

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra počítačů



Bakalářská práce

E-pokladna pro malou obchodní jednotku

E-cash register for small sales unit

Studijní program: Softwarové technologie a management

Studijní obor: Softwarové inženýrství

Vedoucí práce: Ing. Božena Mannová, Ph.D.

Petr Jartym

Praha 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jartym** Jméno: **Petr** Osobní číslo: **420872**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové technologie a management**
Studijní obor: **Softwarové inženýrství**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

E-pokladna malé prodejní jednotky

Název bakalářské práce anglicky:

E-cash register for small sales unit

Pokyny pro vypracování:

Cílem práce je navrhnout a implementovat aplikaci e-pokladny pro malou prodejní jednotku. Pokladna bude schopna dohledávat zboží podle EAN kódu nebo podle jiného číselného označení zboží. Pokladna bude evidovat platbu a vygeneruje účtenku.

Aplikace bude rozdělena se serverovou a klientskou část. Serverová část bude mít uloženo v databázi položky zboží, které bude možné prodávat. Klientská část bude uživatelským rozhraním tohoto systému. Serverová a klientská část poběží výhradně na počítači klienta, pro kterého bude aplikace zprostředkována. Specifikujte a analyzujte konkrétní funkční a nefunkční požadavky na systém. Na základě těchto požadavků navrhnete aplikaci. Pro analýzu a návrh použijte vhodné prostředky SI. Navržený program implementujte a výslednou aplikaci otestujte.

Seznam doporučené literatury:

Ian Sommerville: Software Engineering, Global Edition, Pearson Higher Ed, 2016, ISBN1292096144
Arlow, J., Neustat, I.: UML 2 a unifikovaný proces vývoje aplikací. Computer Press, 2007.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D., Software Engineering and Networking FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **19.01.2018**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Božena Mannová, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 25. května 2018

.....

Podpis autora práce

Poděkování

Rád bych poděkoval vedoucí práce Ing. Boženě Mannové, Ph.D., za její čas a konzultace, které věnovala této, práci. Také bych rád poděkoval rodičům a kamarádům, kteří mě podporovali v napsání práce.

Abstrakt

Tato práce se zabývá návrhem a implementací e-pokladny pro malé obchodní jednotky.

E-pokladna je schopná evidovat tržby do systému pro EET, tisknout účtenky a spravovat tyto údaje. Svou databází položek a historii transakcí má uloženou na serveru, díky čemuž může klientská část běžet na několika počítačích nezávisle na sobě.

Aplikace je určena pro počítače s operačním systémem Microsoft Windows 7 a novější.

Klíčová slova: Java, Spring, EET, Server, Client, Pokladna, aplikace

Abstract

This thesis pursues the designing and implementation of an e-cash register for a small sales unit. The cash register is capable of registering sales to the EET, as well as printing receipts and administrating this data. Its database of items and history of transactions are stored on a server, thanks to which the client's part can run independently and simultaneously on multiple computers. This app is designated for computers with the operational system Windows 7 and higher.

Key words: Java, Spring, EET, Server, Client, Cash desk, application

Obsah

1. Úvod.....	1
1.1 Motivace – e-pokladna.....	1
2. Vize	2
1.3 Cíle práce	2
2.4 Současný stav.....	2
2.5 Budoucí stav.....	3
2.6 Zdůvodnění práce.....	3
2.7 Přínosy práce.....	3
2.8 Konkrétní výstupy práce	3
2. Analýza	4
2.1 SWOT	4
2.2 FURPS	4
Funkčnost.....	4
Vhodnost k použití	4
Spolehlivost.....	5
Výkon.....	5
Schopnost být udržována	5
2.3 Předpoklady	5
2.4 Omezení	5
2.5 Akceptační kritéria.....	5
2.6 Rizika	6
3 Business analýza	7
3.1 Sběr požadavků na práci	7
3.2 Entitní model.....	7
3.3 Business požadavky	8
Jako prodavač potřebuji sestavit nákup z položek, aby mohli být prodány zákazníkovi.....	8
3.4 Funkční požadavky	9
3.5 Role	11
Pokladní	11
Vedoucí	11
Administrátor	11
3.6 Případy užití.....	11
Scénáře případů užití.....	11
3.6 Datový model.....	13
4. Implementace	14

4.1 Volba technologií pro server	14
Java + Spring + Maven	14
H2 databáze.....	14
4.2 Volba technologií pro klienta.....	14
Java + SWING + Maven.....	14
eet-client.....	15
Unirest.....	15
4.3 Konfigurace serveru	15
4.4 Konfigurace klienta.....	16
4.5 Architektura serveru.....	16
4.6 Architektura klienta.....	17
5. Testování.....	18
5.1 Unit testy	18
5.2 Testování vývojářem.....	18
5.3 Akceptační testy	18
5.4 Uživatelské testy	18
5.4 Zhodnocení testů	18
6. Závěr	19
6.1 Zhodnocení práce.....	19
6.2 Budoucnost práce.....	19
7. Seznam citací	20

Seznam příloh

A Terminologický slovník	21
B Instalační příručka	22
C Účtenka z e-pokladny	23
D Obsah přiloženého CD	24

Seznam obrázků

Obrázek 1. Entitní model	8
Obrázek 2. Diagram případů užití.....	12
Obrázek 3. Databázový model	13
Obrázek 4. Příklad evidence do EET	16
Obrázek 5. Účtenka z e-pokladny	23

1. Úvod

V této práci byla zpracována aplikace e-pokladna. E-pokladna je určena pro malou obchodní jednotu, které poskytuje funkce jako je evidence do EET, správu položek, správu uživatelských účtů, správu prodaného zboží a evidenci provedených nákupů. Pokladna obsahuje jednoduché uživatelské rozhraní a je možné ji mít spuštěnou i na několika počítačích současně.

1.1 Motivace – e-pokladna

E-pokladnu pro malou prodejní jednotku jsem si vybral z důvodu aktuálně zvýšené poptávky po těchto systémech, zvýšené mimo jiné také nástupem povinné registrace plateb EET. Jako pravidelný zákazník některých menších obchodů mám příležitost nahlédnout do pokladen, které jsou zde používány. S nástupem EET jsem si všiml, že mnoho obchodů z těch menších např. typicky „Vietnamci“, měla již v minulosti na prodejním stole monitor počítače nebo notebook, ačkoliv ho nutně nepoužívaly pro e-pokladnu. Často jsem vídal, že jej používají pro Skype, nebo pro sledování internetu, během chvil, kdy v obchodu měli minimum, nebo žádné zákazníky. S nástupem EET, jsem si všiml, že většina z nich má nyní nějakou formu E-pokladny na těchto počítačích nainstalovanou.

Při pozorování mě překvapilo, že ačkoliv je o těchto menších prodejcích známo, že velmi spolupracují mezi sebou, a jednají velmi komunitně, tak verze e-pokladen jsou mezi nimi velice různorodé. Nedošlo tedy k tomu, že by si všichni koupili jednu a tu samou verzi, kterou by někdo odzkoušel a doporučil. Dále stále ještě vídám prodejny s „základní pokladnou“, které jsou však připojeny do EET. Trh je tedy velice různorodý, a dokonce vídám obchody, které mají grafiku pokladny ve stylu ms-dos, které zřejmě používají již velmi dlouhou dobu.

Ve své práci budu moci zúročit i svoje znalosti z letní brigády, kdy jsem byl ještě studentem Gymnázia a byl jsem jako brigádník za pokladnou velkého obchodu, která už v té době byla velice pokročilá a obsahovala mnoho funkcionalit.

2. Vize

Práce se bude zabývat tvorbou e-pokladny pro malou prodejní jednotku. Typicky např. různé samoobsluhy a večerky. Pokladna bude muset být schopna dohledávat zboží podle EAN kódu, případně podle číselného kódového označení zboží pro pečivo a zboží neobsahující EAN. Pokladna bude evidovat platbu a vygeneruje účtenku.

Pokladna bude rozdělena se server a klienta pro případ, kdy v prodejně bude více pokladen, což dnes není neobvyklé ani u velmi malých samoobsluh. Z toho důvodu je nutné mít k dispozici vždy aktuální ceny zboží, bez nutnosti pro každou pokladnu nastavovat tyto hodnoty zvlášť.

Serverová část bude mít uloženy v databázi položky zboží, které bude možné prodávat, dále zde bude cena zboží včetně evidence, do jaké spadá kategorie DPH, aktuální množství prodaného zboží a historii plateb.

Klientská část bude uživatelským rozhraním tohoto systému. Bude zde možné sestavit nákup, zaevidování do EET, prohlédnutí si historie prodejů, provedení importu položek a jejich editaci. Dále zde bude možné spravovat uživatelské účty a provést inventuru – výpis prodaného zboží v určitém období.

1.3 Cíle práce

Cílem projektu bude navrhnout aplikaci, zvolit technologie pro vyhotovení. Vytvořit serverovou a klientskou část e-pokladny alespoň o minimální funkcionalitě, tak aby byla použitelná v malé obchodní jednotce. Vzít v potaz nutnou evidenci EET a zjistit možnosti její implementace v rámci e-pokladny a připravit e-pokladnu na její implementování.

Kritéria úspěchu:

- Dodržení časového omezení pro práci
- Dodržení termínu odevzdání bakalářské práce
- Zprovoznění serverové, aby bylo možné evidovat platby a využívat databázi zboží
- Navrhnout klientskou část, aby byla uživatelsky jednoduše a snadno použitelná
- Zprovoznění klientské části, aby byli implementovány základní funkcionality
- Provést testování pokladny

2.4 Současný stav

Aktuální stav malých obchodních jednotek – soukromé samoobsluhy je takový, že po zavedení EET, již prodejci začali používat registrační pokladny do EET, případně e-pokladny s propojením do EET. U malých prodejců je velmi často vidět, že mají v místě prodejního pultu k dispozici počítač a to i když ho nutně nepoužívají pro e-pokladnu. Na trhu existuje nespočet aplikací pro evidenci EET, avšak většinou nenabízí mít možnost zdarma databázi položek k prodeji a evidenci jejich EAN. Často se jedná jen o evidenci pro EET, nebo nutnost mít placenou variantu. Leckde je proto možné narazit i na dvojí vydávání prodejního dokladu, kde jeden doklad slouží jako evidence do EET, kde není nutné uvádět jaké bylo prodáno zboží a

druhý kde je vypsané zboží ale již bez evidence do EET z důvodu nutnosti použití různých systému pro evidenci plateb.

2.5 Budoucí stav

Práce bude umožňovat obchodům využívat výhod e-pokladny bez měsíčních poplatků. Zároveň díky otevřenému zdrojovému kódu, ji bude možné dále upravovat, vylepšovat a testovat.

V e-pokladně jako takové bude možné sestavit běžný nákup. Importovat položky a zřídit uživatelské účty pro různé zaměstnance. Bude zde historie nákupů a možnost editace položek. Půjde dohledat množství prodaného zboží v určitém časovém intervalu. Pokladna bude evidovat do EET, alespoň v běžném režimu.

2.6 Zdůvodnění práce

Práci bych rád vytvořil s otevřeným zdrojovým kódem. Chtěl bych tím tedy umožnit vlastnit relativně funkční elektronickou pokladu s možností dalšího vývoje a integrace dalších služeb, bez nutnosti za ní platit. Pokladna díky tomu bude moci být snadno dále vylepšována a testována. Zároveň se v této práci určitě zdokonalím v oblasti návrhu a implementace aplikací, většího rozsahu, než který je běžný třeba u seminárních prací.

2.7 Přínosy práce

- Případnou možnost pro prodejce pro lepšího využití většinou už dostupného hardware prodejců
- Možnost upravit a otestovat si tuto aplikaci dle potřeby, díky otevřenému zdrojovému kódu
- Napsání práce většího rozsahu.
- Zdokonalení se v návrhu a případně i konfiguraci aplikací.

2.8 Konkrétní výstupy práce

Výstupem bude jak serverová, tak klientská část aplikace. Bude se jednat o zdrojové kódy, které budou zpřístupněné ve formě projektu z vývojového prostředí Netbeans.

2. Analýza

2.1 SWOT

SWOT analýza se používá pro zhodnocení vnitřních a vnějších faktorů, které mohou ovlivnit projekt. Mezi vnitřní faktory patří silné a slabé stránky, mezi vnější faktory patří příležitosti a hrozby. Tuto SWOT analýzu vytvářím za sebe pro tento projektu.

Silné stránky:

- Aktuální téma vzhledem k novým povinnostem pro prodejce
- Možnosti pro další rozvoj práce
- Integrovaná evidence do EET

Slabé stránky:

- První práce většího rozsahu
- Jako jediný vývojář budu muset dbát na časový plán, aby se práce stihla včas

Příležitosti:

- Získání nových zkušeností
- Osvojení nových technologií
- Konzultace s vedoucím práce

Hrozby:

- Nedostatek času na práci
- Příliš vysoká obtížnost
- Neschopnost vyvíjet na čas z nepředpokládaných důvodů

2.2 FURPS

„Položky modelu FURPS jsou chápány jako dimenze kvality, ve kterých je sledována kvalita produktu.“[1]

Funkčnost

Aplikace bude umožňovat prodej v malé prodejní jednotce, typicky samoobsluha nebo večerka. Bude mít klientskou část, která bude umožňovat provedení nákupu zboží a tisk účtenky. Bude propojena s EAN skenerem, kterým bude možné zboží načítat, případně bude možné manuálně zadat EAN kód, nebo číselný kód zboží bez EAN kódu. Bude zde i možnost zadat váhu zboží, které je na váhu, nebo jeho množství.

Serverová část bude mít v sobě uloženy informace o zboží jejich cenu, DPH, množství prodaného zboží, EAN, nebo číselný kód pro zboží bez EAN. Bude provádět evidenci do EET. Bude zde také uložena historie nákupů v obchodě a možnost snadno sledovat množství prodaného zboží.

Vhodnost k použití

Aplikace bude určena pro malou obchodní jednotku a tedy nebude vhodná pro velké obchodní centra, která mají specifický hardware, pro které je nutné vytvořit jiné uživatelské rozhraní i funkcionalitu. Bude spíše vhodná pro obchody s dvěma až třemi pokladnami. Díky tomu, že informace budou uloženy na serveru, nebude nutné pro každou pokladnu zvlášť

přepisovat případné změny cen zboží. Bude moci být použita i pro evidenci množství prodaného zboží.

Spolehlivost

Software by měl být průběžně testován. Pro provoz bude pochopitelně potřeba funkční připojení k internetu pro evidenci do EET, pokladna bude ze začátku podporovat jen běžný režim. Bude zapotřebí také zajistit funkční hardware ze strany prodejce.

Výkon

Rychlost odpovědi na požadavky na server by měla být v řádu desetin vteřiny. Samotná evidence do EET, která je však pouze jednou za celou účtenku, může probíhat řádově vteřiny, zde záleží na rychlosti evidence ze strany státní správy a také rychlosti zpracování ze strany tiskárny. Aplikace by měla být dostatečně výkonná i při souběhu dvou či tří pokladen v prodejně.

Schopnost být udržována

Aplikace bude mít otevřený zdrojový kód. Bude tedy udržovatelná i od třetích stran, nebo dalších poskytovatelů a otevřená k dalšímu rozšiřování či testování.

2.3 Předpoklady

- Potenciální klienti již mají potřebný hardware pro spuštění aplikace
- Potenciální klient bude mít stabilní připojení na internet
- Potenciální klient je malou obchodní jednotkou, s řádově jednotek kas

2.4 Omezení

- Systém je schopen zaevidovat pouze zboží, které má v databázi
- Systém musí běžet na Windows 7 a novější z důvodu zabezpečeného připojení do EET

2.5 Akceptační kritéria

- Aplikace bude muset umožnit provést nákup, za využití informací ze serveru, ze kterých bude načítat cenu zboží, vzhledem k jeho EAN/kódu.
- Aplikace umožní tisk účtenek
- Aplikace bude nějakou formou implementovat evidenci do EET a bude fungovat alespoň v testovacím režimu EET
- Aplikace bude mít snadno ovladatelné uživatelské rozhraní

2.6 Rizika

„Projektová rizika v zahrnují všechny druhy rizik, která mohou jakýmkoliv způsobem ohrozit projekt. Klíčová projektová rizika jsou ta, která ohrožují cíl, čas a náklady projektu.“[2]

- Závažné chyby v systému
 - o Pravděpodobnost: 15%
 - o Dopad:
 - Prodloužení doby vývoje
 - Omezení množství integrované funkcionality
 - o Minimalizace rizika
 - Testování ve vývoji

- Nedodržení termínu pro odevzdání práce
 - o Pravděpodobnost: 15%
 - o Dopad:
 - Nemožnost zpracovat připomínky k práci
 - o Minimalizace rizika
 - Dodržování časového plánu práce

- Chyby v knihovnách třetích stran
 - o Pravděpodobnost: 20%
 - o Dopad:
 - Nemožnost zpracovat včas vybranou funkcionalitu
 - o Minimalizace rizika
 - Vybírat ověřené a testované knihovny

3 Business analýza

3.1 Sběr požadavků na práci

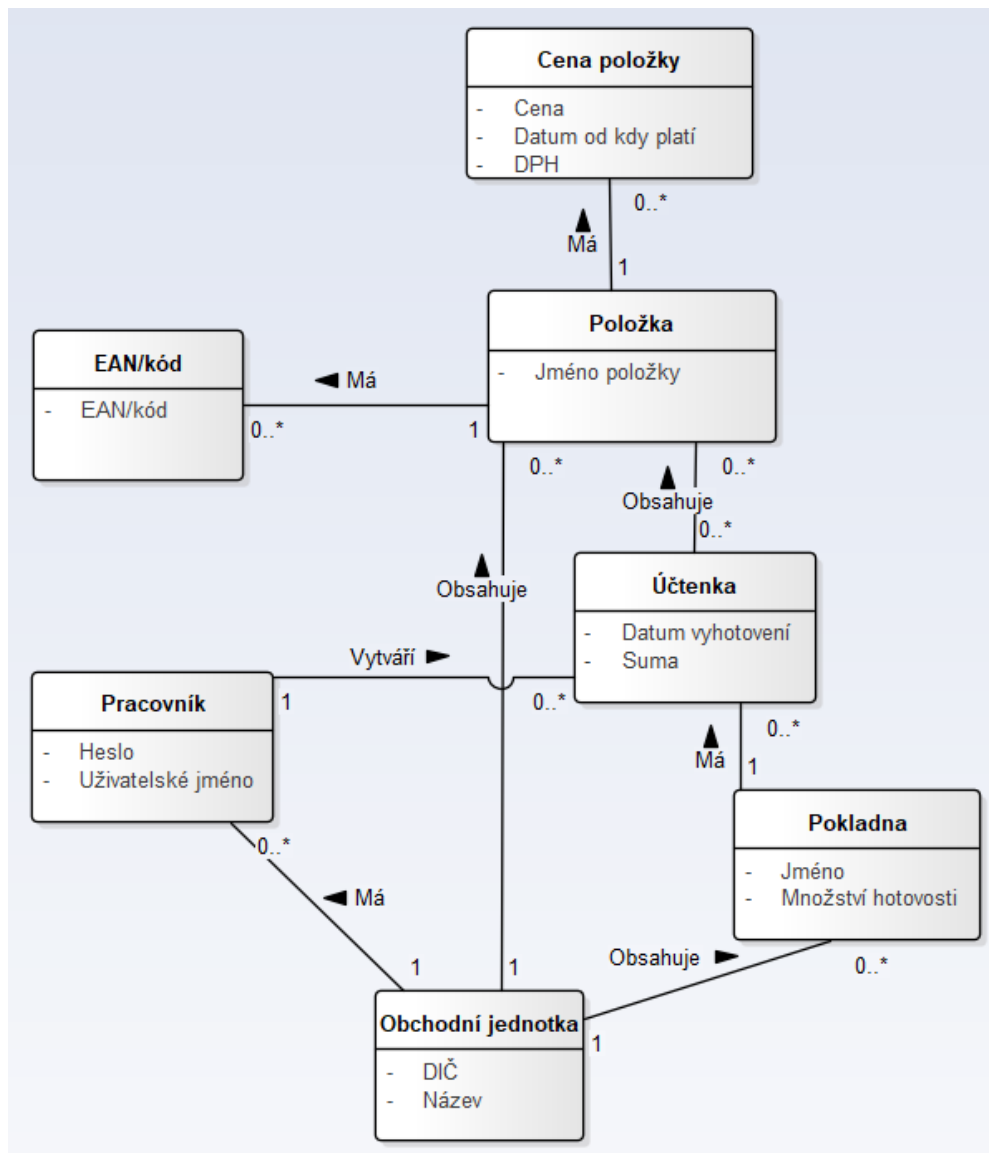
Vzhledem ke zkušenostem i se složitějším pokladním systémem, které jsem získal během brigády, kdy jsem ještě studoval na Gymnáziu, jsem se po prostudování různých recenzí a srovnání pokladen rozhodl navrhnout požadavky na systém sám.

Jako hlavní zdroj srovnání a recenzí jsem použil web eet-ano-ale.cz[3], kde se nachází řádově desítky pokladen, ke kterým jsou zde zpracovány testy a recenze.

3.2 Entitní model

Entitní model v business analýze slouží pro evidenci business entit, definici jejich parametrů a evidenci vazeb, které mezi sebou mají. Slouží pro následnou lepší orientaci v business analýze.

Pracovník	Entita uchovávající informace o uživateli, které spadají pod obchodní jednotku. Budou také uvedeni v účtenkách, které vystaví.
Obchodní jednotka	Entita uchovávající informace ohledně obchodní jednotky. Je určitým způsobem provázána do všech entit. To umožní snadno evidovat v aplikaci i více provozoven naráz.
Pokladna	Entita uchovávající informace ohledně pokladen v obchodní jednotce. Pokladna je také evidována ve vyhotovených účtenkách.
Účtenka	Entita uchovávající informace ohledně účtenek. Je v ní evidováno datum a vyhotovení a suma za účtenku dále položky, které jsou v účtence obsaženy.
Položka	Entita uchovávající informace ohledně zboží.
Cena položky	Entita uchovávající informace ohledně ceny a časové platnosti ceny zboží. Protože může být vystavených víc cen, pro jedno zboží, vždy platí ta, která je dle časové platnosti zboží tou nejnovější.
EAN/kód	Entita uchovávající informace ohledně EAN/kód, patřící ke zboží.



Obrázek 1. Entitní model

3.3 Business požadavky

Zde uvádím business požadavky práce, které jsem navrhl dle zkoumání pokladen na trhu a vlastní zkušenosti. Pokladna může být do budoucna rozšířena o další funkce, zde uvádím požadavky, které aktuálně považuji za důležité pro efektivní využití pokladny.

BP-1 / Zpracování nákupu

Jako prodavač potřebuji sestavit nákup z položek, aby mohli být prodány zákazníkovi.

BP-2 / Tisk účtenky

Jako prodavač potřebuji vytisknout účtenku, aby mohla být předána zákazníkovi.

BP-3 / Importování položek

Jako vedoucí prodejny potřebuji importovat položky do systému, aby mohli být použity pro nákup.

BP-4 / Správa položek

Jako vedoucí prodejny potřebuji editovat položky, když dojde k jejich změně.

BP-5 / Vyhledání výpisu z provedených nákupů

Jako vedoucí prodejny potřebuji vyhledávat výpisy z provedených nákupů, aby mohli být dohledány v případě reklamace.

BP-6 / Vyhledání množství prodaného zboží

Jako vedoucí prodejny potřebuji vyhledat množství prodaného zboží, aby mohla být provedena inventura.

BP-7 / Zaevidování do EET

Jako prodavač potřebuji zaevidovat nákup do EET, abych nedostal pokutu.

BP-8 / Správa uživatelských účtů

Jako vedoucí prodejny potřebuji vytvářet uživatelské účty pro jednotlivé pracovníky

BP-9 / Množství pokladen

Jako vedoucí prodejny potřebuji mít připojeno více pokladen k systému v jednu chvíli.

BP-10 / Správa obchodních jednotek

Jako administrátor, potřebuji vytvořit nový obchod v systému pro každou pobočku

BP-11 / Přehledné uživatelské rozhraní

Jako prodavač, potřebuji přehledné uživatelské rozhraní, aby systém bylo jednoduché ovládat.

3.4 Funkční požadavky

„Funkčními požadavky rozumíme požadavky na věcný, problémový obsah systému.“[4]

Zde uvedené požadavky konkretizují vypsané business požadavky a popisují funkce, které bude systém podporovat.

Funkční požadavky k BP-1 / Zpracování nákupu

Skenovat EAN kód položky

Přidat položku ručně dle EAN kódu

Přidat položku ručně, vyhledanou v systému

Odebrání položky z nákupu

Přijetí platby hotově

Přijetí platby kartou

Stornování platby

Funkční požadavky k BP-2 / Tisk účtenky

Vytiskne účtenku na tiskárně z provedeného nákupu

Opakovaně vytiskne účtenku na tiskárně z provedeného nákupu

Funkční požadavky k BP-3 / Importování položek

Importovat textový soubor s položkami

Funkční požadavky k BP-4 / Správa položek

Přidání a odebrání EAN k položce

Přidání ceny včetně DPH k položce

Vyhledání položky podle jména

Funkční požadavky k BP-5 / Vyhledání výpisu z provedených nákupů

Vyhledání záznamu dle časového intervalu prodeje

Vyhledání záznamu dle položky na záznamu

Funkční požadavky k BP-6 / Vyhledání množství prodaného zboží

Vyhledání počtu všeho prodaného zboží, z určitého období

Vyhledání počtu konkrétního prodaného zboží, z určitého období

Funkční požadavky k BP-7 / Zaevidování do EET

Zaevidování tržby do EET v běžném režimu a získání FIK a BKP pro tisk účtenky

Evidence pokladních Id a Id prodejních jednotek pro systém EET.

Funkční požadavky k BP-8 / Správa uživatelských účtů

Vytvoření uživatelského účtu pod prodejní jednotkou s oprávněním vedoucí nebo prodavač.

Vytvoření nové obchodní jednotky s prvním účtem s oprávněním jako vedoucí.

Funkční požadavky k BP-9 / Množství pokladen

Podpora více pokladen spuštěných naráz.

Funkční požadavky k BP-10 / Správa obchodních jednotek

Vytvoření nové obchodní jednotky.

Správa názvu a DIČ obchodní jednotky.

Funkční požadavky k BP-11 / Přehledné uživatelské rozhraní

Jednoduché a přehledné uživatelské rozhraní, pro různé funkce systému.

3.5 Role

V systému budou dostupné tři úrovně uživatelských rolí. Ty budou sloužit pro to, aby systém mohlo užívat více uživatelů, jen s oprávněními, které se pro ně budou hodit.

Pokladní

Bude moci na pokladně namarkovat nákup, vyhledávat zboží podle názvu a tisknout účtenky. Nebude mít tedy příliš pravomocí a účet může být určen, třeba pro brigádníka, nebo méně spolehlivou osobu, aby nemohla v systému napáchat nějaké škody, nebo nepovolené úpravy.

Vedoucí

Vedoucí bude navíc mít možnost importovat položky a spravovat položky ze systému. Bude moci vyhledávat množství prodaného zboží a také dohledávat výpisy z provedených nákupů. Bude moci zřizovat nové uživatelské účty a blokovat/odblokovávat účty spadající pod stejnou obchodní jednotku.

Administrátor

Administrátor bude v systému pouze jeden, bude se jednat o výchozí účet, vytvořen po spuštění serverových služeb a připojení databáze. Bude moci v systému vytvářet nové obchodní jednotky. Systém bude moci mít tedy v sobě více provozoven spravovaných z jednoho serveru.

3.6 Případy užití

„Případ užití (nebo zkráceně UC) je sada několika akcí, které vedou k dosažení určitého cíle. Use Case může být přidání komentáře k článku, registrování nového uživatele nebo např. vytisknutí dokumentu. Definuje tedy jednu funkcionalitu, kterou by měl navrhovaný systém umět.“[5]

Scénáře případů užití

Z důvodů že aplikaci sám navrhují i implementují uvedu jen některé scénáře. Dále zde uvádím Obrázek 2. Diagram případů užití, zachycující případy užití rozdělené, mezi jednotlivé role.

Vyhledat výpis z provedeného nákupu

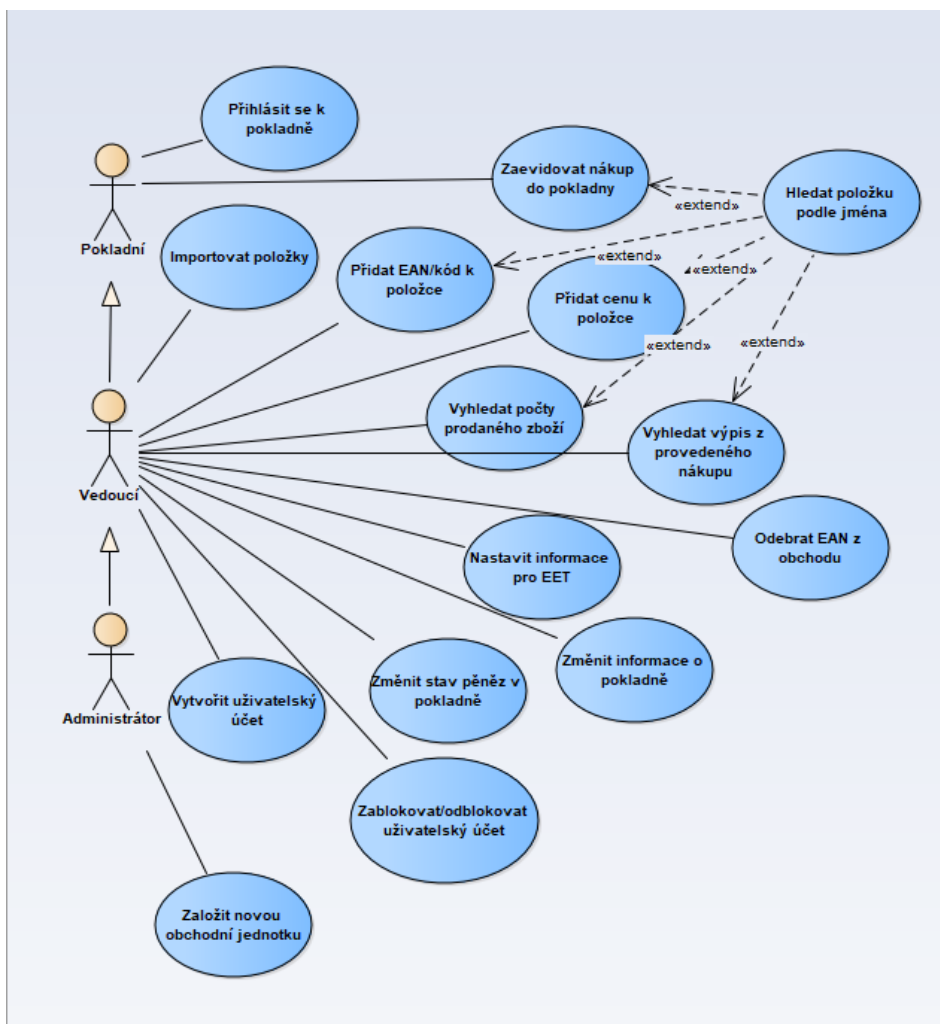
1. Pokladna zobrazí obrazovku pro správu účtenek.
2. Uživatel volitelně vybere časový interval pro hledaný výpis.
3. Uživatel volitelně vybere položku, kterou má výpis obsahovat zadáním EAN/kód. (Může zde použít funkci hledat položku dle jména)
4. Pokladna vyhledá položku.
5. Uživatel může zrušit výběr položky, pokud je nějaká vybraná.
6. Uživatel potvrdí vyhledání záznamů.
7. Pokladna vypíše odpovídající nalezené záznamy.

Přidat cenu k položce

1. Pokladna zobrazí obrazovku pro správu položek
2. Uživatel vybere položku, kterou má výpis obsahovat zadáním EAN/kód. (Může zde použít funkci hledat položku dle jména)
3. Pokladna se pokusí vyhledat položku (pokud není nalezena pokračuje bodem 2.)
4. Uživatel zadá novou cenu s DPH
5. Pokladna zkontroluje, že je zadána cena i DPH (pokud ne pokračuje se bodem 4.)
6. Pokladna zaeviduje novou cenu zboží s platností od této chvíle

Vytvoření nové obchodní jednotky

1. Pokladna zobrazí obrazovku pro nastavení marketu(pobočky)
2. Uživatel zadá nové uživatelské jméno a heslo, pro vytvoření účtu, pod kterým bude nová pobočka.
3. Pokladna zkontroluje, vyplnění uživatelského jména, hesla a jestli je uživatelské jméno. (Pokud ne pokračuje se bodem 2.)
4. Pokladna založí novou pobočku, pod kterou je zaregistrován nový vytvořený uživatel.



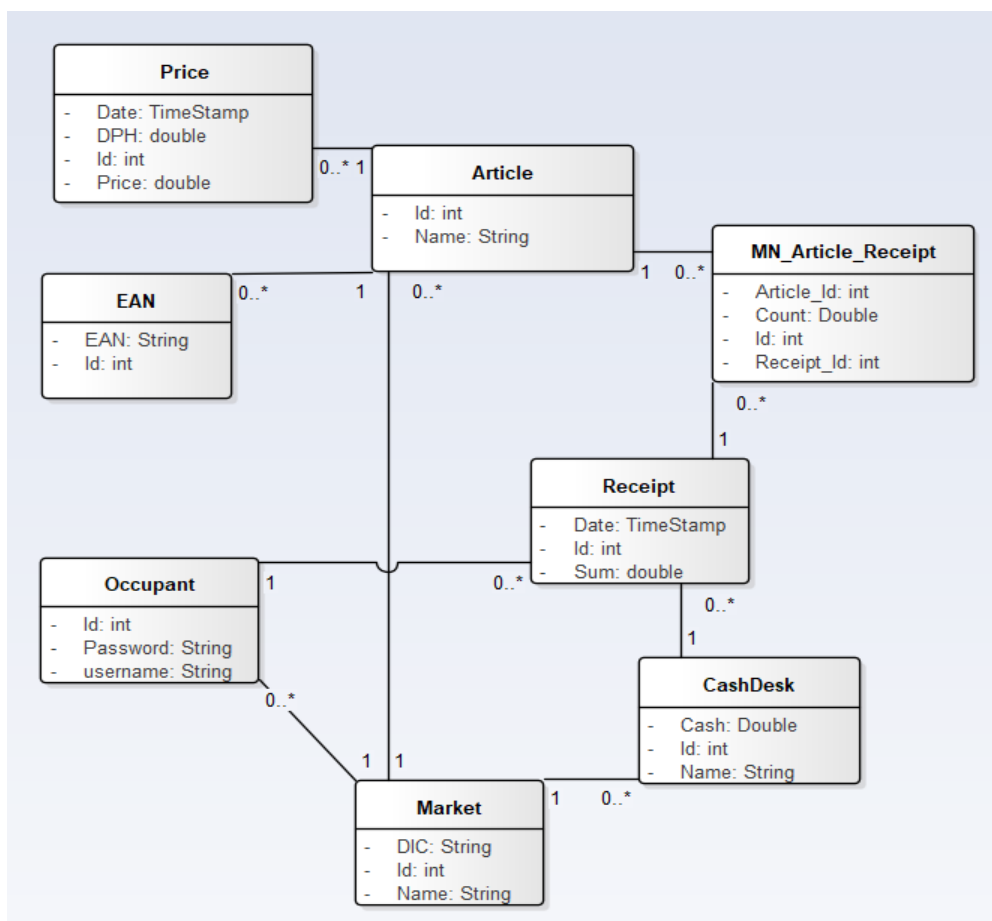
Obrázek 2. Diagram případů užití

3.6 Datový model

Zde uvádím datový model, uložených informací v databázi. Model zobrazuje entity, jejich kardinalitu mezi sebou a atributy s jejich datovým typem. Entity jsou pojmenovány anglicky, aby byl systém snadno udržitelný i vývojáři nehovořících česky.

V aplikaci je implementována abstraktní třída s automaticky generovaným Id pro snazší práci s entitami, které od ní dědí. Tato entita je uvozena anotací @MappedSuperclass, díky tomu není v databázi uložena.

- Occupant** Entita uchovávající informace o uživateli systému.
- Market** Entita uchovávající informace ohledně obchodní jednotky.
- CashDesk** Entita uchovávající informace ohledně pokladen v obchodní jednotce.
- Receipt** Entita uchovávající informace ohledně účtenek.
- MN_Article_Receipt** Entita uchovávající informace o tom, která položka patří do které účtenky a počet, kolikrát byla zaevidována.
- Article** Entita uchovávající informace ohledně názvu zboží.
- Price** Entita uchovávající informace ohledně ceny a časové platnosti ceny zboží.
- EAN** Entita uchovávající informace ohledně EAN/kód, patřící ke zboží.



Obrázek 3. Databázový model

4. Implementace

4.1 Volba technologií pro server

Java + Spring + Maven

Pro server jsem zvolil pro naprogramování programovací jazyk Java s frameworkem Spring. Jazyk je dle mě vhodný, proto že s frameworkem představuje dobrou základnu pro to časem aplikaci rozšířit a implementovat i mnohem více funkcí. Jako asi nejdůležitější vnímám dostupné tutoriály a dokumentaci, protože snadná dostupnost těchto informací umožní mnohem pohodlnější a rychlejší vývoj. Aplikaci je dále možné snadno sestavit pomocí vývojových prostředí a spustit na počítači za pomoci otevřených technologií, což může velmi pomoci lidem, kteří by si chtěli něco upravit v otevřeném zdrojovém kódu práce. Framework Spring umožňuje mnohem lepší přehlednost a správu celého projektu. Jako velice podstatné také vnímám, že si vývojové prostředí díky nástroji pro správu závislostí Maven je schopno samo stáhnout potřebné závislosti a projekt by měl být tím pádem snadno spustitelný pro každého.

H2 databáze

Jako databázi jsem vybral H2 databázi. Výhody této databáze je pro tuto práci hned několik.

1. Jedná se o open-source databázi.
2. Databáze umožňuje uložení ve formě souboru na disku a díky tomu nevyžaduje žádné složité instalace. To ji zároveň umožňuje velice snadno přenášet nebo zálohovat i pro obvyčejného uživatele.
3. Databáze je malá a rychlá, plně postačující pro potřeby této práce, díky tomu nezabere příliš výkonu počítače, na kterém poběží a bez větších problémů může běžet celá aplikace na jednom notebooku, který bude dále využíván i na jiné činnosti.
4. Je dostupné jednoduché webové rozhraní pro správu databáze, které je veliké v řádu jednotek megabajtů.

4.2 Volba technologií pro klienta

Java + SWING + Maven

Pro klienta jsem zvolil opět jak programovací jazyk Javu. Pro klienta je tato volba vhodná z důvodu, že je Java velice rychlá a umožňuje využití mnoha knihoven, zároveň opět obsahuje nástroj pro řízení závislostí Maven, projekty tedy bude snadné rychle zprovoznit a spustit.

U tohoto klienta se předpokládá i implementace funkcí jako import položek a pokladního rozhraní, tudíž Java + SWING zde umožňuje sestavit rychle reagující pěknou aplikaci, která bude moct implementovat mnoho funkcí a bude se jednat o jakési hlavní rozhraní pro pokladnu i do budoucna. Bude se tedy jednat o klienta někde na pomezí mezi tenkým a tlustým, někdy nazývaný chytrým klientem.

Pro budoucí rozvoj si myslím, že je to optimální volbou a že další rozšíření na mobil nebo tablet, je vhodnější učinit pomocí cíleně napsané aplikace na toto zařízení než se teď pokoušet psát webového klienta, který poběží sice všude, ale ne tak dobře.

eet-client

Používám velice dobře zdokumentovanou a napsanou třetí generaci eet-client od Tomáše Dvořáka. Jedná se o knihovnu, která je přímo připojena na servery státní správy. Knihovna umožňuje evidenci v různých režimech, obsahuje ukázkové příklady a mnoho funkcí. Knihovna je dostupná na GitHubu a je ji možné díky JitPacku snadno přidat jako závislost do projektu, tudíž ji není nutné nikde extra stahovat při kompilaci projektu. Zároveň je knihovna pod velice volnou licencí MIT. Pro implementaci volím klientskou stranu, z důvodu, že by se mělo jednat vždy o konkrétní pokladnu, která obdrží své Id od státní správy pro evidenci tržeb a vyšší rychlosti zpracování požadavku.

Unirest

Jedná se o knihovnu zprostředkávající volání http dotazů. Knihovna je opět dobře dokumentovaná, obsahuje mnoho funkcí a je dostupná skrze nástroj Maven, pomocí kterého si ji opět umí projekt sám snadno stáhnout. Je dostupná také pod velice volnou licencí MIT.

4.3 Konfigurace serveru

Pro konfiguraci serveru jsem využil open-source projekt pod licencí GNU verze 3, jedná se o EAR Setup Verifier[6]. Tento projekt obsahuje počáteční konfiguraci pro zprovoznění služeb knihovny Spring a stažení všech potřebných závislostí. Dále obsahuje jednoduchou logiku pro demonstraci služeb Springu a jeho schopností. Projekt jsem tedy upravil a použil pro tuto práci.

Pro konfiguraci zabezpečení, která v EAR Setup Verifier není jsem se inspiroval a využil některých souborů z projektu Reporting Tool[7], jedná se opět o open-source projekt pod licencí GNU verze 3. Soubory, které jsou inspirovány, nebo použity z tohoto projektu jsou balíček zabezpečení a konfiguraci aplikace. Zde uvádím výpis inspirovaných nebo použitých souborů z tohoto projektu, někdy se jedná i jen o prázdné, nebo téměř prázdné třídy, pro potenciální možnost konfigurace systému.

cz.cvut.config

DispatcherServletInitializer.java

SecurityConfig.java

ServiceConfig.java

cz.cvut.security

AuthenticationFailure.java

AuthenticationSuccess.java

DefaultAuthenticationProvider.java

HttpAuthenticationEntryPoint.java

LogoutSuccessHandler.java

SessionTimeoutManager.java

4.4 Konfigurace klienta

Pro konfiguraci klientu bylo pouze nutné použít ukázkové zdrojové kódy a dokumentaci použitých knihoven unirest a eet-client.

Pro unirest jsem čerpal informace z jejich dokumentace[8]. A dle těchto instrukcí nastavil prostředí pro posílání http dotazů.

Pro eet-client jsem využil ukázkového zdrojového kódu Application.java[9], kde jsou demonstrovány možnosti knihovny.

Uvádím zde pro představu možný způsob zaslání požadavku na servery státní správy, pomocí knihovny eet-client. K aplikaci příkládám i certifikáty, které byly přiložené ke knihovně, sloužící pro testovací evidenci tržeb. Pro reálnou evidenci je nutné dodat certifikát, který je provozovně vystaven na základě její žádosti. Vzhledem k zabezpečenému připojení je nutné pro správnou funkci pokladnu spustit na zařízení s Windows 7 a novější.

```
EETClient service = EETServiceFactory.getInstance(clientKey, serverKey);

TrzbaDataType receipt = new TrzbaDataType()
    .withDicPopl("CZ683555118")
    .withIdProvoz(243)
    .withIdPokl("24/A-6/Brno_2")
    .withPoradCis("#135433c/11/2016")
    .withDatTrzby(new Date())
    .withCelkTrzba(new BigDecimal("3264"));

TrzbaType request = service.prepareFirstRequest(receipt, CommunicationMode.REAL);
SubmitResult response = service.sendSync(request, EndpointType.PLAYGROUND);

String bkp = response.getBKP();
String fik = response.getFik();
```

Obrázek 4. Příklad evidence do EET

Pro tiskové služby jsem použil předpřipravený soubor bill_form.java[10], ve kterém byly připraveny parametry pro tisk na 80mm tiskárně. Soubor jsem si upravil a použil pro tisk v aplikaci. Ukázalo se, že ne všechny virtuální tiskárny si rozumí s formátem 80mm na šířku, pro testování se mi osvědčila virtuální tiskárna Microsoft XPS Document Writer, která si byla s neobvyklým požadavkem na velikost papíru schopna poradit asi nejlépe. Tisk je nastaven automaticky na výchozí tiskárnu systému. V příloze C uvádím ukázkovou účtenku, vytištěnou ze systému.

4.5 Architektura serveru

Server bude zpracovávat drtivou část všech požadavků sám, aby bylo možné napsat pouze tenké, případně chytré klienty klienty a usnadnila se tak práce, na možné budoucí aplikaci pro mobil, nebo tablet s různými operačními systémy.

Server bude implementován jako „Vícevrstvá architektura se často označuje jako multi-tier nebo ještě častěji jako n-tier, kde n vyjadřuje počet vrstev, ze kterých se vícevrstvá architektura skládá.“[11] Vrstvy by měli vždy komunikovat jen se sousedními vrstvami a tím dosáhnout mnohem lepší přehlednosti a udržitelnosti kódu. Zde uvádím jednotlivé vrstvy rozdělené do balíčků v aplikaci.

cz.cvut.bc.model	Vrstva, ve které budou modely ukládané do databáze.
cz.cvut.bc.dao	Vrstva, ve které budou objekty pro práci s modely v databázi.
cz.cvut.bc.service	Vrstva, ve které bude business logika aplikace.
cz.cvut.bc.rest	Vrstva, ve které bude rozhraní pro služby REST.

4.6 Architektura klienta

Klientská část této práce je někde na pomezí mezi tenkým a tlustým klientem, někdy nazývaným jako chytrý klient. Uživatelská část tohoto klienta je založena na vzoru MVC. „MVC je velmi oblíbený architektonický vzor, který se uchytil zejména na webu, ačkoli původně vznikl na desktopech.“[12] Uvádím zde rozdělení členění aplikace do jednotlivých balíčků.

Model v této aplikaci případě představuje připojení na služby REST serveru.

Kontrolér je v tomto případě třída Logic a CashDeskService, které zpracovávají informace, které přijímají od uživatele.

View jsou jednotlivé třídy implementující JFrame, do kterých uživatel zadává informace.

5. Testování

Pro testování jsem zvolil z větší části ruční testování z důvodu časové efektivity a rychlosti zkoušení různých testovacích scénářů, které jsem mohl v průběhu rychle upravovat měnit.

5.1 Unit testy

Na serverové straně jsou napsané unit testy pro práci s modely v databázi, což je velice efektivní pro vývoj. Jakýkoliv problém ze strany databáze a schopnosti načítat a ukládat modely je velice rychle odhalen.

5.2 Testování vývojářem

Testování vývojářem proběhlo skrze klientskou aplikaci. Byli zkoušeny různé nastavení a možnosti. Testování probíhalo v průběhu celého vývoje s tím, že na konci proběhlo intenzivnější a hlubší testování všech součástí.

5.3 Akceptační testy

System prošel akceptačními testy, které testovali funkcionality, které systém obsahuje. Byla testována jejich funkčnost a spolehlivost. Testovány byly tyto funkcionality:

- Prodejní pokladna a operace s ní

- Správa položek

- Správa účtenek

- Správa prodaného zboží

- Nastavení pokladny

- Správa uživatelských účtů

5.4 Uživatelské testy

System byl otestován dvěma uživateli, kteří byli ponecháni se systémem, po rychlém předvedení dostupných funkcí. Zkoušeli si různé dostupné funkce a pracovat se systémem. Reakce byli vcelku pozitivní, pochvalovali si intuitivní uživatelské rozhraní.

5.4 Zhodnocení testů

Během testů bylo objeveno a opraveno mnoho chyb, systém je již velice komplexní a vyžaduje dlouhé a soustavné testování pro drtivé většiny chyb. Díky otevřenému zdrojovému kódu práce, si však může každý může snadno pokusit otestovat si tento systém a přispět tak, k dalšímu rozvoji aplikace.

6. Závěr

Vývoj práce byl mnohem těžší a delší, než jsem na začátku přepokládal. Věřím však, že se mi podařilo vypilovat si trochu toužené schopnosti ohledně návrhu a implementace aplikací.

6.1 Zhodnocení práce

Práce se mi podařilo dovést do zdárného konce, takže je již vcelku stabilní a funkční. Obsahuje mnoho různých funkcionalit a je snadné do ní rychle doplnit nové funkcionality v řádu několika málo hodin. Práce je tudíž podle mne úspěšná a v rámci možností i kvalitní.

6.2 Budoucnost práce

Práce může být dále rozšiřována a testována díky otevřenému zdrojovému kódu. Vývoj v této oblasti určitě bude postupovat rychle dále a na trhu se začnou časem objevovat velice osvědčená a kvalitní řešení. Věřím, že i tato práce může pomoci v rozvoji ať už jako práce, která může být někomu k dispozici pro inspiraci, nebo pro další rozvoj.

7. Seznam citací

- [1] VANĚK, Dušan. *Dimenze kvality FURPS+* [online]. 4.6.2012 [cit. 2018-05-20]. Dostupné z: labe.felk.cvut.cz/~marikr/teaching/Y33TSW_10/FEL-04_doplněk_Dimenze_kvality_FURPS.ppt
- [2] Projektová rizika. *Management mania* [online]. 2016 [cit. 2018-05-20]. Dostupné z: managementmania.com/cs/projektova-rizika
- [3] *EET ano, ale...* [online]. 2016 [cit. 2018-05-20]. Dostupné z: www.eet-ano-ale.cz
- [4] INFORMAČNÍ SYSTÉMY. *Katedra informatiky, VŠB-TU Ostrava* [online]. [cit. 2018-05-20]. Dostupné z: wiki.cs.vsb.cz/images/5/58/Inz3.pdf
- [5] Lekce 2 - UML - Use Case Diagram. *ITnetwork* [online]. 2018 [cit. 2018-05-20]. Dostupné z: www.itnetwork.cz/navrh/uml/uml-use-case-diagram
- [6] EAR Setup Verifier. *GitLab na FEL ČVUT* [online]. 23.9.2017 [cit. 2018-05-20]. Dostupné z: gitlab.fel.cvut.cz/ear/setup-project
- [7] Reporting Tool. *GitLab na FEL ČVUT* [online]. 9.12.2016 [cit. 2018-05-20]. Dostupné z: gitlab.fel.cvut.cz/ear/seminar-rest
- [8] Unirest for Java. *Unirest* [online]. [cit. 2018-05-20]. Dostupné z: unirest.io/java.html
- [9] Application.java. *Github* [online]. 18.2.2017 [cit. 2018-05-20]. Dostupné z: github.com/todvora/eet-client-demo/tree/master/src/main/java/cz/tomasdvorak/eet/demo
- [10] Bill_system. *Disk Google* [online]. 28.7.2017 [cit. 2018-05-20]. Dostupné z: drive.google.com/file/d/0ByaTLXUKQ3AEdVFqMXJbFhDVFU
- [11] Vícevrstvá architektura: popis vrstev. *Cleverandsmart* [online]. 2018 [cit. 2018-05-20]. Dostupné z: www.cleverandsmart.cz/vicevrstva-architektura-popis-vrstev/
- [12] MVC architektura. *ITnetwork* [online]. 2018 [cit. 2018-05-20]. Dostupné z: www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor

A Terminologický slovník

BKP	bezpečnostní kód poplatníka, používán pro evidence do EET
ČVUT	České vysoké učení technické v Praze
DAO	data access object
DIČ	daňové identifikační číslo
DPH	daň z přidané hodnoty
EAN	čárový kód
EET	elektronická evidence tržeb
FIK	fiskální identifikační kód, používán pro evidenci do EET
GNU	general public licence
ID	identifikační číslo
MIT	druh licence
MS-DOS	Microsoft Disk Operating System
MVC	druh softwarové architektury
REST	Representational State Transfer

B Instalační příručka

Je potřeba si stáhnout a nainstalovat:

Java 8

NetBeans 8.2

Apache Tomcat 8

Apache Maven 3

Spuštění aplikace:

Otevřete projekty v NetBeans 8.2, proveďte čistou kompilaci (občas je nutné dvakrát). Zapněte projekty z NeatBeans, nejdříve server, poté klienta. Výchozí uživatelský účet je admin, heslo 1234, heslo si co nejdříve změňte.

C Účtenka z e-pokladny

```
-----  
Market  
Uralská 448/11  
Praha 8 - 180 00  
Provozovna: 1  
Pokladna: 51  
Datum a čas: 18/5/2018 14:25:48  
Číslo účtenky: 12  
-----  
Zboží*ks (kg)  cena  DPH  cena s DPH  
-----  
Rohlík*5.0  
8.5Kč 15.0%DPH 10.0Kč s DPH  
Coca-cola 1L*1.0  
21.25Kč 15.0%DPH 25.0Kč s DPH  
Golden Snack 80g*1.0  
13.515Kč 15.0%DPH 15.9Kč s DPH  
-----  
SUMA s DPH: 51.0Kč  
-----  
EET  
FIK: aa2387c5-b086-4178-8e51-61f0d67  
a77b1-ff  
BKP: 2D4530BA-A5CC49FA-5CAC9F00-F1D3  
D740-9205FE38  
Režim tržby: běžný  
*****  
Děkujeme, přijďte zase!  
*****
```

Obrázek 5. Účtenka z e-pokladny

D Obsah přiloženého CD

- 1) Projekty ke spuštění
 - E-pokladna.zip
 - E-pokladna-ui.zip

- 2) Bakalářská práce
 - Bakalářka.pdf
 - Bakalářka.docx