

Sem vložte zadání Vaší práce.

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



Bakalářská práce

Návrh a implementace evidenčního systému zahraničních faktur

Aneta Musilová

Vedoucí práce: Ing. David Buchtela, Ph.D.

11. května 2015

Poděkování

Ráda bych poděkovala svému vedoucímu práce panu Ing. Davidu Buchtelovi, Ph.D. za cenné rady, které mi pomohly při tvorbě této práce. Dále bych ráda poděkovala své mamince za odborné, konstruktivní připomínky k práci a za podporu během celé doby studia. V neposlední řadě bych ráda poděkovala Bc. Jiřímu Sixtovi, který mi byl oporou nejen při vzniku této práce.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 11. května 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Aneta Musilová. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Musilová, Aneta. *Návrh a implementace evidenčního systému zahraničních faktur*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce se zabývá analýzou, návrhem a implementací webové aplikace pro evidenci zahraničních faktur. V první části práce jsou shrnuty zákonné požadavky na aplikaci, další části se zabývají zhodnocením existujících řešení, návrhem a následně samotnou implementací aplikace. V závěru jsou zhodnoceny ekonomicko-manažerské přínosy aplikace.

Klíčová slova evidenční systém, účetnictví, zahraniční faktura, webová aplikace

Abstract

This thesis deals with analysis, design and implementation of a web application for foreign invoices evidence. The first part summarizes legal requirements of the application, in further parts there are comparison of existing solutions, design and implementation of the application. The end of this thesis describes economical and managerial benefits.

Keywords evidence system, accounting, foreign invoice, web application

Obsah

| | |
|--|-----------|
| Úvod | 1 |
| Motivace | 1 |
| Cíl práce | 1 |
| Struktura práce | 2 |
| 1 Analýza | 3 |
| 1.1 Požadavky | 3 |
| 1.2 Případy užití | 5 |
| 1.3 Datový model | 9 |
| 2 Existující řešení | 11 |
| 2.1 Desktopové aplikace | 11 |
| 2.2 Webové aplikace | 11 |
| 2.3 Mobilní aplikace | 12 |
| 2.4 Shrnutí | 12 |
| 3 Návrh | 13 |
| 3.1 Volba technologií | 13 |
| 3.2 Diagram tříd | 14 |
| 3.3 Napojení na portál ČNB | 16 |
| 3.4 Export účetních sestav | 17 |
| 3.5 Návrh uživatelského rozhraní | 17 |
| 4 Realizace | 23 |
| 4.1 Použité nástroje | 23 |
| 4.2 Implementace | 24 |
| 4.3 Validace dat | 25 |
| 4.4 Export | 25 |
| 5 Testování | 27 |

| | | |
|----------|--|-----------|
| 5.1 | Testovací scénáře | 27 |
| 5.2 | Vyhodnocení | 28 |
| 6 | Zhodnocení přínosů | 29 |
| 6.1 | Náklady na vývoj a provoz aplikace | 29 |
| 6.2 | Přínosy aplikace pro zákazníka | 29 |
| | Závěr | 31 |
| | Literatura | 33 |
| | A Seznam použitých zkratk | 35 |
| | B Obsah přiloženého CD | 37 |

Seznam obrázků

| | | |
|-----|---|----|
| 1.1 | Diagram případů užití pro základní úkony | 6 |
| 1.2 | Diagram případů užití pro práci s fakturami | 8 |
| 1.3 | Konceptuální model | 10 |
| 3.1 | Diagram tříd | 14 |
| 3.2 | Domovská stránka | 18 |
| 3.3 | Formulář pro zadání nové faktury | 18 |
| 3.4 | Formulář pro přidání nového subjektu | 19 |
| 3.5 | Formulář pro přidání položek faktury | 19 |
| 3.6 | Náhled detailu faktury | 20 |
| 3.7 | Možnosti hledání faktur | 20 |
| 3.8 | Seznam nalezených faktur | 21 |

Úvod

Motivace

Účetnictví je nástroj sloužící ke sledování a zobrazení stavů, toků a výsledků podnikatelské činnosti v peněžních jednotkách. Podnikatelská činnost je vykonávána dvěma typy účetních jednotek - fyzickou nebo právnickou osobou.

Fyzická osoba má povinnost vést tzv. daňovou evidenci (dříve jednoduché účetnictví) nebo účetnictví (podvojně), povinností právnické osoby je vést účetnictví. Náležitosti procesu vedení účetnictví jsou dány zákonem [1].

V rámci této práce bude z celého systému vedení účetnictví řešena pouze evidence zahraničních faktur. Faktura je podle [1] definována jako účetní doklad vystavený za poskytnutí služby či dodání zboží, který musí mít určité náležitosti dané zákonem a popsané dále v této práci. Obecně je možné faktury rozdělit na přijaté a vydané. Zahraniční fakturou se rozumí takový doklad, který byl přijat nebo vydán z nebo do země Evropské unie i mimo ni. Taková faktura může být v českých korunách nebo v cizí měně.

Vést účetnictví lze pochopitelně „ručně“, tzn. bez pomoci počítačů, to se však již řadu let nepraktikuje. V dnešní době lze využít velké množství dostupných programů. Nevýhodou většiny z nich je však nedostupnost z internetu, tzn. jedná se o aplikace desktopové (lze je používat pouze na zařízení - počítači, na kterém jsou nainstalovány). Právě odstranění této nevýhody je stěžejní bod této bakalářské práce.

Cíl práce

Cílem této bakalářské práce je analýza, návrh a implementace webové aplikace určené k evidenci zahraničních faktur. Dále zahrnuje detailní studium této problematiky vzhledem k zákonným požadavkům a porovnání dostupných, existujících řešení. Na závěr byla aplikace otestována, došlo k vyhodnocení navrženého řešení a ekonomicko-manažerských přínosů aplikace.

Struktura práce

Tato práce je rozdělena na následující kapitoly:

- Analýza - tato kapitola se zabývá rozborem požadavků na aplikaci vzhledem k zákonným požadavkům na evidenci zahraničních faktur. Dále obsahuje popis možných případů užití a konkrétní scénáře.
- Existující řešení - tato kapitola porovnává dostupná, existující řešení zabývající se touto problematikou a hodnotí jejich klady a zápory.
- Návrh - tato kapitola obsahuje konceptuální model samotného evidenčního systému, stejně tak jako návrh napojení na portál ČNB a exportu účetních sestav. Dále popisuje návrh uživatelského rozhraní a volbu implementační platformy.
- Realizace - tato kapitola popisuje jednotlivé části implementace.
- Testování - tato kapitola obsahuje popis použitého testování aplikace.
- Zhodnocení - tato kapitola přináší zhodnocení ekonomicko-manažerských přínosů vytvoření aplikace.

Analýza

Analýza je jednou z nejdůležitějších etap vývoje aplikací. Jedná se o popis činností, které bude daná aplikace vykonávat. S pomocí dobře provedené analýzy se lze vyvarovat chyb při implementaci, jejichž odstranění bývá velmi komplikované.

Analýza softwaru se skládá z popisu funkčních a nefunkčních požadavků, diagramů aktivit popisujících procesy zákazníka, diagramů případů užití a datových modelů. Pro účely aplikace vzniklé v rámci této bakalářské práce je možné vypustit diagram aktivit.

1.1 Požadavky

V této podkapitole jsou popsány veškeré požadavky kladené na vzniklou aplikaci. Dělí se do dvou skupin, na funkční a nefunkční (určují omezení kladená na systém, mají zásadní dopad na návrh architektury).

1.1.1 Funkční požadavky

F1 Zadávání dat

Uživatel zadá údaje do příslušného formuláře. Po vyhodnocení správnosti zadaných dat dojde k jejich uložení do databáze.

F2 Respektování zákonných požadavků

Formulář uvedený v bodu F1 je navržen tak, aby korespondoval s platnými zákonnými požadavky na evidenci zahraničních faktur. Faktura (účetní doklad) obecně musí podle [1] a [2] obsahovat následující data:

- Označení účetního dokladu
- Obsah a subjekty (povinně musí být uveden dodavatel, odběratel není povinný)

1. ANALÝZA

- Peněžní částka a množství (zboží nebo služeb)
- Datum vystavení účetního dokladu
- Datum splatnosti
- Datum uskutečnění plnění
- Základ daně
- Sazba daně

Pokud některé z dat chybí, faktura nesmí být akceptována.

F3 Spolupráce s portálem ČNB

Navržená aplikace automaticky stahuje aktuální denní kurzy měn (EUR, USD, GBP - pro účely této práce pouze tyto tři měny, faktura samozřejmě může být v jakékoli měně) vůči české koruně vyhlášené ČNB. Toto velmi zjednodušuje použití aplikace běžnému uživateli, který by jinak musel kurzy měn zadávat do systému ručně.

F4 Výpočet DPH a kurzových rozdílů

Na základě dat zadaných uživatelem a automaticky stažených denních kurzů měn dokáže aplikace vyhodnotit a dopočítat následující:

- DPH - přidanou hodnotou se rozumí rozdíl ceny zboží nebo služeb na výstupu (prodej) a ceny na vstupu (nákup). Pokud je odběratel plátce DPH (subjekt, jehož obrat za nejvýše 12 po sobě jdoucích kalendářních měsíců přesáhne 1 000 000 Kč), má podle [2] povinnost odvést státu daň z tohoto rozdílu ve stanovené výši:
 - Základní sazba (21%) - položky, na které se nevztahuje žádná snížená sazba - např. elektronika
 - První snížená sazba (15%) - např. potraviny, hromadná doprava, ubytovací služby
 - Druhá snížená sazba (10%) - např. knihy, očkovací látky

Kompletní výčet zboží a služeb, na které se vztahuje některá ze snížených sazeb, je uveden v [2].

Základ daně je pro účely této práce vypočítán jako součet ceny za položku na faktuře a cla, pokud se jedná o dovoz mimo země EU. Zahraníční dodavatel neuvádí na faktuře DPH. V případě dodavatele ze země EU si odběratel sám vypočítá výši DPH podle platné sazby. Pokud se jedná o dodavatele mimo země EU, příslušný celní orgán vypočítá clo a obvykle vyměří i DPH (pokud ne, dopočítá si ho odběratel sám).

- Kurzové rozdíly - kurzy měn vyhlašuje ČNB denně. Vzhledem k tomu, že se datum vystavení a datum zaplacení faktury mohou lišit (v ideálním případě by se měla shodovat data zaplacení a splatnosti), může se vzhledem k neustále se měnícím kurzům měn lišit i přepočtení zahraniční měny na CZK. Rozdílem těchto dvou částek vzniká kurzový rozdíl (zisk nebo ztráta).

Tyto informace jsou klíčové pro správný návrh aplikace.

F5 Zobrazení a úprava dat

Zadaná data zahraničních faktur je samozřejmě možné zobrazit pro zpětnou kontrolu a také dodatečně editovat. K tomuto by však z hlediska účetnictví nemělo docházet, po uzavření účetního období toto není možné vůbec.

F6 Export dat

Dalším požadavkem na vytvořenou aplikaci je export obvyklých účetních sestav (kniha faktur, podklady pro DPH) i jednotlivých faktur do formátu XML.

1.1.2 Nefunkční požadavky

N1 Webová aplikace

Vytvořená aplikace je dostupná přes web s podporou internetových prohlížečů Safari (verze 8.0.0 a vyšší), Google Chrome (verze 40.0 a vyšší) a Mozilla Firefox (verze 35.0 a vyšší).

N2 Validace dat

Zadávaná data jsou kontrolována po odeslání formuláře. V případě chybného zadání některé položky je uživatel vyzván k opravě. V opačném případě jsou data uložena.

N3 Uložení dat

Data zadaná uživatelem jsou ukládána do databáze tak, aby byla pomocí aplikace odkudkoli přístupná.

N4 Intuitivní ovládání

Grafické rozhraní aplikace je navrženo co nejjednodušeji, aby její použití bylo přívětivé i pro méně zkušené uživatele.

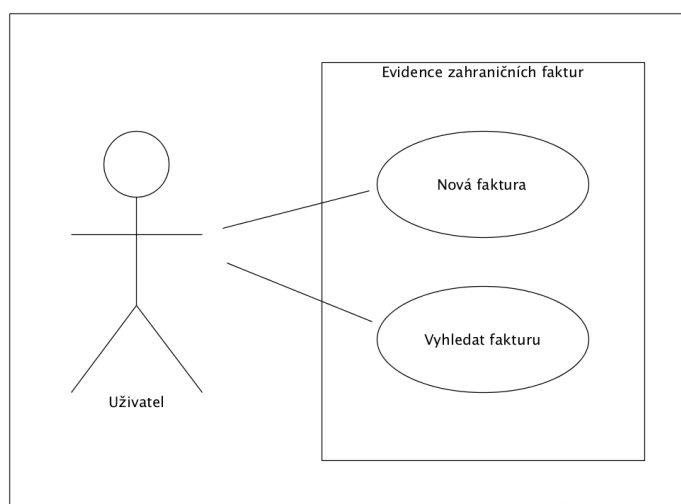
1.2 Případy užití

Model případů užití popisuje funkce aplikace z pohledu uživatele. Vychází především z výše uvedených funkčních požadavků, které detailně vysvětluje. Dále může popisovat jednotlivé uživatelské role. Tento popis není nutné v tomto

konkrétním případě provádět, neboť návrh počítá pouze s jedním typem uživatele.

1.2.1 Základní možnosti uživatele

Po spuštění aplikace má uživatel možnost provést úkony, které ilustruje diagram případů užití na obrázku 1.1.



Obrázek 1.1: Diagram případů užití pro základní úkony

1.2.2 Nová faktura

Uživatel zadá do formuláře požadované údaje, které jsou v případě kladného výsledku ověření správnosti uloženy do databáze.

Hlavní scénář

1. Případ užití začíná, jestliže se uživatel rozhodne přidat do evidence novou fakturu.
2. Systém zobrazí formulář umožňující zadat údaje o dodavateli a odběrateli, variabilní symbol faktury, peněžní částku, množství zboží nebo služeb a datum vyhotovení a splatnosti. Identifikační číslo je generováno automaticky.
3. Systém zkontroluje správnost zadaných dat. Při chybném zadání některé z položek je uživatel vyzván k opravě. Pokud nejsou všechna data správně zadána, při kliknutí na tlačítko *Uložit* je zobrazena chybová hláška.

4. Pokud je formulář vyplněn správně, systém uloží informace o faktuře.
5. Uživatel má dále možnost exportovat právě zadanou fakturu do formátu XML.

1.2.3 Vyhledávání a zobrazení faktury

V této verzi aplikace je možné vyhledávat faktury podle jejich variabilního symbolu (tj. čísla faktury), případně filtrovat podle data (od - do) nebo podle dodavatele. Výsledkem prvního procesu je jedna konkrétní faktura (pokud byla nalezena), dalšími dvěma procesy lze získat více faktur (pokud existují).

Hlavní scénář: Vyhledávání podle VS

1. Příklad užití začíná, jestliže se uživatel rozhodne vyhledat fakturu podle VS.
2. Systém zobrazí formulář, do kterého uživatel zadá VS.
3. V případě, že odpovídající faktura existuje, je zobrazena uživateli. Protože VS je jedinečný, zobrazí se uživateli přímo detail faktury. V opačném případě se po kliknutí na tlačítko *Hledat* objeví informační hláška.

Alternativní scénář 1: Vyhledávání v časovém rozmezí

1. Příklad užití začíná, jestliže se uživatel rozhodne vyhledat fakturu v určitém časovém rozmezí.
2. Systém zobrazí formulář umožňující zadat začátek a konec časového období.
3. Po kliknutí na tlačítko *Vyhledat* se zobrazí seznam faktur pro zadané období. Jestliže zadanému období neodpovídá žádná faktura, objeví se informační hláška.
4. Po kliknutí na tlačítko *Náhled* si uživatel může prohlédnout detail vybrané faktury.

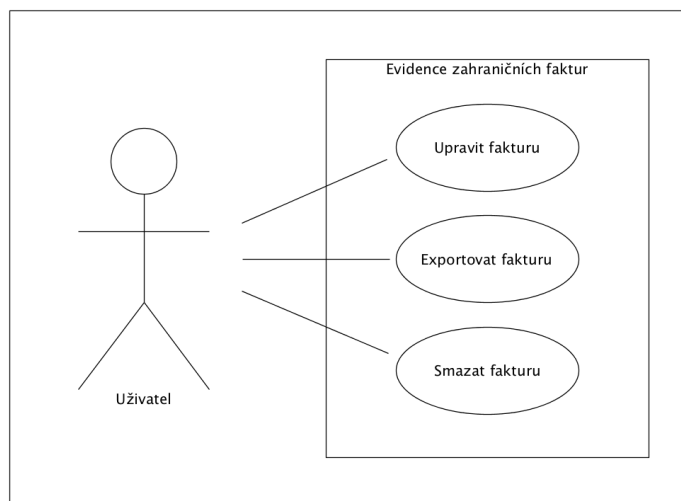
Alternativní scénář 2: Vyhledávání podle dodavatele

1. Příklad užití začíná, jestliže se uživatel rozhodne vyhledat fakturu podle jména dodavatele.
2. Systém zobrazí formulář, kde si uživatel vybere z nabídky konkrétního dodavatele.
3. Pokud zadanému dodavateli odpovídají nějaké faktury, zobrazí se v seznamu po kliknutí na tlačítko *Vyhledat*.

1. ANALÝZA

4. Po kliknutí na tlačítko *Náhled* si uživatel může prohlédnout detail vybrané faktury.

Po nalezení dané faktury a zobrazení jejího detailu může uživatel provést úkony ilustrované v diagramu případů užití na obrázku 1.2.



Obrázek 1.2: Diagram případů užití pro práci s fakturami

1.2.4 Úprava (smazání) faktury

Aplikace umožňuje uživateli v případě potřeby změnit údaje ve faktuře, případně celý záznam smazat. To je však možné pouze do doby před podáním přiznání k DPH nebo uzavřením účetního období.

Hlavní scénář

1. Příklad užití začíná, jestliže se uživatel rozhodne upravit údaje některé z již evidovaných faktur.
2. Uživatel vyhledá fakturu podle jednoho ze scénářů, které jsou popsány v předchozím bodě.
3. Systém zobrazí stejný formulář jako při zadávání nové faktury s tím rozdílem, že v tomto případě jsou údaje již vyplněné.
4. Uživatel opraví vybrané položky a systém zkontroluje správnost zadaných dat. Při chybném zadání některé z položek je uživatel vyzván k opravě. Pokud nejsou všechna data správně zadána, při kliknutí na tlačítko *Uložit* nedojde k uložení editovaných dat do databáze.

5. Pokud je formulář vyplněn správně, systém aktualizuje informace o faktuře.

1.2.5 Export účetních sestav

Evidované faktury lze buď pouze zobrazit nebo také exportovat do formátu XML a využít k dalším výpočtům, případně k „papírové“ evidenci.

Hlavní scénář

1. Případ užití začíná, jestliže se uživatel rozhodne zobrazit, případně exportovat určitou fakturu.
2. Uživatel vyhledá fakturu podle jednoho ze scénářů, které jsou popsány v jednom z předchozích bodů.
3. V zobrazení detailu faktury uživatel klikne na tlačítko *Export*.
4. Vygenerovaný soubor se otevře v novém okně prohlížeče, kde si jej uživatel může stáhnout do počítače.

1.3 Datový model

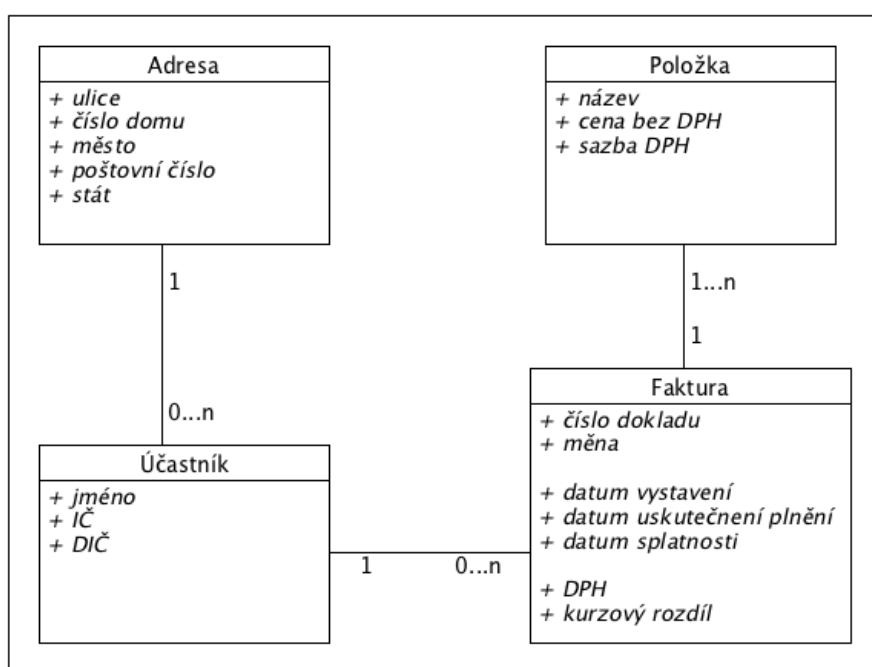
Objekty reálného světa jsou v datových modelech reprezentovány entitami (např. faktura). Každá entita má své atributy, které slouží k její specifikaci. Datový model popisuje vztahy mezi jednotlivými entitami, umožňuje lepší orientaci ve funkčních požadavcích, ze kterých vychází, a je navržen obecně tak, aby z něj bylo možné vycházet ve fázi návrhu (tzn. nezávisle na platformě).

V této podkapitole budou funkční požadavky převedeny do konceptuálního modelu, který je ilustrovaný na obrázku 1.3.

Základem tohoto modelu je *Faktura*, která má atributy kopírující některé z náležitostí uvedených dříve v této práci. Každá faktura má alespoň jednu položku. Více položek se může lišit cenou i výší sazby DPH. Oba tyto atributy jsou použity při výpočtu DPH.

Další důležitou součástí faktury jsou subjekty, tedy *Dodavatel* a *Odběratel*. Každá faktura musí mít dodavatele, odběratel být uveden nemusí (proto vazba 1:N). Zde jsou klíčové údaje o tom, zda se jedná o plátce či neplátce DPH (existence nepovinných atributů *IČ* a *DIČ*) a státní příslušnost (atribut *stát* v entitě *Adresa* značí, zda se jedná o fakturu v rámci EU nebo mimo ni). Na jedné adrese přirozeně může sídlit více firem, kdežto firma sídlí pouze na jedné adrese, proto vazba 1:N.

1. ANALÝZA



Obrázek 1.3: Konceptuální model

Existující řešení

Dostupný účetní software je možné pro účely této práce rozdělit do dvou skupin. Za první takovou skupinu lze považovat programy umožňující komplexní vedení účetnictví (nebo daňové evidence), tj. včetně evidenci faktur, za druhou pak aplikace zaměřené pouze na evidenci faktur. Tato kapitola přináší zhodnocení kladů a záporů produktů v současné době dostupných na trhu.

2.1 Desktopové aplikace

První a největší skupinou jsou aplikace desktopové. Za jejich přednosti lze považovat především komplexnost (ve většině případů se jedná o software pro vedení účetnictví) a bezpečnost lokálně uložených dat.

Ve výčtu nevýhod je potřeba zmínit především nezbytnost instalace takové aplikace na konkrétní zařízení a z toho plynoucí závislost na platformě, dále nedostupnost dat „zvenčí“, tj. online (zde se nabízí polemika o výhodnosti či nevýhodnosti vzhledem k již zmíněnému zabezpečení dat). Často se lze setkat s komplikovaným nastavením a ovládáním aplikace či celkovou nepřehledností zejména pro nové uživatele.

Faktem však je, že mnoho uživatelů si přivyklo používat tyto aplikace a z přechodu na modernější rozhraní a technologie mají přirozeně obavy.

Tuto kategorii zastupují např. Ekonomický systém POHODA [3], Stereo - účetní software [4] nebo Info Office MMI [5].

2.2 Webové aplikace

Do druhé skupiny se řadí webové aplikace. V porovnání s desktopovými aplikacemi mají tyto podstatně jednodušší GUI, ale ve většině případů umožňují pouze evidenci faktur. To může pro některé uživatele představovat zásadní nedostatek.

2. EXISTUJÍCÍ ŘEŠENÍ

Výhodu těchto aplikací lze spatřovat zejména v dostupnosti online odkudkoli, jednoduchosti grafického rozhraní a ovládání či nezávislosti na platformě. Zde je však potřeba věnovat zvýšenou pozornost zabezpečení uložených dat, která v tomto případě nejsou uchována lokálně, nýbrž vzdáleně na serveru.

Mezi zástupce webových aplikací patří FlexiBee (účetnictví) [6], SuperFaktura (pouze evidence faktur) [7] nebo Qfaktury (pouze evidence faktur) [8].

2.3 Mobilní aplikace

Pravděpodobně nejméně rozvinutou skupinou jsou aplikace mobilní. Jejich vlastnosti jsou v podstatě shodné s vlastnostmi webových aplikací s tím rozdílem, že tyto závisí na platformě (iOS, Android, Windows Phone) a je nutná jejich instalace do mobilního zařízení.

Jediným dostupným zástupcem tohoto typu je aplikace iDoklad [9], která umožňuje pouze evidenci faktur.

2.4 Shrnutí

Na závěr této kapitoly je vhodné doplnit, že žádné ze zmíněných řešení se nezabývá přímo evidencí zahraničních faktur a náležitostí s tím spojených. Proto je cílem této práce navrhnout a implementovat takovou aplikaci, která těmto speciálním požadavkům vyhoví a zároveň splní předpoklad online dostupnosti odkudkoli.

Návrh

Návrh je další nedílnou součástí vývoje software. Vychází z analýzy a v tomto případě se skládá z diagramu tříd, návrhu uživatelského rozhraní a volby technologií. Dále je třeba navrhnout způsob napojení na portál ČNB a export účetních sestav.

3.1 Volba technologií

Z nefunkčních požadavků vyplývá, že vyvíjená aplikace má být dostupná pomocí webového prohlížeče. Případaly proto v úvahu např. jazyky ASP.NET, Java nebo PHP [10]. Vzhledem k vlastním zkušenostem jsem zvolila poslední jmenovaný. Jedná se o skriptovací, objektově orientovaný jazyk určený především k vývoji webových stránek. Nabízí širokou škálu použití, zejména v kombinaci se značkovacím jazykem HTML [11] a kaskádovými styly CSS [12]. Spojení těchto technologií splní požadavek na jednoduchou a snadno použitelnou aplikaci.

Z analýzy požadavků dále vyplývá nutnost použití databáze pro uchování dat. Protože nejde o ohromné množství dat ani operací prováděných nad databází, rozhodla jsem se využít databázový server mySQL [13], který se ve spojení s webovými aplikacemi běžně používá.

Při výběru PHP frameworku hrála roli především kvalitní dokumentace, dostupné tutoriály a opět osobní zkušenost, proto jsem dala přednost Symfony2 [14] před Nette [15]. Mezi hlavní výhody Symfony2 patří velké množství nástrojů a tříd, které zjednodušují vývoj webových aplikací.

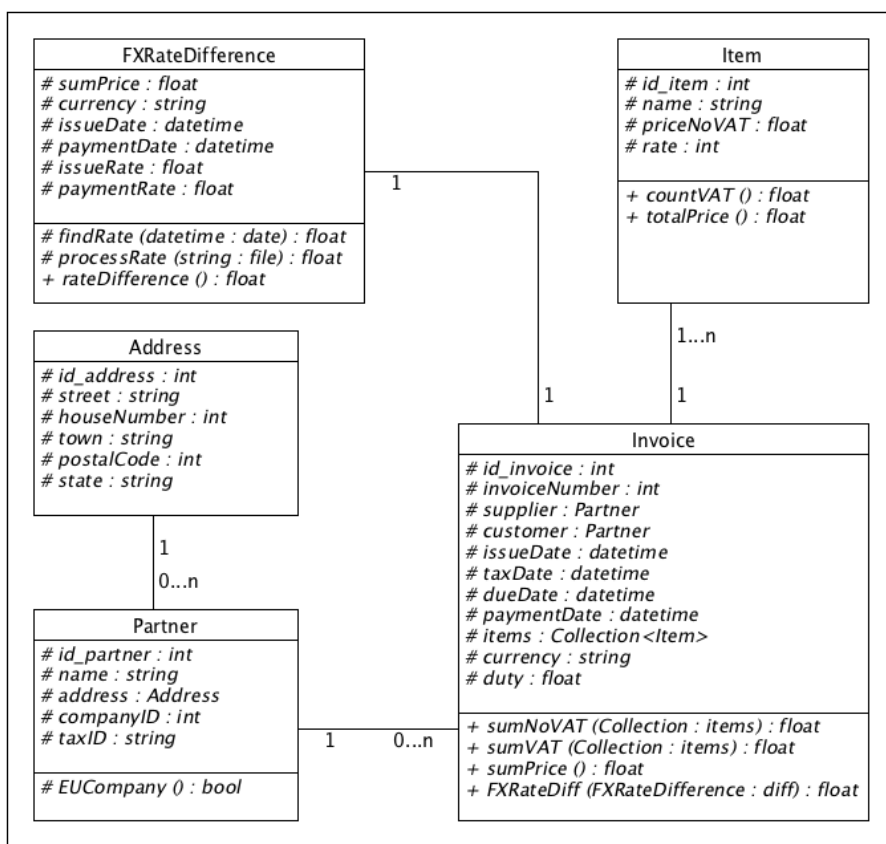
Užitečnou součástí Symfony2 je například Doctrine [16], která zjednodušuje práci s databází, nebo šablonovací systém Twig [17], který se používá pro vykreslení uživatelského rozhraní. Framework dále nabízí použitelné ladící nástroje a umožňuje práci na aplikaci ve vývojovém i produkčním prostředí.

Pro implementaci byla využita aktuální verze Symfony 2.6.

3.2 Diagram tříd

Diagram tříd má předlohu v konceptuálním modelu. Oproti němu je však kompletní (nesmí chybět žádné třídy, atributy či metody) a závislý na platformě.

Z diagramu tříd na obrázku 3.1 přímo vychází databázový model, protože navržené třídy využívají Doctrine a ORM [18] k mapování databáze, a také následná implementace. V diagramu jsou pro lepší přehlednost záměrně vynechány konstruktory a metody *Get* a *Set*.



Obrázek 3.1: Diagram tříd

Kromě třídy *FXRateDifference* odpovídají třídy databázovým entitám. Tyto tzv. datové třídy jsou popsány na následujících řádcích.

3.2.1 Faktury

Třída *Invoice* tvoří základ celého návrhu. Umožňuje vytváření a editaci faktur, využívá se při jejich vyhledávání a zobrazování detailů. Dále je nezbytná

pro export účetních sestav a podkladů pro DPH a také pro výpočty uvedené v analýze požadavků. Obsahuje následující atributy:

id_invoice: primární klíč, automaticky generovaná hodnota.

invoiceNumber: variabilní symbol, odpovídá číslu dokladu uvedenému na faktuře.

supplier: dodavatel, instance třídy *Partner*.

customer: odběratel, instance třídy *Partner*.

issueDate: datum vystavení faktury.

taxDate: datum uskutečnění zdanitelného plnění.

dueDate: datum splatnosti faktury.

paymentDate: datum skutečného zaplacení faktury, výchozí hodnota NULL (tzn. faktura ještě nebyla zaplacená).

items: položky faktury, kolekce instancí třídy *Item*.

currency: měna, 3-znaková zkratka (např. CZK).

duty: clo (pokud je relevantní), nepovinný atribut.

Z těchto atributů je možné vypočítat následující údaje:

sumNoVAT: celková cena za všechny položky na faktuře bez DPH.

sumVAT: celková výše DPH.

sumPrice: celková cena za všechny položky na faktuře včetně DPH.

FXRateDiff: kurzový rozdíl (zisk nebo ztráta), vizte 3.3.

3.2.2 Položky

Třída *Item* představuje položky, které jsou nedílnou součástí každé faktury. Je nutné ji realizovat odděleně od faktury zejména proto, že každá položka (i v rámci jedné faktury) může mít sazbu DPH. Má tyto atributy:

id_item: primární klíč, automaticky generovaná hodnota.

name: název položky.

priceNoVAT: cena položky bez DPH.

rate: odpovídající sazba DPH pro danou položku (10%, 15% nebo 21%).

Z těchto atributů lze vypočítat následující:

countVAT: výše DPH.

totalPrice: cena položky včetně DPH.

3.2.3 Subjekty

Třída *Partner* realizuje subjekty faktury - dodavatele a odběratele. Je důležitá hlavně pro rozlišení dodavatelů z EU a mimo EU zejména z důvodu nutnosti zadání dopočetného cla. Obsahuje následující atributy:

id_partner: primární klíč, automaticky generovaná hodnota.

name: jméno subjektu.

address: sídlo subjektu (adresa).

3. NÁVRH

companyID: IČ - unikátní osmimístné číslo právnické nebo podnikající fyzické osoby.

taxID: DIČ - jednoznačná identifikace plátce DPH, první dva znaky tvoří zkratku země, následuje samotné číslo plátce daně.

Z těchto atributů je možné vypočítat následující údaje:

EUCompany: rozlišení, zda se jedná o subjekt ze země EU nebo mimo ni.

3.2.4 Adresy

Třída *Address* představuje spíše doplnění entity *Partner*. Z hlediska přehlednosti a údržby databáze je výhodnější evidovat adresy mimo instance subjektů. Má tyto atributy:

id_address: primární klíč, automaticky generovaná hodnota.

street: název ulice.

houseNumber: číslo domu.

town: název města.

postalCode: poštovní směrovací číslo.

state: název státu.

3.3 Napojení na portál ČNB

Jedním ze základních požadavků zadavatele je automatické stahování kurzů měn z portálu ČNB a následný výpočet kurzových rozdílů. Toto realizuje třída *FXRateDifference*, která má následující proměnné a metody (oproti popsaným v předchozí podkapitole se nejedná o datové třídy):

sumPrice: celková cena zboží na faktuře, hodnota předána v konstruktoru.

currency: měna faktury - nezbytná pro vyhledání kurzu, hodnota předána v konstruktoru.

issueDate: datum vystavení faktury - nezbytné pro vyhledání kurzu, hodnota předána v konstruktoru.

paymentDate: datum zaplacení faktury - nezbytné pro vyhledání kurzu, hodnota předána v konstruktoru.

Z těchto atributů lze za použití uvedených metod vypočítat následující:

issueRate: kurz pro zadanou měnu v dané datum vystavení faktury, hodnota dopočítána metodou *findRate*.

paymentRate: kurz pro zadanou měnu v dané datum zaplacení faktury, hodnota dopočítána metodou *findRate*.

findRate: metoda starající se o stažení souboru z portálu ČNB pro zadané datum, vrací nalezený kurz dané měny.

processRate: metoda zpracovává soubor stažený pomocí předchozí metody, vrací nalezený kurz dané měny.

rateDifference: metoda počítá a vrací kurzový rozdíl.

3.4 Export účetních sestav

Protože je export do formátu XML realizován v rámci práce s instancemi třídy *Invoice*, je detailnější popis k dispozici v kapitole 4.

3.5 Návrh uživatelského rozhraní

Při tvorbě webové aplikace je důležité navrhnout přehledné uživatelské rozhraní a intuitivní ovládání. Tento návrh se vyplatí udělat ještě před samotnou implementací, protože zamezí nelogickému uspořádání prvků a případnému přepracovávání. K tomuto účelu je použita metoda wireframe [19].

Jedná se o schématický model rozložení prvků na stránce, ze kterého následně vychází grafická podoba webové aplikace. Skládá se ze tří hlavních částí - hlavičky, těla a patičky.

3.5.1 Hlavička

Hlavička (ang. header) obsahuje v levé části název aplikace, který je zároveň odkazem na domovskou stránku. V pravé části se nachází stručný popis aplikace.

3.5.2 Tělo

Tělo neboli obsah (ang. content) se mění v závislosti na úkonech, které uživatel provádí. Toto nejlépe ilustrují následující obrázky.

V levém horním rohu se nachází název stránky, následovaný příslušným obsahem. V případě obrázku 3.2 jsou to tlačítka **Nová faktura** a **Vyhledat faktury**, které představují základní funkcionalitu aplikace.

Pokud je záměrem uživatele přidat novou fakturu, dojde ke zobrazení formuláře podle obrázku 3.3.

V případě potřeby je samozřejmě možné přidat nový subjekt pomocí formuláře na obrázku 3.4. Položky IČ a DIČ jsou nepovinné, jak je vysvětleno v kapitole 1.

Následuje zobrazení formuláře pro přidání položek faktury, jak je ilustrováno na obrázku 3.5.

Po úspěšné validaci dat následuje náhled detailu právě zadané faktury s již dopočítanými hodnotami (aktuální kurzy měn, cena jednotlivých položek včetně DPH, ceny v CZK a kurzový rozdíl) tak, jak ilustruje obrázek 3.6. V tento moment je možné zadanou fakturu upravit pomocí formuláře, který se velmi podobá formuláři pro novou fakturu a následně znovu uložit.

Pokud si uživatel přeje najít či exportovat faktury, použije na domovské stránce tlačítko **Vyhledat faktury**. Následně se rozhodne pro způsob, jakým chce faktury vyhledat, podle obrázku 3.7. Formuláře pro jednotlivé způsoby vyhledávání jsou velmi jednoduché, proto zde nejsou uvedeny.

3. NÁVRH

FOR-INV Evidence zahraničních faktur

Domovská stránka

Nová faktura

Vyhledat faktury

Kontakt
© České vysoké učení technické v Praze

Obrázek 3.2: Domovská stránka

FOR-INV Evidence zahraničních faktur

Nová faktura

Číslo faktury

Dodavatel

Odběratel

Datum vystavení

Datum uskutečnění plnění

Datum splatnosti

Datum zaplacení

Měna

Cio

Přidat subjekt

Přidat položky Zpět

Kontakt
© České vysoké učení technické v Praze

Obrázek 3.3: Formulář pro zadání nové faktury

V případě kladného výsledku vyhledávání dojde ke zobrazení seznamu faktur tak, jak ilustruje obrázek 3.8. V tuto chvíli je možné výsledek vyhledávání exportovat do XML. V opačném případě se uživatel dozví, že podle zadaných parametrů nebylo nic nalezeno.

3.5.3 Patička

Patička (ang. footer) umístěná v dolní části stránky nabízí informaci o autor-
ských právech a kontakt na správce webu.

FOR-INV Evidence zahraničních faktur

Nový subjekt

Název

Adresa

IČO

DIČ

Kontakt
© České vysoké učení technické v Praze

Obrázek 3.4: Formulář pro přidání nového subjektu

FOR-INV Evidence zahraničních faktur

Přidat položky

Název

Cena

Sazba DPH

Název

Cena

Sazba DPH

Kontakt
© České vysoké učení technické v Praze

Obrázek 3.5: Formulář pro přidání položek faktury

3. NÁVRH

| FOR-INV | | Evidence zahraničních faktur | | | |
|---|---------------------------|------------------------------|------------|------------|--|
| Náhled faktury | | | | | |
| Číslo faktury | 12345689 | Položky | | | |
| Dodavatel | Lorem s.r.o. | Tričko | 5.39 | 21% | 6.52 EUR |
| Odběratel | Ipsium s.r.o. | | | | |
| Datum vystavení | 2. 1. 2015 (Kurz: 27.70) | Cena celkem (bez DPH) | 149.30 CZK | | |
| Datum uskutečnění plnění | 2. 1. 2015 | DPH celkem | 31.35 CZK | | |
| Datum splatnosti | 2. 2. 2015 | Cena celkem (s DPH) | 180.60 CZK | 184.45 CZK | |
| Datum zaplacení | 12. 1. 2015 (Kurz: 28.29) | Kurzový rozdíl | 3.85 CZK | | |
| Měna | EUR | | | | |
| Clo | - | | | | |
| | | | | | Upravit Zpět |
| Kontakt © České vysoké učení technické v Praze | | | | | |

Obrázek 3.6: Náhled detailu faktury

| FOR-INV | | Evidence zahraničních faktur | | | |
|---|--|------------------------------|--|--|----------------------|
| Vyhledat fakturu | | | | | |
| Podle čísla faktury | | | | | |
| V časovém rozmezí | | | | | |
| Podle dodavatele | | | | | |
| | | | | | Zpět |
| Kontakt © České vysoké učení technické v Praze | | | | | |

Obrázek 3.7: Možnosti hledání faktur

| FOR-INV | | Evidence zahraničních faktur | | | | |
|---|--------------|------------------------------|-------------|-----------|--------------|---------------------------------------|
| Výsledek vyhledávání | | | | | | |
| Číslo faktury | Dodavatel | Odběratel | Vystaveno | Částka | Částka (CZK) | |
| 123456789 | Lorem s.r.o. | Ipsum s.r.o. | 2. 1. 2015 | 6.52 EUR | 180.60 | <input type="button" value="Náhled"/> |
| 987654321 | Lorem s.r.o. | Dolor a.s. | 12. 3. 2015 | 56.92 USD | 1464.55 | <input type="button" value="Náhled"/> |
| <input type="button" value="XML"/> | | | | | | |
| Kontakt © České vysoké učení technické v Praze | | | | | | |

Obrázek 3.8: Seznam nalezených faktur

Realizace

V této kapitole jsou popsána fakta spojená s realizací webové aplikace navržené v předchozích částech práce. Jak již bylo řečeno, cílem bylo navrhnout takovou aplikaci, která splňuje všechny zadané požadavky, zejména automatické stahování aktuálních kurzů měn a export účetních sestav.

Implementace byla realizována podle návrhu v předchozí kapitole.

4.1 Použité nástroje

Celý proces návrhu a implementace samozřejmě nelze zvládnout pouze s pomocí tužky a papíru. V následujících odstavcích jsou tedy popsány použité nástroje.

4.1.1 UMLet

Umllet [20] je open-source nástroj pro tvorbu různých druhů diagramů. Je multiplatformní a mezi jeho další přednosti patří jednoduché uživatelské rozhraní a intuitivní ovládání.

V této práci je využit k vytvoření datového modelu a diagramu tříd.

4.1.2 Pencil

Pencil [21] je opět open-source nástroj sloužící jak k tvorbě diagramů, tak k vytváření wireframů. Jedná se opět o multiplatformní software, který se vyznačuje svou jednoduchostí a přívětivým ovládáním.

S jeho pomocí jsou vytvořeny veškeré wireframy.

4.1.3 PhpStorm

PhpStorm [22] je komerční vývojové prostředí pro jazyk PHP, mezi jehož hlavní přednosti patří podpora frameworku Symfony2. Tento software není dostupný zdarma, a proto pro jeho používání byla využita školní licence.

4.2 Implementace

Proces implementace lze rozdělit do tří úzce propojených částí - backendu („pozadí aplikace“), frontendu („to, co vidí uživatel“) a databáze.

4.2.1 Backend

Backend část aplikace je v Symfony realizována pomocí tříd zvaných *Controller*, které jsou potomky třídy *Controller*. Tyto třídy ovládají chod celé aplikace s využitím tzv. *Action* metod, které jsou přístupné pomocí cest (tzv. *Route*). Mimo jiné se tyto metody starají o volání šablon, které vykreslují obsah konkrétních stránek.

Dále třídy mohou obsahovat podpůrné metody využívané především ke zpřehlednění kódu. Tyto třídy jsou typicky uloženy ve složce *Controller*.

Další částí backendu aplikace jsou formuláře a jejich zpracování. Formuláře jsou obvykle realizovány jako potomci třídy *AbstractType* a jsou uloženy ve složce *Form*. Zpracování formulářů probíhá v již zmíněných třídách typu *Controller*.

Jako příklad z aplikace vyvinuté v rámci této práce lze uvést například *InvoiceController*, který se stará o správné chování aplikace při práci s fakturami a její metody *newAction* (volá se při vytváření nové faktury) nebo *findAction* (stará se o vyhledávání faktur). Jako příklad třídy formuláře lze uvést například *InvoiceForm*, která realizuje formulář použitý při vytváření nebo editaci faktury.

4.2.2 Frontend

O frontend část aplikace se stará především šablonovací systém TWIG [17]. Šablony jsou volány z *Controller*-ů a starají se o vykreslování obsahu stránek. Typicky je implementována základní šablona, kterou ostatní šablony rozšiřují. V tomto konkrétním případě se jedná o základní šablonu *layout.html.twig* a její potomky *new.html.twig* nebo *find.html.twig*. Šablony jsou rozděleny do složek podle příslušnosti ke *Controller*-ům.

Dále je využit značkovací jazyk HTML, kaskádové styly CSS a jazyk Javascript.

4.2.3 Databáze

Pro jednoduché vytvoření databáze jsou využity možnosti nabízené přímo frameworkem Symfony, které spočívají v přímém vygenerování databáze z datových tříd opatřených anotacemi.

Toto umožní práci přímo s obsahem databáze. Je možné objekty jak vyhledávat, tak upravovat a aktualizovat přímo v kódu programu.

4.3 Validace dat

Pro ověřování správnosti vkládaných dat bylo nutné předem určit pravidla, podle kterých se bude validace řídit.

O jednoduchou validaci (tj. ověření formátu vkládaných dat) se starají anotace uvedené u jednotlivých proměnných v datových třídách a vhodně implementované třídy formulářů.

Pro ověření správnosti dat vyplývající z první kapitoly bylo nutné vytvořit specifická pravidla, aby nedocházelo k ukládání irelevantních údajů. Jako příklad takového pravidla lze uvést například fakt, že datum splatnosti faktury nemůže být dříve než datum vystavení nebo že dodavatel nesmí být ten stejný subjekt jako odběratel.

4.4 Export

Z funkčních požadavků vyplývá nutnost implementace exportu účetních sestav. Toto je realizováno vytvořením *Controller*-u, který volá vykreslení TWIG šablony. Tato šablona se postará o správné vytvoření XML souboru v novém okně prohlížeče.

Tento postup je použit jak k vytvoření XML detailu faktury, tak k vytvoření XML souboru vycházejícího z výsledku vyhledávání faktur (účetní sestava).

Testování

Vzhledem k tomu, že se jedná o webovou aplikaci, byly pro testování použity dva rozdílné webové prohlížeče - Safari verze 8.0.5, Google Chrome verze 42.0.2311 a Mozilla Firefox verze 27.0.1 (všechny na operačním systému OSX).

5.1 Testovací scénáře

Jednou z možností testování aplikací je použití testovacích scénářů. Jejich výhoda spočívá v tom, že simulují reálné použití aplikace a tím pomáhají odhalit případné chyby či nedostatky.

5.1.1 Vytvoření nové faktury

Tento scénář testuje postup vytvoření nové faktury s případě, že v databázi již existují oba subjekty. V rámci testování tohoto scénáře byly odhaleny nedostatky při validaci dat, kdy bylo potřeba zajistit ověření správnosti pro relevantní kontrolu kalendářních dat. Vytvoření položek faktury proběhlo již bez problémů.

5.1.2 Vytvoření nového subjektu

Tento scénář testuje postup vytvoření nové faktury s případě, že v databázi některý ze subjektů neexistuje a je tedy potřeba jej vytvořit. To je možné provést přímo z formuláře pro vytvoření faktury. Poté, co je subjekt úspěšně vytvořen, se zobrazí v příslušné nabídce ve formuláři pro vytváření nové faktury. Testování tohoto scénáře proběhlo bez problémů.

5.1.3 Vyhledávání faktur

Tento scénář je možné rozdělit do tří částí. V první z nich proběhlo testování vyhledávání podle čísla faktury. Po zadání čísla faktury došlo buď k zobra-

zení detailu faktury nebo k oznámení, že faktura zadaného čísla neexistuje, což je správný průchod scénářem.

Při testování vyhledávání v časovém rozmezí byla odhalena potřeba upravit formulář do stávající podoby, tj. na výběr jednotlivých parametrů (den, měsíc, rok) z nabídky, místo zadávání dat ručně, protože převádění textového řetězce na odpovídající objekt neproběhlo vždy úspěšně. Po této drobné úpravě již nenastal žádný další problém.

Podobně tomu bylo i u testování vyhledávání podle dodavatele. Místo původního zadávání textového řetězce byl formulář upraven tak, aby bylo možné konkrétního dodavatele vybrat z nabídky existujících subjektů v databázi. Tímto odpadá nutnost ověřování, zda zadaný dodavatel existuje nebo ne. Další nesrovnalosti nebyly odhaleny.

5.1.4 Export faktur

Testování tohoto scénáře odhalilo pouze drobný nedostatek, a to fakt, že vytvořený XML soubor byl otevřen ve stávajícím okně prohlížeče a tím došlo ke zneumožnění přímého návratu zpět do aplikace. Toto bylo jednoduchou úpravou vyřešeno a další chyby již nebyly nalezeny.

5.2 Vyhodnocení

Testování odhalilo drobné chyby v aplikaci, které se podařilo úspěšně odstranit a aktuální verze aplikace je již neobsahuje. Dále došlo k ověření, že aplikace bez problémů běží v obou testovaných webových prohlížečích.

Dále bude nutné provést testy uživatelského rozhraní, které v rámci této práce nebyly provedeny.

Zhodnocení přínosů

Tato kapitola přináší shrnutí ekonomicko-manažerských přínosů vytvořeného evidenčního systému.

6.1 Náklady na vývoj a provoz aplikace

Náklady na vývoj aplikace se odvíjí zejména od časové náročnosti a finančního ohodnocení vývojářů. Vzhledem k nutnosti provést důkladnou analýzu, detailní návrh a kvalitní implementaci a testování byla časová náročnost odhadnuta na přibližně 50 man-days (česky „člověko-dny“, čas odpovídající práci jedné osoby za jeden pracovní den). Při průměrné mzdě 300 CZK za hodinu práce lze náklady na vývoj aplikace odhadnout na částku 120 000 CZK.

Pro provoz aplikace je nezbytný databázový server a zajištění technické podpory aplikace, případně její další vývoj. Nákup serveru lze jako jednorázovou investici odhadem vyčíslit na 100 000 CZK, náklady na podporu a vývoj mohou dosáhnout až 50 000 CZK měsíčně.

Protože se v současné době nejedná o plnohodnotný účetní software, ale pouze o jeho část pro evidenci zahraničních faktur, pohybuje se odhad ceny za jednu licenci mezi 7 000 a 10 000 CZK.

6.2 Přínosy aplikace pro zákazníka

Hlavním přínosem aplikace pro zákazníka je především zjednodušení práce při zadávání kurzů měn a počítání kurzových rozdílů vzhledem k tomu, že aplikace provádí stahování kurzů měn z portálu ČNB zcela automaticky. Mezi přínosy bezesporu patří také dostupnost aplikace pomocí webového prohlížeče odkudkoli, což je jeden z hlavních požadavků zadání.

Další výhodou představuje přehledné sledování příjmů a výdajů za zahraniční faktury či vývoj kurzových zisků a ztrát.

6. ZHODNOCENÍ PŘÍNOSŮ

V dalších verzích aplikace dojde k revizi stávajících funkcí systému, zjednodušení uživatelského rozhraní na základě vyhodnocení testovacího provozu u zákazníků, případně k návrhu a implementaci dalších funkcionalit na základě požadavků zákazníků.

Závěr

Cílem této práce bylo analyzovat, navrhnout a implementovat evidenční systém zahraničních faktur.

Nejprve jsem provedla analýzu vzhledem k zákonným požadavkům podle [1] a [2] na evidenci zahraničních faktur. Dále jsem zanalyzovala další funkční a nefunkční požadavky podle zadání práce. Následně jsem provedla porovnání již dostupných řešení.

Na základě analýzy jsem navrhla strukturu aplikace, její datový model a také napojení na portál ČNB pro automatické stahování kurzů měn. Následně jsem provedla výběr technologií, které jsem použila při implementaci aplikace. Samotná realizace pro mě byla výzvou a velkým přínosem, protože jsem se do té doby setkala s tvorbou webových aplikací pouze v rámci výuky ve škole.

Výsledkem této bakalářské práce je tedy funkční webová aplikace evidenčního systému zahraničních faktur, která podle výsledku testování vyhovuje všem bodům zadání práce.

Literatura

- [1] Úplné znění zákona č. 563/1991 Sb., o účetnictví.
- [2] Zákon č. 235/2004 Sb., o dani z přidané hodnoty.
- [3] STORMWARE, s.r.o.: POHODA - Účetní program [online]. 2014, [cit. 2015-03-12]. Dostupné z: <http://www.stormware.cz/pohoda>
- [4] KASTNER software, s.r.o.: Stereo - ekonomický software [online]. 2015, [cit. 2015-03-12]. Dostupné z: <http://www.kastnersw.cz/stereo>
- [5] Info Office, s.r.o.: Info Office s.r.o [online]. 2014, [cit. 2015-03-12]. Dostupné z: <http://www.info-office.cz>
- [6] Ing. Jan Bláha: FlexiBee - ekonomický systém a účetnictví online [online]. 2015, [cit. 2015-03-12]. Dostupné z: <https://www.flexibee.info>
- [7] superfaktura.cz, s.r.o.: Faktury online pro živnostníky a malé firmy [online]. 2015, [cit. 2015-03-12]. Dostupné z: <http://www.superfaktura.cz>
- [8] Ing. Jan Daniel: Online vytváření a správa faktur - Qfaktury [online]. 2011, [cit. 2015-03-12]. Dostupné z: <http://www.qfaktury.cz>
- [9] Cígler Software, a.s.: Mobilní fakturace pro iOS, Android a Windows Phone | iDoklad [online]. 2015, [cit. 2015-03-12]. Dostupné z: <https://www.idoklad.cz/mobilni-aplikace>
- [10] Philip Olson: PHP: PHP Manual - Manual [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://php.net/manual/en/>
- [11] HTML Wikipedia, the free encyclopedia [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://en.wikipedia.org/wiki/HTML>
- [12] Cascading Style Sheets - Wikipedia, the free encyclopedia [online]. 2015, [cit. 2015-04-12]. Dostupné z: http://en.wikipedia.org/wiki/Cascading_Style_Sheets

- [13] Oracle Corporation: MySQL :: MySQL 5.6 Reference Manual [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://dev.mysql.com/doc/refman/5.6/en/index.html>
- [14] SensioLabs: Learn Symfony (2.6) [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://symfony.com/doc/current/index.html>
- [15] Nette Foundation: Dokumentace | Nette Framework [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://doc.nette.org/cs/2.3/>
- [16] Jonathan Wage: Doctrine - Doctrine 1.2.4 documentation [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://doctrine.readthedocs.org/en/latest/>
- [17] SensioLabs: Documentation - Twig - The flexible, fast, and secure PHP template engine [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://twig.sensiolabs.org/documentation>
- [18] Object-relational mapping - Wikipedia, the free encyclopedia [online]. 2015, [cit. 2015-04-12]. Dostupné z: http://en.wikipedia.org/wiki/Object-relational_mapping
- [19] Website wireframe - Wikipedia, the free encyclopedia [online]. 2015, [cit. 2015-04-12]. Dostupné z: http://en.wikipedia.org/wiki/Website_wireframe
- [20] UMLet - Free UML Tool for Fast UML Diagrams [online]. 2015, [cit. 2015-04-12]. Dostupné z: <http://www.umlet.com>
- [21] Evolus: Home - Pencil Project [online]. 2012, [cit. 2015-04-12]. Dostupné z: <http://pencil.evolus.vn>
- [22] JetBrains s.r.o.: PHP IDE :: JetBrains PhpStorm [online]. 2015, [cit. 2015-04-12]. Dostupné z: <https://www.jetbrains.com/phpstorm/>

Seznam použitých zkratk

CZK Česká koruna

ČNB Česká národní banka

EU Evropská unie

EUR Euro

DIČ Daňové identifikační číslo

DPH Daň z přidané hodnoty

GBP Great Britain Pound (britská libra)

GUI Graphical User Interface (grafické uživatelské rozhraní)

HTML HyperText Markup Language

IČ Identifikační číslo osoby

ID Identifikační číslo

ORM Objektově-relační mapování

PHP Hypertext Preprocessor

UML Unified Modeling Language

USD United States Dollar (americký dolar)

VS Variabilní symbol

XML Extensible Markup Language (formát souboru, „rozšiřitelný značkovací jazyk“)

Obsah přiloženého CD

| | |
|-------------------------------------|--|
| readme.txt..... | stručný popis obsahu CD |
| doc | |
| ├── for-inv_html..... | dokumentace k programu |
| └── manual.txt..... | uživatelská příručka |
| src | |
| ├── impl..... | zdrojové kódy implementace |
| └── latex..... | zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ |
| text | |
| └── BP_Musilova_Aneta_2015.pdf..... | text práce ve formátu PDF |