

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Návrh systému pro správu financí neziskové organizace

Bc. Čermák Pavel

© 2021 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Pavel Čermák

Systémové inženýrství a informatika
Informatika

Název práce

Návrh systému pro správu financí neziskové organizace

Název anglicky

Design of financial management system for non-profit organization

Cíle práce

Cílem teoretické části práce je uvedení do problematiky správy financí s důrazem na správu financí neziskové organizace a analýza současného stavu existujících procesů a systému pro správu financí organizace.

Cílem praktické části práce je vytvoření doporučení na úpravu procesů organizace směřující k zrychlení a zjednodušení správy financí a tvorba zadávací dokumentace systému, který tyto procesy podpoří.

Metodika

Bakalářská práce pojednává o správě financí neziskové organizace a s tím spojeným informačním systémem. V první řadě tedy nastudují zákonné povinnosti spojené se správou financí neziskových organizací, následně pak nastudují jazyk UML pro modelování diagramů a metodiku BPMN (Business Process Model and Notation) používanou pro modelování procesů. Po prostudování teoretických základů zanalyzují aktuální procesy správy financí v organizaci, následně navrhnu zlepšení těchto procesů a vytvořím zadávací dokumentaci pro systém na základě nově navržených procesů.

Bakalářská práce je rozdělena do 2 částí. V první části se budu věnovat teorii, tedy uvedení do problematiky správy financí neziskových organizací, návrhu informačních systémů a samotné analýze stávajících procesů v organizaci. V rámci druhé části navrhnu změnu procesů za účelem zefektivnění a zjednodušení správy financí a vytvořím zadávací dokumentaci pro systém podporující navržené změny.

Doporučený rozsah práce

30-50 stran

Klíčová slova

informační systém, finanční systém, nezisková organizace, správa financí, analýza finančních procesů

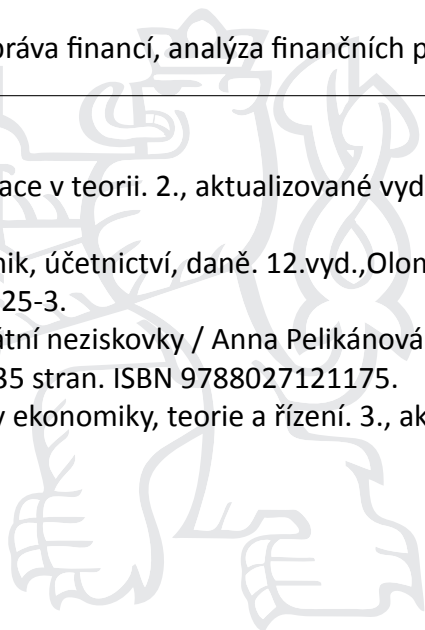
Doporučené zdroje informací

DOBROZEMSKÝ, Václav a Jan STEJSKAL. Nevýdělečné organizace v teorii. 2., aktualizované vydání. Praha: Wolters Kluwer, 2016. ISBN 978-80-7552-103-3.

MERLÍČKOVÁ RŮŽIČKOVÁ, Růžena. Neziskové organizace: vznik, účetnictví, daně. 12.vyd., Olomouc: Nakladatelství ANAG, 2013, 263 s., ISBN 978-80-7263-825-3.

PELIKÁNOVÁ, Anna. Účetnictví, daně a financování pro nestátní neziskovky / Anna Pelikánová. 3., aktualizované vydání. Praha: Grada Publishing, 2018. 335 stran. ISBN 9788027121175.

REKTOŘÍK, Jaroslav. Organizace neziskového sektoru: základy ekonomiky, teorie a řízení. 3., aktualiz. vyd. Praha: Ekopress, 2010. ISBN 978-80-86929-54-5.



Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Dana Vyníkarová, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 20. 02. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Návrh systému pro správu financí neziskové organizace" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne datum odevzdání

Poděkování

Rád bych touto cestou poděkoval Ing. Daně Vynikarové, Ph.D. za odborné vedení, trpělivost a cenné připomínky při tvorbě této práce. Velké díky také patří mé rodině za morální a psychickou podporu po celou dobu studia.

Návrh systému pro správu financí neziskové organizace

Abstrakt

Tato práce se zabývá problematikou správy financí neziskové organizace XYZ. Cílem bylo provést analýzu aktuálního řešení správy financí včetně používaných systémů. Na základě analýzy byl navržen nový systém, který bude sloužit jako centrální místo pro kompletní správu financí dané organizace. Navržený systém odbourává nutnost přistupovat do více systémů pro kompletní správu financí a snižuje pravděpodobnost výskytu chyby.

Klíčová slova: informační systém, finanční systém, nezisková organizace, správa financí, analýza finančních procesů

Design of a financial management system for a non-profit organization

Abstract

This work deals with the issue of financial management of the non-profit organization XYZ. The aim was to analyze the current financial management solution, including the systems used. Based on the analysis, a new system was designed, which will serve as a central place for the organization's complete financial management. The proposed system eliminates the need to access multiple systems for full financial management and reduces the likelihood of error.

Keywords: information system, financial system, non - profit organization, financial management, analysis of economic processes

Obsah

1 Úvod.....	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Postavení neziskové organizace v národním hospodářství	13
3.2 Technologie.....	14
3.2.1 Architektura	14
3.2.1.1 Virtuální server	14
3.2.1.2 Docker	14
3.2.1.3 Porovnání Dockeru a virtuálního serveru.....	14
3.2.1.4 Shrnutí	15
3.2.2 Technologie	15
3.2.2.1 PHP.....	15
3.2.2.2 Javascript	16
3.2.2.3 Srovnání PHP a Node.js	16
3.2.2.4 Shrnutí	16
3.2.3 Databáze.....	16
3.2.3.1 Výkonnost.....	16
3.2.3.2 Analytika	17
3.2.3.3 Shrnutí	17
4 Vlastní práce	18
4.1 Analýza současných procesů.....	18
4.1.1 Tvorba rozpočtu	18
4.1.2 Vystavení faktury.....	18
4.1.3 Přijetí faktury/dokladu	18
4.1.4 Vydání zálohy	19
4.1.5 Zpětné proplacení	19
4.1.6 Pokladna.....	19
4.1.7 Výsledek analýzy aktuálních procesů a systémů	19
4.2 Analýza uživatelských požadavků	20
4.2.1 Funkční požadavky	20
4.2.2 Nefunkční požadavky	21

4.3	Hlavní případy užití.....	21
4.3.1	Seznam účastníků.....	21
4.3.2	Uživatelské scénáře.....	22
4.4	Návrh aplikace.....	27
4.4.1	Model databáze	28
4.4.2	Uživatelské rozhraní	32
5	Závěr.....	37
6	Bibliografie.....	38

Seznam obrázků

Obrázek 1 -	Struktura národního hospodářství (Pelikánová, 2018).....	13
Obrázek 2 -	Docker vs. Virtuální server (Docker vs Virtual Machine – Understanding the Differences, 2019).....	14
Obrázek 3 -	Databázový model.....	31
Obrázek 4 -	Wireframe č.1: vizualizace UC1	32
Obrázek 5 -	Wireframe č.2: vizualizace UC8	32
Obrázek 6 -	Wireframe č.3: vizualizace UC2	33
Obrázek 7 -	Wireframe č.4: vizualizace UC3	34
Obrázek 8 -	Wireframe č.5: vizualizace UC11	35
Obrázek 9 -	Wireframe č.6: vizualizace UC13	35
Obrázek 10 -	Wireframe č.7: vizualizace UC14	36

Seznam tabulek

Tabulka 1 -	Docker vs. Virtuální server (Docker vs Virtual Machine – Understanding the Differences, 2019).....	15
-------------	--	----

Seznam použitých zkratk

BPMN - Business Process Model and Notation

DPH – Daň z přidané hodnoty

IT – informační technologie

NNO – Nestatní nezisková organizace (non-government organization)

SQL – Structured Query Language

1 Úvod

Během mého působení ve vedení organizace XYZ¹ po dobu dvou let, jedna z časově náročných činností bylo opakované tvoření finančních přehledů za účelem zjistit, jak si organizace finančně stojí a zda se plní plánované cíle. V rámci finančních procesů se využívá několik účtů, pokladen a systémů s absencí automaticky generovaných přehledů.

Finanční výkazy jsou stěžejní zdroj informací pro sledování vývoje podniku. Jsou také důležitým podkladem pro rozhodování v rámci řízení podniku a ukazatelem zdraví podniku. Je tedy důležité mít finanční přehledy neustále dostupné a aktuální.

Tato situace mě motivovala k tomu, abych využil své znalosti získané během studia IT oboru a zároveň znalosti procesů organizace po dvouleté zkušenosti ve vedení. V rámci své bakalářské práce jsem zanalyzoval finanční procesy organizace včetně všech využívaných systémů a navrhl jsem jeden komplexní systém, který dokáže obsloužit veškeré finanční potřeby organizace a nabízí vedení automaticky generované přehledy o aktuálním finančním stavu organizace.

¹ Reálný název organizace bude v rámci práce nahrazen pseudonymem XYZ.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem teoretické části práce je uvedení do problematiky správy financí s důrazem na správu financí neziskové organizace a analýzy současného stavu existujících procesů a systému pro správu financí organizace.

Cílem praktické části práce je vytvoření doporučení na úpravu procesů organizace směřující k zrychlení a zjednodušení správy financí a tvorba zadávací dokumentace systému, který tyto procesy podpoří.

2.2 Metodika

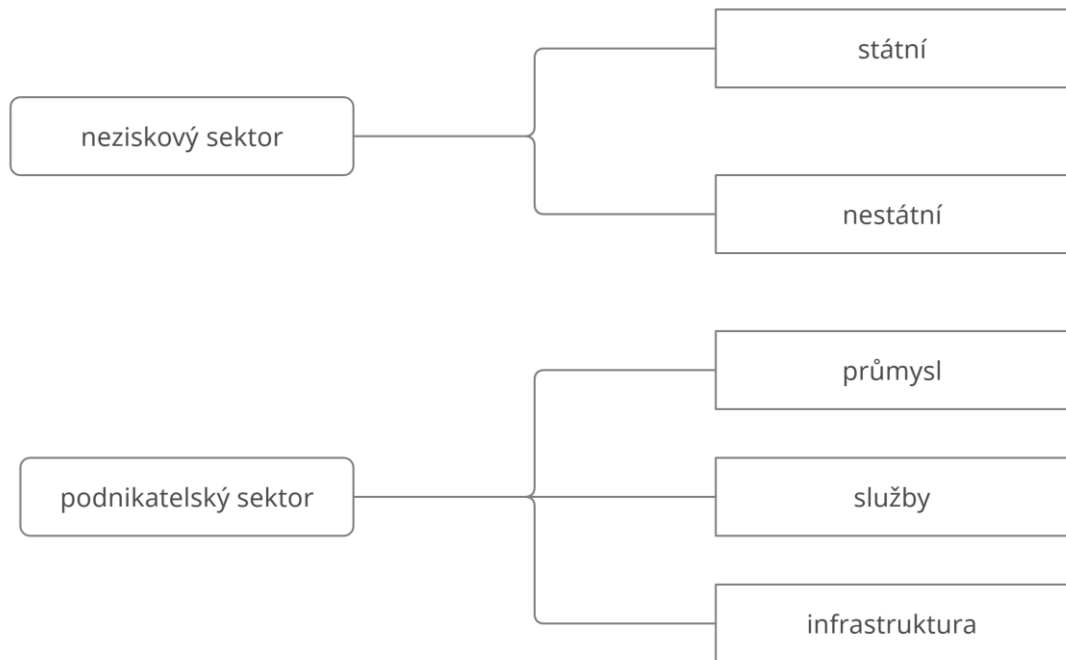
Bakalářská práce pojednává o správě financí neziskové organizace a s tím spojeným informačním systémem. V první řadě tedy nastuduji zákonné povinnosti spojené se správou financí neziskových organizací, následně pak nastuduji jazyk UML pro modelování diagramů a metodiku BPMN (Business Process Model and Notation) používanou pro modelování procesů. Po prostudování teoretických základů zanalyzuji aktuální procesy správy financí v organizaci, následně navrhu zlepšení těchto procesů a vytvořím zadávací dokumentaci pro systém na základě nově navržených procesů.

Bakalářská práce je rozdělena do 2 částí. V první části se budu věnovat teorii, tedy uvedení do problematiky správy financí neziskových organizací, návrhu informačních systémů a samotné analýze stávajících procesů v organizaci. V rámci druhé části navrhu změnu procesů za účelem zefektivnění a zjednodušení správy financí a vytvořím zadávací dokumentaci pro systém podporující navržené změny.

3 Teoretická východiska

3.1 Postavení neziskové organizace v národním hospodářství

Ve smíšené ekonomice se setkáváme s ekonomickou aktivitou různých druhů organizací. Jedná se o soukromé ziskové, soukromé neziskové organizace a organizace veřejného sektoru. (Pelikánová, 2018)



Obrázek 1 - Struktura národního hospodářství (Pelikánová, 2018)

V neziskovém sektoru jsou organizace a instituce, které nejsou primárně založeny za účelem dosahování zisku. Na rozdíl od komerčních organizací, cílem neziskové organizace není vytváření a rozdělování zisku mezi vlastníky, ale přímá produkce užitku. (Pelikánová, 2018)

Omezení v této oblasti je povinnost případný zisk reinvestovat zpět do hlavní činnosti či rozvoje svého smyslu existence. Někdy jsou tato označení nepřesným vyjádřením faktu, že tyto organizace hospodaří se schodkovým rozpočtem, který je pokrýván z veřejných rozpočtů. (Dobrozemský, 2016)

Dalšími používanými pojmy jsou označení státní a nestátní nebo nevládní (mimovládní) organizace. Tato označení vyjadřují explicitní odluku od vlády (včetně regionálních úrovní). V zahraničí se většinou používá pojem non-governmental organization (neboli NNO). (Dobrozemský, 2016)

Dle (Pelikánová, 2018) jsou základními rysy NNO:

- Veřejná prospěšnost
- Dobrovolnost
- Neziskovost
- Samosprávnost
- Soukromé vlastnictví

3.2 Technologie

3.2.1 Architektura

Tato kapitola je zaměřena na srovnání aktuálně používané serverové architektury v organizaci a v současné době populární architektuře využívající umístění aplikací do nezávislých kontejnerů.

3.2.1.1 Virtuální server

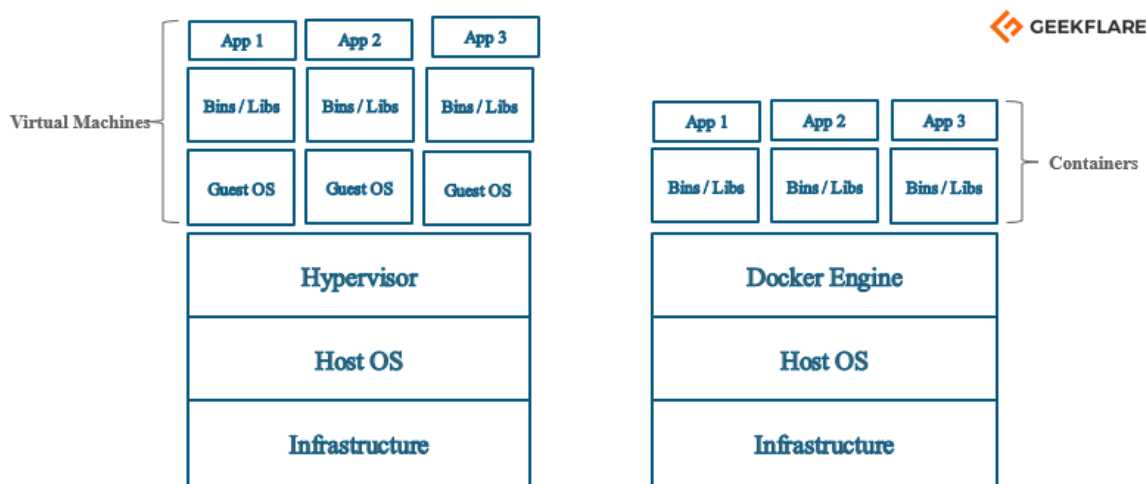
Virtuální server je systém, který se chová stejně jako počítač. Na jednom fyzickém počítači může běžet několik virtuálních serverů, které se však chovají jako samostatná jednotka včetně vlastního operačního systému. (Docker vs Virtual Machine – Understanding the Differences, 2019)

3.2.1.2 Docker

Docker je nástroj, který využívá kontejnery pro tvorbu, nasazení a běh aplikace. Aplikaci včetně jejích závislostí umístí do kontejneru, který se následně může spustit na kterémkoli zařízení, které obsahuje Docker Engine. (Docker vs Virtual Machine – Understanding the Differences, 2019)

3.2.1.3 Porovnání Dockeru a virtuálního serveru

Největší rozdíl Dockeru a virtuálního serveru je přístup k operačnímu systému počítače, na kterém daná architektura běží. Virtuální servery operační systém nesdílí, Docker kontejnery naopak hostující operační systém sdílí. Důsledkem tohoto přístupu jsou Docker kontejnery řádově mnohem menší, neboť obsahují pouze zdrojové kódy aplikace a jejích závislostí, nemusí již však obsahovat samotný systém. Grafické znázornění je vyobrazeno na Obrázek 2.



Obrázek 2 - Docker vs. Virtuální server (Docker vs Virtual Machine – Understanding the Differences, 2019)

V rámci možnosti replikace virtuálních serverů a Docker kontejnerů, pokud aplikace běžící na virtuálním serveru je třeba spustit na jiném stroji, musí se zajistit stejné

prostředí, což v případě starších aplikací a zastaralosti použitých knihoven není jednoduché. Při využití Docker kontejnerů stačí přesunout daný kontejner s aplikací na jiný stroj a následně spustit. Samotný kontejner s aplikací obsahuje veškerý kód potřebný pro běh aplikace včetně jejich závislostí na externích knihovnách. Pro spuštění stačí splnit podmínku, že na daném stroji je nainstalovaný Docker Engine, který daný kontejner dokáže spustit. Krátké shrnutí rozdílů virtuálního serveru a Docker kontejneru je sepsáno v Tabulka 1.

Virtuální server	Docker kontejner
Izolace na úrovni hardwaru	Izolace na úrovni softwaru
Každý virtuální server má vlastní operační systém	Docker kontejnery může sdílet operační systém
Spuštění v řádu minut	Spuštění v řádu vteřin
Velikost řádově v gigabytech	Velikost řádově v megabytech
Výkonnostně více náročné	Výkonnostně méně náročné

Tabulka 1 - Docker vs. Virtuální server (Docker vs Virtual Machine – Understanding the Differences, 2019)

3.2.1.4 Shrnutí

V rámci organizace je velká fluktuace členů, velmi často dochází k situaci, že programátor potřebuje upravit aplikaci a již nemá propojení na původního tvůrce. Z důvodu, že některé aplikace mohou být provozovány i desítky let je těžké vytvořit lokální prostředí tak, aby bylo možné aplikaci spustit na počítači programátora a provádět zde změny. Složitost spuštění starých aplikací na lokálním počítači je většinou způsobena využitím starých verzí programů a knihoven, které již v současné době nejsou dostupné. S ohlednutím na tuto skutečnost i na tendenci organizace přejít do prostředí cloudu bude pro provoz aplikace vhodné využít Docker.

3.2.2 Technologie

V rámci této sekce jsou rozebrány a porovnány technologie, které mohou být využity pro vývoj navrhnutého systému. PHP je technologie aktuálně využívána ve většině aplikací organizace. Javascript patří mezi nejpobulárnější technologie pro vývoj webových aplikací v současnosti (5 Best Programming Languages for Web Development in 2021, 2021).

3.2.2.1 PHP

PHP je zkratka pro Hypertextový preprocesor vytvořený Rasmsem Lerdorfem v roce 1994. Jedná se o open source programovací jazyk navrhnutý specificky pro tvorbu webových stránek. Přestože se jedná o jazyk běžící na straně serveru využívá se i pro běžné programování. (Hooda, 2019)

PHP je až na pár implementačních výjimek synchronní jazyk. Ve chvíli kdy první část kódu trvá déle, zbytek programu čeká na jeho vyhotovení a celkový běh aplikace pak může být pomalejší. (Hooda, 2019)

3.2.2.2 Javascript

Javascript, přesněji framework nesoucí název Node.js je open source jazyk postavený na Javascriptovém Chrome Enginu a využívá se pro tvorbu rychlých a škálovatelných aplikací. První verze frameworku byla zveřejněna roku 2009 autorem Ryanem Dahlem. (Hooda, 2019)

Node.js je asynchronní a v porovnání s PHP jsou aplikace v něm psané rychlejší. (Hooda, 2019)

3.2.2.3 Srovnání PHP a Node.js

Obě technologie běží na straně serveru, jelikož ovšem Node.js je psán v Javascriptu, při psaní webové aplikace může být i část aplikace běžící u klienta psána v Javascriptu, což znamená, že celkově může být aplikace napsána pouze v jednom jazyce. Na rozdíl od Javascriptu je PHP pouze server-side, pro část aplikace běžící u klienta jsou využívány další syntaxe, např. HTML či Javascript. (Hooda, 2019)

V případě využití externích knihoven či modulů, obě technologie disponují doplňky, které se starají o doinstalování či aktualizování všech potřebných balíčků a knihoven. (Hooda, 2019)

Podobně na tom jsou technologie i v rámci frameworků, obě mají několik celosvětově známých a dlouhodobě vyvíjených frameworků díky kterým je vývoj v rámci dané technologie o něco jednodušší. (Hooda, 2019)

V případě možnosti typů databází, Node.js oproti PHP využívá technologii NoSQL, která umožňuje využití i jiných než SQL databází. (Hooda, 2019)

U dřívějších verzí PHP bylo třeba nastavit webový server pro provoz aplikace v tomto jazyce. V dnešní době již jak PHP tak Node.js mají v základě zabudovaných jednoduchý webový server, který je pro provoz aplikace dostačující. (Hooda, 2019)

V rámci aktuálních trendů využívání konkrétních technologií se PHP využívá spíše pro aplikace, které nevyžadují častou interakci mezi klientem a serverem (např. administrační systémy). Node.js je využíváno pro aplikace s vysokou interakcí mezi klientem a serverem (např. chatovací aplikace). (Hooda, 2019)

3.2.2.4 Shrnutí

Převážná většina interních aplikací v organizaci je psána v jazyce PHP a veškeré IT oddělení je zaměřeno na vývoj v tomto jazyce. S přihlédnutím na tento fakt a také na předpoklad, že u aplikace pro správu financí se neočekává častá interakce klienta a serveru bude pro realizaci níže navržené aplikace využito technologie PHP.

3.2.3 Databáze

V následujících kapitolách byly porovnány jedny z nejpoužívanějších databází pro správu a ukládání dat webových aplikací, přesněji se jedná o databázi MySQL a PostgreSQL. Obě databáze jsou také provozovány v rámci organizace pro jiné aplikace.

3.2.3.1 Výkonnost

S každou verzí je rozdíl mezi MySQL a PostgreSQL menší v oblasti čtení a zápisu dat. PostgreSQL ovšem nabízí pár vlastností, které MySQL postrádá. Mezi tyto vlastnosti patří možnost uložení výsledku SQL dotazu jako tabulku, která může být dostupná dalšími

dotazy. PostgreSQL nabízí více druhů indexů, např. částečné indexování, které je vhodné pro filtrování dat. Dalším druhem jsou indexy bitmapové, které jsou vhodné pro práci s kategoriickými daty. Na základě těchto vlastností je pro analyticky náročné úlohy často volena právě databáze PostgreSQL. (PostgreSQL vs MySQL: Which one should you choose?, 2020)

3.2.3.2 Analytika

V rámci analytických operací s daty dle (PostgreSQL vs MySQL: Which one should you choose?, 2020) nabízí PostgreSQL některé funkce navíc:

- funkce, které umožňují výpočty napříč řádky (např. pohyblivé průměry)
- funkce pro práci s časem a daty
- statistické funkce (např. regresní funkce)

3.2.3.3 Shrnutí

MySQL nabízí velmi dobrou uživatelskou podporu a v oblasti jednodušších dotazů je rychlejší čtení a práce jednodušší. (PostgreSQL vs MySQL: Which one should you choose?, 2020)

PostgreSQL na úkor nižší čtecí rychlosti nabízí užitečné funkce, které se dají využít velmi dobře při analytické práci s daty a v rámci složitých dotazů nad větším množstvím tabulek. Práce s ní je složitější z důvodu její komplexnosti. (PostgreSQL vs MySQL: Which one should you choose?, 2020)

V rámci systému pro správu financí jsou očekávány složité dotazy napříč několika databázovými tabulkami a následné tvorby analytických výstupů z uložených dat. Z tohoto důvodu bude využita databáze PostgreSQL, která je pro tento typ operací optimalizována.

4 Vlastní práce

4.1 Analýza současných procesů

Analýza je zaměřena na nejčastější procesy, které v rámci fungování organizace probíhají. Jedná o procesy vystavení faktury, evidence účetního dokladu a jeho případné proplacení. Situace, které nastávají pouze v řádů jednotek případů ročně (např. zahraniční platby, cesty do zahraničí a náklady s tím spojené) do analýzy zahrnuty nebyly. Analýza byla provedena na základě několika diskuzí jak s vedením organizace, tak i s členy týmu financí, kteří mají o všech procesech týkajících se financí detailní přehled.

4.1.1 Tvorba rozpočtu

Rozpočtové období organizace je obvykle jeden rok. Na toto období si organizace vytvoří jeden hlavní rozpočet a ten je následně dělen na části dle aktuální struktury. Typicky se hlavní rozpočet dělí na rozpočty jednotlivých týmů a hlavních projektů. Ty se dělí na menší podskupiny, které jsou již koncové. Aktuálně využívaný systém neumožňuje další dělení podskupin.

Než je rozpočet zaevidován do systému musí být nejdříve navržen a následně schválen. Návrh jednotlivých položek rozpočtů provádí vždy vedoucí týmu nebo projektu, tyto návrhy se zapisují do sdílené tabulky, která na základě navržených položek spočítá očekávané cashflow. Návrh se následně validuje, zda v některých měsících nedochází k zápornému cashflow a zda veškeré navržené položky jsou v souladu s hlavní myšlenkou a cíli organizace.

Po schválení rozpočtu je rozpočet zadán do systému, kde již následně není možnost jeho změny.

4.1.2 Vystavení faktury

Faktury se vystavují na základě objednávek, které se vystavují automaticky v systému na základě vytvoření objednávky v národním systému. Tento systém je v podstatě e-shop pro konkrétní služby pro firmy, které organizace nabízí. Druhá možnost je vytvoření objednávky na míru v případě projektu či služby, která běžně nabízena není.

Pro vystavení faktury se musí člen přihlásit do systému iDoklad, který slouží k veškerému vystavování faktur. Každý člen financí má pro přístup do systému vlastní účet, který v ojedinělých případech sdílí s dalšími členy (např. po skončení většího projektu, kdy se běžně do fakturace zapojí i členové mimo tým financí). Výhoda systému je, že po vystavení faktury sám hlídá její proplacení a při prodlení platby zasílá upozornění.

Vystavená faktura je následně vytištěna a zaslána účetní k zaúčtování.

4.1.3 Přijetí faktury/dokladu

Při přijetí faktury či dokladu se daný výdaj eviduje v interním systému organizace. Každý doklad je naskenován a v systému zaevidován včetně data, kdy byl výdaj realizován, příslušnost ke konkrétnímu rozpočtu, způsob jakým byl výdaj uhrazen, celková výše výdaje a případně sazba DPH. Problém u evidence DPH nastává, že ke každému dokladu v systému lze nastavit pouze jednu sazbu, ve chvíli kdy účtenka obsahuje položky s rozdílnými sazbami DPH musela by se evidovat dvakrát a ručně rozpočítávat, která část

z celkového výdaje je pod konkrétní sazbou. Z tohoto důvodu se výše sazby DPH nezadává. Systém dokáže na základě aktuálních výdajů spočítat předpokládané zálohy finančnímu úřadu, ale z důvodu nezadávání správných sazeb DPH není výpočet spolehlivý.

Ve chvíli kdy člen zaeviduje doklad do systému, musí daný záznam projít schválením týmu financí. Tým zkontroluje, že jsou veškeré údaje o dokladu správně zadány a že mají fyzicky daný doklad či fakturu u sebe, aby ji mohli předat účetní. Když je vše v pořádku, daný doklad schválí a ten je již natrvalo propsán do systému.

4.1.4 Vydání zálohy

V rámci organizace má přístup k bankovním účtům a pokladnám pouze prezident a poté jeden až tři členové týmu financí. Není možné, aby každý člen mohl provádět platbu pomocí karty přímo z účtu organizace. Běžně členové zaplatí výdaj hotově ze svých peněz a ten je jim následně zpětně proplacen. Pokud se jedná o větší částku, může si člen požádat o vydání zálohy. Záloha je mu vyplacena z pokladny a je v souladu s očekávanou výší výdaje.

4.1.5 Zpětné proplacení

Členové běžně doklady uhradí v hotovosti z vlastních peněz a následně při zaevidování dokladu žádají o zpětné proplacení. Člen doklad řádně zaeviduje a pokud je doklad schválen a propsán do systému může požádat o proplacení. Částka může být vyplacena v hotovosti případně na bankovní účet.

4.1.6 Pokladna

Evidence stavu pokladny je aktuálně evidována do papírových archů, kde se zaznamenávají veškeré transakce. Pro získání aktuálního stavu pokladny je nutná fyzická přítomnost v kanceláři a přístup do trezoru.

4.1.7 Výsledek analýzy aktuálních procesů a systémů

Na základě analýzy aktuálních procesů bylo zjištěno, že správa financí aktuálně probíhá na více místech najednou, což zvyšuje riziko chyby. Neexistuje zde žádný automatizovaný mechanismus pro kontrolu aktuálního hospodaření, veškerý přehled aktuálních stavů bankovních účtů, pokladen a rozpočtů si musí vedení vytvářet samo v rámci sdílených tabulek, které plní ručně daty z několika systémů. Z důvodu vysoké pracovního zatížení se intervaly kontroly prodlužují, což také zvyšuje riziko chyb.

Existence více systémů celkový proces evidence financí zhoršuje a prodlužuje čas pro zaškolení nového člena financí. Jednoduchost, přehlednost a možnost rychlého zaučení do procesů je však v rámci organizace klíčová, neboť z důvodu dobrovolnosti je fluktuace členů oproti firmě poměrně vysoká.

Procesy v organizaci splňují veškeré zákonem stanovené povinnosti, chybí však jednotné místo, které by aktuálně stanovené procesy zpřehlednilo a zrychlilo. Tento problém řeší zavedení jednoho systému, který umožní správu veškerých financí, včetně kompletních přehledů o aktuálních finančních tocích, které jsou klíčové pro správné rozhodování.

4.2 Analýza uživatelských požadavků

4.2.1 Funkční požadavky

FP1 – Rozpočet a dělení

Aplikace nebude početně omezovat dělení rozpočtu na kategorie (menší rozpočty).

FP2 – Plánování rozpočtu

V rámci aplikace bude možné rozpočet plánovat. Celkově bude mít rozpočet minimálně tyto stavy: založení, k plánování, ke schválení, schválený, otevřený, uzavřený, archivovaný. Aplikace umožní předvyplnění kategorií a částek z rozpočtů předchozích.

FP3 – V přehledu rozpočtu rozdíl skutečnosti oproti plánu

Aplikace přehledně zobrazí rozdíl plánovaných transakcí oproti skutečným.

FP4 – Kontrola rozpočtu

Aplikace upozorní administrátory v případě překročení/nedodržení rozpočtu.

FP5 – Vyhodnocení a porovnání rozpočtů

Při uzavření rozpočtu aplikace vyhodnotí, jak si organizace za dané období vedla. Porovná očekávání vůči skutečnosti a umožní porovnání vůči předchozím obdobím.

FP6 – Sledování bilance

Aplikace umožní sledování aktuální bilance finančních zdrojů (pokladny, bankovní účty).

FP7 – Pokladna

Aplikace umožní evidenci transakcí probíhajících v pokladnách. Bude možno evidovat více pokladen (více měn). Aplikace nabídne případný přepočtení měny na CZK dle aktuálního kurzu.

FP8 – Zálohy

Aplikace umožní evidovat transakci vydání zálohy. V rámci zálohy je třeba evidovat z jaké pokladny byla vyplacena, účel vydání zálohy a časový termín pro dodání dokladu.

FP9 – Požadavek na zaplacení dokladu

Aplikace umožní u dokladu požádat o proplacení. Touto žádostí se zajistí, že žádná zaevidovaná faktura nezůstane nezaplacená.

FP10 – Možnost nahrání dokladu z mobilu

Do aplikace bude možnost kopii dokladu nahrát nejen jako soubor ale také za využití kamery (např. v mobilním telefonu).

FP11 – Evidence DPH

V rámci dokladu bude možné nastavit více sazeb DPH.

FP12 – Párování transakcí s doklady

Aplikace bude automaticky párovat transakce na zdrojích (pokladny, bankovní účty) s doklady. Ve chvíli kdy automatické párování transakce selže, provede párování manuálně uživatel.

4.2.2 Nefunkční požadavky

NP1 – Použitelnost aplikace

Aplikace musí být použitelná nejen na počítačích ale také na tabletech a mobilních telefonech.

NP2 – Přístupnost aplikace

Aplikace musí být dostupná přes internet po ověření uživatele (přihlášení účtem organizace).

NP3 – Obnovitelnost aplikace

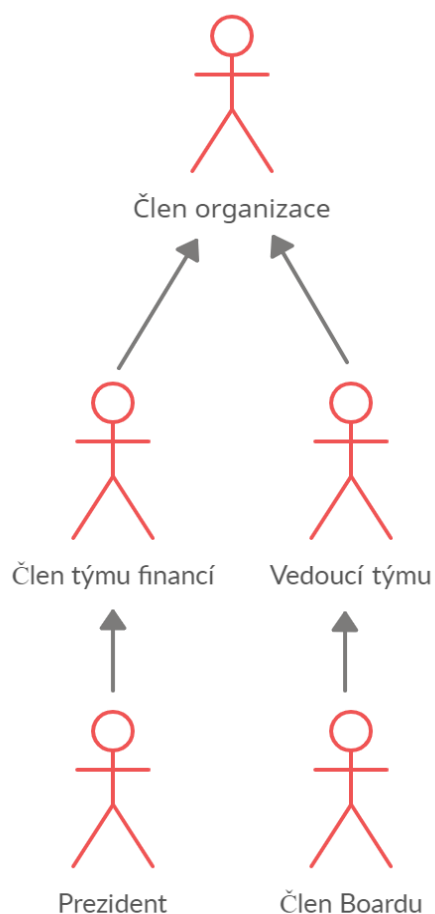
Aplikace bude obsahovat klíčové informace pro běh organizace, musí být pravidelně zálohována. Obnova musí být možná minimálně 7 dní zpětně.

4.3 Hlavní případy užití

Pro zaznamenání případů užití byly využity Use case diagramy. V rámci každého případu užití je uveden seznam účastníků, popis, scénář a samotný diagram.

4.3.1 Seznam účastníků

- **Člen organizace** – Řádový člen organizace, defaultně může vkládat do systému doklady, žádat o zálohu či proplacení jím uhrazeného dokladu.
- **Člen týmu financí** – Člen týmu financí může upravovat vložené (neschválené doklady) doklady, schvalovat doklady.
- **Vedoucí týmu** – Vedoucí týmu má za úkol hlídat aby nedošlo k přečerpání rozpočtu. V rámci svého rozpočtu může peníze v rámci jednotlivých kategorií přerozdělovat.
- **Člen Boardu** – Vidí do všech rozpočtů včetně jejich čerpání, mají přístup k podrobným statistikám aktuálního hospodaření.
- **Prezident** – Hlavní administrátor, který má nejvyšší pravomoci v rámci všech rozpočtu jeho centra. Jedná se o osobu, která je zároveň statutární orgán a je tedy i právně zodpovědná za správné hospodaření organizace.
- **<<System>>** – Zajišťuje synchronizaci údajů s externím systémem iDoklad a bankovními účty. Spouští automatické párování pohybů s jednotlivými doklady.



4.3.2 Uživatelské scénáře

UC1 – Založit rozpočet

Uživatel založí rozpočet v systému a umožní tak jeho plánování.

Aktéři

- Prezident
- Člen Boardu

Scénář

1. Systém uživateli zobrazí formulář pro založení rozpočtu
2. Uživatel vyplní formulář, povinná pole jsou:
 - Název
 - Typ
 - Správce – Osoba odpovědná za daný rozpočet, výchozí nastavení je osoba, která rozpočet zakládá.
 - Sledující – Osoby, které dostávají notifikace o změnách v rozpočtu. Defaultní nastavení je správce, členové Boardu a prezident.
3. Systém ověří, zda jsou vyplněna povinná pole
 - a. Formulář není vyplněn správně – návrat ke kroku 2
4. Systém uloží data o rozpočtu do databáze

UC2 – Zadat plány rozpočtu

Uživatel může plánovat očekávané příjmy a výdej v rámci rozpočtu, případně rozpočet dělit na kategorie.

Aktéři

- Vedoucí týmu
- Člen Boardu
- Prezident

Scénář

1. Systém zobrazí formulář pro zadání jednotlivých plánů rozpočtu
2. Uživatel rozdělí rozpočet na libovolný počet kategorií
3. Uživatel u každé kategorie nastaví očekávané příjmy a výdaje na každý měsíc
4. Systém ověří, že pro každý měsíc jsou vyplněny očekávané příjmy a výdaje
 - a. Některé údaje nejsou vyplněny – návrat ke kroku 3
5. Systém na základě očekávaných příjmů a výdajů kategorií rozpočtu dopočítá celkové příjmy a výdaje rozpočtu
6. Systém údaje o rozpočtu uloží do databáze

UC3 – Zobrazit přehled rozpočtu

Uživatel si může zobrazit přehled rozpočtu, tedy očekávané finanční toky a v případě otevřeného rozpočtu i skutečné finanční toky.

Aktéři

- Vedoucí týmu
- Člen týmu financí
- Člen Boardu
- Prezident

Scénář

1. Systém uživateli zobrazí přehled očekávaných příjmů a výdajů podle kategorií a měsíců
2. Rozpočet je ve stavu „otevřený“
 - a. Systém zobrazí skutečné příjmy a výdaje

UC4 – Schválit rozpočet

Uživatel schválí rozpočet.

Aktéři

- Prezident

Scénář

1. Systém zobrazí přehled rozpočtu (include UC3)
2. Prezident schválí rozpočet
3. Systém označí rozpočet za schválený
4. Datum začátku rozpočtu již nastalo
 - a. Systém označí rozpočet za otevřený
5. Systém uloží údaje o rozpočtu do databáze

UC5 – Porovnat rozpočty

Uživateli se zobrazí přehled několika rozpočtů. Uživatel si pomocí přehledu může zanalyzovat jak si organizace stojí oproti předchozím rokům.

Aktéři

- Člen boardu
- Prezident

Scénář

1. Uživatel zvolí rozpočty, které chce porovnat
2. Systém zobrazí přehledy zvolených rozpočtů

UC6 – Uzavřít rozpočet

Uživatel uzavře rozpočet, tedy do rozpočtu již nepůjde přidávat doklady.

Aktéři

- Prezident

Scénář

1. Uživatel zvolí rozpočet, který chce uzavřít
2. Systém označí daný rozpočet za uzavřený

UC7 – Vyhodnotit rozpočet

Systém vyhodnotí rozpočet, zobrazí tedy přehled jak si organizace během období rozpočtu vedla zobrazením rozdílů mezi očekávanými a skutečnými finančními toky.

Aktéři

- Člen Boardu
- Prezident

Scénář

1. Uživatel zvolí rozpočet, který chce vyhodnotit
2. Systém zobrazí spočítá rozdíly mezi očekávanými a skutečnými finančními toky
3. Systém zobrazí přehled rozdílů mezi očekáváním a skutečností

UC8 – Zaevidovat doklad

Uživatel zaeviduje doklad do systému.

Aktéři

- Člen
- Člen týmu financí
- Vedoucí týmu
- Člen Boardu
- Prezident

Scénář

1. Systém zobrazí formulář pro zaevidování dokladu
2. Uživatel vyplní formulář
3. Uživatel nahraje digitální kopii dokladu
 - a. Uživatel nahraje soubor
 - b. Uživatel použije kameru v zařízení pro vyfocení dokladu
4. Systém zkontroluje řádné vyplnění formuláře
5. Systém označí doklad do stavu Ke schválení
6. Systém uloží informace o dokladu do databáze
7. Systém pošle notifikaci o novém dokladu sledujícím daného rozpočtu

UC9 – Upravit záznamu dokladu

Uživatel upraví záznamy o dokladu.

Aktéři

- Člen týmu financí
- Člen Boardu
- Prezident

Scénář

1. Uživatel zvolí doklad k úpravě
2. Systém zobrazí formulář s předvyplněnými hodnoty z databáze
3. Uživatel upraví údaje ve formuláři
4. Systém uloží upravené informace do databáze

UC10 – Schválit doklad

Uživatel zkontroluje zda se údaje v systému shodují s údaji dokladu, pokud je vše správně doklad schválí.

Aktéři

- Člen týmu financí
- Člen Boardu
- Prezident

Scénář

1. Systém zobrazí informace o dokladu
2. Uživatel klikne na tlačítko schválit doklad
3. Systém zobrazí hlášku, zda uživatel opravdu chce doklad schválit
4. Uživatel potvrdí schválení dokladu
5. Systém označí doklad za schválený a zaeviduje ho do konkrétního rozpočtu

UC11 – Vytvořit žádost o uhrazení dokladu

Uživatel požádá o uhrazení dokladu (převážně se jedná o uhrazení faktury).

Aktéři

- Člen
- Člen týmu financí
- Vedoucí týmu
- Člen Boardu
- Prezident

Scénář

1. Uživatel zvolí doklad, který chce uhradit
2. Systém pošle notifikaci členům týmu financí
3. Systém změní stav dokladu na „čeká na zaplacení“

UC12 – Změnit stav žádosti o uhrazení dokladu

Uživatel změní stav dokladu (převážně z „čeká na zaplacení“ do stavu „uhrazeno“).

Aktéři

- Člen týmu financí
- Člen Boardu
- Prezident

Scénář

1. Uživatel označí doklad za uhrazený
2. Systém změní stav dokladu na „uhrazený“

UC13 – Vytvořit finanční zdroj

Uživatel přidá do systému nový finanční zdroj (např. nová pokladna či bankovní účet).

Aktéři

- Člen týmu financí
- Člen Boardu
- Prezident

Scénář

1. Systém zobrazí formulář pro tvorbu finančního zdroje
2. Uživatel vyplní formulář, povinné položky jsou:
 - Název finančního zdroje
 - Měna finančního zdroje
 - Typ finančního zdroje
3. Systém ověří zda je formulář korektně vyplněn
 - a. Formulář není správně vyplněn – návrat ke kroku 2
4. Systém uloží informace o finančním zdroji do databáze

UC14 – Archivovat finanční zdroj

Uživatel archivuje finanční zdroj (například při zrušení bankovního účtu či poklady).

Aktéři

- Prezident

Scénář

1. Uživatel zvolí finanční zdroj který má být archivován
2. Systém označí finanční zdroj jako archivovaný
3. Systém uloží informace o zdroji do databáze

UC15 – Zadat korekci stavu finančního zdroje

Uživatel zadá korekci finančního zdroje aby se stav v systému shodoval se skutečným stavem finančního zdroje (primárně se jedná o manko/přebytek peněz v pokladně).

Aktéři

- Člen týmu financí
- Člen Boardu
- Prezident

Scénář

1. Systém zobrazí formulář pro korekci finančního zdroje, povinné údaje jsou:
 - Částka pro korekci (plusová či minusová hodnota)
 - Důvod korekce (např. manko, přebytek)
2. Uživatel vyplní formulář
3. Systém upraví aktuální stav finančního zdroje do databáze

UC16 – Vytvořit finanční tok

Uživatel zaeviduje finanční tok (přesun peněz z jednoho finančního zdroje na druhý, převážně se jedná o přesun peněz mezi bankovním účtem a pokladnou).

Uživatel

Aktéři

- Člen týmu financí
- Člen Boardu
- Prezident

Scénář

1. Systém zobrazí formulář pro finanční tok, povinné údaje jsou
 - Zdroj (z jakého finančního zdroje částka bude odečtena)
 - Cíl (na jaký finanční zdroj bude částka přičtena)
 - Částka
2. Uživatel vyplní formulář
3. Systém ověří správnost údajů ve formuláři
 - a. Formulář je chybně vyplněn – návrat ke kroku 2
4. Systém uloží záznam o finančním toku do databáze

UC17 – Anulovat finanční tok

Uživatel anuluje finanční tok, stav cílového a zdrojového finančního zdroje je navrácen do původní hodnoty.

Aktéři

- Prezident

Scénář

1. Uživatel zvolí finanční tok, který má být anulován
2. Systém archivuje finanční tok a vyjme ho ze stavu zdrojového a cílového finančního zdroje
3. Systém uloží informace do systému

UC18 – Spárovat finanční tok s evidovaným dokladem

Finanční tok je spárovat s konkrétním dokladem. O spárování se nejdříve pokusí systém automaticky a v případě neúspěchu provede spárování manuálně uživatel.

Aktéři

- Člen týmu financí
- Člen Boardu
- Prezident
- Systém

Scénář

1. Systém vyhledá doklady v systému, které se shodují datem a částkou
 - a. Systém našel shodu s dokladem – Skok na krok 5
2. Systém označí finanční tok za nespárovaný
3. Systém nabídne doklady v období finančního toku
4. Uživatel zvolí doklad, který odpovídá danému finančnímu toku
5. Systém spáruje doklad a finanční tok
6. Systém uloží informace do databáze

4.4 Návrh aplikace

Tato sekce je zaměřena na samotný návrh samotné aplikace. Nejdříve byla navržena samotná struktura tabulek databáze a následně samotné uživatelské rozhraní. Konkrétní návrhy rozhraní pak byly provedeny pouze pro atypické scénáře, tedy obrazovky, které jsou v dnešní době určitým způsobem standardizované (např. přihlášení uživatele) byly při návrhu vynechány. Při návrhu rozhraní byl kladen důraz na uživatelsky přívětivé rozmístění nikoliv konkrétní grafické zpracování. Každý návrh byl konzultován s vedením a členy organizace a následně upraven na základě jejich připomínek.

4.4.1 Model databáze

V rámci této sekce je vytvořený databázový model včetně popisu jednotlivých entit a jejich atributů.

Status

- id: Identifikátor statusu
- name: Název statusu

Document

- id: Identifikátor dokumentu (dokladu)
- realised_at: Datum, kdy byl doklad vystaven
- amount: Celková částka dokladu
- file_name: Název souboru digitální kopie dokladu
- note: Poznámka k dokladu
- created_at: Datum zaevidování dokladu do systému
- updated_at: Datum poslední úpravy
- Status_id: Identifikátor aktuálního statusu (stavu) dokladu
- Transaction_id: Identifikátor spárované transakce
- Request_id: Identifikátor požadavku na proplacení/úhradu
- Currency_id: Identifikátor měny dokladu

Currency

- id: Identifikátor měny
- name: Název měny
- shortcut: Zkratka měny

Item

- id: Identifikátor položky
- created_at: Datum zaevidování položky do systému
- updated_at: Datum poslední změny položky
- amount: Částka dané položky
- Vat_id: Identifikátor sazby DPH pro danou položku
- Document_id: Identifikátor dokladu pod který daná položka spadá
- Currency_id: Identifikátor měny

Vat

- id: Identifikátor sazby DPH
- rate: Výše sazby DPH

Budget

- id: Identifikátor rozpočtu
- name: Název rozpočtu
- start_at: Datum začátku rozpočtu
- end_at: Datum konce rozpočtu
- expected_income: Očekávaný (plánovaný) příjem v rámci rozpočtu
- expected_outcome: Očekávaný (plánovaný) výdaj v rámci rozpočtu
- real_income: Reálný příjem v rámci rozpočtu
- real_outcome: Reálný výdaj v rámci rozpočtu

- note: Poznámka k rozpočtu
- Status_id: Identifikátor aktuálního statusu (stavu) rozpočtu
- Lc_id: Identifikátor lokálního centra pod které rozpočet spadá

Transaction

- id: Identifikátor transakce
- date: Datum provedení transakce
- amount: Částka
- note: Poznámka k transakci
- Status_id: Identifikátor stavu (statusu) transakce
- User_id: Uživatel, který transakci vytvořil
- User_id2: Uživatel, který transakci exekuoval
- Document_id: Identifikátor dokladu s kterým je transakce spárována
- Resource_id: Identifikátor z jakého zdroje je částka stržena
- Resource_id2: Identifikátor zdroje kam je částka přičtena

Resource

- id: Identifikátor zdroje
- name: Název zdroje
- balance: Aktuální bilance zdroje
- note: Poznámka
- Resource_type_id: Identifikátor typu zdroje
- Currency_id: Identifikátor měny zdroje

Request

- id: Identifikátor žádosti
- created_at: Datum vytvoření žádosti
- updated_at: Datum poslední úpravy žádosti
- Request_type_id: Typ žádosti
- Document_id: Doklad s kterým je žádost spojena
- User_id2: Uživatel, který žádost schválil (exekuoval)
- User_id: Uživatel, který žádost vytvořil
- Status_id: Identifikátor stavu (statusu) žádosti

Lc

- id: Identifikátor lokálního centra
- name: Název lokálního centra

User

- id: Identifikátor uživatele
- username: Uživatelské jméno
- email: E-mail uživatele
- created_at: Datum vytvoření uživatele
- password: Heslo uživatele
- Role_id: Identifikátor role uživatele
- Lc_id: Identifikátor lokálního centra ke kterému uživatel náleží

Resource_type

- id: Identifikátor typu zdroje
- Name: Název typu zdroje

Request type

- id – Identifikátor typu žádosti
- name: Název typu žádosti

Role

- id: Identifikátor role
- name: Název role



Obrázek 3 - Databázový model

4.4.2 Uživatelské rozhraní

WF1 – Založení rozpočtu

Samotný systém pro bude využívat více lokálních center (poboček) v rámci České republiky, proto je třeba u každého rozpočtu evidovat i náležitost k danému centru.

Organizace XYZ [Přehled](#) [Rozpočty](#) [Fin. zdroje](#) [Fin. toky](#) [ZAEVIDOVAT DOKLAD](#) 

Založení rozpočtu

Název:

Začátek: 

Konec: 


LC:

Poznámka:

Obrázek 4 - Wireframe č.1: vizualizace UC1


WF2 – Zaevidování dokladu

Při návrhu obrazovky pro evidence dokladu byl kladen důraz na možnost digitální kopii dokladu pořídit přímo fotoaparátem aktuálně používaného zařízení místo pouhého nahrání souboru. Následně obrazovka nabízí přehledné zobrazení jednotlivých daňových sazeb, které jsou na daném dokladu.

Organizace XYZ [Přehled](#) [Rozpočty](#) [Fin. zdroje](#) [Fin. toky](#) [ZAEVIDOVAT DOKLAD](#) 

Zaevidovat doklad

Název:


Datum: 


Celková částka:

Sazby DPH [Přidat sazbu DPH](#)

	Sazba	Částka
Položka jedna	21%	2 000
Položka 2	15%	500

Kopie dokladu

 **Nahrát**

 **Vyfotit**

Poznámka:

Obrázek 5 - Wireframe č.2: vizualizace UC8

WF3 – Plánování rozpočtu


V rámci obrazovky bylo třeba přehledně zobrazit očekávané příjmy a výdaje kategorií v rámci rozpočtu současně s jednotlivými položky rozpočtu. Pro lepší přehled byl doplněn graf, který zobrazí křivku cashflow na základě očekávaných nákladů daného rozpočtu.



Obrázek 6 - Wireframe č.3: vizualizace UC2

WF4 – Přehled rozpočtu

Obrazovka zobrazuje přehledné informace o čerpání konkrétního rozpočtu včetně porovnání očekávaných versus reálných příjmů a výdajů. Obrazovka obsahuje také přehledný graf cashflow, který obsahuje 3 křivky. Plánované hodnoty, skutečné hodnoty a jejich kombinaci.

Organizace XYZ [Přehled](#) [Rozpočty](#) [Fin. zdroje](#) [Fin. toky](#) [ZAEVIDOVAT DOKLAD](#) 

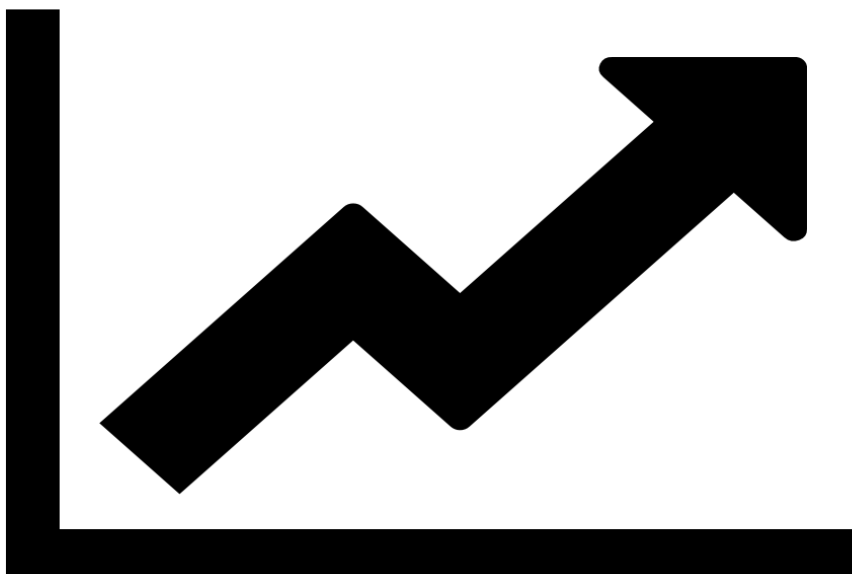
Rozpočet č.4

	Očekávané	Reálné	Rozdíl
Příjmy celkem	25 000	26 500	1 500
Výdaje celkem	25 000	24 000	-1 000
Výsledek		2 500	

Přehled dle kategorií a měsíců

	Leden	Únor	Březen
Kategorie 1	25 000	26 500	1 500
Kategorie 2	-25 000	-24 000	-1 000
Celkem	0	2 500	500

Graf cashflow dle reálných a očekávaných hodnot



Obrázek 7 - Wireframe č.4: vizualizace UC3

WF5 – Žádost o proplacení/uhrazení dokladu

Při žádosti o proplacení či uhrazení dokladu je důležité párování s konkrétním dokladem, aby bylo možné v systému evidovat zda daný doklad byl již proplacen a eliminovat se tak situace neuhrazeného závazku. Pro zjednodušení práce finančníkům je zde možnost vyplnit číslo účtu kam má být částka za doklad zaslána.

Organizace XYZ [Přehled](#) [Rozpočty](#) [Fin. zdroje](#) [Fin. toky](#) [ZAEVIDOVAT DOKLAD](#)

Typ žádosti: Uhrazení Proplacení

Zvolit doklad:

Číslo účtu:

Obrázek 8 - Wireframe č.5: vizualizace UC11

WF6 – Vytvoření finančního zdroje

Organizace pracuje s několika finančními zdroji v různých měnách. Nastávají ale situace, kdy se pro určité projekty používají separátní bankovní účty, pro lepší přehlednost a odlišitelnost byla přidána položka názvu zdroje.

Organizace XYZ [Přehled](#) [Rozpočty](#) [Fin. zdroje](#) [Fin. toky](#) [ZAEVIDOVAT DOKLAD](#)

Typ žádosti: Bankovní úč. Pokladna

Zvolit měnu:

Číslo účtu:


Kód banky:

Název zdroje:

Obrázek 9 - Wireframe č.6: vizualizace UC13

WF6 – Zaevidování finančního toku

V rámci organizace dochází často k realokaci (přesunu peněz) napříč finančními zdroji, tyto transakce je třeba evidovat. Položka „Aktuální kurz“ v rámci obrazovky by měla být doplněna automaticky na základě aktuálního kurzu s možností úpravy uživatelem.

Organizace XYZ [Přehled](#) [Rozpočty](#) [Fin. zdroje](#) [Fin. toky](#) [ZAEVIDOVAT DOKLAD](#) 

Aktuální kurz

Částka

Zdroj

Cíl

Poznámka:

Obrázek 10 - Wireframe č.7: vizualizace UC14

5 Závěr

Cílem práce bylo provést analýzu současného stavu existujících procesů a systému pro správu financí organizace, v rámci praktické části pak provést návrh případné úpravy procesů a nového systému, který procesy podpoří.

Nejprve byla na základě diskuze se současným vedením a členy týmu financí provedena analýza současných procesů v oblasti financí organizace. Během analýzy bylo zjištěno, že pro správu financí není jednotný systém. V rámci financí se využívají tři systémy: pro vystavování faktur, pro evidenci účtenek a pro správu pokladen se využívá evidence do pokladní knihy. Systémy nejsou propojeny a neexistují žádné systémem generované přehledy o aktuálním stavu. Vedení si pro správné rozhodování musí tvořit přehledy ručně a to přepsáním aktuálních údajů ze všech systémů do sdílené tabulky. Tento proces je časově náročný a s vysokým rizikem chybovosti.

Výsledkem práce je návrh systému, který dokáže obsloužit veškeré finanční potřeby organizace a odstraní tak potřebu využití tří systémů. Systém umožní zobrazení přehledů o aktuálním stavu financí organizace v reálném čase a to plně automaticky. Automatizace tvorby finančních přehledů o aktuálním stavu organizace ušetří vedení čas, který budou moci investovat do rozvoje organizace.

6 Bibliografie

DOBROZEMSKÝ, Václav a Jan STEJSKAL, 2016. Nevýdělečné organizace v teorii. 2., aktualizované vydání. Praha: Wolters Kluwer. ISBN 978-80-7552-103-3.

Docker vs Virtual Machine – Understanding the Differences, 2019. In: Geekflare [online]. -: Geekflare [cit. 2021-03-02]. Dostupné z: <https://geekflare.com/docker-vs-virtual-machine/>

HOODA, Parikshit, 2019. PHP vs. Node.js - GeeksforGeeks. GeeksforGeeks [online]. India [cit. 2021-03-02]. Dostupné z: <https://www.geeksforgeeks.org/php-vs-node-js/>

PELIKÁNOVÁ, Anna, 2018. Účetnictví, daně a financování pro nestátní neziskovky. 3., aktualizované vydání. Praha: Grada Publishing. Účetnictví a daně (Grada). ISBN 978-802-7121-175.

PostgreSQL vs MySQL: Which one should you choose?, 2020. Keboola [online]. Praha: Keboola s.r.o. [cit. 2021-03-05]. Dostupné z: <https://www.keboola.com/blog/postgresql-vs-mysql>

5 Best Programming Languages for Web Development in 2021. Medium [online]. -: Medium.com, 2021, 13.01.2021 [cit. 2021-03-13]. Dostupné z: <https://medium.com/javarevisited/top-5-programming-languages-for-web-development-in-2021-f6fd4f564eb6>