

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Univerzální systém pro správu restauračních zařízení

Bc. Daniel Poláček

Diplomová práce

2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Daniel Poláček**
Osobní číslo: **I11404**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Univerzální systém pro správu restauračních zařízení**
Zadávající katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem této práce je návrh systému a vytvoření funkční aplikace, která bude umožňovat kompletní správu restauračního zařízení bez ohledu na typ a velikost tohoto zařízení. Systém bude řízen skupinou modulů, které budou aktivovány dle požadavků zákazníka.

Aplikace bude umožňovat tyto funkcionality:

1. Registrace uživatelů a jejich přístup do systému podle práv a modulů.
2. Správa kapacity zařízení s grafickým zobrazením.
3. Tvorba objednávek sortimentu poskytovaného zařízením.
4. Správa skladových zásob sortimentu.
5. Výpočet aktuálního stavu skladu podle objednávek pro inventurní účely.
6. Statistické vyhodnocení aktuálně nejžádanějšího sortimentu.
7. Sledování nákladů a výnosů na základě cen zboží a zvolené marže.
8. Prognostická kontrola stavu skladů na základě předchozího odbytu sortimentu, která bude dynamicky upravována.
9. Zasílání objednávek vybraného zboží dodavatelům ve volitelných datových formátech.

V úvodní části je nezbytné provést rešerši stávajících systémů, které se zabývají touto problematikou. Rešerši je nutné doplnit o porovnání s nově navrhovaným systémem, který bude předmětem této práce.

Systém bude fungovat na základě samostatných modulů, které bude možné aktivovat libovolně na základě potřeb konkrétního zařízení. Bude se jednat o univerzální systém.

V práci bude provedena důkladná analýza procesů, které budou znázorněny v procesním BPMN diagramu. Diagram bude navržen s cílem dosažení maximální stability a bezpečnosti. Práce bude obsahovat analýzu navrhovaného řešení, popis použitých technologií, návrh databáze a aplikačního řešení. V rámci návrhu databáze bude vytvořen entity-relationship model. Pro vytvoření aplikace bude využit skriptovací jazyk PHP doplněný o jvasriptovou knihovnu jQeury a databáze Oracle, nebo MySQL.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. ŘEPA, Václav. Analýza a návrh informačních systémů. Praha: Ekopress, 1999.
2. PATTINSON, Fiona. Certifying Information Security Management Systems. CISSP, CSDP, Atsec information security corporation. Dostupný z WWW: <http://www.atsec.com>.
3. LACKO, Luboslav. Oracle. Správa, programování a použití databázového systému. Brno: Computer Press, 2007.
4. GROFF, James R. a WEINBERG, Paul N. SQL kompletní průvodce. Brno: Computer Press, 2005.
5. VRÁNA, Jakub. 1001 tipů a triků pro PHP. Brno: Computer Press a.s., 2010.
6. BRYLA, Bob; LONEY, Kevin. Mistrovství v Oracle Database 11g. Jiří Huf. Vydání první. Brno: Computer Press, a.s., 2009.
7. NOWICKI, Steven D. a LECKY-THOMSON, Ed. PHP 6. Programujeme profesionálně. Brno: Computer Press, 2009.
8. DRUSKA, P. CSS a XHTML - tvorba dokonalých webových stránek krok za krokem, Grada 2006.
9. LONEY, K., THERIAULT, M. Mistrovství v Oracle. Praha, Computer Press, 2002.
10. ORACLE - <http://www.oracle.com>

Vedoucí diplomové práce:

Ing. Miloslav Macháček, Ph.D.

Katedra informačních technologií

Datum zadání diplomové práce:

31. října 2013

Termín odevzdání diplomové práce:


16. května 2014



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 16. 04. 2014

Bc. Daniel Poláček

Poděkování

Děkuji vedoucímu diplomové práce panu Ing. Miloslavu Macháčkovi Ph.D. za čas a úsilí, které věnoval mé práci a za jeho odborné rady při vypracování této diplomové práce.

Anotace

Cílem této práce bylo vytvoření univerzálního restauračního systému, který bude možné využít pro podporu provozu libovolného stravovacího zařízení. Po pečlivém prozkoumání situace na trhu jsem začal tvořit vlastní pojetí restauračního systému, na základě získaných informací o provozu různých podniků se zaměřením na gastronomii. Při tvorbě jsem využíval technologie, které umožňují provoz na libovolném počítači, a díky tomu není třeba pořizovat drahá zařízení. Aplikace je postavená na PHP a HTML kódu a databázi MySQL a její tvorba probíhala s využitím OOP programování a návrhu pomocí UML a BPMN diagramů. Celý systém je rozdělený na jednotlivé moduly, které je možné spravovat samostatně a aktivovat je podle potřeby konkrétního podniku.

Klíčová slova

Restaurační systém, správa restaurací, správa skladových zásob, nabídka, objednávky, moduly

Title

A universal system for catering management

Annotation

The aim of this thesis was to create a universal computer system for restaurants, which would be used to support the services of any catering institution. After a thorough examination of the market situation, I began to create my own concept of a catering system, based on the information gained about the management of various companies focused on gastronomy. For creating the system, technologies that allow the usage of any computer were used, and thus there is no need for expensive equipment. The application is based on PHP and HTML code and MySQL database and its formation were carried out using the OOP programming and design using UML and BPMN diagrams. The whole system is divided into individual modules that can be managed separately and activated according to the needs of a particular business.

Keywords

catering system, restaurant management, inventory management, offer, orders, modules

Obsah

Seznam zkratk	8
Seznam obrázků	9
Seznam tabulek	9
Úvod	10
1 Teoretická část	12
1.1 Historický vývoj	12
1.1.1 Vývoj pohostinství.....	12
1.1.2 Vývoj restauračních systémů.....	13
1.1.3 Moderní restaurační systémy.....	14
1.2 Kategorizace systémů dnešní doby.....	15
1.3 Obecná problematika stravovacích zařízení	16
1.3.1 Kategorizace stravovacích zařízení	16
1.3.2 Prostory stravovacích zařízení.....	18
1.3.3 Lidé a obsluha systému	19
1.4 Obecné požadavky na systém.....	22
1.4.1 Ochrana dat a rychlost systému	22
1.4.2 Oblasti, které mohou být automatizované systémem	23
1.5 Řešení restauračních systémů	27
1.5.1 Agnis.....	27
1.5.2 AWIS GASTRO	29
1.5.3 Conto	30
1.5.4 MRP – vizuální účetní systém.....	32
1.5.5 KelEXPRESS	33
1.5.6 Menu55.....	35
1.5.7 Klasické pokladny	36
1.5.8 Souhrnné hodnocení	36
1.6 Porovnání stávajících řešení s mým systémem	38
1.6.1 Systém Agnis.....	38
1.6.2 Systém AWIS	39
1.6.3 Systém Conto.....	40
1.6.4 Systém MRP.....	40

1.6.5	Systém KelEXPRESS	41
1.6.6	Menu55	41
2	Praktická část.....	42
2.1	Cílová skupina vytvářeného systému	42
2.2	Charakteristika aplikace	42
2.3	Proces vývoje.....	43
2.3.1	Požadavky na funkce systému	44
2.3.2	Use Case Diagram	45
2.4	Rozvržení systému.....	47
2.5	ER diagram	49
2.5.1	Databáze bloku uživatelé.....	50
2.5.2	Databáze bloku objednávky.....	51
2.5.3	Databáze bloku nabídky	53
2.5.4	Databáze bloku skladu.....	54
2.5.5	Databáze bloku dodavatelů.....	56
2.6	Složky a soubory	56
2.7	Použité technologie	58
2.7.1	Návrhové aplikace	58
2.7.2	Data a databáze	59
2.7.3	Výroba provoz a testování	59
2.7.4	Programovací jazyky	59
2.7.5	Knihovny a rozšíření	60
2.8	Představení vytvořeného systému	60
2.8.1	Vstupní logika aplikace a základní obsluha příkazů	60
2.8.2	Společné rysy zdrojových kódů.....	62
2.8.3	Obecná charakteristika struktury modulů.....	64
2.8.4	Přihlašování a správa přihlášených uživatelů	65
2.8.5	Správa uživatelů	67
2.8.6	Správa dodavatelů.....	68
2.8.7	Správa skladu.....	69
2.8.8	Správa nabídky	71
2.8.9	Správa rozložení restaurace	71
2.8.10	Správa tvorby objednávek	72

2.9 Požadavky na běh systému	73
2.9.1 HW požadavky na systém	73
2.9.2 SW požadavky	73
2.9.3 Požadavky na server a připojení	74
2.10 Uživatelská dokumentace systému RESYS	74
2.10.1 Přihlašování do systému	74
2.10.2 Práce s uživateli	75
2.10.3 Správa dodavatelů	75
2.10.4 Správa skladu	76
2.10.5 Správa jednotek	77
2.10.6 Tvorba nabídky	78
2.10.7 Rozložení stolů	78
2.10.8 Tvorba objednávek	79
2.11 Instalační příručka	79
2.11.1 Webhosting a základní konfigurace	80
2.11.2 Spuštění instalátoru	81
Závěr	82
Literatura	83
Příloha A – Přehled funkčních a nefunkčních požadavků aplikace	85
Příloha B – ER diagram	86
Příloha C – Use Case Diagram	87
Příloha D – Část metody pro vložení doplňujících polí nabídky	88

Seznam zkratek

CD	Compact Disc
COM	Component Object Model
CSS	Cascading Style Sheets
CSV	Comma-separated values
EAN	European Article Number
GHz	Gigahertz
HTML	HyperText Markup Language
MVC	Model–view–controller
LAN	Local Area Network
ODBC	Open Database Connenctivity
PHP	Personal Home Page
POS	Point of Sale
UPOS	UnifiedPOS
UML	Unified Modeling Language
XML	Extensible Markup Language

Seznam obrázků

Obrázek 1 – Kompletní hierarchie personálu velké restaurace	21
Obrázek 2 – Nejčastější obsazení personálu stravovacích zařízení.....	22
Obrázek 3 – Přehled základních procesů přihlašování	23
Obrázek 4 – Přehled základních procesů správy skladu.....	24
Obrázek 5 – Přehled základních procesů inventarizace	24
Obrázek 6 – Přehled základní procesů nabídky.....	25
Obrázek 7 – Přehled základní procesů rozložení prostoru restaurace	26
Obrázek 8 – Přehled základních procesů objednávek	26
Obrázek 9 – Přehled prioritních požadavků systému	45
Obrázek 10 – Ukázka rozšířeného Use Case diagramu z pohledu vedoucího na sklad	47
Obrázek 11 – Schéma rozvržení aplikace na bloky a moduly.....	49
Obrázek 12 – ER diagram bloku uživatelé.....	51
Obrázek 13 – ER diagram bloku objednávky.....	53
Obrázek 14 – ER diagram bloku nabídky	54
Obrázek 15 – ER diagram bloku skladu.....	55
Obrázek 16 – ER diagram bloku dodavatelů.....	56

Seznam tabulek

Tabulka 1 – Hodnocení testovaných systémů.	37
---	----

Úvod

Ač se to na první pohled nemusí zdát, tak provoz stravovacích zařízení je v dnešní době velice složitý. Většina lidí má o fungování restaurací, barů, kaváren, bufetů a dalších podobných podniků jen rámcovou představu, obsahující nákup zboží, které je pak v různých formách prodávané zákazníkům. Ve skutečnosti je však za podnikáním v tomto oboru skryto mnohem více. Se zvládnutím všech potřebných náležitostí a chodu zařízení pak provozovateli může pomoci restaurační systém. Každý provoz obsahuje několik základních procesů, které jsou dále větvené na podprocesy.

Jedním z procesů provozu stravovacích zařízení je evidence skladových zásob, která musí být přesná, aby provozovatel věděl, jestli má dostatek surovin na pokrytí poptávky. Stejně tak nadbytek nějakého zboží je nežádoucí, protože v něm jsou vázané finanční prostředky. Pro provoz je tedy třeba evidovat zásoby zboží a znát optimální množství každé suroviny.

Po zabezpečení chodu skladu přichází na řadu sestavení nabídky, která musí být cenově vyvážená a samozřejmě atraktivní pro zákazníky. Sestavit dobrou nabídku vyžaduje značnou dávku kreativity a dobrou znalost provozu a poptávky.

Po zpracování problematiky skladu a sestavení nabídky zařízení přichází na řadu další ze základních procesů a to samotný prodej zákazníkům. S prodejem jsou úzce spojené objednávky, které je třeba evidovat a vyhodnocovat. Díky získaným informacím je možné sestavit přehled o prosperitě podniku, žádanosti nabídky a kontrole zaměstnanců, která je v tomto oboru nezbytnou nutností.

Procesů, které jen nutné hlídat, je samozřejmě mnohem více. Když přihlédneme také k všude přítomné konkurenci a nutnosti neustálého sledování nových trendů, zjistíme, že práce kolem restaurace je více, než je mnohdy reálně zvládnout. Při každém provozu stravovacího zařízení se pak velice brzy ukáže, že je nutné práci automatizovat a tím ušetřit čas a peníze. S touto automatizací může provozovateli pomoci restaurační systém.

Pod restauračním systémem si však není možné představit aplikaci, která je striktně určena pro restaurace, ale obecný systém pro stravovací zařízení, který by měl automatizovat provoz restaurací, kaváren, cukráren a dalších podniků spadajících do stravovacích zařízení. I když má každý z podniků trochu jiné požadavky na provoz, jsou zde společné procesy, jež mohou být univerzálně pokryty jedním systémem, který nemusí být vždy chápán jako aplikace, ale může se jednat například o bonovací pokladnu, nebo třeba jiný hardware se specializovaným softwarem.

Všechny systémy, které se v této oblasti vyskytují, nejsou úplně ideální. Některé nepokrývají všechny potřebné oblasti, jiné nejsou uživatelské a další jsou prostě jen finančně nedostupné. Stávající situace vedla k vytvoření této práce, která má za úkol pojmut problematiku stravovacích zařízení a představit takovou aplikaci, která by eliminovala všechny nedostatky aktuálně dostupných systémů.

Teoretická část této práce je zaměřená na vývoj stravovacích zařízení, jejich kategorizaci, rozdělení personálu zařízení, který může se systémem pracovat a obecné požadavky na systém spravující provoz stravovacího zařízení. Po vysvětlení všech potřebných náležitostí jsem se zaměřil na několik stávajících systémů, které jsou běžně používány, a provedl jsem jejich zhodnocení, na základě kterého jsem sestavil porovnání s mým vytvořeným systémem.

V praktické části je představen návrh systému, využívané technologie při jeho tvorbě a samotná tvorba aplikace pro stravovací zařízení. V této části jsou konkrétní informace o fungování aplikace, ukázky návrhu procesů, přehled databáze a samotného kódu. Vše pak doplňuji uživatelskou dokumentací, bez které by systém nebyl kompletní a návodem jak aplikaci snadno instalovat.

1 Teoretická část

1.1 Historický vývoj

I přesto, že zhodnocení historického vývoje stravovacích zařízení není hlavním smyslem této práce, tak je dobré provést alespoň základní přehled historických souvislostí, které předcházely vzniku těchto zařízení ve formě, kterou dnes známe. Může být až překvapivé, jak dlouhou cestu již lidstvo ušlo v oblastech pohostinství a jakým způsobem se vyvíjely požadavky zákazníků a tím i samotné provozovny. Vývoj pak sebou přinesl i postupnou nutnost evidence údajů o chodu zařízení a tím vznikl i základ samotných restauračních systémů.

1.1.1 Vývoj pohostinství

Pohostinství má hluboké kořeny v daleké historii. Již od dávných dob existovala místa, kde se lidé scházeli za účelem nocování a snadného stravování. Zmínky o pohostinství nalézáme už ve starověku a to z oblastí Palestiny a Babylónu, kde vznikala první zařízení, určená hlavně pro poutníky na cestách, kteří zde nalézali bezpečné útočiště (PROSECKÝ, 1999)¹. V této době se však o vývoji restauračních systémů dá jen těžko hovořit.

Výrazný rozmach pohostinství zaznamenalo v době rozkvětu řecké a římské kultury, kdy začala vznikat první soukromá zařízení, kde bylo možné strávit čas, a to až za účelem nocování na cestách nebo za zábavou. Dochází zde však ke změně v samotném využívání těchto zařízení. I když jsou tyto podniky stále určené pro lidi na cestách, tak začínají být stále častěji vyhledávány i čistě za účelem zábavy (KMENT, 2011). Z pohledu restauračních systémů zde však změna není.

V době středověku byly pro pohostinství typické venkovské hostince, kde se již začala značně odlišovat kvalita služeb a provoz takovýchto podniků měl i svá pravidla stanovovaná vládnoucí autoritou (ČURDA, 2004). Právě tato doba mohla vést k nutnosti většího přehledu nad příjmy, výdaji a celkovým chodem podniku. O automatizaci a restauračních systémech zde stále nemůže být řeč, ale zvyšující se nároky na provoz dávají jistý potenciál k počátku jejich vývoje. Stále je zde však technická bariéra, která nedovoluje samotný vývoj strojů, které by provoz ulehčily.

Největší rozkvět pohostinství zaznamenalo v novověku. Nároky klientů se postupně zvyšují a pro samotný běh podniku začalo být nezbytnou nutností přesně evidovat informace o chodu zařízení. V této době se také objevují technologie, které umožňují sestavit první restaurační zařízení v podobě mechanických pokladen. Ty se postupně zdokonalovaly až do systémů, které známe dnes.

¹ První zmínky o hostincích této doby jsou z Chammurapiho zákoníku.

V dnešní době již je provoz stravovacích zařízení “vědním oborem”, který v sobě zahrnuje pečlivou přípravu nabídky, propracované marketingové kampaně obsahující reklamu, logotypy, celkové sladění od vzhledu restaurace až po drobnosti jako loga na talířích a příborech a mnohé další. Mimo to musí být provoz podpořen precizní organizací a přehledem nad poptávkou a nabídkou konkurenčních zařízení. Podniky, které v dnešní době některý z aspektů provozu zanedbají, začnou postupně ztrácet zákazníky, jejich provoz skomírá a dříve nebo později zaniknou.

Z pohledu provozu stravovacího zařízení jsou restaurační systémy nedílnou a velice podstatnou součástí každého úspěšného podniku, protože majiteli či provoznímu pomáhají s pokrytím celé problematiky tohoto oboru a šetří čas a finanční prostředky, což jsou jistě dva důležité aspekty podnikání v dnešní době.

1.1.2 Vývoj restauračních systémů

Určit přesně počátek vývoje restauračních systémů není možné bez definice toho, co si pod restauračním systémem představujeme. Pokud bychom se zaměřili na samotnou evidenci bez ohledu na techniku, tak ta bude téměř stejně stará jako samotné provozování stravovacích zařízení.

Pojem restaurační systém je však právem spojován s technikou a to hlavně s automatizací práce a možností kontroly s minimalizací rizika lidského selhání. Pokud se na problematiku podívám z této strany, tak za původ podobných systémů lze považovat první primitivní mechanické pokladny. Historicky první pokladna začala vznikat v roce 1878 a je až zarážející, jaký mechanismus stál u jejího zrodu.

První myšlenka na způsob fungování pokladny vznikla na obyčejném parníku. Její tvůrce jménem James Ritty si v roce 1878 všiml mechanismu, který na parníku počítal otáčky lodní vrtule a zjistil, že by podobným způsobem šlo počítat prodané zboží a získané peníze. Společně se svým bratrem pak o rok později sestavil první pokladnu, která se skládala z hodin, ukazujících částku a v dolní části bylo několik pák na potvrzení částky.

Stejně zajímavý je i důvod vzniku první pokladny. Ritty byl majitelem úspěšného baru v Daytonu v Ohaiu. Stejně jako ostatní provozovatelé barů se i on potýkal s nepoctivostí svých zaměstnanců a potřeboval nástroj, který by mu pomohl hlídat počty prodaného zboží a částku, kterou má od svých zaměstnanců obdržet (Massachusetts Institute of Technology, 2002). Trend nepoctivosti zaměstnanců se udržel až do dnešní doby a restaurační systémy tady do jisté míry stále plní i své úplně nejstarší poslání.

Jak už to však v historii bývá, tak nic nejde tak lehce, jak se na první pohled může zdát. První prototyp pokladny byl značně nepřesný a ani druhý se moc nepovedl. Úspěšnější a funkční se ukázal až třetí pokus. Ritty však neměl s prodejem pokladen úspěch a nakonec svůj patent prodal. Dalších pět let vývoj stagnoval až do doby, než se k patentu dostal John H. Patterson, který založil firmu National Cash Register Company a začal pokladny šířit na trh. Tím byl odstartován rychlý vývoj pokladen a to ne jen v oboru stravovacích zařízení, ale například i v provozu běžných obchodů.

Pokladny pak byly postupně rozšířeny o papírové pásky, do kterých byly raženy tečky podle toho, jaká částka měla být uhrazena. Následně byl na přední stranu doplněn číselník, který ukazoval zákazníkovi cenu, a v roce 1906 došlo k doplnění pokladen o elektromotor (Almyta systems, 2014). Následovala série dalších úprav, které však pokladny pouze vylepšovaly, ale výrazněji je nezměnily. Hlavní důvodem byla pravděpodobně nedostačující technická vyspělost pro další vývoj.

To se však změnilo kolem roku 1970, kdy se již projevoval technologický pokrok i ve všech oblastech běžného života. Za vývojem pokladem v této době nestál nikdo jiný, než samotná firma IBM, která v roce 1973 představila své produkty IBM 3650 a 3660, které byly základem pro první moderní pokladny. Tato zařízení spolu dokázala komunikovat pomocí sítě LAN (POSmatic, 2014), což byl v této době obrovský krok vpřed. Stále se však jednalo o stroje, které neměly možnost grafického rozhraní a fungovaly pouze na principu předdefinovaných tlačítek.

Následovala série dalších vylepšení a rozšíření pro pokladní systémy. Z pohledu dnešní doby se však jako nejvýznamnější jeví rok 1985, kdy došlo k prvnímu propojení registrační pokladny s dotykovým monitorem a díky tomu se značně ulehčila práce se systémem. V tuto dobu se však většinou využívala technologie tenkých klientů².

1.1.3 Moderní restaurační systémy

Postupný vývoj pak pokladní systémy přiblížil až k těm, které se používají v dnešní době. Moderní systémy jsou bezdrátové, je možné na nich pracovat s kreditními kartami a jejich ovládání probíhá pomocí dotykových barevných monitorů. Díky postupnému vývoji se však objevila i nutnost standardizace komunikace jednotlivých pokladních systémů (ID Innovations Incorporated, 2007).

Stejně jako v mnoha jiných odvětvích, tak i zde nejprve vznikla iniciativa na sjednocení, která byla nazvána UnifiedPOS a zkráceně označována UPOS, kde POS označuje zkratku anglického Point of Sale, tedy místo prodeje. Vzniklá iniciativa však plně neřešila konkrétní spojení, ale hlavně nutnost standardizace. Samotnou standardizaci komunikace pak zabezpečují OPOS a JAVAPOS.

OPOS byl první přijatou normou oblasti komunikace pokladních systémů, kterou začaly využívat společnosti Microsoft, NCR Corporation, Epson, Fujitsu a další. OPOS umožňoval normalizovat komunikace pomocí COM portů za použití programovacích jazyků pro Microsoft. První vydání tohoto standardu bylo nabízeno od roku 1996 v mnoha pokladnách.

JAVAPOS byl vyvíjen společností Sun Microsystems a NCR Corporation od roku 1997. Jeho první verze však byla vydána až v roce 1999. Jak již název napovídá, tak tento standard byl určen pro jazyk Java a díky tomu je do značné míry multiplatformní.

² Tenký klient je silnou vazbou mezi počítačem a programem. Pro běh těchto systémů je tedy třeba speciální hardware.

Zmiňované standardy se v dnešní době stále využívají a již mají nesčetné množství aktualizací. Stejně tak samotné systémy již dostaly velmi mnoho různých tváří a jejich popis by mohl vydat na samostatnou publikaci. Některé ze systémů jsou představeny v dalších částech této práce.

1.2 Kategorizace systémů dnešní doby

V dnešní době je obecně velký trend ukládání a zpracování dat za použití online řešení. Oblast stravovacích zařízení je však v tomto směru poněkud opožděná i přes to, že samotný vývoj byl směřován na síťovou komunikaci mnoho let.

Možná, že právě díky mnoholetému vývoji se restaurační systémy příliš specializovaly a myslím si, že v nich chybí určitá míra otevřenosti a možnosti ke změnám podle potřeb zákazníka. Pro představu o dnešních nabízených systémech je kategorizují do několika skupin.

- “Klasické” registrační pokladny - starší avšak stále používané pokladny, které se většinou prezentují jako jedno zařízení s malým displejem a zabudovanou klávesnicí, kde jsou definované položky z nabídky. Možnosti těchto systémů jsou velice omezené, šance na rozšíření podle přání zákazníka nulová a samotné nastavování je práce na několik desítek hodin. Osobně jsem se setkal s několika systémy tohoto typu a jen samotné spojení s počítačem se ukázalo jako problematické, protože vzájemná komunikace pokladny a dnešní počítačů vyžaduje složité nastavení.
- Restauracní pokladny - modernější systémy, které jsou většinou složené z řídicí jednotky, což bývá počítač, dotykového monitoru, tiskárny na účtenky a myši s klávesnicí. Jedná se o vysoce výkonná zařízení, která pracují s předpřipravenou aplikací. Tuto aplikaci je již možné i rozšířit. Jistým omezením těchto systémů je však pořizovací cena, která se pohybuje v řádech desítek tisíc korun. Mimo to se běžně standardní obslužný program nerozšiřuje, protože jeho úprava bývá nákladná.
- Mobilní restaurační systémy - nejčastěji se jedná o rozšíření restauračních pokladen, které funguje na principu bezdrátových mobilních číšníků, což jsou malá zařízení, která obsluha nosí u sebe a provádí objednávky přímo od stolu. Jedná se o kvalitní, rychlé a spolehlivé rozšíření, které má jistě své velké uplatnění ve větších restauracích. Pro menší zařízení je však tento systém až nadbytečný a finančně nedostupný.
- Restauracní software - běžné aplikace, které se dají instalovat na počítače v restauračních zařízeních. Na trhu jsou většinou verze, které nepokrývají všechny oblasti stravovacích zařízení. Jednoznačnou výhodou je možnost instalace na jakýkoliv počítač. Kvalita programů je však nízká a je zde riziko ztráty dat bez provádění pravidelných záloh.

- Online systémy - online systémů je v dnešní době k dispozici více, ale jen málo z nich je dostupných na českém trhu. Většinou se jedná o aplikace s online přístupem, který umožňuje náhled do systému pomocí internetu, ale systémy nejsou přímo provozované na serveru. Je zde tedy stále nutnost instalace na konkrétní zařízení a online přístup je jen realizován formou vzdálené plochy. Systémů, které fungují výhradně jako webová aplikace přes prohlížeč, je však málo. Vzhledem k tomu, že se však jedná o trend poslední doby, je velká pravděpodobnost, že se tyto systémy budou dále rozvíjet. Jasnou výhodou je provoz na libovolném počítači s připojením k internetu a nízká pořizovací cena. Slabinou je pak napojení na tiskárny účtenek, které pro tento typ systémů nejsou nejvhodněji uzpůsobené³.
- Responzivní řešení - moderní doba sebou přinesla možnost správy stravovacích zařízení přes mobilní technologie, jako jsou chytré telefony a tablety. Většina dostupných systémů je však spíše zaměřena na ovládání ze strany zákazníků restaurace. Zákazník si může objednat, prohlédnout nabídku, zavolat číšníka a využívat další podobné funkce. Veškeré operace zákazníka se pak zaznamenávají do systému a je s nimi možné dále pracovat. Jedná se o zajímavé řešení, ale nevýhodou je zapojení dalších uživatelů do systému⁴.

1.3 Obecná problematika stravovacích zařízení

Každé stravovací zařízení je trochu jiné a má mírně odlišné požadavky na svůj provoz. Pro účely mé další práce uvádím kategorizaci stravovacích zařízení s vysvětlením aspektů, které do provozu vstupují. Díky tomu pak bude snazší pochopit problematiku jako celek a uvědomit si jak velkou roli v dnešní době zastávají stravovací systémy.

1.3.1 Kategorizace stravovacích zařízení

Rozčlenit stravovací zařízení lze mnoha způsoby a do různých kategorií. Jedno ze členění je uvedeno například v publikaci Služby společného stravování (MLEJNKOVÁ, 2005) a je zaměřeno na typologii zařízení. Mimo dalších členění je zde například:

- Členění podle právní formy, které rozděluje podniky na provozované fyzickou osobou a právnickou osobou.
- Členění podle formy stravování na veřejné stravování a provozovny účelového stravování, jako jsou závodní jídelny, školní jídelny a podobně.
- Členění podle formy prodeje na s obsluhou a bez obsluhy
- Členění podle lokalizace na venkovské, městské, lázeňské a horské.
- Členění podle počtu zaměstnanců, kde je však uvedeno jako první rozmezí malé o 0 až 19 zaměstnancích. Což samo o sobě není příliš použitelné, protože podnik s 19 zaměstnanci je již podle mého názoru podnik poměrně velký.

³ Tiskárny většinou komunikují na standardu OPOS, nebo JAVAPOS. Ani jeden z těchto standardů není plně propojitelný s programovacími jazyky pro webové aplikace.

⁴ Každý další přístup do systému klade vyšší nároky na zabezpečení.

Samozřejmě, že podobných členění je uváděno více, ale ta nejsou pro potřeby mé práce nijak klíčová. Z výše uvedených členění je z pohledu restauračních systémů použitelné rozdělení podle formy prodeje, počtu zaměstnanců a formy stravování. Naproti tomu například právní forma podniku je pro vytvoření a použití restauračního systému téměř bezvýznamná.

Mnohem více však pro potřeby mé práce vyhovuje členění podle doporučení upravující základní ukazatele pro kategorizaci hostinských a ubytovacích zařízení (Český statistický úřad, 1994). Jedná se sice o starší členění, ale v dnešní době stále aktuální a v rámci Evropské unie používané. Toto členění dělí stravovací zařízení do několika kategorií a podkategorií a to následovně:

1. Restaurace - hlavním zaměřením této kategorie je prodej pokrmů s možností zakoupení nápojů. S následujícím rozdělením do podkategorií:
 - 1.1. Restaurace - hostinské zařízení, které se specializuje na široký sortiment pokrmů, jež jsou roznášeny personálem. Z pohledu systému tak můžeme očekávat větší požadavky na sestavování nabídky, více skladových položek a větší počet uživatelů.
 - 1.2. Pohostinství - rozšířený typ restaurace nabízející i služby jako je ubytování. Z pohledu obslužné aplikace by bylo výhodnější, aby restaurační systém byl pouze součástí větší aplikace, která by pokrývala také všechny potřeby provozu doplňkových služeb.
 - 1.3. Jídelní restaurační vozy a jiná zařízení pro přepravu osob - poskytování stravování ve veřejných dopravních prostředcích. Zde jsou požadavky na systém specifické v tom, že se jedná o proměnné prostředí, kde není vždy plně možné zaručit všechny požadavky pro chod systému, jako je například připojení k internetu, nebo dostatečný výkon elektrického zdroje. V této kategorii bych online systémy chápal jako nevhodné.
 - 1.4. Motoresty - stravovací zařízení s větší parkovací kapacitou umístěné poblíž větších dopravních komunikací. Z pohledu obslužného systému jsou zde stejné požadavky jako u restaurací.
 - 1.5. Samoobslužné restaurace - zařízení poskytující stravování samoobslužným způsobem. V tomto případě jsou na systém kladeny větší nároky, pokud jde o sestavování nabídky, protože nevíme, jaké suroviny si zákazník vybere. Je tedy nutné, aby se nabídka skládala z více samostatných položek než u restaurací.
 - 1.6. Bufety - samoobslužné stravovací zařízení zajišťující občerstvení a doplňkové stravovací služby. Mohou být dále členěny podle specializace⁵. Takovéto zařízení by mohlo využívat systém, který bude odpovídat požadavkům samoobslužných restaurací.

⁵ Může se jednat o mléčné bufety, rybí bufety a tak dále.

- 1.7. Občerstvení a kiosky - stravovací zařízení zabezpečující poskytování občerstvení. Velice často se jedná o sezonní provozovny nebo jako doplněk stávající provozovny formou výdejního okna. Z pohledu systému se může jednat o kombinaci jídelních restauračních vozů a bufetů. Online systém by v tomto případě opět nebyl nejvhodnější.
2. Bary - jsou specializované na prodej nápojů s jídlem jako doplňujícím sortimentem.
 - 2.1. Denní bar - stravovací zařízení, které má většinou jako dominantní prvek barový pult. Poskytuje občerstvení obslužným způsobem. Denní bary je možné dále dělit podle svého zaměření třeba na gril bar, pizzerie, aperitiv bar a další. Všechny tyto kategorie však mají stejné požadavky na systém. Obecně je u denních barů specifičtější, ale méně obsáhlá nabídka.
 - 2.2. Noční bar - zábavná hostinská zařízení s obsluhou, která servírují jídla i nápoje. Tyto podniky často obsahují taneční parkety a větší barové pulty. Z pohledu systému pak bude kladen velký nárok na rychlost a stabilitu systému při vysokém zatížení. Oproti tomu nebude podstatné rozložení restaurace, protože tyto podniky nemívají objednávky vázané na konkrétní místa.
 - 2.3. Vinárna - zařízení specializovaná na vína a doplňkový sortiment. Systém potřebný pro provoz tohoto zařízení bude mít podobné nároky jako pro restaurace, ale nabídka bude omezenější a jistě by se dalo uvažovat o rozšíření v podobě specifické funkčnosti systému, která pokryje zaměření této kategorie provozoven⁶.
 - 2.4. Kavárna - obslužné hostinské zařízení se zaměřením na prodej teplých nápojů, cukrářských výrobků a studených pokrmů. Bývají často zařízení ve specifickém stylu a zaměřené na určité odvětví. Systém by měl být typově podobný jako v případě vinárny.
 - 2.5. Hostinec - stravovací zařízení specializované na podávání piva a jídel, které pivo vhodně doplňují. Systém využívaný v hostincích by mohl být podobný jako v případě restaurací. Oproti restauracím by bylo vhodné aplikaci rozšířit o část, která se zaměří na nabídku pív.

1.3.2 Prostory stravovacích zařízení

Každá restaurace má originální uspořádání. Pro potřeby restauračního systému specifikují nejběžnější části restauračních prostor. Každý z prostorů má pak vliv na fungování a nastavení systému.

- Kuchyně - místo přípravy jídla. U rozsáhlejších systémů bývá jedna systémová jednotka v kuchyni, aby kuchař hned věděl, jaké jídlo má začít chystat. Někdy bývá využíván monitor, ale častěji se upotřebí tiskárna, která při objednávce jídla vytiskne lístek s informací, co má být připraveno. Většina restaurací však využívá systém napsaného lístku od číšníka, který je donesen do kuchyně. Důvody jsou hned dva, jedním z nich je skutečnost, že pro většinu podniků je koupě dalšího zařízení do kuchyně příliš nákladná a druhým důvodem je prostředí kuchyně, kdy v

⁶ Například specifický modul pro vinný lístek.

kuchyni není dostatek místa a také zde při přípravě jídla dochází prostřednictvím páry, oleje a dalších surovin k opotřebení a znehodnocení techniky.

- Lokál - prostor restaurace, kde jsou obsluhováni hosté. Je to i část provozovny, kde je umístěn počítač, na kterém běží restaurační systém a kde s ním pracuje obsluha restaurace. Z pohledu aplikace by mělo být možné rozlišit prostor lokálu podle stolů, aby byla práce přehledná.
- Bar - prostor v lokálu, který nemusí být ve všech stravovacích zařízeních, ale je velice častý. Z pohledu systému může být chápán jako jeden stůl, různé stoly, které v tomto pohledu rozložení reprezentují židle, nebo je bar veden jako prostor bez místa.
- Zahrádka - venkovní prostor zařízení, který nemusí být obsluhován. Z pohledu systému může být evidován jako oddělená část restaurace (samostatná záložka, nebo jinak oddělené rozdělení restaurace založené na stejném principu jako lokál), jako součást restaurace (stoly jsou umístěné ve stejné části jako lokál, ale odlišené třeba názvem), nebo jako prostor bez místa.
- Ostatní prostory - výdejní okénka, pivní stánky u restaurací, ledové bary. Všeobecně se jedná o prostory s nepravidelným provozem. Z pohledu aplikace mohou být ostatní prostory v systému zohledněné, ale jedná se o nestandardní řešení. Většinou jsou tyto prostory vedeny mimo systém restaurace a zadává se pouze hromadný přehled nebo je situace řešena přidáním jednoho stolu, na který je neustále přiřazována jedna objednávka po druhé s okamžitým uhrazením.

1.3.3 Lidé a obsluha systému

Jako v každém oboru jsou i v pohostinství a provozu stravovacích zařízení základním stavebním kamenem lidé. Setkáváme se zde s obsluhou systému, která má na aplikaci určité požadavky a se zákazníky, kteří se systémem nemusí přímo pracovat a mnohdy ani netuší, co zajišťuje.

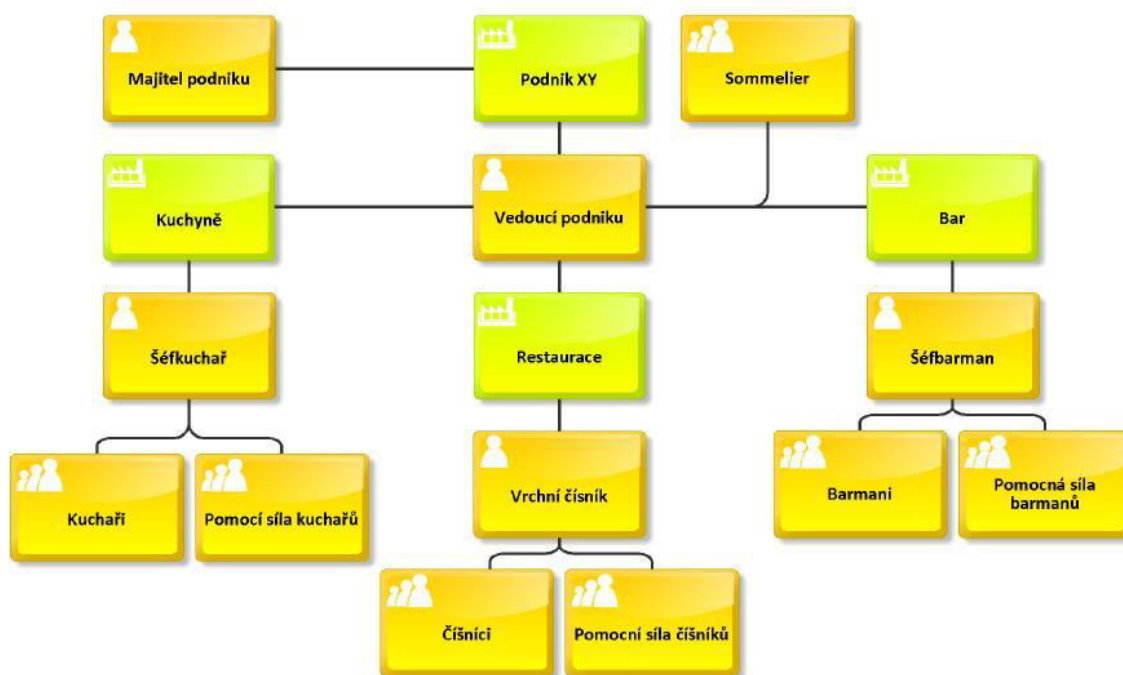
V případě, že se podíváme na systém z pohledu zákazníka, tak zde nejsou žádné nároky. Je však několik odvětví, které může restaurační systém zákazníkovi zprostředkovat. Může se jednat například o tyto služby.

- Rezervační systém míst - restaurační systém může mít moduly, které se starají o rezervaci míst. Z pohledu zákazníka je systém přímo použitelný pouze v případě jeho dostupnosti online.
- Zveřejňování jídelníčku - opět se jedná o rozšiřující funkci, která může být v systému dostupná. Může například docházet k exportu jídelníčku na stránky, kde je zobrazován, nebo k jeho distribuci do e-mailových schránek klientů.
- Novinky e-mailem - z pohledu služeb pro zákazníka systém může rozesílat novinky o akcích na nabídku, nebo třeba chystaný program.

Mnohem obsáhlejší jsou však požadavky, které má na systém samotná obsluha. V obecném měřítku se může například jednat o požadavky na rychlost, snadné ovládání a další. Systém musí pokrývat celou problematiku stravovacích zařízení. Aby bylo možné lépe chápat požadavky na systém, tak bude vhodné definovat uživatelské role, které se systémem pracují. Jednotlivé role se však liší podle velikosti a kategorie zařízení. Následující přehled se týká především větších zařízení restauračního typu (SOUČKOVÁ, 2009).

- Majitel podniku - majitel je osoba rozhodující o financích a stanovující směr vývoje zařízení. Měl by mít dobrý přehled o fungování podniku a provádět kontrolu kvality služeb. S restauračním systémem pracuje na úrovni finančních a statistických přehledů.
- Manažer podniku - přímý podřízený majitele, který má plnit funkci kontroly zaměstnanců a běhu provozu. Jedná se prakticky o provozního ředitele, který je přímo zodpovědný majiteli a měl by zajišťovat běh celého podniku. Systém využívá pro účely zásobování zařízení a sledování celkového chodu.
- Vrchní číšník - stará se o restaurační část a personál s tím spojený. Vrchní číšník se stará o rozplánování směn ostatních číšníků a je odpovědný za chod restaurace. Sám o sobě bývá i pracovní silou jako číšník. Systém využívá pro plánování rozpisu zaměstnanců a pro práci číšníka.
- Číšník - stará se o obsluhu zákazníků v restauračních částech provozovny. Společně s barmany pracuje s restauračními systémy nejčastěji a to na vytváření objednávek.
- Pomocný personál číšníka - pracovní síla, která je do provozoven mnohdy nabíraná jen krátkodobě za účelem personálního zajištění větších akcí jako svatby, firemní večírky a podobné. Jedná se o roli, která se k restauračním systémům většinou vůbec nedostane.
- Šéfbarmán - pracovník starající se o chod baru. V komplexním systému se často vychází z předpokladu, že velké provozovny a komplexy se skládají nejen z restaurace, ale často jsou přidružené i bary. Využití restauračního systému je v tomto případě podobné jako u vrchního číšníka, ale se zaměřením na barovou část.
- Barman - podobná role jako číšník, ale se zaměřením na bar.
- Pomocný barový personál - pomocný personál, který se stará o hladký chod baru tím, že ulehčuje práci barmanů. S restauračními systémy prakticky nepracuje.
- Šéfkuchař - hlavní pozice kuchyně se zaměřením na personální záležitosti a kvalitu podávaných pokrmů. Se systémem pracuje v rámci sestavování menu a kontroly dostatku surovin pro kuchyni.
- Kuchař - stará se o přípravu pokrmů podle rozhodnutí šéfkuchaře. Kuchaři mohou být specializováni na různá odvětví. S restauračními systémy pracují na úrovni získávání informací o tom, co je objednáno od zákazníků a co má být vydáno z kuchyně.
- Pomocný personál v kuchyni - pomocná síla chystající oblohy, odklízějící nádobí a podobně. Se systémem také prakticky nepracuje.

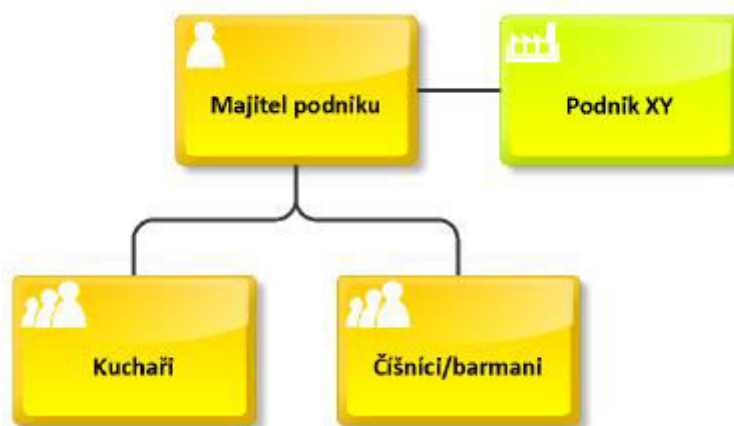
- Sommelier - pracovník starající se o nákup, skladování a podávání vína klientům. Mimo to společně s šéfkuchařem sestavuje jídelní lístek a vybírá vhodná vína k pokrmům (BUREŠOVÁ, 2012). Jeho práce se systémem je obsáhlá. Potřebuje mít přístup do skladů kvůli zásobám, sestavování nabídky stravovacího zařízení a přehledům objednávek restaurace. Tato pozice je však v restauracích velice málo obsazovaná.



Obrázek 1 – Kompletní hierarchie personálu velké restaurace

V běžné praxi však bývá častá mnohem skromnější hierarchie. Jednotlivé role se postupně slučují a stávají se univerzálnějšími. Drtivá většina podniků si vystačí s modelem obsahující role majitel, číšník a kuchař. Neobvyklý není ani model, kdy majitel je buď přímo číšníkem, nebo kuchařem. Pokud se podnik začne rozrůstat, tak zvyšuje počty číšníků a dalšího personálu.

Díky tomu se velice často střetneme s číšníky, kuchaři, barmany, ale málokterý podnik má svého vrchního číšníka, šéfbarmana a již velmi málo zařízení má i svého sommeliéra. Většinou je model restaurace mnohem skromnější.



Obrázek 2 – Nejčastější obsazení personálu stravovacích zařízení

1.4 Obecné požadavky na systém

I když je každé stravovací zařízení jiné a požadavky se tedy mohou lišit, tak i v této problematice je několik bodů, které jsou pro všechna zařízení stejné. Níže uvádím jejich soupis, který mi bude sloužit i jako část kritérií pro hodnocení stávajících systémů. Kvalitní systém by pak měl splňovat všechny níže uvedené body, nebo by měl být postaven tak, aby mohl být snadno rozšířen o požadované funkce.

1.4.1 Ochrana dat a rychlost systému

Stejně jako u ostatních počítačových aplikací tak i u restauračních systémů musí být kladen důraz na dostupnost, ochranu dat a rychlost systému. Rychlost systému je důležitá, protože systém funguje v provozu zařízení a případné prodlevy mohou vést ke zpomalení obsluhy⁷.

U restauračních systémů by se však požadavek na rychlost dal rozdělit do dvou částí. Operace jako sestavování nabídky, operace se skladem nebo statistické přehledy nemají na rychlost tak vysoké nároky. Oproti tomu objednávky, přihlašování do systému a obecně operace pro obsluhu systému za provozu jsou na rychlost velice citlivé a je nezbytně nutné, aby byly co nejrychlejší.

Je třeba, aby v restauračním systému byla spolehlivě uložena také data, která musí být dostupná a zálohovaná. V případě ztráty dat by mohlo dojít ke ztrátě informací o aktuálních objednávkách zákazníků a zboží dostupného ve skladu, což by mohlo vést k nutnosti inventarizace zboží. Případná ztráta by znamenala okamžité komplikace pro provoz.

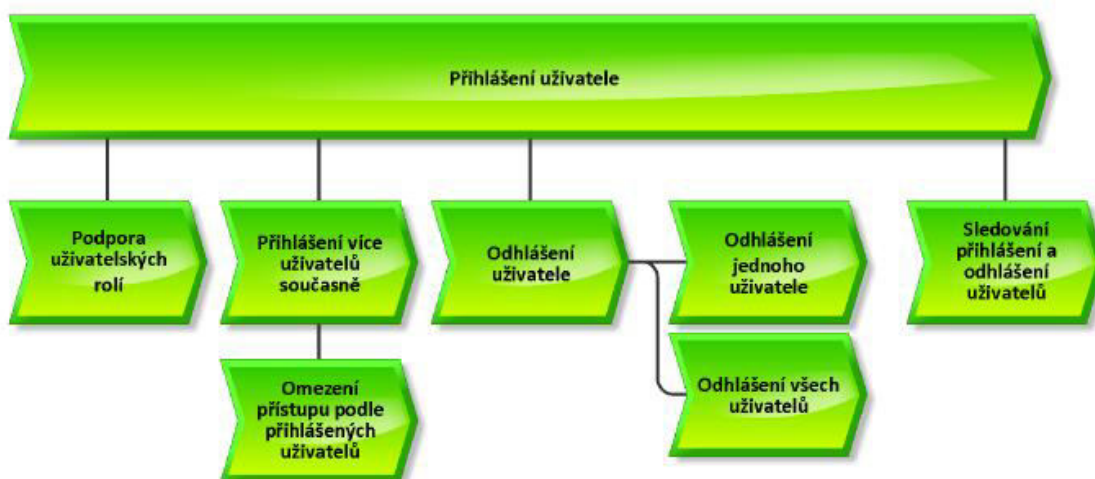
V případě, že by data byla nedostupná, hrozí vznik podobné komplikace jako v případě ztráty uložených dat. Situace, kdy se k objednavce nelze dostat v době, kdy si jí zákazník přeje zaplatit, je stejná, jako by v systému tato informace vůbec nebyla.

⁷ Kdyby byla odezva mezi zadáním požadavku a vrácením výsledku příliš dlouhá, vedla by tato situace k opožděnému zadávání objednávek do systému.

1.4.2 Oblasti, které mohou být automatizované systémem

Oblastí, které mohou být zohledněné v systému je mnoho, ale vytvořit a spojit je všechny do jedné aplikace by bylo zbytečné. Samotný chod stravovacího zařízení si vystačí pouze s malou částí z nich. Je však podstatné, aby byl systém připraven na případná rozšíření⁸. Oproti tomu jsou části, bez kterých by nebyl restaurační systém použitelný. Jednotlivé funkčnosti s jejich základním přehledem obsažených procesů jsem sestavil níže.

- Podpora přihlášení a uživatelské role - rozsáhlé systémy by měly mít mimo jiné zabezpečení na úrovni přihlašování pod různými uživatelskými rolemi, které mají přístupná odlišná práva. Je zřejmé, že stejné údaje, které vidí majitel podniku, by neměl mít dostupné číšník. V případě restauračních systémů je problematika uživatelů poněkud složitější. Systém musí být stavěný na neustálé přihlašování a odhlašování uživatelů. Ve většině stravovacích zařízení je totiž běžný model, kdy s jednou aplikací pracuje více číšníků na stejném stroji, a číšníci se v průběhu dne mění. U menších zařízení může i majitel dělat číšníka souběžně s dalším zaměstnancem a systém musí umět určit kdo je přihlášen a podle toho upravit práva a dostupné části.

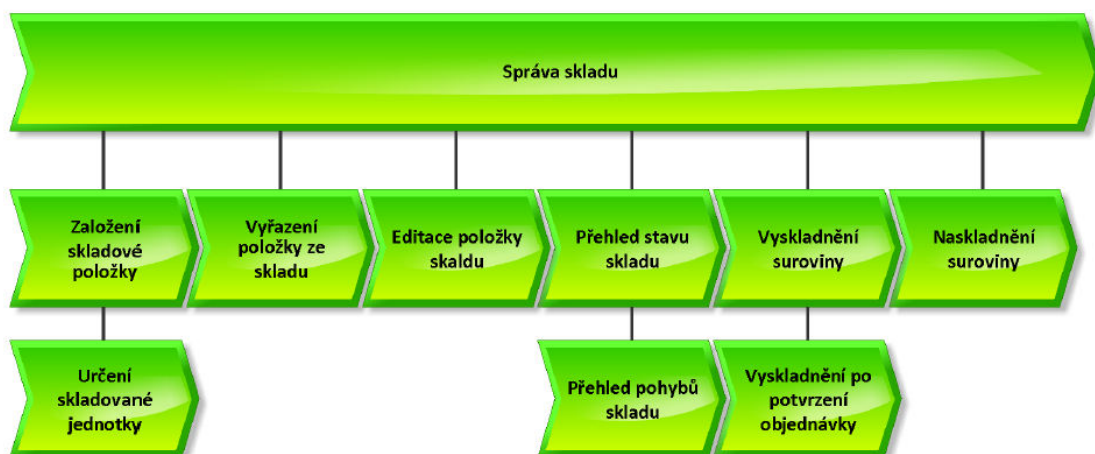


Obrázek 3 – Přehled základních procesů přihlašování

- Správa skladu - restaurační systém by měl mít možnost správy skladových zásob podniku. Základním požadavkem pro skladovou část je evidování jednotlivých surovin a jejich množství. V případě restaurací je častěji nutné evidovat reálné množství v měrných jednotkách suroviny⁹. Evidenci i podle obalů by nebylo technicky možné udržet, protože nevíme, který obal se vezme k prodeji dříve. Samotný sklad by pak měl disponovat možností naskladnění a vyskladnění libovolného množství surovin a evidenci, kdy jaká operace proběhla.

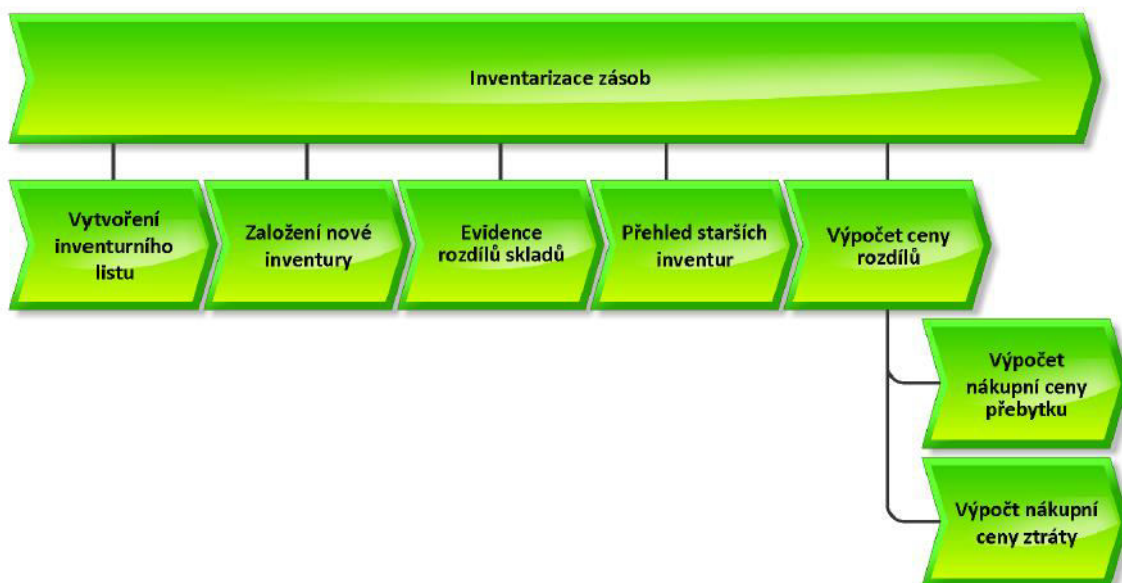
⁸ Příkladem může být modul, který by sommelierovi sloužil pro evidenci vína a pomáhal mu s přiřazováním aktuálně dostupný lahví vína k nabídce.

⁹ To například znamená evidovat, že je ve skladu 5 litrů finské vodky, ale již není rozhodné, jestli jsou v 5 litrových lahvích, nebo v 10 lahvích o objemu 0,5 l.



Obrázek 4 – Přehled základních procesů správy skladu

- Inventarizace skladu - každé stravovací zařízení dělá pravidelné inventury, které slouží k odhalení nepřesností při obsluze zákazníků. Nesrovnalosti ve skutečném stavu skladu a stavu skladu, který by měl být podle systému, vznikají vždy. Někdy se jedná pouze o obyčejnou lidskou chybu jako rozlité nápoje, nepřesně nalitou míru a podobně. Jindy může jít o nepoctivost personálu, která je v tomto oboru velice častá. Systém by na toto měl reagovat vlastním algoritmem, který odhalí nepřesnosti podle zadaných dat v objednávkách s porovnáním vůči skutečným skladovým záznamům¹⁰. Na tato data by pak měl reagovat a sestavit podle nich hlášení, se kterým pak může dále pracovat majitel, nebo vedoucí podniku.



Obrázek 5 – Přehled základních procesů inventarizace

¹⁰ Pro kontrolu se využívají takzvané inventurní listy.

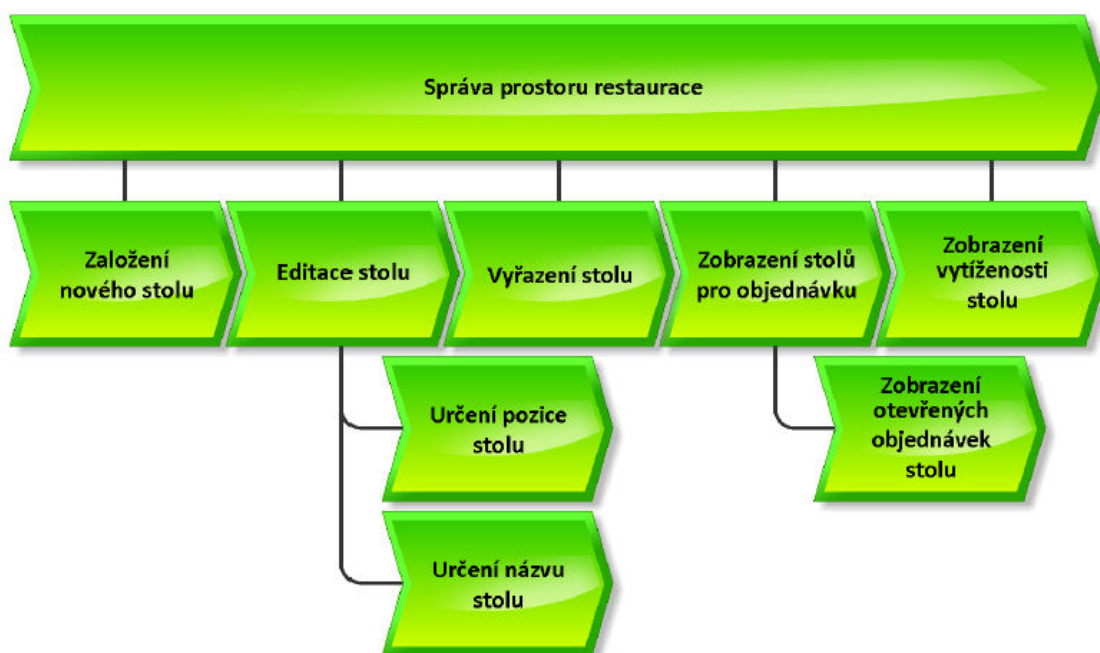
- Tvorba nabídky - aby bylo možné účelně spravovat chod restaurace pomocí restauračního systému, tak je vhodné mít v systému možnost tvorby nabídky stravovacího zařízení. Nabídku by mělo být možné tvořit na základě surovin, které jsou evidované ve skladu, protože to je efektivní způsob, jak provádět snižování stavů surovin ve skladu při objednávkách. Mimo to tvorba nabídky přes systém umožňuje práci se skladovou zásobou¹¹.



Obrázek 6 – Přehled základní procesů nabídky

- Správa rozložení prostoru stravovacího zařízení - restaurační systém by měl být schopný schematicky zobrazit rozložení stolů provozovny a umožnit majiteli případnou změnu tohoto rozložení. Správa rozložení není klíčová pro všechny kategorie stravovacích zařízení, ale je velice častá a měla by být obsažena již v základu využívaného systému.

¹¹ Doplnkovou funkcí pak může být například tvorba jídelního lístku přes systém a jeho zveřejňování nebo rozesílání zákazníkům.



Obrázek 7 – Přehled základní procesů rozložení prostoru restaurace

- Správa objednávek - objednávky jsou základním stavebním kamenem každého stravovacího zařízení. I kdyby systém neuměl nic jiného, tak by měl mít možnost vytvoření objednávky a její evidenci. Pro efektivní práci se systémem je však nutné, aby byly dostupné funkce jako stornování objednávky, rozdělení objednávky, částečné uhrazení, zobrazení přehledu objednávek, filtrování v přehledech objednávek a další.



Obrázek 8 – Přehled základních procesů objednávek

Oblastí, které systém pokrývá, může být i více, ale výše popsané jsou hlavní, které by podle mne měl systém mít, aby byl dobrým nástrojem pro automatizaci provozu stravovacího zařízení.

1.5 Rešerše restauračních systémů

Nyní, když jsem již ve své práci definoval rozdělení a potřeby jednotlivých zařízení, je třeba zaměřit se na to, co nabízí dnešní trh a jaké systémy jsou stále ještě dostupné a využívány. Vzhledem k tomu, že restauračních systémů je velké množství, podívám se jen na část z nich.

Nejčastěji jsou využívány dva typy pokladen. Buď klasické kasy, s malým displejem, nebo restaurační systémy s dotykovou obrazovkou. Dále je pak hojně vyžívaný způsob papírové účtenky, kam obsluha dělá čárky, ale tento postup nelze považovat za restaurační systém.

Díky rozmanitosti stravovacích systémů není jejich porovnávání jednoduché, ale i přes to se dá najít několik společných kritérií, která by měl systém vykazovat, a dají se objektivně zhodnotit.

Pro hodnocení jsem vybral tyto základní kategorie:

1. Existence základních funkcí pro správu stravovacího zařízení.
2. Nároky na provoz systému.
3. Možnost získání testovací aplikace.
4. Uživatelská prostředí a dokumentace.
5. Pořizovací a provozní cena.

1.5.1 Agnis

Systém Agnis¹² svým rozsahem může pokrývat kromě stravovacích zařízení také potřeby zařízení ubytovacích. Aplikace je poskládána z jednotlivých modulů, kde každý plní určitý účel. U této aplikace je vidět její dlouholetý vývoj a propracovanost. Reference uvedené na jejich stránkách jsou důkazem, že se jedná o hojně využívaný software¹³. Modul Restaurace by měl pokrývat potřeby širokého spektra kategorií stravovacích zařízení, ale sám o sobě je zaměřený pouze na správu lokálů, zahrádek a dalších obslužných restauračních prostor. Pro evidenci skladů a potřeb kuchyně jsou třeba další placené moduly a to Kuchyně a Sklad.

Pokud se zaměříme pouze na modul Restaurace, tak ten umožňuje zapojení samotné pokladny pro vyváření objednávek včetně mobilních číšníků, tvaru restauračních prostor podle stolů a židlí a přiřazování objednávek ke stolům a židlím, tisk objednávek, jejich přehledy, tvorba bonů¹⁴ pro kuchyň, tvorby různých ceníků, vyhodnocování tržeb, odpis skladových zásob a identifikaci číšníků přihlášených do systému.

¹² <http://www.agnis.cz/>

¹³ http://www.agnis.cz/2_cestina/4_reference

¹⁴ Bon je v tomto případě papír s informací pro kuchaře co má být připraveno.

Samotná komunikace jednotlivých modulů je však poněkud komplikovanější. Zaměřím se například na vyskladnění zboží ze skladu. Většina restauračních systémů funguje tak, že skladovou zásobu sníží buď při vytvoření objednávky, nebo při jejím potvrzení. Systém Agnis však využívá princip, kdy skladová zásoba není odečítána automaticky, ale je nutné její odečtení spustit manuálně po tom, co některý z uživatelů provede naskladnění zásob do skladu. Tento princip je nevhodný zejména pro plánování množství surovin skladu.

Z pohledu systémových nároků by měl být systém instalovatelný na většinu dnešních počítačů. Potřebná paměť se pohybuje do 400 MB s nároky na procesor optimálně 2 GHz a operační paměť 1 GHz. Systém by měl jít instalovat na běžné počítače se systémem Windows a Linux a měl by podporovat dotykové monitory.

Základní demoverze aplikace je k dispozici zdarma a je zaslána na e-mail, nebo poštou. Rychlost komunikace a zaslání je až překvapivě vysoká. Při žádosti o demoverzi jsem byl kontaktován do 10 minut od zaslání formuláře s žádostí. Demoverze je však pouze orientační a bez patřičného uzpůsobení je velice strohá. Je zde však nabízena možnost bezplatné konzultace a předvedení systému.

Dodavatel systému poskytuje volné minuty pro zaškolení práce se systémem a skype podporu. Po vyčerpání volných minut je další konzultace zpoplatněna. V oblasti dokumentace je k dispozici sekce častých dotazů na webových stránkách a samotná aplikace obsahuje nápovědu v podobě dokumentů, které jsou distribuované společně s instalačním balíčkem aplikace. Podle demoverze se zdá systém celkem příjemný, moderně pojatý a graficky zdařilý. Většina prvků byla pochopitelná i bez náhledu do dokumentace.

Cenová politika tohoto systému je poměrně komplikovaná. Každý modul má své ohodnocení a jeho rozsah cenu navyšuje. Cena modulu restaurace začíná na částce 7 500 Kč, a to v případě, že provozovna má maximálně 50 míst pro hosty. V případě provozovny s místy až pro 150 hostů činí cena modulu 13 500 Kč. Pokud vezmu v potaz cenu kompletního systému, skládajícího se z modulů Restaurace, Kuchyně a Sklad pro zařízení do 100 míst, tak činí celková částka za aplikaci 38 000 Kč.

Pokud aplikaci zhodnotím jako na celek, tak má jistě co nabídnout a pokryje většinu potřeb stravovacích zařízení. Jasnou nevýhodou jsou však jednotlivé moduly, které nejsou samostatně využitelné, a vysoká pořizovací cena aplikace. Mimo to se úplně neztotožňuji se systémem skladových zásob a jejich vyskladnění až ve chvíli, kdy si majitel najde čas a provede potřebné operace. Neaktuální hodnota skladu může velice snadno vést k nedostatku zboží a tím i k ohrožení dostupnosti kompletní nabídky podniku. Systém je zajímavý, ale jeho pořizovací cena mi přijde nepřiměřená.

1.5.2 AWIS GASTRO

Aplikace AWIS GASTRO¹⁵ je celkově rozsáhlá a na první pohled profesionální. Stejně jako u předchozího hodnoceného systému tak i v tomto případě je restaurační systém jen jedním z mnoha samostatných modulů. Zde je však členění produktu trochu jiné. Restaurální systém vystupuje jako jeden modul, který je dále rozdělený na další podmoduly, jenž jsou však dodávány jako celek. Výjimkou jsou pak části zákaznické evidence, mobilních číšníků, kamerového sledování prostor restaurace a dalších modulů, které nejsou pro běžné stravovací zařízení klíčové.

Při testování systému se potvrdily informace, které tvůrce prezentuje na svých stránkách. V tomto případě se skutečně jedná o restaurační systém, který pokrývá vše. Mimo tvorby objednávek, jejich neustálých přehledů a členění, jsou k dispozici funkce jako dopočet částky k vrácení, úprava ceníků podle denní doby, vložení úkolů pro personál, mapa restaurace a mnohé další.

Samotný modul restaurace je pak v základu rozdělen na tyto podmoduly:

- Kasa - obsahuje účtování, přehledy objednávek, rychlé výběry z nabídky a další vlastnosti. Jedná se o pokladní část, která ve všech směrech podporuje provoz stravovacího zařízení.
- Sklad - skladová část systému umožňuje přehled skladovaných surovin, možnost nastavení minimálního množství a objednávání zboží, které dochází. Dále pak export a import do CSV souborů, tvorbu inventur, sestavování nabídek pro objednávky, skladovou marži a další. Skladové možnosti jsou v tomto případě opravdu obsáhlé a pro uživatele zajímavé.
- Ceník - umožňuje tvořit různé cenové kalkulace, akce na nabídku (pravidelné i jednorázové), jídelní lístky a další. Samotné navázání jednotlivých cen pak probíhá na sklad a nabídky přes EAN kódy, které jsou podle mého v tomto směru optimální cestou.
- Uživatelé - správa uživatelů systému. Systém má možnost 21 oprávnění s 250 variantami nastavení, což by mělo bohatě stačit na pokrytí všech rolí, které se ve stravovacích systémech vyskytují.
- Mapa stolů - poskytuje možnost nastavení rozmístění jednotlivých prvků stravovacího zařízení a to podle stolů i židlí. Mimo to dává k dispozici možnost rozčlenit systém stolů na jednotlivé úseky podle číšníků, což umožňuje jednoduchou správu takzvaných rajónů a přehled nad obsluhou.

Systém je dodáváný buď na samostatném CD, nebo jako sestava s počítačem a dotykovým monitorem. Variant sestav je i více podle požadavků provozovny. Mimo to je možné zakoupit síťovou licenci, která umožní propojení více provozoven mezi sebou a tím zprostředkovávat online řešení. V této podobě je však vhodné jen pro větší podniky a řetězce.

¹⁵ <http://www.kasa-pokladna.cz/>

Dodavatel bohužel nikde neuvádí systémové nároky provozu aplikace. Osobně jsem aplikaci zkoušel na systému s 2 GHz operační pamětí a procesorem 1 GHz. Na tomto stroji aplikace fungovala bez jakýchkoliv komplikací.

Uživatelské prostředí je možné ovládat pomocí myši, ale dotyková zařízení budou na běžnou obsluhu jistě lepší. Možná právě proto je systém nabízen i s hardware pro svůj provoz. Aplikace pro svůj běh využívá databázi MySQL s ODBC konektorem. Tento konektor je instalovaný zvlášť, i když je na něj uživatel naveden v průběhu standardní instalace.

Systém je možné si zdarma vyzkoušet. Demoverze je volně ke stažení po registraci na stránkách dodavatele¹⁶. I když se nejedná o plnohodnotnou verzi, tak jistě dobře postačí pro vytvoření přehledu o tom, co tato aplikace dokáže.

Uživatelské zpracování a celková obtížnost práce se systémem jsou na celkem dobré úrovni. Systém je moderně pojatý a ve většině případů i přehledný. Jsou zde však místa, která jsou zbytečně přeplněná a některé možnosti úpravy pracovního prostředí jsou také až zbytečně obsáhlé. Systém samotný je však doplněn dobrou dokumentací a zajímavými videonávody, které představují práci se systémem.

Cenová politika poskytovatele je celkem obsáhlá. Jsou zde možnosti pronájmu systému a to jak krátkodobé, tak dlouhodobé nebo možnost pořízení samostatné aplikace pro instalaci na vlastní zařízení, ale i možnost zakoupení aplikace i s kompletním hardware. V případě koupi samostatné aplikace je základní cena kompletní verze 19 239 Kč. V případě zakoupení základního systému i s hardware začíná cena na 36 179 Kč a roste až za hranici 50 000 Kč.

Celkový dojem ze systému je velice dobrý. Pokrytí oblastí, které jsou důležité pro správu stravovacích zařízení, je úplné, systém je až na některé drobné detaily uživatelský. Tento software však svým konceptem nijak nevybočuje z řady restauračních systémů, které jsou na trhu dostupné. Jeho pořizovací cena je v porovnání s ostatními systémy průměrná. Když vezmu pokladní systémy podobné kategorie, tak se jistě jedná o vyšší třídu v oblasti podpory provozu stravovacích zařízení.

1.5.3 Conto

Tato aplikace je počinem společnosti CÍGLER SOFTWARE¹⁷, která je podepsána například pod účetním systémem Money. Není tedy divu, že aplikace Conto umožňuje XML přenos právě do tohoto systému. Po prozkoumání aplikace jsem zjistil, že je s Money v mnohém podobný.

Aplikace je dodávána jako celek, který pokrývá všechny potřeby stravovacího zařízení a je k němu možné dokoupit rozšiřující části. V tomto případě se však nejedná o části pro samotné stravovací zařízení, ale například pro účetnictví, ubytování a další.

¹⁶ <http://www.kasa-pokladna.cz/cs/zkusebni-verze>

¹⁷ <http://www.money.cz/pos/pokladni-software/conto/>

Samotná aplikace se skládá z konfiguratoru, který umožňuje nastavovat vzhled systému a řídit jeho provoz a obslužné části. Mimo to je zde ještě doplněk, který umožňuje síťové propojení jednotlivých systémů. Každý systém pak však musí mít vlastní zakoupenou obslužnou část. Základní nabídka poskytuje verze, které vždy obsahují konfigurační a obslužnou část. Jejich rozdíl je pak v doplňkových modulech.

Část pro obsluhu je grafické rozhraní, které umožňuje správu objednávek, stolů zařízení a základní skladové operace. Konfigurační část se pak stará o nastavení typů plateb, měn, evidenci skladových surovin a podobně. Z pohledu pokrytí základních potřeb stravovacího zařízení je zde pravděpodobně vše, co by měl restaurační systém ovládat. Uživatel zde najde správu skladu, tvorbu nabídky, možnost inventur, propracovanou správu uživatelů i kompletní přehledy objednávek. Z tohoto pohledu je aplikace kvalitní a propracovaná.

I když je systém možné nainstalovat na většinu běžných zařízení, tak je na první pohled vidět, že hlavně obslužná část je stavěná na dotykové monitory. Obsluha myší je nepraktická a zobrazení značně nekvalitní. Výrobce systémové nároky přímo nezveřejňuje, ale testovací verze, která je časově omezená, ale jinak se jedná o plnohodnotnou aplikaci, není nijak náročná. Pro instalaci stačilo 172 MB volného prostoru. Když vezmu v úvahu již částečné přednaplnění daty, tak mám za to, že pro celkovou aplikaci bude stačit 250 MB prostoru. Systém jsem instaloval na počítač, který měl 2 GHz operační paměti a 1 GHz procesor. Běh systému byl plynulý a stroj nebyl prakticky vůbec během zatížen.

Provozovatel poskytuje plnohodnotnou testovací trial verzi ke stažení na svých stránkách bez nutnosti registrace. Trial je omezen na 30 dní, které postačují pro dostatečné otestování. Samotná instalace je jednoduchá a rychlá. V tomto ohledu jsou poskytované služby nejlepší, se kterými jsem se při porovnávání setkal.

Když se zaměřím na uživatelskou stránku, tak zde systém není dle mého názoru řešen úplně nejlépe. Rozdělení na dvě části je trochu nepraktické, aplikace je graficky strohá a pro uživatele nepřehledná. Vše se nastavuje pomocí systému tří oken, kde první vybírá nastavovanou část, druhé ukazuje položky, které jsou v části evidované a třetí hodnoty, které se dají v položce nastavit. Nastavení je velké množství, ale přehlednost je slabá. Do tohoto ještě vstupují volby hlavního menu, které jsou však z velké části duplicitní s těmi, které jsou v levém menu. Jako poslední je tu možnost změny rozložení obslužné části, která je lépe zpracovaná, ale není uživatelsky jednoduchá.

Obslužná část je na tom podobně. Obsluha systému je nejasná a neúživatelská. Mám za to, že systém by mělo jít do jisté míry ovládat intuitivně, což však o tomto systému rozhodně říci nelze. Grafické prostředí obslužné části je sice rozdělené do bloků a vše je popsáno, ale pro běžného uživatele obsahuje obrovské množství funkcí, které se vzájemně pletou.

Celý systém je popsán pomocí nápovědy, která však vykazuje podobné nedostatky jako samotná aplikace. Nápověda se tváří jako velice obsáhlá, ale celkově je nepřehledná. Některé položky jsou pak vysvětleny jen jedním krátkým a nic neříkajícím odkazem. Dodavatel sice nabízí možnost zaškolení, ale ta je již zpoplatněna.

Cenová politika dodavatele je poměrně jednoduchá. Tři základní aplikace stojí 6 900 Kč (Basic), 8 900 Kč (Standard), 10 900 Kč (Max). K tomu je však nutné započíst ceníkově účtované zaškolení a instalaci systému. V případě síťového propojení a ještě jedné obslužné stanice se pak cena zvedá o dalších 5 400 Kč. Ani tato částka však není konečná, protože pro plné fungování je ještě třeba příslušný hardware a jeho cena se pohybuje od 20 000 Kč výše. Dále je třeba počítat ještě s koupí tiskárny, jejíž cena se pohybuje okolo 5 000 Kč. Za střední verzi systému s levným HW tedy zájemce zaplatí 34 000 Kč, což je průměrná cena podobných systémů.

Celkový dojem z této aplikace je hodně smíšený. Pokrytí oblastí stravovacích zařízení je dobré, systém je přizpůsobitelný, nenáročný a obsahuje mnoho užitečných doplňkových možností. V tomto ohledu se jedná o kvalitní systém. Co se mi však úplně nezamlouvá, je systém správy ve dvou aplikacích, které jsou vzájemně propojené a rozporuplné ovládání systému, které není uživatelské. Je vidět, že výrobce chtěl pokrýt všechny potřebné situace, ale tím vytvořil kolos, který umí vše, ale práce s ním je obtížná. Věřím však, že s trochou praxe by bylo možné systém uzpůsobit a naučit se s ním pracovat. Na druhou stranu představa zaškolení nového uživatele systému by znamenala velkou časovou náročnost.

1.5.4 MRP – vizuální účetní systém

Tento restaurační systém je typickou ukázkou obsluhy stravovacího zařízení pouze přes software bez využití speciálního hardware. Stejně jako v předchozích případech tak i zde je restaurační systém pouze jednou částí zaměření firmy MRP¹⁸.

Výrobce restauračního systému ve své prezentaci tvrdí, že se jedná o software, který zvládne různé kategorie stravovacích zařízení. Mezi základními vlastnostmi systému je tisk účtenek, sumarizace objednávek, uzávěrky, přehledy prodaných jídel a další. Mimoto systém umožňuje propojení se skladovým hospodářstvím.

Z pohledu pokrytí základních potřeb stravovacích zařízení je zde až na možnost napojení na skladovou část prakticky vše. Jsou zde objednávky s grafickým zobrazením stolů, akce na nabídku, sestavování nabídky a příprava jídelníčků, evidence zákazníků a receptury nabídky. Vše se zdá být relativně jednoduché a komplexní. Jedinou nevýhodou, kterou jsem již zmiňoval je fakt, že skladová agenda je samostatným modulem. Systém tedy umožňuje sestavování receptů ze surovin, ale o tom, kolik má provozovatel surovin na skladu nic neví bez dalšího rozšíření.

Samotné nároky na provoz systému jsou podle provozovatele velice nenáročné. Systém požaduje Windows XP nebo novější a pro síťovou verzi propojení počítačů pomocí kabelu. Je trochu zvláštní, že nejsou nároky popsány detailněji, ale pokud se na samotnou aplikaci podívám, tak její struktura a grafická stránka sama o sobě naznačuje fakt, že pokud nějaký počítač zvládne provoz systému Windows XP, tak musí zvládnout i tuto aplikaci.

¹⁸ <http://www.mrp.cz/>

Testovací verzi výrobce nabízí na svých stránkách volně ke stažení bez registrace. Soubor má necelých 25 MB a jeho instalace je kroková a velice jednoduchá. Systém se pak automaticky nespustí, ale vytvoří zástupce na ploše, takže je možné jej spustit ručně. Žádná další nastavení systému nejsou pro jeho zkoušení nutné. Jediné co mne zaskočilo, bylo spuštění aplikace, kde bylo požadováno heslo. Výrobce nějak opomenul tento krok zmínit, ale heslo je zde jen pro úplnost a je prázdné.

Uživatelská stránka mne velice mile překvapila. I když je grafika poněkud starší¹⁹, tak uživatelské stránce se nedá téměř nic vytknout. Ze začátku je možné vybrat uživatele a pak se zobrazil samotný systém, který je jasný a přehledný. Občas jsou některé volby trochu moc zanořené po několika krocích, ale proklikání je logické takže se jedná pouze o drobnost.

Překvapivá je však pořizovací cena této aplikace. Výrobce udává pouhých 3 498 Kč bez DPH. V kombinaci s tím, že se tato aplikace dá nainstalovat prakticky na libovolný počítač, tak se jedná o velice zajímavé podmínky. Pro úplnost aplikace by bylo vhodné zakoupit i skladovou část, kterou je možno pořídit za 1 618 Kč bez DPH.

Když jsem si vybíral tento systém pro zhodnocení a rešerši, tak jsem neočekával zázraky. Webové stránky výrobce jsou zastaralé, grafické zpracování aplikace je velice strohá. Po otestování jsem však zjistil, že se jedná o propracovanou aplikaci, která je jednoduchá a podle mě výborně splňuje všechny podmínky provozu stravovacích zařízení. Až na grafickou stránku jsem byl při testování velice spokojen.

1.5.5 KelEXPRESS

KelEXPRESS²⁰ je v první řadě ekonomický software pro středně velké firmy, který má modul Restaurace. Ten je pak zaměřen na provoz stravovacích zařízení. Opět je zde i podobný scénář, který se týká modulového oddělení části restaurace a části sklad, i když v tomto případě je nabídka postavená rovnou tak, aby si potencionální zájemce mohl zakoupit oba moduly v jednom balíčku. I přes mnoho podobností je zde samotný princip fungování provozu aplikace trochu odlišný. Výrobce aplikaci provozuje na základě měsíčních pronájmů.

Z pohledu pokrytí oblastí provozu stravovacích zařízení je v tomto případě základ celkem dobrý. Je k dispozici přístup podle práv a jejich nastavování, tvorba rozložení restaurace, objednávky a jejich administraci na úrovni přesouvání, částečného uzavírání a podobné, dále pak tisk dokladů, číselník receptur, ceníky a bony pro kuchyň. Myslím si, že z tohoto pohledu je v systému obsaženo vše. Ovšem ani tento výrobce si neodpustil do modulu restaurace přidat základní správu pro ubytovací zařízení. Což samozřejmě není na škodu, ale osobně to považuji za zbytečné. Jako lepší řešení bych chápal situaci, kdyby byl do modulu restaurace více zasazen modul skladu.

¹⁹ Celá aplikace má starší uživatelské rozhraní.

²⁰ <http://www.keloc-software.cz/produkty/kelexpress/>

Jak již bylo řečeno, tak modul sklad je opět zvlášť, ale výrobce nabízí balíček, kde jsou oba moduly spolu. Sklad pokrývá chybějící nedostatky modulu restaurace v oblasti kontroly skladových zásob a přímého vyskladňování.

Podle dodavatele je systém kompatibilní s běžně používaným hardware používaném ve stravovacích zařízeních. I když systém může běžet na klasickém počítači, tak je uzpůsoben na dotykové monitory. Ostatní nároky aplikace jsou zanedbatelné stejně jako u ostatních systémů. Jediná informace, která mi zde chybí, a myslím si, že je celkem podstatná, je který operační systém tuto aplikaci podporuje. Vzhledem k tomu, že aplikace funguje na databázi MS SQL, očekávám, že bude striktně vyžadovat operační systém Windows.

Výrobce poskytuje možnost otestování aplikace pomocí připojení na vzdálený server. Ke zprovoznění stačí připojení k internetu a malá aplikace pro vzdálenou plochu, která je k dispozici na stránkách výrobce a sama se automaticky připojí, kam je třeba. Tato myšlenka je rozhodně zajímavá, a mám za to, že je funkční. Jen samotná aplikace byla díky vzdálené ploše trochu pomalejší, než by bylo vhodné.

Uživatelská stránka aplikace je velice jednoduchá. Obsahuje přehledy pro objednávání, zobrazení stolů i židlí, výběr a sestavování nabídky. Vše je na první pohled jednoduché, ale mám za to, že aplikace je někdy nepředvídatelná. Ve chvíli, kdy uživatel chce uzavřít objednávku na stůl, se zobrazí donastavení účtů. Při jakékoliv akci pak byl konečný výsledek stejný, znovu se zobrazil přehled stolů. V aplikaci tedy buď chybí nějaký mezikrok s informací o tom, že akce byla provedena, nebo je aplikace chybná. V tomto případě jsem však přesvědčen o tom, že se jedná pouze o nějaké nastavení, které je díky více přístupům od testujících uživatelů značně nestabilní.

I když je přehlednost systému vysoká a dokumentace dobrá, tak musím vytknout grafickou stránku zpracování. Grafika je zastaralá a až nepříjemně uniformní. Mimo to, že je určená pro dotykové monitory, tak jsou zde prvky, které bez myši nebude možné plně ovládat. Jedná se hlavně o horní dropdown menu, které zůstává otevřené na základě přítomnosti myši. Vyzkoušet systém na dotykovém zařízení bohužel není přímo možné, ale jsem přesvědčený, že toto menu bude komplikací.

Na rozdíl od všech ostatních systémů je zde cena nastavená na základě měsíčního pronájmu, který se mění podle tří pásem s tím, že verze jsou odlišené podle počtu dat, která mohou obsahovat. Pak zde jsou ještě takzvané oborové balíčky, které poskytují specializovaná řešení pro různá odvětví podnikání. Hodnocený restaurační systém je možné pronajmout ve dvou verzích. Základní verze vychází na 455 Kč měsíčně bez DPH a profi verze je za 1 170 Kč za měsíc bez DPH. K tomu je třeba počítat i s pořízením hardware, který tato firma vůbec nenabízí. Na myšlence pronájmu je dobré, že pronájem obsahuje nejnovější aktualizace systému. Na druhou stranu se jedná o pravidelný náklad, který bude nutné hradit po celou dobu provozu stravovacího zařízení. Cena základní verze za celoroční provoz je 5 460 Kč bez DPH. S tím, že u restauračních zařízení je nutné počítat s delším provozem. Za deset let tedy cena činí 54 600 Kč.

1.5.6 Menu55

Tento systém je již na první pohled jiný než všechny ostatní²¹. Je to jediné české cloudové řešení, na které jsem při testování narazil. Systém funguje na libovolném operačním systému a mobilních zařízeních a je určen jak pro obsluhu, tak pro zákazníka. Celkově by se mohlo jednat o velice zajímavý systém, který bude hodně využíván, ale bohužel má pár nedostatků, které by bylo třeba dořešit.

Jako první se zaměřím na poskytované funkce pro obsluhu. Systém by mohl mít potencionálně vše, tedy objednávky a rozvržení stolů stejně jako ve všech ostatních systémech. V této oblasti žádnou větší slabinu nevidím, nedostatky ale shledávám u skladu. Ten je totiž v této aplikaci pouze formou nabídky, ale skladové přehledy surovin úplně chybí. Naproti tomu jsou zde funkce jako objednávka podle menu přímo zákazníkem nebo SMS zavolání číšníka, které jsou určitě zajímavé, a jistý potenciál zde vidím, ale pouštět do systému přímo zákazníka mi přijde neuvážené. Jednak mohou vznikat nepřesnosti ve chvíli, kdy zákazník bude systém testovat a zadá požadavek, o který ve skutečnosti nemá zájem. Rovněž přivolávání číšníka prostřednictvím SMS považuji za nepraktické, když je zřejmé, že tento postup bude narušovat systém jeho práce a tedy zpomalovat obsluhu.

Jistou výhodou, která se systému nedá upřít, jsou nároky na provoz. Stačí libovolný počítač nebo větší tablet a provoz může začít. Na druhou stranu je třeba dobré připojení k internetu, protože systém není možné provozovat lokálně.

Díky cloudovému řešení je příjemná možnost otestovat si systém přímo na stránkách provozovatele. Musím říct, že systém je skutečně přehledný a na práci velice příjemný. Všechny funkce jsou snadno k nalezení bez využití nápovědy. V tomto ohledu je systém na vysoké úrovni.

Výrobce tohoto systému neprodává hardware, ale čistě software, takže je u ceny nutné počítat ještě s nákladem na zařízení, kde systém poběží. Obecně by mělo stačit jakékoliv zařízení s obrazovkou větší než 7 palců. Samotný provoz je pak placen měsíčně za přístup na server a to podle rozsahu od 350 Kč do 480 Kč za měsíc. V tomto ohledu je částka nízká a oproti ostatním systémům velice výhodná.

Celkový dojem z tohoto systému mám dost smíšený. Cena a pokrytí služeb je výborné, až na slabé zpracování skladu. Myšlenka cloudu a provozu je také určitě plus. Na druhou stranu pouštět zákazníka do systému mi přijde velice nepraktické.

²¹ <http://www.menu55.cz/>

Problém vidím i v ukázkové kalkulaci. Výrobce počítá s nasazením tabletu jako mobilního číšníka a jeden tablet pro kuchaře kvůli objednávkám. Z pohledu provozu zařízení to je však těžko realizovatelné. Standardní mobilní číšník je stavěný tak, aby byl odolný na vliv běžného provozu. Standardní tablet by v provozu přežil pouhých pár měsíců. V případě kuchyně je tablet nevyužitelný, protože v každé kuchyni je značné množství mastnoty, páry a dalších vlivů, které tablet velice rychle zničí.

1.5.7 Klasické pokladny

Ne všechna stravovací zařízení mohou využívat moderní restaurační aplikace. Je tedy vhodné představit i zástupce starších systémů. V tomto případě se jedná o klasické pokladny. Jak již bylo zmíněno v kategorizaci, tak se jedná o jeden stroj s klávesnicí a malým displejem. Fungování takovéto kasy je léty prověřený způsob provozu v mnoha zařízeních a tyto kasy dovedou mnohem více, než se na první pohled zdá. Mohou například evidovat objednávky, kontrolovat skladovou zásobu, rozpoznávat číšníky a mnoho dalšího. Mimo to jsou tyto kasy technicky mnohem odolnější a jejich pořízení levnější.

Bohužel jejich nevýhoda je již v samotné administraci pokladny a získávání dat. Na začátku této práce jsem uvedl, že POS systémy využívají pro svou komunikaci standardy pro COM konektory. Takže k napojení na dnešní počítače je třeba využít přepojku COM na USB. Další komplikací je samotné spojení aplikace pokladny s aplikací v počítači. Tyto aplikace byly obecně stavěny pro mnoho typů pokladen, takže správně nastavit obě aplikace na správný typ pokladny je zdoluhavá záležitost. Test spojení mnohdy probíhá i minutu, takže zkusit různé varianty je časově náročné. Nakonec je problémem samotná komunikace s provozovatelem nebo obsluhou. Displej je malý a díky tomu pojme jen malé množství informací, takže se k získání dat používají tisknuté reporty. Obsluze pak vyjede mnoho informací, ve kterých se musí pracně vyhledávat.

Navzdory svým výhodám a nevýhodám je celkem jasné, že se tento typ pokladen bude používat ještě velice dlouho. Jsou totiž bytelné a pro provoz mnohdy dostačující, tak že provozovatelé nebudou mít důvod tento způsob automatizace měnit.

1.5.8 Souhrnné hodnocení

Restauračních systémů je na českém trhu poměrně hodně. Ze začátku bylo těžké se v systémech orientovat. Jejich struktura je obecně poměrně chaotická, protože se nejedná o samostatné aplikace, ale jsou většinou k něčemu přidružené.

Až na systém Menu55, který se svým konceptem poněkud vymyká standardu, jsou si všechny systémy velice podobné až na drobné odlišnosti ve funkcích, porovnání menu a celkovém obsahu aplikace.

Pokrytí základních oblastí, které by měl systém automatizovat, je obecně na dobré úrovni i když úplně nesouhlasím s tím, že jsou některé funkce vsazeny do modulů, které se musí do systému dokupovat zvlášť.

Docela mne překvapilo, jak malé nároky na běh systémy obecně mají. Když opomenou vhodnost dotykových monitorů, kde je vyšší pořizovací cena, tak by většina systémů mohla fungovat na těch nejméně výkonných počítačích, které je v dnešní době možné pořídit. Jedinou podmínka je pak většinou použití systému Windows. Podpora jiných platforem je spíše výjimkou.

Vesměs všichni výrobci nabízí možnost otestování jejich systému a to jak formou demoverze, trial verze nebo náhledu v předpřipraveném prostoru. V tomto směru je pokrytí dobré, ale zarazilo mě, že ne všechny verze nabízené k odzkoušení jsou vždy funkční. Takovéto pochybení pak vede k jedinému cíli a to ztrátě potenciálního zákazníka.

Pokud jde o grafické provedení a uživatelskou jednoduchost, tak zde jsou nedostatky poměrně značné. Nejčastějším prohřeškem jsou nepřehledná menu, nejasné hlášky při nějaké situaci, nebo neúplná dokumentace. Ze všech systémů, které jsem vyzkoušel nebo jsem prošel jejich uživatelské rozhraní, jsem narazil pouze na jeden originálnější program a to menu55. Jinak jsou si všechny aplikace výrazně podobné.

Pořizovací cena systémů je obecně velice podobná. I když se to na první pohled zdát nemusí, protože jsou zde rozdíly v ceně samotné části aplikace pro správu restauračních zařízení, tak při bližší zkoumání je zřejmé, že se jsou velké rozdíly v tom, jaké moduly je třeba dokoupit, aby byl systém kompletní. Společně s hardware se pak uživatel dostane na částku kolem 35 000 Kč. Druhou možností je pronájem systému, kde se většinou jedná o částky kolem 500 Kč za měsíc bez potřebného zařízení.

Pro lepší přehled jsem sestavil tabulku, kde je zobrazeno bodové hodnocení mnou popisovaných systémů. Hodnocení jsem zvolil na deseti úrovněové stupnici s tím, že hodnota 10 je nejlepší. Pro cenu jsem volil vždy střední třídu nabízené aplikace s moduly a nejlevnější hardware. Ceny jsem pak porovnal mezi sebou.

	Pokrytí základních oblastí	Nároky na provoz	Možnost ozkoušení	Uživatelská jednoduchost	Cena
Agnis	6	8	3	5	7
AWIS GASTRO	8	8	7	6	7
MRP	9	9	9	8	10
Conto	8	7	9	7	7
KelEXPRESS	7	8	8	7	6
Menu55	7	10	10	8	9

Tabulka 1 - Hodnocení testovaných systémů

1.6 Porovnání stávajících řešení s mým systémem

V době, kdy jsem tvořil první návrh svého systému, jsem měl vlastní ucelenou představu o tom, co všechno by aplikace měla umět, aby pokryla potřeby restauračních zařízení. Moje zkušenosti v té době obsahovaly znalost dvou pokladních systémů, velice dobrý přehled ve fungování stravovacích zařízení a možnost konzultovat s provozovateli různých podniků.

Celou svou vizi jsem začal pomalu zpracovávat do formy, která by pro provoz mohla být použitelná, a záměrně jsem ještě v této fázi nezjišťoval, co všechno umí podobné aplikace. Ve chvíli, kdy jsem si udělal průzkum nabízených systémů, jsem zjistil, jak obsáhlou aplikaci jsem vlastně začal tvořit.

Mnoho funkcí, které mají velké aplikace s mnohaletým vývojem, jsem zcela intuitivně zabudoval do základu mého software a na rozdíl od všech zkoumaných systémů jsem vytvořil jednu aplikaci, která je kompletní a není nutné k ní dokupovat další moduly a rozšíření. V tomto případě bych rád poukázal speciálně na sklad, který je nedílnou součástí mé aplikace, ale všechny zkoumané systémy měly sklad pro evidenci surovin oddělený.

I když v mém softwaru v porovnání s podobnými systémy nějaké části chybí, tak se jedná o taková rozšíření, která nepovažuji za podstatná a mám za to, že provozovatelé restauračních zařízení by o podobné funkce ani nejevili velký zájem. Jedná se zde například o modul zákazníci, kam se evidují pravidelní návštěvníci stravovacích zařízení a jejich útraty.

Některé ze systémů obsahují modul zaměřující se na akce a slevy, kde je možné měnit cenu určité nabídky v případě velkých akcí, nebo časově omezené nabídky. Mám za to, že tento modul je vhodným doplněním systému.

Při samotném porovnávání není samozřejmě možné objektivně zhodnotit vše. Mimo to není vždy úplně možné porovnat mou aplikaci s restauračním systémem, který již nějakou dobu běží.

1.6.1 Systém Agnis

Tento systém je typickou ukázkou restauračních systémů pro dotykové monitory. Oproti mému systému je oddělena skladová část, kterou je nutné dokoupit zvlášť, což je běžné i pro mnoho jiných systémů, ale spíše princip vyskladňování zboží není podle mého názoru ideální. Můj systém funguje tak, že sníží stav skladu s každou uhrazenou objednávkou. Zatímco Agnis snižuje sklad manuálně. Vedoucí naskladní suroviny, pak spustí porovnání skladu a ukáže se mu zůstatek. V tomto ohledu je můj systém jistě lépe udělaný. Stejně tak samotná kuchyně a její oddělení do samostatného modulu. Podle mého je tento krok úplně zbytečný a ve své aplikaci jsem sklad, kuchyni i restauraci spojil do jedné části.

Systému se obecně nedá upřít fakt, že je hojně používaný a má za sebou dlouholetý vývoj. Pojetím, technologiemi a celkovou konstrukcí však svůj systém hodnotím jako přínosnější pro provoz stravovacích zařízení.

1.6.2 Systém AWIS

V tomto případě se jedná o podobný systém jako je první porovnávaná aplikace, ale v některých ohledech je mé práci bližší. Aplikace je také rozdělena na moduly, ale tváří se jako jeden celek, který pokrývá celou problematiku stravovacího zařízení. Moduly jsou pak jednotlivé části, které jsou od sebe odstíněné. Stejný koncept používám i ve své práci. Moduly spolu spolupracují, ale nikde nejsou provázány dál než do sousedního kooperujícího.

Některé rozšiřující funkce systému AWIS bych v budoucnu chtěl využít i ve své aplikaci, ale pro samotný provoz nejsou klíčové a jsou tedy pouze jakýmsi bonusem. Jedná se například o napojení kamerového systému pro možnost sledování obsluhy nebo třeba rozšíření mapy restaurace, které je v tomto systému propracovanější. Na druhou stranu toto rozšíření také není pro běžný provoz nijak klíčové.

Pokud se zaměřím na práva, tak zde vidím poměrně velkou slabinu oproti mé aplikaci. V systému AWIS je předpřipravených 21 práv s více jak 250 kombinacemi. Takováto variabilita bude provozovateli spíše překážet než pomáhat. Můj systém, kde má provozovatel možnost definovat vlastní práva, je tedy v tomto ohledu jistě lepší.

Po technické stránce se opět jedná o systém, který je mnohem náročnější než můj vytvářený. Systémové nároky na výkon a paměť nejsou reálně vysoké, ale pro plné ovládání nepostačuje použití myši a je třeba dotykového monitoru. S jeho pořízením je však spojeno zvýšení nákladů. Technické parametry této aplikace neumožňují, aby byla využívána na mobilních zařízeních. I když je poskytováno síťové řešení, tak je nutné jej dokoupit za příplatek, a nejedná se stále o online přístup.

Z uživatelského pohledu je AWIS obsáhlejší, ale díky tomu také méně přehledný. I když vzhledem k možnostem, které poskytuje, nemůžu objektivně porovnat obě aplikace. Je možné, že v případě dopracování stejných funkcí do mé aplikace, by také došlo ke snížení přehlednosti. Na druhou stranu takto rozsáhlý systém je pro drtivou většinu podniků nevyužitelný a z pohledu filozofie použití mé aplikace se ani tímto směrem nebudu dále vydávat.

Když se podívám na celkové porovnání systému AWIS s mojí aplikací, tak musím uznat, že AWIS je mnohem obsáhlejší. Na druhou stranu někdy až moc přeplněný velkým množstvím funkcí, které nejsou podle mého třeba. Z pohledu modulů, které mám v aplikaci a jsou srovnatelné se systémem, je funkcionalita srovnatelná. V mém systému jsem však zachoval vyšší přehlednost. V čem však systém AWIS propadá je cena. Jedná se sice o obsáhlý a jistě kvalitní software, ale základní cena přesahující hranici 35 000 Kč je pro většinu podniků nepřijatelně vysoká.

1.6.3 Systém Conto

Conto je na první pohled typickou restaurační aplikací, která má menu postavené pro dotykový monitor a podobné zobrazení. Po otestování se však ukázalo, že koncept tohoto systému je vzdálený nejen od ostatních restauračních systémů, ale oproti mému má velice odlišnou strukturu.

Na rozdíl od mé aplikace je systém ovládán pomocí dvou nezávislých aplikací. Jedna pro obsluhu a druhá pro vedení. Tato myšlenka jde úplně proti univerzálnosti mého systému. Osobně dávám přednost rozdělení podle práv. Ovládání přes dva systémy je velice nepraktické.

Pokud jde o uživatelskou stránku, tak v té Conto také zaostává. Mimo samotného konceptu ovládání je zde mnoho položek a systém hlášení je nepřehledný. V mém systému jsem vytvořil taková hlášení a ovládání, aby byla co nejjednodušší. S tímto je pak spojená i grafická stránka, která je v porovnání s mojí aplikací, kde je grafika sice jednoduchá, ale atraktivní a účelná, řekl bych až nevyhovující.

Věřím, že aplikace Conto má jistě své místo na trhu. Z pohledu systému, který jsem navrhl je však pro mou cílovou skupinu nevhodná a myslím, že má aplikace lépe pokrývá potřeby provozu stravovacích zařízení.

1.6.4 Systém MRP

U tohoto software se dostáváme mezi zástupce klasických programů, které se v oblasti restaurační systémů příliš nevyskytují. K navrhované aplikaci se MRP přibližuje v myšlence jednoduchého použití na libovolném počítači. Je zde však omezení na operační systém Windows.

Při porovnání samotných vlastností je zde velký rozdíl v oblasti skladu. Má aplikace obsahuje skladovou obsluhu na úrovni hlídání stavů skladů. Podobné funkce však MRP obsahuje pouze okrajově na úrovni skládání nabídky. Správa skladového množství však není k dispozici.

Po uživatelské stránce jsou oba systémy porovnatelné. Je zde zachována dobrá ovladatelnost a přehlednost. Vytknout však lze grafickou stránku. MRP je oproti mé aplikaci strohý a graficky nezajímavý.

Při celkovém pohledu a porovnání systémů je tato aplikace celkově slabší, než můj systém. V ovladatelnosti se oba systémy podobají, ale všechny ostatní oblasti nejsou u MRP z mého pohledu úplné. Předchozí porovnávané aplikace měly některé vlastnosti srovnatelné a objevovaly se i části, kde byly lepší, ale zde se jedná o program, který má všechny své funkce srovnatelné, nebo horší než má aplikace.

1.6.5 Systém KeEXPRESS

KeEXPRESS je spíše než systém pro správu stravovacích zařízení účetním systémem. Je zde patrné nepřímé zaměření na gastronomii a tím se KeEXPRESS dostává v některých oblastech za mou aplikaci. Můj systém je čistě pro podniky se zaměřením na stravování, zatímco zde je potenciál k účetnictví a dokonce ubytovací činnosti.

V pokrytí oblastí je aplikace KeEXPRESS jistě dostačující. Ve srovnání s mojí aplikací je systém celkem podobný. Umožňuje sestavování nabídky ze skladových surovin, přímé vyskladňování, evidenci objednávek a další. Vše je postaveno na podobném principu jako v mém systému. Je zde ovšem nutnost zakoupit modul pro restauraci a modul pro sklad. V mé aplikaci mám tyto moduly společně.

Z pohledu uživatelské stránky je tento software příjemný. Od mé aplikace se poměrně liší v možnosti volby položek do objednávek a dalších uspořádání ovládání. To je však dáno již samotným principem. Online aplikace bude vždy trochu jiná než systém pro dotykové monitory. Myslím si však, že z pohledu uživatele se bude jednat spíše o příjemnou změnu, než o krok špatným směrem. Administrativní stránku ovládání mám však zpracovanou přehledněji a pro uživatele logičtěji.

Pokud oba systémy postavím vedle sebe a porovnáme je jako celek, tak si myslím, že jsou v základních principech srovnatelné. Aplikace KeEXPRESS je v některých ohledech obsáhlejší.

1.6.6 Menu55

Menu55 se svou myšlenkou nejvíce přibližuje mé aplikaci. Jedná se o jedno z mála online řešení a jediné použitelné, na které jsem narazil při průzkumu českého trhu. V porovnání s mou aplikací je však méně propracovaný sklad a systém obsahuje velké množství zbytečných vlastností. Na druhou stranu je aplikace Menu55 plně responzivní, což u mé aplikace nebylo záměrem. I přes to, že má aplikace na mobilních zařízeních funguje, tak pro ně nebyla optimalizována.

Obsah systému a pokrytí základních potřeb stravovacích zařízení je na tom však trochu hůře. Systém je sice velice jednoduchý, ale ne úplně kompletní. Chybí skladová část a přehledy, které jsou v mé aplikaci jedním ze základních prvků a pro provoz stravovacích zařízení klíčové. Tvorba nabídky také zůstává oproti mému systému pozadu a s tím je spojená celá administrace podniku.

Při pohledu na obsah systému je pak patrné, že má aplikace je na tom lépe v oblasti podpory restaurace jako celku. I přes to, že menu55 je velice zajímavým počinem, tak není pro kompletní provoz středního podniku úplně ideální. V tomto ohledu je můj systém propracovanější.

2 Praktická část

2.1 Cílová skupina vytvářeného systému

Systém, který jsem postavil, nese pracovní název RESYS, což je zkratka slov Restaurační SYStém. RESYS jsem vytvořil pro provozovatele všech uvedených kategorií stravovacích zařízení, které jsem uvedl v teoretické části této práce. Jedinou výjimkou jsou velké komplexy, které se skládají z více různých kategorií provozoven. Optimální je nasazení v zařízeních, kde se počítá s kapacitou do 150 míst a obsluhou do 20 zaměstnanců. Aplikace by samozřejmě zvládala i větší podniky, ale původním cílem bylo vytvořit levný a jednoduchý systém, který zvládne základní potřeby jednotlivých stravovacích zařízení. Z tohoto důvodu aplikaci chybí například modul pobočky, který by umožňoval evidenci více provozoven.

Cílovou skupinu jsem si vybral z důvodu špatného pokrytí tohoto segmentu restauračními systémy a je zde tedy největší prostor pro zlepšení a prosazení systému na trhu. Důvody jsou celkem jednoduché. Malé a střední podniky nemají většinou finanční prostředky na velké systémy. Někdy také není v podniku dostatek prostoru na samotné umístění pokladny. Mimo to tyto podniky potřebují pouze základní správu svého provozu a mnoho modulů velkých systémů je pro jejich provoz nadbytečné.

2.2 Charakteristika aplikace

Důvody, proč jsem si vybral právě tuto práci a mou cílovou skupinu, jsem již zmínil v textu dříve. Zde se již budu zabývat samotným systémem, jeho fungováním a celkovým smyslem mé práce. Hlavním cílem mé práce bylo sestavit takovou aplikaci, která bude pokrývat všechny základní potřeby provozu libovolného stravovacího zařízení, bude možné ji využívat na většině dostupných počítačů a její zprovoznění nebude časově ani finančně náročné.

Jako asi v každé aplikaci, tak i v té mé, jsou rozhodující dva faktory. Prvním je to, co samotná aplikace vlastně umí a druhým jsou lidé, kteří s aplikací pracují. Oba aspekty je pak možné dále větvit na menší a detailnější celky, ale základ je vždy stejný. Nejprve se pokusím popsat můj systém jako celek. Jeho fungování a faktory, které jej ovlivňují, jsou popsány níže.

Stravovací zařízení obecně obsahuje poměrně rozsáhlou problematiku, do které vstupuje mnoho aspektů. V tuto chvíli se nebudu zabývat kategoriemi stravovacích zařízení, ale pokusím se vytvořit obecný přehled.

Prvním ze společných faktorů, který stravovací zařízení ovlivňují, jsou lidé, kteří se z pohledu této problematiky člení na zákazníky a obslužný personál. Zákazníci od podniku čekají určité služby, kvůli kterým provozovnu navštěvují. Na zákazníka ještě přímo působí dva faktory. Jednak jsou to poskytované služby restaurace, jako je kvalita a rychlost obsluhy, kvalita nabídky, vzhled podniku a tak dále. Druhým je konkurence, kterou definuji níže.

Zaměstnanci jsou druhou částí tohoto faktoru. Z pohledu systému je důležité jim poskytnout jen omezený přístup do aplikace podle práv a hlavně evidovat jejich práci, aby nedocházelo k případným nesrovnalostem ve stavu skladu a finančním ztrátám.

Druhým společným aspektem je konkurence, která přímo ovlivňuje nutnost zvyšování kvality a dává zákazníkům lepší představu o tom, jakou kvalitu mohou poskytované služby mít, jaké ceny mohou být za jaké produkty nabídky a podobně. Restaurační systém do konkurence přímo nezasahuje, ale svým provozem dává konkurenční výhodu.

Vedlejším aspektem je pak povinnost evidovat výdaje a zisk pro daňové účely. Na tuto část se můj systém přímo nezaměřuje. Jsou v něm k dispozici pouze přehledy objednávek, které reprezentují zisky.

Samozřejmě není možné opomenout ani samotný podnik. Z obecného pohledu zde hraje roli jeho rozložení, které je možné velice dobře modelovat mou aplikací, dále pak obsluha, která pracuje v jednotlivých částech podniku a je reprezentovaná podle rolí a v neposlední řadě také samotné umístění počítače, na kterém aplikace běží a případně dalších komponentů, které jsou s aplikací spojené.

2.3 Proces vývoje

Při vývoji mé aplikace jsem využíval spirálový přístup. Vzhledem k tomu, že jsem systém vyvíjel po modulech, mi přišel právě tento postup vhodnější. Nejprve jsem vytvořil základní přehled v podobě sepsání požadavků, vytvoření CRC štítků, vytvoření seznamu potencionálních uživatelů a dalších akcí, které jsou nezbytně nutné pro počátek tvorby. Následně jsem si vzal vždy jeden z chystaných modulů, ten jsem analyzoval, zhodnotil a zkontroloval s provozovateli. Následně jsem jej naprogramoval a připravil podklady pro další modul. Tímto způsobem jsem postupně systém rozšiřoval a zkvalitňoval.

Při vývoji jsem využíval UML jazyk pro tvorbu diagramů a přehledů. Na základě postupné analýzy systému jsem následně vytvořil samotnou aplikaci za použití objektově orientovaného programování a jeho koncepce, která obsahuje objekty, zapouzdření, skládání, delegování a tak dále. Samotný návrh aplikace byl poměrně rozsáhlý, tak jej nyní představím po částech tak, jak šly za sebou.

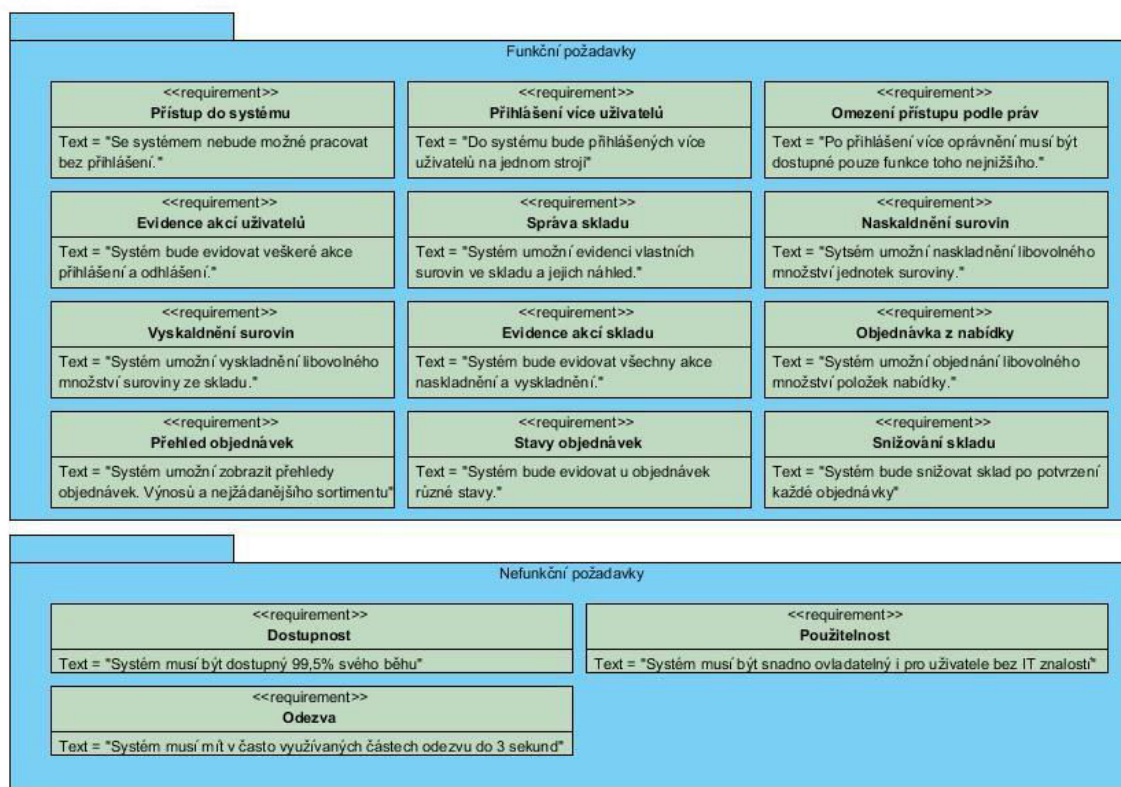
2.3.1 Požadavky na funkce systému

Požadavky na systém jsem skládal na základě konzultací s provozovateli restauračních zařízení a také podle vlastních zkušeností. Jak bývá dobrým zvykem, tak jsem požadavky rozdělil na dvě části a to funkční a nefunkční. Nefunkční požadavky zachycují aspekty samotného provozu systému, zatímco ty funkční se vztahují k možnostem, které aplikace nabízí.

Vzhledem k charakteru provozu stravovacích zařízení bych mezi nejdůležitější nefunkční požadavky řadil dostupnost, odezvu a použitelnost. Pokud bude systém nedostupný, tak dojde k ochromení celého provozu podniku. V případě špatné odezvy dojde ke zpomalení obsluhy, a tím k nespokojenosti zákazníků. Posledním z prioritních nefunkčních požadavků je použitelnost ve smyslu snadného pochopení aplikace a logického uspořádání často používaných operací tak, aby nebyla omezena rychlost provozu tím, že obsluha bude některou volbu dlouho hledat.

U funkčních požadavků považuji za důležité všechny, ale přece jen bych zde několik z nich postavil o něco výše. Jednak to je správa skladu, která je nutná k uhlídání zásob a kontrole zaměstnanců, dále pak požadavky týkající se objednávek, kde je riziko špatného výpočtu ceny a nepřesného vyskladnění suroviny. Obě rizika by vedla ke ztrátě zisku a to je pro restaurační systém logicky nepřípustné. Jako třetí část požadavků, které jsou prioritnější, jsou ty, které se týkají přihlašování. Špatné přihlášení uživatele nebo zobrazení modulů, kam nemá mít uživatel přístup, by dávalo prostor k potencionálnímu napadení systému.

Přehled prioritních funkčních i nefunkčních požadavků je znázorněn v grafice níže. Prioritu požadavků své práce jsem vybíral na základě konzultací s provozovateli stravovacích zařízení. Nebyl to těžký výběr, protože názory na důležité aspekty systému byly u provozovatelů jednotné. Kompletní přehled všech požadavků aplikace se nachází na konci této práce jako příloha. Požadavků by však bylo možné definovat i více, ale tím bych se dostal ke tvorbě mnohem obsáhlejší aplikace, než bylo zadáním této práce.



Obrázek 9 – Přehled prioritních požadavků systému

2.3.2 Use Case Diagram

Po vyjasnění všech požadavků, které by mohl systém na svůj provoz mít, jsem vytvořil návrh Use Case Diagramu. Ten mi objasnil, jaké uživatelské role budou v systému figurovat, a jaké požadavky budou pokrývat. Při vytváření diagramu jsem se zaměřil pouze na klíčové oblasti. Samozřejmě, že některé z případů užití by se mohly dále větvit. Z mého pohledu je však podstatné představit atypické případy a není tedy podle mne nutné vypisovat například u všech položek se správou modulu případy jako editace záznamu, smazání záznamu a další opakující se akce.

Určení přesných uživatelských rolí a jejich možností je u mé aplikace poněkud obtížnější. Každé stravovací zařízení má své pevné pracovní pozice, které je možné pojmenovat univerzálně a rozdělit jim přístupy do systému. Jenže jednotlivých kategorií podniků je poměrně hodně a každá kategorie má požadavky trochu jiné. Vzhledem k tomu, že jsem stavěl univerzální systém, tak jsem potřeboval mechanismus, který umožní vytvořit uživatelské role vedoucímu nebo majiteli provozovny, přesně podle specifických potřeb a požadavků daného podniku.

Pro potřeby návrhu systému jsem vytvořil čtyři uživatelské role a samozřejmě roli systém, která je nezbytná²². Nejvyšším oprávněním pro správu aplikace je administrátor. Ten má k dispozici všechny funkce aplikace. Pod tímto oprávněním je role vedoucí, která spravuje

²² Uživatelská role systém by podle mého měla být v každém návrhu, protože ne všechny operace vykonává vždy některý z uživatelů.

jednotlivé části týkající se samotného provozu systému. Další dvě oprávnění jsem určil pouze pro ukázkou a nazval jsem je vyšší oprávnění restaurace a vyšší oprávnění kuchyně. V Use Case Diagramu figuruje ještě jeden aktér, kterého jsem nazval uživatel a vyjadřuje nepřihlášeného návštěvníka systémů. Po přihlášení jsou z tohoto aktéra odvozeni aktéři ostatní, podle toho o jakou roli se jedná.

Aktér uživatel má ve svých případech užití přihlášení a odhlášení. Přihlášení se následně dělí na dvě varianty. Buď se jedná o první přihlášení uživatele, nebo je již do systému někdo přihlášen a další uživatel se dopřihlašuje navíc²³. S přihlašováním a odhlašováním potom souvisí i uložení záznamu o této činnosti, který je v tomto případě rozšiřující a přímo jí vykonává aktér systém. Mimo to je zde ještě požadavek, aby aktér systém provedl kontrolu a omezil dostupné moduly podle toho, jaký uživatel je přihlášený nebo se přihlásil jako další.

Následujícím aktérem je administrátor, který má stejné požadavky jako aktéři nižší, ale navíc má přístup do správy modulů systému, aby mohl aktivovat a deaktivovat jednotlivé moduly. Nižším oprávněním je potom vedoucí. Ten má k dispozici veškerou správu jednotlivých částí systému jako například správu dodavatelů, uživatelů, rozložení provozovny, sestavování nabídky, správu skladu a tak dále.

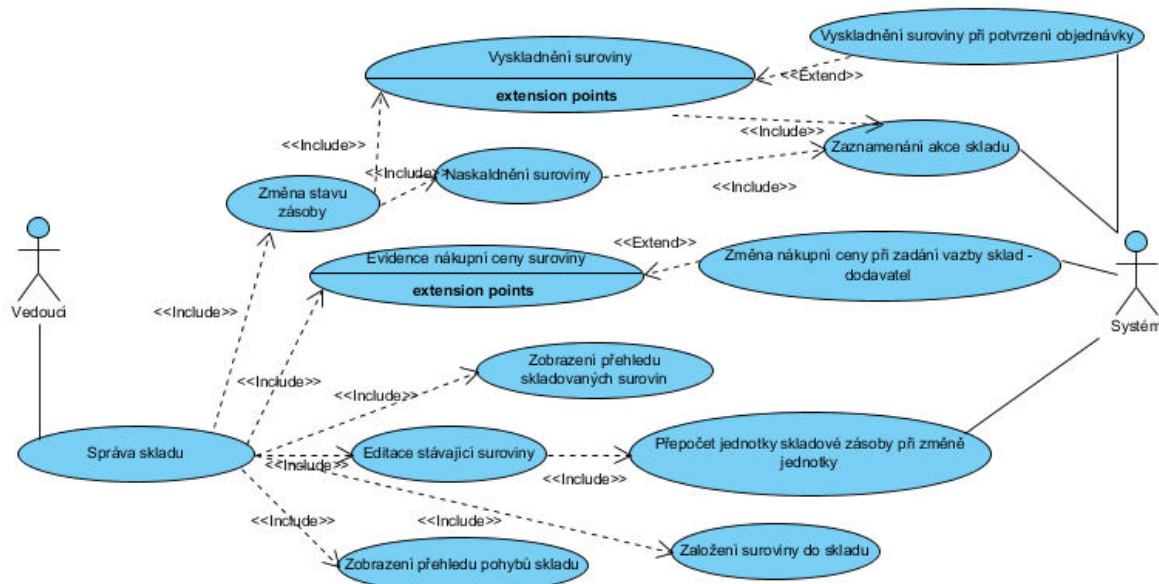
Pod vedoucím jsou pak dva již zmiňovaní aktéři, kteří jsou pojati univerzálně a slouží pro ukázkou, jak by systém mohl být optimálně nastaven a jak bude pravděpodobně využíván. Jako první představím vyšší oprávnění restaurace, které zahrnuje obsluhu hostů v podobě číšníků, barmanů a podobných rolí. Ti budou se systémem pracovat nejčastěji a mají požadavky na správu objednávek a přehled skladových zásob. Druhým aktérem je pak vyšší oprávnění skladu, které má přístup do skladových zásob a přehledu objednávek. Pod tímto aktérem mohou být kuchaři, sommeliéři a další, kteří do systému nemusí přímo zasahovat.

Tímto jsem objasnil aktéry a jejich základní smysl a nyní se budu věnovat některým vazbám mezi případy užití. Jako první se zaměřím na správu skladu. Základní požadavek na správu skladu obsahuje tři rozšíření. Prvním je zobrazení skladových zásob pro určité aktéry, dále pak možnost naskladnění a vyskladnění, které mají další rozšíření a to požadavek na zaznamenání operace do databáze, které vykonává systém.

Jako druhý zde máme požadavek na správu objednávek, která v sobě obsahuje rozšíření v podobě možnosti zobrazení přehledu objednávek podle stavu. Navíc je zde požadavek na přiřazení objednávky ke stolu zařízení a možnost správy stavů objednávek jako storno, potvrzení, rozdělení a další běžně používané, do kterých je možné objednávku přeradit. K akci potvrzení je ještě vázaný požadavek na vyskladnění ze skladu v případě potvrzení objednávky. Tento požadavek opět vykonává automaticky přímo systém.

²³ V rámci možnosti multipřihlášení.

Kompletní Use Case diagram, který jsem popsal, je v příloze na konci této práce a jak již bylo zmíněno, tak v zájmu zachování určité míry přehlednosti obsahuje pouze klíčové nebo nějakým způsobem netypické případy užití. Pro představu, jak by mohl vypadat detailní Use Case Diagram, jsem si vybral část z kompletního diagramu a rozšířil jsem ji o všechny požadavky, které se mohou u této části vyskytnout.



Obrázek 10 – Ukázka rozšířeného Use Case diagramu z pohledu vedoucího na sklad

2.4 Rozvržení systému

Dříve, než představím další části vývoje své aplikace, tak určím logické rozvržení systému. Vzhledem k tomu, že aplikace, kterou jsem vytvořil je poměrně rozsáhlá, tak jsem si jí musel rozdělit na menší celky. Nutnost rozdělení aplikace podle modulů, které se starají o jednotlivé části běhu systému, se projevila právě po tom, co jsem vytvořil případy užití, a chystal jsem se na návrh databáze. Z tohoto důvodu uvádím rozdělení i na tomto místě mé práce.

Univerzální aplikace nemusí nutně znamenat, že vždy budou dostupné všechny úplně všechny funkce, ale systém bude možné upravit tak, aby přesně vyhovoval potřebám konkrétního stravovacího zařízení. Z tohoto důvodu jsem celou aplikaci postavil na principu modulů, které lze snadným způsobem aktivovat nebo deaktivovat dle potřeby.

Díky tomuto mechanismu je možné systém uzpůsobit podle specifických požadavků každého zařízení. Rozdělení na moduly je v některých ohledech podobné jako u restauračních systémů, které jsem testoval v rámci tvorby teoretické části této práce. Rozdíl je pak to, že moduly jsou vždy dostupné, ale v případě potřeby se dají vypnout²⁴.

²⁴ Testované aplikace často nutí k dokoupení jednotlivých modulů, nebo některé vůbec neobsahují.

Celý systém jsem rozdělil na celkem 11 samostatných modulů, kde každý z nich vykonává svou specifickou činnost. Vypnutí modulu je možné prakticky kdykoliv a systém tím nijak nebude strádat. Musí však být zachována podmínka na zachování dat. I když je modul vypnut, tak musí mít alespoň základní data, která jsou dostupná při instalaci. Jinak není možné zaručit bezproblémový chod celého systému.

Dělení na moduly však není jediné. Systém je dále dělený na pět větších logických bloků, které není možné deaktivovat nebo s nimi podobným způsobem pracovat. Tyto bloky slouží spíše pro lepší orientaci a odstínění celé aplikace. Funkce jednotlivých modulů a rozložení ještě detailně zmíním v dalších částech této práce.

Nyní se budu věnovat samotnému dělení. Jako první blok představím správu uživatelů. Ta se stará o kompletní uživatelské záležitosti jako jsou přihlašování, evidence uživatelů, práva, evidence akcí uživatelů a další. Tento blok obsahuje tři samostatné moduly. První obsahuje samotné rozhraní modulů a slouží pouze pro jejich aktivaci a deaktivaci. Sám o sobě je dostupný pouze administrátorovi a nikomu jinému. Jako druhý zde mám modul práva, který se stará o sestavování jednotlivých oprávnění systému. Jako třetí je zde modul uživatelé, kde jsou evidováni všichni uživatelé systému s jejich přístupy a základními údaji.

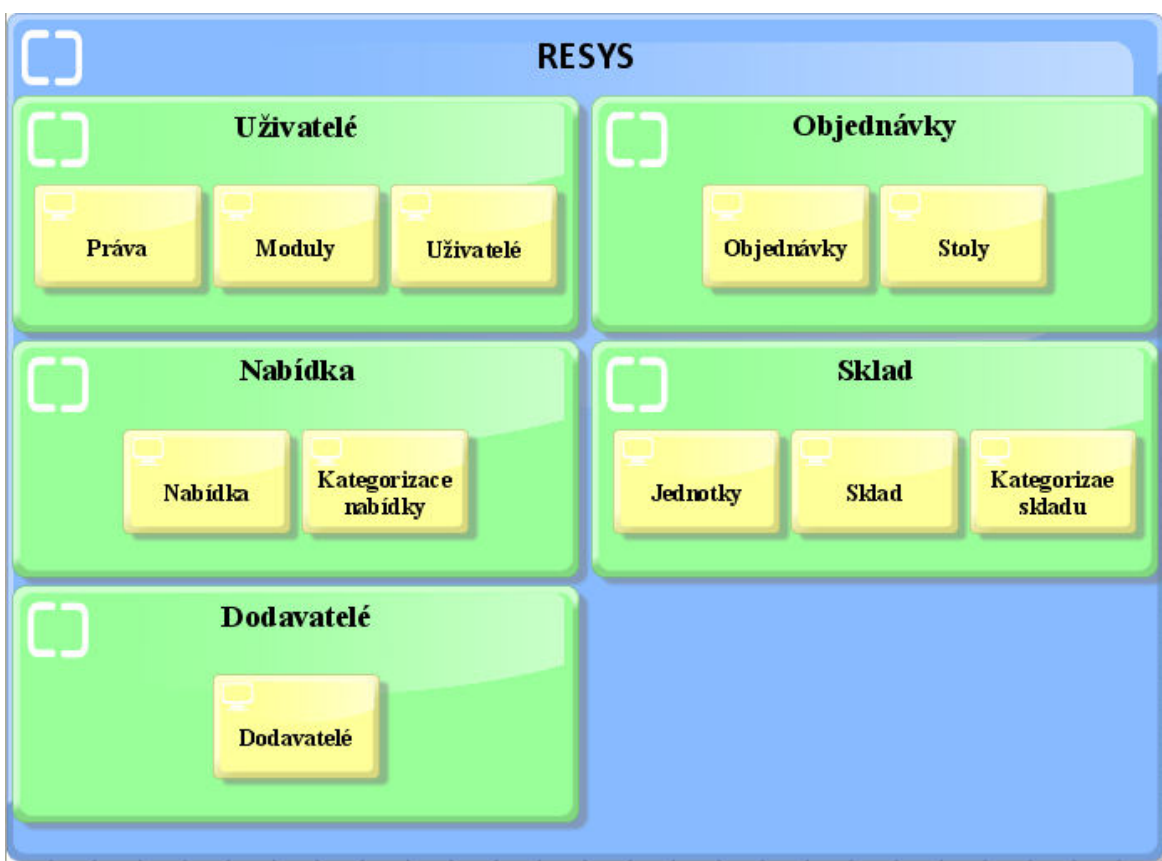
Druhý blok sdružuje akce kolem objednávek. Obsahuje a zpracovává vytvořené objednávky, jejich přiřazení ke stolům, navázání na nabídku a správu rozložení provozovny. Jedná se tedy o klíčovou část provozu, která je složena ze dvou samostatných modulů. Prvním je modul stoly, který je rozložením prostoru a druhým je modul objednávky, který eviduje veškeré prodané položky z nabídky a komplexně vše, co je od objednávek ve stravovacím zařízení očekáváno.

Třetí blok obsahuje vše potřebné kolem nabídky provozovny. Stará se o určení, kolik jednotek jaké suroviny bude v dané položce obsaženo a za jakou cenu bude daná položka prodávána. Tento blok vykonává funkci spojky mezi objednávkou a skladem a je rozdělen na dva moduly. Prvním je samotná nabídka, která se skládá ze surovin skladu a váže se na objednávku a druhým modulem je kategorizace nabídky, kde je možné určit dělení nabídky.

Čtvrtý blok se stará o organizaci skladu a všeho kolem něj. Jsou zde obsažené jednotky využívané v systému a veškeré informace o surovinách, které má daná provozovna k dispozici. Mimo to jsou zaznamenané pohyby na skladu a sklad je samozřejmě kategorizován. Tento blok se pak skládá ze tří samostatných modulů. První se stará o kompletní správu jednotek, které jsou dostupné v systému. Druhý slouží pro kategorizaci skladovaných položek a třetí je samotný sklad, který eviduje veškeré suroviny, které jsou uloženy ve skladu.

Poslední logický blok je zároveň i nejmenším a stará se o dodavatele. Slouží pouze pro evidenci dodavatelů a způsob, jakým je u nich možné objednat. Mimo to umožňuje samotné objednání vystavením CSV souboru, XML souboru, nebo zasláním objednávky na e-mail. Blok sám o sobě obsahuje pouze jeden modul. Samostatně jsem tuto část dal z důvodu, že zde očekávám do budoucna další vývoj.

Pro lepší přehled jsem vytvořil jednoduchý diagram, který znázorňuje rozdělení mé aplikace na bloky a moduly. Modré pole znázorňuje celý systém RESYS, zelená pole jsou pak logické bloky a žlutá pole jsou samotné moduly.



Obrázek 11 – Schéma rozvržení aplikace na bloky a moduly

2.5 ER diagram

Po sestavení požadavků, vytvoření případů užití a rozvržení systému jsem se zaměřil na ER diagram a databázovou část. Vzhledem k tomu, že celý model je až příliš obsáhlý, tak jej popisuji podle logických bloků, které jsem zmínil v kapitole výše. Náhled na celý model je k dispozici v přílohách této práce.

Pokud se na model podívám obecně, tak zde mám několik společných rysů celé databáze. Většina z těchto rysů je spojena s normálními formami. Jedná se například o fakt, že až na tabulky, které jsou čistě vazební a tabulku s přihlašovacími údaji, jsem všude použil specifický primární klíč, který je automaticky generován²⁵. Všechny relace a jejich atributy jsem rozložil do maximální možné míry.

Stejně tak datové typy a jejich rozsahy jsem vybíral tak, aby nedocházelo ke zbytečnému prostoru pro napadení databáze nebo alokaci zbytečně velké části paměti. V některých případech jsem si přeci jen rezervu pro jistotu nechal, kdyby bylo třeba vložit větší hodnotu, než jsem očekával při návrhu. Mimoto jsem se snažil dodržet přesné vazby mezi jednotlivými relacemi. Kardinalita a parcialita je velice důležitá a při tvorbě návrhu databázové části jsem na ní kladl zvláštní důraz.

2.5.1 Databáze bloku uživatelé

Jako první představím databázovou část, která pokrývá blok uživatelů. Jak již bylo řečeno, v tomto bloku jsou tři moduly, ale tabulek je zde dohromady šest. S tím, že tabulky moduly, práva, uživatelé, log uživatelů a login pokrývají hlavní datovou část bloku a tabulka modul_mn_prava má na starosti vazby práv na moduly.

Tabulka moduly obsahuje informace o modulech systému. Je složená z identifikátoru modulu, jeho názvu a příznaku, jestli je modul aktivní nebo ne. Podobně na tom je i tabulka práva, která nese informace o všech vytvořených právech v systému. Mimo atributů, které má tabulka moduly je zde navíc možnost vložení popisu práva. Třetí datová tabulka je nazvaná uživatel. Obsahuje přehled všech uživatelů systému, jejich jména a základní kontaktní údaje.

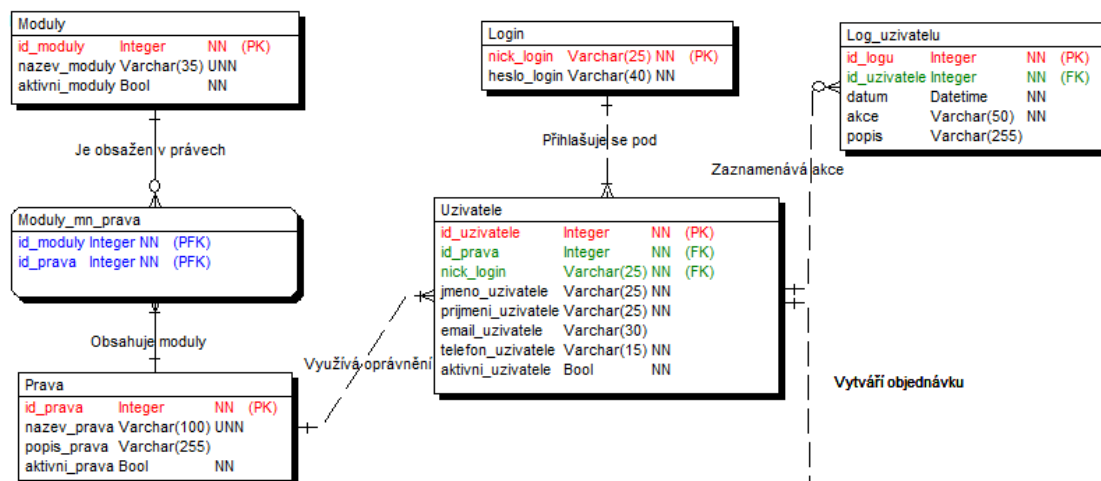
Tabulka uživatelů je vázána na tabulku login, která obsahuje přihlašovací jméno a heslo uživatele. V tomto případě je vazba provedena na základě přihlašovacího jména, které v mém systému musí být jedinečné. Samotná vazba vyjadřuje stav, kdy uživatel má jedny přihlašovací údaje a přihlašovací údaje musí patřit minimálně jednomu uživateli, ale teoreticky by mohli patřit i více uživatelům. Tuto možnost jsem si však v systému nechal pouze pro případné rozšíření. Reálně však využívám vazbu 1:1.

Další tabulkou tohoto bloku, kterou jsem ještě nepopsal, je log uživatelů. Tato tabulka je určena pro veškeré akce, které bude uživatel v systému provádět. Mimo základního identifikátoru obsahuje informace o tom, jaký uživatel akci provedl, kdy došlo k akci, jaká akce byla provedena a popis této akce. Vesměs se jedná o klasický log, který je nutné pro případné zjištění, co se v systému dělo. Z toho logicky vyplývá, že akce bude vždy náležet jednomu konkrétnímu uživateli, ale uživatel bude moci mít více různých akcí. Nebo taky žádnou, pokud uživatel bude jen založen, ale ještě se například nepřihlásí do systému.

²⁵ Za využití možnosti AUTO_INCREMENT

Z celého bloku nyní zbývají zmínit vazební tabulku mezi moduly a právy, vazbu mezi právy a uživateli a nakonec vazbu uživatelů na objednávky, které již spadají mimo tento blok. Začnu od vazební tabulky mezi moduly a právy. Vzhledem k tomu, že v mém systému může být jeden modul ve více právech a jedno oprávnění může nést více modulů, tak jsem zde využil vazby M:N a díky tomu bylo nutné vytvořit i vazební tabulku. Ta neobsahuje nic jiného, než primární klíče obou vázaných tabulek s tím, že právo musí mít vždy vybraný alespoň jeden modul, ale modul nemusí být přiřazený žádnému právu.

Z vazeb zde ještě mám jednu mezi uživateli a právy. Politika mého systému je taková, že každý uživatel musí mít pouze jedno oprávnění, ale oprávnění může náležet více uživatelům, nebo také žádnému. Nakonec je ještě třeba zmínit vazbu mezi celým blokem uživatelů a blokem objednávek. Vazba zde vyjadřuje skutečnost, že objednávky vytváří uživatel systému a to tak, že jeden uživatel může vytvořit více objednávek, ale objednávku vždy vytvoří právě jeden uživatel. Pro snadnější orientaci v problematice popsaného bloku přidávám část ER diagramu.



Obrázek 12 – ER diagram bloku uživatelé

2.5.2 Databáze bloku objednávky

Jako druhý je na řadě blok objednávky, který se skládá ze dvou modulů, ale celkem je rozložen do čtyř tabulek. Tabulky stoly a objednávky slouží pro uchování základních dat, tabulka objednavka_mn_stul je pak čistě vazební pro násobnou vazbu mezi stoly a objednávkami. Poslední tabulka zde je objednavka_mn_nabidka, která v mém prvním návrhu byla čistě vazební, ale při vývoji jsem pak zjistil, že bude nutné její rozšíření.

S popisem začnu u tabulky stoly. Ta obsahuje veškeré informace o rozvržení prostoru ve stravovacím zařízení. Kromě nezbytného identifikátoru zde máme atribut název stolu, který slouží pro uživatele, aby si mohl rozlišit stoly dle svého uvážení. Atribut pro určení aktivity je v tomto případě celkem jasný. Slouží pro dočasnou deaktivaci stolu bez nutnosti jeho smazání.

Trochu matoucí však mohou být další atributy tabulky. Ty slouží k určení polohy stolu vůči určenému prvku stránky. Pozice x , y určují pozici pravého horního rohu a šířka s výškou určují velikost pole, které znázorňuje stůl. Hodnoty jsou zadávány v pixelech, takže je mám určené jako celočíselné.

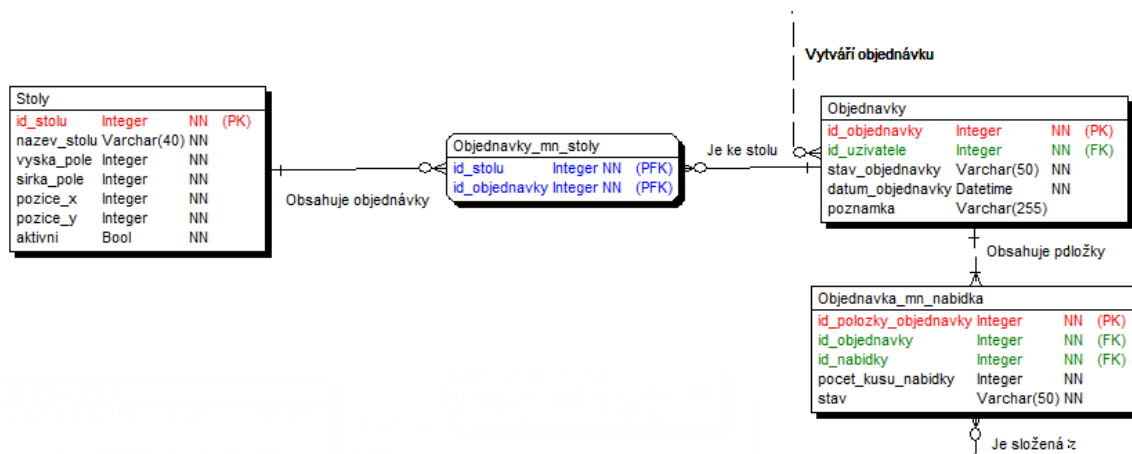
Vzhledem k tomu, že jsem chtěl podchytit situaci, kdy jedna objednávka bude moci být na více stolů a jeden stůl může mít více objednávek, tak jsem v aplikaci počítal s vazební tabulkou pro M:N vazbu. Stejně tak může být v systému objednávka bez stolu a stůl bez objednávky. Reálně však můj systém využívá pouze situaci, kdy objednávka může být jen u žádného nebo jednoho stolu a stůl může mít více objednávek. Tato část databáze slouží čistě pro možnost rozšíření.

Nyní se dostávám k samotné tabulce objednávek. Ta obsahuje atribut identifikátoru záznamu a identifikátoru uživatele, který objednávku založil. Tento identifikátor je cizím klíčem. Mimo to je zde atribut stav objednávky, který je pro rozlišení, zda je objednávka stornovaná, uhrazená nebo stále otevřená, atribut datum určující, kdy byla objednávka založena a atribut poznámka pro případné vyjádření od obsluhy k objednávce²⁶.

Aby bylo možné určit, co objednávka vlastně obsahuje, tak mám v návrhu tabulku `objednavka_mn_nabidka`. Ta byla původně pouze vazební, ale nyní slouží ke kompletnímu určení, co je v objednávce. Protože mohou být objednávány některé položky s časovým odstupem, tak jsem doplnil vlastní identifikátor, protože cizí klíče nemusí být nutně unikátní. Atribut počet kusů nabídky určuje množství z nabídky, které je v objednávce obsaženo. Vzhledem k tomu, že se jedná o počty kusů, tak zde stačí celočíselný typ. Trochu zavádějící zde může být atribut stav, který je i u samotné objednávky. Stav využívám v systému dva a nejedná se tedy o duplicitu. Stav v této tabulce určuje stav konkrétní části objednávky, tedy případ, kdy je v objednávce jedna část uhrazená a další ještě neuhrazené. Bez tohoto stavu by nebylo možné určit, jestli je již objednávka celá uhrazená nebo ne a co již bylo uhrazeno.

Nakonec představím samotné vazby. Je celkem jasné, že objednávka musí mít minimálně jednu objednanou položku z nabídky a položka objednávky se vztahuje pouze k jedné konkrétní objednávce. U vazby na nabídku, která je již mimo tento blok, je situace odlišná. Položka z objednávky se vztahuje vždy na jednu nabídku, ale nabídka nemusí být v žádné objednávce, ale stejně tak může být jedna položka z nabídky ve více objednávkách.

²⁶ Někdy je nutné zapsat k objednávce specifický požadavek hosta.



Obrázek 13 – ER diagram bloku objednávky

2.5.3 Databáze bloku nabídky

Tato část databáze ukládá informace o nabídce podniku. Celý blok se skládá ze dvou modulů a celkově čtyř tabulek. Klíčová zde je tabulka nabídka a vazební tabulka sklad_mn_nabidka, která určuje složení nabídky. Další dvě z tabulek slouží pouze pro možnost kategorizace nabídky.

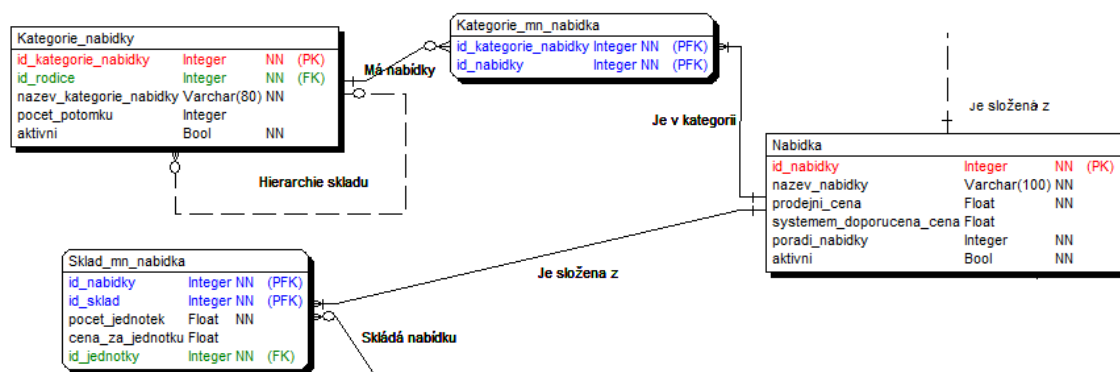
Se samotným představením této části databáze začnu právě u tabulky pro kategorizaci nabídky. V mé aplikaci jsem udělal pro kategorizace stromovou strukturu a tomu musí odpovídat i návrh databáze. Tabulka kategorizace_nabidky má z tohoto důvodu vazbu sama na sebe s tím, že rodič může mít libovolné množství potomků nebo žádného, ale potomek by měl mít pouze jednoho rodiče. V této části je však nutné uvažovat i nad samotným kořenem, který nemusí mít potomky. Vazba zde tedy znázorňuje fakt, že prvek nemusí mít žádného rodiče.

Tabulka kategorizace_nabidky není nijak složitá. Obsahuje jednoznačný identifikátor jako primární klíč, atribut pro název kategorie, informaci o počtu potomků a informaci o tom, zda je kategorie aktivní. Mimo to zde mám atribut id_rodice, který je cizím klíčem a určuje vazbu tabulky samu na sebe. Vzhledem k tomu, že mám v mém systému možnost zadat nabídku do více kategorií, tak zde mám i násobnou vazbu v podobě tabulky kategorie_mn_nabidka. Ta obsahuje pouze primární cizí klíče z obou vázaných tabulek a vazby s informací o tom, že kategorie nemusí být využita v žádné nabídce, ale může být i ve více a nabídka musí být minimálně v jedné kategorii.

Nyní již k samotné tabulce nabídky. Smyslem této tabulky je uchovávat informace o tom, co je součástí jídelního, nebo nápojového lístku. Tabulka obsahuje pouze přehledové, řekl bych však nosné informace. Mám zde atribut určující název položky v nabídce, její prodejní cenu, její systémem doporučenou cenu, určení pořadí položek v nabídce a informaci o tom, zda je položka nabídky aktivní. Ceny v této části jsem volil jako desetinná čísla. V českých podmínkách není sice desetinné číslo pro stravovací zařízení nutné, ale v případě plateb v eurech by se již jednalo o značný rozdíl a je tedy podle mne vhodné zachovat desetinné hodnoty.

Tabulka nabídky by však nebyla nic bez tabulky sklad_mn_nabidka, která určuje, jaké suroviny se v nabídce nachází a také jaké množství od dané suroviny. Tato tabulka původně vznikla jako vazební, ale časem se ukázalo, že je potřeba její rozšíření. Hlavním smyslem tabulky je určit, kolik jednotek od jaké suroviny je obsaženo v konkrétní nabídce. Tabulka si vystačí s primárním klíčem, který je složený z dvou primárních cizích klíčů, protože nemůže nastat situace, kdy bude surovina ve stejné nabídce dvakrát, nebo nabídka bude obsahovat dvě rozdílná množství jedné suroviny. Mimo cizích klíčů jsou v této tabulce obsaženy atributy určující počet jednotek dané suroviny v nabídce, o jakou jednotku se jedná a nákupní cenu jednotky.

Tento databázový blok je dále vázaný na blok skladu a to hned na dvou místech. Obě vazby však vychází ze stejné tabulky a to sklad_mn_nabidka. První vazba je na tabulku skladu, kde je zachycen stav, kdy skladová položka nemusí být v žádné nabídce, ale může být i ve více nabídkách najednou. Druhá vazba je na tabulku jednotek, kde je určeno, jaká jednotka bude u jednotlivých složek nabídky. S tím, že jedna položka nabídky může mít vždy jen jednu jednotku, ale jednotka může být u více položek z nabídky.



Obrázek 14 – ER diagram bloku nabídky

2.5.4 Databáze bloku skladu

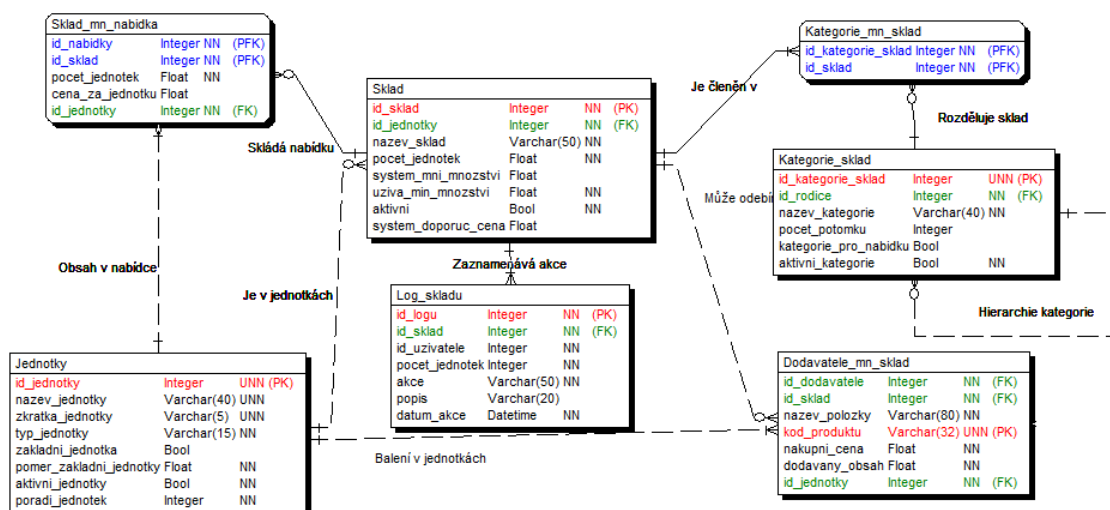
Předposlední a zároveň největší blok databázové části se stará o sklad. Blok obsahuje tři moduly a pět tabulek. Základní tabulkou bloku je tabulka sklad. Sklad je kategorizovaný podobně jako nabídka a to za použití tabulky kategorizace_sklad a vazební tabulky kategorie_mn_sklad. Dále jsou zde ještě dvě tabulky. První nese název Log_skladu a obsahuje záznamy akcí naskladnění a vyskladnění surovin. Druhá tabulka s názvem jednotky slouží nejen v tomto bloku, ale také v bloku nabídky a dodavatelů a nese měrnou logiku aplikace.

S popisem začnu u kategorizace skladu, která je podobná jako v případě nabídky. Je zde jedna tabulka, která obsahuje kategorie a je navázaná opět sama na sebe. Jediným rozdílem zde je atribut `kategorie_pro_nabidku`, který určuje, jestli je kategorie k dispozici při sestavování nabídky nebo se jedná o skladové položky, které nejsou přímo určené pro prodej²⁷. Vazby jsou zde stejné jako v případě kategorizace nabídky.

Tabulka `log_skladu` je v tomto případě obdobná jako u logu akcí uživatelů. Uchovávají se zde informace o tom, kdo naskladnil nebo vyskladnil jaké zboží, kdy tak učinil a je zde možnost doplnit informace proč k akci došlo.

Další tabulka slouží pro uchování jednotek. Vzhledem k tomu, že jsem vytvářel opravdu univerzální systém, tak mně přišlo vhodné umožnit i tvorbu vlastních jednotek a vztahů mezi jednotkami. Tabulka uchovává informace o názvu jednotky, používané zkratce a typu, který určuje, zda se jedná o obsahovou, hmotnostní nebo jinou jednotku. Logicky jsem zařadil i položku pro určení, zda je jednotka základní jednotkou pro daný typ a případně jaký je poměr převodu vůči základní jednotce. Hodnoty tabulky jsou využívány ve skladu, nabídce a u dodavatelů, protože zde se mohou jednotky lišit podle potřeby.

Nakonec k samotné tabulce `sklad`, která je jádrem tohoto bloku. Sklad je v mém systému určen tak, aby nerozděloval typy balení, ale pouze rozpoznával množství skladované suroviny. Primární klíčem tabulky je uměle vytvořený atribut `id_sklad`. Cizím klíčem je `id_jednotky`, který určuje v jaké jednotce je surovina skladována. Další atributy jsou název, který pomáhá v identifikaci pro uživatele, počet jednotek, který udává aktuální stav skladu, systémové minimální množství a uživatelské minimální množství, kde je hodnota, kterou hlídá systém vůči aktuální skladové zásobě. S tím, že uživatelskou si určuje uživatel sám podle svého uvážení a systémovou určuje systém podle odbytu.



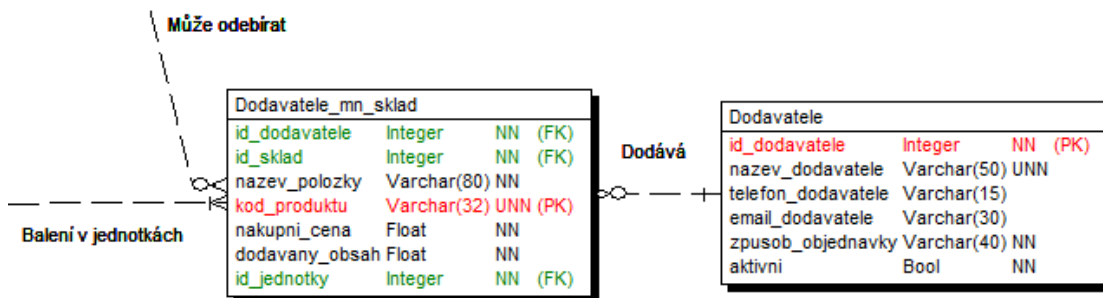
Obrázek 15 – ER diagram bloku skladu

²⁷ Ve skladu mohou být i sanitární potřeby, nádobí, přístroje a další podobné položky.

2.5.5 Databáze bloku dodavatelů

Poslední blokem, ke kterému chybí popsat databázovou část je blok dodavatelů. Ten je ve své podstatě velice jednoduchý. Obsahuje pouze jeden modul a dvě databázové tabulky. První tabulka dodavatelé obsahuje informace o všech dodavatelích, které mohou dodávat zboží do provozovny. Tato tabulka obsahuje pouze základní informace jako je název dodavatele, telefonní a e-mailový kontakt a informaci, zda je dodavatel aktivní nebo ne. Mimo to je zde atribut způsob objednávky, který říká, jakým způsobem dodavatel přijímá objednávku na své zboží. Ta mu může být zaslána e-mailem, vystavená v podobě XML na server, zaslána ve formátu CSV souboru, nebo do CSV pouze vygenerována.

Druhou tabulkou bloku je vazební mezi dodavateli a skladem. Zde vycházím z předpokladu, že nějaká položka skladu nemusí být dodávána žádným dodavatelem, ale může být dodávána i od více dodavatelů. Stejně tak dodavatel nemusí dodávat žádné zboží, ale může dodávat více druhů zboží. Samotná vazební tabulka nese název dodavatele_mn_sklad a záznamy v ní jsou určeny pomocí čárového kódu produktu. Mimo to nese atributy, kde je uveden název produktu, jeho nákupní cena, obsah ve kterém je produkt dodáván a identifikátor jednotky, ve kterém je určen obsah. Zde totiž nastává situace, kdy ve skladu máme jednotku určenou v litrech, ale dodávaný obsah je třeba 500 mililitrů. Je tedy třeba určit reálné hodnoty a ty mezi sebou převést tak, aby došlo k získání skutečné hodnoty skladu.



Obrázek 16 – ER diagram bloku dodavatelů

2.6 Složky a soubory

Pro úplnost ještě představím stromovou strukturu mého systému. Protože se jedná o velký systém a mám v plánu jej rozšiřovat i mimo rámec této práce, tak jsem navrhl a vytvořil strukturu, která mi pomůže při orientaci v souborech a složkách i při dalším vývoji.

Jednotlivé složky a soubory vychází vždy od kořenového adresáře serveru. Zde jsem ponechal jeden soubor index.php, který slouží pouze jako vstupní brána do editoru. Z tohoto souboru se časem může stát prezentace provozovny, která bude přístupná všem návštěvníkům stránek.

V této úrovni je dostupná složka grafika a editor. Adresář grafika jsem nechal prázdný, protože se bude jednat o grafiku samotné prezentace. Složka editor pak obsahuje administrativní část celého systému. Z toho důvodu probíhá přihlášení do systému na adrese www.vasedomena.cz/editor. Po otevření této složky se zobrazí celkem rozsáhlá struktura, která obsahuje podsložky generované, grafika, instalator, java, konfigurace a moduly. Mimo to zde mám základní soubory pro chod aplikace.

Jedná se o soubory `hlavicka.php`, který obsahuje hlavičku stránky včetně metatagů a dalších náležitostí, `menu.php`, kde je hlavní rozcestník aplikace a logika přístupů, `index.php`, který je hlavním přístupovým bodem do systému a obsahuje vyhodnocení přihlášení a příkazů z menu a nakonec `paticka.php`, kde je konec stránky a prostor pro doplnění patičky.

V této úrovni je několik dalších adresářů. Jako první je adresář generované, který souží pro ukládání CSV a XML souborů pro objednávky a na serveru je volně dostupný. Další je grafika, kde jsou umístěny CSS soubory se styly aplikace. Celé styly mám rozdělené do dvou souborů a to `layout.css`, kde jsou veškeré styly kromě těch, které využívá jQuery. Styly pro jQuery zobrazení pak jsou ve stejnojmenném souboru tedy `jQuery.css`. Tento adresář obsahuje ještě jeden podadresář, který nese název `obrazky` a je pro uložení obrázků používaných v administrační části.

Následuje složka instalator, kde je jediný soubor `index.php`. Ten je určen pro naplnění systému daty, která jsou nezbytná pro jeho základní fungování. V souboru je jednoduchý mechanismus, který dokáže vložit potřebné tabulky a jejich data do databáze, která však musí být předem založena.

Na stejné úrovni jako instalátor jsou pak složky java a konfigurace. Složka java obsahuje soubory s JavaScriptovými kódy. Soubory jsou rozloženy zvlášť podle funkce, kterou vykonávají, aby bylo zobrazení přehledné. Ve složce je například soubor `vyberNabidky.js`, který se stará o zobrazení více položek skladu při jejich výběru pomocí multiselect boxu.

Další adresář konfigurace obsahuje soubory `TMySQLKonektor`, který slouží pro správu třídy, která realizuje připojení k databázi a vykonávání dotazů, `jadro.php`, kde je mechanismus pro includování souborů a jako poslední je zde `konfigurace.inc`, která obsahuje konstanty pro nastavování systému.

Posledním adresářem této úrovně je adresář moduly. Ten obsahuje jednu složku pro každý modul obsažený v systému. Ve složce modulu je pak zpravidla minimálně 7 souborů, které jsou velice podobné a to jak názvem, tak strukturou²⁸. Mimo to je vždy obsažen soubor pojmenovaný `obsluhaNazevModulu.php`, který slouží pro hlavní obslužnou logiku modulu. Dále pak soubory s názvy `TNazveModuluModel.php`²⁹, který je modelem daného modulu, `TNazveModuluKontroler.php`, který je kontrolerem daného modulu a

²⁸ Vzhledem k tomu, že využívám MVC model, tak každý modul obsahuje minimálně třídu modelu, kontroleru a pohledu s rozhraními.

²⁹ Velké T na začátku označuje třídu

TNazveModuluPohled.php, který obsahuje funkce pro zobrazení informací, která modul spravuje.

Ke všem třídám jsou v adresářích obsažena také jejich rozhraní, která jsou pojmenovaná stejně jako třídy, ale místo počátečního určujícího písmena T je písmeno I, které dává najevo, že se jedná o rozhraní tedy interface.

2.7 Použité technologie

Pro splnění požadavků na snadnou rozšiřitelnost, přenositelnost a běh na běžných počítačích, jsem si vybral online řešení. Takto je možné, aby systém běžel na libovolném počítači, nebo více počítačích současně a v případě selhání jednoho stroje pak mohl být nahrazen libovolným jiným a není nutné čekat na dlouhou opravu, jako u většiny jiných restauračních systémů. Požadavkům na přenositelnost jsem uzpůsobil i výběr nástrojů na tvorbu. Díky tomu může RESYS běžet na většině hostingů, které jsou v dnešní době dostupné.

Při tvorbě jsem využil nástroje, které jsou k dispozici zdarma. Tyto nástroje lze rozdělit do několika skupin podle svého účelu a níže je představím. Samotná aplikace je psaná v programovacích jazycích PHP, HTML, JavaScript s využitím databáze MySQL.

2.7.1 Návrhové aplikace

Pro návrh systému RESYS jsem používal tři aplikace, které jsou v dnešní době poměrně rozšířené. Prvním je ARISExpress verze 2.4, která je pro nekomerční užití zdarma a byla doporučena vedoucím práce. Jako druhý nástroj jsem používal VisualParadigm for UML CE verze 10.2 a třetím Toad Data Modeler Freeware verze 2.24. Všechny nástroje byly pro práci příjemné a intuitivní. I když mají všichni za účel zpřehlednit pohled na systém, tak každý z nich má trochu jiné zaměření.

- ARIS Express - tento nástroj je zaměřený na tvorbu diagramů BPMN a BP, které mají sloužit pro prezentaci systému objednavateli, managementu a ostatním uživatelům, kteří potřebují znát systém a nemají vždy potřebné znalosti z oblasti informačních technologií. Mimo to jsem v této aplikaci vytvořil přehledy uživatelů, základních procesů jednotlivých odvětví, které by měl restaurační systém pokrývat bussines procesy fungování sestavené aplikace.
- VisualParadigm for UML CE - složí pro ujasnění potřeb systému a kompletní návrh z pohledu UML metodiky. Jedná se o rozsáhlý nástroj, který pokrývá celou problematiku tvorby aplikací od zadání přes Use Case Diagramy, sekvenční diagramy až po dokumentaci a časový harmonogram. V tomto programu jsem tvořil návrhy tříd, interakce tříd mezi sebou a základní logiku systému.

- Toad Data Modeler Freeware - je nástroj, který se používá pro návrh databází. Velice snadno se v něm tvoří ER diagramy a je použitelný pro generování DDL dat databáze. Veškeré databázové části byly navrženy právě v této aplikaci. I když se jedná o poměrně jednoduchý nástroj, tak jsem si jej vybral právě z důvodu snadné kontroly a generování dat. Mimo to jsem již Toad Data Modelerem již dříve pracoval a znám jeho vlastnosti.

2.7.2 Data a databáze

- MySQL - MySQL je volně dostupný databázový systém, který je v dnešní době hodně využíván. Aktuálně je tato databáze zařazena pod firmou Oracle. I když je MySQL mnohem chudší na dostupné funkce a má i další omezení oproti databázi Oracle, tak jsem si jí vybral z důvodu snadného nasazení na hosting. Většina dnešních hostingů podporuje právě MySQL. Provoz na databázi Oracle by byl rozhodně bezpečnější a spolehlivější, ale bohužel na úkor snadného nasazení a provozní ceny, což byly klíčové požadavky mé práce.
- PHP adminer - jedná se o jednoduchý nástroj, který usnadňuje práci s databází. Pro svou práci jsem si jej vybral hlavně kvůli jednoduché instalaci, která spočívá v nahrání jednoho souboru do složky, která je pomocí aplikace XAMPP simulována jako server.

2.7.3 Výroba provoz a testování

Pro provoz a testování systému je třeba serverový prostor. Vzhledem k tomu, že je při vývojových pracích mnohdy nutné dělat operace, které by mohly brzdit provoz serveru, tak jsem se rozhodl k jejich vytváření přímo na lokálním serveru v počítači.

- XAMPP - jedná se o instalační balík pro systém Windows, který obsahuje databázi MySQL, Apache server a PHP. V jednoduchosti řečeno umožní vytvořit z běžného počítače server. Celá aplikace byla tvořena za pomoci XAMPP, protože umožňuje rychlejší odpovědi od serveru a je možné pracovat i bez přístupu na internet.
- NetBeans - vývojové prostředí, které mi usnadňovalo tvorbu zdrojových kódů a to ve všech použitých programovacích jazycích.
- Chrome - webový prohlížeč, který je v dnešní době velice populární. Tuto aplikaci jsem využíval v rámci celé tvorby systému a to ne jen na samotné zobrazování stránek, ale také na optimalizaci a ladění javascriptových částí.

2.7.4 Programovací jazyky

Vzhledem k tomu, že jsem aplikaci tvořil jako online systém, tak jsem tomu uzpůsobil i použité programovací jazyky. I když jsem se mohl vydat cestou servletů, asp a dalších, tak jsem se rozhodl zůstat u častěji používaných jazyků, s jejichž tvorbou mám dřívější zkušenosti.

- HTML 5 - Hypertext MarkupLanguage je hlavním z jazyků, které jsou v dnešní době používány pro vytváření webových stránek. Jedná se o neprocedurální jazyk, který slouží čistě pro zobrazení jednotlivých prvků statické stránky.

- CSS – Cascading Style Sheets je jazyk, který slouží ke grafické úpravě webových stránek, nebo přesněji HTML tagů. Jedná se vlastně o jazyk, který může každému prvku stránky určit jeho pozici, velikost, barvu a mnohé další vlastnosti. HTML stránka bez CSS by nebyla ničím jiným, než černobílým listem papíru.
- PHP - je scriptovacím programovacím jazykem, který dává stránkám dynamiku. Aktuální verze je XX. Má aplikace je tvořena ve verzi XX. Dříve měl tento jazyk velká omezení, ale od verze 5 došlo k zavedení prvků OOP programování a to vedlo k obrovskému rozšíření možností práce s tímto jazykem. Velkou nevýhodou však zůstává fakt, že PHP běží na straně serveru. Pokud tedy dojde ke spuštění scriptu, který je napsaný objektově, tak se objekty vytvoří, provedou se požadované operace a po ukončení scriptu se všechny objekty zase zničí. Zjednodušeně řečeno, instance objektu na serveru existuje jen po dobu běhu scriptu.
- JavaScript - je objektově orientovaný multiplatformní scriptovací jazyk, který je vhodné využívat při tvorbě některých prvků stránek. Jeho největší výhodou je, že běží na straně klienta a umožňuje načítat data a provádět operace dynamicky bez nutnosti znovunačtení celé stránky. V aplikaci jej využívám třeba na dynamické zobrazování seznamů nabídek a podobně.

2.7.5 Knihovny a rozšíření

- JQuery - je výkonná JavaScriptová knihovna, která usnadňuje práci s JavaScriptem. Instalace je velice jednoduchá. V principu se dají využít dva způsoby. Buď je jQuery knihovna stažena a uložena jako JavaScriptový soubor k ostatním souborům na serveru nebo je vložena jako externí odkaz na soubor, který je k dispozici online. V své aplikaci jsem zvolil online soubory. JQuery využívám napříč celým systémem. Nejčastěji však jako rozšíření jQueryValidate a jQuery UI pro zobrazování některých grafických prvků a validaci vstupních dat.
- SimpleXML - rozšíření jazyka PHP, které dovoluje snadnou práci s XML soubory. V systému jej využívám pro generování XML objednávce pro dodavatele.
- PDO - je rozhraní pro snadnou práci s připojením k databázi. Od verze 5 je již v PHP pevně zakořeněná, ale pořád se PDO dá považovat za specifické rozšíření. PDO je univerzální přístup k datům v databázi bez ohledu na typ databáze. V systému přes PDO inicializuji veškeré interakce s databází.

2.8 Představení vytvořeného systému

Nyní, když jsem poukázal na postup vývoje a rozložení celé aplikace, tak je čas představit i vytvořený program. Jako první se zaměřím na základní kameny, které slouží pro chod aplikace a následně se přesunu k jednotlivým blokům a modulům. U těch popíši jejich účel, představím je z pohledu business procesů a se zaměřením na zajímavá místa jejich provedení. Nakonec ještě zhodnotím celkové požadavky na běh systému.

2.8.1 Vstupní logika aplikace a základní obsluha příkazů

Asi jako každá webová aplikace postavená na HTML kódu má i můj systém základní strukturu složenou z hlavičky, těla a patičky. Hlavička je uložena zvlášť v souboru

hlavicka.php. Její struktura je poměrně jednoduchá. Celá je čistě v HTML kódu bez využití PHP, protože jeho využití zde skutečně není potřeba. Samotné tagy DOCTYPE, html, head a body není třeba nijak zvlášť představovat. Za zmínku však stojí fakt, že hlavička končí tagem<body> bez uzavření. Tento tag je uzavřený v souboru paticka.php, který je vložen na konec stránky.

Nejdůležitější částí hlavičky je kód mezi tagy<head></head>. Jako první zde mám metatag pro určení kódování stránky³⁰, jazykové mutace, copyrightu, autora a také robots nastavený tak, aby se stránky neindexovaly a nebyly vyhledatelné přes prohlížeč³¹. Následuje jeden samostatný tag<title>, který udává titulek stránky, a pod ním jsou tři skupiny dalších tagů. První jsou tagy pro určení cesty k css stylům, následují tagy pro načtení potřebných částí jQuery a jako třetí jsou tagy, které vkládají JavaScriptové soubory.

Od samotné hlavičky přejdu k hlavní vstupní bráně aplikace a to k souboru index.php, který je umístěn v adresáři editor. Tento soubor obsahuje kompletní rozcestník celé aplikace. Jedním z mých cílů byla však maximální možná decentralizace systému, proto jsem zde zachoval konstrukci, která není nijak obsáhlá a slouží pouze jako rozcestník.

Vycházím z principu, kdy se pro zobrazení libovolného modulu kontroluje právo pro modul a podle zvolené volby se pak načte obsah. Než je však načtený samotný modul, tak musí být vloženo několik dalších souborů, které jsou nutné pro běh aplikace jako celku. Jako první se načítá soubor jadro.php, který obsahuje mechanismus pro includování dalších souborů a soubor konfigurace.inc, kde jsou používané konstanty. Po načtení jádra a konfigurace se načte soubor hlavičky a tím se určí potřebné metainformace.

Vzhledem k tomu, že pro práci se systémem musí být uživatel přihlášen, tak je nejprve vyhodnocen stav proměnné \$_SESSION, jestli je přihlášení aktivní. Pokud není, dochází ke vložení souboru pro obsluhu přihlášení a zobrazení přihlašovacího formuláře. K samotnému principu nastavení \$_SESSION se dostanu při popisu přihlašování. Aktuálně je potřebná pouze informace o tom, že přihlášení je rozlišováno na základě existence \$_SESSION['prihlaseni']

Pokud je přihlášení vyhodnocené kladně, tak dojde ke vložení souboru menu.php, kde jsou jednotlivé volby systému. Na základě přihlášeného oprávnění se pak volby zobrazují nebo skrývají.

Princip zobrazení podle akce systému a vkládání modulů je pak vždy řízeno ve dvou fázích. První je zobrazení samotného modulu, které řeší právě popisovaný soubor index a to díky jednoduchému rozhodování za použití příkazu switch, který vyhodnocuje proměnnou \$_SESSION['aktualni_pozice']. V podstatě se jedná o situaci, kdy je kontrolováno zaslání informace o kliku na položky hlavního menu³² a nastavení proměnné

³⁰ V celé aplikaci využívám kódování UTF-8

³¹ Je nežádoucí, aby část správy aplikace byla dostupná jinak než určeným odkazem, nebo přes URL adresu.

³² Za využití PHP metod post a get.

pro pozici v SESSION. Pokud je zachycená akce z menu, tak dojde ke změně SESSION. Tím je pak při vyhodnocení includován požadovaný soubor.

Druhá fáze probíhá na úrovni konkrétního vloženého modulu. Zde přichází na řadu vyhodnocení proměnné POST nebo GET podle toho, o jakou akci se jednalo. Na základě vyhodnocení je pak provedena určitá funkce.

Na závěr souboru index.php se pak nachází vložení obsluhy přihlášení, která se zobrazuje buď samotný přihlašovací formulář, nebo možnost pro přihlášení dalšího uživatele a vložení patičky stránky.

2.8.2 Společné rysy zdrojových kódů

Vzhledem k podobnosti jednotlivých modulů je zde i jistá podobnost zdrojových kódů. O využití MVC architektury a objektově orientovaného programování jsem se již zmiňoval. S tím jsou však spojené další charakteristiky aplikace.

Všechny metody, které jsem ve své aplikaci použil, jsou určené pouze k jednomu účelu a nikdy nevykonávají dvě činnosti. Z tohoto důvodu má aplikace obsahuje více metod, ale jejich provedení je vždy poměrně jednoduché.

V mé práci jsem jen minimálně využíval návratové hodnoty. Veškeré výsledky akcí jsou vyhodnocovány pomocí výjimek. V tomto případě se nejedná o standardní postup, ale vzhledem k charakteru mé aplikace jsem se rozhodl pro tento postup, protože výjimka zastaví běh aplikace vždy, když je vyvolána. Zatímco návratové hodnoty by musely být hlídané napříč celým systémem. Pokud by někde došlo ke špatnému vyhodnocení návratové hodnoty, tak by mohlo dojít k poškození dat, která považuji za klíčová.

Struktura kódu jednotlivých modulů je v jednotlivých částech podobná. Samozřejmě, že každý modul má své specifické konstrukce, ale základní myšlenka je vždy stejná. Kódy se nejvíce přibližují v jednotlivých částech na úrovni modelů, kontrolerů a pohledů.

Moduly vždy obsahují privátní atributy a veřejné metody pro přístup k atributům. Mimo to zde využívám parametrický konstruktor, který nastavuje všechny parametry při vytvoření instance objektu. Metody set jsou tedy v mé aplikaci pouze doplňující, ale pro úplnost jsem je zde ponechal. Metody set a get jsou vždy nastavované samostatně i přesto, že PHP 5 umožňuje využití předdefinovaných metod __set a __get. Jejich konstrukce se mi nezdá úplně vhodná a v samotném kódu působí spíše zmatek než ulehčení.

Kontroler pak téměř vždy obsahuje atribut tří polí podle toho, zda se jedná o aktivní, neaktivní nebo všechny záznamy z modulu, atribut pro připojení k databázi a atribut pro aktuální prvek seznamu. V konstruktoru pak nastavuji atribut připojení do databáze, inicializuji pole na prázdnou hodnotu a zakládám novou prázdnou instanci aktuálního prvku.

V každém kontroleru mám dále gettry k atributům polí a k atributu aktuální prvky. V tomto případě jsem se rozhodl jít cestou, kdy gettry jsou zároveň i settry, protože PHP běží na straně serveru a neudrží instanci. Kdybych tedy prováděl nejprve set a pak get metodu, tak by docházelo pouze k situaci, kdy v jedné metodě atribut nastavím, druhou jí předám a po doběhnutí scriptu pak instance stejně skončí destruktorem, takže by se jednalo pouze o několik výpočetních kroků navíc.

Pole záznamů pak nastavuji tak, že v jejich těle volám privátní metodu nactiPrvky, která vybere potřebné informace z databáze podle zadaného parametru a zařadí je do příslušného seznamu. Tento seznam je pak vrácen příslušnou metodou, která je veřejná, takže jí využívají ostatní třídy.

Aktuální prvek je pak nastavován trochu odlišně a to samostatným dotazem na konkrétní záznam databáze. Dříve jsem využíval možnost procházení celého seznamu prvků a výběru konkrétního záznamu, ale tento přístup není vhodný pro PHP, protože dochází k situaci, kdy jsou vždy načtené všechny záznamy a pak z nich vybírám ten správný. Je tedy vhodnější rovnou vybrat jeden záznam z databáze.

Mezi ostatní metody, které využívají kontroler, patří přidání, editace, odebrání a změna aktivity záznamu. Mimo to zde jsou podpůrné funkce pro kontrolu duplicitních údajů. Jednotlivé metody jsou strukturou poměrně podobné. Na začátku vždy hlídají správnost všech parametrů a možnost vykonání dané akce. Následně dochází k sestavení SQL dotazu, který je vykonán, a tím jsou upravena potřebná data. Konkrétní specifické metody jsou popsány v textu níže.

Třída pohledů pak ve většině případů obsahuje metodu na výpis prvků z tabulky a formuláře pro editaci a přidání. Výpis prvků je strukturován do tabulky a probíhá na bázi jedné proměnné \$vypis, kam jsou postupně přidávány jednotlivé části v podobě swingových řetězců, které jsou na konci metody vypsané příkazem echo \$vypis. Díky tomuto principu je možné využívat libovolně další prvky PHP- jako jsou podmínky a cykly a přitom přímo sestavit potřebný výstup. Stejným postupem je pak sestavený i formulář pro přidání nebo editaci.

Tím jsou popsány moduly, ale zbývá ještě databázová část. Veškeré databázové metody jsou vloženy ve třídě TMySQLKonektor.php. Připojení do databáze a vznášení dotazů probíhá pomocí extenze PDO, která poskytuje jednotné API k různým databázovým systémům a poskytuje funkce právě pro zpracování dat do databáze.

Třída TMySQLKonektor obsahuje proměnnou \$options, která obsahuje část nastavení týkající se kódování. Ostatní prvky pro realizaci připojení jsou řešeny formou globálních proměnných. Metody třídy jsou potom rozděleny podle využití.

První je metoda vyberZDatabaze, která vybírá data z databáze a má jako parametr proměnnou \$dotaz. Zpracování pak probíhá tak, že je nejprve vytvořena instance PDO, která realizuje připojení do databáze. Následně je pak využitý příkaz query, který dotaz

zpracuje. Pokud není výsledek prázdný, tak je vráceno pole dat, které byly získány z databáze. Ukončení připojení pak probíhá na základě volání destruktoru nad instancí PDO.

Ostatní metody jsou pak strukturou podobné. Metoda `upravavDatabazi` slouží pro provádění dotazů `update` a místo příkazu `query` je vykonávána příkazem `exec`. Jako jedna z mála metod v mé aplikaci pak vrací zpět hodnotu, která vyjadřuje počet upravených záznamů. Metoda `pridejDoDatabaze` je prováděna stejným příkazem jako úprava, ale vyhazuje výjimku v případě, kdy je vrácena prázdná hodnota³³. Jako poslední je k dispozici metoda `pridejSNulovymNavratem`, která slouží pro zakládání dat do databáze pro instalátor.

Připojení do databáze jsem již prošel, ale nyní k tomu, jak jsou tvořené dotazy zasílané do databáze. Všechny dotazy pro výběr z databáze jsou realizované z důvodů vyšší bezpečnosti. Pokud jde o vkládání hodnot, tak zde jsou všechny stringové hodnoty uvedeny v uvozovkách. Dotazy jsou pak tvořené vždy v metodě, která se stará o danou operaci.

2.8.3 Obecná charakteristika struktury modulů

Po přechodu do modulu dojde k zobrazení doplňkového menu pod menu hlavním a seznamu položek daného modulu, které jsou aktivní. Na začátku každého záznamu je vždy možnost pro editaci záznamu, jeho aktivaci a deaktivaci, a pokud to vazby systému dovolují, tak pro odebrání. Ve specifických případech jsou na tomto místě další akce, které odpovídají požadavkům daného modulu.

Možnost založení nového záznamu do modulu se pak vždy nachází jako první položka podmenu. Po volbě této akce nastává situace, kdy se zobrazí formulář pro přidání nového záznamu a v horním podmenu se objeví tlačítko zpět, které provede návrat na výpis bez uložení formuláře.

Samotný formulář pro přidávání je vždy trochu jiný, protože každý modul má jinou strukturu svých záznamů. Společné je však ošetření vstupních dat. Veškerá vstupní pole mé aplikace jsou ošetřena ve dvou úrovních. Prvním kontrolu provádí `jQuery` knihovna `Validation`, který umožňuje snadnou kontrolu pomocí `jQuery`. Část metod této knihovny je předdefinovaných, ale je zde možnost dopsání metod vlastních, které jsem v případě mé práce využil třeba pro ošetření desetinných čísel.

Vzhledem k tomu, že `jQuery` může v některých případech selhat³⁴, tak jsem do systému zařadil ještě druhou kontrolu na úrovni PHP. Ta je prováděna vždy v kontroleru modulu před tím, než je vznesen dotaz do databáze. Dvojitá kontrola může být zbytečnou zátěží na systém, ale data jsou podle mne velice cenná na to, abych dovolil jejich ohrožení na úkor malého zlepšení výkonu.

³³ Prázdná hodnota nastane v případě, kdy vložení není úspěšné.

³⁴ V případě nedostupnosti zdrojového souboru, nebo při chybném zásahu ze strany programátora.

Aplikaci jsem stavěl tak, aby při nesprávném vyplnění jedné nebo několika položek zůstaly hodnoty správně vyplněných položek zachované. Pod formulářem se pak pouze zobrazí hlášení, které poukazuje na nesprávně vyplněné hodnoty. Samozřejmě jsou v prostoru pro hlášení jmenovitě vypsány jednotlivé hodnoty, které nebyly vyplněné dle očekávání.

Editaci záznamu v mém systému jsem postavil na podobném principu jako zakládání záznamů. Formulář pro editaci je z pravidla stejný jako pro zakládání a stejně tak kontrola jednotlivých polí. Na rozdíl od založení nové položky editační formulář vychází z již předvyplněných hodnot, které jsou načtené podle identifikátoru záznamu.

Nad výpisem položek se pak nachází filtry pro zobrazení aktivních, neaktivních a všech položek. Mezi seznamem a menu je pak vždy podmenu, které obsahuje minimálně možnost pro přidání nového záznamu. Jedinou výjimkou je samotný modul modulů, který je pevně zadán bez možnosti odebírání a přidávání. Jedinou možností je zde editace názvu a deaktivace/aktivace modulu.

2.8.4 Přihlašování a správa přihlášených uživatelů

Má aplikace umožňuje kromě standardního přihlášení také přihlášení více uživatelů na jednom stroji, což není u webových aplikací příliš obvyklé. Z pohledu systému pro stravovací zařízení je tato možnost však nezbytně nutná, protože je častá situace, kdy obsluhu provádí více jak jeden zaměstnanec.

Problematika přihlášení je mírně odlišná od ostatních částí systému. Přihlášení není samostatným modulem a obslužný soubor této části je do systému vložen vždy bez ohledu na to, s jakým modulem uživatel aktuálně pracuje. Obsluha přihlášení pak rozlišuje stavy a vybírá, co bude zobrazeno podle toho, zda je již nějaký uživatel přihlášený nebo ne.

Pokud není žádný uživatel přihlášený, tak defaultně dochází k zobrazení přihlašovacího formuláře. Když už je některý z uživatelů přihlášený, tak systém zobrazí v levé části monitoru rozevírací panel, který umožňuje přihlášení dalšího uživatele, změnu aktuálního uživatele nebo odhlášení jednoho z přihlášených uživatelů. Tato část systému tedy neslouží pouze k přihlašování, ale řídí celou logiku přihlašování, odhlašování, dostupných modulů a aktuálního uživatele.

Celé přihlašování funguje na principu vkládání záznamů do proměnné SESSION, která udržuje potřebné informace po celou dobu práce se systémem. Ze samotné logiky fungování SESSION ji využívám jako pole, kde držím informaci o aktuálně přihlášeném uživateli v proměnné SESSION['aktualni'], všech přihlášených uživatelů v proměnné SESSION['vsichni'] a nakonec proměnnou SESSION['uzivatelske_moduly'], kde jsou moduly dostupné přihlášeným uživatelům. S tím, že aktuální uživatel a všichni přihlášení jsou spojeni do společného pole s názvem SESSION['prihlaseni'].

Nejprve však k tomu, jaké situace mohou při přihlašování nastat a jak na ně systém reaguje. Celý proces jsem zaznamenal do BP diagramu, který je přílohou této práce. Popis celého procesu by byl až příliš obsáhlý, proto se nyní zaměřím jen na klíčové části. Celý diagram vychází z předpokladu, že se do systému chce přihlásit nějaký, aktuálně pro systém neznámý uživatel. Podle toho, jestli je již někdo na počítači do systému přihlášen dochází k větvení, kdy je buď zobrazen přihlašovací formulář přímo, nebo je třeba jej vyvolat pomocí příslušné akce.

Po zadání údajů následuje akce kontroly vyplnění údajů. Pokud jsou přihlašovací údaje zadané, tak přihází na řadu ověření, zda tyto údaje existují v databázi. Pokud je i tato podmínka splněná, dochází k ověření, jestli přihlašovací údaje nepatří uživateli, který je již do systému jednou přihlášen. V případě kladného výsledku všech kontrol dochází k rozvětvení procesu za využití možnosti souběžného běhu dvou větví, kdy v jedné větvi dochází k zaznamenání logu o přihlášení do databáze a z neznámého uživatele se stává uživatel s konkrétní rolí v systému. V druhé větvi je vyhodnocena situace, zda je již nějaký uživatel do systému přihlášen nebo ne. Pokud není, tak dojde k nastavení přístupu do systému podle právě přihlášeného uživatele. Pokud je, tak dojde k omezení práv na principu, kdy každý přístup do modulu musí být obsažen zároveň v obou oprávněních.

S přihlašováním je spojena také správa přihlášených uživatelů. V mém systému je vždy jeden uživatel označený jako aktivní a veškeré akce jsou v aplikaci vykonávány, jako by je vykonával on. Aktivní uživatel je vždy vypsán v panelu na levé straně obrazovky, kde je také možné provést přepnutí mezi více přihlášenými uživateli. Stejně tak k přepnutí aktivního uživatele dochází ve chvíli nově přihlášeného uživatele, který je automaticky vybrán jako aktivní a odhlášení aktivního uživatele, kdy je jako aktivní vybrán poslední přihlášený uživatel. Samotná akce změny aktivního uživatele nemá vliv na změnu pozice v systému.

Celý proces přihlášení má obecně tři hrozby - pokus o útok na systém, který mám zabezpečený přes jQuery validaci, duplicitní přihlášení stejného uživatele, které by vedlo k nemožnosti určení aktivního uživatele a jako třetí zde je riziko nesprávně určeného přístupu při přihlášení více uživatelů.

Z databáze je v procesu využito pět tabulek. Tabulka obsahující přihlašovací údaje uživatelů, kde dochází ke kontrole správnosti zadaných údajů, tabulka uživatelů pro zjištění údajů uživatele, tabulka akcí uživatelů, kam je zaznamenáno přihlášení, tabulka oprávnění a tabulka modulů, kde jsou zjištěné moduly, které danému právu náleží.

Z pohledu zdrojových kódů se o přihlašování starají tři třídy s jejich rozhraními, soubor obsahující obslužnou logiku přihlášení a jQuery validace. Hlavní obsluha je přiřazena na PHP kódy s HTML tagy a jQuery příkazy.

Ze specifických metod se zde vyskytuje například metoda `vypisPrihlasene`, která se stará o zobrazení panelu s přihlášenými uživateli. Dále pak metoda `zkontrolujPrihlaseni`, která se stará o kontrolu přihlašovacích údajů podle postupu, který je popsán výše. Tato metoda pak využívá privátní metody `nactiUzivatelePrihlaseni`, která získává informace o přihlašovaném uživateli a metoda `nastavUzivatskeModuly`, která provede nastavení modulů, které jsou uživateli zpřístupněné.

2.8.5 Správa uživatelů

Správa uživatelů se skládá ze tří specifických modulů, které jsem popsal v několika následujících odstavcích. Jako první představím modul `moduly`. Ten je ve svém principu jednoduchý, protože slouží pouze pro evidenci modulů a jejich vypínání a zapínání. Možnost přidávání modulů a odebírání není standardně k dispozici, protože moduly jsou stavební kameny systému a rozšíření v této oblasti znamená vždy nutnost zásahu do kódu aplikace.

Podstatná informace z tohoto modulu je pouze ohledně deaktivace modulu. Pokud je modul vypnut, tak je i automaticky odebrán ze všech oprávnění kromě administrátora. Při opětovném zapnutí modulu se však systém nevrátí do původního nastavení, ale moduly je nutné přidat k právům ručně.

Druhým modulem správy uživatelů jsou práva. Tento modul má na starosti evidenci oprávnění v aplikaci. Svým konceptem se jen mírně odlišuje od ostatních modulů. Výpis práv je poměrně strohý. Jsou v něm zobrazeny informace o názvu oprávnění, o tom jestli je právo aktivní a počet uživatelů, kteří mají právo přiřazené.

Přidání a editace práva je umožněno všem uživatelům, kteří mají do tohoto modulu přístup. Samotné přidání obsahuje všechny položky, které jsou k dispozici ve výpisu, a navíc je zde možnost zadání popisu oprávnění. Editace je pak stejná, ale pole jsou předvyplněná hodnotami podle toho, jaká položka je editována.

Pokud jde o omezení modulu, tak těch jsem musel zavést hned několik. První omezení je na možnost editace, odebírání a mazání oprávnění a to ve dvou případech. Jednak není možné editovat, mazat ani deaktivovat právo administrátora, protože toto oprávnění musí být v systému vždy zachované. Dále jsem ještě zvolil stejné omezení pro všechna oprávnění v případě, že je přihlášený více jak jeden uživatel. Druhým výrazným omezením je možnost přiřadit právu maximálně ty moduly, které má uživatel provádějící přidání, nebo editaci práva jiného.

Třetí modul a poslední modul této části je modul `uzivateli`. Tento modul se stará o správu všech uživatelů systému. V mé aplikaci nedovoluji do systému registraci, takže se jedná pouze o uživatele, kteří jsou do systému vloženi odpovědnou osobou. Samotnou strukturou se tento modul neodlišuje od konceptu ostatních. Je zde možnost přidání nových záznamů, jejich editace, odebírání a změny aktivity. Vzhledem k tomu, že je nutné mít pro správu systému minimálně jednoho uživatele, a to s oprávněním `admin`, tak jsem vytvořil jeden univerzální záznam, který není možno odstranit ani editovat.

Při zakládání záznamu dochází ke standardní kontrole na vyplnění údajů a jejich správnost. Systémem zde hlídám několik důležitých omezení. Prvním je nemožnost přidání do systému oprávnění vyšší, než má aktuálně přihlášený uživatel, dalším je omezení na nemožnost práce se záznamy v případě přihlášení více uživatelů.

Správa uživatelů je rozmístěna celkem mezi pět databázových tabulek, a to tabulka modulů, práv, uživatelů, jejich přihlašovacích údajů a nakonec samotné logování. Diagram s ukázkou tříd v tomto případě nevkládám, protože je identický jako v případě přihlašování.

Z pohledu zdrojových kódů je však tato část trochu obsáhlejší. Celkem je zde zapojeno deset tříd s rozhraními, tři obslužné soubory a samozřejmě jQuery validace. Až na jQuery je použit čistě jazyk PHP a HTML. Většina tříd a metod zde se nijak neodlišuje od standardního chování, až na samotné uživatele. Zde je model postaven tak, aby v sobě obsahoval ještě instance třídy TLoginModel a instanci třídy práva. Uživatel tedy ve většině případů nese tyto dvě instance, ale je zde i metoda, která vrátí uživatele bez těchto vazeb.

Pokud jde o atypické metody, tak zde stojí za zmínku metoda `predejUzivateleBezVazeb`, která má jako parametr `id_uzivatele`. Na základě tohoto parametru jsou pak z databáze vybraná data, ze kterých je vytvořena instance třídy `TUzivatelModel`. Do této instance však není doplněna vazba na oprávnění a přihlašovací údaje. Díky tomu je načítán menší objem dat, čehož využívám v dalších částech aplikace, kde není nutné, aby byly avizované vazby k dispozici.

2.8.6 Správa dodavatelů

Správa dodavatelů je relativně malou částí skládající se pouze z jednoho modulu šesti tříd a dvou databázových tabulek. Tento modul má za úkol udržovat data o dodavatelích a umožňovat objednávku jejich sortimentu pro zásobování podniku.

Modul dodavatelé se odlišuje od ostatních dvěma rozšiřujícími funkcemi, které jsou spolu přímo provázané. Prvním z nich je možnost nazvaná sklad, která se nachází u každého záznamu o dodavateli. Tato položka umožňuje navázat jednotlivé skladové suroviny, které jsou evidované do skladu na konkrétního dodavatele. Díky této vazbě je pak možné snadno objednat potřebné suroviny.

Při zakládání vazeb na skladové suroviny je nejprve vypsán seznam skladových položek, které je možné od dodavatele teoreticky objednat. Na základě výběru konkrétní položky z muliselectboxu pak dochází pomocí JavaScriptu k zobrazení dalších formulářových oken, která umožňují zadat hodnoty pro název položky, tak jak jí eviduje dodavatel, EAN kód položky pro identifikaci produktu, cenu složky³⁵ a v jakých jednotkách je surovina

³⁵ Na základě této ceny je počítána systémem doporučená cena za jednotku suroviny a dále za samotnou položku v nabídce.

dodávána³⁶. Po odeslání hodnot dochází k jejich kontrole a založení záznamu do příslušné tabulky. Tyto záznamy pak slouží pro samotné vytváření objednávek.

Druhá odlišnost, která je charakteristická pro tuto část aplikace je možnost objednávky. Ta funguje na základě určených vazeb mezi skladem dodavatelem. Princip ovládání je celkem jednoduchý, ale obslužná logika už je o něco těžší.

U každého záznamu je možnost volby akce objednávka. Po jejím využití se zobrazí výběr položek, které jsou k dodavateli přiřazené jako vazby na sklad. Po výběru ze seznamu dochází k zobrazení dalších polí pomocí JavaScriptu. Do těchto polí je možné vybrat počet kusů produktu, který bude objednán od dodavatele. Po potvrzení objednávky pak může nastat několik situací.

Podle toho, jaký je vybraný typ pro objednávání při založení dodavatele, může být buď odeslán e-mail s objednávkou v textové podobě na adresu dodavatele, nebo je vygenerován CSV soubor pro uložení do počítače, nebo XML soubor, který se uloží do připravené složky na serveru a zpřístupní se dodavateli, nebo je generovaný soubor pro tisk.

Správa dodavatelů je kombinací PHP, HTML a Javascriptových kódů a je zde poměrně hodně specifických metod. Jednak se jedná o metody, které spravují navázání surovin na dodavatele, ale také o metody k samotným objednávkám produktů od dodavatelů.

Z metod, které spravují zmíněné navázání, se jedná o metodu `zalozVazbuNaSklad`, kde dochází jedna k založení samotné vazby, což by nebylo nic neobvyklého, ale dále je zde aktualizovaná prodejní cena suroviny podle nákupní ceny.

U objednávkových metod to pak jsou například `pripravCSVObjednavku`, která generuje CSV soubor pro možnost jeho stažení, nebo `pripravXMLObjednavku`, která generuje XML soubor a ukládá jej na server. Tato metoda používá třídu `SimpleXMLElement` a generovaný soubor je postavený na základu přenosu objednávky do účetního systému Money S4.

2.8.7 Správa skladu

Správa skladu je z pohledu mé aplikace rozšířením, které není v základních vlastnostech podobných systémů obvyklé. Samotná o sobě se jedná o poměrně jednoduchou část, která má jen několik málo rozšíření oproti ostatním. Celkem jsou zde obsažené tři samostatné moduly a to sklad, jeho kategorizace a jednotky, které přímo se skladem souvisejí³⁷.

Prvním z modulů, který představím je kategorizace skladu. Ta vychází ze stromové struktury, takže je zde i trochu jiná struktura modelu a kontroleru. Model má kromě jednoduchých atributů také pole potomků a odkaz na svého rodiče. Pokud jde o pole potomků, tak to plním v modelu tak, že zasílám celé vytvořené pole najednou pomocí

³⁶ Samozřejmě i s obsahem balení.

³⁷ Jednotky jsou dále provázány na dodavatele a nabídku, ale vždy však ve spojení se skladem.

metody set. Složitější konstrukce postupného plnění nemá v PHP smysl, protože by se prováděly další kroky navíc s instancí objektu, která se neudrží v paměti.

Kontroler se pak liší ve způsobu načítání a udržované struktuře záznamů. Kontroler neobsahuje atributy jednotlivých seznamů s aktivními, nebo neaktivními záznamy, ale atribut pro kořen stromu. Načítání pak funguje na principu postupného načítání podřízených potomků.

Nejprve je načten kořen stromu, nebo v případě načítání podstromu prvek, který kořen reprezentuje. Následně je vyhodnoceno, jestli má tento prvek potomky. Pokud ano je volána metoda pro načtení potomků, která se tak dlouho rekurzivně opakuje, dokud nejsou načteny potomci všech záznamů.

Mimo strukturované načítání je samozřejmě i k dispozici metoda, která z databáze vybere jednu konkrétní kategorii, nebo třeba všechny kategorie, které jsou určené pro sestavování nabídky. Veškeré operace pro ukládání a odebírání jsou pak uspořádané tak, aby nemohlo dojít například k situaci, kdy se kategorie mezi sebou zacyklí, nebo vzniknou podobné komplikace.

Jako druhý představím modul jednotky, který se stará o kompletní správu měrné soustavy aplikace. Tento modul v sobě obsahuje záznamy o jednotkách, které jsou v systému používány a jejich vzájemných vztazích. Každý z typů jednotek má svojí základní jednotku, která je využívána jako převodová mezi jednotlivými jednotkami.

Nakonec k samotnému modulu skladu. Ten je v základu stejný jako ostatní moduly, které jsem popsal v obecných charakteristikách kódu. Navíc je zde však několik metod, které umožňují provádět pohyb nad skladovanými surovinami. Tyto metody nejsou nijak záludné, ale jsou jistě specifické.

Tuto část aplikace jsem psal v PHP a HTML kódu. JavaScript jsem využil pouze okrajově ve formě validace polí a upozornění na přepočty stavu skladu při změně jednotky. Většina metod v této části je stejných jako v částech ostatních. Mezi specifické metody této části patří například `nactiKategorieSkladu` z modulu `kategorie skladu`, kterou jsem popsal výše.

Dále pak metody, které se starají o funkce naskladnění a vyskladnění surovin na skladu, které jsou celkem tři³⁸. Jako první zde mám metodu `naskladniVlastniMnozstvi`, která slouží pro naskladňování surovin do skladu. Další dvě metody pak slouží pro vyskladňování. První se nazývá `vyskladniMnozstvisJednotkou` a slouží pro ruční vyskladnění surovin a druhá `vyskladniVlastniMnozstvi` a ta je volána pro vyskladnění zboží v případě potvrzení objednávky.

³⁸ Všechny tři tyto metody volají metodu pro zápis logu skladu.

2.8.8 Správa nabídky

Správa nabídky je částí, kde se střetávají suroviny ze skladu s objednávkami. Tato část se skládá ze dvou modulů. Jednak ze samotné nabídky a dále z kategorizace nabídky. Vzhledem k tomu, že kategorizace nabídky funguje na stejném principu jako u kategorizace skladu, tak se jí nebudu dále zabývat.

Samotný modul nabídky je specifický v tom, že je jakýmsi můstkem mezi skladem a objednávkou. Celý modul funguje tak, že na základě dostupných surovin skladu je možné složit nabídku, které se přiřadí cena a specifický název. U jednotlivých surovin je pak nutné určit, v jaké jednotce budou jako součást nabídky a v jakém množství. Tyto údaje se do systému zadávají již při samotném vytváření nabídky.

U nabídky je vždy ohlídáno, aby nebylo možné zadat více jednotek stejné suroviny do jedné nabídky. V takovém případě by pak nebylo reálně možné uhlídat, kolik surovin má být ze skladu vyskladněno a kolik jich zbývá.

Z pohledu zdrojových kódů se jedná o kombinaci PHP, HTML a JavaScriptu, který je v této části využíván hlavně na zobrazení doplňujících polí při zakládání nabídky. Ze zajímavých metod je zde obsažena například `upravSystemovouCenu`, což je metoda, která upraví cenu u nabídek podle změny nákupní ceny nějaké suroviny.

2.8.9 Správa rozložení restaurace

Správa rozložení prostor restaurace, která může být také označována jako mapa stolů, je v mé aplikaci řešena rozdílně oproti ostatním modulům. V celé práci řadím tento modul do bloku nabídek, ale zde jej představím samostatně.

Jak již bylo řečeno, tato část je pouze jediným blokem, který se stará o grafické zobrazení rozložení stolů v restauraci. Po přechodu do modulu nedochází ke zobrazení klasického výpisu, ale objeví se plocha restaurace, která je v mé aplikaci pevně definovaná. Na tuto plochu je pak možné vkládat jednotlivé stoly, měnit jejich pozici, název a rozměr.

Změna pozice je limitována velikostí definované plochy a velikost stolu je určena v JavaScriptovém souboru `rozsireni.js` na hodnoty minimálně 90 pixelů šířky a stejně tak výšky. U maximálního rozměru je pak hodnota nastavena na 300 pixelů. U názvu je pak pouze jediné omezení a to na název. Pokud toto pole zůstane prázdné, tak nedojde k uložení akce z důvodů využití JavaScriptového volání `post`, které odesílá data přímo pro obslužný script, které se zastaví na podmínce, která vyhazuje výjimku.

Mimo vložení nového stolu je zde ještě možnost stůl odebrat, deaktivovat a zobrazit výpis deaktivovaných stolů pro zpětnou aktivaci. Výpis se nijak neodlišuje od standardních, které jsem popsal v této práci již několikrát. Odebrání stolu je limitováno přítomností objednávky, která byla na stůl vytvořena. Součástí pohledu, který poskytuje přehled o stolech je totiž i počet objednávek z vazební tabulky mezi objednávkou a stoly. V kódu je pak podmínka, která vyhodnocuje přítomnost objednávky a podle toho dojde k zobrazení tlačítka pro odeslání. Deaktivace stolu je pak možná kdykoliv bez omezení.

Z pohledu databáze jsou v této části pouze dvě tabulky, a to samotná tabulka stoly, která obsahuje informace o stolech a vazební tabulka mezi objednávkou a stoly, která určuje k jakému stolu je objednávka přiřazena.

Z pohledu zdrojových kódů jsou zde pouze tři třídy se svým rozhraním a k tomu obslužný soubor. Samotná obsluha dat této části a výpisy jsou řešené přes PHP a HTML. Výjimkou je však editace stolů, která je kompletně řešena přes jQuery a JavaScript obecně. V tomto případě se jedná o metody resizable a change. S tím, že metoda resizable umožňuje změnu velikosti a posun div bloku podle zadaných limitujících rozměrů a to jak na velikost div, tak na samotnou podkladovou plochu. Po ukončení změny velikosti je volaná sekvence stop, která odešle do databáze aktuální data podle pozice levého horního rohu a výšky a šířky prvku. Metoda change hlídá změnu pole input pro uvedení názvu. Po provedení změny jsou opět data odeslána obslužnému souboru metodou post. Na základě těchto dat se pak obslužný soubor postará o volání příslušné metody kontroleru a samotnému uložení dat.

2.8.10 Správa tvorby objednávek

Tvorba objednávek je klíčovým prvkem celé mé aplikace. Celkově se tato část skládá ze dvou kooperujících modulů a to z modulu objednávka a modulu správa stolů. Modul pro správu stolů jsem však popsal již výše, takže nyní se budu věnovat pouze části objednávek.

Modul objednávek funguje na základě stolů a nabídek. Nejprve je zobrazen přehled stolů, na které je možné vytvořit novou objednávku. To výběru stolu je zobrazen nový formulář kde se zobrazí položky nabídky podle kategorií. Na základě výběru nabídky se pak zobrazují doplňující informace s možností zadat počet kusů z dané nabídky, který bude v objednávce obsažen. Po založení nabídky je pak naplněná příslušná vazební tabulka a objednávka je vedena jako otevřená.

Na objednávku je pak možné zpětně přejít a otevřít si její obsah s rozšiřujícími možnostmi. Je zde možné stornovat část objednávky nebo jí potvrdit. Stejně tak je možné najednou potvrdit celou objednávku nebo jí stornovat. Do samotné objednávky se lze dostat buď přes konkrétní stůl, nebo výpisem všech objednávek, který je dostupný nad přehledem stolů v objednávkové části. V tomto výpisu je také možné objednávky filtrovat dle stavů a data založení.

Ještě je třeba zmínit, že objednávka je vždy vytvářena pod konkrétním uživatelem, který je v systému nastaven jako aktuální. Tohoto uživatele je možné změnit na začátku vytváření objednávky před výběrem stolu nebo i při samotném založení nebo editaci objednávky.

Z objednávek je automaticky vytvářen i odhad skladové zásoby, kterou vypočítává sám systém. Tento odhad je však značně závislý na počtu objednávek v systému³⁹. Počáteční odhady tohoto mechanismu jsou značně nepřesné. Z toho důvodu je mechanismus nastaven tak, aby začal odhady tvořit až při určitém počtu objednávek. Odhad pak je sestavován na základě dne v týdnu a počtů dnů, které jsou reprezentované konstantou POCET_DNU_ZAVOZ v konfiguraci.

Pro plné zprovoznění a definitivní odladění této funkčnosti bude třeba minimálně tři měsíční testovací provoz aplikace. Výpočet není pro chod aplikace klíčový, ale bude dobrým pomocníkem hlídání stavů zásob, protože velice přesně určí, kolik suroviny bude minimálně potřeba pro několik následujících dní a tím pomůže od zbytečného skladování zboží, které nebude v brzké době spotřebované.

Z pohledu tříd a souborů je zde využita klasická kombinace pohled, model, kontroler, obslužný soubor obsluhaObjednavky.php, soubor pro jQuery validaci a JavaScriptový soubor pro zobrazování polí pro zadání nabídky a kontrolu, zda se již nabídka v objednávce nevyskytuje, aby mohla být případně spárována.

Z pohledu metod, které jsou typické pro tento modul zde, mám metody pridejDoObjednavky, potvrdObjednavku, stornujObjednavku a další, které se starají o akce objednávky. Dále pak metodu nactiObjednavkyPodleData, která dostává jako své parametry \$od a \$do, které reprezentují datum. Na základě parametrů pak vybírá zadané záznamy, které jsou dále přes nabídky přenášeny až do skladu pro vyhodnocení predikce skladové zásoby.

2.9 Požadavky na běh systému

2.9.1 HW požadavky na systém

Vzhledem k tomu, že se jedná o online řešení, tak hardwarové nároky na systém nejsou nijak vysoké. Pro fungování aplikace stačí běžný počítač s myší, který zvládne běh webového prohlížeče. V případě rozšíření o tisk účtenek bude ještě potřeba standardní tiskárna účtenek s XML rozhraním.

2.9.2 SW požadavky

Systém RESYS je postaven a odladěn na prohlížečích Chrome 34.0.1847.116 a Internet Explorer 11. Pro práci s restauračním systémem tedy doporučuji jeden z těchto prohlížečů. Práce na jiném prohlížeči by měla být také možné, ale není možné garantovat kvalitu zobrazení a některé složitější funkce.

³⁹ Stejně jako provozovatel, ze začátku provozu nemůže, přesně odhadnout kolik surovin bude třeba, tak ani systém nemůže sestavit odhady bez dostatečné zásoby dat

2.9.3 Požadavky na server a připojení

Systém RESYS je postaven na online řešení a pro svůj běh vyžaduje umístění na serveru. Vzhledem ke své konstrukci je nezbytná podpora PHP verze 5 a MySQL verze XX. Pokud bude systém instalován na počítač bez připojení k internetu, je nutné doplnit jQuery knihovny přímo do aplikace. Aktuálně jsou totiž jQuery odkazované na online zdroje.

Vzhledem k tomu, že systém je citlivý na rychlost připojení a dostupnost dat, tak je vhodné využívat takový server, který bude mít dostupnost minimálně 99.5% a nejrychlejší dostupné připojení. Systém není přímo limitován rychlostí připojení, ale pomalé připojení znamená zdržování při práci se systémem.

2.10 Uživatelská dokumentace systému RESYS

Systém RESYS je postaven s důrazem na uživatelskou stránku. Se systémem bude pracovat mnoho uživatelů, kteří nemusí mít ani základní znalosti v užívání počítačových aplikací a bylo tedy nutné sestavit takový systém, který je vysoce intuitivní.

Systém má v základním nastavení pouze jedno oprávnění. Ostatní práva je třeba definovat na základě vlastních potřeb a to za pomoci principu správy modulů, které si můžete představit jako samostatné části. Každý modul pak ovlivňuje specifické nastavení. Jednotlivé moduly jsou pak sdružované do větších celků a ty pak tvoří celý systém RESYS.

2.10.1 Přihlašování do systému

Systém RESYS disponuje funkcí násobného přihlášení. Je tedy možné přihlášení více uživatelů na jednom počítači. Po otevření systému je uživatel vyzván k prvnímu přihlášení. Na základě přihlášení jsou uživatelům zpřístupněny jednotlivé části systému podle určeného nastavení a v pravé části je k dispozici panel s informací o aktuálním uživateli. Po rozkliknutí tohoto panelu se zobrazí možnost přihlášení dalšího uživatele, změna aktuálního uživatele a možnost odhlášení vybraného uživatele a možnost odhlášení všech uživatelů.

Při otevření možnosti přihlášení dalšího uživatele dojde k zobrazení přihlašovacího formuláře stejně jako při prvním přihlášení. Systém pak kontroluje duplicitní přihlášení stejného uživatele. Při přihlášení dalšího uživatele dochází automaticky k omezení práv. Omezení funguje na systému dostupnosti jednotlivých modulů, které jsou dostupné pro všechny přihlášené. Pokud tedy jeden uživatel má část systému dostupnou a druhý ne, tak v případě jejich souběžného přihlášení nebude daná část dostupná ani jednomu. Stejně tak může nastat situace, kdy některé funkce systému nebudou dostupné při vícenásobném přihlášení.

S možností principu přihlášení více uživatelů je spojená nutnost rozlišit, který uživatel z uživatelů je aktuální pro práci se systémem. Pokud je v systému přihlášených několik různých uživatelů, tak jsou všechny operace evidované, jako by je dělal právě jeden uživatel. Informace o tom, který z uživatelů je aktuální, se nachází v pravé části monitoru na prvku, který jsem již zmiňoval. Na tomto místě je pak možné jednoduše přepnout aktuálního uživatele nebo některého z uživatelů odhlásit.

2.10.2 Práce s uživateli

Ze začátku práce se systémem je vhodné definovat oprávnění uživatelů, kteří v systému budou pracovat. Práva se definují v menu pod položkou práva, kde dojde k zobrazení výpisu stávajících práv a k možnosti jejich přidání. Právo se dá vložit pomocí tlačítka přidat položku, které vyvolá formulář, kam je možné zadat název oprávnění, popis oprávnění a vybrat moduly, které budou v právu obsažené. Podobným způsobem pak je možné práva také editovat.

Po definici práv je pravý čas na vložení uživatelů do systému. To uděláte pod položkou v menu s názvem Uživatelé. Její volba vyvolá seznam všech uživatelů systému, kde je možné opět uživatele přidávat nebo je editovat stejně jako tomu je u práv. Přidání uživatele obsahuje možnost zadání jména uživatele, základních kontaktních údajů jako je e-mail a telefon a pomocí výběrového pole je možné zvolit oprávnění uživatele ze seznamu aktivních práv systému.

Dále je zde možnost určit přihlašovací jméno uživatele a jeho heslo zadáním stejného hesla do obou polí pro heslo. Je zde však podmínka na jedinečnost přihlašovacího jména. Pokud uživateli zadáte příznak aktivní, tak je možné se pod ním ihned přihlásit po jeho založení.

Editace uživatele je tak trochu odlišná. Zde je formulář rozdělený na dvě části s tím, že první část slouží k editaci údajů pro uživatele a druhá pro změnu hesla. Přihlašovací jméno není možné u uživatele standardně měnit.

2.10.3 Správa dodavatelů

Nyní se přesunu k druhé části aplikace, kde se nacházejí dodavatelé. Ty je možné měnit pomocí volby položky Dodavatelé v hlavním menu. Výpis je opět stejný jako všude v systému, ale je zde přeci jen pár odlišností.

Při zakládání dodavatele je třeba vybrat položku Způsob objednávky, která určuje, jakým způsobem bude objednáváno zboží od dodavatele. Na výběr je celkem pět možností a to zaslání objednávky e-mailem, kde dojde vygenerování e-mailu, který je odeslán na e-mailovou adresu dodavatele a obsahuje informace o tom, jaké zboží a jaký počet zboží má být doručen. Další možnost je vygenerování CSV, nebo XML souboru. I když se jedná o rozdílné struktury, tak princip je zde podobný. Vygenerovaný CSV soubor je možné stáhnout do počítače a dále s ním pracovat.

Zatímco XML soubor je uložený do složky generované na server a dodavatel si jej může stáhnout a zpracovat podle svého uvážení. U XML souboru jsem použil strukturu XML pro přenos objednávek do systému Money S4, takže je s tímto software kompatibilní. Pojmenování souboru vychází z data a názvu dodavatele⁴⁰.

Před samotným objednáváním však musí být v systému určeno, jaké suroviny dodavatel dodává, pod jakým názvem, kódem a za jakou cenu. Tyto informace se zadávají přímo ke každému dodavateli a to pomocí volby Sklad u libovolného záznamu. Touto volbou se zobrazí část, která umožňuje vybrat libovolné suroviny, určit v jakém objemu je dodává dodavatel a pod jakým názvem.

Na základě těchto vazeb pak dochází k zpřístupnění možnosti objednávky zboží od dodavatele, ale také k výpočtu doporučené ceny nabídky a skladové jednotky jako takové. Mimo to je zde počítána doporučená prodejní cena suroviny a také celkové nabídky⁴¹.

Objednávka suroviny je již následně celkem snadná. Stačí ze seznamu vybrat dodavatele, od kterého chcete objednat a pomocí akce Objednat vyvolat příslušný formulář. Do něj vyberete suroviny, které se přejete objednat a jejich množství. Následně stačí objednávku potvrdit a dále jí zpracovat podle toho, jaký způsob objednání je u dodavatele vybrán.

2.10.4 Správa skladu

Skladované suroviny jsou v systému základním prvkem, od kterého se odvíjí další chod celé aplikace. Před nastavováním samotného skladu je nejprve nutné založit jednotlivé kategorie skladu, kam budou suroviny přiřazovány. To je možné udělat přes položku kategorie skladu v hlavním menu.

Kategorie jsou řešené stromovou strukturou, kde každá kategorie může mít libovolný počet podkategorií. Pro zakládání první úrovně kategorií je k dispozici kategorie hlavní, která je označena jako nultá a není možné ji v systému vyžít jinak, než pro navázání dalších kategorií. Správa kategorií skladu je jednoduchá. Jediné, co by zde mohlo působit zmatečně, je volba Zobrazit v nabídce. Po jejím zatržení se pak budou suroviny zobrazovat v možnosti sestavení nabídky pod danou kategorií. Tato volba se však vztahuje pouze na danou kategorii a již ne na podkategorie. Je tedy třeba provádět toto nastavení pečlivě.

Po nastavení kategorií je možné nastavit samotný sklad. To je možné provést pod volbou sklad z hlavního menu. Po jejím otevření dojde k zobrazení výpisu všech skladovaných surovin. Záznamy zde je možné přidávat, editovat, naskladňovat, vyskladňovat, deaktivovat a odebírat. Odebírání je však možné pouze v případě, že se daná skladová položka nevyskytuje v žádné nabídce. Jinak tato volba není dostupná.

⁴⁰ Tato hodnota nebude reálně použitelná pro přenos. Před propojením objednávek s dodavatelem je třeba ujasnit, kde bude soubor nahráván a jak bude pojmenován.

⁴¹ Toto nastavení musí být povoleno v souboru konfigurace.inc.

Z možností při založení suroviny do skladu bych rád upozornil na možnost minimálního skladového množství, kam je možné zadat počet jednotek, které musí být vždy skladem. Jak hranice klesne pod tuto úroveň, tak dojde ke hlášení o tom, že je třeba doplnit zásobu dané suroviny. Aktuální stav je pak možné prohlížet v podmenu pod položkou Stav skladu.

Dále je zde možnost prohlížet veškeré pohyby na skladech, a to v podmenu pod položkou Operace skladu, kde dojde k zobrazení všech akcí, které byly se skladem vykonávány a to s možností filtrování podle výdejových a příjmových akcí a časových filtrů. Mimo operací skladu je v podmenu k dispozici akce Stav skladu, která vypíše seznam surovin, kde se skladová zásoba blíží minimálnímu množství. Výpis probíhá ve dvou fázích. Pokud je hranice aktuálního stavu skladu nižší než hodnota nastavená v konfiguraci⁴² tak je položka zařazena jako kritická. Pokud je aktuální stav skladové zásoby suroviny nižší, než je stanovené minimum v databázi, ale nebyla dosažena kritická hodnota 50 procent zůstatkové zásoby, tak dochází k zobrazení této položky jako nutné suroviny pro objednání.

Při naskladňování je možné naskladnit libovolné množství na úrovni skladované jednotky. Pokud je skladován nápoj v centilitrech, tak je možné naskladnit libovolný počet decilitrů dané suroviny. S tím, že vyskladňování je pak na podobném principu.

Editace suroviny je pak podobná jako v případě zakládání záznamu. Je zde však omezení na změnu jednotky, kterou je možné měnit vždy v rámci jednoho typu, aby bylo možné jednotky převádět. Při uložení, kde je odeslána jiná jednotka, než byla původní, dojde k přepočítání skladové hodnoty na danou jednotku.

2.10.5 Správa jednotek

Správa jednotek slouží pro určení měrného systému napříč celou aplikací a je jedinou částí aplikace, která je standardně přednastavena. Tato část systému slouží k tomu, aby bylo možné v systému využívat i netradiční jednotky a přímo si je i do systému založit.

Modul sám o sobě je podobný jako všechny ostatní při zakládání nové položky. Co je však mírně odlišné, je samotný formulář. Mimo to pochopit logiku jednotek není vždy úplně snadné. Nejprve k zadávání údajů do formuláře pro zakládání nebo editaci jednotek. Pro založení jednotky je třeba doplnit název jednotky, její zkratku, která je pak zobrazována v systému u jednotlivých položek, informaci o tom, zda se jedná o objemovou, hmotnostní nebo kusovou jednotku, určení, zda je jednotka základní pro celou soustavu a nakonec převodní poměr mezi jednotkami.

Princip je pak následující. Každý typ jednotky může mít pouze jednu základní jednotku, která má převodní poměr 1. U každé další zadávané jednotky, která není jednotkou hlavní, se pak zadává poměr, který určuje vztah k hlavní jednotce. Tímto vztahem jsou pak veškeré jednotky v systému mezi sebou přepočítávány.

⁴² Hodnota je nastaven v konfiguraci aplikace.

Jednotky a jejich převody jsou využívány v objednávkách od dodavatelů, kde je určeno v jakých jednotkách dodavatel dodává, ve skladu, kde je informace o tom jaké jednotky jsou skladované a nakonec ve vazbě na nabídku, kde je určeno v jakých jednotkách je surovina v nabídce a tím i kolik jednotky bude po potvrzení objednávky na danou objednávkou vyskladněno.

Je zde však podmínka, aby jednotky k dané surovině byly ve všech částech převoditelné. Jinak by nebylo možné určit poměr, který má být převáděn.

2.10.6 Tvorba nabídky

Tvorba nabídky je přímo spojená se skladem. Aby bylo možné tvořit nabídku zařízení, tak je třeba definovat nejprve skladové položky a zařadit je k takovým skladovým kategoriím, které mají aktivní příznak, že položky této kategorie jsou určené pro sestavení nabídky. Pokud jsou dostupné položky skladu, tak je nutné ještě definovat kategorie nabídky. V této části systému je počítáno s kategoriemi jako například Nápoje, Nealkoholické nápoje a další. Systém je stavěn tak, aby bylo možné kategorie zadávat jako strom, kde je možné určit v jedné kategorii více podkategorií.

Po definování všech náležitostí je možné zakládat samotnou nabídku. Ta je k nalezení pod položkou Nabídka, která se nachází v hlavním menu. Po přechodu do této části dojde k zobrazení výpisu tak, jako jinde v systému. Možnost přidání položky se opět nachází v podmenu jako u jiných modulů. Samotný formulář je však trochu rozdílný. Nabídky je nutné skládat z jednotlivých skladových položek a zadat jí základní údaje. Jako první je tedy třeba vyplnit unikátní název nabídky, cenu, za jakou bude v systému prezentována a kategorii, kde bude nabídka zařazena.

Po vyplnění těchto polí je třeba vybrat složení nabídky podle skladových položek. Ty jsou zobrazené ve výběrových polích níže a rozřazeny podle jednotlivých kategorií skladu. Po výběru položky se pak v další části formuláře zobrazí vybraná položka, kam je nutné zadat, jaké množství dané suroviny a v jakých jednotkách bude položka evidována. Tato hodnota pak slouží pro odečítání stavu skladu při objednání nabídky. Po vyplnění všech polí stačí jen nabídku uložit a tím se zpřístupní pro objednávání.

V samotném výpisu nabídek není nic neobvyklého. Jediným, co by mohlo být nejasné, je skrývání možnosti odebrání u některých nabídek. Nabídku je možné odebrat jen do doby, než je objednána. Ve chvíli vytvoření objednávky na nabídku jí dále půjde jen skrýt, ale odebrání není možné.

2.10.7 Rozložení stolů

Správa rozložení jednotlivých stolů je celkem snadná. Stačí přejít do položky Stoly z hlavního menu a dojde k zobrazení základní plochy, která znázorňuje plochu restaurace. Pomocí tlačítka Nový stůl dojde k přidání nového stolu na pracovní plochu. Nová položka nese název Nový a je určena na defaultní pozici.

Změna názvu stolu, jeho pozice na ploše a velikost se ukládá automaticky ihned po změně. Stačí tedy kliknout na libovolnou část stolu a tažením stůl umístit kam je třeba. Přes pravý dolní roh je poté možné tažením měnit velikost stolu a pro změnu názvu stačí prostě jen přepsat pole s názvem.

Název je vhodné udržovat jedinečný, protože slouží obsluze k identifikaci stolu při vytváření objednávky. Stejně tak název nesmí být prázdný. Pokud dojde ke vložení prázdné hodnoty, tak se neuloží žádná akce prováděná při editaci stolu.

Dále je možné stůl odebrat tlačítkem odstranit, které se nachází přímo na ploše stolu⁴³ nebo deaktivovat. Deaktivací se stůl přesune do seznamu, který je možné zobrazit druhým tlačítkem v pomocném menu. Z tohoto seznamu je pak stůl možné znovu aktivovat tak, jako se provádí aktivace napříč celým systémem.

2.10.8 Tvorba objednávek

Pro založení nové objednávky musí být dostupná nabídka, na kterou je objednávka vytvářena a stůl, který bude sloužit pro navázání objednávky. Pokud jsou tyto podmínky splněné, pak je možné začít se zakládáním samotné objednávky.

Objednávku založíte pomocí tlačítka Nová⁴⁴, které se nachází na poli konkrétního stolu. Po volbě této akce dojde k zobrazení formuláře, kde je nutné vybrat položky nabídky, které budou součástí objednávky. Po volbě konkrétního záznamu dojde k zobrazení rozšíření, které umožní vybrat ještě počet kusů dané nabídky. Po vybrání, vyplnění všech záznamů a uložení objednávky dojde k přesměrování zpět na výpis stolů.

Přehled nabídek je možné zobrazit pomocí dvou způsobů. Buď přes tlačítko Výpis objednávek, které je nad rozložením stolů nebo přes možnost Otevřené u konkrétního stolu. Oba způsoby vedou k zobrazení výpisu otevřených objednávek v systému s možností přechodu do samotné objednávky.

V detailu objednávky je pak možné provést její storno, potvrzení, částečné potvrzení, nebo částečné storno, nebo dodání dalších položek nabídky podle toho jakou akci vyžaduje samotný provoz. U akcí, které nepracují objednávkou jako celkem, je pak vždy nutné zadat, s jakou částí nabídky v objednávce bude dále pracováno.

2.11 Instalační příručka

Vzhledem k tomu, že jsem v průběhu vývoje aplikace musel neustále mazat databázi a znovu ji plnit, jsem se rozhodl si tento proces automatizovat. Vytvořil jsem jednoduchý instalátor, který má pouze dvě funkce. Jednou se celá databáze maže a druhou se plní. I když se jedná o poměrně primitivní nástroj, tak jsem jej v aplikaci ponechal, aby ho bylo možné snadno zprovoznit na libovolném hostingu. Před tím, než je možné instalátor použít, je nutné provést ještě další kroky, které jsou popsány níže.

⁴³ Pouze pokud na stůl neexistuje objednávka v jakémkoliv stavu. V takovém případě je možné stůl pouze deaktivovat.

⁴⁴ Objednávka je vždy zakládána pod aktuálním uživatelem systému.

2.11.1 Webhosting a základní konfigurace

Aplikace je tvořena v PHP a jako databázi jsem zvolil MySQL, protože je běžně dostupná na většině hostingů. Pro počátek instalace tedy potřebujete optimální hosting, který bude poskytovat dobrou dostupnost, pravidelnou zálohu dat, bude podporovat PHP minimálně verze 5 a umožní vytvořit MySQL databázi. Paměťové nároky nejsou v případě mého systému nijak závratné. Běžný hosting poskytuje v základním balíčku úložný prostor v řádech desítek GB. Pro instalaci aplikace by však mělo stačit zhruba 5 MB a pak rezerva na samotná data. I v případě velkého množství dat by však měla stačit kapacita do 200 MB.

Pokud je k dispozici vhodný hosting, tak jako první krok nahrajte obsah složky RESYS, která je k dispozici na disku přiloženém k této práci, do kořenového adresáře hostingu. Soubory v této složce jsou již uspořádané dle potřeby a není nutné je nijak měnit.

Před samotnou instalací je třeba vytvořit databázi, která bude podporovat kódování UTF-8 a nastavit k ní potřebné přihlašovací údaje. Dále pak otevřete soubor s názvem konfigurace.inc.php ve složce editor/konfigurace. Jedná se o hlavní konfigurační soubor, který zprostředkovává připojení do databáze a umožňuje další nastavení.

V konfiguračním souboru vás nyní zajímá část “nastavení pro instalaci”. Jako první nastavte databázové údaje. Proměnná HOST_MYSQL určuje IP adresu databáze a jméno databáze.

UZIVATEL_MYSQL je pak pro uživatelské jméno přístupu k databázi a HESLO_MYSQL pro vámi zvolené heslo k databázi. Údaje, které zadáváte, jste zvolili při nastavování databáze. Jedinou výjimkou je IP adresa, kterou bude možné zjistit od poskytovatele hostingu.

Nyní jste ve fázi, kdy je nastavené připojení do databáze a je třeba nastavit cesty k souborům. Jako první nastavte proměnnou CESTA_EDITOR, která má vést do složky editor. Pokud máte systém nainstalovaný přímo do kořenové složky serveru, tak stačí zvolit jako konec cesty /editor. Stejným způsobem pak nastavte i cestu URL_EDITOR, která má obsahovat hodnotu ve tvaru <http://vasedomena.cz/editor>.

Volitelně pak můžete přenastavit proměnnou HESLO_INSTALACE, která určuje heslo pro provedení vytvoření databázových tabulek nebo jejich odstranění. Toto heslo je standardně nastavené na hodnotu “resys”. Změnu tohoto hesla však doporučuji, protože tím zabezpečíte systém před případným útokem od někoho, kdo systém zná a ví defaultní heslo. Nyní můžete konfigurační soubor uložit a zavřít.

2.11.2 Spuštění instalátoru

Nyní přejděte na adresu vasedomena.cz/editor/instalator⁴⁵, kde se vám zobrazí pole pro heslo a dvě tlačítka. Do pole pro heslo zadejte zvolené heslo z konfigurace a zvolte možnost “Instaluj”. Po několika málo okamžicích dojde k založení všech potřebných tabulek, pohledů a naplnění základních dat. Stejným postupem pak lze celou databázi také odinstalovat.

Pokud se vám v průběhu instalace zobrazí další pokyny, tak postupujte podle nich. V opačném případě lze nyní přejít na adresu vasedomena.cz/editor a tím dojde k zobrazení přihlašovacího formuláře do systému. Defaultní jméno administrátora pro první přihlášení je admin a heslo je stejné. Po instalaci jsou k dispozici pouze základní data nutná pro běh aplikace. Pro využití všech možností systému je tedy nejprve nutné projít jednotlivé části a vložit vlastní požadovaná data.

⁴⁵ Za předpokladu, že byla dodržena stejná struktura souborů jako na přiloženém CD.

Závěr

Počáteční návrh aplikace, kterou jsem vytvořil, se vztahoval na co možné nejuniverzálnější pokrytí všech oblastí, jenž obnáší problematika provozu stravovacích zařízení. Postupem času se však návrh začal stávat velice rozsáhlý a bylo nutné jej dále členit a rozšiřovat. Vzhledem k tomu, že vybrané téma jsem volil na základě znalosti provozů restaurací, většina návrhů vznikala z již jasné představy o tom, jak bude systém fungovat.

V průběhu návrhu jsem si udělal jen zběžný přehled o tom, jaké systémy jsou na dnešním trhu dostupné, abych nebyl ovlivněný stávajícími funkcemi, které jsou v podobných systémech využívány. Tento krok se postupem práce ukázal jako přínosný, protože jsem nezávisle na ostatních aplikacích vytvořil takový systém, který obsahuje všechny potřebné vlastnosti jako souhrn modulů, které spolu vytváří mohutný, ale lehce upravovatelný systém.

Celkový rozsah systému pokrývá veškeré běžné procesy restaurací a dokáže na ně pružně reagovat. Díky použitému návrhu je možné systém snadno rozšířit nebo upravit pro specifické potřeby každého provozu a to bez nutnosti složitého zjišťování stávajících vlastností. I bez dalších úprav se nyní jedná o rozsáhlou aplikaci, která by měla stačit opravdu každému provozu.

Z pohledu splnění zadání této práce bylo splněno vše podle zadání. Bohužel však některé funkce jako například predikce skladových zásob bude možné prohlásit za přínosnou až po několika měsíčním provozu aplikace, protože je závislá na velkém množství reálných dat a jejich následném porovnání se skutečnou potřebou skladových zásob.

Vytvořený systém je připravený k provozu, do kterého bude v blízké době i reálně zaveden. Již nyní mám k dispozici dva zájemce, kteří svolili, že se budou podílet na testování systému v běžném provozu. Předpokládám, že tento provoz odhalí skryté nedostatky, které se v rámci stávajícího testování neprojeví. Nemyslím však, že se bude jednat o klíčové části, ale spíše o úpravu uživatelského rozhraní systému.

Pokud se na celou práci podívám jako na celek, tak jsem až sám překvapen, jak rozsáhlý a unikátní systém se mi podařilo v rámci zadání vytvořit. Aplikace, která by byla v podobném rozsahu dostupná, na českém trhu není. Mimo to se jedná pravděpodobně o jedinou takto rozsáhlou aplikaci, která je stavěna pro běh na serveru a tím by byla pro obsluhu dostupná z libovolného počítače.

Práci hodnotím jako přínosnou a to jak z pohledu mého osobního rozvoje v oblasti tvorby rozsáhlých systémů, tak z pohledu vytvoření systému, který je univerzální a na dnešním trhu unikátní.

Literatura

A.W.I.S. Správa, systémy s.r.o. 2013. Pokladní systémy pro restaurace, bary, kluby a diskotéky. *AWIS pokladní systémy*. [Online] 2013. [Citace: 04. 03. 2014.] <http://www.kasa-pokladna.cz/cs/pokladni-systemy-pro-restaurace-kluby-bary>.

Agnis s.r.o. 2013. Pokladna - restaurační software. *Hotelové a restaurační informační systémy*. [Online] Agnis, 2013. [Citace: 04. 03. 2014.] http://www.agnis.cz/2_cestina/17_programy/3_texty/20_pokladni-systemy,-registracni-pokladna/.

Almyta systems. 2014. POINT OF SALE SYSTEMS. *Almyta systems*. [Online] 2014. [Citace: 09. 05 2014.] http://systems.almyta.com/Point_of_Sale_Software.asp.

BORONCZYK, Timothy. 2009. *PHP 6, MySQL, Apache: Vytváříme webové stránky*. Brno : Computer press, 2009. 978-80-251-2767-4.

BUREŠOVÁ, Pavla, Ing. 2012. Klub sommelierů. *Vysoká škola hotelová v Praze*. [Online] 2012. [Citace: 05. 04. 2014.] <http://www.vsh.cz/cz/klub-sommelieru-nav41/>.

CÍGLER SOFTWARE. 2014. Conto: pokladní systém pro restaurace. *Money*. [Online] 2014. [Citace: 04. 03. 2014.] <http://www.money.cz/pos/pokladni-software/conto/>.

COCKBURN, Alistair. 2005. *Use Cases: jak efektivně modelovat aplikace*. . Brno : Computer Press, 2005. 80-251-0721-3.

Český statistický úřad. 1994. Doporučení upravující základní ukazatele pro kategorizaci hostinských a ubytovacích zařízení. *cestovni-ruch*. [Online] 07. 12 1994. [Citace: 09. 05. 2014.] <http://www.cestovni-ruch.cz/kategorizace/doporuceni.htm>.

ČURDA, Dušan a HOLUB, Karel. 2004. *Stručné dějiny oborů Potravinářství a hotelnictví*. Praha : Scientia, 2004. 80-7183-292-8.

GROFF, James R. a WEINBERG, Paul N. 2005. *SQL kompletní průvodce*. Brno : Computer press, 2005. 80-251-0369-2.

ID Innovations Incorporated. 2007. Online Support Assistant. *OPOS Drivers*. [Online] ID Innovations Incorporated, 2007. [Citace: 09. 05. 2014.] <http://www.idinnovations.com/webhelp/index.htm#page=clOPOSDrivers.htm>.

KMENT, Zdeněk. 2011. *Hospody a jejich historická úloha v české společnosti, aneb, Hostince v Čechách, šenky na Valašsku a hospody ve Valašském Meziříčí*. Valašské Meziříčí : autor neznámý, 2011. str. 303. 9788025484913.

KOSEK, Jiří. 1999. *PHP - tvorba interaktivních internetových aplikací*. Praha : Grada Publishing, 1999. 80-7169-373-1.

Management Consulting Group, s.r.o. 2008. Školení a vzdělávání pracovníků v cestovním ruchu. *Pohostinství pro cestovní ruch*. [Online] 2008. [Citace: 05. 04. 2014.] <http://www.mmr.cz/getmedia/aeaa9373-332e-4e6d-bf12-972463c0d3da/GetFile32.pdf>.

Massachusetts Institute of Technology. 2002. Cash Register. *Lemelson-MIT*. [Online] Massachusetts Institute of Technology, 05. 2002. [Citace: 08. 05. 2014.] <http://web.mit.edu/invent/iow/ritty.html>.

Menu55. 2013. Pokladní systém pro restaurace a bary. *Menu55*. [Online] 2013. [Citace: 04. 03. 2014.] <http://www.menu55.cz/>.

MLEJNKOVÁ, Lenka. 2005. *Služby společného stravování*. Praha : VŠE, 2005. 80-245-0870-2 .

POSmatic. 2014 . What is Point of Sale (POS)? *Point of Sales*. [Online] POSmatic, Inc. , 2014 . [Citace: 09. 05. 2014.] <http://www.posmatic.com/point-of-sale/what-is-point-of-sale.php>.

PROSECKÝ, Jiří a kolektiv. 1999. *Encyklopedie starověkého Předního východu* . Praha : Libri, 1999. str. 260. 80-85983-58-3.

SOFTWARE AG. 2014. ARIS Express. *ARIS Community*. [Online] 2014. [Citace: 05. 12. 2013.] <http://www.ariscommunity.com/aris-express/tutorials>.

SOUČKOVÁ, Jitka. 2009. GASTROPROFESOR. *Gastronomie je týmová hra*. [Online] LINKMAN MEDIA s.r.o. , 2009. [Citace: 02. 05. 2014.] http://www.gastroprofesor.cz/clanek/gastronomie_podnik_h372022322.

STANÍČEK, Petr. 2003. *CSS Kaskádové styly*. Brno : Computer press, 2003. 80-7226-872-4.

STEPHENS, Ryan a PLEW, Roland. 2004. *Naučte se SQL za 21 dní*. Brno : Computer Press, 2004. 80-7226-870-8.

VRÁNA, Jakub. 2012. *1001 tipů a triků pro PHP*. Brno : Computer Press, 2012. 978-80-251-2940-1.

World Wide Web Consortium. 2014. PHP 5 Tutorial. *W3Schools*. [Online] World Wide Web Consortium, 2014. [Citace: 09. 02. 2014.] <http://www.w3schools.com/PHP/>.

ZAEFFERER, Jörn. 2013. validate(). *jQuery Validation Plugin*. [Online] 23. 05. 2013. [Citace: 06. 02. 2014.] <http://jqueryvalidation.org/validate/>.

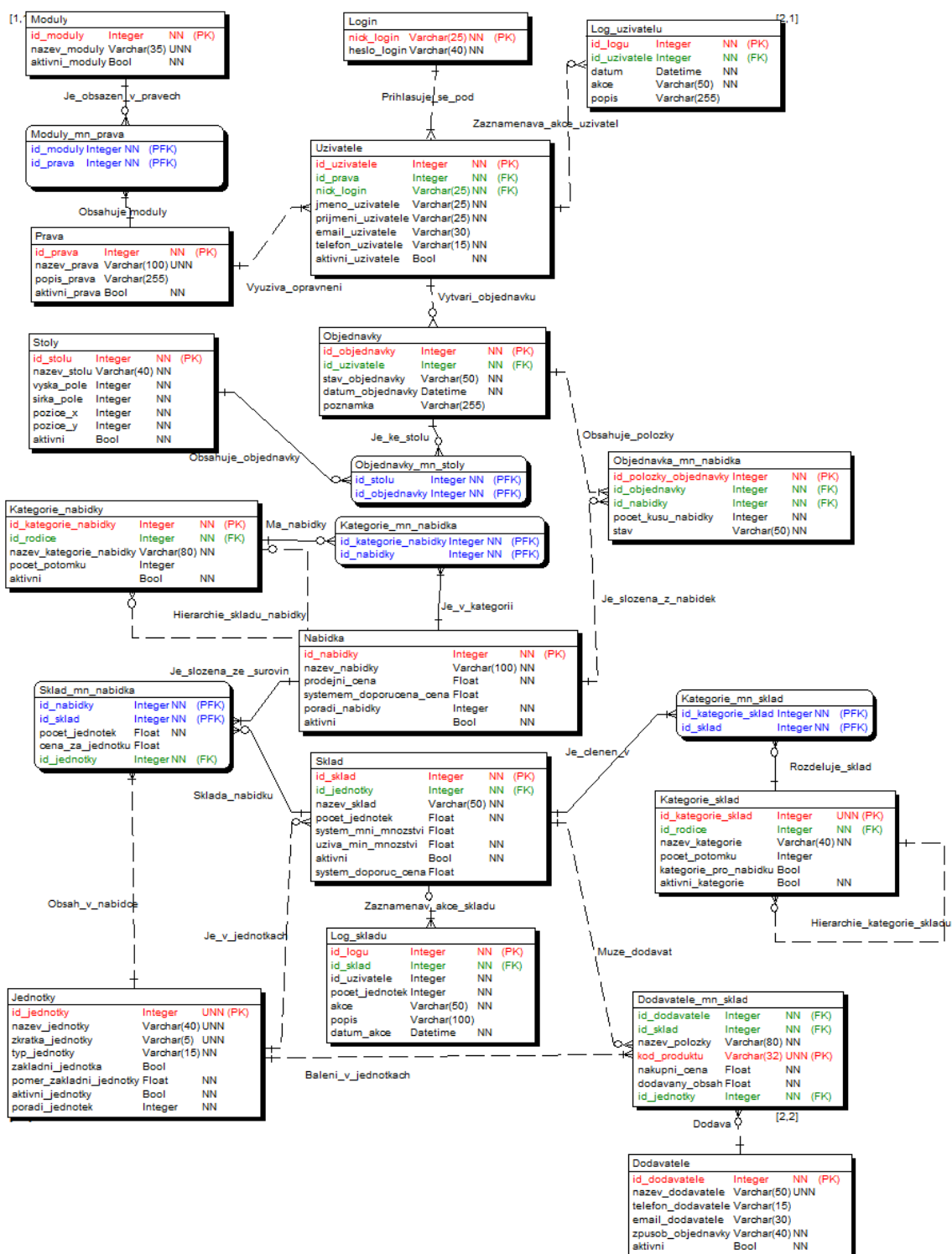
Příloha A – Přehled funkčních a nefunkčních požadavků aplikace

Visual Paradigm for UML Community Edition [not for commercial use]

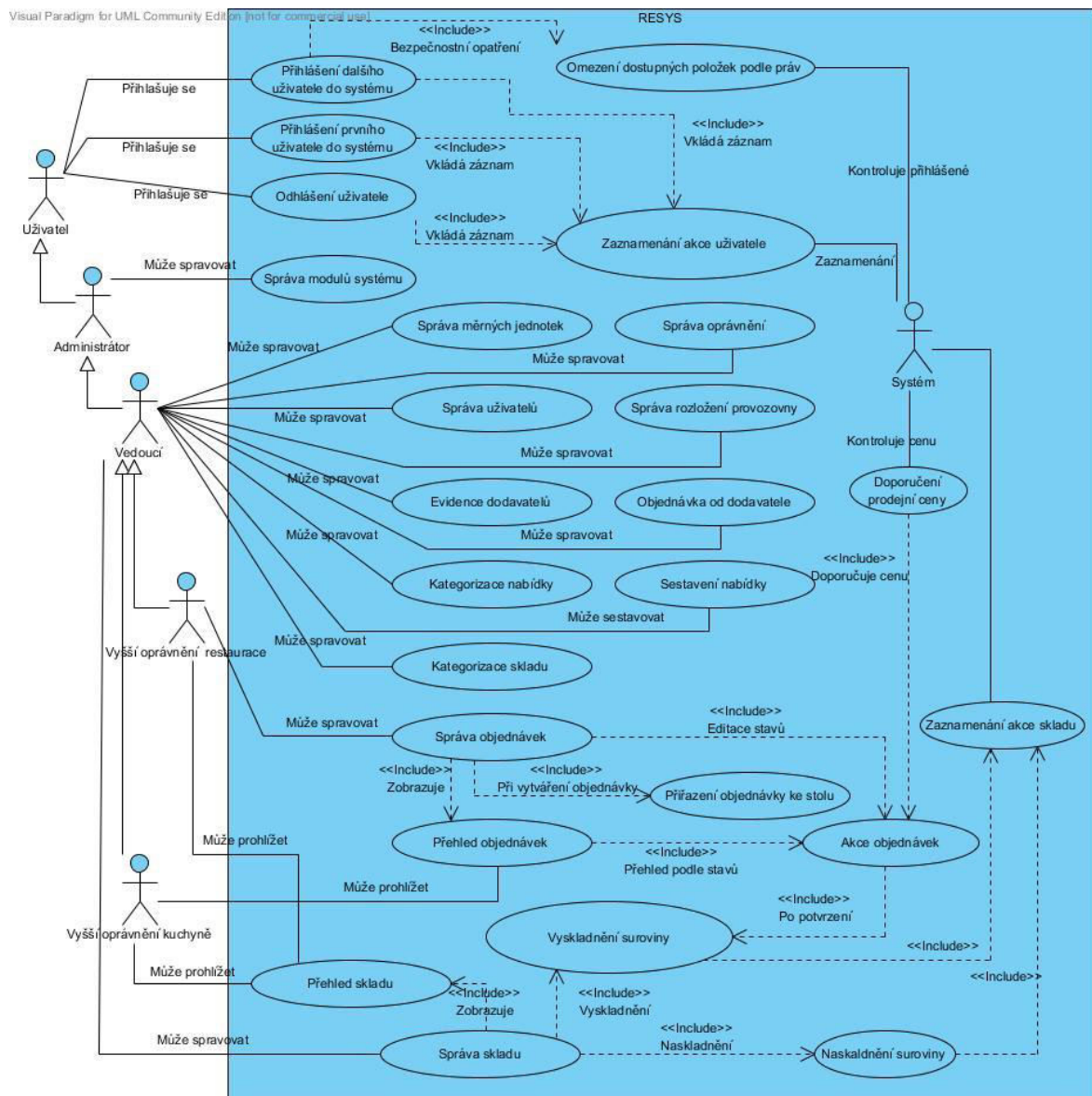
Funkční požadavky		
<<requirement>> Přístup do systému Text = "Se systémem nebude možné pracovat bez přihlášení."	<<requirement>> Přihlášení více uživatelů Text = "Do systému bude přihlášených více uživatelů na jednom stroji"	<<requirement>> Omezení přístupu podle práv Text = "Po přihlášení více oprávnění musí být dostupné pouze funkce toho nejvyššího."
<<requirement>> Evidence akcí uživatelů Text = "Systém bude evidovat veškeré akce přihlášení a odhlášení."	<<requirement>> Správa měrných jednotek Text = "Systém umožní definování vlastních měrných jednotek a jejich převodového systému."	<<requirement>> Správa skladu Text = "Systém umožní evidenci vlastních surovin ve skladu a jejich náhled."
<<requirement>> Naskladnění surovin Text = "Systém umožní naskladnění libovolného množství jednotek suroviny."	<<requirement>> Vyskladnění surovin Text = "Systém umožní vyskladnění libovolného množství suroviny ze skladu."	<<requirement>> Evidence akcí skladu Text = "Systém bude evidovat všechny akce naskladnění a vyskladnění."
<<requirement>> Kategorizace skladu Text = "Systém umožní vlastní kategorizaci skladových položek."	<<requirement>> Evidence dodavatelů Text = "Systém umožní evidenci dodavatelů a způsobu jakým lze u dodavatele objednat"	<<requirement>> Objednávka od dodavatelů Text = "Systém umožní snad objednat suroviny od dodavatele ve zvoleném formátu."
<<requirement>> Doporučení ceny Text = "Systém dopočítá doporučenou cenu suroviny na základě ceny dodavatele"	<<requirement>> Skládání nabídky Text = "Systém umožní složit nabídku ze surovin zařízení včetně určení množství suroviny."	<<requirement>> Kategorizace nabídky Text = "Systém umožní vlastní kategorizaci nabídky."
<<requirement>> Správa rozložení provozovny Text = "Systém umožní vlastní rozložení stolů provozovny."	<<requirement>> Objednávka z nabídky Text = "Systém umožní objednání libovolného množství položek nabídky."	<<requirement>> Objednávka ke stolu Text = "Systém umožní přiřazování nabídek ke stolům."
<<requirement>> Přehled objednávek Text = "Systém umožní zobrazit přehledy objednávek. Výnosů a nejžádanějšího sortimentu"	<<requirement>> Stavy objednávek Text = "Systém bude evidovat u objednávek různé stavy."	<<requirement>> Snižování skladu Text = "Systém bude snižovat sklad po potvrzení každé objednávky"
<<requirement>> Správa uživatelů Text = "Systém umožní základání a editaci uživatelů."	<<requirement>> Správa oprávnění Text = "Systém umožní vlastní definici rolí systému."	

Nefunkční požadavky	
<<requirement>> Dostupnost Text = "Systém musí být dostupný 99,5% svého běhu"	<<requirement>> Použitelnost Text = "Systém musí být snadno ovladatelný i pro uživatele bez IT znalostí"
<<requirement>> Odezva Text = "Systém musí mít v často využívaných částech odezvu do 3 sekund"	<<requirement>> Bezpečnost Text = "Systém musí být zabezpečen proti útoku a ztrátě dat"
<<requirement>> Udržitelnost Text = "Rozvržení systému a dokumentace pro údržbu systému"	<<requirement>> Rozšiřitelnost Text = "Systém musí být rozšiřitelný, nebo modifikovaný bez ovlivnění stávajících funkcí"
<<requirement>> Provoz Text = "Systém musí běžet na většině běžných počítačů."	

Příloha B – ER diagram



Příloha C – Use Case Diagram



Příloha D – Část metody pro vložení doplňujících polí nabídky

```
if (je_stejna_slozka) {

    selectList += '<input type="text" name="slozky[]" id="" value="' +
    slozeni[i] + '" readonly>';
    selectList += '<input type="text" name="cehoText[]" id="" value="'
    + textySlozeni[i] + '" readonly>';

    selectList += '<input type="text" name="mnozstvi_slozky[]" id=""
    value="' + mnozstvi[pozice_seznamu] + '">';
    selectList += '<select name="jednotka_slozky[]">';

    for (var x = 0; x < jednotky_id.length; x++) {

        var je_nastavena = false;

        if (jednotky_id[x] === jednotkyPrednatsavene[pozice_seznamu]) {

            je_nastavena = true; }

        if (je_nastavena) {
            selectList += "<option value=" + jednotky_id[x] + " selected
            >" + jednotky_hodnoty[x] + "</option>";}

        else {
            selectList += "<option value=" + jednotky_id[x] + ">" +
            jednotky_hodnoty[x] + "</option>"; } } }

        else {
            selectList += '<input type="text" name="slozky[]" id=""
            value="' + slozeni[i] + '" readonly>';
            selectList += '<input type="text" name="cehoText[]" id=""
            value="' + textySlozeni[i] + '" readonly>';
            selectList += '<input type="text" name="mnozstvi_slozky[]"
            id="" value="1">';
            selectList += '<select name="jednotka_slozky[]">';

            for (var x = 0; x < jednotky_id.length; x++){
                selectList += "<option value=" + jednotky_id[x] + ">" +
                jednotky_hodnoty[x] + "</option>";}}

        selectList += "</select><br />";}
$('#seznamJednotek').html(selectList);
```