LOGA (OBSÁHLÝ ČLÁNEK). UNIE GRAFICKÉHO DESIGNU [online]. [cit. 2017-05-19]. Dostupné z: <http://unie-grafickeho-designu.cz/vlastnosti-dobreho-loga-obsahly-clanek/#.WR9fcNLyIU>
- [85] VPS Server hosting. RapidTables [online]. [cit. 2017-05-20]. Dostupné z: <http://www.rapidtables.com/web/hosting/vps-server.htm>
- [86] Web sites are made of lots of things — frameworks, libraries, assets, and utilities. Bower manages all these things for you. Bower [online]. [cit. 2017-05-19]. Dostupné z: <https://bower.io/>
- [87] What is a QR Code? Unitag [online]. [cit. 2017-05-19]. Dostupné z: <https://www.unitag.io/qr-code/what-is-a-qr-code>
- [88] What is Docker? Docker [online]. [cit. 2017-05-19]. Dostupné z: <https://www.docker.com/what-docker>
- [89] What is jQuery? JQuery [online]. [cit. 2017-05-19]. Dostupné z: <https://jquery.com/>
- [90] What is php framework. PHP Frameworks [online]. [cit. 2017-05-19]. Dostupné z: <http://www.phpframeworks.com/>

-
- [91] What is PHP? PHP [online]. [cit. 2017-05-19]. Dostupné z: <http://php.net/manual/en/intro-what-is.php>
- [92] Working with JSON data. Mozilla developer network [online]. [cit. 2017-05-19]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>

11 Seznam obrázků

Obr. 1	Pracovní plocha aplikace sMoneyBox	19
Obr. 2	Rozhraní přihlášeného uživatele aplikace eÚčty.cz	20
Obr. 3	Rozhraní přihlášeného uživatele aplikace Fakturoid	22
Obr. 4	Rozhraní aplikace iDoklad	23
Obr. 5	Princip fungování OAuth2 protokolu [60]	35
Obr. 6	Use case aplikace	40
Obr. 7	Drátěný model pro obrazovku přihlášení uživatele	41
Obr. 8	Drátěný model pro obrazovku s výpisem příjmů a výdajů	42
Obr. 9	Logo aplikace	45
Obr. 10	Předdefinované odstavcové styly v grafickém návrhu	46
Obr. 11	Návrh obrazovek aplikace, rozdělený do návrhových ploch programu Adobe Photoshop	47
Obr. 12	Schéma databáze	49
Obr. 13	Hierarchie presenterů aplikace	53
Obr. 14	Ruční přiřazení kategorie transakce uživatelem	56
Obr. 15	Varianty modálního okna pro přidání závazku.	60
Obr. 16	Rozdělní faktury na výřezy	62

12 Seznam tabulek

Tab. 1	Souhrn parametrů požadavku při zadávání platebního příkazu u Fio banky	32
Tab. 2	Přehled affiliate programů [70]	67