



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Martin Vašina

System správy objednávek pro restaurační zařízení

Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: doc. RNDr. Petr Hnětynka, Ph.D.

Studijní program: Informatika

Studijní obor: Programování a softwarové systémy

Praha 2021

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Poděkování

Děkuji doc. RNDr. Petru Hnětynkovi, Ph.D. za vedení této práce, za jeho cenné připomínky a za pomoc s jejím vznikem. Dále děkuji svému bratrovi Tomáši Vašinovi a čajovně U Bílého draka za podnět k práci a zpětnou vazbu při užívání softwarového díla. Nakonec bych chtěl vyjádřit poděkování autorům softwaru třetích stran, bez nichž by tato práce nemohla vzniknout.

Název práce: Systém správy objednávek pro restaurační zařízení

Autor: Martin Vašina

Katedra: Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: doc. RNDr. Petr Hnětynka, Ph.D., Katedra distribuovaných a spolehlivých systémů

Abstrakt: Práce se zabývá implementací softwaru pro evidenci objednávek a tržeb v restauračním zařízení a je zaměřena především na podniky typu čajovna/kavárna, tedy zařízení která nejsou členěna na kuchyň a bar, ale mají pouze jednu přípravnu. Systém, umožňující rychlou a přehlednou evidenci objednávek a tržeb, pomáhá udržet přehled o financích podniku a požadavcích zákazníků. Navíc vhodný systém je klíčem ke kvalitnějším službám podniku. Na trhu již existuje mnoho produktů pro správu objednávek, avšak kvůli velké variabilitě vlastností těchto systémů je obtížné najít právě takový, který bude odpovídat přesným požadavkům podniku. V rámci této práce byl navržen a implementován systém Orsy. Dílo vzniklo na požadavky konkrétní čajovny, kde je nyní se spokojeností využíváno. Navíc tak přispívá k inovaci v oblasti informačních systémů pro restaurace.

Klíčová slova: restaurace, objednávka, POS, objednávkový systém

Title: A system for orders management in a restaurant

Author: Martin Vašina

Department: Department of Distributed and Dependable Systems

Supervisor: doc. RNDr. Petr Hnětynka, Ph.D., Department of Distributed and Dependable Systems

Abstract: The work is about the implementation of a system for orders management in a restaurant, and is focused primarily on companies such as teahouse/-cafe, ie facilities that are not divided into kitchen and bar, but have only one preparation room. The system, which enables fast and clear registration of orders and sales, helps to keep track of the company's finances and customer requirements. In addition, a suitable system is the key to better business services. There are already many order management softwares on the market, but due to the great variability of the properties of these systems, it is difficult to find just one that will meet the exact requirements of the company. Within this work was designed and implemented a system Orsy. The work was created for requirements of a specific tea house, where it is now used with satisfaction. In addition, the system contributes to innovation in the area of information systems for restaurants.

Keywords: restaurant, order, POS, ordering system

Obsah

1	Úvod	3
1.1	Struktura textu	3
2	Analýza požadavků	4
2.1	Popis a zaměření softwarového díla	4
2.2	Motivace ke vzniku softwarového díla	4
2.3	Požadavky na systém	4
2.3.1	Prostředí pro nasazení informačního systému	4
2.3.2	Definice pojmů	5
2.3.3	Požadované vlastnosti informačního systému	5
2.3.3.1	Otevírání a zavírání směn	5
2.3.3.2	Správa účtů a objednávek	6
2.3.3.3	Evidence tržeb	6
2.3.3.4	Grafická mapa stolů	6
2.3.3.5	Kreditní zákaznické účty	6
2.3.3.6	Zobrazení statistik a jejich export	6
2.3.3.7	Tisk účtenek	6
2.3.3.8	Měsíční výkazy tržeb	7
2.3.3.9	Evidence skladových položek	7
2.3.3.10	Zálohování dat systému	7
2.3.3.11	Elektronická evidence tržeb (EET)	7
2.3.3.12	Mobilní číšník	7
2.3.3.13	Další požadované vlastnosti	8
2.4	Uživatelé a interakce se systémem	8
3	Návrh systému	9
3.1	Hlavní funkcionalita	9
3.2	Prostředí aplikace	10
3.3	Použité technologie	10
3.3.1	Programovací jazyk - Java	10
3.3.2	Grafické uživatelské rozhraní - JavaFX	10
3.3.3	Databáze - HyperSQL	11
3.3.4	ORM framework - Hibernate	11
3.3.5	Další softwarové knihovny	11
3.3.6	Android	12
3.4	Apache Ant	13
3.5	Architektura projektu	13
3.5.1	Desktopová aplikace	14
3.5.2	Mobilní číšník	15
3.5.3	Databázové relační schéma	15
3.6	Komunikační rozhraní	16

3.6.1	Bezpečnost	17
3.7	Services	18
3.7.1	EETService	18
3.7.2	PrinterService	18
3.7.3	EmailService	18
3.7.4	BackupService	19
3.7.5	PDFService	19
3.8	Datové soubory	19
4	Použití systému	21
4.1	Podobné existující systémy	21
4.2	Instalace a spuštění systému	21
4.2.1	Desktopová aplikace	21
4.2.2	Mobilní číšník	22
4.3	Požadavky na zabezpečení dat	24
4.4	Analýza výkonnosti systému	24
4.4.1	Předpokládané nároky na výkon aplikace	24
4.4.2	Otestování výkonnosti aplikace	25
4.5	Reference	26
5	Závěr	28
	Seznam použité literatury	29
	Seznam obrázků	30
A	Přílohy	31
A.1	Obsah elektronické přílohy	31
A.2	Diagram komunikačního protokolu	32

1. Úvod

V restauračních zařízeních je žádoucí aby obsluha měla dokonalý přehled o objednávkách jednotlivých hostů. S tím souvisí vedení informací o tom, jaká objednávka patří ke kterému stolu a co má obsluha naučtovat zákazníkovi při placení. V základu by bylo možné si tyto informace pouze pamatovat a nebo je sepisovat na papír, avšak pro rychlejší zpracování služeb a pro menší pravděpodobnost chyby je vhodné využít počítačového programu. Tento software by měl poskytnout navíc i další funkcionalitu, jako například přehled o financích podniku, sledování prodejnosti produktů a tisk účtenek. Díky těmto vlastnostem je možné dosáhnout celkově lepší kvality služeb podniku.

Každý takovýto podnik navíc musí podávat hlášení správci daně o svých tržbách. V současné době byl státem zaveden systém EET (Elektronická evidence tržeb), který spravuje Finanční správa ČR. Je tedy přínosné aby objednávkový systém podniku rovnou umožňoval odesílání EET účtenek správci daně. Finanční správa nevyžaduje certifikaci dodavatele pokladních zařízení, a tedy softwarové nebo hardwarové řešení je oprávněn nabízet kdokoliv.[1]

Cílem této práce je navrhnout a implementovat objednávkový systém podle požadavků konkrétní čajovny. Tento systém ponese pracovní název Orsy (ze slov ORdering SYstem) a bude zaměřen především na jednodušší podniky typu čajovna/kavárna, tedy zařízení která nejsou členěna na kuchyň a bar, ale mají pouze jednu přípravnu.

1.1 Struktura textu

Text práce je rozdělen následujícím způsobem. V sekci 2. jsou diskutovány a rozebrány požadavky na informační systém určený pro správu objednávek v restauračním zařízení. Sekce 3. popisuje návrh a architekturu systému implementovaného podle požadavků z druhé sekce. Ve 4. sekci je poté popsáno praktické použití tohoto systému včetně otestování výkonnosti systému.

2. Analýza požadavků

2.1 Popis a zaměření softwarového díla

Jak již bylo zmíněno v úvodu, softwarové dílo je určeno pro správu objednávek v restauračním zařízení, a je zaměřeno především na podniky typu čajovna/kavárna, obecně na zařízení která nejsou členěna na bar a kuchyň, ale mají pouze jednu přípravnu, a tedy nevyžadují komplexnější distribuovaný systém.

2.2 Motivace ke vzniku softwarového díla

K tomuto tématu jsem se dostal z praktického důvodu potřeby nasazení systému správy objednávek pro jednu čajovnu. I když již podobných systémů na trhu existuje mnoho, všechny se liší v mnoha parametrech a vybrat takový systém, který by pokrýval všechny základní potřeby při zachování uživatelské přívětivosti a jednoduchosti, a s ohledem na cenu je docela problém.

Rozhodl jsem se tedy vzít si tuto problematiku jako téma a vytvořit implementaci přímo na míru, a pro potřeby konkrétního typu podniku.

2.3 Požadavky na systém

U větších restauračních zařízení, která jsou například členěna na bar a kuchyň, je vhodné mít informační systém distribuovaný pro více jednotlivých částí restauračního zařízení. Například v restauraci, která by byla členěna na hlavní bar, kuchyň a venkovní bar s terasou, by bylo vhodné navrhnout distribuovaný systém, který bude mít na jednom počítači hlavní server s databází, a na barech a v kuchyni budou klientské aplikace komunikující s tímto serverem.

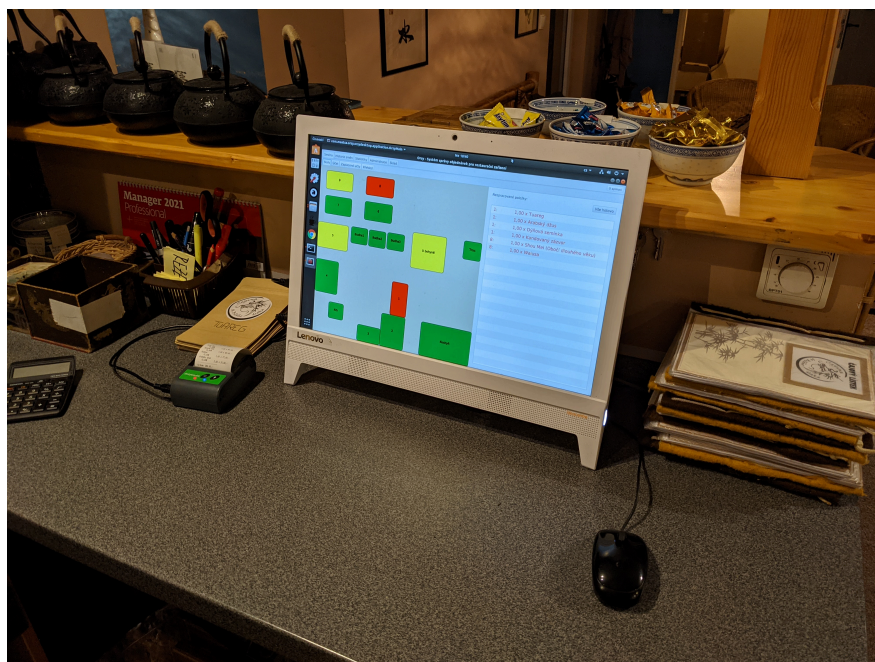
V menších restauračních zařízeních jako jsou čajovny/kavárny postačuje mít objednávkový informační systém na jednom počítači. Obsluha, která pracuje s tímto počítačem, používá jeden program ve kterém eviduje vše potřebné.

Následující požadavky byly posbírány na základě diskuse s provozovatelem čajovny.

2.3.1 Prostředí pro nasazení informačního systému

Informační systém by měl být nasazen na klasickém osobním počítači (PC), předpokládá se použití počítače typu All In One PC. Na obrázku 2.1 je ukázka z prostředí pro nasazení systému.

Aplikace by měla být pokud možno nezávislá na použitém operačním systému a tedy by neměla klást žádné další technické a nebo finanční požadavky na základní softwarové vybavení počítače.



Obrázek 2.1: Příklad sestavy s informačním systémem

2.3.2 Definice pojmů

Směna označuje souvislé období práce, zpravidla odpovídá jednomu kalendářnímu dni. *Směna* může být buď ve stavu *otevřená* nebo *uzavřená*.

Položka je prvek jídelního lístku nebo *jednotková cena*. Každá *položka* obsahuje název, cenu a sazbu DPH.

Objednávka označuje množinu *položek* přidanych na účet v rámci *směny*. Každá *objednávka* může být buď ve stavu *nezpracovaná* nebo *zpracovaná*.

Jednotková cena je speciální druh *položky*, která není v jídelním lístku. Umožňuje rychle vytvořit *objednávku* s konkrétní cenou bez nutnosti vytvářet novou *položku* v jídelním lístku s touto cenou.

Skladová položka je prvek jídelního lístku, u kterého systém eviduje množství. Při zaplacení *objednávky* s touto položkou dojde k odečtení množství.

2.3.3 Požadované vlastnosti informačního systému

2.3.3.1 Otvírání a zavírání směn

V jeden okamžik může být otevřena nejvýše jedna směna. Směnu tedy bude možné otevřít pouze pokud předchozí směna byla uzavřena. Směnu bude možné uzavřít pouze pokud nebudou existovat žádné objednávky.

Nebude-li žádná směna otevřena, pak nebude možné vytvářet objednávky.

2.3.3.2 Správa účtů a objednávek

Na stolech půjde vytvářet účty s daným jménem. Účet bude možné přejmenovat, odstranit, sloučit s jiným účtem, přesunout na jiný stůl, oddělit položky nebo zaplatit.

Na účet bude možné přidat objednávku. Položky, budou rozděleny do kategorií. Po rozkliknutí kategorie se zobrazí seznam položek v kategorii a po kliknutí na danou položku dojde k přidání této položky na účet.

Při platbě bude na výběr zda zaplatit celý účet nebo pouze část účtu. Platbu bude možné provést hotově, bezhotovostně pomocí externího platebního terminálu a nebo kreditem ze zákaznické karty. K dispozici bude zobrazení přehledu zaplacených účtů v čteně jejich objednávek.

2.3.3.3 Evidence tržeb

System na každé směně bude evidovat tržbu. Přehled tržeb bude možné zobrazit v historii směn, případně ve statistikách.

2.3.3.4 Grafická mapa stolů

Konfigurovatelná mapa stolů odpovídající rozložení stolů v provozovně. Po kliknutí na stůl se zobrazí účty a objednávky na daném stole

2.3.3.5 Kreditní zákaznické účty

System bude umožňovat pro pravidelné zákazníky mít vložený kredit, kterým lze platit objednávky. Kredit bude možné na účet vkládat a vybírat z účtu. Také bude k dispozici zobrazení historie provedených plateb z dané zákaznické karty.

2.3.3.6 Zobrazení statistik a jejich export

Pro optimalizaci služeb podniku je vhodné mít zpětnou vazbu. K tomu poslouží generování statistik z dat uložených v databázi. System by měl umět vygenerovat alespoň základní statistiky tržeb a prodejnosti položek. Navíc by tato vygenerovaná data mělo být možné exportovat do vhodného formátu, který bude uživatelsky přívětivý a nebude závislý na softwaru ani hardwaru. Příkladem takového formátu je PDF¹.

2.3.3.7 Tisk účtenek

K systému bude možné připojit tiskárnu, na které bude po zaplacení účtu vytisknuta účtenka. Účtenku půjde vytisknout i zpětně z historie provedených plateb na směně. U účtenky by se měla dát nastavit hlavička účtenky a zda tisknout seznam položek na účtence. Cena za platbu objednávky bude tisknuta vždy.

¹PDF - Portable Document Format je souborový formát pro ukládání dokumentů nezávisle na softwaru i hardwaru

2.3.3.8 Měsíční výkazy tržeb

Zobrazení přehledu tržeb v rámci měsíců. Pro každý měsíc ve kterém byla vytvořena alespoň jedna platba, dojde k vytvoření výkazu obsahujícím sumu všech provedených plateb v tomto měsíci.

Přínosná by byla možnost odeslání měsíčního výkazu emailem přímo z aplikace.

2.3.3.9 Evidence skladových položek

Systém bude podporovat funkci skladu. Z jídelního lístku bude možné vybrat množinu skladových položek. U těchto položek bude evidováno množství. Při zaplacení objednávky s danou položkou dojde k odečtení množství. Skladové položky lze naskladnit (přidat množství) nebo vyskladnit (odebrat množství).

2.3.3.10 Zálohování dat systému

Podpora pro vytváření záloh databáze a datových souborů. Zálohování může být například založeno na nakopírování dat na externí úložiště vždy po uzavření směny.

2.3.3.11 Elektronická evidence tržeb (EET)

V České republice byla od roku 2016 zavedena elektronická evidence tržeb. Ta spočívá v odesílání informací o každé transakci obchodníka do systému Finanční správy ČR. Je tedy velmi přínosné, aby systém implementoval funkcionalitu pro odesílání a vydávání EET účtenek. Systém bude umožňovat nakonfigurování EET klienta. Po vytvoření platby dojde k odeslání EET účtenky do systému správce daně a následně bude vytištěna účtenka s povinnými údaji EET.

2.3.3.12 Mobilní číšník

Součástí systému by měl být mobilní číšník, tedy aplikace pro mobilní telefony, kterou bude možné připojit prostřednictvím internetu, a pomocí které bude možné vytvářet vzdáleně objednávky například přímo od stolu.

Aplikace by měla splňovat tyto základní požadavky:

- Obrazovka pro připojení k serveru / hlavnímu programu
- Zobrazení seznamu stolů
- Zobrazení seznamu účtů na daném stole
- Vytvoření a odeslání nové objednávky

Příkladem použití mobilního číšníka může být následující scénář:

Obsluha bude mít nainstalovanou aplikaci mobilního číšníka například v mobilním telefonu. Desktopová aplikace i mobilní telefon budou připojeni ke stejné WiFi síti. Obsluha na mobilním telefonu otevře aplikaci mobilního číšníka a zadá IP adresu počítače, na kterém běží desktopová aplikace. Pokud bude otevřena směna, tak dojde k připojení mobilní aplikace k serveru desktopové aplikace.

Po připojení se zobrazí na mobilní aplikaci seznam stolů. Po kliknutí na stůl se zobrazí objednávky na posledním účtě přidaném na daný stůl. Seznam účtů bude možné zobrazit a změnit pomocí rozbalovacího seznamu v horní části obrazovky. V dolní části obrazovky bude tlačítko „Nová objednávka“. Po kliknutí na toto tlačítko se zobrazí dva seznamy. Jeden seznam bude obsahovat položky vytvářené objednávky. Druhý seznam bude obsahovat seznam kategorií, ze kterých bude možné rozkliknutím přidávat nové položky do objednávky. Tlačítkem „Odeslat“, dojde k odeslání objednávky na server. Po přijetí objednávky serverem dojde v desktopové aplikaci k informování, že na daném stole byla přijata nová objednávka.

2.3.3.13 Další požadované vlastnosti

Dalšími požadavky na systém je důraz na uživatelskou přívětivost, intuitivní ovládání a jednoduchou administraci. Speciálním požadavkem zákazníka je absence přihlašování a nutnosti používání hesel a absence vedení účtů pracovníků na směnách.

2.4 Uživatelé a interakce se systémem

Vlastník restaurace nebo vedoucí osoba provozu, zajistí instalaci systému v restauračním zařízení včetně jeho nakonfigurování.

Na počátku každé směny vedoucí této směny spustí desktopovou aplikaci a otevře novou směnu. Obsluhující personál během směny využívá systém především pro správu objednávek, tedy vytváří nové objednávky, manipuluje s objednávkami v systému a provádí platby objednávek. Většinu těchto operací provádí personál přes desktopovou aplikaci. Pro vytváření objednávek může personál používat zařízení (například mobilní telefon) s aplikací mobilního číšníka. Na konci směny, vedoucí této směny zkontroluje stav kasy s tržbami uvedenými v systému a uzavře směnu.

Vedoucí osoba provozu nebo jiná odpovědná osoba podle potřeby exportuje měsíční výkazy, nahlíží do statistik a historie směn a obsluhuje administraci systému.

3. Návrh systému

V rámci této práce byl vytvořen informační systém s pracovním názvem Orsy, který je dostupný v příloze A.1.

System obsahuje dvě části:

- Desktopová aplikace - Java aplikace určená pro počítače s operačním systémem Linux nebo Windows.
- Mobilní číšník - aplikace určená pro mobilní zařízení s operačním systémem Android

3.1 Hlavní funkcionalita

System implementuje všechny požadavky, které byly podrobněji analyzovány v předchozí kapitole 2.3.3

Zde je pro přehled seznam vlastností systému:

- Správa účtů a objednávek
- Mapa stolů
- Otevírání a zavírání směn, evidence tržeb
- Kreditní zákaznické účty
- Mobilní číšník
- Statistiky a jejich export do PDF
- Elektronická evidence tržeb (EET)
- Tisk účtenek na systémové tiskárně
- Měsíční výkazy
- Evidence skladových položek
- Zálohování

Detailní popis používání všech možností systému je k dispozici v uživatelské příručce, která je součástí přílohy A.1.

3.2 Prostředí aplikace

Desktopová aplikace je určena pro nasazení na běžném osobním počítači. Není závislá na typu operačního systému, ale pro svůj běh vyžaduje instalaci Java Virtual Machine. Aplikace je primárně určena pro operační systémy Linux, Windows a všechny další systémy, pro které je k dispozici běhové prostředí Javy.

Mobilní klient je určen pro zařízení s operačním systémem Android. K desktopové aplikaci je možné jej připojit prostřednictvím internetu.

3.3 Použité technologie

V této sekci budou rozebrány konkrétní technologie použité v softwarovém díle. U každé technologie bude popsán její účel a případně i porovnání s možnými alternativami.

3.3.1 Programovací jazyk - Java

Při výběru programovacího jazyka byly kladeny nároky na podporu objektivě orientovaného programování a také na dostatečnou popularitu jazyka, se kterou je spojeno větší množství nástrojů usnadňujících vývoj jako jsou například různá vývojová prostředí, buildovací nástroje a v neposlední řadě softwarové knihovny třetích stran. Především kvůli osobním preferencím byl zvolen programovací jazyk Java. Další kandidáti byli C# a C++.

Java nabízí robustnost a nezávislost na architektuře. Další bezespornou výhodou je rozsáhlý opensource ekosystém Javy, který přináší velké množství frameworků a softwarových knihoven. Dále Java nabízí automatickou správu paměti, která omezuje množství chyb.

Výhodou Javy je i její podpora operačním systémem Android. Mobilní číšník pro tuto platformu bude moci být také naprogramován v Javě, a tedy desktopová aplikace a Android klient budou moci sdílet část kódu.

Jazyk C# je především určen pro vývoj softwaru na platformách Microsoftu. Především je vhodnější pro vývoj her, nežli informačních systémů.[2]

C++ by nabízelo možnost kompilace do nativního kódu, která by zjednodušovala instalaci.

3.3.2 Grafické uživatelské rozhraní - JavaFX

Java nabízí dva populární frameworky pro vytváření grafického uživatelského rozhraní (GUI).

Jedním z nich je starší framework Swing, který začal být vyvíjen v roce 1997 a je nedílnou součástí Java SE od verze 1.2.

Naproti tomu je o 10 let mladší framework JavaFX. Od verze Java 11 byl odstraněn ze standardní distribuce Javy, a pro použití aplikace s JavaFX je tedy

nutné stáhnutí externí knihovny. Velikou výhodou JavaFX je podpora architektury MVC¹. Další výhodou JavaFX je volně dostupný a uživatelsky přívětivý drag&drop editor JavaFX Scene Builder, který umožňuje vytváření grafického uživatelského rozhraní v JavaFX aplikacích bez potřeby psaní kódu.

Vzhledem k těmto dvěma zmíněným výhodám byl použit framework JavaFX.

3.3.3 Databáze - HyperSQL

Velmi důležitou součástí aplikace je databáze, která musí zajistit konzistentní uchování dat a zabránit jejich poškození a ztrátě. Pro tento účel je vhodné využít nějakou SQL databázi. Databázové stroje mohou různě podporovat dva základní režimy „client/server“ a „embedded“. První typ databází běží jako samostatný proces a vyžadují komplexnější instalaci. Embedded databáze je integrována jako součást aplikace.

Při výběru databázového stroje byly brány požadavky na kompatibilitu s jazykem Java a také aby databáze umožňovala být zaintegrována přímo do aplikace.

S ohledem na tyto požadavky se nabízelo několik možností jako jsou HyperSQL, SQLite, H2, Apache Derby a další. Vybrána byla nakonec databáze HyperSQL v režimu ukládání na disk. Malý a rychlý databázový stroj HyperSQL je napsaný v Javě a podporuje dialekt SQL definovaný standardy SQL 92, 1999, 2003 a 2008.[3] Navíc umožňuje spuštění ve dvou režimech a to jak v paměti, tak přímo na disku.

3.3.4 ORM framework - Hibernate

Při použití relační databáze a objektově orientovaného programovacího jazyka je velmi vhodné využít techniku objektově relačního mapování (ORM), která zajišťuje automatickou konverzi dat mezi používanými objekty (entitami) a jejich reprezentací v databázovém systému. Pro Javu existuje mnoho různých ORM frameworků. Při vývoji tohoto systému byl použit framework Hibernate, který je jeden z nejpopulárnějších ORM frameworků.

3.3.5 Další softwarové knihovny

Mezi požadavky na systém je i generování PDF dokumentů (viz 2.3.3.6), odesílání emailů (viz 2.3.3.8) a implementace elektronické evidence tržeb (viz 2.3.3.11). Pro tuto funkcionalitu není přímo ve standardní knihovně Javy podpora, a tedy bylo nutné využít externích knihoven.

- OpenPDF - Pro generování PDF dokumentů byla použita opensource knihovna OpenPDF², která vychází z knihovny iText³ verze 4. Knihovna

¹MVC - softwarová architektura Model-View-Controller

²<https://github.com/LibrePDF/OpenPDF>

³<https://itextpdf.com>

iText nabízí velmi bohaté prostředky pro práci s PDF dokumenty, včetně kvalitní dokumentace, avšak její použití je omezeno licencí AGPL[4].

Knihovna OpenPDF dostatečně pokrývá potřebné funkce pro generování PDF dokumentů, a navíc je oproti knihovně iText dostupná pod „svobodnější“ licencí LGPL[5].

- Jakarta Mail - Knihovna Jakarta Mail⁴ slouží k odesílání a přijímání emailů prostřednictvím protokolů SMTP, POP3 a IMAP. Jakarta Mail je volně dostupná opensource knihovna, která je obsažena v platformě Java Enterprise Edition (Java EE). Použití v JavaSE vyžaduje stažení externí knihovny.

V desktopové aplikaci je využita k odesílání měsíčních výkazů emailem prostřednictvím SMTP serveru.

- eet-client - Systém elektronické evidence tržeb (EET) využívá Webovou Službu SOAP⁵. Dokument WSDL⁶ definující rozhraní pro komunikaci s EET⁷ systémem je dostupný na URL adrese:

- Playground:

<https://pg.eet.cz/eet/services/EETServiceSOAP/v3?wsdl>

- Production:

<https://prod.eet.cz/eet/services/EETServiceSOAP/v3?wsdl>

Podle tohoto dokumentu by bylo možné implementovat vlastního klienta pro komunikaci s rozhraním Finanční správy. Avšak EET klient pro Javu již existuje, takže bylo možné využít volně dostupné knihovny. Zdrojové soubory včetně dokumentace této knihovny jsou dostupné na adrese GitHubu: <https://github.com/todvora/eet-client>

Další informace ohledně EET lze nalézt na oficiálním portálu⁷.

3.3.6 Android

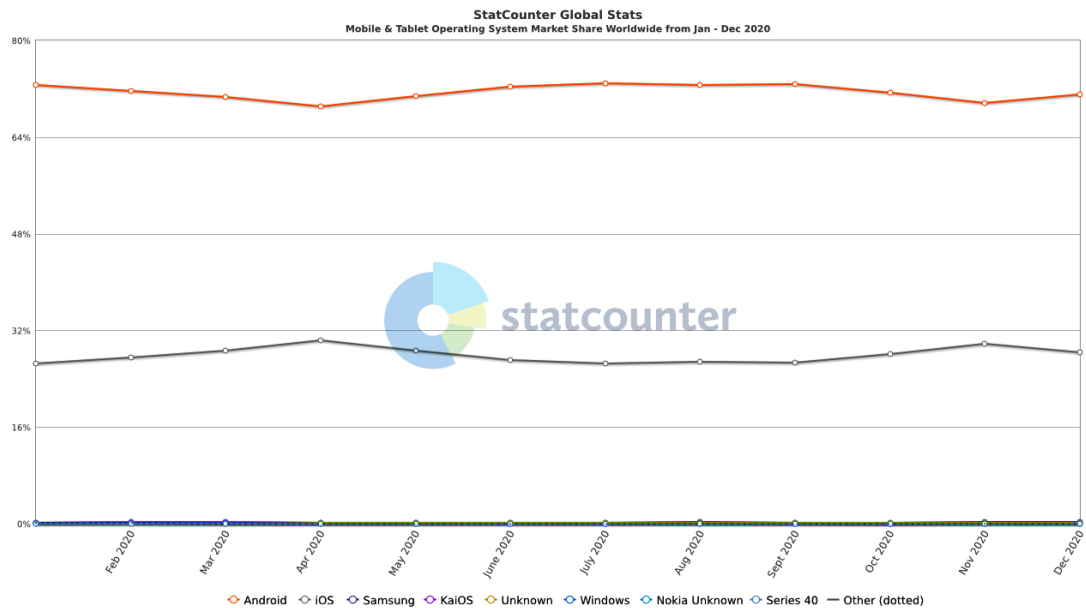
Platformou pro implementaci mobilního číšníka byl vybrán operační systém Android. Mezi dalšími možnostmi bylo v nabídce především iOS a dále Samsung, KaiOS, Windows a další minoritní zástupci. Hlavní předností operačního systému Android je jeho vysoký poměr zastoupení na trhu mobilních operačních systémů. V roce 2020 činil tento podíl na trhu pro mobilní telefony a tablety v průměru více jak 69%. Údaj, včetně následujícího grafu 3.1 pochází ze serveru StatCounter[6].

⁴<https://eclipse-ee4j.github.io/mail>

⁵SOAP - Simple Object Access Protocol je protokolem pro výměnu zpráv založených na XML prostřednictvím sítě

⁶WSDL - Web Services Description Language je jazyk ve formátu XML pro popis webových služeb

⁷<https://www.etrzby.cz>



Obrázek 3.1: Graf podílu mobilních operačních systémů na celosvětovém trhu v roce 2020

Tento značný podíl zaručuje velikou základnu mobilních zařízení, na které bude možné aplikaci mobilního číšníka nainstalovat.

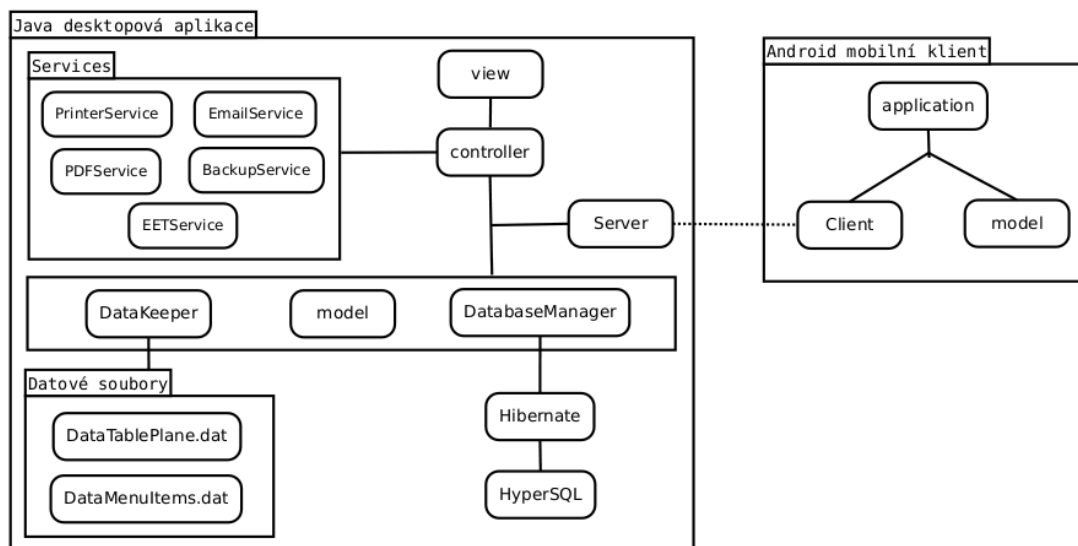
Aplikace pro Android mohou být primárně vyvíjeny v jazycích Kotlin, Java nebo C++.[7] V současnosti Google preferuje pro vývoj Android aplikací jazyk Kotlin[8]. Vzhledem k osobním preferencím a tomu, že desktopová aplikace je naprogramovaná v jazyce Java, tak pro vývoj mobilního číšníka byl také zvolen jazyk Java.

3.4 Apache Ant

Apache Ant je nástroj pro sestavování softwarových aplikací. Součástí zdrojového kódu desktopové aplikace je konfigurační soubor pro Ant build.xml. Pomocí tohoto nástroje lze zkompilovat zdrojové soubory, vytvořit archivy pro distribuci, spustit aplikaci a také je možné vygenerovat programátorskou dokumentaci ve formátu JavaDoc.

3.5 Architektura projektu

Na následujícím diagramu 3.2 je zobrazena struktura celého systému. Ta zahrnuje dvě základní části a to desktopovou aplikaci a mobilního číšníka, které komunikují pomocí protokolu popsaného v kapitole 3.6.



Obrázek 3.2: Diagram architektury projektu

3.5.1 Desktopová aplikace

Desktopová aplikace je napsaná v programovacím jazyce Java a grafické uživatelské rozhraní je postaveno na softwarové platformě JavaFX. Výhodou JavaFX je podpora softwarové architektury Model-View-Controller (MVC).

MVC[9] je architektonický vzor, který rozděluje aplikaci do tří logických komponent: Model, View a Controller. Model obsahuje datovou logiku systému. Zahrnuje databázové dotazy, výpočty, validace a podobně. View se stará o zobrazení výstupu uživateli. Controller působí na rozhraní mezi View a Model. Zajišťuje interakci mezi uživatelským rozhraním a datovou logikou.

View: V aplikaci je komponenta View tvořena především šablonami ve formátu FXML. FXML je jazyk platformy JavaFX postavený na technologii XML⁸. Základem uživatelského rozhraní je komponenta TabPane, která rozděluje aplikaci do několika karet (Tab). Každá tato karta je definována ve vlastním FXML souboru. Dále ve zdrojovém kódu aplikace jsou použity 3 generické šablony pro vytváření dialogů. Tyto šablony také spadají do komponenty View.

Model: Model aplikace je tvořen objekty v balíku com.maatsa.orsy.model.

Dále aplikace pro přístup a správu dat používá 2 třídy DataKeeper a DatabaseManager. Datakeeper obsluhuje data uložená v souborech DataMenuItems.dat a DataTablePlane.dat. DatabaseManager je třída obsluhující data uložená v databázi.

⁸XML - Extensible Markup Language je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C

Controller: Ke každé kartě (Tabu definovaného FXML souborem) je přiřazen vlastní controller, který obsluhuje požadavky z uživatelského rozhraní.

Pro datovou logiku využívají především instance tříd DataKeeper a Database-Manager.

Controllery využívají i dalších služeb, které jsou rozebrány v sekci 3.7.

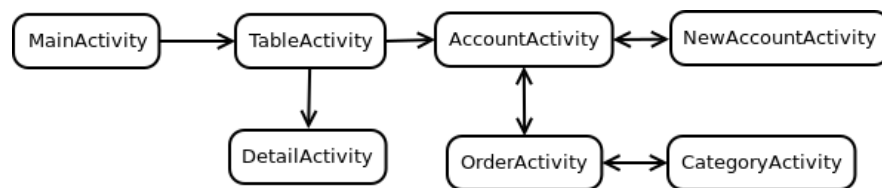
Server: V samostatném vlákně, které je vytvořeno a spuštěno po inicializaci aplikace, běží server, který odpovídá na požadavky klientů podle komunikačního protokolu (viz 3.6).

3.5.2 Mobilní číšník

Mobilní číšník je aplikace pro Android podporující verze 4.4 a vyšší. Aplikace je naprogramovaná v jazyce Java.

Základem uživatelského rozhraní v Android aplikacích je komponenta Activity, která deklaruje jednu obrazovku uživatelského rozhraní. Rozložení prvků na obrazovce je popsáno zvlášť v souborech ve formátu XML.

Na následujícím diagramu 3.3 jsou zobrazeny všechny Activity v aplikaci, včetně přechodů mezi nimi.



Obrázek 3.3: UML Activity Diagram

Singleton třída Client zajišťuje komunikaci se serverem. Při požadavku nějaké Aktivity se klient připojí k serveru a podle komunikačního protokolu si předají serializované objekty.

Balík com.maatsa.orsy.model obsahuje datové objekty a je shodný s balíkem v desktopové aplikaci.

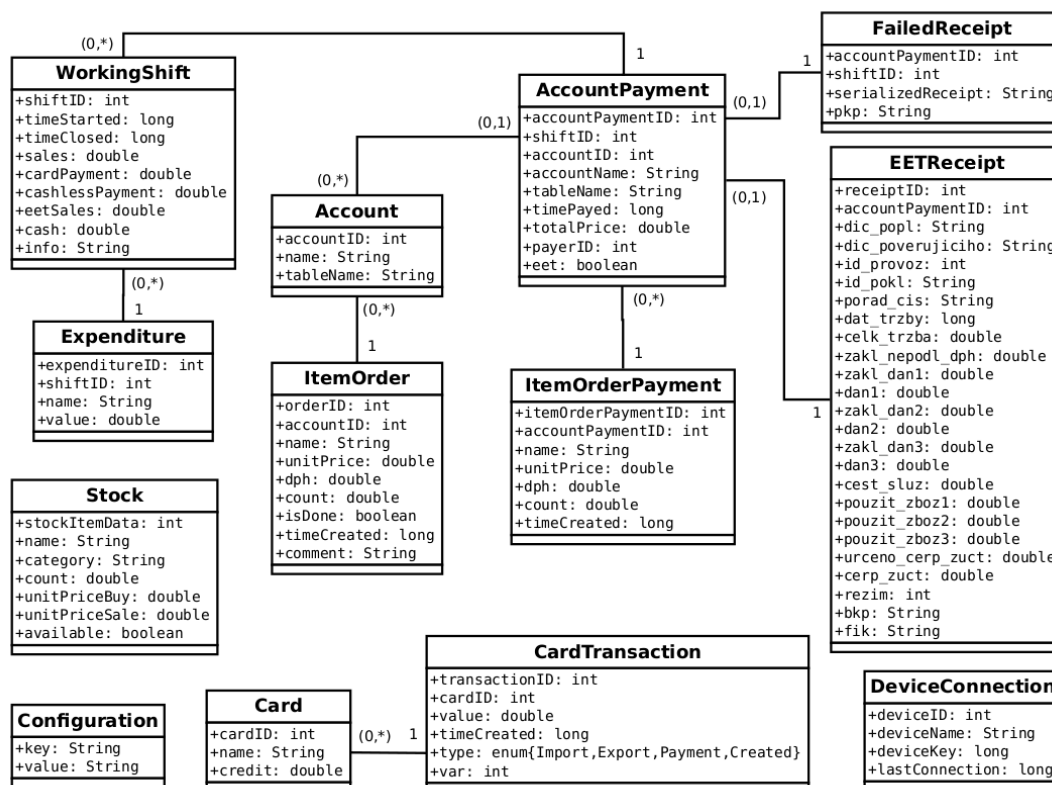
3.5.3 Databázové relační schéma

Desktopová aplikace používá HyperSQL databázi v režimu ukládání do souboru na disk. Databáze je přímo integrována do aplikace, a tedy neobsahuje server běžící jako samostatný proces, ale přístup k databázi je vykonáván z procesu, který volá funkce knihovny HyperSQL. To umožňuje aplikaci větší kontrolu nad databází, především si aplikace při prvním spuštění může databázi sama vytvořit a nakonfigurovat, a není tak nutné databázi instalovat a konfigurovat zvlášť.

Schéma databáze je inicializováno při spuštění aplikace. Pokud na disku neexistuje databáze, tak dojde k jejímu vytvoření a inicializaci databázového schéma.

Databázové tabulky jsou mapovány na objekty/entity pomocí frameworku Hibernate.

Typy objektů uložených v databázi a vztahy mezi nimi znázorňuje následující UML⁹ diagram 3.4.



Obrázek 3.4: Databázové relační schéma

3.6 Komunikační rozhraní

K desktopové aplikaci lze připojit Android klienta. Mobilní klient má omezené funkce pouze na vytváření nových účtů a objednávek. Komunikace je založena na TCP socketech a využívá serializovaných Java objektů. V rámci desktopové aplikace běží ServerSocket, který obsluhuje požadavky klientů. Při každém spojení je vytvořen nový Socket, ve kterém dojde podle komunikačního protokolu k předání dat, a poté je zase uzavřen.

Komunikační protokol má dvě části. V první části dojde k předání celočíselného kódu charakterizujícího co se bude dít v části druhé, kde poté podle kódu dojde k předání dat ve formě posloupnosti serializovaných Java objektů.

V příloze A.2 je k dispozici vizualizace komunikačního protokolu pomocí sekvenčního diagramu.

Aby se k desktopové aplikaci nemohlo připojit libovolné zařízení s nainstalovanou aplikací klienta, tak bylo nutné přidat proces autentizace. Jedna z možností by byla zadávání bezpečnostního klíče, avšak v rámci uživatelské přívětivosti byla

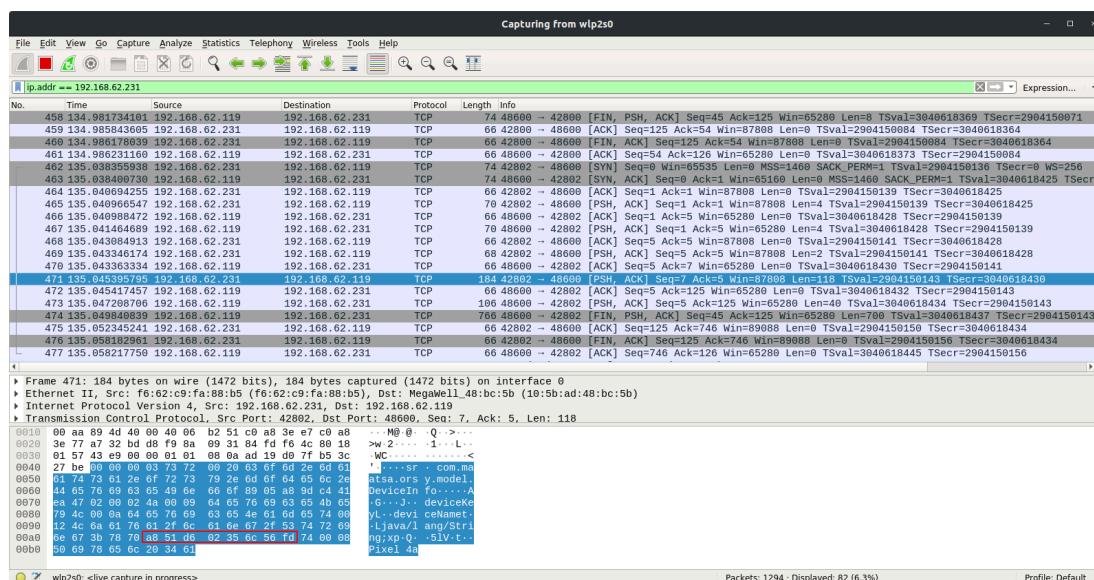
⁹UML - Unified Modeling Language je modelovací jazyk pro grafické zobrazení návrhu systému.

zvolena následující varianta: Klientská aplikace si po nainstalování vygeneruje náhodný bezpečnostní klíč, pomocí kterého bude vždy identifikována. Při každém spojení v první fázi klient odešle tento svůj bezpečnostní klíč. Server si v databázi eviduje whitelist bezpečnostních klíčů, podle kterých při každém spojení ověří klienta. Zaregistruje-li server neznámý klíč, vyžádá si od klienta informace o zařízení a zeptá se uživatele zda jej chce přidat na whitelist. Ve whitelistu lze vidět klientské aplikace a čas jejich posledního spojení; nevhodná zařízení lze odebrat.

3.6.1 Bezpečnost

Vhodným způsobem jak otestovat bezpečnost aplikace, je využití penetračního testování. Tímto typem testování se simulují možné útoky na daný systém.

Na následujícím obrázku 3.5 je příklad zachycení bezpečnostního klíče klientské aplikace z TCP paketu.



Obrázek 3.5: Zachycení bezpečnostního klíče z TCP paketu programem Wireshark

V aktuální verzi aplikace (která je součástí přílohy) je komunikace nešifrovaná. Pro zamezení případu, že by někdo mohl odposlouchávat komunikaci se nyní očekává, že klient i server poběží v oddělené síti, kam nikdo jiný nemá přístup.

Nyní jsou pro komunikaci využívány třídy Socket a ServerSocket. Vhodnější by bylo využít tříd SSLSocket a SSLServerSocket, které implementují šifrovanou komunikaci. Případně celkově předělat komunikační protokol a pro komunikaci využít technologie Webových Služeb, například protokol SOAP, který by umožňoval implementaci mobilního číšníka i na další platformy nepodporující Javu.

Tato oblast je tak možnou iniciativou pro budoucí práci.

3.7 Services

Desktopová aplikace využívá další služby pomocí tříd, které jsou rozepsány v následujících podsekcích.

3.7.1 EETService

Systém podporuje odesílání EET transakcí podle požadavků, které byly rozebrány v sekci 2.3.3.11). Pro komunikaci se systémem Finanční správy byla použita knihovna `eet-client` (viz 3.3.5).

Inicializace klienta vyžaduje certifikáty klienta i serveru pro podepisování a ověřování zpráv. Klientský certifikát má podnik svůj vlastní, který dostane přidělený od Finanční správy, a je nutné jej přidat v administraci aplikace. Certifikáty serveru jsou obsaženy v rámci knihovny `eet-client`.

EET zahrnuje dva režimy, produkční (PRODUCTION) a vývojový (PLAYGROUND), a dále pro každý režim jsou k dispozici dva komunikační módy, reálný (REAL) a testovací (TEST).[1]

V aplikaci je nastaven jako výchozí produkční režim. Pro přepnutí na vývojový režim lze přidat argument aplikace „playground“ (viz programátorská dokumentace v příloze A.1).

Testovací mód umožňuje otestování spojení aplikace se serverem EET pomocí testovací účtenky. Reálný mód slouží pro běžné odesílání EET účtenek.

V případě výpadku připojení je systém EET navrhnut tak, že dojde k vydání „offline“ účtenky a od restauračního softwaru se poté očekává, že tuto účtenku uloží do databáze a odešle ji znovu, jakmile bude mít systém opět přístup k Internetu.

Třída `EETService` implementuje metody pro inicializaci EET klienta a pro vytvoření a následné odeslání EET účtenky, včetně podpory „offline“ režimu. Dále implementuje metodu pro odeslání testovací účtenky, která umožňuje ověřit správné nakonfigurování EET klienta.

3.7.2 PrinterService

Třída `PrinterService` implementuje metody pro zobrazení seznamu systémových tiskáren, tisk textu na dané tiskárně a vytisknutí účtenky zaplacené objednávky. Viz požadavky v sekci 2.3.3.7.

3.7.3 EmailService

Třída `EmailService` obsahuje statickou metodu pro odeslání emailu ze SMTP serveru. Využita je knihovna `Jakarta Mail` (viz 3.3.5). Aplikace podporuje port 465 s šifrováním SSL a port 587 se zabezpečením `StartTLS`. [10] Odeslat email lze tedy například ze serveru `smtp.seznam.cz:465`, `smtp.google.com:587` nebo `smtp.mail.com:587`.

3.7.4 BackupService

V administraci aplikace lze nastavit automatické vytváření zálohy dat systému, které je provedeno vždy po uzavření směny (viz požadavky v sekci 2.3.3.10). Třída BackupService implementuje metody pro nakopírování databáze a datových souborů do daného úložiště.

3.7.5 PDFService

Třída PDFService obsahuje statickou metodu, která generuje PDF dokument obsahující statistiky přehledu tržeb a prodejnosti položek (viz požadavky v sekci 2.3.3.6). Pro vytváření PDF dokumentu využívá knihovny OpenPDF (viz 3.3.5).

3.8 Datové soubory

Informace o konfiguraci mapy stolů a jídelního lístku jsou uloženy v textových souborech DataTablePlane.dat a DataMenuItems.dat. Při startu desktopové aplikace jsou data načtena do odpovídajících objektů. V administračním prostředí jsou pro každý soubor uživatelsky přívětivé editory, které načtou data ze souboru a po změně je zase uloží a aktualizují právě používané instance.

Mapa stolů je reprezentována dvěma celými čísly (šířka, výška) a množinou stolů, kde každý stůl obsahuje informace (jméno, souřadnice levého horního rohu, souřadnice pravého dolního rohu). Každý stůl má jedinečné jméno (při kolizi jmen je použit pouze první stůl s daným jménem a zbytek je ignorován).

Formát souboru DataTablePlane.dat:

```
šířka
výška
jméno stolu 1
x1
y1
x2
y2
jméno stolu 2
x1
y1
x2
y2
...
```

Lístek menu je tvořen kategoriemi obsahujícími položky (jméno, cena, sazba DPH v procentech - pouze celé číslo). Každá položka má unikátní jméno (v případě kolize jsou další ignorovány). Každá kategorie má také unikátní jméno, ale zde v případě kolize jmen dojde ke sloučení kategorií. Jméno, cena a DPH položky jsou odděleny středníkem.

Formát souboru DataMenuItems.dat:

jméno kategorie 1

položka v kategorii 1; cena; sazba DPH

položka v kategorii 2; cena; sazba DPH

položka v kategorii 3; cena; sazba DPH

...

jméno kategorie 2

položka v kategorii 1; cena; sazba DPH

položka v kategorii 2; cena; sazba DPH

...

4. Použití systému

4.1 Podobné existující systémy

Na trhu existuje již mnoho různých systémů pro restaurace. Např.: Dotykačka[11], POS Expert[12], Gastro[13], Datona[14], Agnis Pokladna[15], AWIS[16], FINTA[17], Fusion[18], Menu55[19], ...

Všechny tyto systémy nabízejí společnou základní vlastnost - správa objednávek a účtů. Z tohoto pohledu jsou prakticky totožné, avšak liší se především způsobem nasazení aplikace, uživatelským rozhraním a cenou. Většina těchto systémů navíc vyžaduje speciální hardware. Dále mohou různě podporovat například následující vlastnosti: správa skladu, mobilní klient, zákaznické účty, grafická mapa stolů, a další.

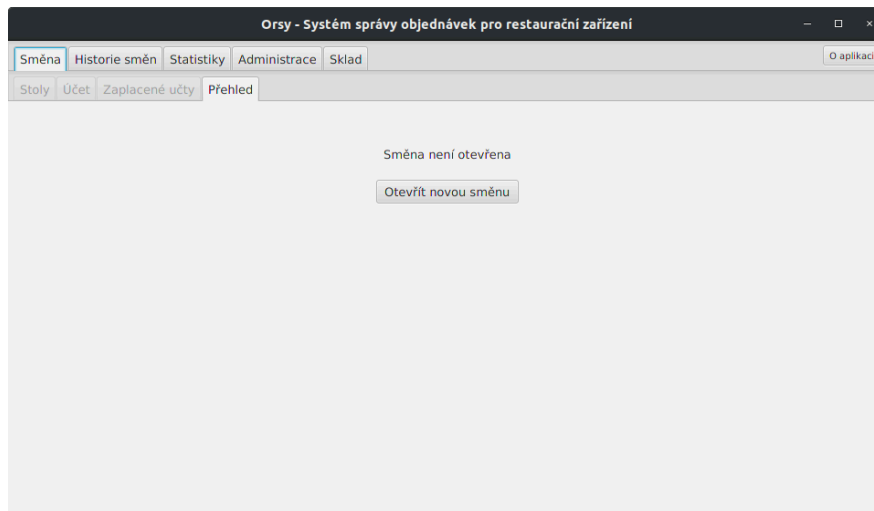
Při výběru optimálního systému musí uživatel brát v úvahu mnoho parametrů a nakonec většinou přistoupit k nějakému kompromisu. Systém Orsy vznikl na konkrétní požadavky čajovny, kde je nyní se spokojeností využíván, a jeho předností je jednoduchá instalace a administrace při zachování všech podstatných funkcí, které jsou vyžadovány v menších restauračních zařízeních. Navíc tento systém přispívá k inovaci v oblasti informačních systémů pro restaurace.

4.2 Instalace a spuštění systému

Projekt obsahuje dvě části, desktopovou aplikaci a mobilního číšníka. Desktopovou aplikaci je možné nainstalovat na počítač s operačním systémem Linux nebo Windows. Aplikaci mobilního číšníka je možné nainstalovat na zařízení s operačním systémem Android.

4.2.1 Desktopová aplikace

Po spuštění desktopové aplikace se zobrazí úvodní obrazovka s přehledem směny. Pokud se aplikace spouští poprvé, tak směna je uzavřená. Na obrázku 4.1 je obrazovka aplikace po prvním spuštění.



Obrázek 4.1: První spuštění desktopové aplikace

Okno programu je členěno na jednotlivé karty podle logických částí systému. V sekci Administrace je možné nakonfigurovat jídelní lístek, plánek stolů, tisk účtenek, EET a zálohování. Dále je v Administraci možné zobrazit přehled měsíců a spravovat zákaznické karty. V sekci Sklad lze nastavit a spravovat skladové položky. Sekce Statistiky umožňuje generování statistik prodejnosti položek a přehledu tržeb. Tato vygenerovaná data lze exportovat do souboru PDF.

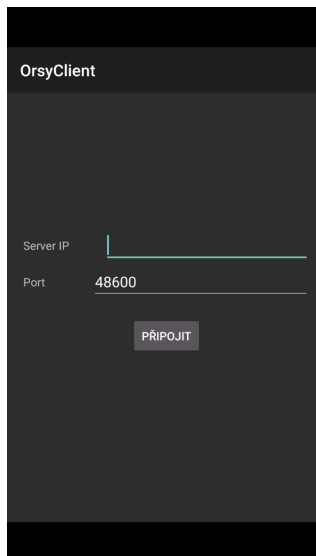
Desktopová aplikace podporuje tisk účtenek na libovolné tiskárně, kterou lze připojit k operačnímu systému. Jako příklad vhodné tiskárny je Mobilní Bluetooth tiskárna 58 mm Dotykačka, která je na následujícím obrázku 4.2.



Obrázek 4.2: Mobilní Bluetooth tiskárna 58 mm Dotykačka

4.2.2 Mobilní číšník

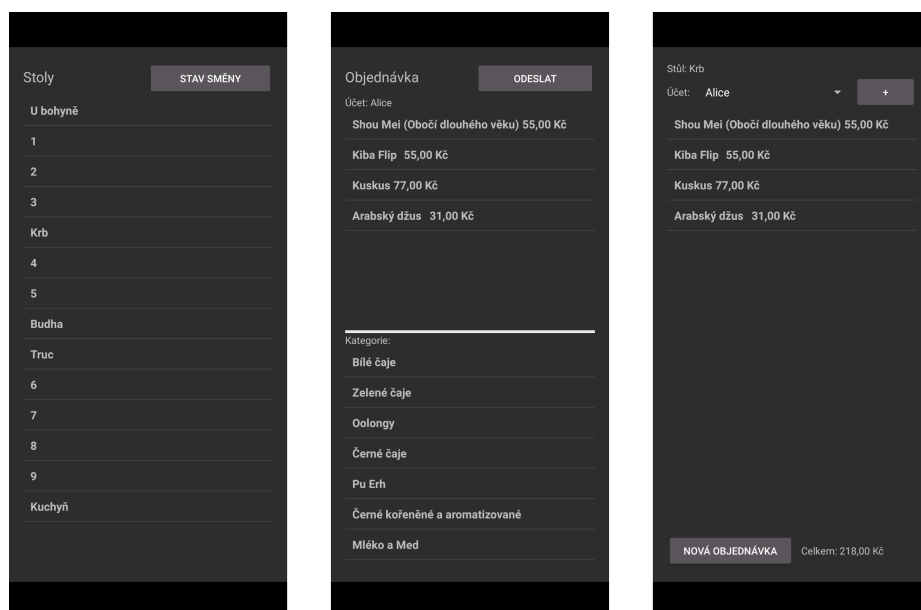
Mobilní číšník je aplikace pro Android. Po spuštění se zobrazí obrazovka (na obrázku 4.3) pro připojení k desktopové aplikaci přes internet.



Obrázek 4.3: První spuštění aplikace mobilního číšníka

Připojení k serveru vyžaduje IP adresu počítače, na kterém běží desktopová aplikace (server), a port serveru, který je možné změnit v administraci desktopové aplikace. Pokud se zařízení s aplikací mobilního číšníka připojuje poprvé, tak po odeslání požadavku o připojení v desktopové aplikaci vyskočí dialog s žádostí o povolení. Po schválení žádosti je zařízení přidáno na whitelist, ze kterého je možné zařízení zpětně odebrat.

Připojí-li se mobilní číšník úspěšně k desktopové aplikaci, zobrazí se seznam stolů (obrazovka vlevo na následujícím obrázku 4.4). Po kliknutí na daný stůl se zobrazí seznam s objednávkami (obrazovka vpravo na obrázku 4.4). Tlačítko „Nová objednávka“ zobrazí obrazovku pro vytvoření a odeslání nové objednávky (obrazovka uprostřed na obrázku 4.4).



Obrázek 4.4: Obrazovky mobilního číšníka

Více informací o uživatelském rozhraní, konfiguraci a ovládání obou aplikací lze nalézt v uživatelské příručce, která je součástí elektronické přílohy A.1.

4.3 Požadavky na zabezpečení dat

Program pracuje s důležitými daty, které se nesmí ztratit ani při nechtěném vypnutí aplikace nebo výpadku počítače. Proto aktuální stav dat je synchronizován s databází v souborovém režimu. Adresář databáze je inicializován ve stejném adresáři kde se nachází aplikace. Aby nemohlo dojít ke ztrátě těchto dat v databázi důsledkem smazání uživatelem nebo poruchou disku, je doporučeno uložit program do adresáře, ke kterému nemá běžný uživatel přístup a pravidelně vytvářet zálohy na jiné úložiště.

Komunikace mezi mobilním číšníkem a desktopovou aplikací používá nešifrovanou komunikaci. Pro zabezpečení dat je tedy vhodné při nasazení v provozovně využít například samostatný WiFi kanál, ke kterému nebude mít veřejnost přístup.

4.4 Analýza výkonnosti systému

4.4.1 Předpokládané nároky na výkon aplikace

Aplikace je určena spíše pro menší podniky.
Příkladem může být podnik s následujícími parametry:

- 20 stolů
- 20 kategorií položek
- 100 položek celkem ve všech kategoriích
- 1 pracovní směna denně
- 50 účtů za směnu
- 10 položek na každém účtu
- 20 zákaznických kreditních karet

Je tedy nutné otestovat a odladit, aby aplikace byla schopná udržet si svoji rychlost a stabilitu i po několika letech pravidelného používání.

V úvahu je potřeba brát i ohled na výkon počítače, na kterém je aplikace instalována.

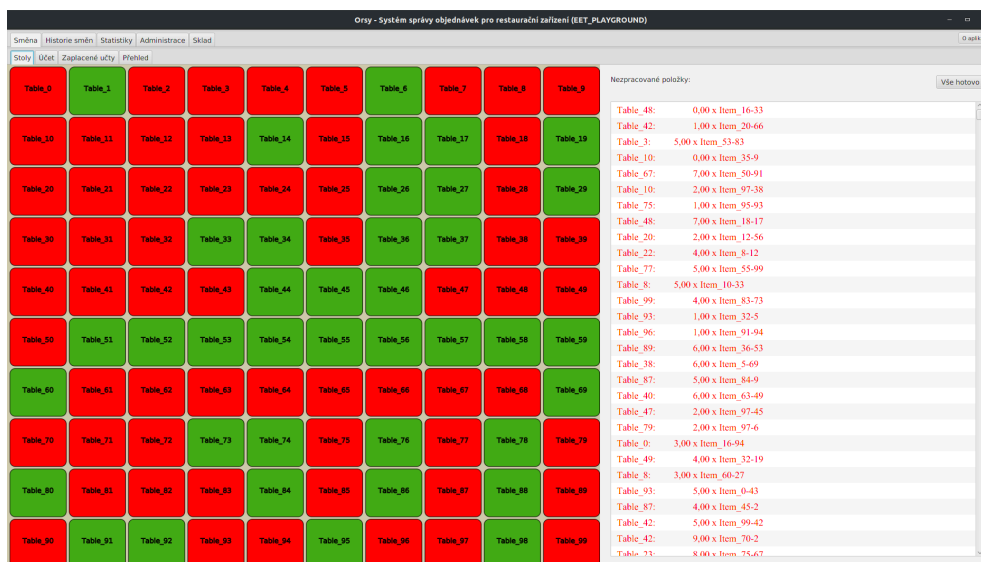
4.4.2 Otestování výkonnosti aplikace

Při testování aplikace bylo vygenerováno větší množství dat, které by mělo dostatečně pokrýt nároky na pravidelné používání systému i po dobu několika let.

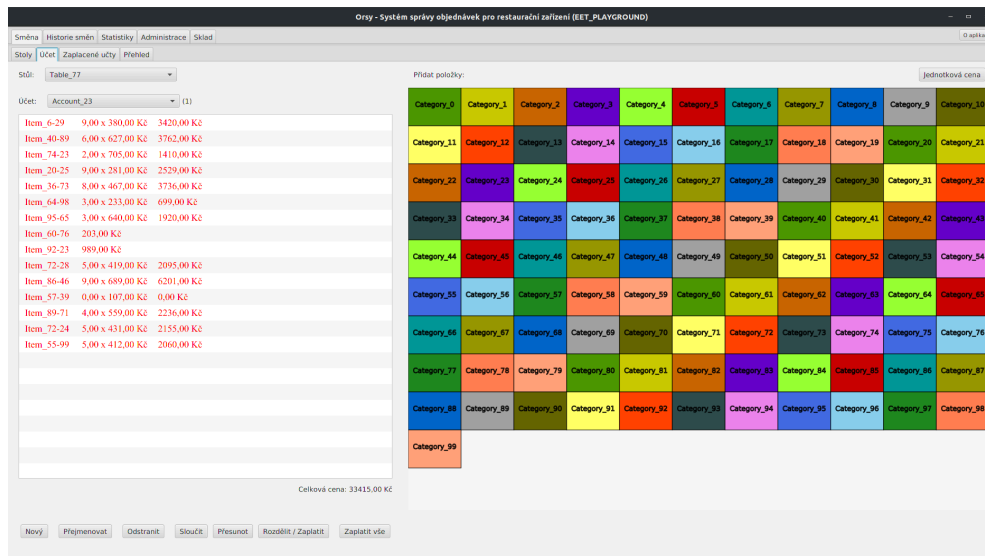
Množství nagenерованých dat bylo následující:

- 1 000 zákaznických kreditních karet
- na každé kartě bylo vytvořeno 1 000 transakcí pro import a export kreditu (celkem 1 000 000 transakcí)
- 100 stolů
- 100 kategorií položek
- 100 položek v každé kategorii (celkem 10 000 položek)
- bylo nasimulováno 3650 směn (odpovídá zhruba 1 směně denně po dobu 10-ti let), poslední směna byla ponechána otevřená
- na každé směně bylo vytvořeno 100 účtů a celkem 1 000 položek (celkem 3 650 000 položek v databázi)

S těmito nagenерованými daty byla následně aplikace spuštěna a otestována. Byl použit běžný osobní počítač s procesorem Intel Core i5-8265U, 16 GB RAM a SSD pevným diskem. Na obrázcích 4.5 a 4.6 jsou zobrazeny obrazovky aplikace s testovacími daty.



Obrázek 4.5: Desktopová aplikace s testovacími daty - Mapa stolů



Obrázek 4.6: Desktopová aplikace s testovacími daty - Účet

Aplikace se na testovacích datech nijak nesekala, a byla plně uživatelsky spolehlivá. Načítání aplikace trvalo cca 40 sekund, většinu času pokrylo načítání dat z databáze. Jediná karta, která musela déle načítat data, byla karta Přehled měsíců. Po kliknutí na kartu trvalo zpracování výsledků zhruba 2 vteřiny. Delší čas na zpracování byl nutný při generování statistik prodejnosti položek, tato úloha trvala zhruba 25 sekund.

Testováno bylo i připojení mobilního číšníka k desktopové aplikaci s těmito daty. Mobilní aplikace byla svižná a se serverem komunikovala bez problémů.

4.5 Reference

Systém Orsy byl již nasazen ve dvou provozovnách, kde je plnohodnotně využíván. V čajovně, která je na následujícím obrázku 4.7, byl nasazen začátkem srpna 2018 na operační systém Ubuntu. Dále v září 2019 byl systém nainstalován v menší horské restauraci na operační systém Windows.



Obrázek 4.7: Nasazení systému Orsy v čajovně

Zákazníci jsou s tímto systémem spokojeni. Prozatím má větší nedostatky pouze mobilní číšník, kvůli jeho horší uživatelské přívětivosti. Do budoucna to je tak oblast, kterou by bylo vhodné ještě vylepšit.

5. Závěr

V rámci této práce byla provedena analýza požadavků na systém správy objednávek pro restaurační zařízení. Požadavky byly posbírány také na základě diskuse s provozovatelem čajovny.

Dále na základě těchto požadavků byl navržen a naprogramován systém Orsy, který je především zaměřen na jednodušší podniky typu čajovna/kavárna, tedy zařízení která nejsou členěna na kuchyň a bar, ale mají pouze jednu přípravnu.

Navíc systém podporuje i Elektronickou evidenci tržeb, která je v současnosti na území ČR běžnou součástí restauračních systémů, a která je nutností aby systém mohl konkurovat komerčním řešením.

Systém byl již nasazen ve dvou provozovnách, kde je nyní se spokojeností využíván. Systém je tedy provozuschopný a navíc je i plnohodnotným konkurentem komerčních systémů.

I přes tuto provozuschopnost a konkurenceschopnost stále existují oblasti, které mohou být vylepšeny. Především se jedná o uživatelské rozhraní mobilního číšníka, které by mohlo být uživatelsky přívětivější. Dále by bylo vhodné zabezpečit komunikaci mezi desktopovou aplikací a mobilním číšníkem, nejlépe rovnou využít například technologie Webových Služeb, které by i navíc zobecnily komunikační protokol, a umožnily tak implementaci mobilních klientů i pro další zařízení mimo ekosystém Javy. Do budoucna by bylo také vhodné využít nástroj Maven, který by především zajistil lepší správu používaných knihoven.

V systému Orsy jsou tedy ještě oblasti, které je možné vylepšit, avšak už i nyní je systém užitečný a přispívá k inovaci v oblasti informačních systémů pro restaurace.

Seznam použité literatury

- [1] Elektronická evidence tržeb: Základní informace pro vývojáře. <https://www.etrzby.cz/cs/zakladni-informace-pro-vyvojare>.
- [2] Guru99: Java vs C# - 10 Key Differences between Java and C#. <https://www.guru99.com/java-vs-c-sharp-key-difference.html>.
- [3] The HSQL Development Group: HyperSQL User Guide. <http://hsqldb.sourceforge.net/doc/2.0/guide/guide.html>.
- [4] Wikipedia: Affero General Public License. https://en.wikipedia.org/wiki/Affero_General_Public_License.
- [5] Wikipedia: GNU Lesser General Public License. https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License.
- [6] StatCounter. <https://gs.statcounter.com>.
- [7] Google Developers, Android: Documentation for app developers. <https://developer.android.com/docs>.
- [8] TechCrunch: Kotlin is now Google's preferred language for Android app development. <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development>.
- [9] Guru99: MVC Tutorial for Beginners: What is, Architecture & Example. <https://www.guru99.com/mvc-tutorial.html>.
- [10] SendGrid: What's the Difference Between Ports 465 and 587? <https://sendgrid.com/blog/whats-the-difference-between-ports-465-and-587>.
- [11] Dotykačka. <https://www.dotyacka.cz>.
- [12] POS Expert. <https://www.posexperts.cz>.
- [13] Gastro. <http://www.gastrosystemy.cz>.
- [14] Datona. <https://www.datona.cz>.
- [15] Agnis Pokladna. <https://www.agnis.cz/hotelovy-software/pokladni-systemy>.
- [16] AWIS. <https://www.registracni-pokladna.cz>.
- [17] FINTA. <http://www.finta.cz/restauracni-system>.
- [18] FUSION. <https://www.horeca-fusion.cz/pokladni-systemy-system>.
- [19] Menu55. <https://menu55.cz>.

Seznam obrázků

2.1	Příklad sestavy s informačním systémem	5
3.1	Graf podílu mobilních operačních systémů na celosvětovém trhu v roce 2020	13
3.2	Diagram architektury projektu	14
3.3	UML Activity Diagram	15
3.4	Databázové relační schéma	16
3.5	Zachycení bezpečnostního klíče z TCP paketu programem Wireshark	17
4.1	První spuštění desktopové aplikace	22
4.2	Mobilní Bluetooth tiskárna 58 mm Dotykačka	22
4.3	První spuštění aplikace mobilního číšníka	23
4.4	Obrazovky mobilního číšníka	23
4.5	Desktopová aplikace s testovacími daty - Mapa stolů	25
4.6	Desktopová aplikace s testovacími daty - Účet	26
4.7	Nasazení systému Orsy v čajovně	27

A. Přílohy

A.1 Obsah elektronické přílohy

V elektronické příloze je k dispozici kompletní software Orsy včetně zdrojových kódů a dokumentace. Programátorská dokumentace je součástí zdrojového kódu desktopové aplikace a lze ji vygenerovat pomocí programu Ant.

Elektronická příloha obsahuje následující soubory:

- Orsy_source_v.5.0.zip - Zdrojový kód desktopové aplikace včetně programátorské dokumentace
- OrsyClient_source_v.5.0.zip - Zdrojový kód mobilního číšníka
- Orsy_v.5.0.zip - Instalační archiv desktopové aplikace pro Linux a Windows
- OrsyClient_v.5.0.apk - Instalační aplikace mobilního číšníka pro Android
- Orsy_UserManual_source_v.5.0.zip - Uživatelská příručka - zdrojový kód pro LaTeX
- Orsy_UserManual_v.5.0.pdf - Uživatelská příručka
- Orsy_Documentation_v.5.0.zip - Programátorská dokumentace

A.2 Diagram komunikačního protokolu

