

Aplikace pro správu osobních financí

Bc. Filip Tesař

Diplomová práce
2014



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Filip Tesař**
Osobní číslo: **A12737**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Aplikace pro správu osobních financí**

Zásady pro vypracování:

1. Analyzujte existující řešení pro správu osobních financí.
2. Provedte analýzu požadavků na aplikaci pro správu osobních financí a navrhnete vlastní řešení.
3. Věnujte pozornost způsobu zabezpečení aplikace.
4. Seznamte se s technologií ASP.NET.
5. Vytvořte prototyp aplikace.
6. Vyhodnoťte vytvořený prototyp.
7. Navrhnete další možný rozvoj aplikace.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **MACDONALD, Matthew, Adam FREEMAN a Mario SZPUSZTA. ASP.NET 4 a C# 2010: tvorba dynamických stránek profesionálně. Vyd. 1. Překlad Jan Pokorný. Brno: Zoner Press, 2011, 880 s. Encyklopedie Zoner Press. ISBN 978-80-7413-131-8.**
2. **SKEET, Jon. C# in depth: Second Edition. 2. vyd. Stamford, CT: Manning, 2011, 554 s. ISBN 19-351-8247-1.**
3. **KRUG, Steve. Web design: nenuťte uživatele přemýšlet!. 2. aktualiz. vyd. Překlad Jan Škvařil. Brno: Computer Press, 2006, 167 s. ISBN 80-251-1291-8.**
4. **NIELSEN, Jakob a Marie TAHIR. Použitelnost domovských stránek. Vyd. 1. Brno: Zoner Press, 2005, 323 s. Encyklopedie webdesignera. ISBN 80-868-1518-8.**
5. **LACKO, Luboslav. SQL hotová řešení: k okamžitému použití. Vyd. 1. Brno: Computer Press, 2003, 298 s. ISBN 80-722-6975-5.**
6. **CROFT, Jeff, Ian LLOYD a Dan RUBIN. Mistrovství v CSS: pokročilé techniky pro webové designéry a vývojáře. Vyd. 1. Překlad Josef Bábík. Brno: Computer Press, 2007, 409 s. ISBN 978-80-251-1705-7.**
7. **MSDN – Microsoft Developer Network [online]. Microsoft, 2014 [cit. 2014-02-02]. Dostupné z: <http://msdn.microsoft.com>.**
8. **The Official Microsoft ASP.NET Site [online]. Microsoft, 2014 [cit. 2014-02-02]. Dostupné z: <http://www.asp.net>.**

Vedoucí diplomové práce:

Ing. Radek Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů


Datum zadání diplomové práce:

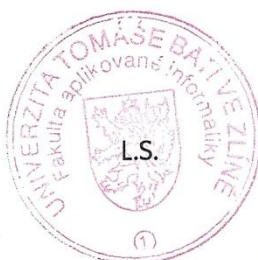
21. února 2014

Termín odevzdání diplomové práce:

20. května 2014

Ve Zlíně dne 21. února 2014


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Příjmení a jméno: TESAR FILIA

Obor: INFORMAČNÍ TECHNOLOGIE

PROHLÁŠENÍ

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby ¹⁾;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3 ²⁾;
- beru na vědomí, že podle § 60 ³⁾ odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 ³⁾ odst. 2 a 3 mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Ve Zlíně 7.3.2014.....



1) zákon č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, § 47 Zveřejňování závěrečných prací:

(1) Vysoká škola nevydělečně zveřejňuje disertační, diplomové, bakalářské a rigorózní práce, u kterých proběhla obhajoba, včetně posudků oponentů a výsledku obhajoby prostřednictvím databáze kvalifikačních prací, kterou spravuje. Způsob zveřejnění stanoví vnitřní předpis vysoké školy.

(2) Disertační, diplomové, bakalářské a rigorózní práce odevzdané uchazečem k obhajobě musí být též nejméně pět pracovních dnů před konáním obhajoby zveřejněny k nahlédnutí veřejnosti v místě určeném vnitřním předpisem vysoké školy nebo není-li tak určeno, v místě pracoviště vysoké školy, kde se má konat obhajoba práce. Každý si může ze zveřejněné práce pořizovat na své náklady výpisy, opisy nebo rozmnoženiny.

(3) Platí, že odevzdáním práce autor souhlasí se zveřejněním své práce podle tohoto zákona, bez ohledu na výsledek obhajoby.

2) zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, § 35 odst. 3:

(3) Do práva autorského také nezasahuje škola nebo školské či vzdělávací zařízení, užije-li nikoli za účelem přímého nebo nepřímého hospodářského nebo obchodního prospěchu k výuce nebo k vlastní potřebě dílo vytvořené žákem nebo studentem ke splnění školních nebo studijních povinností vyplývajících z jeho právního vztahu ke škole nebo školskému či vzdělávacímu zařízení (školní dílo).

3) zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, § 60 Školní dílo:

(1) Škola nebo školské či vzdělávací zařízení mají za obvyklých podmínek právo na uzavření licenční smlouvy o užití školního díla (§ 35 odst.

3). Odírá-li autor takového díla udělit svolení bez vážného důvodu, mohou se tyto osoby domáhat nahrazení chybějícího projevu jeho vůle u soudu. Ustanovení § 35 odst. 3 zůstává nedotčeno.

(2) Není-li sjednáno jinak, může autor školního díla své dílo užít či poskytnout jinému licenci, není-li to v rozporu s oprávněnými zájmy školy nebo školského či vzdělávacího zařízení.

(3) Škola nebo školské či vzdělávací zařízení jsou oprávněny požadovat, aby jim autor školního díla z výdělku jím dosaženého v souvislosti s užitím díla či poskytnutím licence podle odstavce 2 přiměřeně přispěl na úhradu nákladů, které na vytvoření díla vynaložily, a to podle okolností až do jejich skutečné výše; přitom se přihlédne k výši výdělku dosaženého školou nebo školským či vzdělávacím zařízením z užití školního díla podle odstavce 1.

ABSTRAKT

Tato diplomová práce se zabývá aplikacemi pro správu osobních financí. V teoretické části je provedena analýza relevantních existujících webových řešení a rozbor největších bezpečnostních zranitelností webových aplikací podle nezávislé organizace OWASP. V praktické části je navrženo vlastní řešení problematiky ve formě webové aplikace. Jsou vypracovány případy užití a navrženo uživatelské rozhraní. Dále se práce věnuje výběru technologií pro implementaci a vytvoření funkčního prototypu aplikace, který je využit pro uživatelské testování. Výsledky tohoto testování jsou v práci popsány spolu s návrhem dalších rozšíření aplikace.

Klíčová slova:

Osobní finance, webové aplikace, ASP.NET, uživatelská použitelnost, uživatelské testování

ABSTRACT

This dissertation describes applications for personal finance management. The theoretical part consists of an analysis of existing and relevant web solutions and reviews of the largest security vulnerabilities in web applications according to the independent OWASP organization. The practical part depicts a solution of this problem in the form of a web application. Use case scenarios are described as well as user interface design concept. This dissertation also describes a selection of technologies for implementing and creating a functional prototype of an application, which is used for user testing. Results of these tests are described in this work along with further extension concepts for the application.

Keywords:

Personal finance, web applications, ASP.NET, user experience, usability testing

Děkuji vedoucímu mé diplomové práce Ing. Radku Šilhavému, Ph.D. za jeho odborné vedení a cennou zpětnou vazbu, kterou mi dával po celou dobu tvorby této práce. Dále bych chtěl poděkovat své přítelkyni a rodině za jejich podporu a vytvořené zázemí.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

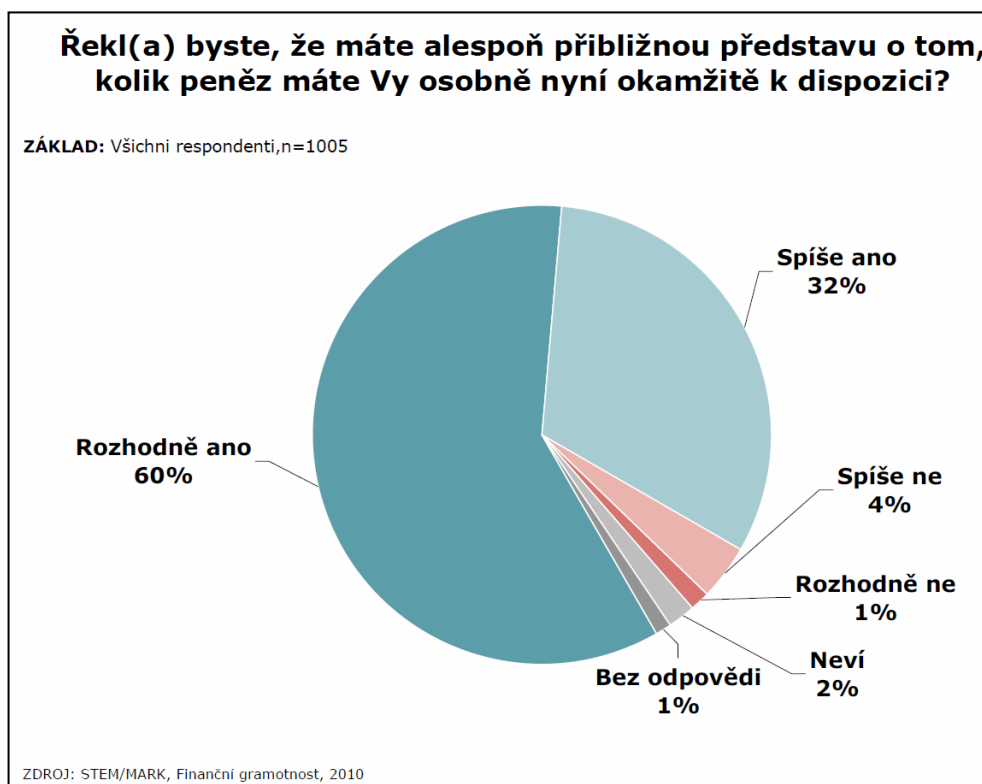
ÚVOD	11
I TEORETICKÁ ČÁST	13
1 POŽADAVKY NA APLIKACI PRO SPRÁVU FINANČÍ	14
2 TYPY APLIKACÍ PRO SPRÁVU FINANČÍ	16
2.1 DESKTOPOVÉ APLIKACE	16
2.2 MOBILNÍ APLIKACE	17
2.3 WEBOVÉ APLIKACE	18
3 PŘEHLED DOSTUPNÝCH ŘEŠENÍ	19
3.1 EÚČTY.CZ	19
3.1.1 Evidence transakcí	19
3.1.2 Statistiky a plánování	20
3.1.3 Zabezpečení a zálohování	20
3.1.4 Další funkce	20
3.2 MOŘE FINANČÍ	21
3.2.1 Evidence transakcí	21
3.2.2 Statistiky a plánování	22
3.2.3 Zabezpečení a zálohování	22
3.2.4 Další funkce	22
3.3 TOSHL FINANCE	22
3.3.1 Evidence transakcí	23
3.3.2 Statistiky a plánování	23
3.3.3 Zabezpečení a zálohování	24
3.3.4 Další funkce	24
3.4 SHRNUÍ	24
4 ZABEZPEČENÍ WEBOVÝCH APLIKACÍ	25
4.1 INJECTION	25
4.1.1 Ukázka útoku	25
4.1.2 Jak předcházet této zranitelnosti	26
4.2 ZRANITELNÁ AUTENTIZACE NEBO SPRÁVA RELACE	26
4.2.1 Ukázky útoku	27
4.2.2 Jak předcházet této zranitelnosti	27
4.3 CROSS-SITE SCRIPTING (XSS)	28
4.3.1 Ukázka útoku	28
4.3.2 Jak předcházet této zranitelnosti	28
4.4 NEZABEZPEČENÁ PŘÍMÁ REFERENCE OBJEKTŮ	29
4.4.1 Ukázka útoku	29
4.4.2 Jak předcházet této zranitelnosti	29
4.5 ŠPATNÁ KONFIGURACE ZABEZPEČENÍ	30
4.5.1 Ukázka útoku	30
4.5.2 Jak předcházet této zranitelnosti	30
4.6 ÚNIK CITLIVÝCH DAT	31
4.6.1 Ukázka útoku	31

4.6.2	Jak předcházet této zranitelnosti.....	31
4.7	KONTROLA OPRÁVNĚNÍ NA APLIKAČNÍ ÚROVNI	32
4.7.1	Ukázka útoku.....	32
4.7.2	Jak předcházet této zranitelnosti.....	32
4.8	CROSS-SITE REQUEST FORGERY (CSRF)	32
4.8.1	Ukázka útoku.....	33
4.8.2	Jak předcházet této zranitelnosti.....	33
4.9	KOMPONENTY SE ZNÁMÝMI ZRANITELNOSTMI.....	34
4.9.1	Jak předcházet této zranitelnosti.....	34
4.10	NEOVĚŘENÉ PŘESMĚROVÁNÍ	34
4.10.1	Ukázka útoku.....	34
4.10.2	Jak předcházet této zranitelnosti.....	35
II	PRAKTICKÁ ČÁST	36
5	FUNKCIONALITA APLIKACE	37
5.1	FUNKČNÍ POŽADAVKY	37
5.1.1	Uživatelé	37
5.1.2	Správa účtů.....	37
5.1.3	Správa transakcí	38
5.1.4	Statistiky	38
5.1.5	Plánování.....	39
5.1.6	Správa vozidla	39
5.1.7	Import a export dat	40
5.2	NEFUNKČNÍ POŽADAVKY	40
5.3	PŘÍPADY UŽITÍ.....	40
5.3.1	Uživatelé	41
5.3.2	Správa účtů.....	49
5.3.3	Správa štítků.....	53
5.3.4	Správa transakcí	57
5.3.5	Správa opakovaných transakcí	62
5.3.6	Správa rozpočtů.....	65
5.3.7	Správa vozidel	68
5.3.8	Zálohování a obnova dat.....	73
6	UŽIVATELSKÉ ROZHRANÍ.....	75
6.1	PŘEHLED A EVIDENCE TRANSAKČÍ.....	76
6.2	SPRÁVA	79
6.3	VOZIDLA.....	79
6.4	GRAFICKÝ STYL UŽIVATELSKÉHO ROZHRANÍ.....	80
7	NÁVRH IMPLEMENTACE	81
7.1	POUŽITÉ TECHNOLOGIE A KNIHOVNY	81
7.1.1	ASP.NET a C#	81
7.1.2	ASP.NET MVC.....	82
7.1.3	ASP.NET Web API	82
7.1.4	Microsoft SQL Server.....	83
7.1.5	HTML5 a CSS3.....	83
7.1.6	Bootstrap a jQuery.....	84

7.1.7	Angular.js.....	84
7.1.8	Google Charts.....	84
7.2	IMPLEMENTACE.....	84
7.2.1	Uživatel.....	85
7.2.2	Účty.....	85
7.2.3	Transakce a štítky	86
7.2.4	Rozpočty	87
7.2.5	Vozidla.....	88
7.3	ZABEZPEČENÍ APLIKACE	88
7.3.1	Injection	88
7.3.2	Zranitelná autentizace nebo správa relace	88
7.3.3	Cross-Site Scripting (XSS)	89
7.3.4	Nezabezpečená přímá reference objektů a kontrola oprávnění.....	89
7.3.5	Špatná konfigurace zabezpečení	89
7.3.6	Únik citlivých dat	89
7.3.7	Cross-Site Request Forgery (CSRF).....	90
7.3.8	Komponenty se známými zranitelnostmi.....	90
7.3.9	Neověřené přesměrování	90
8	PROTOTYP APLIKACE	91
8.1	REGISTRACE UŽIVATELE A PŘIHLÁŠENÍ	91
8.2	VYTVORENÍ ÚČTŮ	92
8.3	EVIDENCE TRANSAKČÍ.....	93
8.4	ZOBRAZENÍ STATISTIK.....	95
8.4.1	Filtrování transakcí	95
8.4.2	Grafy	96
8.5	ZMĚNA HESLA A ODHLÁŠENÍ.....	98
8.6	CELKOVÝ DOJEM UŽIVATELŮ	99
9	DALŠÍ ROZVOJ APLIKACE.....	100
	ZÁVĚR	101
	CONCLUSION.....	102
	SEZNAM POUŽITÉ LITERATURY	103
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	105
	SEZNAM OBRÁZKŮ.....	106
	SEZNAM PŘÍLOH	107

ÚVOD

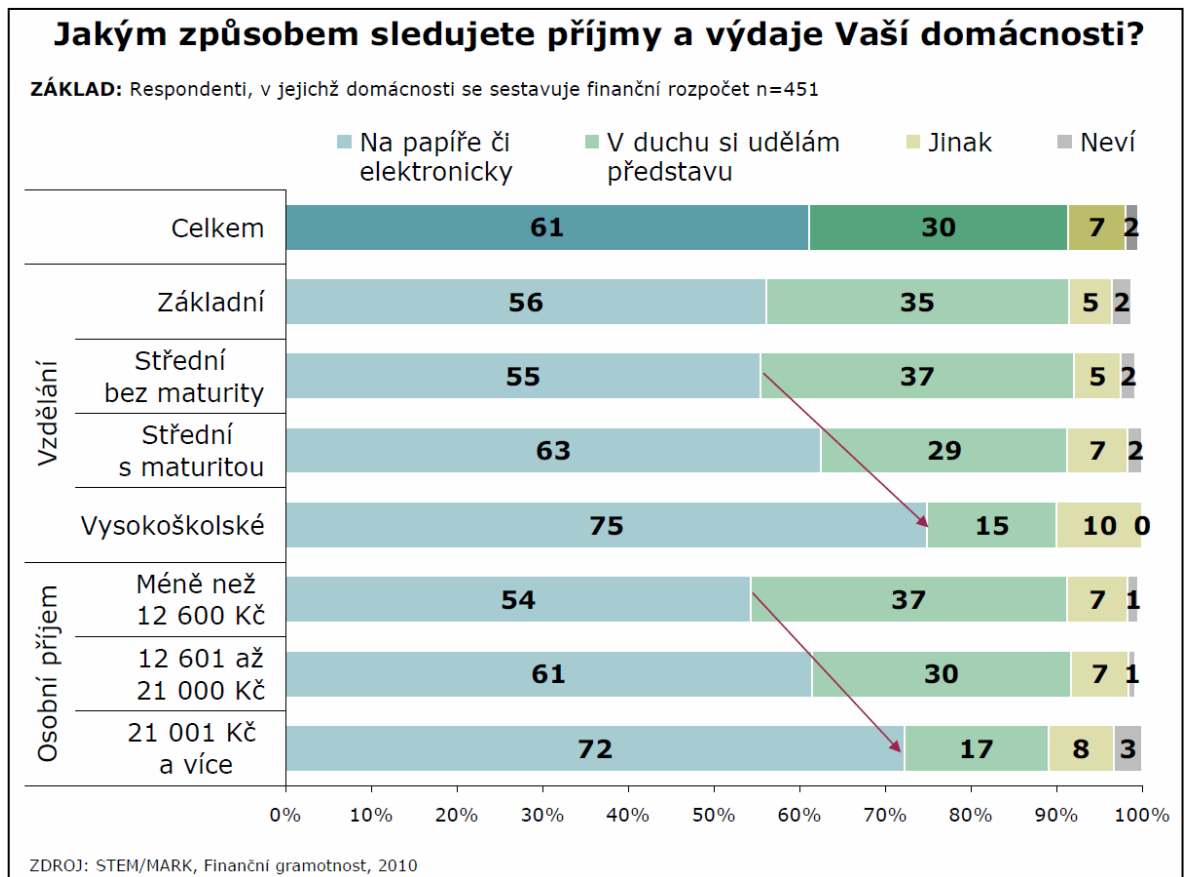
Osobní finance jsou velmi citlivé a značně individuální téma každého z nás. Podle výzkumu finanční gramotnosti v ČR provedeného společností STEM/MARK pro Ministerstvo financí ČR a Českou národní banku v roce 2010, má o svých financích přehled celých 92% lidí (viz Obr. 1). Celkem 45% domácností si tvoří dopředu rozpočty a 95% z nich alespoň někdy kontroluje jejich dodržování. [1]



Obr. 1. Graf zobrazující kolik osob v ČR má přehled o svých financích. [1]

Podle stejného výzkumu [1] lidé, kteří si sestavují rozpočty, sledují své příjmy a výdaje v 61% na papíře nebo elektronicky (viz Obr. 2).

Já sám se řadím právě mezi tuto skupinu, své příjmy a výdaje si důsledně eviduji již několik let a používám k tomu tabulky v programu Microsoft Excel. Není to proto, že by byl tento způsob evidence osobních financí dokonalý, ale spíše z toho důvodu, že jsem zatím nenašel žádný jiný, který by mi vyhovoval. Otázkou je, zda ti lidé, kteří si evidují své finance elektronicky či na papíře, k tomu používají opravdu nástroj co jim vyhovuje, nebo jsou na tom podobně jako já.



Obr. 2. Graf zobrazující zastoupení jednotlivých způsobů sledování financí. [1]

Cílem této diplomové práce je zanalyzovat dostupné aplikace pro správu osobních financí (především na českém trhu) a navrhnout takové vlastní řešení této problematiky, které by mohlo být pozitivně přijato uživateli evidujícími si své finance jinými prostředky a případně i těmi, kteří si neevidují finance vůbec, protože pro to zatím nenašli žádný atraktivní nástroj.

I. TEORETICKÁ ČÁST

1 POŽADAVKY NA APLIKACI PRO SPRÁVU FINANČÍ

Každý člověk přistupuje ke svým osobním financím jiným způsobem a podle toho má také různé očekávání od aplikace pro jejich správu. Dalo by se skoro říci, že ideální aplikace by byla vytvořena právě na míru konkrétnímu člověku a uměla by přesně to, co od ní požaduje, ale nic navíc. S aplikací určenou pro správu financí bude člověk pracovat denně a právě proto je velmi důležité, aby pro něj byla co nejlépe použitelná. Řčení, že software není dokonalý tehdy, když do něj nelze nic přidat, ale až tehdy, když z něj nelze nic odebrat v tomto případě platí dvojnásob.

Existují dvě skupiny potencionálních uživatelů aplikace. Jedni by chtěli, aby byl výsledný software velmi komplexní a nabízel jim mnoho funkcí pro pokročilou analýzu jejich financí. Druzí by nejraději měli k dispozici software, který bude co nejjednodušší a tím pádem také velmi snadno použitelný. Požadavky těchto dvou skupin jdou proti sobě, protože čím bude aplikace komplexnější a čím více funkcionality bude nabízet, tím méně přehlednou se může stát pro uživatele hledajícího jednoduchost a naopak.

Je třeba najít průnik požadavků těchto dvou skupin. Jako důležité bych označil i nefunkční požadavky, například výbornou uživatelskou použitelnost výsledné aplikace (UX – User Experience). Výborná uživatelská použitelnost znamená, že aplikace má dobře navrženou informační architekturu – tedy že uživatel je vždy schopen najít právě tu funkci, kterou chce použít, práce s aplikací je pro něj přirozená a zbytečně jej nezatěžuje [2]. Dále musí být aplikace pro uživatele atraktivní, aby ji měl chuť každodenně používat. Aplikace nesmí uživateli klást žádné překážky – a to ani v případě, pokud by byl handicapovaný – měla by tedy splňovat alespoň základní požadavky přístupnosti [3].

Dalším důležitým nefunkčním požadavkem, na kterém se i v tomto případě jistě shodne většina uživatelů, jsou vysoké nároky na zabezpečení aplikace. Osobní finance jsou jedním z nejcitlivějších témat každého člověka. Případný únik takto osobních dat představuje velký problém, protože mohou být snadno zneužita. Žádný uživatel nebude svěřovat svá důvěrná data aplikaci, která jej nepřesvědčí o vysoké úrovni jejich zabezpečení.

Co se funkčních požadavků týče, tak základem všech aplikací pro správu financí je evidence příjmů a výdajů v čase. Téměř každý má pravidelné příjmy nebo výdaje, jistá automatizace evidence těchto položek se tudíž přímo nabízí. Aplikace by měla zvládat pracovat s více bankovními účty, které mohou chtít uživatelé evidovat zvlášť (hotovost může být považována také za zvláštní účet). Jisté zjednodušení by uživateli přinesl

automatický import dat z výpisů bankovních účtů. Důležitá je i možnost exportu dat pro případ, že by uživatel již nechtěl aplikaci dále používat, ale chtěl by si archivovat svá data. V tomto případě by mu také mělo být umožněno uživatelský účet zrušit a všechna data odstranit.

Samotná elektronická evidence příjmů a výdajů uživateli nepřináší nic navíc oproti například papírové evidenci (která může být v podstatě i ve formě archivovaných účtenek a výplatních pásek). Teprve možnost agregovaného přehledu příjmů a výdajů dává uživateli motivaci aplikaci reálně používat. Tento přehled může být v nejjednodušší verzi například jen součtem příjmů a výdajů ve zvoleném časovém období. Mnohem větší možnosti se ale nabízí v případě, kdy by jednotlivé příjmy a výdaje byly kategorizované. Například vývoj nákladů na bydlení a stravu v čase je pro uživatele určitě zajímavější, než prostá tabulka se všemi výdaji. Ještě lepší by bylo, kdyby tyto přehledy uměla aplikace nabídnout v atraktivní grafické formě.

Od retrospektivního přehledu příjmů a výdajů je jen krůček k jejich plánování do budoucnosti. Aplikace by tedy měla zvládnout vytváření časově ohraničených rozpočtů na jednotlivé kategorie výdajů a v případě překročení stanoveného rozpočtu uživatele upozornit. Kromě rozpočtu, který může být chápán jako restrikce na straně výdajů, by bylo možné podobný systém využít i na straně příjmů, při budování finančních rezerv nebo spoření. Například pokud by si uživatel stanovil cíl během příštího roku naspořit určitou částku na nové auto, viděl by, kolik peněz již má naspořeno a kolik mu ještě k jeho cíli zbývá ušetřit.

Zajímavou funkcionalitou by mohla být evidence nákladů za provoz zařízení, která vyžadují pravidelnou údržbu nebo investice. Typickým příkladem může být automobil, který má stanovený plán pravidelné údržby a je zatížen dalšími provozními náklady jako jsou pohonné hmoty, pojištění, dálniční poplatky apod. Všechny náklady související s automobilem budou v systému již evidované a stačí je použít například pro spočítání průměrných nákladů za ujetý kilometr včetně amortizace automobilu.

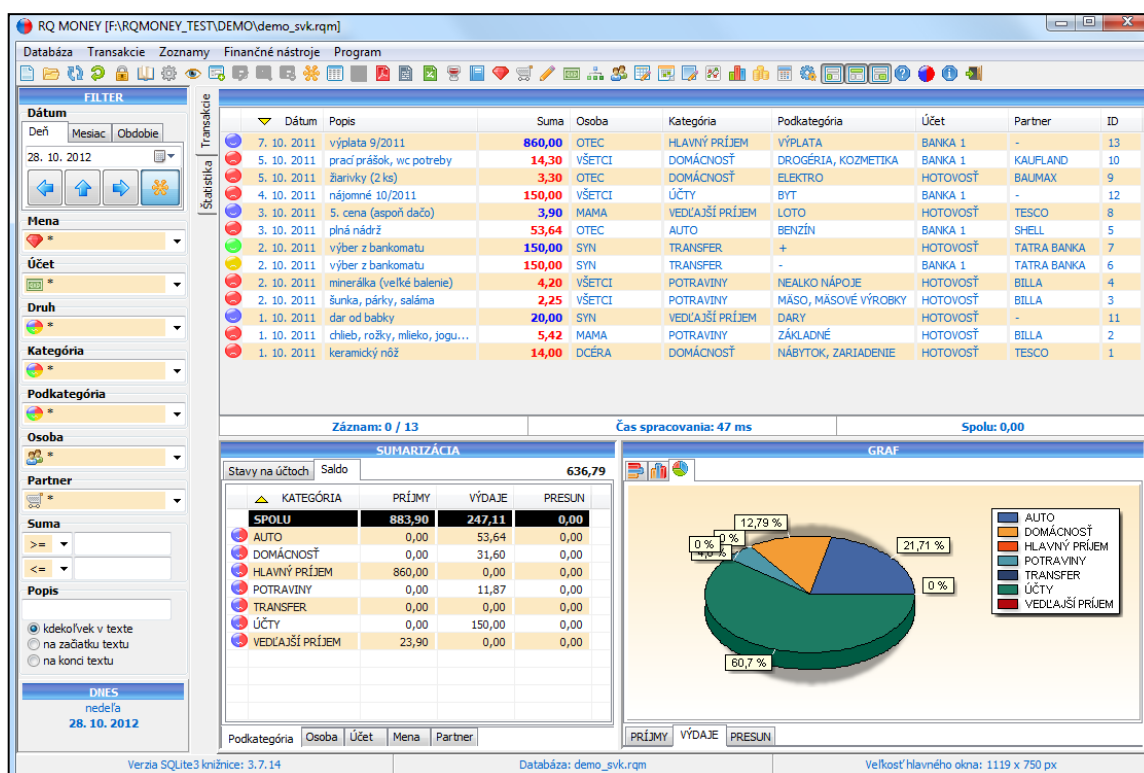
Dalším usnadněním by pro uživatele mohla být například evidence pohledávek a dluhů nebo spotřeby energií v domácnosti. Jistě existuje spousta dalších možností, které jsem nezmínil, nicméně v dnešní době jsou úspěšné především aplikace, které se snaží řešit jeden problém, ale řeší jej opravdu dobře.

2 TYPY APLIKACÍ PRO SPRÁVU FINANČÍ

Dostupné aplikace pro správu osobních financí by se daly rozdělit do tří skupin podle způsobu jejich použití a distribuce. Jsou to desktopové aplikace, mobilní aplikace a webové aplikace. Každá z těchto skupin je něčím specifická.

2.1 Desktopové aplikace

Jedná se o programy pro stolní počítače a notebooky. Většinou se musí instalovat, což samo o sobě může některé nezkušené uživatele odradit. Výhoda těchto aplikací je současně jejich nevýhodou – všechna data jsou uložena pouze lokálně na daném počítači. To znamená, že nad nimi uživatel má plnou kontrolu a žádá třetí osoba k nim nemá přístup. Nevýhodou tohoto řešení je, že uživatel zadává data pouze na svém počítači. To může být velmi nepraktické například na dovolené nebo služební cestě, kdy je nutné výdaje evidovat nějakým jiným způsobem a poté je zpětně zaznamenat do aplikace. Taktéž o zálohování těchto dat se uživatel musí starat sám.



Obr. 3. Ukážka programu RQ Money. [4]

Tyto aplikace většinou vznikly ještě před masivním rozšířením chytrých mobilních telefonů a z hlediska uživatelské přívětivosti jsou dnes poměrně zastaralé. Typickým

zástupcem tohoto druhu aplikací je například velmi rozšířený Microsoft Excel nebo specializovaná aplikace RQ Money [4].

2.2 Mobilní aplikace

Mobilní aplikace pro správu financí začaly vznikat po širokém rozšíření chytrých mobilních telefonů. Existuje mnoho takových aplikací pro všechny běžné operační systémy jakými je Android, iOS a Windows Phone. Přístup k datům je u těchto aplikací podobný jako u desktopových aplikací – uživatel aplikaci nainstaluje do svého zařízení a data jsou uložena přímo na jeho lokálním úložišti. Samotná instalace nečiní problém ani nezkušeným uživatelům, protože probíhá plně automaticky po zvolení požadované aplikace v repositáři dané platformy (Google play, App Store nebo Windows Store).

Oproti desktopovým aplikacím mají mobilní aplikace výhodu v tom, že mobilní telefon nosí většinou lidé stále u sebe a proto mohou evidovat své výdaje bezprostředně po jejich vzniku a nemusí je zadávat zpětně. Omezením je u nich většinou velikost displeje mobilních telefonů, které nejsou vhodné k zobrazování velkého množství dat. Představitelem takové aplikace je například My Budget Book, což je aktuálně nejoblíbenější aplikace Google play v kategorii finance [5].



Obr. 4. Ukázka mobilní aplikace My Budget Book. [5]

2.3 Webové aplikace

Webové aplikace se od zástupců předchozích dvou kategorií zásadně odlišují. Jedná se o serverové aplikace, které jsou hostované buď na dedikovaných serverech nebo v cloudu. Uživatel k nim přistupuje přes internetový prohlížeč, nemusí tedy nic instalovat a s aplikací může pracovat z jakéhokoli zařízení, které je připojeno k internetu. To dnes znamená nejen z domácích počítačů, ale také z mobilních telefonů. Pokud má aplikace navíc optimalizovanou mobilní verzi, je práce s ní na mobilním telefonu stejně pohodlná jako s klasickou instalovanou mobilní aplikací.

Nevýhodou těchto aplikací může být z hlediska některých uživatelů fakt, že nemají svá data pod plnou kontrolou. Ta jsou uložena na serverech třetí strany a přestože autoři aplikací deklarují, že jsou data v bezpečí, tak uživatel nikdy nemůže mít jistotu, jestli k nim nemá přístup i někdo jiný. Stejně tak může mít pochybnosti o tom, zda jsou zálohována a v mnohých případech je ani nemůže zálohovat sám, protože aplikace nenabízí žádný způsob jejich exportování. I kdyby byla data pravidelně zálohována, tak bez možnosti jejich exportu je uživatel ztratí v momentě, kdy se provozovatel rozhodne poskytování služby ukončit nebo změnit podmínky jejího užívání tak, že na ně uživatel nebude ochotný přistoupit. V tomto případě je uživatel plně závislý na autorovi aplikace, na rozdíl od běžných aplikací, které fungují dále, přestože se jejich autor rozhodne je již nepodporovat.

Přes výše zmíněné nevýhody je ale obecně příklon k webovým aplikacím trendem. Pro uživatele je tahle forma poskytování software přitažlivější, především kvůli jeho dostupnosti kdykoliv a odkudkoliv. Tyto aplikace také s využitím nových webových technologií svým uživatelům většinou nabízí atraktivní uživatelské prostředí. Pro vývojáře je výhoda také jasná – nad aplikací mají stále plnou kontrolu, spravují ji centralizovaně a nemusí vyvíjet a podporovat více různých mutací aplikace, což je právě největší problém mobilních aplikací. Dále tedy bude tato práce zaměřena jen na webové aplikace určené pro správu osobních financí.

3 PŘEHLED DOSTUPNÝCH ŘEŠENÍ

V této kapitole budou stručně představeny vybrané existující webové služby pro správu osobních financí, dvě české a jedna zahraniční.

3.1 eÚčty.cz

Služba eÚčty.cz je nejpoužívanější na českém trhu, přitom je provozována soukromou osobou a dostupná svým uživatelům zdarma. Do veřejného provozu byla uvedena 7. 1. 2008 a od té doby je stále vyvíjena. [6]

Seznam transakcí

Výběr účtu: Banka

Nový výdaj Nový příjem Vyrovnání stavu Převod na jiný účet Import dat Hromadné operace

Datum	Uživatel	Kategorie	Popis	Částka
9.3.2014	Mirek Hujer	Bez kategorie		- 100 Kč
23.2.2014	Mirek Hujer	Banka	Penzijní	- 300 Kč
21.2.2014	Mirek Hujer	Bez kategorie	Převod na účet Peněženka	- 6 000 Kč
20.2.2014	Mirek Hujer	Bydlení	Energie	- 5 000 Kč
17.2.2014	Mirek Hujer	Bydlení	Kabelova TV	- 150 Kč
17.2.2014	Mirek Hujer	Bez kategorie	Převod na účet Peněženka	- 6 000 Kč
16.2.2014	Mirek Hujer	Benzín		- 742 Kč
14.2.2014	Mirek Hujer	Výplata		25 146 Kč
12.2.2014	Mirek Hujer	Benzín		- 801 Kč
11.2.2014	Mirek Hujer	Bez kategorie	Převod na účet Peněženka	- 3 800 Kč
3.2.2014	Mirek Hujer	Bez kategorie	Převod na účet Peněženka	- 6 000 Kč
1.2.2014	Mirek Hujer	Auto	Dálniční známka	- 1 500 Kč
1.2.2014	Mirek Hujer	Bez kategorie	Vyrovnání stavu na účtu	20 000 Kč

1

Příjmy v seznamu: 45 146 Kč
 Výdaje v seznamu: 8 593 Kč
 Příjmy - výdaje: 14 753 Kč
 Celkový stav na účtu: 14 753 Kč

Označ vše pro hromadné změny

www.eucty.cz | provozovatel | napište nám

Obr. 5. Ukázka přehledu transakcí v aplikaci eÚčty.cz. [6]

3.1.1 Evidence transakcí

Transakce je možné evidovat pod různými účty, které mohou být v různých měnách. Každou transakci lze kategorizovat, přičemž si uživatel předem vytváří vlastní hierarchii kategorií. Vždy ale platí, že jedna transakce může být přiřazena nejvýše jedné kategorii. Systém umožňuje vnitřní převod mezi účty, který se nezapočítává do statistik.

Transakce je možné importovat z CSV souboru, přičemž aplikace podporuje výstupní formát několika českých bank. Dále lze automatizovat vytváření pravidelných transakcí. Služba není optimalizována pro mobilní telefony, ale nabízí jednoduchou aplikaci pro operační systém Android, kterou lze využít pro zadávání transakcí.

3.1.2 Statistiky a plánování

Aplikace umožňuje zobrazení grafu stavu na jednotlivých účtech v čase a přehled příjmů a výdajů v jednotlivých dnech, týdnech, měsících a letech. Statistiky pro kategorie transakcí jsou na výběr měsíční a roční ve formě tabulky. Přehlednější forma grafu podílu kategorií na výdajích je dostupná pouze pro kategorie nejvyšší hierarchické úrovně.

Plánování je zde zahrnuto ve formě jednoduché kalkulačky, do které lze vložit aktuální stav účtu.

3.1.3 Zabezpečení a zálohování

Aplikace umožňuje anonymní registraci při které není vyžadována emailová adresa. Účet je chráněn kombinací uživatelského jména a hesla, přičemž není vyžadována jeho minimální síla nebo délka. Aplikace při přihlašování ani při práci v chráněné sekci standardně nepoužívá šifrovaný přenos, čímž se vystavuje možnosti úniku přihlašovacích údajů nebo krádeži relace (viz kapitola 4.2). Přitom lze chráněného kanálu přes protokol HTTPS využít (doména má certifikát od uznávané autority), ale neděje se tak automaticky, ani to autor nezmiňuje v bezpečnostních doporučeních. Dále jsem objevil zranitelnost k XSS útoku v popisu transakce (viz kapitola 4.3), čehož by se dalo zneužít například pomocí infikování importovaného souboru.

Data lze z aplikace exportovat do CSV souboru.

3.1.4 Další funkce

Speciální funkcí aplikace je možnost vytvoření rodiny, v rámci které lze sdílet účty mezi více uživateli. Dále aplikace umožňuje evidovat tankování automobilu (při tankování se zadává počet natankovaných litrů pohonných hmot a stav tachometru) a dopočítává jeho průměrnou spotřebu a cenu za kilometr. Do této ceny jsou započítány pouze náklady za pohonné hmoty, nikoliv další s ním spojené výdaje. Zajímavou funkcí jsou alarmy, které se aktivují na základě stanoveného data nebo při dosažení určitého počtu kilometrů na stavu

tachometru. Aplikace eÚčty.cz navíc umožňuje jednoduchou evidenci spotřeby energií a záruky na zakoupené zboží.

3.2 Moře financí

Další český projekt, který vznikl na začátku roku 2011 a rychle si získal oblibu. Kromě evidence financí nabízí možnost spolupráce s finančními poradci, kdy funguje jako prostředník pro poptávku po určitém bankovním, investičním nebo pojišťovacím produktu. Registrovaní finanční poradci vidí poptávky uživatelů a mohou na ně reagovat. Uživatel si tedy může prostřednictvím tohoto systému porovnat více různých nabídek. [7]

The screenshot shows the Moře financí.cz application interface. At the top, there is a navigation bar with the logo and user information (hujer.mirek@gmail.com). Below the navigation bar, there are tabs for 'PŘEHLED', 'DATA', 'ÚSPORA', 'NASTAVENÍ', 'PROFIL', and 'FAQ'. A secondary navigation bar contains links like 'Vložit data', 'Záznamy ve frontě', 'Grafy', 'Záznamy', 'Kategorie', 'Součty kategorií', and 'Celkové součty'. The main content area features a search and filter section with dropdowns for 'Částka' (Příjmy a výdaje), 'Datum' (Vše), and 'Účty' (Hlavní účet | Účet | TESTOVACÍ účet). There are also input fields for 'od' and 'do' dates. Below this is a table of transactions with columns: Datum, Účet, Kategorie, Popis, and Částka. The total amount shown is 24 335 Kč. On the right side, there is an advertisement for 'Srovnejte ceny plynu!' and a logo for 'PATRIA ONLINE Váš investiční portál'.

	Datum ↓	Účet	Kategorie	Popis	Částka
<input type="checkbox"/>	15.03.2014	Hlavní účet	židle - (NAKUPOVÁNÍ - Nábytek)		-100 Kč
<input type="checkbox"/>	24.01.2011	TESTOVACÍ účet	Potraviny - (NAKUPOVÁNÍ)	PLATBA KARTOU TESCO	-253 Kč
<input type="checkbox"/>	18.01.2011	TESTOVACÍ účet	Potraviny - (NAKUPOVÁNÍ)	TESCO PRAHA	-1 010 Kč
<input type="checkbox"/>	17.01.2011	TESTOVACÍ účet	Nájem / Hypotéka - (BYDLENÍ)	platba nájem	-6 400 Kč
<input type="checkbox"/>	17.01.2011	TESTOVACÍ účet	VÝBĚR Z BANKOMATU	CSOB SPALENA 3 PRAHA 1	-3 000 Kč
<input type="checkbox"/>	17.01.2011	TESTOVACÍ účet	Potraviny - (NAKUPOVÁNÍ)	TESCO PRAHA NARODNI	-1 558 Kč
<input type="checkbox"/>	10.01.2011	TESTOVACÍ účet	BANKY A POPLATKY	CSOB VODICKOVA	-1 000 Kč
<input type="checkbox"/>	10.01.2011	TESTOVACÍ účet	Knihy, časopisy, noviny - (NAKUPOVÁNÍ)	PALAC KNIH LUXOR PRAHA 1	-258 Kč

Obr. 6. Ukázka přehledu transakcí v aplikaci Moře financí. [7]

3.2.1 Evidence transakcí

Transakce je opět možné evidovat pod různými účty, ale všechny pouze v českých korunách. K hotovosti je v tomto případě přístupováno jinak, protože není vedena jako

speciální účet. Pro výběr hotovosti z účtu se používá speciální typ transakce, která je ihned evidována jako výdaj. Chybí tedy přehled aktuální hotovosti. Převody mezi vlastními účty nejsou jednoduše dostupné, například v případě převodu financí z běžného účtu na spořicí je nutné vytvořit dvě nezávislé transakce ručně (výdej na běžném účtu a příjem na spořicím účtu), čímž dochází ke zkreslení celkových příjmů a výdajů. Každou transakci lze kategorizovat, uživatel si předem vytváří vlastní hierarchii kategorií. Jedna transakce může být přiřazena právě jedné kategorii.

Transakce je možné importovat z CSV souboru, přičemž aplikace podporuje výstupní formát několika českých bank. Lze automatizovat vytváření pravidelných transakcí. Existuje zjednodušená verze služby optimalizovaná pro mobilní telefony.

3.2.2 Statistiky a plánování

Aplikace umožňuje zobrazení grafů vývoje bilance, příjmů a výdajů v čase. Statistiky týkající se vývoje transakcí v jednotlivých kategoriích jsou dostupné za libovolné období ve formě tabulky a grafu. Podíl kategorií na výdajích je zobrazen pro všechny úrovně jejich hierarchie. Pro plánování je možné vytvořit jeden rozpočet pro každou kategorii.

3.2.3 Zabezpečení a zálohování

Aplikace vyžaduje pro registraci ověření emailové adresy. Pro přihlášení se používá kombinace uživatelského jména a hesla, které musí splňovat minimální délku a při jeho nastavování je zobrazen indikátor síly hesla. Během přihlašování a po něm je veškerá komunikace přenášena šifrovaně pomocí ověřeného certifikátu. Aplikace neumožňuje provést zálohu dat.

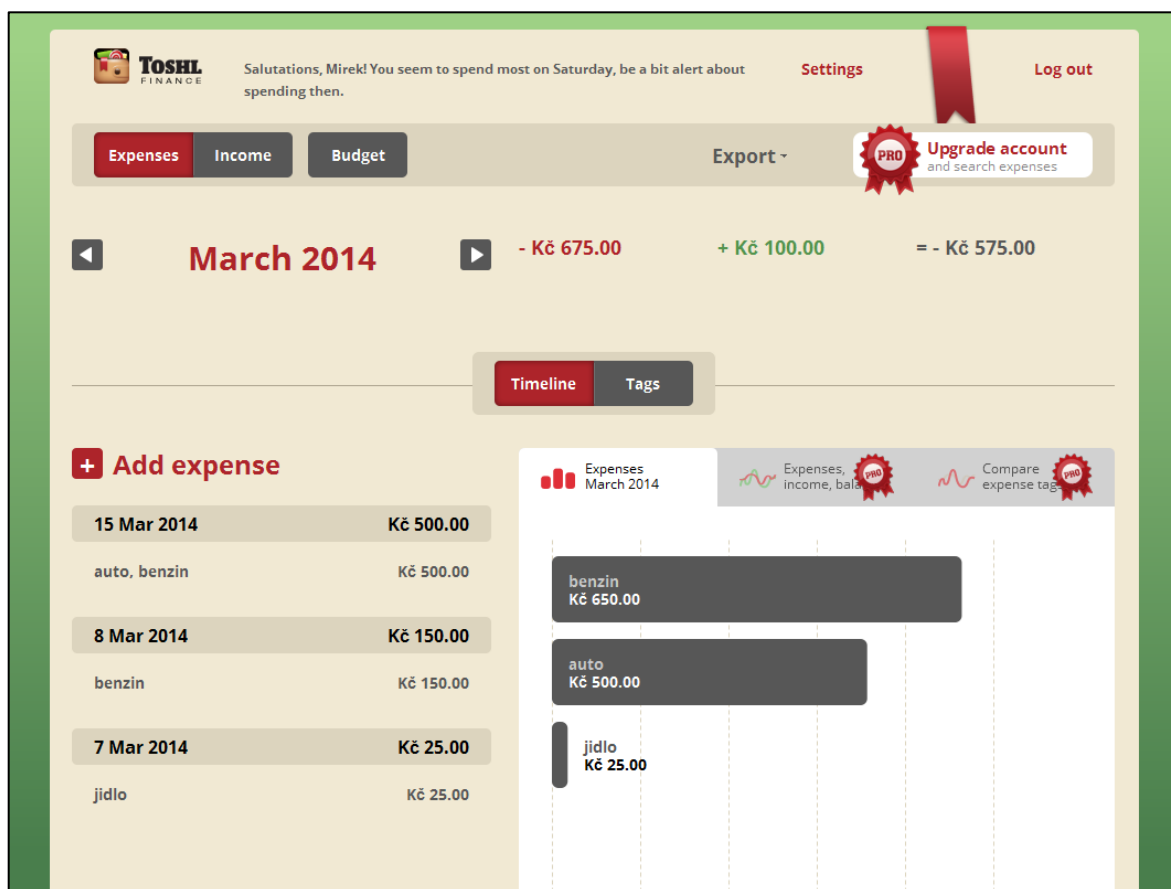
3.2.4 Další funkce

Speciální funkcí aplikace je možnost zadání poptávky po finančních produktech.

3.3 Toshi Finance

Toshi je webová služba pro správu financí, která je propojena s mobilními klienty dostupnými pro všechny rozšířené platformy. Vznikla v roce 2010 ve Slovinsku. V roce 2012 se zakladatelé přemístili do Silicon Valley v San Franciscu, kde se s úspěchem připojili do startup akcelérátoru „500 Startups“. V San Franciscu je nyní sídlo společnosti,

vývoj probíhá ve Slovinsku a její servery jsou umístěny v Německu. Jedná se o uznávaný mezinárodní projekt s několika oceněními. [8]



Obr. 7. Ukázka přehledu transakcí v aplikaci Toshl Finance. [8]

3.3.1 Evidence transakcí

Toshl sází na jednoduchost a uživatelskou přívětivost. Dokáže evidovat pouze jeden bankovní účet v jedné měně, kterou si uživatel může zvolit. Toshl přichází místo kategorií s tagy (štítky). Každé transakci musí být přiřazen alespoň jeden štítek. Štítky nejsou hierarchické a nelze je nijak uspořádat. Importování transakcí ze souboru není možné, ale lze automatizovat pravidelné transakce. Transakce lze spravovat přes mobilní aplikaci, která je dostupná pro všechny rozšířené mobilní platformy.

3.3.2 Statistiky a plánování

Aplikace zobrazí grafy vývoje příjmů, výdajů a bilance v čase. Dále jsou dostupné grafy vývoje výdajů za jednotlivé štítky a porovnání součtu všech výdajů za zvolené časové období podle jednotlivých štítků. Pokud je transakce označena například třemi štítky, tak je

započítána do třech sloupců tohoto grafu. Je také možné zobrazit graf podílu jednotlivých štítků na celkových výdajích.

Plánování je přítomno v podobě rozpočtů, které mohou být denní, týdenní, měsíční, roční a jednorázové. U každého rozpočtu je nastavena finální částka a případně seznam štítků, na které se rozpočet vztahuje. Výsledkem rozpočtu je graf ukazující jaká část stanoveného obnosu již byla utracena.

3.3.3 Zabezpečení a zálohování

K registraci pro používání aplikace je nutné zadat email a heslo, nebo se přihlásit pomocí Google účtu. Veškerá komunikace probíhá šifrovaně. Data je možné exportovat do několika formátů.

3.3.4 Další funkce

Aplikace umožňuje nastavení zasílání měsíčních reportů na emailovou adresu. Základní verze aplikace je dostupná zdarma, nicméně je vhodná pouze pro evidenci transakcí s omezením na jeden příjem měsíčně. Statistiky a rozpočty jsou dostupné pouze v placené verzi za necelých 20 dolarů.

3.4 Shrnutí

Každé z popsaných řešení nabízí různé funkce. Toshl Finance je zaměřené na dobrou použitelnost, přehlednost a jednoduché ovládání. eÚčty.cz je komplexnější a má více funkcí za cenu menší přehlednosti a vyšší složitosti pro nenáročného uživatele. Moře financí není natolik složité ani komplexní jako eÚčty, ale přitom nedosahuje použitelnosti srovnatelné s Toshl Finance.

4 ZABEZPEČENÍ WEBOVÝCH APLIKACÍ

Webové aplikace jsou oproti běžným klientským aplikacím obecně snáze zneužitelné, protože jsou dostupné odkudkoliv nejen svým uživatelům, ale také potenciálním útočníkům. Bezpečnosti aplikace je nutné věnovat obzvláště zvýšenou pozornost, jestliže uchovává citlivá data, jakými informace o osobních financích bezesporu jsou.

Problematikou bezpečnosti internetových aplikací se zabývá neziskové sdružení Open Web Application Security Project (OWASP), které se snaží šířit osvětu o tom, jak vyvíjet bezpečné internetové aplikace. Jedním z jejich projektů je OWASP Top 10, který na základě podkladů od společností zabývajících se bezpečnostními audity a výrobců SaaS řešení mapuje 10 nejvíce problémových zranitelností webových aplikací. První verze tohoto dokumentu vyšla v roce 2003, další v roce 2004, 2007, 2010 a poslední v roce 2013. Do tohoto výběru se zranitelnosti nevybírají pouze podle toho, že jsou nejrozšířenější, ale také podle toho, jak velkou hrozbu jejich zneužití pro webovou aplikaci představuje. [9]

Těchto 10 nejdůležitějších zranitelností z poslední verze OWASP Top 10 [10] bude představeno v této kapitole. U každé z nich je uvedeno v čem zranitelnost spočívá, je nastíněna ukázka jejího zneužití a seznam doporučení, kterými se lze řídit, aby se zranitelnosti v aplikaci předešlo.

4.1 Injection

Injection je technika napadení webových stránek vsunutím (anglicky "injection") nebezpečného kódu přes neošetřený vstup, čímž dojde k vykonání tohoto kódu. Této techniky lze využít v případě, kdy se hodnota vstupu používá při dynamicky vytvářených příkazech interpretovaného jazyka. Typicky je spojována se SQL databázemi, nicméně stejným způsobem lze pozměnit i LDAP, Xpath nebo NoSQL dotazy, SMTP hlavičky apod. [11]

4.1.1 Ukázka útoku

Útočník využije neošetřeného vstupního formuláře nebo URL parametru, který se přímo použije při vytváření následujícího SQL dotazu:

```
string query = "SELECT * FROM Users WHERE UserID = " +  
Request.QueryString.Get("id");
```

Při skládání dotazu se předpokládá, že hodnota vstupu "id" je číslo, ale není to ověřováno. Pokud útočník podvrhne vstup "id", může docílit například smazání celé tabulky "Users":

```
http://example.com/userview?id=1; DROP TABLE Users;
```

Výsledný dotaz bude vypadat následovně:

```
SELECT * FROM Users WHERE UserID = 1; DROP TABLE Users;
```

Kromě nastíněného destruktivního útoku lze samozřejmě zranitelnost využít také k neoprávněnému získání dat, pozměnění uložených dat a podobně.

4.1.2 Jak předcházet této zranitelnosti

1. Je nutné předpokládat, že veškerá vstupní data mohou obsahovat škodlivý kód a podle toho s nimi nakládat.
2. Dobrou praktikou je úplně se vyhnout dynamicky vytvářenému interpretovanému kódu nebo dotazům. Pokud je nutné interpretovaný kód použít, tak místo přímo skládaného kódu je vhodné použít API, které nabízí parametrizované rozhraní (například parametrické SQL dotazy nebo využít uložených procedur).
3. Jestliže není parametrizované rozhraní dostupné, tak je vždy nutné validovat správný typ vstupních hodnot a v případě řetězců escapovat speciální znaky daného jazyka (black list - v případě SQL například apostrof, který ukončuje řetězec). Další možností je opačný přístup (white list), kdy jsou povolené jen určité nezávadné znaky a všechny ostatní uživatel nesmí použít.

[10, s. 7]

4.2 Zranitelná autentizace nebo správa relace

Využití zranitelnosti v procesu autentizace nebo správě relace (session) typicky vede k podvrhnutí identity uživatele. Protokol HTTP je bezstavový (každý požadavek je prováděn samostatně bez návaznosti na předešlé požadavky uživatele), proto musí webové servery implementovat mechanismus, který zajišťuje uchování uživatelského kontextu mezi jednotlivými požadavky. Každá taková relace je jednoznačně identifikována a uživatelův prohlížeč musí s každým požadavkem zasílat také identifikátor relace, ke které požadavek patří. Pokud útočník získá tento identifikátor, může ve webové aplikaci vystupovat a jednat pod identitou uživatele, kterému byl identifikátor odcizen. Další

možností jak podvrhnout identitu uživatele, je odcizení přímo jeho přihlašovacích údajů. [12]

4.2.1 Ukázky útoku

- Webová aplikace pro uchovávání identifikátoru nepoužívá cookies, ale předává si jej jako URL parametr. Uživatel při používání aplikace pošle odkaz na zajímavý obsah serveru svým přátelům, přičemž netuší, že součástí odkazu je i identifikátor jeho relace. Pokud přes tento odkaz někdo k serveru přistoupí, tak bude vystupovat jako přihlášený uživatel a dostane se k jeho citlivým údajům.
- Aplikace nemá nastavenou omezenou platnost relace. Uživatel se přihlásí na veřejně dostupném PC (například v internetové kavárně), ale zapomene se odhlásit a místo toho jen zavře prohlížeč. Útočník později použije stejný prohlížeč, ve kterém přistoupí na adresu aplikace a jedná jako přihlášený uživatel.
- Útočník získá přístup k databázi aplikace (může se jednat i o nepoctivého administrátora), přičemž přihlašovací údaje uživatelů jsou uloženy v otevřené podobě.

4.2.2 Jak předcházet této zranitelnosti

1. Neukládat přihlašovací údaje uživatelů v otevřené podobě.
2. Neumožnit jednoduše změnit přihlašovací údaje, pro případ, že by útočník získal přístup k aktivní relaci uživatele. V praxi to znamená vyžadovat opětovné zadání přihlašovacích údajů před tím, než je uživateli umožněno provádět následující akce:
 - a. změnu hesla,
 - b. změnu emailové adresy v případě, že přes ni lze obnovit zapomenuté heslo,
 - c. zobrazení nebo změnu kontrolních otázek a odpovědí pro případ ztráty hesla.
3. Nevystavovat identifikátor relace v URL.
4. Vygenerovat nový identifikátor relace při každém přihlášení uživatele.
5. Omezit platnost každé relace (relace se invaliduje po určité době nečinnosti uživatele a také po odhlášení uživatele).
6. Přihlašovací údaje uživatele a identifikátor relace zasílat přes šifrovaný kanál. Jinak je aplikace náchylná na "men-in-the-middle" útok [13].

4.3 Cross-Site Scripting (XSS)

Útok Cross-Site Scripting je typem Injection útoku, ve které se škodlivý skript vloží do jinak neškodných a důvěryhodných webových stránek. Útočník tedy použije webovou aplikaci k doručení škodlivého kódu, obvykle ve formě skriptu interpretovaného prohlížečem, nic netušícímu koncovému uživateli. Tato zranitelnost je velmi rozšířená a může se vyskytovat všude tam, kde webová aplikace používá vstup od uživatele k vygenerování výstupního kódu stránky. Prohlížeč zasaženého uživatele nemůže tento škodlivý skript rozpoznat od jiných skriptů na stránce a dojde k jeho spuštění. Vzhledem k tomu, že skript je přímo součástí napadené stránky, má přístup ke všem cookies, identifikátoru relace a dalším citlivým údajům uloženým v prohlížeči touto stránkou. Tento skript může přepsat HTML obsah stránky, ale také může být použit pro krádež relace (session hijacking) – viz kapitola 4.2. [14]

4.3.1 Ukázka útoku

Aplikace po zadání textu do vyhledávacího pole vypisuje výsledky vyhledávání, přičemž do stránky vloží i neošetřený text zadaný uživatelem:

```
<h2>Results for <%= Request.QueryString.Get("search")%></h2>
```

Útočník do URL parametru „search“ vloží škodlivý kód a odkaz pošle oběti útoku. Pokud oběť odkaz otevře, útočník získá její uložená cookies (v těch se může vyskytovat i identifikátor relace):

```
http://example.com/?search=<script>document.location='http://  
www.attack.com/record?cookies='+document.cookie;</script>
```

4.3.2 Jak předcházet této zranitelnosti

1. Před tím, než se jakýkoliv uživatelský vstup stane součástí obsahu stránky, je nutné jej validovat nebo escapovat. Nejedná se pouze o vstup, který by byl přímo vypisován, jak je uvedeno v příkladu, ale také o veškeré hodnoty formulářových prvků, skrytých polí apod.
2. V případě, že je nutné umožnit uživatelům vytvářet přímo obsah stránky (například autorům v publikačním systému, kteří vytváří přímo HTML kód), tak je dobré alespoň filtrovat značky umožňující vložení spustitelného kódu, jako jsou `<script>`, `<embed>`, `<object>`, `<applet>`, `<iframe>` apod.

3. Při ukládání citlivých údajů (jako je například identifikátor relace) do cookies, je možné nastavit ukládané cookie vlastnost HttpOnly, nebude potom dostupná klientským skriptům.

[10, s. 9]

4.4 Nezabezpečená přímá reference objektů

Webové aplikace běžně používají hodnoty databázových klíčů jako referenci na uložené objekty. Ne vždy však kontrolují, zda má daný uživatel právo pracovat s konkrétním objektem. Typickým příkladem tohoto přístupu je například situace, kdy má uživatel k dispozici seznam objektů, které může upravit (ze všech uložených objektů se filtrují jen ty, na které má uživatel oprávnění) a po zvolení konkrétního objektu je přesměrován na editační formulář, kterému je předán klíč zvoleného objektu. Tento formulář už ale nekontroluje oprávnění uživatele pracovat s předaným objektem.

4.4.1 Ukázka útoku

Editační formulář slouží k úpravě objektu, jehož klíč je předán URL parametrem "id". Útočník získá adresu tohoto formuláře (například tím, že upravuje objekt, na který má práva) a změní hodnotu parametru "id", čímž se dostane k datům objektu, ke kterému nemá mít přístup.

4.4.2 Jak předcházet této zranitelnosti

1. Před tím, než dojde k načtení objektu, je nutné zkontrolovat, jestli má uživatel právo s objektem pracovat. Nelze spoléhat na to, že se dané stránce předává správný identifikátor a že uživatelovo oprávnění již bylo ověřeno na předchozí stránce.
2. Nepoužívat přímou referenci objektů pomocí snadno odhadnutelných identifikátorů jako je databázový primární klíč, místo něj například použít GUID nebo nepřímou referenci specifickou pro přihlášeného uživatele.

[10, s. 10]

4.5 Špatná konfigurace zabezpečení

Chyby v zabezpečení mohou vzniknout na jakékoliv aplikační úrovni, zahrnující platformu, webový server, aplikační server, databázi, použitý framework a vlastní kód aplikace. Problémem je neaktualizovaný software, zbytečné aktivní služby, nezabezpečený přístup k souborovému systému, nesmazané výchozí účty, ukázkové aplikace nebo testovací data, zapnutý ladící režim apod.

4.5.1 Ukázka útoku

- Administrativní konzole aplikačního serveru je nainstalována automaticky a není nikdy použita, ale ani odinstalována. Útočník na serveru objeví standardní administrační rozhraní a zkusí se přihlásit s výchozím účtem, což bude úspěšné a získává přístup k serveru.
- Na serveru je povolené procházení obsahu složek. Útočník si postupně stáhne obsah složek, kde je dostupný i kód aplikace (v případě jeho zkompileované verze může útočník použít dekompilátor). Po prostudování kódu aplikace objeví bezpečnostní chybu, kterou zneužije.
- Je zapnutý ladící mód aplikace, útočník tedy dostává k dispozici pro něj důležité informace o architektuře aplikace, schématu databáze apod., díky čemuž jednodušeji objeví zranitelnosti aplikace.
- Na serveru jsou nainstalované ukázkové aplikace, které mají známé bezpečnostní chyby, těch může útočník snadno využít.

4.5.2 Jak předcházet této zranitelnosti

1. Vždy používat aktuální software, ať už jde o operační systém serveru, webový server, platformu, framework nebo komponenty třetích stran.
2. Deaktivovat veškerou nepotřebnou funkcionalitu (otevřené porty, běžící služby, nepoužívané administrativní stránky a účty).
3. Smazat nebo deaktivovat výchozí administrátorské účty.
4. Vypnout ladící režim a v případě chyby nezobrazovat zbytečné podrobnosti.
5. Zkontrolovat bezpečnostní nastavení použité platformy a frameworku.

4.6 Únik citlivých dat

Většina aplikací uchovává alespoň nějaké citlivé údaje o svých uživateli, ať už se jedná o jejich přihlašovací údaje, telefonní čísla, emailové adresy nebo čísla platebních karet. Pokud útočník získá přístup k těmto datům, tak mohou být velmi snadno zneužita. Například když uživatel používá pro přístup k aplikaci stejné heslo jako do své emailové schránky, tak k ní útočník získá přístup také a využije ji k infiltraci dalších uživatelem využívaných služeb. V krajním případě je možné ukrást celou jeho virtuální identitu.

4.6.1 Ukázka útoku

- Útočník získá přístup k údajům v databázi (například provedením SQL Injection útoku nebo díky špatnému nastavení zabezpečení serveru). Přihlašovací údaje jsou uloženy v otevřené podobě nebo se využívá velmi slabého šifrování, tudíž útočník získává přístup k účtu uživatele.
- Aplikace sice citlivá data před uložením šifruje velmi silným šifrovacím algoritmem, ale nepoužívá zabezpečený přenos dat mezi klientem a serverem. Útočník monitoruje provoz na síti (například na nezabezpečené bezdrátové síti) a zachytí buď přímo přihlašovací údaje uživatele, nebo identifikátor jeho relace, díky čemuž opět získává kontrolu nad jeho účtem.

4.6.2 Jak předcházet této zranitelnosti

1. Všechny zneužitelné uživatelské údaje (přihlašovací hesla, čísla platebních karet, čísla dokladů apod.) ukládat v šifrované podobě. Přenos těchto dat musí být zabezpečen, například pomocí šifrování TLS/SSL.
2. Neukládat zbytečně údaje, které nutně není třeba shromažďovat. Co není uloženo, nemůže být zneužito.
3. Pro šifrování využívat silný algoritmus a zabezpečit, aby i šifrovací klíče nemohly být zneužity.
4. Vypnout funkci automatického dokončování u formulářů, které slouží k zadávání citlivých dat.

4.7 Kontrola oprávnění na aplikační úrovni

Tato zranitelnost částečně souvisí s již popsanou nezabezpečenou přímou referencí objektů, jen s tím rozdílem, že se jedná o nezabezpečený přístup k funkcím aplikace. Části aplikace poskytující danou funkcionalitu spoléhají na to, že pro neoprávněné uživatele nejsou dostupné v uživatelském rozhraní a samy už žádnou kontrolu neprovádí.

4.7.1 Ukázka útoku

Útočník (uživatel systému) má právo prohlížet si články v redakčním systému, ale nemá právo je ze systému smazat. Při přístupu k článku se používá URL parametr označující akci, kterou chce uživatel vykonat:

```
http://www.example.com/articles?id=1051&action=view
```

Útočník změní název akce následovně:

```
http://www.example.com/articles?id=1051&action=delete
```

Vzhledem k tomu, že se oprávnění k mazání článků kontroluje pouze na úrovni uživatelského rozhraní, ale ne na aplikační úrovni, dojde ke smazání článku.

4.7.2 Jak předcházet této zranitelnosti

Oprávnění uživatele k využití funkcionality musí být kontrolováno vždy i přímo na aplikační úrovni, nikoliv pouze na prezentační úrovni.

[10, s. 13]

4.8 Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery se nazývá útok, který podvrhne požadavek na zranitelnou aplikaci s cílem provedení akce v kontextu přihlášeného uživatele, přičemž k podvrhnutí tohoto požadavku stačí, aby uživatel přistoupil na stránky ovládané útočníkem a současně byl ve stejném prohlížeči přihlášený ke zranitelné aplikaci. Samotná zranitelnost spočívá v tom, že k provedení dané akce stačí, aby na server přišel požadavek s validním identifikátorem relace přihlášeného uživatele bez dalšího tajného kódu (autorizačního tokenu) pro ověření autentičnosti požadavku. K provedení tohoto útoku musí mít útočník určitou znalost napadené aplikace, aby byl schopný správně sestavit požadavek. [15]

4.8.1 Ukázka útoku

Aplikace umožňuje přeposílání kreditů (virtuálního platidla) mezi svými uživateli, přičemž k odeslání kreditů se využívá GET požadavku na následující adresu:

```
http://example.com/transferCredit?amount=1000&account=12345
```

URL parametr "amount" určuje počet převáděných kreditů a "account" identifikuje účet, na který se kredity převádí. K této stránce mají přístup pouze přihlášení uživatelé, kredity se tedy vždy převádí z účtu aktuálně přihlášeného uživatele.

Útočník na svoji vlastní stránku obsahující například sbírku vtipných obrázků vloží mezi běžné obrázky následující kód:

```

```

Uživatel, který je přihlášen ke zranitelné aplikaci, ve stejném prohlížeči přistoupí na podvodnou stránku. Mezitím, co si prohlíží vtipné obrázky se převedlo jeho 1000 kreditů na útočnickův účet, protože jeho prohlížeč při snaze stáhnout podvodný obrázek zaslal požadavek na zranitelnou aplikaci žádající převod kreditů. Prohlížeč k požadavku přidal automaticky identifikátor relace platný pro doménu "example.com" a zranitelná aplikace nemůže rozlišit mezi autentickým a podvrhnutým požadavkem uživatele.

4.8.2 Jak předcházet této zranitelnosti

1. Pro odeslání dat na server používat POST metodu HTTP protokolu místo GET.
2. Každému formuláři, který umožňuje provedení nějaké akce předat náhodný autorizační token, který se může uložit například do skrytého pole. Tento token musí být specifický pro přihlášeného uživatele a formulář. Po odeslání formuláře je nutné kontrolovat, zda zasláný token odpovídá tokenu, který byl uživateli předán při vykreslování formuláře.
3. Před provedením akce vyžadovat opětovné přihlášení uživatele nebo vyplnění CAPTCHA.

4.9 Komponenty se známými zranitelnostmi

Málokterá aplikace nepoužívá žádnou komponentu třetí strany. Většina vývojářů aplikací použije hotové řešení nějaké funkcionality například ve formě knihovny, než aby vyvíjeli svoje vlastní. Problémem je, že nad kódem komponent třetích stran nemají kontrolu (pokud se nejedná o open source řešení). Ve kterékoliv použité komponentě může být bezpečnostní chyba, které bude útočník moci využít tak dlouho, dokud nebude komponenta aktualizována na novější verzi opravující tuto chybu.

4.9.1 Jak předcházet této zranitelnosti

1. Identifikovat veškeré použité komponenty třetích stran a jejich verze, které aplikace využívá.
2. Sledovat vývoj těchto komponent, zjišťovat, jestli se u používané verze neobjevila bezpečnostní chyba a v případě vydání nové verze komponentu co nejdříve aktualizovat.
3. Komponentu, která trpí bezpečnostními nedostatky a již se dále nevyvíjí nahradit za její aktivně vyvíjenou nebo alespoň udržovanou alternativu.

[10, s. 15]

4.10 Neověřené přesměrování

Internetové aplikace mohou při přístupu k určité stránce přesměřovat své uživatele na jinou stránku. Nejčastěji se tak děje pomocí zaslání HTTP odpovědi s kódem v rozsahu 300-307, typicky je to 302. Další možný způsob přesměrování je vložení javascriptu do stránky, který provede změnu adresy po jejím načtení. Oba dva tyto způsoby vyžadují zaslání odpovědi ze serveru prohlížeči, který si sám vyžádá novou stránku. Třetím využívaným způsobem je přesměrování, které se provádí pouze na straně serveru, kdy server prohlížeči předá již výslednou stránku, klientský prohlížeč se tedy o tomto druhu přesměrování nedozví. Problém nastává v případě, kdy je možné pomocí dat zaslaných na server ovlivnit adresu, na kterou bude uživatel přesměrován.

4.10.1 Ukázka útoku

V aplikaci existuje stránka, která provede přesměrování na adresu zadanou jako URL parametr. Vzhledem k tomu, že stránka neprovádí žádnou kontrolu této adresy, může útočník parametr podvrhnout a přesměrovat uživatele na svoji útočnou stránku:

```
http://www.example.com/redirect?url=attacker.com
```

Takový odkaz se pro uživatele tváří věrohodně, protože směřuje na jemu známou doménu, takže jej s větší pravděpodobností otevře, čímž se ve skutečnosti dostane na stránky útočníka. Ten na své doméně může například provozovat kopii originálního webu za účelem sběru přihlašovacích údajů (tzv. phishing).

4.10.2 Jak předcházet této zranitelnosti

1. Jestliže je to možné, přesměrování se vyhnout. Pokud je přesměrování nezbytné, nesmí parametry požadavku ovlivnit adresu, na kterou přesměrování proběhne.
2. Když je nutné předávat adresu výsledné stránky v parametrech, je třeba tuto adresu validovat, například pomocí seznamu povolených adres. V případě, kdy není možné vytvořit tento seznam, lze například využít asymetrické kryptografie. URL parametry jsou podepsány privátním klíčem a stránka provádějící přesměrování kontroluje pomocí veřejného klíče, jestli nejsou podepsaná data podvrhnutá.

[10, s. 16]

II. PRAKTICKÁ ČÁST

5 FUNKCIONALITA APLIKACE

Cílem této diplomové práce je navrhnout vlastní řešení aplikace pro správu financí s přihlédnutím k existujícím řešením. V této kapitole budou specifikovány požadavky na funkcionalitu aplikace.

5.1 Funkční požadavky

Neformální funkční požadavky jsou pro přehlednost rozděleny podle modulů.

5.1.1 Uživatelé

- Uživatelé aplikace se budou moci sami volně registrovat a přihlašovat.
- Každý uživatel bude identifikován svým unikátním uživatelským jménem, pro ověření bude použito heslo. Volitelně bude moci uživatel vyplnit i svoji emailovou adresu.
- V aplikaci budou existovat dvě uživatelské role – běžný uživatel a správce. Běžný uživatel bude spravovat svá data a bude moci sám odstranit svůj vlastní účet. Uživatel s rolí správce bude spravovat uživatele v systému, konkrétně bude moci odstranit libovolného uživatele ze systému, ale nebude moci prohlížet data jiných uživatelů.
- Pokud uživatel zapomněl své přihlašovací heslo a poskytl emailovou adresu, tak si bude moci zažádat o vytvoření nového hesla. Na zaregistrovanou emailovou adresu mu přijde odkaz pomocí kterého si bude moci vytvořit nové heslo.

5.1.2 Správa účtů

- Aplikace bude umožňovat správu více účtů (hotovost bude možné vést jako jeden z účtů).
- Všechny účty budou vedeny ve stejné měně.
- Počet účtů nebude nijak omezen a uživatel si je bude moci volně vytvářet, upravovat i odstraňovat.
- Mezi účty bude možné jednoduše převádět prostředky aniž by byly zkresleny statistiky (převod mezi vlastními účty nebude evidován jako výdaj a příjem).
- Uživatel si bude moci zobrazit přehled aktuálního stavu prostředků na jednotlivých účtech.

5.1.3 Správa transakcí

- Uživatelé budou pomocí aplikace evidovat své příjmy a výdaje (obecně transakce). Každá transakce bude přiřazena k jednomu z účtů, případně ke dvěma pokud půjde o převod prostředků mezi účty.
- U transakce bude evidována nominální částka, datum uskutečnění transakce a volitelně její popis.
- Transakce budou rozlišeny pomocí štítků, přičemž jeden z nich bude hlavní kategorizační štítek transakce. Volitelně bude možné transakci navíc označit i dalšími upřesňujícími štítky.
- Štítky se budou vytvářet dynamicky při jejich prvním použití. Vytvořený štítek se bude uživateli nabízet při označování dalších transakcí.
- Uživatel bude moci spravovat své vytvořené štítky, bude je moci přejmenovat nebo smazat (transakce označené smazaným štítkem v systému zůstanou, odstraní se pouze jejich označení).
- Uživatel si bude moci zobrazit všechny své evidované transakce ve zvoleném časovém období, standardně se budou zobrazovat transakce za posledních 30 dní. Transakce bude možné filtrovat podle účtů a štítků.
- Evidovanou transakci bude možné upravit nebo smazat.
- Bude možné vytvářet automaticky opakované transakce s denní, týdenní a měsíční frekvencí opakování.
- Každou evidovanou transakci bude možné rozdělit do více samostatných transakcí.

5.1.4 Statistiky

- Graf vývoje příjmů, výdajů a bilance v časovém intervalu, který si uživatel zvolí. Příjmy a výdaje budou agregované buď po dnech, týdnech, měsících a nebo letech. V případě bilance bude použit stav účtu po poslední transakci v daném agregovaném období.
- Graf vývoje příjmů a výdajů označených zvolenými štítky v časovém intervalu, který si uživatel určí. Příjmy a výdaje budou agregované buď po dnech, týdnech, měsících, nebo letech.
- Koláčový graf podílu jednotlivých hlavních kategorizačních štítků na celkových výdajích za zvolené časové období.

- Sloupcový graf podílu jednotlivých štítků na celkových výdajích za zvolené časové období.

5.1.5 Plánování

- Uživatel si bude moci definovat výdajové rozpočty. Každý rozpočet bude mít zadanou částku, popis, období platnosti a volitelný seznam štítků, na které se vztahuje.
- Jednorázový rozpočet bude mít explicitně zadané období. Další možností bude pravidelný rozpočet (týdenní, měsíční, roční) se stanoveným startovacím datem.
- Uživatel bude graficky informován o plnění rozpočtu.

5.1.6 Správa vozidla

- Aplikace umožní uživateli pokročilé sledování výdajů za vozidlo. Uživatel si bude moci definovat libovolný počet vozidel. Každé vozidlo bude mít svůj název, podle kterého jej bude uživatel identifikovat. Vozidla bude možné vytvářet, upravovat a odstraňovat.
- Po vytvoření vozidla se uživateli automaticky vytvoří štítek s jeho názvem, pomocí kterého bude moci označovat transakce související s daným vozidlem.
- Při zadávání transakce bude možné přiřadit k ní operaci tankování pohonných hmot. Pokud uživatel tuto možnost zvolí, bude muset kromě částky zadat také buď cenu za litr pohonné hmoty, nebo počet natankovaných litrů. Po zadání jedné z těchto hodnot se druhá dopočítá. Dále bude uživatel muset označit transakci štítkem jednoho ze svých vozidel a vybrat, zda se jedná o natankování plné nádrže (v tom případě uživatel zadá i stav tachometru vozidla).
- Aplikace bude zobrazovat vypočítanou průměrnou spotřebu vozidla mezi jednotlivým tankováním do plné nádrže a dále celkovou průměrnou spotřebu za vybrané období. Tento údaj bude dostupný pouze v případě, že se v daném období uskutečnily minimálně dvě tankování do plné nádrže.
- Aplikace bude zobrazovat vývoj ceny pohonných hmot a kilometrového nájezdu vozidla dle evidovaných tankování.
- Dále bude možné zobrazit průměrnou cenu za kilometr ve vybraném období. Do průměrné ceny bude možné zahrnout jak pohonné hmoty, tak i další výdaje spojené s vozidlem na základě transakcí označených jeho štítkem.

5.1.7 Import a export dat

- Bude možné importovat data pomocí CSV souboru.
- Bude možné exportovat data do CSV souboru.

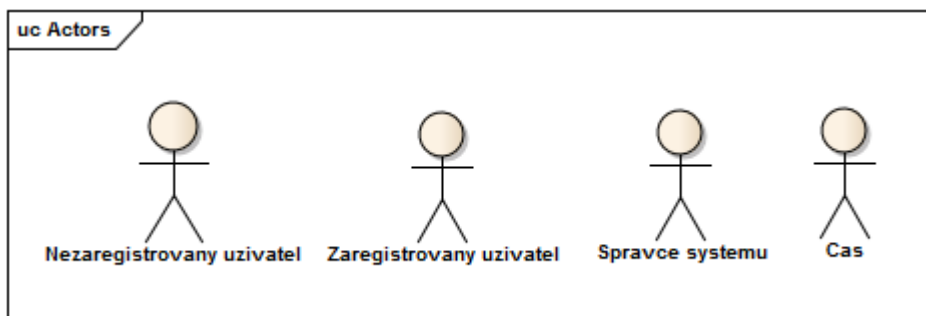
5.2 Nefunkční požadavky

Kromě funkčních požadavků by měla aplikace také splňovat následující nefunkční požadavky.

- Aplikace bude postavena na webové technologii Microsoft ASP.NET, verze 4.5.
- Jako datové úložiště bude použit Microsoft SQL Server 2008 R2 nebo novější.
- Aplikace bude dostatečně zabezpečená, to znamená, že bude respektovat doporučení uvedená v kapitole 4 Zabezpečení webových aplikací.
- Transakce bude možné do systému zadávat prostřednictvím mobilního telefonu.

5.3 Případy užití

Na základě výše popsaných neformálních požadavků byl vytvořen diagram aktérů a případů užití, který je přílohou této práce. Jednotlivé případy užití (dále jen UC – use case) byly rozpracovány do podoby scénářů.



Obr. 8. Diagram aktérů.

5.3.1 Uživatelé

V této podkapitole budou popsány scénáře týkající se správy uživatelů aplikace, jejich registrace a autentizace.

UC01: Vytvoření uživatelského účtu

UC umožňuje nezaregistrovanému uživateli založení nového uživatelského účtu.

Akteři

Nezaregistrovaný uživatel

Podmínky pro spuštění

Nezaregistrovaný uživatel se nachází na úvodní stránce aplikace.

Základní tok

1. Uživatel zvolí možnost vytvořit nový účet.
2. Systém vygeneruje formulář.
3. Uživatel vyplní své uživatelské jméno, heslo a volitelně emailovou adresu a formulář odešle.
4. Systém provede validaci zadaných údajů. Uživatelské jméno musí být unikátní a heslo musí být minimálně 8 znaků dlouhé a musí obsahovat alespoň jedno malé písmeno, jedno velké písmeno a jedno číslo.
5. Systém vytvoří nový uživatelský účet.

Alternativní tok 1

- 4.1 Pokud vstup uživatele neprošel validací, systém na tuto skutečnost uživatele upozorní a nedovolí účet vytvořit.
- 4.2 Uživatel opraví neplatný vstup a pokračuje na bodu 3 základního toku.

Podmínky po dokončení

Je vytvořen uživatelský účet a uživatel se může přihlásit do systému.

UC02: Přihlášení uživatele

UC umožňuje přihlášení do systému již zaregistrovanému uživateli. Po přihlášení bude uživatel moci používat funkcionalitu aplikace.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Zaregistrovaný uživatel se nachází na úvodní stránce aplikace.

Základní tok

1. Uživatel zvolí možnost přihlášení do systému.
2. Systém vygeneruje formulář.
3. Uživatel zadá své uživatelské jméno a heslo a formulář odešle.
4. Systém autentizuje uživatele.

Alternativní tok 1

4.1 Uživatele nelze autentizovat, protože nezadal správnou kombinaci uživatelského jména a hesla. Uživatel je upozorněn na nesprávnost zadaných údajů.

4.2 Uživatel opraví zadané údaje a pokračuje na bodu 3 základního toku.

Podmínky po dokončení

Byla ověřena identita uživatele na základě znalosti kombinace uživatelského jména a hesla. Uživatel je přesměrován na hlavní stránku chráněné části aplikace.

UC03: Změna údajů uživatele

UC umožňuje přihlášenému uživateli změnit své přístupové údaje.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost úpravy svého uživatelského účtu.
2. Systém vygeneruje formulář pomocí kterého uživatel může změnit nebo doplnit své údaje.
3. Uživatel musí zadat své stávající přístupové heslo. Dále uživatel může vyplnit své nové uživatelské heslo nebo novou emailovou adresu. Po zadání údajů, které chce změnit, formulář odešle.
4. Systém provede validaci zadaných údajů.
5. Systém uloží nově zadané údaje.

Alternativní tok 1

4.1 Jestliže uživatel zadal špatné stávající přístupové heslo nebo nové heslo, které nespĺňuje požadavky na minimální sílu hesla, případně pokud zadal nevalidní emailovou adresu, je na tento fakt upozorněn a údaje nebudou změněny.

4.2 Uživatel opraví zadané údaje a pokračuje krokem 3 základního scénáře.

Podmínky po dokončení

V systému jsou uloženy nové údaje o uživateli.

UC04: Obnovení uživatelského účtu při zapomenutí hesla

UC umožňuje uživateli obnovit přístup k uživatelskému účtu při zapomenutí přístupového hesla.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Zaregistrovaný uživatel dříve systému poskytl svoji emailovou adresu a nyní se nachází na úvodní stránce aplikace.

Základní tok

1. Uživatel vybere možnost obnovení zapomenutého hesla.
2. Systém vygeneruje formulář.
3. Uživatel do formuláře zadá své uživatelské jméno.
4. Systém validuje zadaný údaj.
5. Systém odešle na emailovou adresu uživatele zprávu obsahující odkaz s unikátním kódem. Tento odkaz je platný po dobu jedné hodiny od vygenerování.
6. Uživatel přejde na zaslanoou adresu.
7. Systém validuje předaný unikátní kód a vygeneruje formulář.
8. Uživatel do formuláře zadá své uživatelské jméno a nové heslo a formulář odešle.
9. Systém validuje zaslanoé údaje.
10. Systém změní přístupové heslo uživatele.

Alternativní tok 1

4.1 Jestliže uživatel zadal neexistující uživatelské jméno nebo u daného uživatele není dostupná informace o jeho emailové adrese, tak UC končí. Uživatel na tuto skutečnost není ale nijak upozorněn, aby nebylo možné tímto způsobem zjistit existující uživatelská jména uložená v systému.

Alternativní tok 2

7.1 Pokud vypršela časová platnost unikátního kódu, je na tuto skutečnost uživatel upozorněn a UC končí.

Alternativní tok 3

9.1 Pokud uživatel zadal heslo, které nesplňuje požadavky na minimální sílu hesla, je na tento fakt upozorněn a údaje nebudou změněny.

9.2 Uživatel opraví zadané údaje a pokračuje krokem 8 základního scénáře.

Alternativní tok 4

9.1 Pokud uživatel vyplnil neplatné uživatelské jméno nebo předaný unikátní kód nebyl vygenerován pro daného uživatele, tak UC končí. Uživatel na tuto skutečnost není ale nijak upozorněn, aby nebylo možné tímto způsobem zjistit existující uživatelská jména uložená v systému.

Podmínky po dokončení

Uživateli je umožněno přihlásit se do systému pomocí nového hesla.

UC05: Odstranění uživatelského účtu

UC umožňuje uživateli odstranění jeho uživatelského účtu včetně všech uložených souvisejících dat.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost odstranění uživatelského účtu.
2. Systém vygeneruje formulář a upozorní uživatele na nenávratnost tohoto kroku a doporučí mu provést zálohu jeho dat.
3. Uživatel zadá své přihlašovací heslo a odešle formulář.
4. Systém validuje zadaný údaj.
5. Všechna data uživatele včetně jeho uživatelského účtu jsou odstraněna ze systému.
Uživatel je odhlášen a přesměrován na úvodní stránku aplikace.

Alternativní tok 1

- 4.1 Jestliže uživatel zadal špatné přístupové heslo je na tento fakt upozorněn a účet nebude odstraněn.
- 4.2 Uživatel opraví zadané údaje a pokračuje krokem 3 základního scénáře.

Podmínky po dokončení

Uživatel je odhlášen ze systému a všechna jeho data jsou odstraněna.

UC06: Odstranění uživatelského účtu jiného uživatele

UC umožňuje správci systému odstranit uživatelský účet jiného uživatele v systému. Slouží především k odstranění neaktivních uživatelů systému.

Akteři

Správce systému

Podmínky pro spuštění

Správce systému je přihlášený.

Základní tok

1. Systém vygeneruje seznam všech uživatelů systému. U každého zobrazí jeho uživatelské jméno, emailovou adresu a datum posledního přihlášení.
2. Správce vybere uživatele, kterého chce odstranit.
3. Systém si vyžádá potvrzení kroku, při kterém vypíše uživatelské jméno, emailovou adresu a datum posledního přihlášení vybraného uživatele.
4. Správce potvrdí svůj záměr odstranit vybraného uživatele.
5. Všechna data vybraného uživatele včetně jeho uživatelského účtu jsou odstraněna ze systému.

Alternativní tok 1

4.1 Pokud správce nepotvrdí záměr odstranit vybraného uživatele, tak UC končí.

Podmínky po dokončení

Vybraný uživatel je odstraněn ze systému.

UC07: Odhlášení ze systému

UC umožňuje přihlášenému uživateli odhlášení ze systému.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost odhlášení ze systému.
2. Systém uživatele odhlásí ze systému.
3. Uživatel je přesměrován na úvodní stránku aplikace.

Podmínky po dokončení

Uživatel je odhlášen ze systému a nachází se na úvodní stránce aplikace.

5.3.2 Správa účtů

V této podkapitole budou popsány scénáře týkající se správy účtů. Účet je abstrakcí jakéhokoliv způsobu držení finančních prostředků, které chce uživatel sledovat zvlášť. Může tak jít o běžný účet, spořicí účet, stavební spoření, hotovost obecně, hotovost v peněžence, v trezoru apod.

UC08: Vytvoření nového účtu

UC umožňuje uživateli vytvořit nový účet, ke kterému bude moci přiřazovat transakce.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost vytvoření nového účtu.
2. Systém vygeneruje formulář.
3. Uživatel vyplní název účtu, počáteční stav finančních prostředků a odešle formulář.
4. Systém vytvoří účet a uloží zadané údaje.

Podmínky po dokončení

Nový účet uživatele je uložen v systému.

UC09: Editace účtu

UC umožňuje uživateli upravit údaje o existujícím účtu.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu účtů vybere účet, který si přeje upravit.
2. Systém vygeneruje formulář, jehož editační pole jsou vyplněné aktuálními údaji o účtu.
3. Uživatel může změnit název účtu nebo počáteční stav finančních prostředků a odešle formulář.
4. Systém uloží změněné údaje.

Podmínky po dokončení

V systému jsou uloženy aktualizované údaje o účtu.

UC10: Odstranění účtu

UC umožňuje uživateli odstranit existující účet včetně všech s ním spojených dat.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu účtů vybere účet, který si přeje odstranit a zvolí u něj možnost odstranit účet.
2. Systém uživatele informuje o nenávratnosti tohoto kroku a vyžádá si potvrzení uživatele.
3. Uživatel potvrdí svůj úmysl odstranit vybraný účet.
4. Vybraný účet je odstraněn ze systému včetně všech s ním spojených dat.

Alternativní tok 1

3.1 Pokud uživatel nepotvrdí svůj úmysl odstranit vybraný účet, tak UC končí.

Podmínky po dokončení

Účet je odstraněn ze systému včetně všech s ním spojených dat.

UC11: Zobrazení grafů vývoje příjmů, výdajů a bilance

UC umožňuje uživateli zobrazit grafů vývoje příjmů, výdajů a bilance na zvolených účtech.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí zobrazení statistik o účtech.
2. Systém vygeneruje formulář.
3. Uživatel ve formuláři vybere účty, které chce do grafů zahrnout a zvolí časové období.
4. Systém vygeneruje grafy vývoje součtu příjmů, výdajů a celkové bilance na všech vybraných účtech za zvolené časové období. Data budou agregovaná buď po dnech, týdnech, měsících nebo letech dle délky zvoleného časového období.

Alternativní tok 1

4.1 Pokud ve vybraném období nejsou na zvolených účtech evidovány žádné transakce, nebudou grafy zobrazeny.

Podmínky po dokončení

Uživateli jsou zobrazeny grafy vývoje příjmů, výdajů a bilance na zvolených účtech za zvolené časové období.

5.3.3 Správa štítků

V této podkapitole jsou popsány scénáře, které se týkají správy štítků. Štítky slouží ke kategorizaci transakcí.

UC12: Vytvoření nového štítku

UC umožňuje uživateli vytvořit nový štítek. Po jeho vytvoření k němu bude možné přiřazovat transakce.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost vytvoření nového štítku.
2. Systém vygeneruje formulář.
3. Uživatel vyplní název štítku a odešle formulář.
4. Systém ověří unikátnost zadaného názvu a vytvoří nový štítek.

Alternativní tok 1

4.1 Jestliže uživatel zadal název již existujícího štítku, je na tuto skutečnost upozorněn a musí zvolit jiný název. UC pokračuje krokem 3 základního toku.

Podmínky po dokončení

Nový štítek je uložen v systému a je možné k němu přiřazovat transakce.

UC13: Editace štítku

UC umožňuje uživateli upravit údaje o existujícím štítku.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu štítků vybere štítek, který si přeje upravit.
2. Systém vygeneruje formulář umožňující změnit název štítku.
3. Uživatel změní název štítku a odešle formulář.
4. Systém ověří unikátnost nového názvu a uloží změny.

Alternativní tok 1

4.1 Jestliže uživatel zadal název již existujícího štítku, je na tuto skutečnost upozorněn a musí zvolit jiný název. UC pokračuje krokem 3 základního toku.

Podmínky po dokončení

V systému je uložen nový název štítku.

UC14: Odstranění štítku

UC umožňuje uživateli odstranit existující štítek ze systému.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu štítků vybere štítek, který si přeje odstranit a zvolí u něj možnost odstranit štítek.
2. Systém uživatele informuje o nenávratnosti tohoto kroku a vyžádá si jeho potvrzení.
3. Uživatel potvrdí svůj úmysl odstranit vybraný štítek.
4. Vybraný štítek je odstraněn ze systému. Všechny transakce, které byly tímto štítkem označeny zůstávají nadále v systému, ale už nejsou daným štítkem označeny.

Alternativní tok 1

2.1 Pokud uživatel chce odstranit štítek, který je spojen s existujícím vozidlem, tak je informován o nemožnosti odstranění daného štítku a UC končí.

Alternativní tok 2

2.1 Pokud uživatel chce odstranit štítek, který je spojen s existujícím rozpočtem, tak je informován o nemožnosti odstranění daného štítku a UC končí.

Alternativní tok 3

3.1 Pokud uživatel nepotvrdí svůj úmysl odstranit vybraný štítek, tak UC končí.

Podmínky po dokončení

Štítek je odstraněn ze systému a nejsou jím označeny žádné existující transakce.

UC15: Zobrazení grafů vývoje příjmů a výdajů za zvolený štítek

UC umožňuje uživateli zobrazení grafů vývoje příjmů a výdajů za vybraný štítek ve zvoleném časovém období.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu štítků vybere štítek, pro který chce zobrazit grafy a zvolí možnost zobrazení statistik.
2. Systém vygeneruje formulář.
3. Uživatel ve formuláři zvolí časové období a vybere účty, které se do grafu mají zahrnout a formulář odešle.
4. Systém zobrazí grafy vývoje příjmů a výdajů označených vybraným štítkem za zvolené časové období. Příjmy a výdaje jsou agregované po dnech, týdnech, měsících nebo letech v závislosti na délce zvoleného časového období.

Alternativní tok 1

4.1 Pokud ve zvoleném časovém období nejsou evidovány žádné transakce označené vybraným štítkem, tak grafy nejsou zobrazeny.

Podmínky po dokončení

Uživateli jsou zobrazeny grafy vývoje příjmů a výdajů označených vybraným štítkem za zvolené časové období.

5.3.4 Správa transakcí

V této podkapitole jsou popsány scénáře, které se týkají správy transakcí. Transakce jsou základním kamenem aplikace a slouží k evidování jednotlivých pohybů na účtech.

UC16: Vytvoření nové transakce

UC umožňuje uživateli vytvořit novou transakci.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost vytvoření nové transakce.
2. Systém vygeneruje formulář.
3. Uživatel zvolí, zda se jedná o příjem, výdaj nebo o převod mezi účty, dále vybere účet, ke kterému se transakce vztahuje (v případě převodu mezi účty uživatel vybere zdrojový a cílový účet), nominální hodnotu a datum uskutečnění transakce. Volitelně uživatel přidá popis transakce a označí transakci štítky. První zadaný štítek je považován za hlavní kategorizační štítek, další štítky jsou upřesňující. Uživateli jsou nabízeny existující štítky, ale přitom může zadat jakýkoliv text, který se použije pro případné vytvoření nového štítku. Nakonec uživatel odešle formulář.
4. Systém vytvoří novou transakci. V případě, že uživatel transakci označil štítkem, který ještě neexistuje, tak je automaticky vytvořen nový štítek.

Alternativní tok 1

3.1 Uživatel se může rozhodnout současně vytvořit novou opakovanou transakci, v tom případě je scénář rozšířen o UC21: Vytvoření nové opakované transakce.

Alternativní tok 2

3.1 Uživatel se může rozhodnout současně s transakcí evidovat také tankování vozidla, v tom případě je scénář rozšířen o UC30: Evidence tankování vozidla.

Podmínky po dokončení

Nová transakce a je uložena v systému. V případě, kdy uživatel označil transakci neexistujícím štítkem, je tento štítek vytvořen.

UC17: Editace transakce

UC umožňuje uživateli upravit existující transakci.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu transakcí vybere transakci, kterou si přeje upravit.
2. Systém vygeneruje formulář umožňující změnit nominální hodnotu, datum uskutečnění transakce, její popis a označení transakce štítky.
3. Uživatel provede požadované změny a odešle formulář.
4. Systém uloží změny.

Podmínky po dokončení

V systému jsou uloženy změny provedené uživatelem.

UC18: Odstranění transakce

UC umožňuje uživateli odstranit existující transakci ze systému.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu transakcí vybere transakci, kterou si přeje odstranit a zvolí u ní možnost odstranit transakci.
2. Systém uživatele informuje o nenávratnosti tohoto kroku a vyžádá si potvrzení uživatele.
3. Uživatel potvrdí svůj úmysl odstranit vybranou transakci.
4. Vybraná transakce je odstraněna ze systému.

Alternativní tok 1

3.1 Pokud uživatel nepotvrdí svůj úmysl odstranit vybranou transakci, tak UC končí.

Podmínky po dokončení

Transakce je odstraněna ze systému.

UC19: Rozdělení existující transakce

UC umožňuje rozdělit existující transakci na více samostatných transakcí. To může být vhodné například v případě, kdy chce uživatel evidovat některé položky z jednoho nákupu zvlášť. Původní transakce vždy zůstává v systému, ale je snížena její nominální hodnota o hodnotu nově vytvořené transakce.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu transakcí vybere transakci (příjem nebo výdaj – převod mezi účty rozdělit nelze), kterou si přeje rozdělit a zvolí u ní možnost rozdělit transakci.
2. Systém vygeneruje formulář určený pro vytvoření nové transakce.
3. Uživatel ve formuláři vyplní informace o nové transakci, konkrétně nominální hodnotu, datum, volitelně popis a označí transakci štítky. Nakonec uživatel odešle formulář.
4. Systém vytvoří novou transakci na základě odeslaných údajů. Transakce je automaticky přiřazena k účtu podle rozdělované transakce. Nominální hodnota rozdělované transakce je snížena o hodnotu nově vytvořené transakce.

Podmínky po dokončení

V systému je evidována nová transakce. Součet hodnoty nové a původní transakce je stejný jako u původní transakce před jejím rozdělením.

UC20: Zobrazení grafu podílu štítků na celkových výdajích

UC umožňuje uživateli zobrazit graf podílu jednotlivých štítků na jeho celkových výdajích.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost zobrazení grafu podílu štítků na jeho výdajích.
2. Systém vygeneruje formulář.
3. Uživatel vybere časové období, zvolí účty, které chce do grafu zahrnout a formulář odešle.
4. Systém vygeneruje koláčový graf podílu jednotlivých hlavních štítků a sloupcový graf podílu všech štítků na celkových výdajích za zvolené časové období na vybraných účtech.

Alternativní tok 1

4.1 Pokud v systému nejsou evidované žádné transakce ve zvoleném časovém období, tak grafy nejsou zobrazeny.

Podmínky po dokončení

Uživateli je zobrazen koláčový graf podílu jednotlivých hlavních štítků a sloupcový graf podílu všech štítků na celkových výdajích.

5.3.5 Správa opakovaných transakcí

V této podkapitole jsou popsány scénáře, které se týkají správy opakovaných transakcí. Na základě uložených opakovaných transakcí systém automaticky vytváří běžné transakce dle zadané frekvence opakování. Opakovaná transakce je pouze šablonou pro automaticky zapisované běžné transakce, tudíž její změna nebo odstranění nemá vliv na již automaticky vytvořené běžné transakce.

UC21: Vytvoření nové opakované transakce

UC umožňuje uživateli vytvořit novou opakovanou transakci. Tento scénář rozšiřuje scénář UC16: Vytvoření nové transakce.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel zahájil scénář UC16: Vytvoření nové transakce.

Základní tok

1. Systém k formuláři pro zadávání transakce vygeneruje další formulář pro vytváření opakovaných transakcí.
2. Uživatel zvolí možnost vytvoření opakované transakce, vybere frekvenci jejího opakování (denní, týdenní, měsíční) a odešle formulář.
3. Systém kromě běžné transakce vytvoří i opakovanou transakci, která bude obsahovat stejné údaje jako právě vytvořená běžná transakce.
4. Systém bude automaticky vytvářet nové běžné transakce dle nastavení opakované transakce.

Podmínky po dokončení

Nová opakovaná transakce je uložena v systému a systém na základě ní automaticky vytváří nové běžné transakce.

UC22: Editace opakované transakce

UC umožňuje uživateli upravit existující opakovanou transakci, která slouží jako šablona pro automaticky vytvářené transakce.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu opakovaných transakcí vybere transakci, kterou si přeje upravit.
2. Systém vygeneruje formulář umožňující změnu nominální hodnoty, data uskutečnění první transakce, jejího popisu, frekvence opakování a označení transakce štítky.
3. Uživatel upraví údaje a odešle formulář.
4. Systém uloží změny.

Podmínky po dokončení

V systému jsou uloženy změny provedené uživatelem.

UC23: Odstranění opakované transakce

UC umožňuje uživateli odstranit existující opakovanou transakci ze systému. Tím nedojde k odstranění již automaticky vytvořených běžných transakcí na základě odstraněné opakované transakce.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu opakovaných transakcí vybere transakci, kterou si přeje odstranit a zvolí u ní možnost odstranit transakci.
2. Systém uživatele informuje o nenávratnosti tohoto kroku a vyžádá si potvrzení uživatele.
3. Uživatel potvrdí svůj úmysl odstranit vybranou transakci.
4. Vybraná transakce je odstraněna ze systému.

Alternativní tok 1

3.1 Pokud uživatel nepotvrdí svůj úmysl odstranit vybranou transakci, tak UC končí.

Podmínky po dokončení

Opakovaná transakce je odstraněna ze systému a nebudou se na základě ní již automaticky vytvářet další běžné transakce.

5.3.6 Správa rozpočtů

V této podkapitole jsou popsány scénáře, které se týkají správy rozpočtů. Rozpočty slouží k omezení výdajů ve zvoleném časovém období. Pomocí rozpočtu uživatel zjistí, jakou část stanovené cílové částky již vydal a může podle toho přizpůsobit výdaje ve zbytku stanoveného období. Rozpočty mohou být jednorázové nebo opakované, například měsíční. Opakovaný rozpočet se po uplynutí stanovené doby sám resetuje a počítá výdaje opět od nuly. Je možné, aby se do rozpočtu započítávaly pouze určité transakce označené vybraným štítkem.

UC24: Vytvoření nového rozpočtu

UC umožňuje uživateli vytvořit nový rozpočet.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost vytvoření nového rozpočtu.
2. Systém vygeneruje formulář.
3. Uživatel vyplní název rozpočtu, omezující částku a zvolí, zda se jedná o jednorázový nebo opakovaný rozpočet. U jednorázového rozpočtu musí zadat datum začátku a konce, u pravidelného je třeba zadat počáteční datum a po jakém období se bude rozpočet opakovat (dny, týdny, měsíce). Volitelně může uživatel vyplnit popis rozpočtu a specifikovat výčet štítků, na které se bude vztahovat. Nakonec uživatel odešle formulář.
4. Systém vytvoří nový rozpočet.

Podmínky po dokončení

Nový rozpočet je uložen v systému.

UC25: Editace rozpočtu

UC umožňuje uživateli upravit existující rozpočet.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu rozpočtů vybere rozpočet, který si přeje upravit.
2. Systém vygeneruje formulář, ve kterém je možné změnit název rozpočtu, omezující částku, popis rozpočtu a výčet štítků, na které se vztahuje. U jednorázového rozpočtu je také možné upravit datum začátku a konce rozpočtu, u pravidelného počáteční datum a frekvenci opakování.
3. Uživatel provede požadované změny a odešle formulář.
4. Systém uloží změny.

Podmínky po dokončení

V systému jsou uloženy změny provedené uživatelem.

UC26: Odstranění rozpočtu

UC umožňuje uživateli odstranit existující rozpočet ze systému.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu rozpočtů vybere rozpočet, který si přeje odstranit a zvolí u něj možnost odstranit rozpočet.
2. Systém uživatele informuje o nenávratnosti tohoto kroku a vyžádá si potvrzení uživatele.
3. Uživatel potvrdí svůj úmysl odstranit vybraný rozpočet.
4. Vybraný rozpočet je odstraněn ze systému.

Alternativní tok 1

3.1 Pokud uživatel nepotvrdí svůj úmysl odstranit vybraný rozpočet, tak UC končí.

Podmínky po dokončení

Rozpočet je odstraněn ze systému.

5.3.7 Správa vozidel

V této podkapitole jsou popsány scénáře, které se týkají správy vozidel. Pokud uživatel eviduje v systému vozidlo, může sledovat s ním související náklady. Každé vozidlo je spojeno se štítkem, kterým uživatel označuje transakce týkající se vozidla.

UC27: Vytvoření nového vozidla

UC umožňuje uživateli vytvořit nové vozidlo.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost vytvoření nového vozidla.
2. Systém vygeneruje formulář.
3. Uživatel vyplní název vozidla a odešle formulář.
4. Systém ověří, zda neexistuje vozidlo se stejným názvem a vytvoří nové vozidlo a nový štítek.

Alternativní tok 1

4.1 Pokud v systému již existuje vozidlo se stejným názvem, je na tuto skutečnost uživatel upozorněn a musí zvolit jiný název. UC dále pokračuje krokem 3 základního toku.

Alternativní tok 2

4.1 Pokud v systému již existuje štítek se stejným názvem, jako je název vozidla, bude vozidlo spojeno s ním a nedojde k vytvoření nového štítku.

Podmínky po dokončení

Nové vozidlo a štítek se stejným názvem jsou uloženy v systému.

UC28: Editace vozidla

UC umožňuje uživateli upravit existující vozidlo.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu vozidel vybere vozidlo, které si přeje upravit.
2. Systém vygeneruje formulář umožňující změnu názvu vozidla.
3. Uživatel změní název a odešle formulář.
4. Systém ověří, zda neexistuje vozidlo a štítek se stejným názvem a změní název vozidla a jemu přiřazeného štítku.

Alternativní tok 1

4.1 Pokud v systému již existuje vozidlo s novým názvem, je na tuto skutečnost uživatel upozorněn a musí zvolit jiný název. UC dále pokračuje krokem 3 základního toku.

Alternativní tok 2

4.1 Pokud v systému již existuje štítek se stejným názvem, jako je nový název vozidla, je na tuto skutečnost uživatel upozorněn a musí zvolit jiný název. UC dále pokračuje krokem 3 základního toku.

Podmínky po dokončení

Vozidlo a přiřazený štítek mají nový název.

UC29: Odstranění vozidla

UC umožňuje uživateli odstranit existující vozidlo ze systému.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu vozidel vybere vozidlo, které si přeje odstranit a zvolí u něj možnost odstranit vozidlo.
2. Systém uživatele informuje o nenávratnosti tohoto kroku a vyžádá si potvrzení uživatele.
3. Uživatel potvrdí svůj úmysl odstranit vybrané vozidlo.
4. Systém odstraní vybrané vozidlo a s ním spojený štítek.

Alternativní tok 1

3.1 Pokud uživatel nepotvrdí svůj úmysl odstranit vybrané vozidlo, tak UC končí.

Alternativní tok 2

3.1 Pokud existuje alespoň jedna transakce označená štítkem vozidla, tak nedojde k odstranění štítku a ten tedy zůstává nadále v systému.

Podmínky po dokončení

Vozidlo je odstraněno ze systému. Štítek vozidla je odstraněn, pokud jím nejsou označeny žádné transakce.

UC30: Evidence tankování vozidla

UC umožňuje evidovat tankování pohonných hmot vozidla. Tento scénář rozšiřuje scénář UC16: Vytvoření nové transakce.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel zahájil scénář UC16: Vytvoření nové transakce a v systému eviduje alespoň jedno vozidlo.

Základní tok

1. Systém k formuláři pro vytvoření transakce vygeneruje navíc formulář pro evidenci tankování pohonných hmot.
2. Uživatel do formuláře zadá počet natankovaných litrů nebo cenu za litr (druhá hodnota je vždy dopočítána automaticky). Dále vybere vozidlo, kterého se tankování týká a zaznamená, zda se jedná o natankování plné nádrže a pokud ano, tak zadá stav tachometru vozidla. Nakonec uživatel odešle formulář.
3. Systém kromě vytvoření transakce zaznamená údaj o tankování pohonných hmot.

Podmínky po dokončení

V systému je evidováno tankování pohonných hmot vybraného vozidla.

UC31: Zobrazení statistik vozidla

UC umožňuje uživateli zobrazit statistiky vybraného vozidla.

Akteři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel v seznamu vozidel vybere vozidlo a zvolí možnost zobrazení statistik.
2. Systém vygeneruje formulář.
3. Uživatel ve formuláři vybere časové období, pro které chce statistiky zobrazit a formulář odešle.
4. Systém vygeneruje graf vývoje ceny pohonných hmot a kilometrového nájezdu vozidla dle evidovaných tankování. Dále zobrazí průměrnou spotřebu vozidla, průměrnou cenu za kilometr, celkovou cenu pohonných hmot, cenu dalších nákladů (transakce označené štítkem vozidla) za vybrané období a graf průměrné spotřeby mezi jednotlivým tankováním do plné nádrže.

Alternativní tok 1

4.1 Pokud uživatel nezaznamenal alespoň dvě tankování do plné nádrže ve vybraném období, tak systém nezobrazí údaje týkající se kilometrového nájezdu vozidla, průměrné spotřeby vozidla a nákladů na kilometr.

Podmínky po dokončení

Uživateli jsou zobrazeny dostupné statistiky o vozidle.

5.3.8 Zálohování a obnova dat

V této podkapitole jsou popsány scénáře, které se týkají zálohování a obnovy dat.

UC32: Exportování dat do CSV souboru

UC umožňuje zálohovat uživatelská data do CSV souboru.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost exportování dat do CSV souboru.
2. Systém vygeneruje CSV soubor obsahující všechna data uživatele.
3. Uživatel si uloží vygenerovaný soubor.

Podmínky po dokončení

Uživatel má uložený soubor se zálohou svých dat.

UC33: Importování dat z CSV souboru

UC umožňuje uživateli obnovit zálohovaná uživatelská data.

Aktéři

Zaregistrovaný uživatel

Podmínky pro spuštění

Uživatel je přihlášen do systému.

Základní tok

1. Uživatel zvolí možnost importování dat z CSV souboru.
2. Systém vygeneruje formulář.
3. Uživatel pomocí formuláře odešle CSV soubor s daty na server.
4. Systém ověří kompatibilitu souboru a provede import dat.

Alternativní tok 1

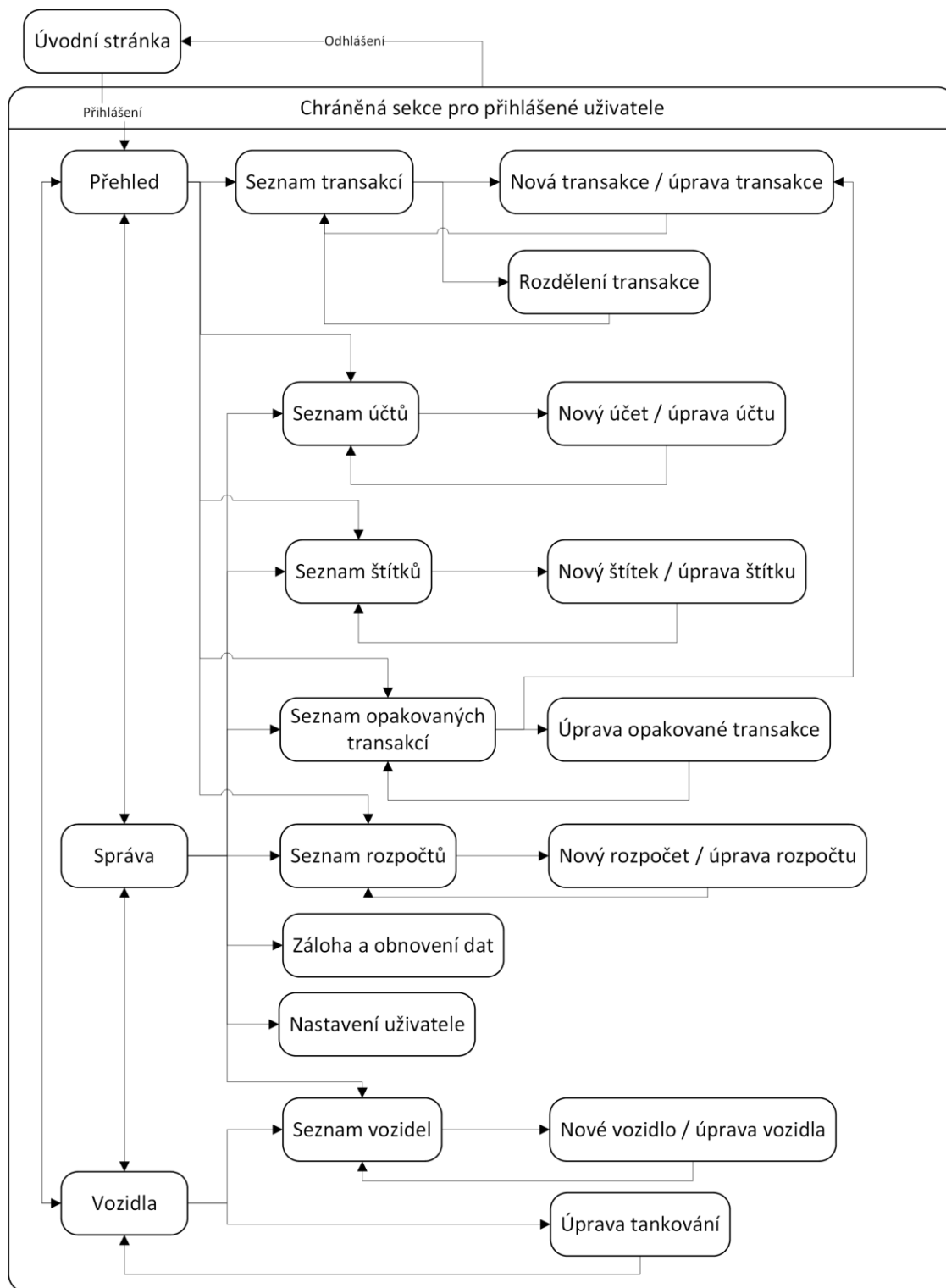
4.1 Pokud uživatel odeslal poškozený nebo nekompatibilní soubor, je na tuto skutečnost upozorněn a UC pokračuje krokem 3 základního scénáře.

Podmínky po dokončení

V systému jsou uložena data z CSV souboru.

6 UŽIVATELSKÉ ROZHRAŇÍ

Po specifikování funkčních požadavků kladených na aplikaci a jejich rozpracování do formy uživatelských scénářů v předchozí kapitole, bude tato kapitola věnována návrhu uživatelského rozhraní aplikace.



Obr. 9. Schéma propojení jednotlivých stránek uživatelského rozhraní.

Na Obr. 9 je znázorněno schéma propojení jednotlivých stránek webové aplikace. Výchozím bodem je veřejná úvodní stránka, ze které se uživatel po přihlášení přesune do chráněné sekce. Z každé stránky v chráněné sekci se uživatel bude moci odhlásit. Chráněná sekce je rozdělena na tři hlavní části – *Přehled*, *Správa* a *Vozidla*. Uživatel bude moci kdykoliv přejít z jedné části do druhé pomocí hlavního menu nebo křížových odkazů, které se v jednotlivých částech vyskytují.

V této kapitole budou dále prezentovány náčrty některých důležitých stránek aplikace. Všechny náčrty uživatelského rozhraní byly vytvořeny v open-source aplikaci Pencil [16] a při jejich vytváření byl brán ohled na doporučení v knihách *Webdesign Nenutíte uživatele přemýšlet* [2] a *Použitelnost webových stránek* [17].

6.1 Přehled a evidence transakcí

Stránka *Přehled* je hlavní stránkou chráněné sekce aplikace a slouží ke správě transakcí a zobrazení statistik. Jednou z nejdůležitějších funkcí aplikace je evidence transakcí, návrh formuláře pro zadávání nového výdaje je na Obr. 10.

The image shows a web form titled "Nový výdaj" (New transaction) from the application "OsobníFinance.eu". The form is divided into two sections: "Nový výdaj" and "Tankování" (Refueling). The "Nový výdaj" section includes fields for "Účet" (Account), "Částka" (Amount), "Šřítky" (Receipt), "Datum" (Date), and "Popis" (Description). The "Tankování" section includes fields for "Vozidlo" (Vehicle), "Litry" (Liters), "Cena za litr" (Price per liter), "Stav tachometru" (Mileage), and a checkbox for "Plná nádrž" (Full tank). The form also features a "Opakovaná transakce" (Recurring transaction) section with radio buttons for "Den" (Daily), "Týden" (Weekly), and "Měsíc" (Monthly), and checkboxes for each day of the week. The form is styled with a dark header and footer, and includes "Uložit" (Save) and "Zpět" (Back) buttons.

OsobníFinance.eu Přehled Správa Vozidla

Uložit Zpět

Nový výdaj

Účet: Účet 1

Částka: 300 Kč

Šřítky: Šřítek 1

Datum: 20.1.2014 Dnes Včera #

Popis:

Opakovaná transakce

Opakovat každý: Den
 Týden
 Měsíc

Tankování

Vozidlo: Škoda Octavia

Litry: 35

Cena za litr: 36,70

Plná nádrž

Stav tachometru: 190 250 km

Uložit Zpět

Obr. 10. Náčrt formuláře pro vytvoření nového výdaje.

OsobníFinance.eu
Přehled Správa Vozidla
Jméno uživatele Odhlásit se

Filtr
 Příjmy a výdaje
 Plánování
 Statistiky

Od # Do #

ÚčtyVšechny Zrušit výběr

ŠtítkyVšechny Zrušit výběr

Příjmy a výdaje

Příjmy					Výdaje							
30.1.2014	Štítek 2, Štítek 3	Popis transakce	1500 Kč	<input type="button" value="Upravit"/>	<input type="button" value="Rozdělit"/>	30.1.2014	Štítek 2, Štítek 3	Popis transakce	150 Kč	<input type="button" value="Upravit"/>	<input type="button" value="Rozdělit"/>	
20.1.2014	Štítek 2	Popis transakce	2500 Kč	<input type="button" value="Upravit"/>	<input type="button" value="Rozdělit"/>	20.1.2014	Štítek 2	Popis transakce	250 Kč	<input type="button" value="Upravit"/>	<input type="button" value="Rozdělit"/>	
2.1.2014	Štítek 5, Štítek 3	Popis transakce	15000 Kč	<input type="button" value="Upravit"/>	<input type="button" value="Rozdělit"/>	2.1.2014	Štítek 5, Štítek 3	Popis transakce	1000 Kč	<input type="button" value="Upravit"/>	<input type="button" value="Rozdělit"/>	
Celkem			19 000 Kč				Celkem			1 400 Kč		

Plánování

Plánované transakce

Datum	Účet	Štítky	Popis	Částka	
20.2.20	Účet 1	Štítek 2	Energie	3000 Kč	<input type="button" value="Upravit"/>
21.2.20	Účet 2	Štítek 2	Spořeni	1500 Kč	<input type="button" value="Upravit"/>

Rozpočty

Rozpočet 1

Rozpočet 2

Statistiky

Příjmy, výdaje a bilance

483 x 206

Celkem příjmy: 19 000 Kč Celkem výdaje: 1 400 Kč Bilance: 17 600 Kč

Stav na účtech

526 x 130

Účet	Počáteční stav	Koncový stav	Rozdíl
Účet 1	2 000 Kč	6 000 Kč	4 000 Kč
Účet 2	36 000 Kč	33 000 Kč	-3 000 Kč
Celkem	38 000 Kč	39 000 Kč	1 000 Kč

Podíl hlavních štítků na výdajích

479 x 180

Podíl štítků na výdajích

524 x 163

Obr. 11. Náčrt stránky Přehled zobrazené na monitoru s vysokým rozlišením.

Na Obr. 11 je celkový náčrt stránky *Přehled*. V horní části je filtr pro výběr časového období, účtů a štítků. Následují tabulky s přehledem příjmů, výdajů a plánovaných transakcí, které odpovídají zvoleným podmínkám. Dále bude zobrazeno plnění aktuálních rozpočtů a veškeré statistiky, které budou vypočítané pro právě zobrazené transakce ve formě grafů, které mohou být doplněny tabulkami.

Tato stránka bude také optimalizována pro zobrazení na mobilních zařízeních. Obr. 12 ukazuje její zobrazení na tabletu, kdy dojde ke změně dvousloupcového rozložení na jednosloupcové a Obr. 13 na mobilním telefonu, kdy budou viditelné pouze ty nejdůležitější prvky určené pro evidenci transakcí.



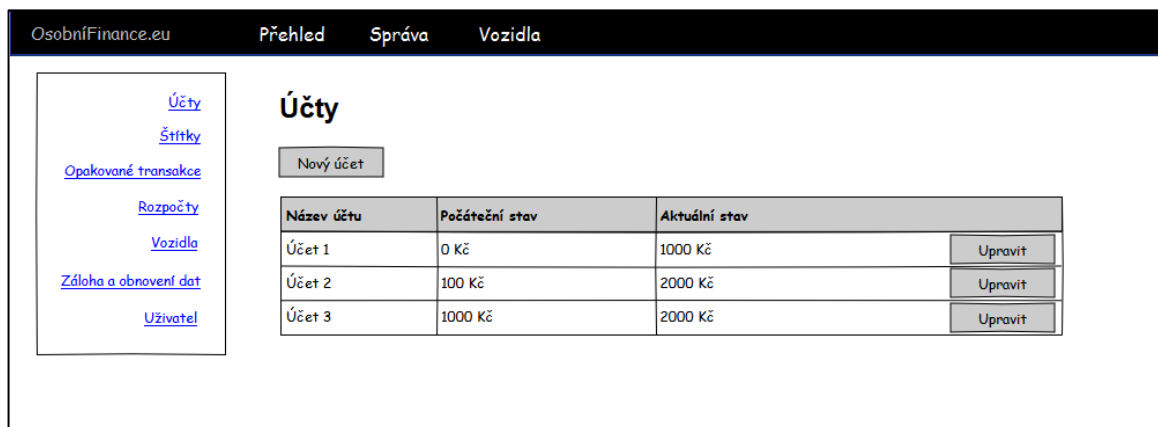
Obr. 12. Náčrt stránky Přehled zobrazené na tabletu.



Obr. 13. Náčrt stránky Přehled zobrazené na mobilním telefonu.

6.2 Správa

V sekci *Správa* bude uživatel moci spravovat účty, štítky, opakované transakce, rozpočty, vozidla a svůj uživatelský účet. Také zde bude mít dostupné funkce pro zálohování a obnovu dat. Uživatelské rozhraní této sekce bude založeno na tabulkách zobrazujících uložené objekty, jak ukazuje Obr. 14 a na formulářích pro jejich vytvoření nebo úpravu, jak ukazuje Obr. 15.



Obr. 14. Náčrt seznamu účtů v sekci *Správa*.

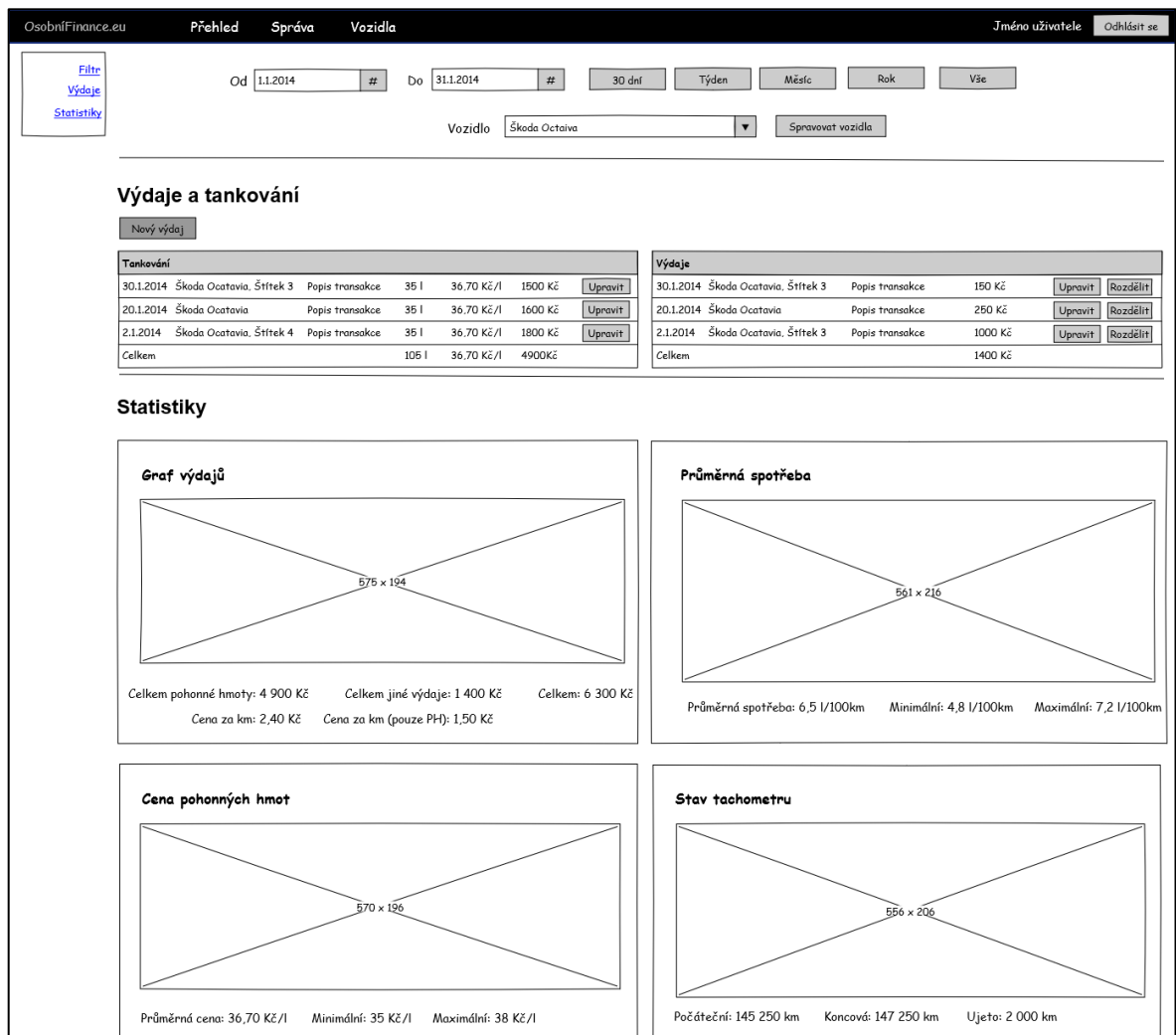


Obr. 15. Náčrt formuláře pro úpravu účtu v sekci *Správa*.

6.3 Vozidla

Stránka *Vozidla* bude sloužit k zobrazení statistik o vybraném vozidle. Náčrt stránky je na Obr. 16. V její horní části je filtr, pomocí kterého uživatel zvolí časové období a vybere vozidlo. Následují tabulky s evidovaným tankováním a ostatními výdaji, které se vztahují k vybranému vozidlu a časovému období. Dále budou na stránce zobrazeny statistiky o vozidle, jakými je například graf výdajů za vozidlo v čase včetně vypočítaných nákladů

za ujetý kilometr, graf průměrné spotřeby mezi tankováním do plné nádrže, graf ceny pohonných hmot v čase a graf stavu tachometru vozidla v čase.



Obr. 16. Náčrt stránky Vozidla sloužící pro zobrazení statistik o evidovaném vozidle.

6.4 Grafický styl uživatelského rozhraní

Uživatelské rozhraní bude vyvedeno v graficky jednoduchém stylu za použití design frameworku Bootstrap [18]. Konkrétní vzhled aplikace bude doladěn v průběhu vytváření prototypu, nicméně schéma uživatelského rozhraní by se již měnit nemělo.

7 NÁVRH IMPLEMENTACE

Navrhnutá aplikace pro správu financí bude implementována jako webová aplikace, půjde tedy o serverovou aplikaci, ke které budou uživatelé přistupovat prostřednictvím webového prohlížeče. K implementaci takového řešení je dnes možné přistoupit z různých úhlů pohledu a s použitím mnoha dostupných technologií a programovacích jazyků. V této kapitole budou mimo jiné popsány technologie, které navrhuji při implementaci použít a bude u nich zmíněno proč. Bude také nastíněno, jakým způsobem provést implementaci tak, aby se vyloučily zranitelnosti uvedené v kapitole 4 Zabezpečení webových aplikací.

7.1 Použité technologie a knihovny

V této podkapitole budou popsány použité serverové a klientské technologie a knihovny.

7.1.1 ASP.NET a C#

Platformu .NET a framework pro webové aplikace ASP.NET společnost Microsoft uvedla již v roce 2002 a od té doby je neustále vyvíjí. Velkou výhodou ASP.NET frameworku je právě podpora Microsoftu a velké komunity vývojářů. Každá jeho nová verze rozšiřuje nabídku již implementované často používané funkcionality, díky čemuž se mohou vývojáři stále efektivněji zaměřovat na řešení problémů především z obchodního hlediska namísto implementování již stokrát implementovaného. [19]

Pro psaní ASP.NET aplikací lze využít více programovacích jazyků, ale nejrozšířenější je C#. Ten stejně jako samotný framework prochází vývojem a nyní je dostupná již jeho pátá verze. Jedná se o objektově orientovaný jazyk, který se kompiluje do Common Intermediate Language (CIL), což je byte kód platformy .NET, který je spouštěný ve virtuálním stroji Common Language Runtime (CLR). CLR potom interpretuje CIL a převádí jeho instrukce na strojový kód. Jednou z výhod tohoto řešení je například automatická správa paměti, takže se vývojář nemusí starat o její alokaci a uvolňování. Jednotlivé aspekty programovacího jazyka C# jsou výborně popsány v knize C# in Depth [20].

Díky použití ASP.NET a jazyka C# je možné jádro aplikace implementovat velmi podobným způsobem jako kdyby se jednalo o klasickou desktop aplikaci.

7.1.2 ASP.NET MVC

ASP.NET nabízí několik možností, jak implementovat webové aplikace. V této práci bude využit framework ASP.NET MVC, který implementuje architektonický vzor Model-View-Controller (MVC). MVC striktně odděluje datový model, uživatelské rozhraní a aplikační logiku. Každý požadavek klienta je pomocí routing systému využívajícího zaregistrovaných vzorů v URL, předán k vyřízení správnému kontroléru, což je třída implementující definované rozhraní. Nad instancí této třídy je zavolána metoda, opět identifikovaná pomocí vzoru v URL. Zodpovědností metod kontroléru je získání požadovaných dat (modelu), případně vytvoření modelu ze zaslaných dat. Tento model kontrolér následně předá pohledu (view), který z něj vytvoří požadovaný výstup v uživatelském rozhraní. Výsledek je poté vrácen jako odpověď na klientův požadavek. [21]

ASP.NET MVC bylo zvoleno právě kvůli jasnému oddělení prezentační vrstvy od datového modelu a aplikační logiky. To přináší výhody především z hlediska přehlednosti a udržitelnosti aplikace, protože lze velmi snadno pokrýt aplikační logiku unit testy. Framework ASP.NET MVC má navíc odlišný vydávací cyklus od základního ASP.NET frameworku, takže nové verze jsou uvolňovány podstatně rychleji.

7.1.3 ASP.NET Web API

ASP.NET Web API je framework, pomocí něhož lze jednoduše vytvářet REST služby, které mohou svým konzumentům nabídnout různou funkcionalitu aplikace skrze protokol HTTP. Typickým konzumentem takových služeb může být například webová stránka, která dynamicky načítá data pomocí asynchronních požadavků. Další možností je například mobilní aplikace, která má vlastní uživatelské rozhraní a pomocí služby přistupuje k datům na serveru nebo s nimi manipuluje. Web API služba vrací odpovědi ve formátu JSON nebo XML v závislosti na klientovi, který požadavek odeslal. [21]

Web API bude použito u přehledových stránek aplikace, konkrétně na stránkách *Přehled* a *Vozidla*, na kterých není žádoucí, aby se uživatelé načítala celá stránka znovu v případě, kdy si například změní výčet zobrazených štítků. V takovém případě bude následovat asynchronní požadavek na Web API službu, která bude požádána o dodání chybějících dat. Tato data budou poté dynamicky přidána k již zobrazeným datům. Tento přístup zrychlí odezvu na akce uživatele a také bude šetřit množství přenesených dat, což je důležité

především v případě, kdy uživatel k aplikaci přistupuje prostřednictvím mobilního telefonu s pomalým nebo omezeným datovým spojením.

7.1.4 Microsoft SQL Server

Jako datové úložiště bude využit Microsoft SQL Server. ASP.NET s ním dobře spolupracuje a v případě, kdy by aplikace obsluhovala mnoho uživatelů, by mohl být výkon běžně dostupných hosting balíčků nedostatečný. V takové situaci by bylo možné aplikaci přesunout do cloud řešení Microsoft Azure, které jako databázový server využívá Microsoft SQL Azure a migrace databáze z Microsoft SQL Server na SQL Azure je řešitelná poměrně lehce. Jiné databázové servery se více či méně liší, srovnání zápisů konkrétních jazykových konstrukcí mezi systémy Microsoft SQL Server, Oracle a MySQL ukazuje kniha SQL Hotová řešení [22].

7.1.5 HTML5 a CSS3

Klientská část aplikace bude postavena na HTML5, který oproti svým starším variantám nabízí především mnohem lepší možnosti při vytváření přístupných stránek.

HTML5 zavádí nové sémantické značky, jako je `<header>` (hlavička stránky), `<footer>` (patička stránky), `<main>` (hlavní obsah), `<nav>` (navigace), `<section>` (sémanticky oddělená část stránky) a další. Dále je značně rozšířen výčet typů ovládacího prvku `<input>`, například o `tel` (zadávání telefonních čísel), `search` (vyhledávání), `url` (zadání URL adresy), `email` (zadání emailové adresy), `datetime` (zadání data a času), `date` (zadání data), `number` (zadání čísla) a mnohé další. [23]

Nové vlastnosti HTML5 pomáhají čtečkám obrazovky rozlišit význam jednotlivých prvků na stránce, navíc je využívají prohlížeče mobilních telefonů, které například při zadávání čísla zobrazí pouze numerickou klávesnici namísto kompletní klávesnice, což kromě přístupnosti zlepšuje i použitelnost.

Co se týče definice vzhledu aplikace, tak bude použito CSS3, které stejně jako HTML5 oproti svému předchůdci přináší spoustu vylepšení. Například zakulacené rohy a stín blokových elementů se ve starším CSS musely řešit pomocí různých speciálních technik, které jsou například detailně popsány v knize Mistrovství v CSS [24], zatímco v CSS3 na jejich vytvoření stačí aplikování jediného pravidla.

Je ovšem na místě zmínit také fakt, že HTML5 a CSS3 nejsou podporované staršími prohlížeči, jmenovitě například Internet Explorer většinu nových vlastností podporuje až od verze 10.0. Při používání nové vlastnosti je tedy dobré zvážit, zda stojí její využití za ztrátu kompatibility například s nějakým dosud rozšířeným prohlížečem. Při tomto rozhodování skvěle poslouží nástroj Can I use... [25].

7.1.6 Bootstrap a jQuery

Bootstrap je front-end framework poskytující nástroje pro rychlou tvorbu responsivních webových aplikací. Jedná se o propojení CSS stylů a JavaScript kódu, díky kterému vývojáři stačí jen respektovat určitá pravidla při psaní HTML markupu a bez nutnosti vytváření dalších stylů nebo skriptů získá v základu již dobře vypadající webovou stránku s kvalitní typografií, která bude mobile first (to znamená, že je výchozí styl primárně optimalizován pro mobilní telefony a až dodatečně upraven pro desktop prohlížeče pomocí media queries). Bootstrap nabízí spoustu grafických i funkčních komponent, které mají dobře dokumentované JavaScript API. [18]

Pro manipulaci s DOM (Document Object Model) stránky bude používána JavaScript knihovna jQuery [26], kterou také interně používá Bootstrap.

7.1.7 Angular.js

O několik odstavců výše jsem zmínil výhody MVC architektury na straně serveru a také jsem navrhnul využití Web API v přehledových stránkách. Klientská část aplikace na přehledových stránkách bude používat framework AngularJS, který zásadně usnadňuje implementaci MVC přístupu v JavaScriptu. [27]

7.1.8 Google Charts

Pro zobrazování grafů bude použit nástroj Google Charts, který na stránce dokáže vykreslit interaktivní a velmi široce nastavitelné grafy různých typů. Má dobře dokumentované JavaScript API a veškeré zpracování dat a vykreslování se odehrává přímo v klientském prohlížeči, žádná data nejsou odesílána na cizí servery. [28]

7.2 Implementace

Samotná implementace aplikace bude sestávat z několika knihoven, které budou oddělovat jednotlivé vrstvy architektury aplikace. Striktně budou především separované třídy

představující datový model aplikace a aplikační logika, pro kterou se tím pádem budou snáze psát unit testy. Prezentační vrstva bude knihovny z nižších vrstev používat pro manipulaci s daty, ale už nebude přidávat další aplikační logiku. Díky tomu bude možné současně prezentovat výstupy aplikace jednak pomocí klasických webových stránek, ale stejně tak zpřístupnit funkcionalitu aplikace přes služby Web API pro klientské skripty nebo v budoucnu třeba i jiným aplikacím (viz kapitola 9 Další rozvoj aplikace).

Na Obr. 17 je znázorněno schéma datového modelu aplikace. Podle něj budou data uložena v databázi a obdobně budou vypadat datové třídy, se kterými bude aplikace pracovat. Jednotlivé třídy detailněji popíše níže.

7.2.1 Uživatel

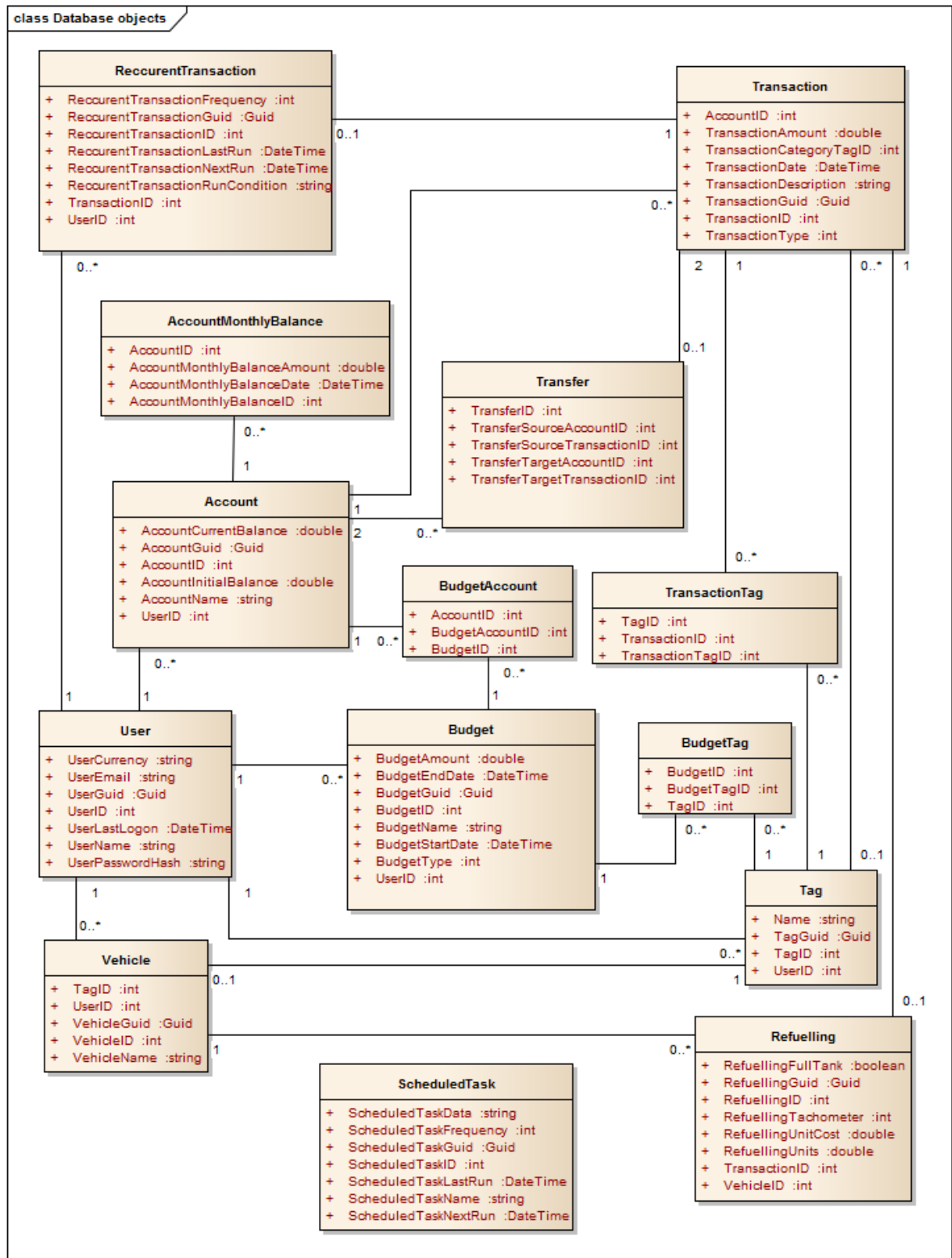
`User` představuje uživatele aplikace, ukládá jeho unikátní uživatelské jméno, hash hesla, email, datum a čas posledního přihlášení a měnu, kterou bude uživatel používat.

Uživatelské jméno a hash hesla správce aplikace bude uložen v konfiguračním souboru aplikace, protože správce nebude z bezpečnostních důvodů moci používat aplikaci jako běžný uživatel a tudíž není nutné mít jeho údaje uložené v databázi.

7.2.2 Účty

`Account` představuje účet, na kterém jsou soustředěny finanční prostředky. Kromě názvu je u něj uložen jeho počáteční stav a aktuální stav, který je aktualizován při zápisu každé transakce.

Do `AccountMonthlyBalance` se průběžně ukládá hodnota, o kterou se změnil stav účtu v daném měsíci. Aktualizace této hodnoty probíhá při každém uložení transakce. Tato třída je v aplikaci z důvodu vylepšení výkonu při získávání počátečního a koncového stavu účtu v uživatelem zvoleném časovém období. Běžně by bylo potřeba sečíst všechny transakce, které byly vytvořeny od založení účtu až po začátek vybraného období. S použitím této třídy stačí sečíst hodnoty za každý měsíc od vytvoření účtu a poté přičíst pouze transakce, které proběhly od začátku měsíce do dne zvoleného jako začátek období. To může přinést značnou úsporu, pokud vezmeme v potaz, že po ročním používání se jedná o součet 12 čísel místo stovek transakcí.



Obr. 17. Datový model aplikace.

7.2.3 Transakce a štítky

Třída `Tag` moduluje štítek. `Transaction` představuje transakci, která je vždy navázána na účet a může mít přiřazený hlavní kategorizační štítek a volitelně také neomezený počet

doplňujících štítků (vazbu mezi transakcí a štítkem modeluje třída `TransactionTag`). Vlastnost `TransactionAmount` obsahuje částku, o kterou se změnil zůstatek účtu a `TransactionType` rozlišuje jednotlivé druhy transakcí (běžná transakce, převod mezi účty nebo opakovaná transakce).

`Transfer` ukládá informaci o převodu mezi vlastními účty, kdy propojuje zdrojový s cílovým účtem a také zdrojovou s cílovou transakcí. Díky tomu je možné mít transakce nezávislé a přitom spárované. Při smazání jednoho z účtů se z převodu stane běžná transakce, zároveň ale při změně částky je možné aktualizovat obě dvě spárované transakce současně. Dále lze díky informaci o zdrojovém a cílovém účtu efektivně odfiltrovat převody mezi účty, které má uživatel zobrazené současně. Potom není žádoucí, aby převody viděl jako výdaje a příjmy, protože aplikace vždy se současně zobrazenými účty pracuje jako kdyby se jednalo o jeden virtuální účet.

Opakované transakce jsou uloženy jako `ReccurentTransaction` a jsou navázány na transakci, která slouží jako jejich předpis. Vlastnosti této třídy tedy slouží pouze pro určení, kdy se má provést automatický zápis transakce, ale neduplikují její nastavení.

S opakovanými transakcemi souvisí také třída `ScheduledTask`, kterou bude aplikace interně používat pro spouštění obecných opakovaných úloh. V praxi bude jedno vlákno běžící v aplikační doméně vyhrazeno pouze pro tyto účely a bude periodicky kontrolovat, zda se v tabulce `ScheduledTask` nevyskytuje záznam o úloze, kterou je třeba spustit (čas naplánovaného běhu je ve vlastnosti `ScheduledTaskNextRun`). Pokud takový záznam najde, provede spuštění dané naplánované úlohy pomocí reflexe. Úloha bude specifikovat název assembly a třídy implementující speciální rozhraní, která se bude instancovat. Zodpovědností instancované třídy potom bude například projít všechny opakované transakce a zapsat ty, které jsou naplánované na dnešní den. Výhodou tohoto řešení je především obecnost a jednoduchá rozšiřitelnost o další typy naplánovaných úloh.

7.2.4 Rozpočty

Rozpočet reprezentuje třída `Budget`. Kromě vlastností definujících omezující částku (`BudgetAmount`) a dobu platnosti (`BudgetStartDate` a `BudgetEndDate`) obsahuje i vlastnost `BudgetType`, která určuje zda se jedná o jednorázový nebo opakovaný týdenní, případně měsíční rozpočet. Rozpočet je standardně aplikován na

všechny transakce všech účtů, ale může být omezen jen na vybrané účty a štítky pomocí vazeb modelovaných třídami `BudgetTag` a `BudgetAccount`.

7.2.5 Vozidla

Vozidla jsou modelována třídou `Vehicle`. Každé vozidlo má vazbu na štítek, kterým jsou označovány transakce související s jeho provozem. Tankování vozidla představuje třída `Refuelling`, která je vždy navázána na transakci a rozšiřuje ji o počet natankovaných litrů (vlastnost `RefuellingUnits`), jejich jednotkovou cenu (vlastnost `RefuellingUnitCost`) a v případě dotankování plné nádrže (označeno příznakem `RefuellingFullTank`) také o stav tachometru (`RefuellingTachometer`).

7.3 Zabezpečení aplikace

V této kapitole bude popsáno, jakým konkrétním způsobem se při implementaci aplikace za použití navržených technologií zajistí její odolnost vůči bezpečnostním zranitelnostem uvedeným v kapitole 4 Zabezpečení webových aplikací.

7.3.1 Injection

Datová vrstva bude používat výhradně parametrizované databázové dotazy, které .NET framework nabízí, případně bude použit ORM Entity Framework nebo LINQ to SQL, které zabezpečení proti této zranitelnosti zajistí automaticky. Veškeré uživatelem zadávané hodnoty budou také validovány jak v klientské, tak v serverové části aplikace. K přístupu k databázi bude používán účet, který bude mít pouze nutná oprávnění pro čtení a zápis do potřebných tabulek.

7.3.2 Zranitelná autentizace nebo správa relace

Pro přihlašování budou použity vestavěné mechanismy ASP.NET frameworku, nebude implementován vlastní způsob autentizace. ASP.NET používá pro identifikaci relace cookies, které mají automaticky příznak `HttpOnly`. Jiné způsoby identifikace relace, například pomocí identifikátoru v URL, nebudou povoleny. Relace bude mít krátkou dobu platnosti a nebude umožněno trvalé přihlášení uživatele.

Uživatelské jméno a heslo bude muset splnit požadavky na minimální délku a použití dostatečně velké množiny znaků. Před změnou přihlašovacích údajů se bude muset uživatel znovu přihlásit, aby ověřil svoji identitu (viz UC03: Změna údajů uživatele).

Při obnovení přístupu k uživatelskému účtu bude využit token s omezenou platností zasláný na uživatelskou emailovou adresu (viz scénář UC04: Obnovení uživatelského účtu při zapomenutí hesla), aplikace nebude nikdy zasílat uživatelské jméno nebo heslo na email.

7.3.3 Cross-Site Scripting (XSS)

ASP.NET automaticky validuje všechny HTTP požadavky a odmítne ty, které obsahují nebezpečné znaky. Před vložením dat do stránky se vždy provede enkódování HTML značek. V případě, kdy se budou hodnoty předávat klientským skriptům se ještě navíc budou enkódovat také speciální znaky JavaScriptu, jakými jsou například uvozovky nebo apostrofy ukončující řetězec.

7.3.4 Nezabezpečená přímá reference objektů a kontrola oprávnění

Při přístupu ke každému datovému objektu bude kontrolováno, zda patří přihlášenému uživateli. Pokud se bude uživatel snažit přistoupit k objektu jiného uživatele, bude se aplikace chovat stejně, jako kdyby daný objekt v systému vůbec neexistoval. Aplikace bude interně používat reference pomocí číselných identifikátorů, ale z pohledu uživatele budou objekty referencovány pomocí jejich GUID.

7.3.5 Špatná konfigurace zabezpečení

V produkčním prostředí bude vypnuto ladění a v případě chyby nebudou uživateli zobrazeny žádné technické detaily, ty se pouze zalogují na serveru pro interní použití. Citlivé údaje v konfiguračním souboru aplikace, jako jsou například připojovací řetězce k databázi, budou šifrované. Administraci serveru bude možné provádět pouze pomocí zabezpečených kanálů, nezabezpečené služby jakými je například FTP, nebudou vůbec aktivované.

7.3.6 Únik citlivých dat

Přihlašovací heslo uživatele bude v databázi uloženo jako hash se solí, která bude unikátní pro každého uživatele. Pro hashování bude použit algoritmus SHA-2 s délkou 512 bitů. Veškerá data budou přenášena mezi serverem a klientem šifrovaným kanálem HTTPS. Při přístupu přes protokol HTTP bude uživatel automaticky přesměrován na zabezpečenou verzi stránek. Doména aplikace bude zabezpečena pomocí DNSSEC.

7.3.7 Cross-Site Request Forgery (CSRF)

Všechna data budou na server odesílána pomocí POST (nebo PUT) požadavku, nikoliv GET požadavkem. Každý formulář vygeneruje unikátní token, který bude validován při odeslání formuláře na server. Tuto funkcionalitu poskytuje přímo ASP.NET MVC framework, tudíž ji není třeba implementovat, ale pouze správně použít.

7.3.8 Komponenty se známými zranitelnostmi

Aplikace bude hostována na profesionálním hostingu nebo v cloudu Microsoft Azure, který garantuje určitou míru zabezpečení a automatickou aplikaci dostupných aktualizací. Budou použity nejnovější verze všech frameworků a knihoven, nejlépe pomocí nuget balíčků, které zajistí jednoduché přechody na budoucí novější verze.

7.3.9 Neověřené přesměrování

Aplikace nebude nikdy přesměřovávat na adresu přijatou nějakou formou v požadavku. Veškeré případné přesměrování bude probíhat pouze na předem definované adresy přímo ve zdrojovém kódu aplikace.

8 PROTOTYP APLIKACE

V rámci diplomové práce byl vytvořen funkční prototyp aplikace určený k provedení uživatelského testování navrhnutého řešení. Toto testování bylo zaměřeno na ověření pochopitelnosti koncepce aplikace a dobré uživatelské použitelnosti základních případů užití týkajících se evidence příjmů a výdajů.

Doporučení pro vedení uživatelských testování jsou například v knize Webdesign Nenutíte uživatele přemýšlet [2]. Testování proběhlo na čtyřech uživateli, kteří se k němu dobrovolně přihlásili. Každý z nich měl za úkol v aplikaci projít vytisknutý testovací scénář a přitom nahlas vyslovovat vše, co ho v danou chvíli napadne – většinou to znamená odpovědi na otázky typu: „Co vidím na obrazovce a jaký to má význam?“, „Co zde mohu udělat?“, „Proč klikám zrovna sem?“, „Jak na mě aplikace působí?“ apod.

Dále budou popsány jednotlivé kroky testovacího scénáře a každý z nich bude vyhodnocen vzhledem ke zpětné vazbě od uživatelů.

8.1 Registrace uživatele a přihlášení

Uživatelé byli vyzváni k vytvoření nového uživatelského účtu a přihlášení do aplikace. K přihlášení slouží obrazovka zobrazená na Obr. 18. Žádný z uživatelů neměl problém se splněním tohoto úkolu, všichni okamžitě zvolili možnost vytvoření nového účtu a vyplnili registrační formulář (Obr. 19). Po jeho odeslání je aplikace automaticky přihlásí.



OsobníFinance.eu
Osobní finance pod kontrolou.

Přihlašte se prosím

Uživatelské jméno

Heslo

Přihlásit se

[Vytvořte si účet](#) pokud žádný ještě nemáte.

Obr. 18. Přihlašovací obrazovka.



The image shows a registration form for the website OsobníFinance.eu. The header includes the site name and the tagline 'Osobní finance pod kontrolou.' Below this, the title 'Vytvoření nového účtu' is displayed. The form consists of three input fields: 'Uživatelské jméno', 'Heslo', and 'Potvrzení hesla'. A green button labeled 'Vytvořit účet' is positioned at the bottom of the form.

Obr. 19. Registrační formulář.

8.2 Vytvoření účtů

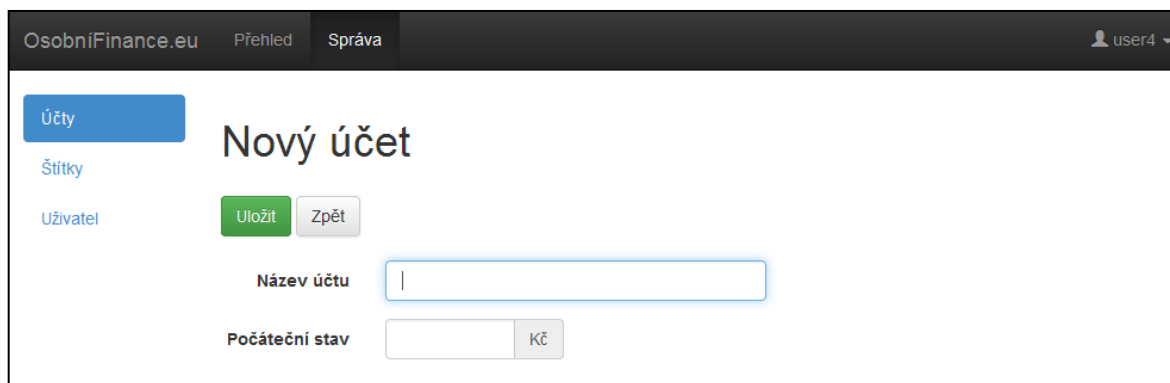
Uživatelé byly vyzváni, aby v aplikaci evidovali 7 výdajů a jeden příjem, přičemž některé výdaje platili v hotovosti a jiné bezhotovostně z bankovního účtu. Seznam těchto pohybů financí dostali ve formě tabulky. Bylo na rozhodnutí jednotlivých uživatelů, zda budou chtít evidovat hotovost a bankovní účet zvlášť nebo dohromady. Všichni zvolili možnost oddělených účtů.



The image shows the dashboard for the 'Účty' (Accounts) section in the OsobníFinance.eu application. The top navigation bar includes the site name, 'Přehled', 'Správa', and a user profile icon labeled 'user4'. On the left, there is a sidebar with 'Účty' selected, along with 'Štítky' and 'Uživatel'. The main content area features a green 'Nový účet' button and a message: 'Zatím nemáte vytvořené žádné účty. Účty slouží k oddělenému sledování financí. Můžete mít například pouze jeden účet představující všechny Vaše finanční prostředky nebo můžete chtít sledovat pohyb peněz na běžném účtu, spořicího účtu a v hotovosti zvlášť.' Below the message is a blue button labeled 'Začněte vytvořením nového účtu'.

Obr. 20. Úvodní obrazovka po prvním přihlášení.

Po přihlášení do aplikace je uživatel v případě, kdy nemá vytvořený žádný účet, přesměrován do správy účtů, aby si mohl účet rovnou vytvořit (Obr. 20). Na této obrazovce uživatelé neměli problémy a kliknuli buď na tlačítko *Nový účet* nebo *Začněte vytvořením nového účtu*. Tím se dostali na obrazovku pro vytvoření nového účtu (Obr. 21). Na této obrazovce nebyly zaznamenány žádné nejasnosti.



Obr. 21. Vytvoření nového účtu.

Po uložení svého prvního účtu je uživatel automaticky přesměrován do sekce *Přehled*, aby mohl hned začít zadávat transakce. Vzhledem k tomu, že všichni uživatelé chtěli pro evidenci vytvořit více účtů, přišlo jim toto přesměrování nevhodné, protože čekali, že budou moci vytvořit další účet. Museli se proto vrátit zpět do sekce *Správa*, k čemuž museli použít menu v horním panelu. Většinou se zpět dostali rychle a bez potíží, ale v jednom případě uživatel strávil necelou půl minutu hledáním možnosti jak se dostat zpět do správy účtů, než si všiml horního menu. Toto menu by tedy mohlo být více výrazné.

8.3 Evidence transakcí

Pro evidenci transakcí je nutné jít na stránku *Přehled*, což zvládli všichni uživatelé, protože již věděli, že k tomu mají použít horní menu. Na této stránce je možné zadávat nové a filtrovat již existující transakce a zobrazovat si k nim statistiky, jak bylo navrženo v kapitole 6.1 *Přehled a evidence transakcí*. Konečnou podobu této stránky je možné vidět na Obr. 22. Formulář pro vytvoření nového výdaje je na Obr. 23.

S vytvořením transakcí neměl žádný uživatel vážné potíže, nicméně našlo se několik připomínek. Především by podle uživatelů bylo dobré podrobněji vysvětlit funkci štítků, například nápovědou přímo u pole *Štítky* formuláře pro zadávání transakcí. Dále uživatelům na začátku nebylo jasné, zda musí ve filtru transakcí vybrat nejdříve účet a až

poté zvolit možnost *Nový výdaj* (účet se ve skutečnosti vybírá až ve formuláři pro vytvoření transakce a filtr v přehledu na něj nemá žádný vliv).

Výdaje

Datum	Štítky	Popis	Částka	Akce
30. 04. 2014	ZÁBAVA	Hobbit	230 Kč	[edit] [delete]
29. 04. 2014	AUTO Pojištění	Povinné ručení	3 800 Kč	[edit] [delete]
26. 04. 2014	POJIŠTĚNÍ		620 Kč	[edit] [delete]
25. 04. 2014	DOMÁCNOST Elektřina Inkaso		1 200 Kč	[edit] [delete]

Příjmy

Datum	Štítky	Popis	Částka	Akce
20. 04. 2014			18 000 Kč	[edit] [delete]
Celkem			18 000 Kč	

Obr. 22. Přehled evidovaných transakcí.

Nový výdaj

Uložit Zpět

Účet: Bankovní účet Hotovost

Částka: 1200| Kč

Štítky: × Domácnost × Elektřina

Datum: 04. 05. 2014 Dnes Včera

Popis: [text input field]

Obr. 23. Formulář pro vytvoření nového výdaje.

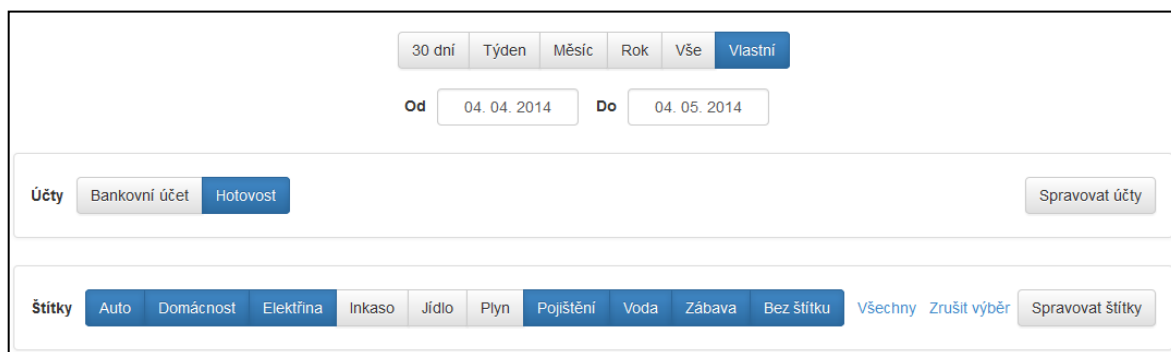
Jeden uživatel si myslel, že když má ve filtru vybrané dva účty, tak výdaj vytvoří pro oba účty současně. Nicméně poté, co uživatelé zjistili, že je nutné vybrat účet explicitně až při vytváření výdaje, jim byl systém vytváření transakcí jasný. Při vytváření transakce by ale ocenili, kdyby jim aplikace automaticky vybrala účet, který měli zvolený ve filtru přehledu transakcí. Jednomu uživateli chybělo pole *Název*, které by využíval pro jednoduchou orientaci v transakcích, protože *Popis* chápal jako doplňující nedůležité údaje, které nebudou zobrazeny přímo v tabulce transakcí. Naopak uživatelé uvedli, že se jim líbí možnost rychlé volby dnešního a včerejšího data pomocí tlačítek.

Po vyzvání k úpravě a smazání existující transakce uživatelé okamžitě klikli na správná tlačítka na řádku s transakcí, kterou chtěli upravit nebo smazat. Stejně tak přejmenování a odstranění existujícího štítku pro ně nepředstavovalo žádný problém.

8.4 Zobrazení statistik

Uživatelé k zobrazení statistik používali testovací účet s již připravenými daty, aby měli všichni stejné výchozí podmínky. Byly vyzvány k rozboru evidovaných výdajů za určité období. Zde vyvstalo několik nejasností, které budou popsány dále.

8.4.1 Filtrování transakcí



The screenshot displays a user interface for filtering transactions. At the top, there are buttons for time periods: '30 dní', 'Týden', 'Měsíc', 'Rok', 'Vše', and 'Vlastní'. Below these are input fields for 'Od' (04. 04. 2014) and 'Do' (04. 05. 2014). Underneath, there are buttons for account types: 'Účty', 'Bankovní účet', and 'Hotovost', along with a 'Spravovat účty' button. At the bottom, there are buttons for tags: 'Štítky', 'Auto', 'Domácnost', 'Elektřina', 'Inkaso', 'Jídlo', 'Plyn', 'Pojštění', 'Voda', 'Zábava', 'Bez štítku', 'Všechny', 'Zrušit výběr', and 'Spravovat štítky'.

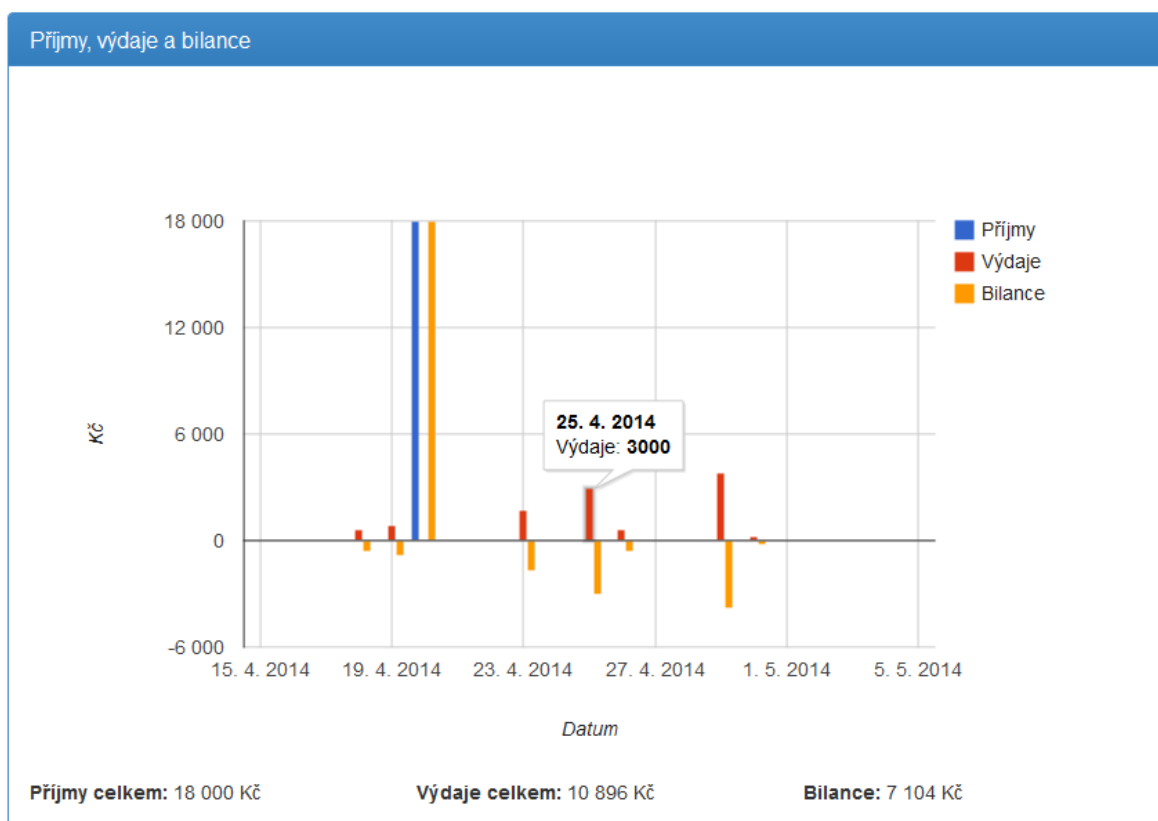
Obr. 24. Filtr transakcí.

Veškeré statistiky se počítají pro transakce, které jsou vybrané na základě filtru (Obr. 24). V tomto filtru je možné zvolit časové období, účty a štítky, které uživatele zajímají. Tohle není na první pohled zřejmě dostatečně jasné, protože uživatelé nebyli při pohledu na jednotlivé statistiky schopni říci, pro jaké období jsou statistiky platné. Pomohlo by uvádět u každé statistiky explicitně období, na jaké se vztahuje, protože filtr je na začátku stránky a statistiky až na jejím konci. Případně by bylo možné udělat časový filtr fixní, potom by zůstával stále přichycený k horní části obrazovky a uživatelé jej měli na očích.

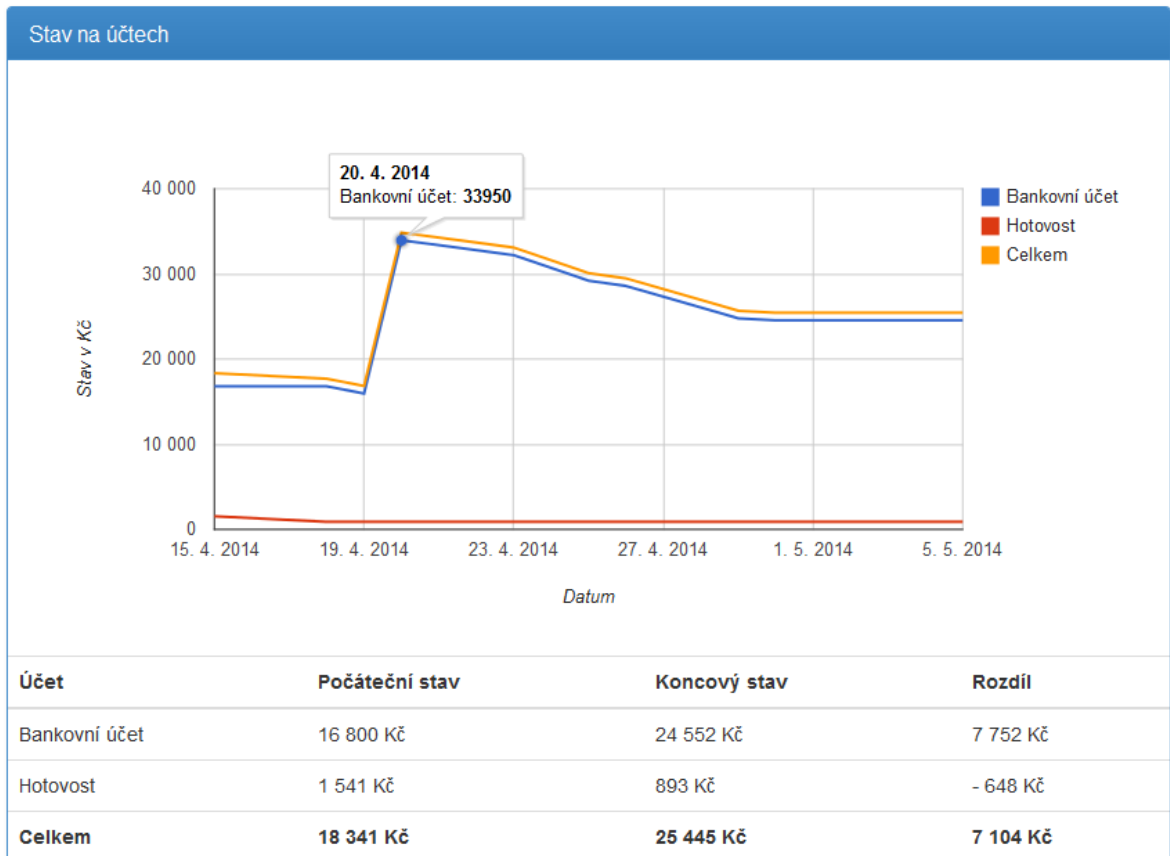
Dalším problémem se ukázala aktuální implementace filtrování štítků a účtů, kdy modře zvýrazněné štítky nebo účty jsou zahrnuty do statistik a bílé nikoliv. Tohle bylo uživatelům jasné. V případě, kdy ale uživatel zruší výběr úplně všech štítků (případně účtů), se filtr chová stejně, jako kdyby byly vybrány všechny, což uživatele mátló. Jasnější jim přijde varianta, kdy takové nastavení filtru nikdy nevrátí žádnou transakci. Uživatelé ale ocenili možnost rychle vybrat všechny štítky nebo zrušit výběr všech štítků pomocí akčních tlačítek, což jim usnadnilo například vybírání pouze jednoho štítku.

8.4.2 Grafy

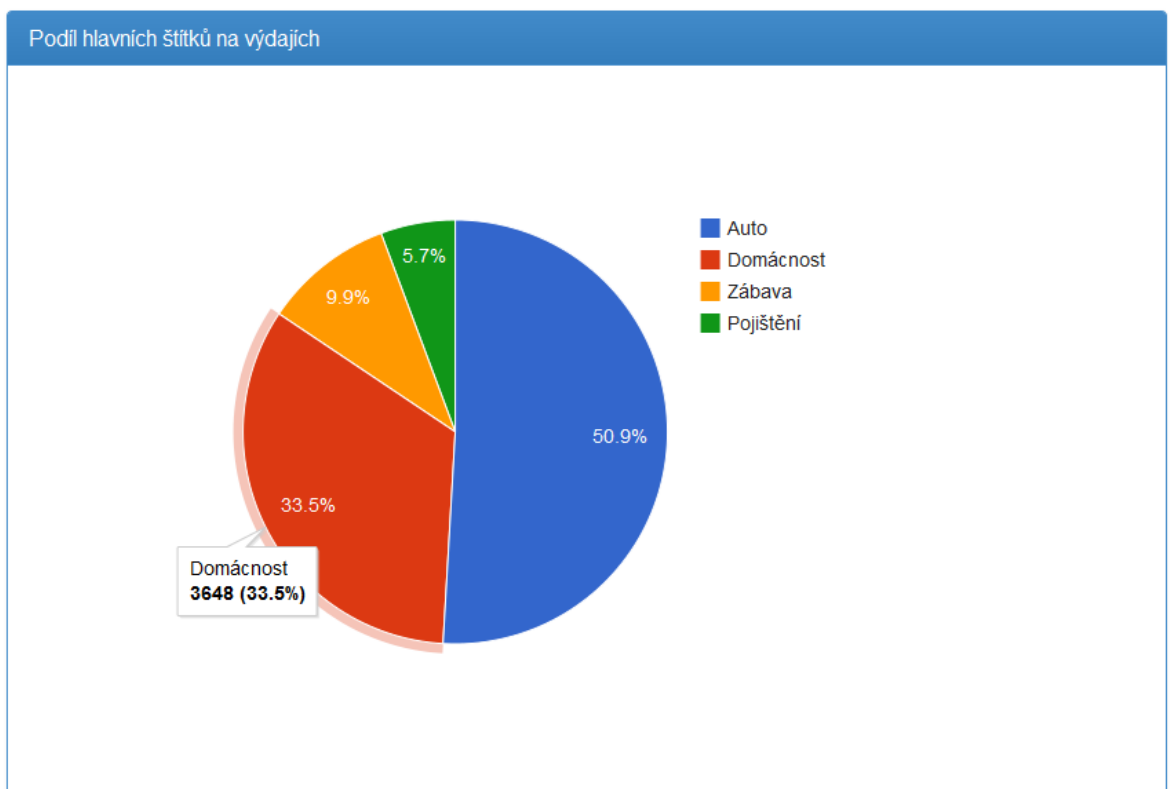
Aplikace dokáže z transakcí vybraných na základě filtru vytvořit několik grafů. Graf příjmů, výdajů a bilance (Obr. 25) byl uživatelům jasný, pouze v jednom případě uživatel nerozuměl pojmu „bilance“ a ocenil by spíše použití slova „rozdíl“. Graf vývoje stavu účtu (Obr. 26) byl pochopitelný pro všechny uživatele.



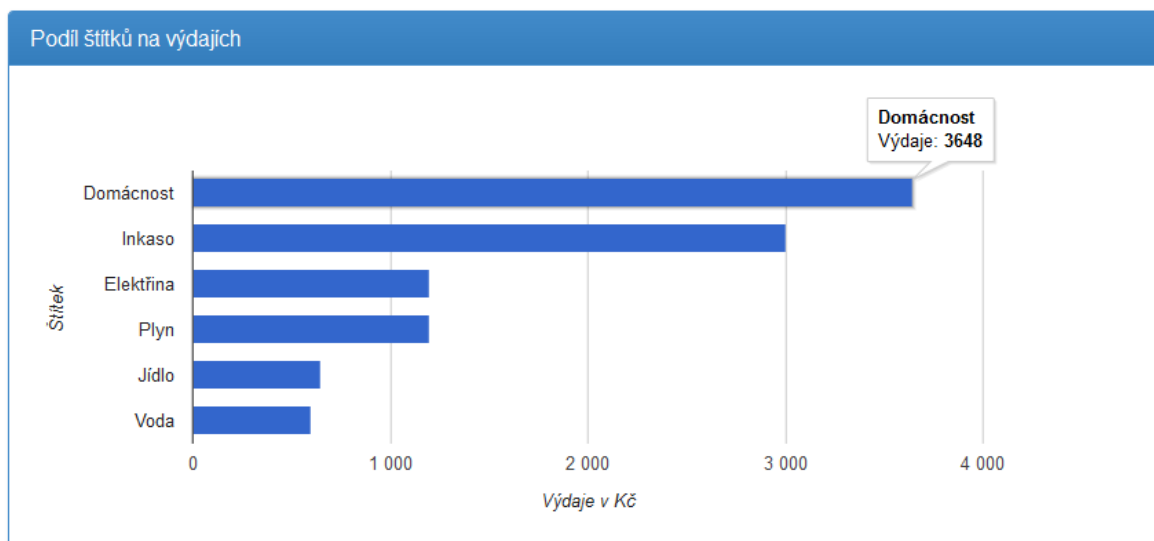
Obr. 25. Graf příjmů, výdajů a bilance s aktivní anotací.



Obr. 26. Graf vývoje stavu účtů s aktivní anotací.



Obr. 27. Graf podílu hlavních štitků na výdajích s aktivní anotací.



Obr. 28. Graf podílu jednotlivých štítků na výdajích s aktivní anotací.

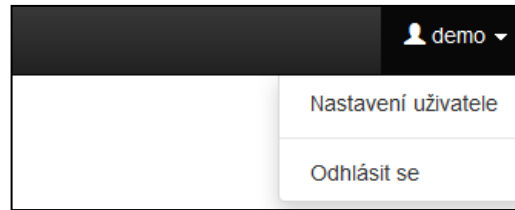
Graf podílu hlavních štítků na výdajích (Obr. 27) uživatelé označili za nejzajímavější, protože z něj snadno zjistí strukturu svých výdajů. Padl námět na zlepšení, aby po najetí na část koláčového grafu aplikace zobrazila všechny transakce, které byly do dané části grafu započítány.

Nejasnosti pro některé uživatele přinesl graf podílu štítků na výdajích (Obr. 28), protože jednotlivé transakce jsou započítány ve více sloupcích grafu. Například transakce označené štítky *Voda*, *Plyn* a *Elektřina* jsou současně označeny štítkem *Inkaso* a *Domácnost*, takže jsou započteny také v těchto dvou sloupcích. Kromě toho je ve výdajích za *Domácnost* započítáno také *Jídlo*, které má ale přitom v grafu svůj vlastní sloupec. Toto chování vychází z podstaty zvolení nehierarchické struktury kategorizace výdajů, na což tito uživatelé nejsou zvyklí a je pro ně náročnější tento graf pochopit. Řešení by mohlo spočívat v tom, že po najetí na sloupec by byl zobrazen seznam transakcí, které jsou do něj zahrnuty, případně by si uživatel ke každé transakci přiřadil pouze jeden štítek a potom by sloupce tohoto grafu byly disjunktní.

Uživatelé intuitivně přišli na možnost anotovat zobrazená data po najetí kurzorem myši na bod grafu, který je zajímavý a ocenili ji.

8.5 Změna hesla a odhlášení

Nakonec byli uživatelé vyzváni ke změně hesla a odhlášení z aplikace. Obě dvě tyto možnosti jsou dostupné z vysouvacího uživatelského menu, které je umístěno v pravém horním rohu obrazovky (Obr. 29) a všichni testovaní uživatelé je intuitivně použili.



Obr. 29. Uživatelské menu.

8.6 Celkový dojem uživatelů

Celkový dojem uživatelů je veskrze pozitivní. Aplikaci označili za intuitivní a dokázali ji používat bez předchozích znalostí. Velmi oceňovali čistý a jednoduchý vzhled aplikace. Během seznamování se s aplikací vyvstaly sice některé nejasnosti, ale na všechny dokázali uživatelé najít odpovědi sami tím, že aplikaci chvíli používali. Výjimkou je snad jen systém nehierarchických štítků, který by si zasloužil lepší vysvětlení přímo v aplikaci. Uživatelé shodně uvedli, že by aplikaci používali již v takovém stavu, v jakém je nyní, protože nezaznamenali žádnou zásadní překážku, která by jim v použití aplikace bránila.

9 DALŠÍ ROZVOJ APLIKACE

Aplikace by v budoucnu mohla být rozšířena o další funkcionalitu, která s evidencí osobních financí souvisí. Příkladem může být evidence spotřeby energií v domácnosti, ze které by bylo možné dělat zajímavé statistiky.

Správa vozidel by mohla být rozšířena o možnost vytváření vlastních servisních plánů jednotlivých vozidel, kde by si uživatel mohl sám určit akce, které je třeba provádět po uplynutí určité doby nebo ujetí daného počtu kilometrů. Aplikace by potom uživatele s předstihem upozorňovala na blížící se servis vozidla, jaké úkony bude nutné provést vzhledem k ujetým kilometrům a zobrazila by předpokládané výdaje vycházející z historických dat o předešlých servisech.

Důležitou funkcí, kterou zmínili také uživatelé při testování prototypu, by byl import transakcí z výpisů účtu českých bank nebo ještě lépe automatický import za využití API nabízeného některými z nich. Případně by k automatizaci také mohlo být využito informačních emailů, které banky zasílají při změně stavu účtu.

Dále by aplikace mohla podporovat vedení účtů ve více měnách současně nebo hromadné akce, pomocí kterých by uživatel mohl například přiřadit štítek více transakcím najednou.

Přestože je aplikace od začátku navrhována s ohledem na mobilní zařízení, je stále nutné přistupovat k ní online. V budoucnu by bylo určitě dobré doplnit ji o velmi jednoduchou mobilní aplikaci, která by umožňovala offline zadávání výdajů a ty poté automaticky synchronizovala se serverem.

ZÁVĚR

Tato diplomová práce se zabývá aplikacemi pro správu osobních financí. Nejdříve je nastíněna motivace, proč se má smysl touto problematikou vůbec zabývat a je uvedeno, jaké mají uživatelé na tento druh aplikací požadavky. Dále jsou popsány typy aplikací, které pro správu osobních financí existují a výsledky analýzy relevantních webových řešení dostupných na českém trhu a jednoho zahraničního startupu.

Vzhledem k tomu, že osobní finance spravované ve webové aplikaci jsou velmi citlivé téma z hlediska bezpečnosti, bylo detailně rozebráno deset největších bezpečnostních zranitelností webových aplikací podle nezávislé organizace OWASP. U každé zranitelnosti bylo vysvětleno v čem spočívá, jak jí lze zneužít a jakým způsobem se proti ní bránit. Díky tomuto rozboru bylo možné později vlastní aplikaci navrhnout tak, aby zmíněné zranitelnosti neobsahovala.

Bylo navrženo vlastní řešení problematiky, jehož funkcionalita byla detailně rozpracována ve formě jednotlivých uživatelských scénářů. Na základě vytvořených scénářů a nefunkčních požadavků na aplikaci bylo navrženo uživatelské rozhraní aplikace ve formě náčrtů. Na těchto náčrtech bylo validováno, zda obsahují veškerou požadovanou funkcionalitu aplikace.

Pro implementaci byla vybrána vhodná kombinace webových serverových a klientských technologií a dostupných frameworků. Byl také vytvořen funkční prototyp aplikace, který implementuje nejdůležitější uživatelské scénáře a ověřuje navržený výběr technologií.

Vytvořený prototyp byl využit pro uživatelské testování aplikace na vzorku uživatelů, které mělo ověřit správnost celého konceptu navrženého řešení. Při testování byly objeveny některé nedostatky implementace, které mohou být opraveny do další verze. Samotný koncept aplikace byl ale validován s pozitivním výsledkem. Všichni uživatelé shodně uvedli, že by dané řešení používali pro evidenci svých financí.

Nakonec byly v práci uvedeny náměty na další rozšíření aplikace, které částečně také vychází ze zpětné vazby od uživatelů účastnících se testování prototypu aplikace.

CONCLUSION

This dissertation describes applications for personal finance management. The motivation for dealing with this topic is described at the beginning and also outlines users' requirements for this type of applications. The next part details already existing applications for personal finance management and analysis results of relevant web solutions available on the Czech market and one foreign startup.

Due to the fact that personal finance managed in a web application is a very sensitive topic in terms of security, this work describes in detail the ten largest security vulnerabilities of web applications according to the independent OWASP organization. Each vulnerability is thoroughly explained, how it can be misused and how to fight it. It was later possible to design the application avoiding these vulnerabilities thanks to this in depth analysis.

The solution of the problem was suggested and its functionality developed in detail in individual use case scenarios. The user interface was designed in sketches based on these scenarios and non-functional requirements for the application. These sketches validate all of the required functionality of the application.

A combination of web server and client technologies and frameworks were used to implement this application. The created functional prototype of the application implements the most important use case scenarios and checks the suggested technology selection.

Created prototype was used for user testing on a sample of users. Such testing was planned to check the correctness of the entire solution concept. A few drawbacks were found during the testing, which can be fixed before the next version. The application concept itself was validated with a positive result. All users agreed that they would use this application for their personal finances management.

More thoughts on further expansion of the application are described in the last section. These thoughts are partially based on the feedback from users participating in the prototype testing.

SEZNAM POUŽITÉ LITERATURY

- [1] STEM/MARK, A.S.. *Finanční gramotnost v ČR: Kvantitativní výzkum – Finanční gramotnost obyvatel ČR* [online]. 2010, 88 s [cit. 2014-03-09]. Dostupné z: http://www.mfcr.cz/assets/cs/media/Odborne-vyzkumy_2010-12_Zaverecna-zprava-z-vyzkumu-plne-zneni-STEMMARK.pdf
- [2] KRUG, Steve. *Web design: nenuťte uživatele přemýšlet!*. 2. aktualiz. vyd. Brno: Computer Press, 2006, 167 s. ISBN 80-251-1291-8.
- [3] ŠPINAR, David. *Tvoříme přístupné webové stránky: připraveno s ohledem na novelu Zákona č. 365/2000 Sb., o informačních systémech veřejné správy*. Vyd. 1. Brno: Zoner Press, 2004, 360 s. ISBN 80-868-1511-0.
- [4] SVETLÍK, Slavomír. *RQ Money* [online]. 2014 [cit. 2014-02-22]. Dostupné z: <http://www.rq.sk/>
- [5] Google. My Budget Book. *Google Play* [online]. 2014 [cit. 2014-02-22]. Dostupné z: <https://market.android.com/details?id=com.onetwoapps.mh>
- [6] LAŠ, Jan. *eÚčty.cz* [online]. 2014 [cit. 2014-03-15]. Dostupné z: <http://eucty.cz>
- [7] Fresh Garden s.r.o.. *Moře financí* [online]. 2014 [cit. 2014-03-15]. Dostupné z: <https://www.more-financi.cz/>
- [8] Toshl Inc.. *Toshl Finance: Personal Finance Manager and Expense Tracker* [online]. 2014 [cit. 2014-03-22]. Dostupné z: <https://toshl.com/>
- [9] OWASP Foundation Inc.. *OWASP* [online]. 2014 [cit. 2014-03-23]. Dostupné z: <http://www.owasp.org>
- [10] OWASP FOUNDATION INC.. *OWASP Top 10 - 2013: The Ten Most Critical Web Application Security Risks* [online]. 2013 [cit. 2013-11-11]. Dostupné z: <http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf>
- [11] SQL Injection. *OWASP* [online]. 2013 [cit. 2013-11-14]. Dostupné z: https://www.owasp.org/index.php/SQL_Injection
- [12] Session hijacking attack. *OWASP* [online]. 2011 [cit. 2013-11-14]. Dostupné z: https://www.owasp.org/index.php/Session_hijacking_attack
- [13] Man-in-the-middle attack. *OWASP* [online]. 2009 [cit. 2014-03-23]. Dostupné z: https://www.owasp.org/index.php/Man-in-the-middle_attack
- [14] Cross-site Scripting (XSS). *OWASP* [online]. 2013 [cit. 2013-11-14]. Dostupné z: https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29
- [15] Cross-Site Request Forgery (CSRF). *OWASP* [online]. 2013 [cit. 2013-11-14]. Dostupné z: <https://www.owasp.org/index.php/CSRF>

[16] Evolus. *Pencil Project* [online]. 2012 [cit. 2014-04-10]. Dostupné z: <http://pencil.evolus.vn/>

[17] NIELSEN, Jakob a Marie TAHIR. *Použitelnost domovských stránek*. Vyd. 1. Brno: Zoner Press, 2005, 323 s. ISBN 80-868-1518-8.

[18] *Bootstrap* [online]. 2014 [cit. 2014-04-10]. Dostupné z: <http://getbootstrap.com/>

[19] MACDONALD, Matthew, Adam FREEMAN a Mario SZPUSZTA. *ASP.NET 4 a C# 2010: tvorba dynamických stránek profesionálně*. Vyd. 1. Brno: Zoner Press, 2011, 880 s. ISBN 978-80-7413-131-8.

[20] SKEET, Jon. *C# in depth: Second Edition*. 2. Stamford, CT: Manning, 2011, 554 s. ISBN 19-351-8247-1.

[21] *MSDN – Microsoft Developer Network* [online]. Microsoft, 2014 [cit. 2014-02-02]. Dostupné z: <http://msdn.microsoft.com>

[22] LACKO, Luboslav. *SQL hotová řešení: k okamžitému použití*. Vyd. 1. Brno: Computer Press, 2003, 298 s. ISBN 80-722-6975-5.

[23] W3C. *HTML5* [online]. 2013 [cit. 2014-04-22]. Dostupné z: <http://www.w3.org/TR/html5/>

[24] CROFT, Jeff, Ian LLOYD a Dan RUBIN. *Mistrovství v CSS: pokročilé techniky pro webové designéry a vývojáře*. Vyd. 1. Brno: Computer Press, 2007, 409 s. ISBN 978-80-251-1705-7.

[25] DEVERIA, Alexis. *Can I use...* [online]. 2014 [cit. 2014-04-22]. Dostupné z: <http://caniuse.com/>

[26] The jQuery Foundation. *jQuery* [online]. 2014 [cit. 2014-04-22]. Dostupné z: <http://jquery.com/>

[27] Google. *AngularJS: Superheroic JavaScript MVW Framework* [online]. 2014 [cit. 2014-04-22]. Dostupné z: <https://angularjs.org/>

[28] Google. Google Charts. *Google Developers* [online]. 2014 [cit. 2014-04-22]. Dostupné z: <https://developers.google.com/chart/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Application Programming Interface – programové rozhraní systému určené pro integraci s jinými systémy.
CAPTCHA	Turingův test používaný na internetu pro odlišení lidí od robotů.
CSS	Cascading Style Sheets – kaskádové styly definující vzhled webu.
CSV	Comma-separated values – formát souboru, kde se hodnoty oddělují čárkou.
DOM	Document Object Model – Objektově orientovaná reprezentace XML nebo HTML dokumentu.
GUID	Globally Unique Identifier – identifikátor, který na základě pravděpodobnosti zaručuje praktickou unikátnost v celosvětovém měřítku.
JSON	JavaScript Object Notation – platformě nezávislý datový formát.
LDAP	Protokol pro ukládání a přístup k datům na adresářovém serveru.
NoSQL	Obecné označení pro nerelační databáze.
ORM	Object-relational mapping – technika, která zajišťuje automatickou synchronizaci relační databáze s objektovým modelem aplikace.
OWASP	Open Web Application Security Project – neziskové sdružení zabývající se bezpečností internetových aplikací.
SaaS	Software as a Service – Software nabízený formou předplacené služby.
SMTP	Protokol určený pro přenos zpráv elektronické pošty.
TSL/SSL	Kryptografické protokoly používané pro zabezpečení komunikace na internetu.
UC	Use Case – případ užití.
URL	Uniform Resource Locator – řetězec určující umístění souboru nebo služby na internetu.
UX	User Experience – uživatelská spokojenost nebo použitelnost.
XML	Extensible Markup Language – obecný značkovací jazyk.

SEZNAM OBRÁZKŮ

Obr. 1. Graf zobrazující kolik osob v ČR má přehled o svých financích. [1]	11
Obr. 2. Graf zobrazující zastoupení jednotlivých způsobů sledování financí. [1].....	12
Obr. 3. Ukázka programu RQ Money. [4].....	16
Obr. 4. Ukázka mobilní aplikace My Budget Book. [5]	17
Obr. 5. Ukázka přehledu transakcí v aplikaci eÚčty.cz. [6].....	19
Obr. 6. Ukázka přehledu transakcí v aplikaci Moře financí. [7].....	21
Obr. 7. Ukázka přehledu transakcí v aplikaci Toshl Finance. [8].....	23
Obr. 8. Diagram aktérů.	40
Obr. 9. Schéma propojení jednotlivých stránek uživatelského rozhraní.	75
Obr. 10. Náčrt formuláře pro vytvoření nového výdaje.	76
Obr. 11. Náčrt stránky Přehled zobrazené na monitoru s vysokým rozlišením.....	77
Obr. 12. Náčrt stránky Přehled zobrazené na tabletu.	78
Obr. 13. Náčrt stránky Přehled zobrazené na mobilním telefonu.	78
Obr. 14. Náčrt seznamu účtů v sekci Správa.	79
Obr. 15. Náčrt formuláře pro úpravu účtu v sekci Správa.....	79
Obr. 16. Náčrt stránky Vozidla sloužící pro zobrazení statistik o evidovaném vozidle.	80
Obr. 17. Datový model aplikace.	86
Obr. 18. Přihlašovací obrazovka.	91
Obr. 19. Registrační formulář.	92
Obr. 20. Úvodní obrazovka po prvním přihlášení.....	92
Obr. 21. Vytvoření nového účtu.....	93
Obr. 22. Přehled evidovaných transakcí.	94
Obr. 23. Formulář pro vytvoření nového výdaje.	94
Obr. 24. Filtr transakcí.....	95
Obr. 25. Graf příjmů, výdajů a bilance s aktivní anotací.	96
Obr. 26. Graf vývoje stavu účtů s aktivní anotací.....	97
Obr. 27. Graf podílu hlavních štítků na výdajích s aktivní anotací.....	97
Obr. 28. Graf podílu jednotlivých štítků na výdajích s aktivní anotací.....	98
Obr. 29. Uživatelské menu.	99

SEZNAM PŘÍLOH

- Příloha I: Diagram případů užití
- Příloha II: DVD (obsahuje text práce, zdrojové soubory vytvořeného prototypu aplikace a instalační soubory software třetích stran, který je potřeba pro spuštění prototypu)

PŘÍLOHA I: DIAGRAM PŘÍPADŮ UŽITÍ

