Mendelova univerzita v Brně
Provozně ekonomická fakulta

Vytvoření objednávacího systému pro restaurace a stravovací zařízení

Diplomová práce

Vedoucí práce:
doc. Ing. František Dařena, Ph.D.

Bc. Daniel Kyzlink

Brno 2015

Zde bude vloženo zadání práce.

Na tomto místě děkuji všem, akademickým pracovníkům, kteří mě provázeli celým studiem, zvláště panu doc. Ing. Františku Dařenovi, Ph.D. za vedení této práce a za odbornou pomoc při její konzultaci. Velké díky patří také celé komunitě, která stojí za vývojem Nette frameworku a která ochotně zodpovídala mé dotazy na Nette fóru.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Vytvoření objednávacího systému pro restaurace a stravovací zařízení**
vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*
Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.
Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 3. ledna 2015 .....

Abstract

Kyzlink, D. Creating ordering system for restaurants and catering facilities. Brno, 2015.

The topic of this thesis is design and implementation of web application for managing orders in restaurants. Application will accelerate and simplify work and it will work as a complete POS system. The thesis describes current accessible POS systems for restaurants, then outlines how such system would be implemented. Then there are mentioned selected implementation techniques, system functions and the application is designed and implemented. Finally, there is described proposed hardware and software for the final run in restaurant.

Abstrakt

Kyzlink, D. Vytvoření objednávacího systému pro restaurace a stravovací zařízení. Brno, 2015.

Tématem diplomové práce je návrh a implementace webové aplikace pro správu objednávek v restauračních zařízeních. Hotová aplikace bude číšníkům zrychlovat a zjednodušovat práci a bude sloužit jako kompletní pokladní systém. Práce zahrnuje současný stav již hotových pokladních systémů pro restaurace, poté nastiňuje jak by se dal takovýto systém implementovat. Dále jsou vybrány implementační techniky, jsou stanoveny veškeré funkce systému a na jejich základě je aplikace navržena a následně naimplementována. Nakonec je navrženo hardwarové a softwarové vybavení pro finální spuštění v restauraci.

Obsah

1 Úvod a cíl práce

1.1 Úvod do problematiky

Informační technologie se v posledních letech dostávají postupně snad do všech odvětví průmyslu i služeb. Pomáhají zlevňovat, zrychlovat a zjednodušovat veškeré procesy. Snad každý větší podnik má svůj informační systém, přes který řeší všechny problémy. Tento trend se dotkl už i nemalého množství restauračních zařízení a restauračního průmyslu celkově. Mnoho restaurací má už pokladny napojené na počítač do kterého se objednávky zapisují a přiřazují k jednotlivým stolům. Tak nějak (ale značně zjednodušeně) to fungunje už na poměrně starých pokladnách. Velmi často lze vidět pokladny s dotykovým displejem. Taková pokladna je skutečně velkým přínosem, protože je uživatelsky velmi přívětivá. Objednávání je skutečně jednoduché a intuituvní. Není problém podívat se na aktuální útratu, zaplatit jen část objednávky a na každou část vystavit zvlášť pokladní doklad. Není to sice nic neobvyklého, ale rozhodně to není samozřejmost. I v těchto řešeních ale zůstává číšníkovy v ruce papír a tužka. Vše musí na baru přepsat do počítače.

Už mnohem méně časté je řešení, kdy má u sebe číšník tablet a objednávky rovnou zadává do systému. Díky tomuto značnému vylepšení pokladního systému už není potřeba neustále opisovat data z papírků do počítače a také se může významně zvýšit rychlost obsluhy. Zatímco jeden číšník obchází stoly a píše si objednávky, druhý už je nosí na stůl. Efektivita je určitě nesporná.

Některé systémy jdou ještě dál a nechávají zákazníka aby si objednával sám, buď ze zabudovaného tabletu ve stole, nebo dokonce z vlastního mobilního telefonu. Číšník je samozřejmě stále v restauraci, ale tento doplněk může v některých případech proces objednávky opět urychlit.

První řešení (bez tabletů) má již velkou konkurenci a mnoho hotových profesionálních a spolehlivých řešení. Druhé je zatím na svém začátku a používá ho jen pár restauračních zařízení. Obě řešení jsou poměrně drahé a pro spoustu restaurací to může být významná a někdy snad i riziková investice. Začíná to specializovaným hardwarem a končí licencemi na softwarové vybavení. Mnoho především menších podníků dále od centra proto zůstává u běžných objednávek na kusu papíru.

Systémy, které dovolí aby si zákazník objednával sám jsou velmi ojedinělé a zatím mají mizivou konkurenci. To se ovšem může rychle změnit. Na druhou stranu je ale otázkou, zda-li je cesta tímto směrem krokem vpřed ke zvyšování efektivity vyřizování objednávek. Na toto si s jistotou nedokáži odpovědět a myslím, že to ukáže až čas.

1.2 Cíl práce

Cílem této diplomové práce je návrh a implementace restauračního systému pro správu objednávek. Celý systém by měl zastávat funkci pokladny, evidenci objednávek, správu jídelního lístku. Dále by měl sloužit jako komunikační kanál mezi

číšníkem, barem a kuchyní. Objednávky, které se připravují v kuchyni do ní budou automaticky zaslány. V případě velké restaurace s více kuchyněmi bude možné u každé položky jídelního lísku nastavit, kde se má objednávka zpracovat, respektive do které kuchyně poslat.

Systém bude zpracovávat objednávky v reálném čase, takže každá objednávka bude vidět okamžitě u baru i v kuchyni. Objednávání bude možné i z mobilních zařízení. Součástí systému bude také skladová evidence. Důležitou funkcí bude zobrazování statistik prodejnosti jednotlivých položek ve zvoleném časovém období a zobrazení obratu restaurace. Tyto přehledy budou zobrazitelné v administrační části systému.

Obsahem práce bude i zmíněné řešení, které umožňuje aby si zákaznící restaurace objednávali pomocí vlastních telefonů. Nebude se ovšem jednat o objednávání prostřednictvím nějaké aplikace, pro kterou by musel mít zákazník telefon s určitým minimálním rozlišením a musel by se v aplikaci nějak orientovat. Bude se jednat o jednoduché požadavky, které bude zákazník posílat do systému pomocí QR kódů natisklých v jídelním lístku, připadně v jiné nabídce restaurace.

Cílem není aby si zákazník mohl bez obsluhy číšníkem objednávat celá menu včetně polévky a dezertu, ale spíš aby si mohl objednat nějaký nápoj, menší nenáročné jídlo nebo balíček slaných tyčinek. Bude se jednat pouze o zpestření prodeje a možnost rychlého objednání věcí, při kterých není nutná komunikace s číšníkem.

Výstupem by měl být hotový, funkční systém připraven ke spuštění v restauraci. Nasazení aplikace je již domluveno ve vinotéce U Leopolda v Brně, kde bude první měsíc systém spuštěn v testovacím provozu a po jeho odladění bude systémem nahrazena veškerá papírová evidence objednávek a účtů.

2 Analýza současného stavu

V každé restauraci ale i kavárně je potřeba objednávky od zákazníků přijímat a posléze je většinou někde za barem vyřizovat. Pokud se jedná o restauraci, bude v ní kromě baru ještě kuchyně s kuchařem. Mezi těmito místy musí existovat nějaký tok informací, který může mít každá restaurace mírně odlišný.

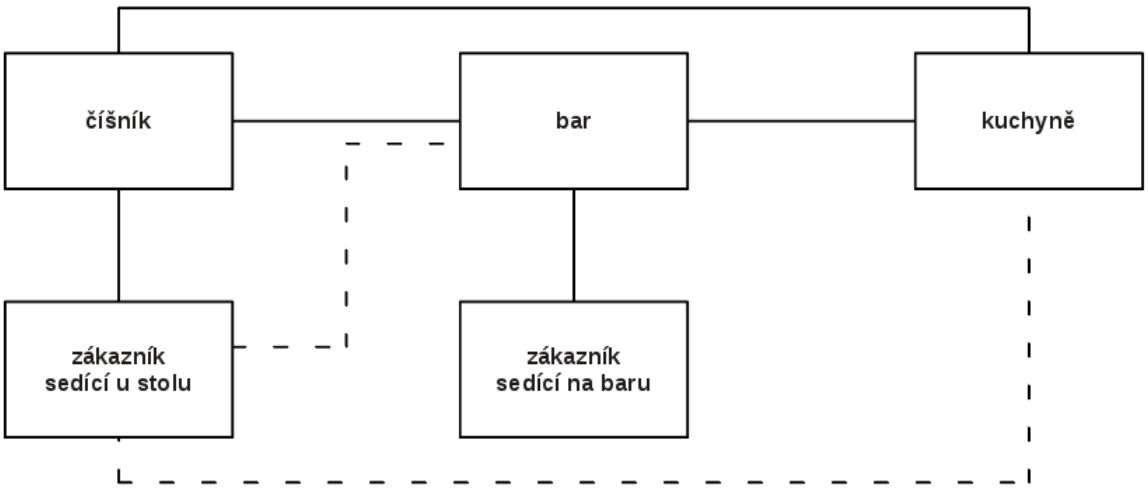
Například v Restauraci U Malchrů v Brně-Židenicích to funguje následovně. Restaurace má 3 místnosti, které obsluhují dva až tři číšníci, dále má bar, za kterým stojí jedna barmanka, potom kuchyni s jedním a někdy dvěma kuchaři a nakonec kancelář, kde je vždy na začátku a konci provozní doby restaurace jeden z provozovatelů.

Informační tok funguje přes číšníka. Ten přijímá objednávky, pokud jsou do kuchyně, osobně je tam předá, pokud k baru, objednávku nepředává, ale sám ji vyřídí. Zákazník si ale může objednat i u baru, ovšem ne jídlo z kuchyně. Objednávky u baru vyřizuje barmanka a rovnou vyžaduje platbu.

V některých restauracích se objednávky do kuchyně vyřizují telefonicky. Děje se tak většinou tam, kde je kuchyně v jiném patře a jídlo se vozí dopravním výtahem.

Pomocí pokladních systémů, se může s kuchyní komunikovat o něco rychleji pomocí počítače. V případě mobilního číšníka se může informace do kuchyně dostat dokonce ihned po objednání. To už zajisté nějaký čas uspoří.

Pokud do systému přidáme objednávání samotnými zákazníky, docílíme toho, že z určitého procenta objednávek můžeme číšníka úplně vyřadit (neuvažuji samotnou donášku připravené objednávky), protože zákazník bude komunikovat přímo s barem a kuchyní.



Obr. 1: Tok informací v restauraci. Čárkovaná čára ukazuje tok informací za předpokladu, že si zákazník objednává sám pomocí mobilního zařízení.

2.1 Klasické pokladny

Tyto pokladny obvykle zachovávají koncept takový jako známe například z potravin, tedy s tlačítky a malým displejem. Do restaurace jsou zpravidla upraveny tím způsobem, že mají omyvatelnou plochu s tlačítky (ty jsou pouze mírně vystouplé a překryté folií). Další, pro restaurace nezbytnou úpravou je vedení účtu každého hosta respektive skupinu hostů. Pokladna má zpravidla vestavěnou termotiskárnu, podporuje několikery měny a někdy umí i zálohovat na paměťovou kartu. Je to tedy řešení „vše v jednom“. Tato pokladna stojí kolem desítí tiscíc korun a stačí ji postavit na pult a strčit do zásuvky. (Pokladna SHARP XE-A217B)

2.2 Dotykové pokladní systémy

Taková poklada zpravidla obsahuje všechny funkce jako klasická pokladna zmíněná výše. Velkou výhodou je větší dotykový displej, díky kterému je systém přehlednější. Všechny úkony jsou intuitivní a zaučení je otázkou chvilky. Tyto systémy jsou většinou instalovány do počítačů „All in one“ (vše v jedné skříni). Není tedy nutná žádná kabeláž (vyjma napájení). K takovému počítači nebýva připojena ani klávesnice a vše se ovládá přes dotykovou obrazovku. Systém může být vyvíjen například v prostředí Delphi v programovacím jazyce Object Pascal (ABX software). I přesto, že jsou tyto pokladní systémy implementovány pod OS Windows a šli by tedy sprovoznit na většině osobních počítačů, jsou dodávány vyhradně se speciálním hardwarem. Tento hardware má sice svoje výhody, ale taky znatelně zvyšuje cenu. Tyto pokladní systémy začínají na ceně okolo třiceti tisíc korun. (Pokladna ABX)

2.3 Pokladní softwarové systémy

Jak již název napovídá, zde už se nejedná o pokladnu jak si ji bežně představíme, ale „pouze“ o softwarové vybavení. Provozovatel restaurace koupí pouze software, který si může rozběhnout na svém hardawaru. Máli zájem, může si k systému zakoupit i doporučený hardware výrobce, ten je totiž již připraven na provoz v restauraci. (Pokladní systém Septim)

2.4 Mobilní číšník

Jak bylo zmíněno v úvodu, mobilní číšník nám přináší efektivitu především v rychlosti vyřizování objednávek. Ne každá z profesionálních pokladen tuto funkci nabízí. Zatím to rozhodně není standartem a ve většině restauracích jsem toto řešení neviděl. Nicméně je několik pokladních systému, které mobilního číšníka mají. Jsou v podstatě dva přísupy. První nabízí mobilního číšníka jako hardwarové přenosné zařízení s potřebným softwarem, druhý nabízí software, který lze nainstalovat na poměrně bohatou škálu tabletů.

Kompletní systémy včetně hardwaru vyrábí například firma Orderman. Takové řešení stojí kolem 30 tisíc korun, zařízení je robustnější a tedy i odolnější než obyčejný

tablet. Komunikuje na ISM pásmu (433 Mhz). Dosah zařízení je 300m, ale není problém ho prodloužit za použití přídávných stanic (v případě řešení Orderman se jedná o zařízení Base Station3). Velkou výhodou je také možnost dokoupení kapesní tiskárny, která lze na mobilního číšníka jednoduše napojit. (Orderman-Accessories)

Obr. 2: Mobilní číšník orderman Don. (Mobilní terminál Orderman DON)

Druhé řešení, tedy software, který lze nainstalovat na tablet je poněkud levnější a jednodušší. Tablet není tak odolný, má o něco menší výdrž baterie a většinou je náchylný na zvýšenou vlhkost. Výhodou tabletu může být jeho cena, ale také velikost displeje. Jako příklad bych uvedl mobilního číšníka od společnosti AWIS pokladní systémy, který stojí 4500 Kč a běží pod OS Android. (Mobilní číšník) (Kasy, pokladny a pokladní systémy s tradicí)

Je tedy z čeho vybírat, a v případě vytíženější restaurace se investice vrátí velice rychle.

2.5 Přehled nalezených řešení

Ke všem nalezeným řešením jsem získal poměrně málo informací. Nepodařilo se mi totiž najít žádné open source řešení ani žádné podrobně zdokumentované. I tak se ale pokusím vystihnout alespoň rámcově funkce nalezených řešení.

Septim restaurace

Software pro restaurace Septim je vlajkovým produktem společnosti ASW Systems, který vyvinuli pro užití v náročném restauračním provozu. Celý restaurační a pokladní systém je optimalizován pro dotykové pokladny a dělí se na dvě hlavní části, pokladna a manažer.

Modul pokladna se nejvíce blíží klasické představě o pokladně. Má řadu funkcí jako například tisk a kopie účtu, storno chybně namarkovaných položek, dělení účtu, přesun či spojení účtů, vzkazy do kuchyně a mnoho dalších. Jako pokladnu je možné používat i již zmíněné mobilní číšníky Orderman.

Část manažer obsahuje informace o skladu, možnost jeho doplnění, zobrazuje tržby a obraty, ziskovost jednotlivých produktů, evidence zákazníků, cenové kategorie a další. (Pokladní systém Septim)

Pokladní systém Septim vyniká kvalitní technickou podporou, zaštiťuje ho stabilní společnost s mnohaletou historií. Jedná se o produkt na profesionální úrovni.

AWIS GASTRO

Komplexní pokladní systém vhodný pro jakýkoliv druh gastronomického provozu. AWIS GASTRO obsahuje program pro obsluhu „Kasa“, kterým se provádí účtování na stoly. Dále obsahuje například automatické odečítání surovin ze skladu (či více skladů), databázi zákazníků (jejich útratu, počet návštěv, aj.) či rozvážkový systém. Druhá část pokladního systému pro restaurace obsahuje manažerskou část „Office“ pro statistiky, tržby, fakturace, inventury, normování, tvorbu kalkulací a směsí, nastavení měny, platebních karet a stravenek, denní přehledy a grafy, docházku zaměstnanců a další propracované funkce. V ceně pokladního systému je automaticky také síťové řešení a propojení přes internet. Propojení přes internet je ceněn zejména pro provoz a kontrolu z domova či jiného pracovního místa. (Pokladní systémy pro restaurace, bary, kluby a diskotéky)

K programu AWIS GASTRO se dá dokoupit pokladní modul MOBILNÍ ČÍŠNÍK pro tablety. Obsluhující zadává do aplikace objednávky, které posílá prostřednictvím sítě Wi-fi rovnou na bar nebo do kuchyně, rovněž může účtovat a kasírovat zákazníky kdekoliv v terénu. Modul lze také připojit na přenosný pokladní terminál pro bezhotovostní platbu kartou nebo na přenosnou tiskárnu účtenek. (Pokladní modul MOBILNÍ ČÍŠNÍK pro tablety a terminály)

AWIS GASTRO společně s modulem MOBILNÍ ČÍŠNÍK je taktéž poměrně profesionální řešení od společnosti s dlouhou historií. K veškerému softwaru dodávají i specializovaný hardware.

Fusion

Základem je elektronická pokladna – standardní počítač nebo specializované pokladní řešení (např. od IBM). K této pokladně se číšník přihlásí svým heslem nebo bezdrátovým čipem a markuje objednané položky. Fusion řešení disponuje

přehledným zobrazením statistik prodeje. Krom nezbytných funkcí umožňuje platit účty kombinací několika plateb (hotovost, stravenka, platební karta), tisk náhledu účtenky, spojování a rozdělování účtů, pojmenování a přesouvání účtů mezi stoly, jazykové mutace účtenek, dopisování poznámek k objednávkám a mnoho dalšího. Součástí řešení je i mobilní číšník. (Číšnická pokladna)

Celá tato aplikace beží na webovém serveru a je dostupná přes prohlížeč. Na jednotlivé stanice se nic neinstaluje. Službu poskytuje společnost i-Technologies, která je zaměřena především na webové aplikace.

Agnis

Řešení Agnis obsahuje taktéž spoutu zajímavých funkcí. Například objednávky jídla posílá rovnou do kuchyně, můžeme v něm provádět rezervaci stolů, pracuje se skladem, můžeme zjišťovat ziskovost jednotlivých položek. Hlavní obrazovka pokladny nabízí přehledný pohled na reálné rozmístění stolů v provozu. Tím zrychluje obsluhu, snižuje chybovost a zpříjemňuje práci obsluze pokladny. Červenou barvou označuje aktivní stoly s ještě neuzavřenými účty. (Systém AGNIS - Pokladna)

Program plně využívá možností, které nabízí a poskytuje operační systém Windows – grafické rozhraní, ovládání nejmodernějších periferií, vazbu na internet, databázový SQL jazyk apod. Informační systém Agnis se skládá z modulů, které mohou být používány v celku nebo samostatně, na jednom PC nebo v rozsáhlé počítačové síti. Za systémem Agnis stojí téměř 20 let vývoje a více než 700 spokojených zákazníků. (Systém AGNIS - Představení systému)

HarSys

Pokladní systém HarSys se skládá ze dvou částí – pokladní a manažerské. Do manažerské části nemají číšníci přístup. Zobrazují se v ní statistiky, sortiment, sklady, uzávěrky aj. Apliakce má několik možností instalace.

- Jednouživatelská
  Celý pokladní systém běží na jednom počítači.
- Síťová
  Jeden nebo více počítačů slouží jako pokladna, další počítač nebo i více počítačů jsou v kanceláři. Jeden z počítačů v síti slouží jako server.
- Síťová s propojením přes internet
  Některé z počítačů se k serveru připojují prostřednictvím internetu. Předpokladem je dostatečně rychlá linka.
- Víceuživatelská
  S dávkovým přenosem dat pomocí souborů. Manažerské úkony se dějí např. na notebooku (tvorba ceníků) – podle potřeby se předávají ve formě souborů. Počítače na sobě nejsou závislé a nemusejí být trvale propojeny.

K systému lze připojit i mobilního číšníka, který obsahuje tytéž možnosti jako pokladna. (Pokladní systém Harsys)

Menu55

Menu55 je poměrně nový systém, který se od předchozích vzájemě podobných nejvíce liší. Přichází s myšlenkou kdy si zákazníci mohou objednávat sami a to i z vlastního zařízení přes internet. Objednávku mohou zaslat z restaurace po uvedení čísla a hesla stolu případně mimo restauraci po uvedení telefonního čísla. Systém dále obsahuje pokladnu a mobilního číšníka. Celá služba běží u provozovatele a vše se dělá přes webový prohlížeč. Můžeme si tedy k systému napojit libovolné množství vlastních zařízení a platíme stále stejný, poměrně malý měsíční poplatek. (Pokladní systém pro restaurace a bary)

Systém se nijak neinstaluje, stačí se zaregistrovat a zaplatit. Velkou výhodu vidím v jednoduchosti a ceně. Oproti předchozím systémům ho není problém po půl roce přestat používat a nepřijít přitom o velké peníze. Pro profesionální provozy by ale mohla být překážkou mladá a malá společnost bez delší historie na trhu.

2.6 Zhodnocení nalezených řešení

Většina nalezených řešení je poměrně drahá a jejich pořízení může být pro většinu menších a začínajících podniků riskantní. Značná část taky neumožňuje instalaci na vlastní zařízení a je potřeba pořídit si dražší zařízení přímo od dodavatele softwaru. Z toho vybočuje akorát Menu55, které je cenově dostupné a je možno provozovat jej na vlastním zařízení. Je ale prakticky nemožné aplikaci nějak upravit na přání zákazníka, protože beží na serveru provozovatele. Zaroveň při výpadku internetu je aplikace nedostupná, což by za plného provozu restaurace mohlo mít velmi nepříjemné následky.

Nenašel jsem jedinou variantu open source řešení, která by se alespoň vzdáleně touto problematikou zabývala. Žádné z nalezených řešení také neumožnuje alternativní objednávání přímo zákazníky bez použití aplikace, například pomocí zmíněných QR kódů.

3 Možnosti implementace

Možností jak implementovat danou problematiku je nepřeberně. V této kapitole bych rád popsal alespoň některé z nich.

3.1 Aplikace bežící přímo na PC

Asi nejtradičnější variantou je aplikace která se spustí přímo na PC pod některým z operačních systémů. Taková aplikace by se nainstalovala na jednotlivé stanice a serverová část na server. Výhodou by bylo efektivní využití HW na stanicích, bezpečnost a také možnost pracovat offline. Tím by se předešlo problémům například při výpadku sítě. Možnost pracovat offline je tedy nesporná výhoda. Jako programovací jazyk vhodný k použití bych uvedl C++, Java, Delphi, C. Z těchto bych vypíchl jazyk Java, který je interpretovaný a překládá se až při spuštění. Jeho start je pomalejší, ale jeho velkou výhodou je multiplatformnost. Není tedy problém program v něm napsaný spustit na různých operačních systémech.

3.2 Aplikace bežící přes http(s) protokol

Další méně častou variantou je aplikace bežící přes internetový prohlížeč. Takovou aplikaci můžeme spustit na kterémkoliv zařízení umožňujícím zobrazovat internetové stránky. Nároky na HW jednotlivých počítačů (výjma serveru) jsou minimální. Nevýhodu je závislost na síťovém spojení. Tím vzniká riziko nejen dostupnosti systému ale také riziko bezpečnostní. Obě se ale dají významě snížit během aplikace pouze na vnitřní sítí (LAN). Většinou ani není důvod aby aplikce bežela po internetu (nicméně výše zmíněné Menu55 tak funguje). Odstínění aplikace od OS jednotlivých stanic a její běh pouze na serveru sebou přináší i výhody a to především v ceně HW, kdy potřebujeme jen jeden výkonější počítač. Všechny ostatní stanice můžou být už jakékoliv i starší s libovolným OS. Případně stačí zakoupit „tenké klienty“ (např. OptiPlex FX170).

Co se týče výběru programovacího jazyka, opět je z čeho vybírat. Mezi nejrozšířenějšími je ASP.NET, Perl, PHP, Ruby, Python. V klientské části (v prohlížeči) bych jako vhodný jazyk zmínil pouze JavaScript s knihovnou JQuery.

3.3 Mobilní aplikace

I mobilní číšník by mohl bezproblému fungovat přes internetový prohlížeč. Byl by jen potřeba vhodně navržený responzivní design, který by i na malém zařízení zajistil přehlednost a intuitivní používání. Nicméně by na mobilním telefonu bylo praktičtější mít aplikaci, která by na web přistupovala. Přecejen by nás tam jinak mohl rušit neustále vyskakující adresový řádek, který nás v aplikaci vůbec nezajímá. Bylo by vhodné podporovat systémy Android, iOS a Windows Phone (a to právě v tomto pořadí). (VŠETEČKA, Roman)

3.4 Databáze

Základním kamenem systému bude dobře navržená a rozšiřitelná databáze. Nároky na ni nejsou velké, systém nebude obsahovat stovky tabulek ani do něho nebude tisíce přístupů denně. Přece jen, bude to systém pro jednu restauraci. Můžeme si tedy vybrat v podstatě libovolnou relační databázi.

4 Metodika práce

Tato kapitola se bude zabývat zvolenými technologiemi. Pokusím se je zde nějak popsat a nastínit, proč jsem si vybral právě tyto. Dále se bude kapitola zabývat jakými fázemi bude aplikace v průběhu vývoje procházet.

4.1 PHP

PHP je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací. Při použití PHP pro dynamické stránky jsou skripty prováděny na straně serveru. PHP jazyk není dílem žádné firmy, ale jednotlivců, a je udržován jako technologie s otevřenými zdrojovými kódy. Protože vyniká jednoduchostí a funguje na různých operačních systémech i různých webových serverech, je velmi oblíben a značně rozšířen. Takto PHP prezentuje (PONKRÁC, Miloslav.) v první kapitole. Pan Ponkrác v knize ovšem neuvažuje objektově orientované programování, které je v PHP5 už na dobré úrovni.

Objektově orientované PHP bylo často kritizováno. Je to způsobeno i tím, že PHP4 ve kterém se objevila objektová orientace poprvé v celé historii PHP, postrádalo některé základní funkce, zkrátka mu scházeli nástroje skutečného objektově orientovaného jazyka. Příchod PHP5 toto tvrzení ale změnil. PHP5 je už skutečně plnohodnotný objektově orientovaný programovací jazyk. Smyslem objektové orientovanosti v PHP není snaha udělat z PHP Javu nebo něco podobného, ale nabídnout pořádný nástroj všem webovým vývojářům. Objektová orientovanost je další strategií pro přijetí současného stavu vývoje webů. (LAVIN, Peter.)

Současné verze PHP jsou 5.4, 5.5 a 5.6. Verze 5.4 vyšla v březnu 2012. Byly z ní odstraněny některé zastaralé funkce a direktivy, zvýšil se výkon a snížily požadavky na paměť. Verze 5.5 vyšla v červnu 2013 a jako novinku přinesla například operátor yeld pro vytváření generátorů a blok finally pro ošetřování vyjímek. Verze 5.6 je ještě horkou novinkou vydánou na konci srpna 2014, která na spoustě hostinzích není ještě podporována. (PHP)

PHP jazyk jsem zvolil vzhledem k jeho osvěčenosti a rozšířenosti pro webové aplikace. Na PHP běží například wikipedia.org, nebo také CMS wordpress. PHP jazykem se už pár let zabývám a rád bych v něm zkusil OOP a taky bych se rád naučil pracovat s nějakým frameworkem.

4.2 Nette framework

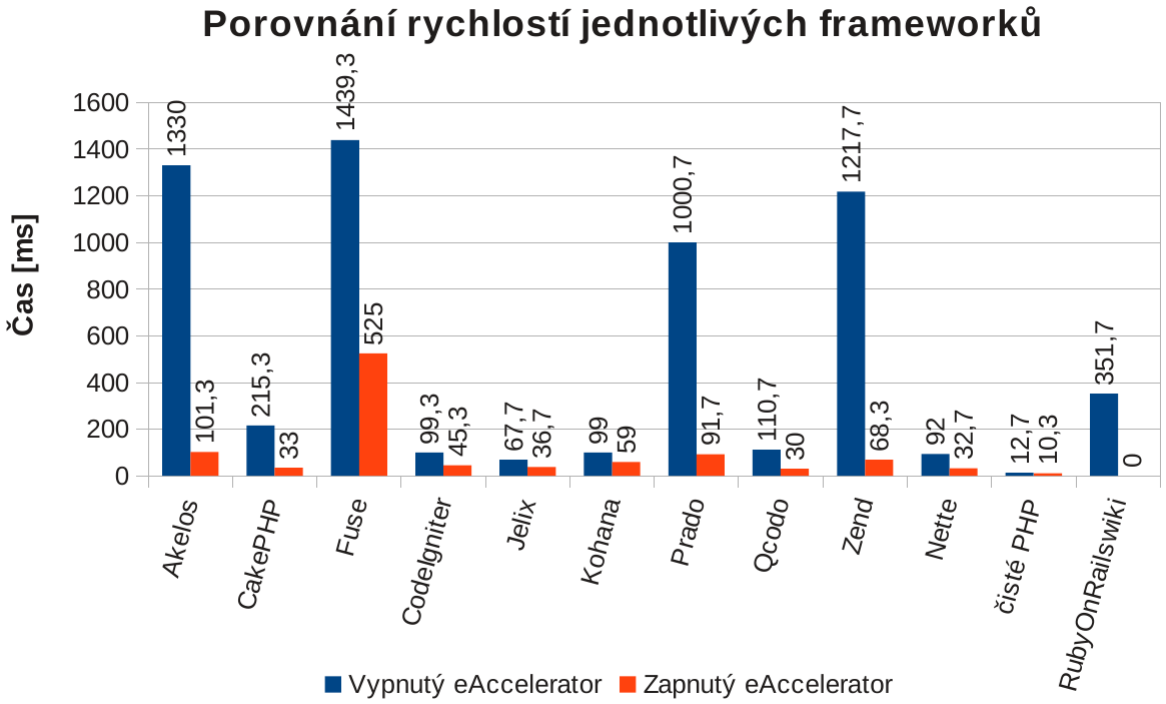
Framework je sada nástrojů, knihoven, konvencí a osvědčených postupů, které se snaží abstraktní a rutinní úkoly vložit do obecných modulů a ty mohou být znovupoužívány. Cílem je, aby se programátor soustředil na úkoly, které jsou specifické pro daný projekt. Takto definuje frameworky obecně Jeff Croft na svém blogu.

Programátor pracující v čistém PHP musí například bezpodmínečně ošetřovat veškeré vstupy do aplikace a i přesto, že PHP má spoustu funkcí, musí stále

programovat triviální úlohy používané velmi často. Většina takových programátorů používá svoje naprogramované věci v různých projektech a je to vlastně takový jejich vlastní framework.

Pokud se chce programátor odstínit od rutinních úloh má možnost sáhnout po nějakém již existujícím propracovaném frameworku. A výběr je velký. Já jsem si pro tvorbu aplikace zvolil Nette framework. Vůbec prvotním důvodem je poměrné velká česká komunita a taky jeho rozšířenost v Česku. Nette framework je totiž dílem českého vývojáře Davida Grudla. Výhodou je v porovnání s ostatními frameworky, ale i s čistým PHP také jeho rychlost jak je vidět na obrázku 3.

Za zmínku ale stojí i Zend Framework. Ten je zase jeden z nejpopulárnějších frameworků celosvětově. Je to robustní framework, řešící opravdu mnoho problémů, na úkor toho je ale pomalejší. Dokumentaci má stejně jako Nette velmi dobrou. K Nette bych ještě zmínil pravidlené „Posoboty“. Jsou to pravidlené srazy konající se ve větších městech Česka (nejčastěji v Praze a Brně) na kterých je možné osobně konzultovat se zkušenějšími, poslechnout si zajímavou přednášku a také se pobavit. Jedné Posoboty v Brně jsem se v rámci seznámování s Nette zůčástnil.



Obr. 3: Rychlosti frameworků. (DANĚK, Petr.)

Nette usnadňuje práci hned na několika místech. Eliminuje výskyt bezpečnostních děr a jejich zneužití, jako je například XSS, CSRF, session hacking, session fixation a další. Disponuje vlastním ladícím nástrojem (laděnka), zobrazuje chyby

přehledněji než PHP a také upozorní i na věci, které se od PHP nedozvím vůbec. Nette je moderní framework, který podporuje práci s AJAX, má jednoduché nastavování tvaru URL (vhodné pro SEO) a používá MVC softwarovou architekturu. Dále má k dispozici mnoho doplňků, které můžeme v aplikacích používat. No a nakonec můžeme díky němu vhodně rozdělovat práci mezi programátory a HTML kodéry. (GRUDL, David.)

Model-View-Controller (MVC)

Model-View-Controller je softwarová architektura, která vznikla z potřeby oddělit u aplikací s grafickým rozhraním kód obsluhy (controller) od kódu aplikační logiky (model) a od kódu zobrazujícího data (view). Tím aplikaci zpřehledňuje, usnadňuje budoucí vývoj a umožňuje testování jednotlivých částí zvlášť.

Model je datový a zejména funkční základ celé aplikace. Je v něm obsažena aplikační logika. Jakákoliv akce uživatele (přihlášení, vložení zboží do košíku, změna hodnoty v databázi) představuje akci modelu. Model si spravuje svůj vnitřní stav a ven nabízí pevně dané rozhraní. Voláním funkcí tohoto rozhraní můžeme zjišťovat či měnit jeho stav. Model o existenci view nebo kontroleru neví.

View, tedy pohled, je vrstva aplikace, která má na starost zobrazení výsledku požadavku. Obvykle používá šablonovací systém a ví, jak se má zobrazit ta která komponenta nebo výsledek získaný z modelu.

Controller je řadič, který zpracovává požadavky uživatele a na jejich základě pak volá patřičnou aplikační logiku (tj. model) a poté požádá view pro vykreslení dat. Obdobou kontrolerů v Nette Frameworku jsou presentery. (GRUDL, David.)

4.3 JavaScript

JavaScript (zkracovaný jako JS) je multiplatformní, interpretovaný, dynamický, objektově orientovaný jazyk s first-class funkcemi, známý především jako skriptovací jazyk pro webové stránky. (Mozilla developer network) Spouští se ve webovém prohlížeči klinta, to pro nás znamená, že si nemůžeme být jisti jeho spuštěním a proto se na něho nemůžeme spoléhat. Webová aplikace musí fungovat i bez podpory JavaScriptu. JavaScript nám ale pomáhá udělat aplikaci příjemnější, intuitivnější a použitelnější. V posledních letech se k němu začali psát různé knihovny a velmi se rozšířil. Vedle CSS a HTML se bere jako třetí nazbytná část moderního webu. Existují k němu ale alternativy jako například VBScript (ten už nyní upadá, prohlížeče ho nepodporují) a CoffeeScript, který je spíše novinkou a začíná se používat. Já ale zůstal u JavaScriptu se kterým už mám nějakou zkušenost.

AJAX

AJAX zanamená Asynchronní JavaScript a XML. Namísto XML se ale častěji používá formát JSON, proto existuje i zkratka AJAJ, ta se ale příliš často nepoužívá. Nejedná se ani o programovací jazyk ani o knihovnu. AJAX pouze určuje způsob jak

využívat současné standarty. Popisuje jak si vyměňovat data se serverem a aktualizovat části webové stránky bez nutnosti jejího znovunačtení. Je to moderní technika jak tvořit rychlé a dynamické webové stránky. AJAX dnes využívá většina webů, jako je facebook, google služby, youtube. Ale existují i služby jako třeba novinky.cz od společnosti Seznam, které AJAX nevyužívají. Například při pouhém hodnocení komentáře se musí celá stránka načíst znovu, což je často nepříjemné a nepřehledné. AJAXový požadavek lze samozřejmě napsat v čistém JavaScriptu, většinou ale používáme nějakou knihovnu, která nám tyto požadavky maximálně zjednoduší.

JQuery

Knihovna jQuery je napsaná v již zmíněném programovacím jazyce JavaScript. Tato knihovná výrazně ulehčuje tvorbu JavaSriptových aplikací včetně AJAXových požadavků. Knihovna je velmi populární také pro své jednoduché a účinné možnosti tvorby efektů a animací. (MARGORÍN, Marián.) Pan Margorín Marián Dále také uvádí, že jQuery nám díky svým animacím často nahradí težkopádný flash.

Tuto knihovnu dnes používají i světové servery jako například www.amazon.com. To co bysme v JavaScriptu psali hodiny na desítky řádků, to v jQuery napíšeme v mžiku na pár řádků. Efektivita práce je naprosto fascinující.

Jedním z nejmocnějších aspektů knihovny jQuery je lehký výběr elementů v modelu DOM a to pomocí selektorů. Díky tomu můžeme jednoduše přistupovat k jednotlivým oblastem webové stránky a tyto části dynamicky modifikovat, plnit obsahem či animovat.

Protože považuji jQuery knihovnu za velmi významný krok v efektivnosti práce i času při tvorbě dynamických stránek a také ve zpřístupnění JavaScriptu šikovným kodérům uvedu porovnání dvou kódů vytvářejících jednoduchou animaci. Jeden v čistém JavaScriptu, druhý za použití knihovny jQuery.

Takto by vypadala animace v knihovně jQuery.

```
$("#priklad").animate({
  left: '500px',
  height: '150px'
}, 500);
```

A takto vypadá v čistém javaSriptu.

```
var test = null;
function pohyb() {
  test.style.left = parseInt(test.style.left)+500+'px';
  test.style.height = parseInt(test.style.height)+150+'px';
  setTimeout(pohyb,20);
}

function init() {
  test = document.getElementById('priklad');
```

```
  test.style.top = '0px';
  test.style.left = '0px';
  pohyb();
}

windiw.onload = init;
```

Tento kód navíc ani nebere v úvahu různé prohlížeče, takže odladěný výsledek by mohl být ještě delší. Při složitějších animacích by byla výhoda jQuery o mnoho markantnější a to především v přehlednosti kódu. (MARGORÍN, Marián.)

4.4 CSS3

Kaskádové styly (CSS) představují jednoduchý mechanismus přidávání stylů (např. fontů, barev, mezer) do webových dokumentů. Veškerý vzhled webu by se měl definovat právě pomocí CSS. (Cascading Style Sheets)

CCS3 přichází s poměrné zajímavými novinkami, které webdesignerskou činnost posouvají o třídu výše. Prvky, které jsme na webu museli dříve zobrazovat pomocí obrázků můžeme nyní nastylovat v CSS. A to díky možnosti nadefinovat zakulacené rohy objektů, stínování, vložení více obrázků na pozadí, tvorba barevných přechodů, průhlednost elementů a další.

CSS3 taky významě napomáhá při tvorbě responzivního designu a to především díky *Media Queries*. Media Queries nám umožňuje definovat styly pro jednotlivá zařízení a různá rozlišení. Díky tomu můžeme velmi jednoduše skrývat, či jinak stylovat elementy pro nižší rozlišení na mobilních telefonech.

Tato nová verze CSS taky překonala všechny problémy spjaté se zobrazováním webového písma (to znamená používání fontu dostupném na webovém serveru) jehož počátky sahají už do roku 1998. Jelikož na to ale nebyl vydán žádný standat a každý si to dělal po svém nebylo v podstatě možné to rozumně používat. S příchodem CSS3 to již problém není. Vše jde jednoduše pomocí directivy *@font-face*.

Kaskádové styly se neustále vyvíjejí a je spousta novinek, které ještě nejsou zahrnuty ve specifikaci W3C. Prohlížeče tyto novinky zpřístupňují jako testovací vlastosti a každý prohlížeč pro ně má svůj prefix. Díky tomu je můžeme již nyní bez problémů používat. (CASTRO, Elizabeth a Bruce HYSLOP.)

```
div{
  -moz-border-radius: 10px;
  -webkit-border-radius: 10px;
  border-radius: 10px;
}
```

Ukázka definice stylu *border-radius* nejdříve pro Firefox dále pro prohlížeče s jádrem WebKit (Safari, Chrome) a nakonec definice, která bude platná až se vlastnost dostane do specifikace W3C.

Novinek je samozřejmě celá řada a můžete se s nimi seznámit například na webu http://www.w3schools.com/css/css3_intro.asp.

4.5 HTML5

HTML5 je nyní nejaktuálnější verze značkovácího jazyka pro tvorbu webových stránek. Značky jazyka HTML slouží k popisu sémantiky webu. Ve starších verzích se v HTML definoval i vzhled, postupně se ale veškeré definice vzhledu přesunuly do kaskádových stylů a HTML5 už klade velký důraz na výhradně sémantické použití a o vzhled se nestará vůbec. Z toho důvodu přibyly nové elementy jako například *article* či *section*. Element *article* slouží k označení jednoho článku a *section* k označení nějakého logického celku webu. Některé elementy HTML5 také úplně vypustilo. Například element *big* znamenal mírné zvětšení písma, toto ale dnes do HTML vůbec nepatří a řeší se přes CSS. Velkou novinkou je také podpora přehrávání zvukových a video souborů přímo v moderních webových prohlížečích a to bez nutnosti instalace doplňků. K tomu jsou zavedeny nové elementy *video* a *audio*. Jazyk HTML5 také přináší lepší podporu vektorové grafiky v otevřeném formátu SVG. (CASTRO, Elizabeth a Bruce HYSLOP.)

Velmi důležitou změnou je tedy především striktní oddělení semántiky od stylystiky. A proto vznikly nové elementy popisující různé typy obsahu.

4.6 XML

Extensible Markup Language (XML) je velmi jednoduchý, flexibilní textový formát, původně navržen pro publikaci většího množství dokumentů. Nyní hraje XML velmi významnou roli při výměně dat mezi jednotlivými často webovými aplikacemi. (Extensible Markup Language)

XML tagy nejsou nikde definovány a určujeme si je sami podle potřeby. Jazyk je navržen jako sebepopisující. Výsledný dokument jsou pouze holá data, která nic nedělají. Pro jejich zobrazení či zpracování je potřeba použití dalšího programu. (Introduction to XML)

XML komunikace s ekonomickým systémem POHODA

Program POHODA využívá komunikační rozhraní XML, díky kterému můžeme přijímat i odesílat data z a do různých aplikací. V současné době je tato komunikace upřednostňována před importy a exporty tabulek i přímým přístupem do databáze. Důvodem je vyšší bezpečnost operací, neboť při XML komunikaci dochází ke kontrole a validaci dat tak, jako by šlo o operace vykonávané přímo v prostředí programu. Například při importu dokladů ve formátu XML tak dochází ke kontrole vstupních údajů, dopočítání chybějících informací a v případě chybovosti importovaných dat také k upozornění na nutnost doplnění nebo přepracování XML souboru bez jeho

importování. Bezpečnost této formy komunikace je srovnatelná s bezpečností a možnostmi ručního zápisu dat v prostředí programu POHODA. (XML komunikace se systémem Pohoda)

Vytvářený restaurační systém bude mít jako jednu z funkcí export dat ve formátu XML připravených pro import do účetního softwaru Pohoda. Učetní software Pohoda jsem zvolil na základě diplomové práce na téma Srovnání nejpoužívanějších účetních softwarů v ČR z hlediska možnosti zpracování účetních dat od autorky (BĚHOUNKOVÁ, Martina.). V případě požadavku klienta na přípravu XML pro jiný účetní systém není problém program rozšířit.

4.7 Spirálový model

Celý vývoj systému bude řízen metodikou spirálového modelu. Tento model pokrývá nedostatky vodopádového modelu, je založen na iterativním přístupu díky kterému se aplikace postupně vylepšuje dokud není hotová. Jeho hlavní fáze jsou:

- určení cílů, alternativ, omezení (determine objectives, alternatives, constraints),
- vyhodnocení alternativ, identifikace a řešení rizik (evaluate alternatives, identify, resolve risks),
- vývoj a verifikace další úrovně produktu (develop, verify next-level product) a
- plánování následujících fází (plan next phases).

Po každé fázi následuje testování, hodnocení a konzultace se zadavatelem projektu. (BOEHM, Barry W.)

První fáze modelu byla uvedena v určení cílů této závěrečné práce. Je potřeba přesně vymezit co má aplikace umět. To je dále vymezeno v Use case diagramech.

Druhá fáze je probrána v kapitole implementační nástroje, kde jsou zváženy možnosti jak systém naprogramovat. Bezpečností ryzika jsou z velké části pokryty Nette frameworkem.

V třetí fázi budu řešit logické rozvržení aplikace. Návrh databáze a vazeb v ní, návrh tříd systému a rozčlenění do modulů. Podle tohoto návrhu bude aplikace napsána.

Nakonec naplánuji další funkcionalitu a po otestování naprogramované části aplikace se bude pokračovat další iterací.

Při řešení veškerých problémů budou použity co nejmodernejší a nejbezpečnější techniky řešení. To znamená, že se budu snažit využít především Nette doplňků a přistupovat k databázi budu výhradně pomocí Nette/database.

5 Realizace aplikace

Tato kapitola se zabývá samotnou realizací restauračního systému včetně jeho analýzy a stanovení funkcí. Po neformální specifikaci jsou uvedeny případy užití (use case) pro jednotlivé aktéry systému. Pod každým případem užití je jeho specifikace. Tímto jsou jasně definovány všechny funkce, které bude systém umět. Následují diagramy tříd a entitně relační diagram databáze, které už ukazují vnitřní strukturu systému. V druhé části kapitoly je popsána již hotová aplikace. Popis je rozdělen podle modulů, kde jsou u každého modulu popsány jeho funkce.

Aplikace je navržena, tak aby se dala co nejjednoduššeji rozšířit a zdokonalit. Tomu významně pomáhá Nette Framework, který k takovému návrhu směřuje sám od sebe. Po testovacím spuštění ve vinotéce U Leopolda bude proveden další vývojový cyklus a celý návrh aplikace se posune zase o něco blíže k reálným potřebám provozovatele restaurace.

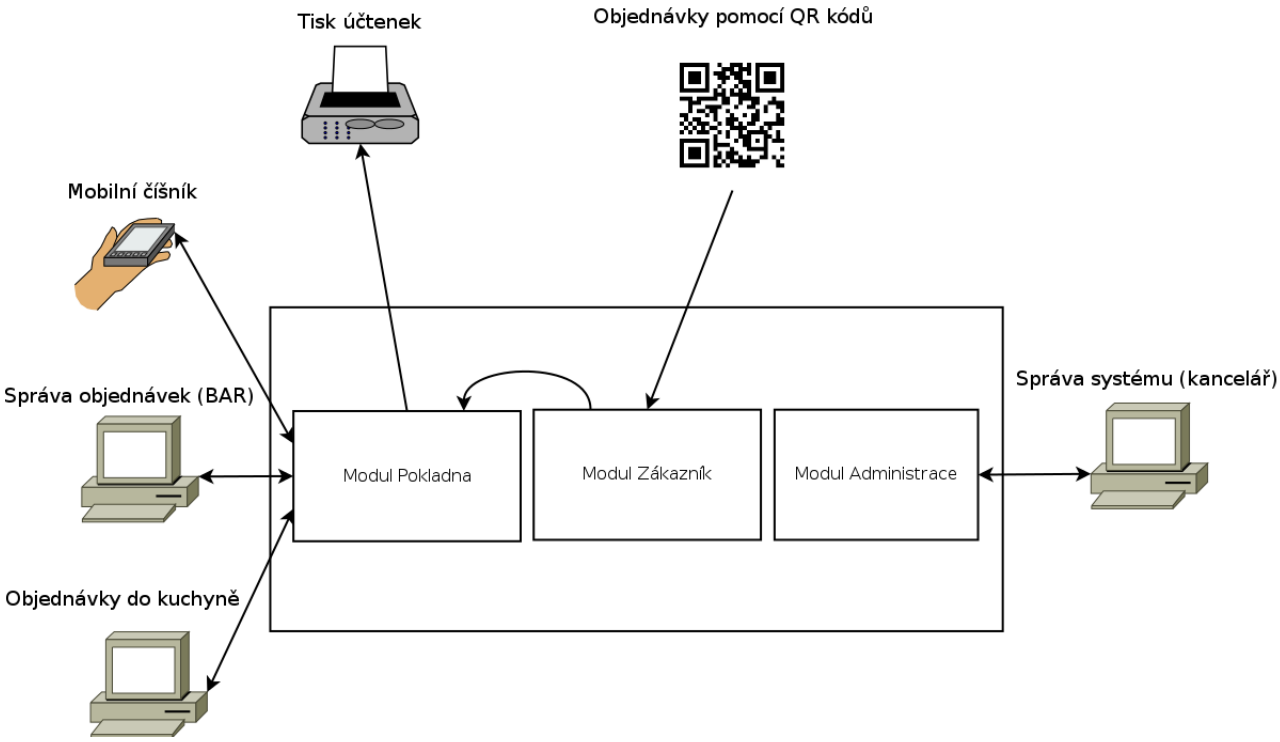
5.1 Neformální specifikace systému

Do celého systému můžou data vstupovat hned z několika míst. Objednávky do systému vstupují pomocí počítače ke správě objednávek co bude umístěn u baru, skrz tablet, se kterým číšník obchází zákazníky (mobilní číšník) a nakonec mohou zákazníci zadávat některé požadavky sami pomocí QR kódů umístěných v jídelních lístcích. Dále je systém možno konfigurovat z počítače u baru, nebo dalšího počítače v kanceláři provozovatele.

Každý objekt, který do systému vstupuje, komunikuje s určitým modulem. Nejjednodušší je Modul zákazník, který pouze přijímá požadavky zaslané pomocí QR kódů, tyto požadavky zkontroluje a uloží do databáze. Dále s nimi už pracuje modul Pokladna. Na modul Pokladna je připojen mobilní číšník, správa objednávek (bar) a kuchyně. Poslední modul (Administrace) se stará o správu a nastavení celé aplikace, o výpisy statistik a generování XML výstupů.

Celý systém bude ovlivňován z několika míst, které popisuje následujíci seznam a obrázek.

- Kancelář – Počítač, ze kterého se bude systém nastavovat.
- Bar – Pro zadávání a vyřizování objednávek.
- Kuchyně – V kuchyni se budou objednávky připravovat případně odmítat.
- Mobilní číšník – Pro zadávání objednávek přímo od stolu zákazníka.
- Zákazník – Odesílání QR objednávek.



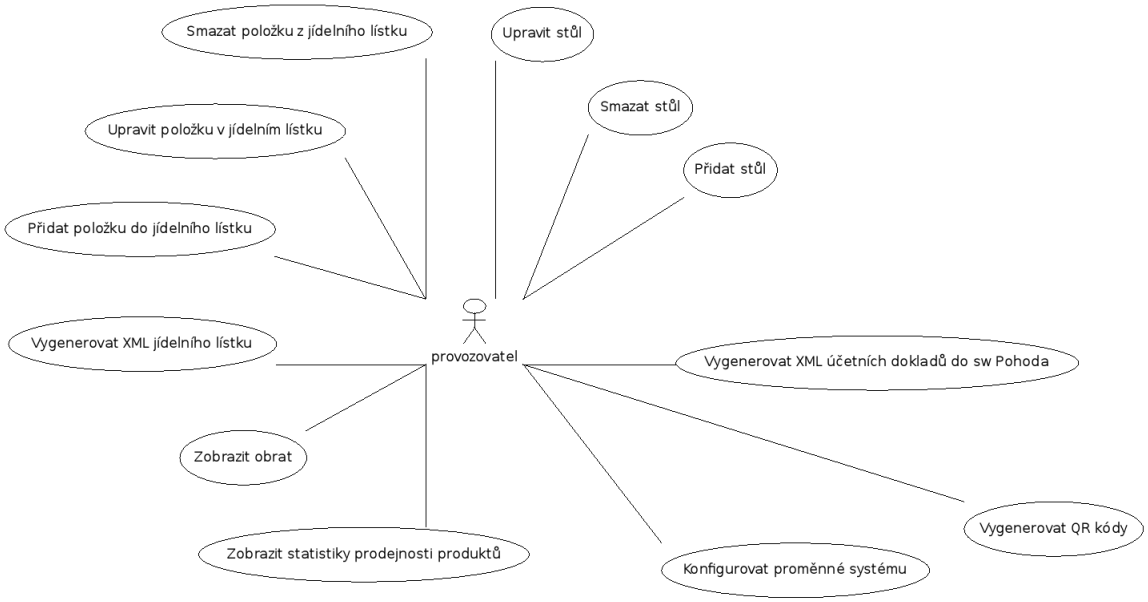
Obr. 4: Systém a jeho okolí.

5.2 Use case diagramy

Use case diagram zobrazuje chování systému tak, jak ho vidí uživatel. Účelem diagramu je popsat funkcionalitu systému, tedy co od něj klient nebo my očekáváme. Diagram vypovídá o tom, co má systém umět, ale neříká jak to bude dělat. Proto je to většinou první diagram, který při návrhu informačního systému vytváříme. Je důležité se nejprve shodnout na tom, co má náš systém umět. Až potom má smysl se ptát, jak to vlastně uděláme. (ČÁPKA, David)

V následujících use case diagramech můžeme vidět co který aktér v systému může provádět za operace. Za každým use casem následuje jeho slovní specifikace, která obsahuje jeho obecný popis, co do něj vstupuje a jak se operace provede. Do celého systému vstupují celkem čtyři aktéři a to následující:

- provozovatel,
- číšník,
- kuchař a
- zákazník.



Obr. 5: Use case s aktérem provozovatel.

Přidat položku do jídelního lístku

- Popis – Vloží novou položku do jídelního lístku.
- Vstup – Název, podnázev, popis, cena, množství, typ stravy, měna, jednotka, daňová třída.
- Provedení – Po odeslání se zkontroluje, zda-li jsou vyplněny správně všechny povinné položky. Poté se nová položka a vazba na typ stravy uloží do databáze.

Upravit položku v jídelním lístku

- Popis – Upraví položku jídelního lístku.
- Vstup – Název, podnázev, popis, cena, množství, typ stravy, měna, jednotka, daňová třída.
- Provedení – Po odeslání se zkontroluje, zda-li jsou vyplněny správně všechny povinné položky. Poté se provede update databáze.

Smazat položku z jídelního lístku

- Popis – Smaže položku jídelního lístku.
- Vstup – Id položky jídelního lístku.
- Provedení – Položka se v databázi označí jako smazaná a dále se v systému nezobrazuje.

Přidat stůl

- Popis – Přidá nový stůl.
- Vstup – Název, počet míst, souřadnice X, souřadnice Y, rotace.
- Provedení – Po odeslání se zkontroluje, zda-li jsou vyplněny správně všechny povinné položky. Poté se nový stůl uloží do databáze.

Upravit stůl

- Popis – Upraví parametry stolu.
- Vstup – Název, počet míst, souřadnice X, souřadnice Y, rotace.
- Provedení – Po odeslání se zkontroluje, zda-li jsou vyplněny správně všechny povinné položky. Poté se provede update databáze.

Smazat stůl

- Popis – Smaže stůl.
- Vstup – Id stolu.
- Provedení – Stůl se v databázi označí jako smazaný a dále se v systému nezobrazuje.

Vygenerovat XML jídelního lístku

- Popis – Vygeneruje XML všech položek jídelního lístku a zobrazí jej v okně prohlížeče.
- Vstup – Žádný.
- Provedení – Vybere z databáze všechny nesmazané položky jídelního lístku a předá šabloně, ta je zobrazí ve formátu XML.

Vygenerovat XML účetních dokladů do sw Pohoda

- Popis – Vygeneruje XML účetních dokladů pro jejich import do ekonomického softwaru Pohoda.
- Vstup – Datum od, datum do.
- Provedení – Vybere z databáze všechny účetní doklady, které spadají do zvoleného datového rozpětí. Data předá šabloně a ta vygeneruje XML specifikované ekonomickým softwarem Pohoda.

Vygenerovat QR kódy

- Popis – Vygeneruje QR kódy ve formátu png pro všechny položky, které je možné pomocí QR kódu objednat. Sada QR kódů položek je vygenerována pro jednotlivé stoly.
- Vstup – Žádný.
- Provedení – Vybere z databáze všechny položky, u kterých je nastaveno QR objednání. Pro každou položku vygeneruje png obrázek pomocí rozšíření QrCode a ty následně zobrazí.

Zobrazit statistiky prodejnosti produktů

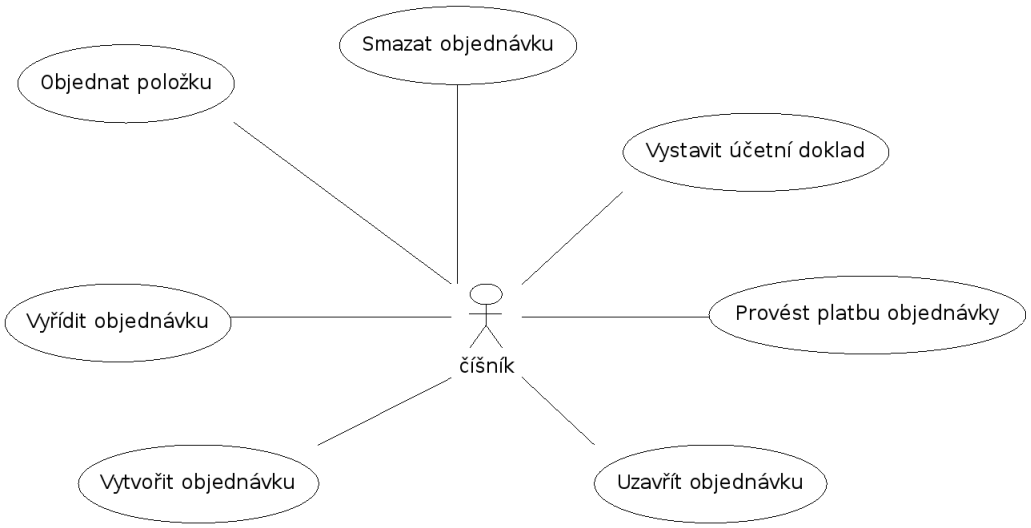
- Popis – Zobrazí sloupcový graf po hodinách, dnech nebo měsících a ukáže v něm jak moc se daný produkt nebo skupina produktů v jednotlivých časových úsecích prodává.
- Vstup – Datum od, datum do, pole produktů, typ přehledu.
- Provedení – Ze všech uzavřených objednávek se vyberou ty co uživatel požaduje a co spadají do zvoleného časového rozpětí. Tato data se roztřídí po hodinách, dnech v týdnu nebo měsících a zobrazí ve sloupcovém grafu vygenerovaném pomocí doplňku Google Charts.

Zobrazit obrat

- Popis – Zobrazí celkový obrat restaurace ze všech objednávek za dané časové období.
- Vstup – Datum od, datum do.
- Provedení – Vybere všechny objednávky které spadají do zvoleného časového období a jsou jíž vyřízeny a zaplaceny. Z těchto vybraných objednávek vypočítá celkový obrat a zobrazí ho.

Konfigurovat proměnné systému

- Popis – Umožní konfiguraci proměnných systému jako je například ip adresa ze které bude systém přijímat QR objednávky.
- Vstup – Jednotlivé proměnné systému.
- Provedení – Všechny vstupy se zkontrolují a validují zda-li obsahují povolené hodnoty, poté se uloží do databáze.



Obr. 6: Use case s aktérem číšník.

Vytvořit objednávku

- Popis – Vytvoří novou objednávku na konkrétním stole.
- Vstup – Id stolu.
- Provedení – Vytvoří v databázi novou objednávku na stole s daným id. Zapíše čas vytvoření objednávky.

Objednat položku

- Popis – Objedná položku k dané objednávce.
- Vstup – Pole položek.
- Provedení – Vytvoří záznam ve spojovací tabulce mezi objednávkou a jídelním lístkem.

Vyřídit objednávku

- Popis – Objednanou položku nastaví jako vyřízenou.
- Vstup – Id položky.
- Provedení – V databázi nastaví položku jako vyřízenou. Položka se přestane zobrazovat v aplikaci nevyřízených objednávek.

Smazat objednávku

- Popis – Objednanou položku smaže.

- Vstup – Id položky.
- Provedení – Položku s daným id nastaví jako smazanou.

Provést platbu objednávky

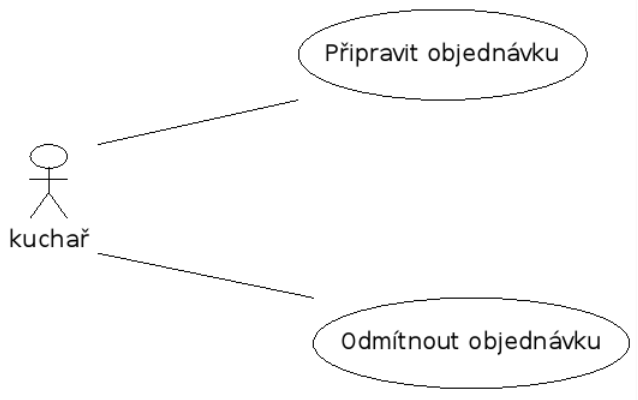
- Popis – Nastaví vybrané položky v objednávce jako zaplacené.
- Vstup – Pole položek k zaplacení, id objednávky.
- Provedení – Sečte ceny vybraných položek, zkontroluje jestli jsou vyřízené a označí je jako zaplacené. Nakonec do databáze uloží účetní doklad.

Vystavit účetní doklad

- Popis – Vystaví účetní doklad. Umožní ho poslat emailem nebo připravit k tisku.
- Vstup – Id účetního dokladu.
- Provedení – Z databáze vybere všechny položky které náleží k danému účetnímu dokladu a z nich vygeneruje účtenku.

Uzavřít objednávku

- Popis – Uzavře objednávku a dále ji už nezobrazuje na daném stole.
- Vstup – Id objednávky.
- Provedení – Zkontroluje, zda-li jsou na stole všechny vyřízené položky zaplacené. Pokud ano upozorní na nevyřízené položky a po odsouhlasení označí objednávku jako uzavřenou.



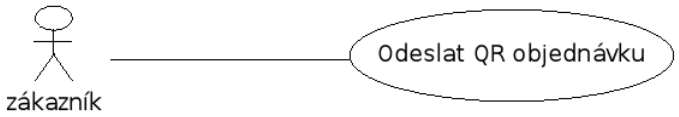
Obr. 7: Use case s aktérem kuchař.

Připravit objednávku

- Popis – Objednanou položku označí jako připravenou k servírování.
- Vstup – Id objednané položky.
- Provedení – Objednanou položku nastaví v databázi jako připravenou.

Odmítnout objednávku

- Popis – Objednanou položku odmítne. Položka se nebude vyřizovat.
- Vstup – Id objednané položky.
- Provedení – V databázi nastaví objednanou položku jako odmítnutou.



Obr. 8: Use case s aktérem zákazník.

Odeslat QR objednávku

- Popis – Vytvoří v systému novou QR objednávku.
- Vstup – GET požadavek s id stolu a id objednané položky.
- Provedení – Zkontroluje, zda-li je na daném stole aktivní objednávka a zda-li lze zvolená položka objednat pomocí QR kódu. Pokud je vše v pořádku vloží do systému novou objednávku a nastaví ji přiznak QR objednávka.

5.3 Class diagramy

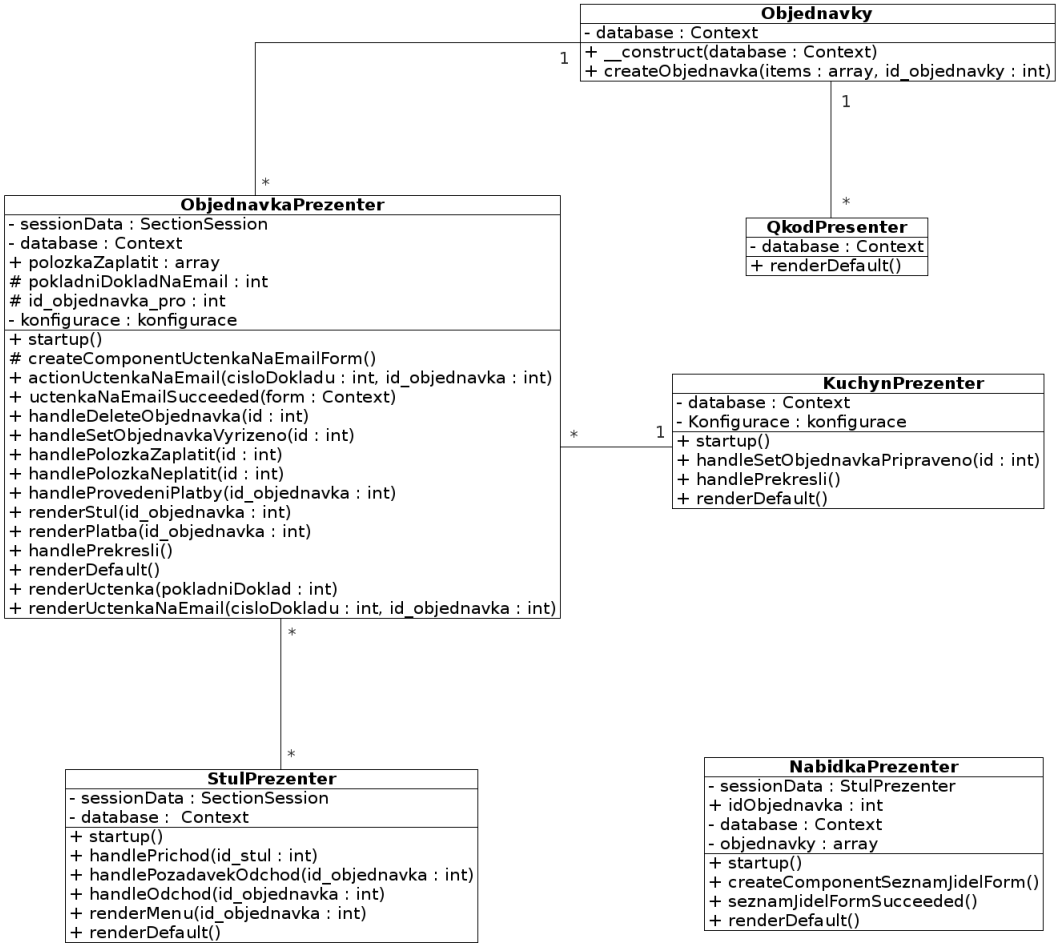
Následující diagramy tříd jsou o něco zjednodušené. Existuje v nich především spousta dědičnosti, která vychází z použití Nette Frameworku. Dále nejsou v diagramech znázorněny vazby na třídy komunikující s databází, která je taktéž součástí Nette frameworku.

Class diagram už nepopisuje systém z pohledu uživatele jak to bylu u use case diagramu. Class diagram se zaměřuje na systém z pohledu programátora. Většinou ho vytváří zkušenější programátoři, protože se od něj do jisté míry odvíjí kvalita celé aplikace. Programovat systém podle diagramu tříd už můžou provádět i programátoři s méně zkušenostmi.

Class diagram modulu Pokladna a Zákazník

Tento class diagram obsahuje 6 tříd. Z toho je 5 presenterů a jeden model. Modelová třída *Objednavky* se stará o tvorbu objednávek v databázi a využívají ji některé prezentery. Modul zákazník obsahuje pouze prezenter *QkodPresenter*, a proto jsem class diagramy obou modelů sjednotil do jednoho. Tento prezenter využívá třídu *Objednavky*, která je součástí modelu společného pro všechny moduly.

Každý prezenter obsahuje nejméně jednu metodu s prefixem *render*, která zajišťuje vykreslení šablony.



Obr. 9: Class diagram modulu Pokladna a Zákazník.

Class diagram modulu Administrace

Class diagram modulu administrace obsahuje 7 tříd, které se starají o veškerou administraci systému. Jejich metody jsou především k tvorbě a zpracovávání formuláře pro vytváření nových položek v databázi připadně jejich update. Dále slouží ke generování QR kódů a statistik.

Největší třída je *jidelniListekPresenter*, která se stará o veškeré nastavení jídelního lístku, měn, jednotek, typů stravy a dalších parametrů položek v jídelním lístku.

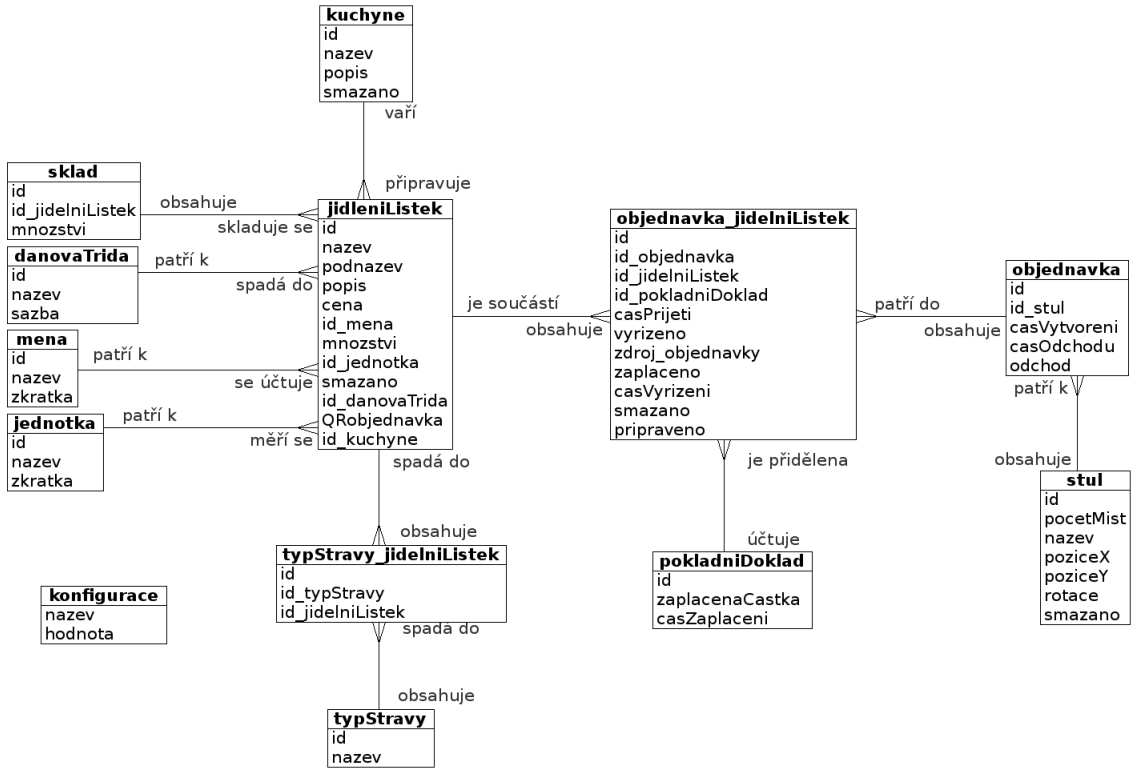
Opět každý prezenter obsahuje nejméně jednu metodu s prefixem *render*, která zajišťuje vykreslení šablony.

Obr. 10: Class diagram modulu Administrace.

5.4 Struktura databáze

Jako systém řízení báze dat, jsem zvolil MySQL. Celá databáze obsahuje 13 tabulek, z toho jedna je pouze pro data obsahující parametry nastavení systému. V databázi je zajištěno entitní integritní omezení a návrh databáze splňuje 3. normální formu. Ve většině tabulkách se záznamy nemažou, ale pouze se nezobrazují pokud je ve sloupci *smazáno* hodnota „1“.

V kapitole jsou po entitně relačním diagramu popsány jednotlivé tabulky (stručně je popisuje také tabulka 1) a jejich sloupce.



Obr. 11: Entitně relační diagram znázorňující všechny tabulky v databázi.

Tabulka jidelniListek

Tabulka *jidelniListek* obsahuje informace o jednotlivých jídlech a nápojích, případně i služeb. V této tabulce je tedy veškerá nabídka restaurace.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- nazev – Obsahuje název jídla či nápoje (např. guláš).
- podnazev – Obsahuje podnázev jídla či nápoje. Může to být jakýkoliv dodatek názvu.

Tab. 1: Přehled databázových tabulek a jejich stručný popis.

Název tabulky	**Stručný popis tabulky**
jidelniListek	Tabulka jidelniListek obsahuje položky jídelního lístku.
danovaTrida	Každá položka z tabulky jidelniListek musí spadat do nějaké daňové třídy.
mena	Každá položka z tabulky jidelniListek má měnu.
jednotka	Každá položka z tabulky jidelniListek má jednotku.
sklad	Každá položka z tabulky jidelniListek má v této tabulce uložen počet kusů co je na skladě.
typStravy	Tabulka obsahuje seznam typů jídel a nápojů.
typStravy_jidelniListek	Tabulka typStravy_jidelniListek zajišťuje vazbu mezi tabulkami jidelnilistek a typStravy.
objednavka	Tabulka objednavka obsahuje jednotlivé objednávky, nikoliv však konkrétní položky.
objednavka_jidelniListek	Tabulka objednavka_jidelniListek propojuje jednotlivé objednávky s položkami jidelního lístku.
stul	Tabulka stul obsahuje parametry jednotlivých stolů.
pokladniDoklad	Do tabulky pokladniDoklad se ukládají údaje o zaplacených objednávkách.
kuchyne	Tabulka kuchyne obsahuje názvy a popisy jednotlivých kuchyní v restauraci.
konfigurace	Tato tabulka slouží pro nastavení celého systému. Jedná se o uživatelské nastavení aplikace.

- popis – Zde může být nějaký delší popis položky jídelního lístku.
- cena – Cena za jeden kus.
- id_mena – Id použité měny.
- mnozstvi – Kolik množství obsahuje jedna porce.
- id_jednotka – Id použité jednotky.
- smazano – Může nabývat hodnot nula nebo jedna. V připadě smazání položky z jídelního lístku tam položka fyzicky zůstane, ale její hodnota ve sloupci smazáno se změní na jedničku.
- id_danovaTrida – Id použité daně.
- QRobjednavka – V případě, že nabývá hodnoty jedna. Je povoleno tuto položku objednat pomocí QR kódu.
- id_kuchyne – Id kuchyně, ve které se jídlo připravuje.

Tabulka danovaTrida

Tabulka *danovaTrida* obsahuje všechny možné daňové sazby, které jsou pak přiřazeny ke konkrétním položkám jídelního lístku.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- nazev – Obsahuje název daňové třídy.
- sazba – Sazba v procentech.

Tabulka mena

Tabulka *mena* obsahuje všechny typy měn, které restaurace akceptuje.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- nazev – Obsahuje název měny.
- zkratka – Obsahuje zkratku měny.

Tabulka jednotka

Tabulka *jednotka* obsahuje různé množstevní jednotky jako jsou například gramy, litry, kusy, aj.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- nazev – Obsahuje název jednotky.
- zkratka – Obsahuje zkratku jednotky.

Tabulka sklad

Tabulka *sklad* obsahuje informace o skladových zásobách jednotlivých položek z jídelního lístku.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- id_jidelniListek – Id jídla, které je ve skladu.
- mnozstvi – Obsahuje informaci o množství dané položky na skladu.

Tabulka typStravy

Tabulka *typStravy* obsahuje kategorie položek z jídelního lístku. Například to může být zákusek, studený nápoj, předkrm, aj.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- nazev – Obsahuje název typuStravy.

Tabulka typStravy_jidelniListek

Tabulka *typStravy_jidelniListek zajišťu*je vazbu mezi tabulkami *jidelniListek* a *typStravy*. Pokud se v budoucnu rozhodneme, že chceme k nějaké položce přiřadit dva typy stravy nebude to díky této spojovací tabulce problém.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- nazev – Obsahuje název typu stravy.

Tabulka objednavka

Tabulka *objednavka* obsahuje jednotlivé objednávky. Objednávka je vytvořena při první objednávce daného klienta v restauraci. Uzavřena je při zaplacení celé objednávky a odchodu od stolu.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- id_stul – Id stolu, na kterém je objednávka.
- casVytvoreni – Při vytvoření objednávky se zaznamená čas.
- casOdchodu – Při odchodu se zaznamená čas.
- odchod – Může nabývat hodnot nula nebo jedna. Po příchodu je ve sloupci hodnota nula. Po zaplacení a uzavření objednávky se do sloupce zapíše hodnota jedna.

Tabulka objednavka_jidelniListek

Tabulka *objednavka_jidelniListek* propojuje jednotlivé objednávky s položkami jidelního lístku. Přiděluje tím každé objednavce objednané položky. Je to stěžejní tabulka aplikace, protože obsahuje veškeré objednané položky.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- id_objednavka – Id objednávky na stole.

- id_jidelniListek – Id přiřazeného jídla.
- id_pokladniDoklad – Id vyhotoveného pokladního dokladu.
- casPrijeti – Při objednání položky se zaznamená čas.
- vyrizeno – Může nabývat hodnot nula nebo jedna. Nulu mají nevyřízené položky, jedničku vyřízené.
- pripraveno – Může nabývat hodnot nula, jedna nebo dva. Nulu mají nepřipravené položky, jedničku připravené a dvojku ty, co se v kuchyni nijak nepřipravují.
- zaplaceno – Může nabývat hodnot nula nebo jedna. Nulu mají nezaplacené položky, jedničku zaplacené.
- casVyrizeni – Při vyřízení položky se zaznamená čas.
- smazno – Může nabývat hodnot nula nebo jedna. V případě smazání objednávka v tabulce zůstane, ale její hodnota ve sloupci smazáno se změní na jedničku a bude skrytá.
- zdroj_objednavky – Obsahuje jedničku v případě, že se jedná o QR objednávku.

Tabulka stul

Tabulka *stul* obsahuje parametry jednotlivých stolů a je propojena s tabulkou objednávky. Každá objednávka má přiřazen jeden stůl. Na stole může být ale více vytvořených objednávek zároveň.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- pocetMist – Obsahuje informaci o počtu míst u stolu.
- poziceX – Souřadnice stolu v plánku restaurace na ose X.
- poziceY – Souřadnice stolu v plánku restaurace na ose Y.
- rotace – Rotace stolu v plánku ve stupních.
- smazno – Může nabývat hodnot nula nebo jedna. V případě smazání stůl v tabulce zůstane, ale jeho hodnota ve sloupci smazáno se změní na jedničku a bude skryt.

Tabulka pokladniDoklad

Do tabulky *pokladniDoklad* se ukládají údaje o zaplacených objednávkách. Objednávky co jsou placeny v rámci jedné platby jsou na společném dokladu. Díky tomu, mohu objednávku rozdělit na více dokladů, připadně platit na jeden doklad více objednávek.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- zaplacenaCastka – Celková hodnota zaplacené objednávky.
- casZaplaceni – Při zaplacení se zaznamená čas.

Tabulka kuchyne

V tabulce *kuchyne* jsou uloženy informace o kuchyních v restauraci. Ukládá se pouze název a popis.

Tabulka obsahuje následující sloupce.

- id – Má nastavený auto increment a je primárním klíčem tabulky.
- nazev – Název kuchyně.
- popis – Popis kuchyně.
- smazno – Může nabývat hodnot nula nebo jedna. V případě smazání kuchyň v tabulce zůstane, ale její hodnota ve sloupci smazáno se změní na jedničku a bude skryta.

Tabulka konfigurace

V tabulce *konfigurace* jsou nastaveny všechny uživatelské parametry systému. Může to být například pravidlo pro řazení položek čekajících na vyřízení, dále třeba čas, po kterém je nevyřízená položka zvýrazněna a další nastavení.

Tabulka obsahuje následující sloupce.

- nazev – Název daného parametru je zároveň primárním klíčem tabulky.
- hodnota – Uživatelem nastavená hodnota.

Vazby mezi tabulkami

Tabulka *jidelniListek* má vazbu do skladu, k daňové třídě, měně a jednotce. U každé položky v jídlením listu je totiž potřeba tyto informace uchovávat. Dále má vazbu na kuchyň, která slouží k určení, ve které kuchyni se položka připravuje a kam se má zaslat požadavek k jejímu zpracování. Nakonec je vazba k typu stravy. Tato vazba je přes spojovací tabulku, díky které je možné, aby položka spadala do více typů. Tabulka *objednávka* má vazbu na stůl, na kterém byla vytvořena. Tabulka *objednavka_jidelniListek* vytvoří po jejím zaplacení vazbu do tabulky s pokladními doklady, kde je tím přiřazena na jeden konkrétní doklad. Dále má položka vazbu do jidelního lístku, což určuje konkrétní položku a vazbu k objednávce, ke které objednané jídlo patří.

5.5 Modulové rozdělení

Aplikace je rozdělena do třech logických modulů. Nette framework má pak pro každý modul vlastní presentery a šablony. Nad všemi moduly je jediný model. Výhodou je, že v každém modulu dědí všechny presentery od BasePresenteru a díky tomu mohu nějaké vlastnosti, jako například uživatelské přihlášení naimplmentovat do něj a promítnou se v celém modulu, ale ne v modulech ostatních. Další výhodou je vlastní layout šablona pro každý modul a tím pádem, bezproblémové nastavení různých layoutů pro jednotlivé moduly. V neposlední řadě je systém pro programátory mnohem přehlednější. Rozhodl jsem se systém rozdělit do třech modulů:

- modul Administrace,
- modul Pokladna a
- modul Zákazník.

V modulu Administrace je veškeré nastavení systému, včetně generování statistik, XML výstupů a QR kódů. V modulu Pokladna je vše co bude přístupné z mobilního číšníka, hlavní pokladny a kuchyně. Tento modul obsahuje objednávání a vyřizování objednávek, platby a vystavování pokladních dokladů. Poslední modul je pro přijímání a zpracovávání QR kódů s objednávkami zaslánými přímo zákazníkem restaurace.

5.6 Modul Administrace

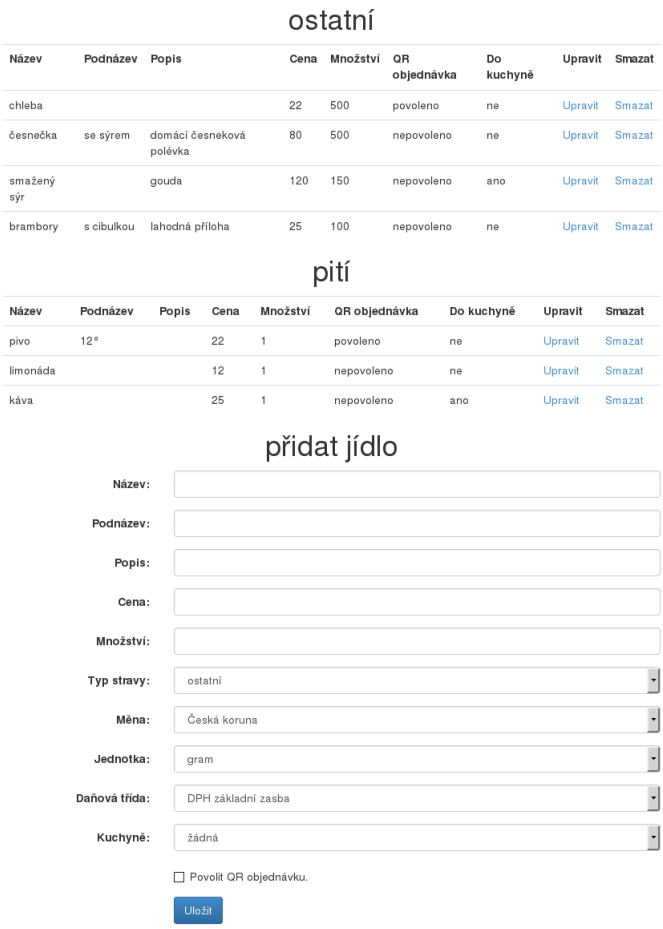
V modulu administrace lze přidávat a editovat položky jídelního lístku včetně typů, měn, jednotek a sazby DPH. Stejně tak lze editovat a přidávat jednotlivé stoly. Dále v administraci nastavujeme parametry systému, které jsou uloženy v tabulce *konfigurace*. Důležitou součástí jsou XML výstupy. A to XML jídelního lístku pro případné naimportování na web restaurace, ale také XML pro import účetních dokladů do účetního serveru Pohoda.

V administraci systému můžeme také sledovat statistiky obratu a prodejnosti produktů. Obrat restaurace lze sledovat v námi zvoleném časovém období. U prodejnosti jednotlivých produktů můžeme sledovat, konkrétní položky, či celé skupiny položek. Sledujeme kdy se nejvíce objednávají a to pro jednotlivé měsíce, dny, či hodiny. Díky tomu můžeme velice efektivně upravovat sezonní jídelní lístky, případně informační tabule v restauraci nebo různé slevové akce.

V již zmíněné části editace jídelního lístku můžeme u každého jídla nastavit, že je možno objednat jej pomocí QR kódů. Přehled všech QR kódů pak nalezneme v části správa QR kódů. Pro každý stůl jsou zde vygenerovány kódy všech povolených jídel. Tyto kódy může pak grafik, který zpracovává jídelní lístek, jednoduše stáhnout a zakomponovat do nabídky restaurace.

Editace jídelního lístku

V sekci editace jídelního lístku lze přidávat nové položky do jídelního lístku, měnit či mazat stávající. Dále lze měnit, přidávat a mazat jednotlivé typy položek a ostatní parametry.



Obr. 12: Editace jídleního lístku.

Při vkládání do jídelního lístku se zapisuje do tabulky *jidelniListek* a dále do tabulek *typStravy_jidelniListek*, *sklad*, *mena*, *danovaTrida* a *jednotka*.

Při mazání položky se vždy smažou i všechny vazby do tabulky *typStravy*. Položka zůstane v databázi, ale ve sloupečku *smazano* se hodnota změní na jedničku. Pokud chci smazat typ stravy, provede se nejdříve kontrola jestli na daný typ neexistuje vazba z jídelního lístku, pokud existuje, smazání není možné. U ostatních atributů navázaných na další tabulky to funguje analogicky. Díky tomu nemůže vzniknout v databázi nekonzistence.

O celý tento proces se stará třída *JidelniListek*, která se stará o vykreslování a generování formulářů a jejich následnou kontrolu a zpracování. Zkrácený formulář pro vložení nového jídla vypadá následovně.

```
$form = new UI\Form;
$form->addText('nazev', 'Název:')
    ->setRequired('Zadejte název jídla');
$form->addText('popis', 'Popis:');
$form->addText('podnazev', 'Podnázev:');
$form->addText('cena', 'Cena:')
    ->setRequired('Zadejte cenu jídla')
    ->addRule(UI\Form::INTEGER, 'Cena musí být číslo');
$form->addText('mnozstvi', 'Množství:')
    ->setRequired('Zadejte množství jídla')
    ->addRule(UI\Form::INTEGER, 'Množství musí být číslo');

$typStravy = $this->database->table('typStravy')->fetchPairs('id', 'nazev');
$form->addSelect('typStravy', 'Typ stravy:', $typStravy);

$mena = $this->database->table('mena')->fetchPairs('id', 'nazev');
$form->addSelect('mena', 'Měna:', $mena);

$jednotka = $this->database->table('jednotka')->fetchPairs('id', 'nazev');
$form->addSelect('jednotka', 'Jednotka:', $jednotka);

$form->addSubmit('ulozit', 'Uložit');
$form->onSuccess[] = $this->jidelniListekFormSucceeded;
```

Nejprve se do proměnné *form* uloží nová instance UI\Form a do ní se následně přidávají jednotlivé prvky. Metoda *addText* přidá do fomuláře HTML entitu *text input*. Metoda *setReguired* nastaví formulářový prvek jako poviný. Parametrem metody je chybové hlášení při nevyplnění pole. Velmi mocnou a jednoduchou metodou je *addRule* pomocí níž lze efektivně přidávat pravidla pro validaci vložené hodnoty. Lze použít buď standardní validační metodu obsaženou v Nette frameworku, nebo můžeme hodnotu validovat regulárním výrazem.

V druhé části formuláře je metoda *addSelect* která přidá formulářový prvek *select*. Před každým zavoláním metody se nejdříve naplní pole možností z patřičné tabulky v databázi. Nakonec se vygeneruje odesílací tlačítko a zadá se metoda, která bude formulář zpracovávat.

Kdybychom si takovýto formulář zkusili naprogramovat v čistém php včetně všech validací a escapování hodnot, vyprodukovali bychom mnohem delší, složitější a méně přehledný kód. Zde je vidět, že se díky frameworku můžeme zaměřit na to co skutečně potřebujeme (v tomto případě tvorba formuláře) a nemusíme se zdržovat rutinními úlohami jako je escapování a validace vložených hodnot.

Tab. 2: Přehled validačních pravidel obsažených v Nette Frameworku. (GRUDL, David.)

Validační pravidlo	**Popis pravidla**
Form::FILLED	Je prvek vyplněn?
Form::EQUAL	Je hodnota rovna uvedené?
Form::IS$_IN$	Testuje, zda hodnota spadá do výčtu.
Form::VALID	Je prvek vyplněn správně?
Form::MIN$_LENGTH$	Minimální délka.
Form::MAX$_LENGTH$	Maximální délka.
Form::LENGTH	Délka v daném rozsahu nebo právě tato délka.
Form::EMAIL	Je hodnota platná e-mailová adresa?
Form::URL	Je hodnota absolutní URL?
Form::PATTERN	Test oproti regulárnímu výrazu.
Form::INTEGER	Je hodnota celočíselná?
Form::FLOAT	Je hodnota číslo?
Form::RANGE	Je hodnota v daném rozsahu?

Editace stolů

V této sekci lze upravovat atributy jednotlivých stolů, mazat či přidávat stoly. Při mazání stolu se kontroluje jestli na něm není aktivní objednávka. I když se nepředpokládá, že se stoly budou měnit za provozu restaurace, jsou všechny situace ošetřené aby databáze byla vždy v pořádku.

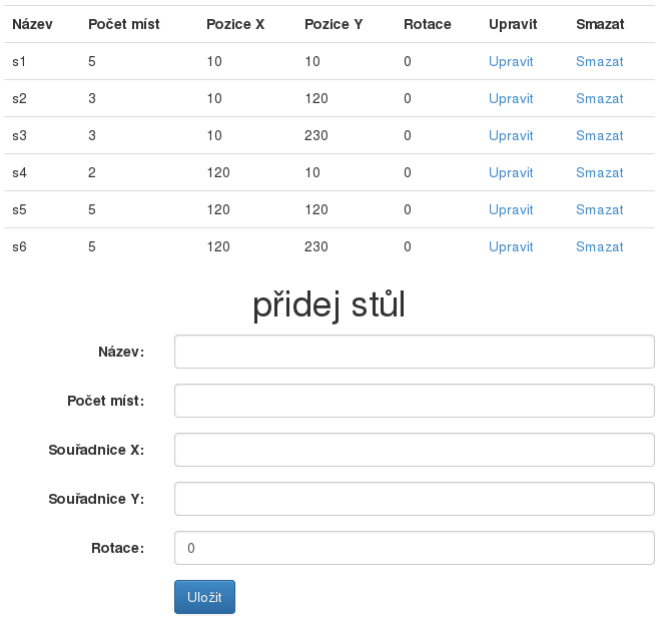
Pokud chci upravovat parametry stolu, zavolám metodu *renderUpravStul*. V následujícím kódu můžete vidět jak se v metodě kontroluje, zda na stole není nějaká objednávka. V případě že je, zavolá se metoda *flashMessage*, která vypíše varovné hlášení a následně nás *redirect* přesměruje zpět na šablonu default. Nakonec do šablony předáme id stolu, který chceme upravovat aby v případě odeslání formuláře k úpravě stolu odešla i informace o jeho id.

```
public function renderUpravStul($id_stul)
{
  $jeAktivni = $this->database->table('objednavka')
    ->where('id_stul', $id_stul)
    ->where('odchod', '0');
  if($jeAktivni->fetch())
  {
    $pouzivan = 1;
    $this->flashMessage('Na stole jsou otevřené objednávky.', 'error');
    $this->redirect('default');
  }
  $this->template->id_stul = $id_stul;
}
```

Následné potvrzení úprav už je poměrně jednoduchý SQL dotaz.

```
$values = $formUprav->getValues();
$row = $this->database->table('stul')->get($values->id_stul);
$row->update(array(
    'nazev' => $values->nazev,
    'pocetMist' => $values->pocetMist,
    'poziceX' => $values->poziceX,
    'poziceY' => $values->poziceY,
    'rotace' => $values->rotace,
));
$this->flashMessage('Úspěšně upraveno.', 'success');
$this->redirect('default');
```

Vkládání nového stolu vypadá prakticky stejně. Není ale třeba nic kontrolovat a to ani za plného provozu restaurace. Stoly je možné ze sytému také mazat. Před mazáním se provede kontrola zda-li na něm není aktivní objednávka, pokud není zapíše se do sloupce *smazano* jednička a pak se stůl již nikde nezobrazuje.



Obr. 13: Editace stolů.

Editace kuchyní

Tato sekce slouží k přidávání, úpravě a mazání kuchyní. Jako kuchyň nemusíme brát celou místnost, ale můžeme mít jednu fyzickou kuchyň rozdělenou na jednotlivé úseky a ke každému úseku můžeme zobrazovat relevantní objednávky. Kuchyní

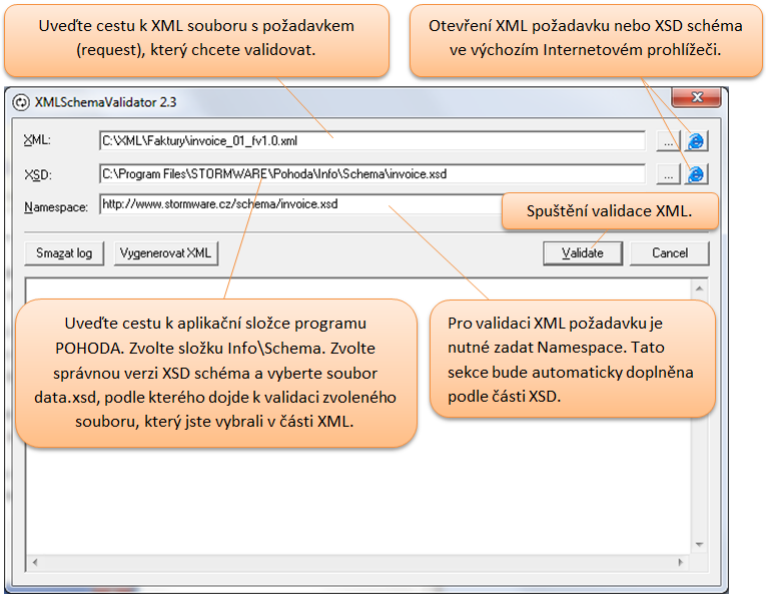
v systému nebude moc a pravděpodobně se nebudou ani často měnit. Tato část aplikace se tedy bude používat především při prvotním nastavení systému.

XML výstupy

XML výstup jídleního lístku má kořenový tag *jidelniListek* a každá položka je uzavřena v tagu *polozkaJidelnihoListku*. Další vnořené tagy mají názvy odpovídající názvům tabulek v databázi. XML jísleního lístku s jednou položkou by vypadalo následovně.

```
<jidelniListek>
  <polozkaJidelnihoListku>
    <nazev>Limonáda</nazev>
    <podnazev>pomerančová</podnazev>
    <popis>Popis produktu<popis/>
    <cena>22</cena>
    <mena>CZK<mena/>
    <mnozstvi>3</mnozstvi>
    <jednotka>dcl<jednotka/>
  </polozkaJidelnihoListku>
</jidelniListek>
```

XML výstup pro účetní systém Pohoda je vygenerován na základě specifikace účetního softwaru. Pro otestování XML výstupu je k dispozici XML Schema Validator.



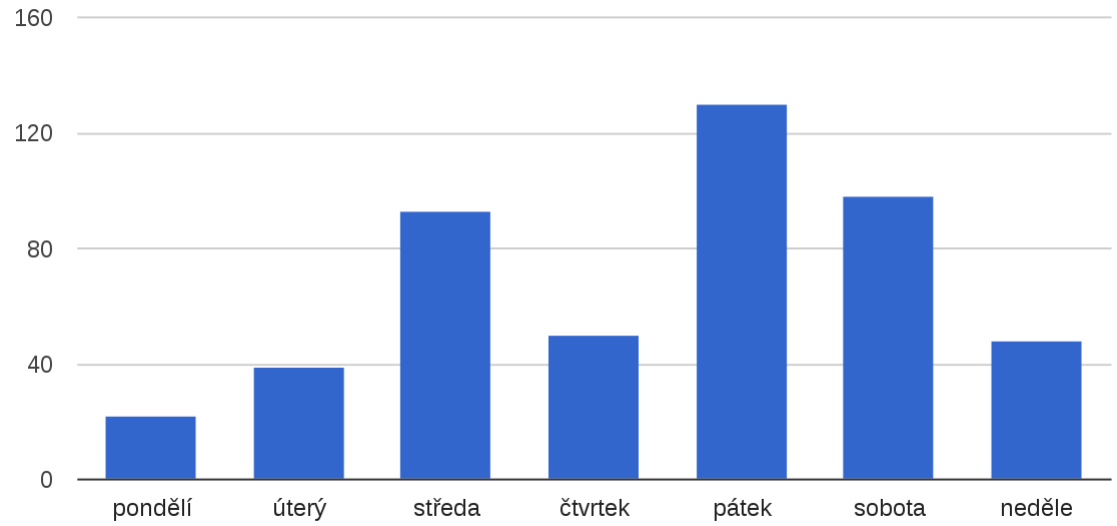
Obr. 14: Ukázka XML Schema Validatoru. Zdroj: http://www.stormware.cz/xml/xmlvalidator.aspx

Statistiky

Ve statistikách lze sledovat vývoj obratu a tím okamžitě zjistit zda-li se restauraci daří či nikoliv. O něco zajímavější statistikou je ale prodejnost jednotlivých produktů.

Uživatel si nastaví, které produkty ho zajímají, vybere datové rozpětí se kterým chce pracovat a jestli chce výstup vypsat po měsících, dnech nebo hodinách. Generování statistik funguje tak, že se nejdříve vyberou všechny prodeje v daném časovém rozpětí z databáze a následně se roztřídí do požadovaného zobrazení (hodiny, dny, měsíce), takto roztříděná data se zobrazí v grafu.

Grafy jsou v aplikaci generovány pomocí Google Charts. Díky této službě stačí zvolit typ grafu, naplnit ho daty a zobrazit. Grafy jsou navíc interaktivní, takže po najetí myší se zobrazí doplňující informace. Více informací o Google Charts je na stránkách projektu https://developers.google.com/chart/.



Obr. 15: Ukázka statistiky počet objednaných piv po jednotlivých dnech v týdnu.

QR kódy

Zkratka „QR“ pochází z anglického „quick response“ neboli „rychlá reakce“. Charakteristickým prvkem QR kódu jsou detekční vzory umístěné ve třech rozích sloužící k určení polohy a umožňujici snímání v různých polohách natočení.

Podle množství zakódované informace je vygenerována jedna ze čtyřiceti odpovídajících velikostí symbolu, které se také nazývají verze. Čím vyšší je číslo verze, tím větší je počet modulů (černé a bílé čtverečky).

Moduly jsou nejmenší prvky výsledného symbolu, černý čtvereček odpovídá logické jedničce a bílý nule. Nejmenší verze 1 má 21 ku 21 modulů, největší verze

40 má 177 ku 177 modulů. S každým vyšším číslem verze přibývají 4 další moduly v obou směrech.

Kolem symbolu musí být dodržena klidová zóna neboli prostor bez tisku, o velikosti 4 modulů na všech stranách.

Pro aplikace s malým množstvím dat můžeme zvolit Micro QR kód, který je ve velikostech 11 ku 11 až 17 ku 17 modulů a obsahuje jen jeden detekční vzor.

V každém QR kódu je daná informace uložena s určitou redundancí. Díky tomu lze kód přečíst i v případě špatného načtení za horších světelných podmínek. Při generaci kódu se dá nastavit jedna ze 4 úrovní „odolnosti“ proti chybám:

- úroveň L – správně přečte i v případě až 7 % poškozené plochy,
- úroveň M – správně přečte i v případě až 15 % poškozené plochy,
- úroveň Q – správně přečte i v případě až 25 % poškozené plochy,
- úroveň H – správně přečte i v případě až 30 % poškozené plochy.

(ŠOŠOLÍK, Petr.)

Pro tuto aplikaci by kapacita MicroQR kódu nestačila, generované kódy mají velikost 37 ku 37 modulů. Jako úroveň odolnosti proti chybám jsem zvolil úroveň M. Vzhledem k tomu, že se QR kódy budou snímat v interiéru, nebude vytisklý kód příliš ztrácet kvalitu a světelné podmínky budou poměrně dobré. Pro tyto podmínky by bylo zbytečné volit verzi s vyšší redundancí.

Tato část aplikace tedy slouží k vygenerování a následném zobrazení všech QR kódů, které systém akceptuje. Zobrazuje QR kódy pro každé povolené jídlo ke každému stolu. QR kódy jsou generovány do bezztrátového formátu PNG o šířce 320px včetně bílých okrajů.

Možností implementace je hned několik. Jako tu nejjednodušší bych označil generování javaScriptem s knihovnou jQuery a pluginem *jquery.qrcode*. Celá implementace by vypadalo následovně.

```
<script type="text/javascript" src="jquery.qrcode.min.js"></script>
<div id="qrcode"></div>
jquery('#qrcode').qrcode("Text/odkaz, který QR kód obsahuje.");
```

(jquery.qrcode)

Je to velice jednoduché a rychlé řešení, které potěší především kodéry. Toto řešení ale generuje kódy až na klientské stanici a tím je méně spolehlivé.

Další dvě řešení, která jsem zkoumal generují QR kódy v PHP za pomocí knihovny GD2. Vše se tedy odehrává na serveru a na klientský počítač se odesílá už hotový obrázek. První řešení je *PHP QR Code*, které je poměrně propracované, dobře zdokumentované ale jeho poslední aktualizace je z roku 2010. Proto jsem zvolil druhé řešení, které je stále aktualizované (poslední commit na GitHubu je před dvěma měsíci) a subjektivně mi přišlo přívětivější. (PHP QR Code) (endroid/QrCode)

Obr. 16: Takto vypadá QR kód pro objednání chleba na stole číslo 1.

Konfigurace

V sekci konfigurace mohu nastavovat jednotlivé parametry systému. Konfiguraci obsluhuje třída *KonfiguracePresenter*, která funguje velice jednoduše. Pouze se v ní generuje formulář, ve kterém se parametry nastavují a po jeho zpracování zaktualizuje tabulku konfigurace. Třída používá pouze jednu zobrazovací šablonu.

5.7 Modul Pokladna

Modul pokladna slouží k zadávání a vyřizování objednávek, přijímání plateb a vystavování pokladních dokladů. První jeho částí je zobrazení stolů a přiřazených objednávek. Druhou částí je zobrazení konkrétní objednávky, práce s ní. A třetí je zobrazení všech objednávek a jejich vyřizování a připravování. Tento modul je připraven k nasazení na počítač u baru, v kuchyni a na tablet, případně větší mobilní telefon (mobilní číšník). Z toho důvodu je modul pokladna nakódován v responzivním designu.

Seznam stolů

Seznam stolů nám zobrazí veškeré stoly v grafické podobě na jejich pozicích a s případnou rotací. Na každém stole se zobrazuje informace o jeho názvu, počtu míst a především přiřazených objednávkách. U každého stolu mohu vytvořit další objednávku případně si otevřít objednávku již vytvořenou. Na obrázku 17 můžeme vidět stoly vložené v aplikaci.

Práce s databází je v Nette/Database poměrně jednoduchá a přehledná. Pro vytvoření objednávky stačí jednoduchý kód, který načte aktuální datum, jako parametr potřebuje id stolu, kde se objednávka vytváří a poslední sloupec nutný pro úspěšné vytvoření objednávky na stole je její id, které se vytvoří pomocí auto increment.

```
public function handlePrichod($id_stul){

    $data = [
        'casVytvoreni' => new DateTime,
        'id_stul' => $id_stul
        ];

    $this->database->table('objednavka')
        ->insert($data);

    $this->redirect('this');

}
```

Tento kód je z třídy *VyberStul*. Tato třída se dále stará o uzavření objednávky, kdy kontroluje, zda-li jsou všechny objednávky vyřízené a hlavně zaplacené. V případě, že objednávka obsahuje nevyřízené objednávky, systém číšníka upozorní, ale i tak je možné pokračovat a objednávku uzavřít. V případě, že se v objednávce vyskytují vyřízené a nezaplacené objednávky, systém nedovolí uzavřít stůl. Při odchodu se nejdříve zavolá kontrolní metoda *handlePozadavekOdchod* a v případě, že je vše v pořádku provede se uzavření objednávky metodou *handleOdchod*. Tato třída se dále stará o vykreslení seznamu stolů a o vykreslení nabídky k požadované objednávce.

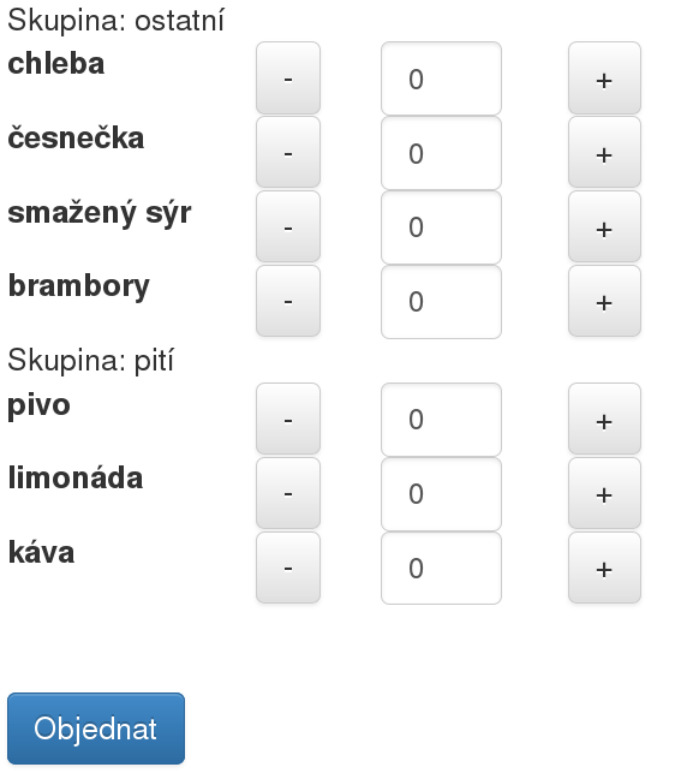
Obr. 17: Seznam všech stolů v restauraci.

Práce s objednávkou

Po otevření vytvořené objednávky lze objednávat jednotlivé položky, prohlížet si objednané položky a jejich stav či zaplatit část, nebo celou objednávku. Pokud je vše zaplaceno, mohu objednávku uzavřít.

V případě objednávání vykreslí prezenter *Nabidka* formulář, po jehož odeslání se volá metoda *seznamJidelFormSucceeded* která všechny označené položky uloží do objednávky, tedy vytvoří záznamy v tabulce *objednavka_jidelniListek*.

Formulář nabídky jídel se vytváří ve foreach cyklu, kde se všechny položky řadí podle skupin do kterých náleží. Díky tomu se nabídka zobrazí přehledně ve skupinách.



Obr. 18: Objednávání jednotlivých položek.

V sekci objednáno jsou k nahlednutí všechny objednané položky seřazeny podle stavu (vyřízeno/nevyřízeno) a dále pak podle data přijetí. Každou objednávku lze označit jako vyřízenou případně ji smazat. K tomu jsou vytvořeny dvě metody *handleDeleteObjednavka* a *handleSetObjednavkaVyrizeno*. Obě tyto akce ale bude většinou provádět číšník u baru u přehledu všech objednávek, kde je celý přehled propracovanější. Tato část aplikace bude sloužit spíše pro informování zákazníků co všechno mají objednané a vyřízené. Určitě se ale najdou i situace, kdy vznikne nějakým překlepem či nepozorností chybná objednávka a číšník ji bude moct jednoduše smazat přímo ze zařízení mobilního číšníka.

V sekci platby lze platit buď objednávky po jedné, nebo vše najednou. Ke každé platbě se generuje platební doklad. Proces placení funguje tak, že se nejdříve vyberou položky které chce zákazník zaplatit (v případě, že se bude objednávka platit po částech). Tyto položky se vloží do pole *polozkaZaplatit* pomocí metody *handlePolozkaZaplatit* a následně označí jako zaplacené metodou *handleProvedeniPlatby*, která ihned po zaplacení nabídne tisk účtenky, případně její odeslání na email zákazníka. V případě, že jsou všechny vyřízené položky zaplacené lze objednávku uzavřít.

Přehled všech objednávek

V tomto přehledu jsou všechny objednávky napříč celou restaurací. Tato část aplikace je navržena pro spuštění na počítači umístěným na baru a to z toho důvodu, že zobrazuje přehledně všechny objednávky podle času přijetí. Zaroveň se barevně zvýrazňují ty, které už na vyřízení čekají příliš dlouho. Číšník, který je zrovna na baru může na přijaté objednávky obratem reagovat.

Pokud je nějaká objednávka zaslána samotným zákazníkem pomocí QR kódu, je vedle objednávky ikona, která tuto skutečnost zdůrazní. Seznam všech objednávek je AJAXově překreslován každou vteřinu. To zajišťuje jednoduchý javaScriptový, který funkcí *setInterval* periodicky volá metodu *handlePrekresli*.

```
<script>
    setInterval(function() {
        $.nette.ajax({link prekresli!});
    }, 1000);
</script>
```

Nette.ajax.js je doplněk Nette frameworku pro snažší práci s AJAXovými požadavky. V Nette frameworku se pracuje s takzvanými snippety, což jsou výřezy šablony které lze AJAXově posílat. V tomto případě je jako snippet *objednávky* označena tabulka nevyřízených objednávek. Tato část stránky se periodicky aktualizuje, zbytek ovšem zůstává statický. Díky tomu se přenáší jen ta data, která je potřeba aktualizovat.

Na obrázku 19 jsou vidět všechny nevyřízené objednávky systému podbarvené podle toho, jak dlouho čekají. Barva podbarvení i čas, kdy se má objednávka zvýraznit se nastavuje v konfiguraci systému.

Přehled objednávek do kuchyně

Tento přehled je velmi podobný přehledu všech objednávek. Zobrazuje ale pouze položky, které čekají na přípravu v dané kuchyni a u každé umožňuje nastavit ji jako připravenou, případně ji odmítnout. Stejně jako u přehledu všech objednávek se snippet s přehledem AJAXově neustále aktualizuje.

Nevyřízené

Položka	Vyřízeno	Smazat	č.stolu	čeká	
chleba	není - vyřídit	smazat	2	před okamžikem	
česnečka	není - vyřídit	smazat	5	před okamžikem	
káva	není - vyřídit	smazat	3	před minutou	
brambory	není - vyřídit	smazat	5	před minutou	
pivo	není - vyřídit	smazat	1	před 9 minutami	
chleba	není - vyřídit	smazat	4	před 14 dny	
smažený sýr	není - vyřídit	smazat	4	před 14 dny	

Vyřízené

Položka	Vyřízeno	Smazat	č.stolu
česnečka	je	smazat	4
káva	je	smazat	4

Obr. 19: Přehled všech objednávek v restauraci.

5.8 Modul Zákazník

Modul zákazník se stará pouze o přijímání objednávek zaslaných zákazníky pomocí QR kódů. Modul zkontroluje přijatý GET požadavej, zda-li je v něm obsažena položka, která má povolenou QR objednávku a zkontroluje jestli na stole, ze kterého je objednávka odeslána existuje právě jedna aktivní objednávka. Pokud je vše v pořádku uloží objednávku do databáze a označí ji jako objednanou pomocí QR kódu. V sekci konfigurace se dá také nastavit IP adresa, ze které jsou objednávky akceptovány, je to z toho důvodu aby šlo objednávky objednávat pouze z restaurace, respektive z wifi sítě v restauraci.

5.9 Nasazení aplikace do restaurace

Na realizaci celé aplikace budou použity 4 osobní počítače a dva tablety. Jeden počítač bude sloužit jako server a manažerská stanice v jednom. Bude se jednat o plnohodnotnou stanici v PC skříni Midi Tower s operační pamětí 4–8 GB, SSD 128 GB pro běh operačního systému. Dále bude v počítači HDD 500 GB pro případné zálohy databáze a pro potřeby manažera restaurace. Na stanici bude nainstalován OS Linux Debian, na kterém poběží webový server Apache a MySQL databázový server. Další dva počítače budou do kuchyně a poslední počítač bude na baru jako hlavní pokladna. Pro tyto tři stanice jsem vybral dotykový 19,5" all in one PC *Lenovo IdeaCentre N300 Touch Black* s OS Google Android. Na těchto stanicích

poběží pouze internetový prohlížeč, takže stačí slabší stroj, za to je velmi důležitá dotyková obrazovka. Jako mobilní číšníky, jsem vybral tablety s velikosti úhlopříčky 7" taktéž s OS Google Android.

Všechna zařízení budou síťově propojena bez nutnosti připojení k síti Internet.

6 Diskuze

Výsledkem této závěrečné práce je komplexní pokladní systém pro restaurace a podobná stravovací zařízení. Aplikaci se podařilo dotáhnout do stavu, kdy je možné provést její záverečné otestování v provozu a následné ostré spuštění. Pokladní systém je naimplementován jako webová aplikace, což zaručuje naprostou multiplatformnost a možnost spuštění na téměř všech počítačích i tabletech. Kromě serveru není třeba na stanice nic instalovat. To zaručuje velmi nízkou cenu a bezproblémovou a rychlou výměnu vadné stanice za novou (v případě náhlé poruchy není problém využít jakýkoliv notebook).

Výhodou systému je možnost zasílání objednávek prostřednictvím dnes populárních QR kódů, což může být zvláště pro mladé lidí zajímávé vylepšení, které může zaujmout. Další užitečnou funkcí je možnost přiřazení objednávek do různých kuchyní. Takto si může velmi efektivně rozdělovat práci i více různě specializovaných kuchařů v jedné kuchyni. Celá webová aplikace je responzivní a to z důvodu možnosti použít mobilního číšníka. Díky tomu se restaurace úplně obejde bez papírových objednávek a může se vše zadávat rovnou do systému. Díky použití Nette frameworku je aplikace dobře zabezpečená a snadno rozšiřitelná.

Jako jedno z prvních rozšíření aplikace bude zpracování účetních dokladů. Momentálně systém nabízí XML export účetních dokladů do ekonomického softwaru Pohoda, ale samotná aplikace účetní doklady neumí nijak zpracovat. To bude vyžadovat mimo jiné i spolupráci s odborníkem na předmětnou doménu, který bude schopen přesně specifikovat požadavky na informace poskytované systémem i na samotný proces zpracování.

7 Závěr

Cílem práce byl návrh a implementace restauračního systému pro správu objednávek. Tento cíl se splnit podařilo. Na samém začátku jsem provedl analýzu všech hotových řešení, nastudoval jejich funkce a snažil se zjistit na jakých technologiích se dají provozovat. Dále jsem si zvolil prostředky, které k tvorbě aplikace využiji. Aplikaci jsem vytvořil jako webovou v jazyce php. A to z důvodu multiplatformnosti, nutnosti instalace aplikace pouze na server a díky responzivnímu designu jednoduché zprovoznění na mobilních zařízeních. Následně jsem si k programování zvolil framework Nette. Programování bez frameworku by bylo mnohem náchylnější na vnešení bezpečnostích děr a musel bych řešit spoustu rutinních záležitostí. Díky frameworku jsem se mohl více soustředit na to co má aplikace umět a méně na zabezpečení aplikace, validaci dat a možnost případného jedoduchého rozšíření aplikace. Nette framework jsem zvolil z důvodu jeho rychlosti, bezpečnosti a taky vzhledem k jeho velké a ochotné české komunitě.

K návrhu aplikace jsem využil UML diagramů, kde jsem nejdříve vytvořil Use case diagramy, dle kterých celý vývoj pokračoval. Snažil jsem se řídit spirálovým vývojovým modelem, díky kterému měl vývoj určitý řád. K návrhu databáze jsem zvolil entitně relační diagram.

K designu aplikace a především k řešení její responzivity jsem využil nástroj Twitter Bootstrap (soubor stylů a javaSriptů). Díky němu se dá responzivní design řešit opravdu elegantně.

Systém umí přijímat a zpracovávat objednávky. Pokud je to u dané objednávky nastaveno, zašle ji do kuchyně. Objednávky mohou zadávat i samotní návštěvníci pomocí QR kódů uvedených například v jídelním lístku. Všechny objednávky lze platit po částech, kdy je ke každé části vystaven pokladní doklad, ten může být vytištěn, nebo poslán zákazníkovy emailem. Provozní restaurace si může prohlížet statistiky obratu a prodejnosti jednotlivých produktů, může generovat XML pokladních dokladů do ekonomického softwaru Pohoda a také si může jeho webová prezentace stahovat aktuální jídelní lístek ve formátu XML.

Nakonec měl být systém nasazen ve vinárně U Leopolda. Vzhledem k odložení jejího otevření na léto se nasazení nemohlo uskutečnit. Stále, ale věřím, že se vinárna otevře a tento restaurační systém v ní bude použit. Pro mě osobně to bude úspěch a těším se na doladění celé aplikce dle konrétnějších představ provozovatele vinárny. Této práci vděčím především za hlubší porozumnění Nette frameworku, který jsem na začátku teprve poznával. Ve vývoji webových aplikací jsem se díky tomuto vývoji znatelně posunul vpřed.

8 Literatura

PONKRÁC, Miloslav *PHP a MySQL: bez předchozích znalostí : [průvodce pro samouky]* Vyd. 1. Brno: Computer Press, 2007, 221 s. ISBN 978-80-251-1758-3..

LAVIN, Peter *PHP - objektově orientované: koncepty, techniky a kód.* 1. vyd. Praha: Grada, 2009, 211 s. ISBN 978-80-247-2137-8..

CASTRO, Elizabeth a Bruce HYSLOP *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek.* Vyd. 1. Brno: Computer Press, 2012, 439 s. ISBN 978-80-251-3733-8..

MARGORÍN, Marián *JQuery bez předchozích znalostí : [průvodce pro samouky]* Vyd. 1. Brno: Computer Press, 2011, 253 s. ISBN 978-80-251-3379-8..

BOEHM, Barry W. *A Spiral Model of Software Development and Enhancement.* Wiley-IEEE Computer Society Press. 2007. 832 s. ISBN: 978-0-470-14873-0.

BĚHOUNKOVÁ, Martina *Srovnání nejpoužívanějších účetních softwarů v ČR z hlediska možnosti zpracování účetních dat.* České Budějovice, 2007. Diplomová práce. JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH. Vedoucí práce Ing. Hana Hlaváčková..

ŠOŠOLÍK, Petr *Použití dvourozměrných kódů v praxi.* Zlín, 1. 6. 2009. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Bronislav Chramcov..

CROFT, Jeff *Frameworks for Designers. In: A List Apart [online].* 12. 6. 2007 [cit. 2014-05-17]. Dostupné z: http://alistapart.com/article/frameworksfordesigners.

DANĚK, Petr *Root.cz: Velký test PHP frameworků: Zend, Nette, PHP a RoR [online].* 11. 9. 2008 [cit. 2014-05-17]. Dostupné z: http://www.root.cz/clanky/velky-test-php-frameworku-zend-nette-php-a-ror.

VŠETEČKA, Roman *mobil.idnes.cz [online].* 5. 1. 2010 [cit. 2014-04-12]. Dostupné z: http://mobil.idnes.cz/podil-mobilni-operacnich-systemu-2014-dbu-/mob_tech.aspx?c=A140130_175418_mob_tech_vok.

GRUDL, David *Nette framework-Úvod [online].* c 2008–2014 [cit. 2014-05-17]. Dostupné z: http://www.nette.org.

ČÁPKA, David *UML - Use Case Diagram [online].* c 2014 [cit. 2014-12-15]. Dostupné z: http://www.itnetwork.cz/uml-use-case-diagram.

Wikipedie: Otevřená encyklopedie [online]. San Francisco (CA): Wikimedia Foundation, 1.5.2014 [cit. 2014-05-10]. Dostupné z: http://cs.wikipedia.org/wiki/PHP.

JavaScript [online]. 13. 5. 2014 [cit. 2014-05-17]. Dostupné z: https://developer.mozilla.org/en/docs/Web/JavaScript.

Cascading Style Sheets [online]. c 1994–2014, 23.10.2014 [cit. 2014-10-31]. Dostupné z: http://www.w3.org/Style/CSS.

Extensible Markup Language [online]. 10.3.2014 [cit. 2014-10-31]. Dostupné z: http://www.w3.org/XML.

Introduction to XML [online]. c 1999–2014 [cit. 2014-10-31]. Dostupné z: http://www.w3schools.com/xml/xml_whatis.asp.

XML komunikace s ekonomickým systémem POHODA [online]. c 2014 [cit. 2014-10-31]. Dostupné z: http://www.stormware.cz/xml.

jquery.qrcode - jquery plugin for pure browser qrcode generation on Jerome Etienne blog [online]. 7. 4. 2011 [cit. 2014-12-01]. Dostupné z: http://notes.jetienne.com/2011/04/07/jquery-qrcode.html.

PHP QR Code - QR code generator, an LGPL PHP library [online]. c 2006–2010 [cit. 2014-12-01]. Dostupné z: http://phpqrcode.sourceforge.net.

endroid/QrCode · GitHub [online]. c 2014 [cit. 2014-12-01]. Dostupné z: https://github.com/endroid/QrCode.

Zdroje k restauračním systémům

Váhy-Pokladny-HARON [online]. 2013 [cit. 2014-03-16]. Dostupné z: http://www.vahy-pokladny-haron.com/pokladny-restauracni.php?ukaz=Katalog-SHARP_XE-A217B.

ABX software [online]. c 2009 [cit. 2014-03-16]. Dostupné z: http://www.abx.cz/171-pokladna_abx_1.

Orderman-Accessories [online]. c 2014 [cit. 2014-03-16]. Dostupné z: http://www.orderman.com/en/handhelds/leo2plus/accessories.html.

Mobilní terminály a přenosné pokladny [online]. 2013 [cit. 2014-03-16]. Dostupné z: http://www.mobilni-cisnik.cz/p/software-program.html.

AWIS pokladní systémy [online]. c 2013 [cit. 2014-03-16]. Dostupné z: http://www.kasa-pokladna.cz/cs/.

ASW Systems [online]. [cit. 2014-03-29]. Dostupné z: http://www.septim.cz/cs/reseni/reseni_pro_restaurace.

Mobilní pokladny – ASW Systems [online]. c 2008 [cit. 2014-03-22]. Dostupné z: http://www.septim.cz/cs/produkty/pokladny/mobilni_pokladny.

AWIS pokladní systémy [online]. c 2013 [cit. 2014-03-29]. Dostupné z: http://www.kasa-pokladna.cz/cs/pokladni-systemy-pro-restaurace-kluby-bary.

AWIS pokladní systémy [online]. c 2013 [cit. 2014-03-29]. Dostupné z: http://www.pokladny-systemy.cz/modul-mobilni-cisnik.

fusion [online]. [cit. 2014-03-29]. Dostupné z: http://www.horeca-fusion.cz/pokladni-systemy-cisnicka-pokladna.

Restaurační systémy Agnis [online]. c 2010 [cit. 2014-03-29]. Dostupné z: http://www.restauracni-systemy.cz/system-agnis/pokladna.html.

Restaurační systémy Agnis [online]. c 2010 [cit. 2014-03-29]. Dostupné z: http://www.restauracni-systemy.cz/system-agnis/.

ABX software [online]. c 2009 [cit. 2014-03-30]. Dostupné z: http://www.ab-x.cz/40-pokladni_system_harsys_6_-_podrobny_popis.

Menu 55 [online]. c 2013 [cit. 2014-03-30]. Dostupné z: http://www.menu55.cz/index.php/funkce-restauracniho-systemu.