

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Informační systém potravinářské firmy

Petr Ruttkay

© 2021 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Petr Ruttkay

Systémové inženýrství a informatika
Informatika

Název práce

Informační systém potravinářské firmy

Název anglicky

Food company information system

Cíle práce

Cílem této bakalářské práce je navržení informačního systému potravinářské firmy s využitím objektového jazyka PHP. Systém bude pracovat rovněž s databází MYSQL. Všichni uživatelé budou pracovat se stejnou databází. Systém by měl být přívětivý k uživateli a měl by poskytovat informace, které uživatel potřebuje.

Metodika

V první řadě je potřeba provést analýzu potřeb dané firmy. Na základě výsledku budou vytvořeny požadavky dané firmy na informační systém.

Pro analýzu a návrh se použije UML.

Poté co budeme mít kompletně vytvořený návrh se započne s implementací.

Systém se bude implementovat za použití jazyka PHP. Při vytváření systému se bude klást důraz na všechny potřeby dané firmy.

Systém by měl být pro uživatele přehledný přívětivý. Uživatel by měl být informovaný ohledně zboží. K tomu bude využita databáze MYSQL, kde budou uloženy data ohledně produktů, které daná firma poskytuje.

Doporučený rozsah práce

30-40 stran

Klíčová slova

informační systém, potravinářská firma, PHP, databáze, MYSQL

Doporučené zdroje informací

- BASL, J. – BLAŽÍČEK, R. – ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI. *Podnikové informační systémy : podnik v informační společnosti*. Praha: Grada, 2008. ISBN 978-80-247-2279-5.
- BÖHMER, M. *Návrhové vzory v PHP : [23 vzorových postupů pro rychlejší vývoj]*. Brno: Computer Press, 2012. ISBN 978-80-251-3338-5.
- GILMORE, W J. – POKORNÝ, J. *Velká kniha PHP 5 a MySQL : kompendium znalostí pro začátečníky i profesionály*. Brno: Zoner Press, 2011. ISBN 978-80-7413-163-9.
- KOSEK, J. *PHP a XML*. Praha: Grada, 2009. ISBN 978-80-247-1116-4.
- SODOMKA, P. – KLČOVÁ, H. *Informační systémy v podnikové praxi*. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.

Předběžný termín obhajoby

2021/22 ZS – PEF

Vedoucí práce

Ing. Marek Pícka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelíkán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelíkán, Ph.D.

Děkan

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Informační systém potravinářské firmy" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.11.2021

Poděkování

Rád bych touto cestou poděkoval Ing. Markovi Píckovi, Ph.D. za cenné rady, pomoc, ochotu a trpělivost při zpracování mé bakalářské práce. A dále bych chtěl poděkovat svým rodičům za podporu a velikou trpělivost během mého studia.

Informační systém potravinářské firmy

Abstrakt

Tato bakalářská práce je zaměřena na vytvoření informačního systému. Ten bude pracovat s vlastní databází. Následná aplikace je zobrazena ve webovém prostředí. Teoretická část se zabývá použitými technologiemi pro tvorbu IS na webu a dále rozšiřuje povědomí ohledně IS.

Praktická část je zaměřena na analýzu potřeb potravinářské firmy. Z toho vychází UML diagram s potřebnými třídami. Podle schéma se vytvoří databáze vyplněná tabulkami a potřebnými atributy. Další krok už je samotná implementace informačního systému v PHP. Pomocí PHP funkcí se XML formuláře zobrazí na HTML stránkách, které uživatel vidí ve webovém prohlížeči.

Ke konci jsou zobrazeny a popsány části systému z vizuálního pohledu.

Klíčová slova: informační systém, potravinářská firma, PHP, databáze, MYSQL, internet, HTML, CSS, UML, XML

Food company information system

Abstract

This bachelor thesis focuses on the creation of an information system. It will work with its own database. The subsequent application is displayed in a web environment. The theoretical part deals with the technologies used to create IS on the web and further expands the awareness regarding IS.

The practical part focuses on the needs analysis of a food company. From this, a UML diagram with the necessary classes is developed. According to the diagram, a database filled with tables and necessary attributes is created. The next step is the actual implementation of the information system in PHP. Using PHP functions, the XML forms are displayed on HTML pages that the user sees in a web browser.

Towards the end, parts of the system are displayed and described from a visual perspective.

Translated with www.DeepL.com/Translator (free version)

Keywords: information system, food company, PHP, database, MYSQL, internet, HTML, CSS, UML, XML

Obsah

1 Úvod.....	13
2 Cíl práce a metodika	14
2.1 Cíl práce	14
2.2 Metodika	14
3 Teoretická východiska	15
3.1 Informační systém firmy	15
3.1.1 Změna ve využívání počítačové techniky.....	15
3.1.2 Druhy informačních systémů.....	16
3.1.3 Základní klasifikace informačních systémů	20
3.2 Potravinářství	22
3.2.1 Marketingový informační systém.....	22
3.2.2 Interní databáze:.....	23
3.3 TECHNOLOGIE PRO TVORBU IS ve webovém prostředí	24
3.3.1 Apache	24
3.3.2 HTML	25
3.3.3 Java script	26
3.3.4 JQuery.....	27
3.3.5 CSS	27
3.3.6 PHP.....	29
3.3.6.1 Eclipse IDE for PHP Developers	29
3.3.7 MySQL	29
3.3.7.1 HeidiSQL.....	30
3.3.8 UML.....	30
4 Vlastní práce	32
4.1 Požadavky na IS	32
4.1.1 Dosledovatelnost.....	32
4.1.2 Jakostní údaje mouky a obilí	32
4.1.3 Normy spotřeb (receptury).....	33
4.1.4 Složení výrobku a nutriční údaje	33
4.2 Řešení Desledovatelnosti potravinu	34
4.2.1 Identifikace šarže produktu.....	35
4.3 UML diagramy	37
4.4 Popis databáze.....	39
4.5 Softwarová realizace IS.....	46
4.5.1 Struktura definice formuláře v XML	46

4.5.2	Programová část.....	50
4.6	Tiskové výstupy IS.....	58
4.7	Zabezpečení systému.....	58
4.7.1	Login do IS.....	58
4.7.2	Šifrování.....	59
4.7.3	Přístupová práva k menu IS pro jednotlivé uživatele	59
5	Výsledná aplikace.....	60
5.1	Odkaz na aplikaci IS na webu	60
5.2	Přihlášení do aplikace.....	60
5.3	Rozvržení XML formuláře na webu	61
5.4	Funkční tlačítka používané v aplikaci IS	62
5.5	Uživatelé.....	63
5.6	Menu aplikace IS.....	64
5.7	Filtrování (výběr) dat v IS	65
5.8	Zápis příjmového skladového dokladu.....	67
6	Závěr.....	69
7	Seznam použitých zdrojů.....	70
8	Přílohy	71

Seznam obrázků

Obrázek 1: Změny v ISP [1]	15
Obrázek 2: Tříúrovňový pyramidový model [2].....	17
Obrázek 3 - Pyramidový model s pěti úrovněmi [2].....	18
Obrázek 4: Ukázka UML diagramu [24]	31
Obrázek 5: Schéma fungování šarži.....	35
Obrázek 6 - Referenční schéma vazeb mezi tabulkami	37
Obrázek 7: EER diagram	38
Obrázek 8 - Ukázka přihlašovacího HTML formuláře	58
Obrázek 9 - Přihlášení do aplikace	60
Obrázek 10: Rozvržení webové stránky dle XML definice.....	61
Obrázek 11: Funkční tlačítka v IS.....	62
Obrázek 12 - Ukázka editace nebo přidání uživatele IS	63
Obrázek 13 - Ukázka formuláře pro zadání nebo editaci menu.....	64
Obrázek 14 - Ukázka formuláře pro zadání výběru.....	65
Obrázek 15: Příjemka do skladu	67

Obrázek 16: Řádka příjemky	68
----------------------------------	----

Seznam tabulek

Tabulka 1: Příklad jakostních údajů	33
Tabulka 2 - Popis údajů skladové karty	36
Tabulka 3 - Popis šaržových parametrů skladu	36
Tabulka 4: Vybrané tabulky z databáze.....	39
Tabulka 5 - Přehled tagu pro definici formuláře.....	46
Tabulka 6 - Přehled funkčních tlačítek v aplikaci IS.....	62

Seznam použitých zkratk

Ajax – Asynchronous JavaScript and XML
API – Application Programming Interface
BIC – Business Identification Code
BIRT – Business Intelligence and Reporting Tools
CAD – Computer-aided design
CSS – Cascading Style Sheets
DB – Databáze
DBMS – database management system
DMT – Datum minimální trvanlivosti
DOM – Document Object Model
DOM – Document Object Model
DPH – Daň z přidané hodnoty
EAN – European Article Number
EER – Enhanced Entity-Relationship
EU – Evropská unie
FIFO – First In, First Out
HTML – Hypertext Markup Language
HTTP – Hypertext Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
IČO – Identifikační číslo odběratele
ID – Identifikace
IS – Informační systém

JC – Jednotková cena
JS – JavaScript
MIS – Marketingový informační systém
MJ – Měrná jednotka
MYSQL – My Structured Query Language
NS – Norma spotřeby
OM – Odběrní místo
PČ – Pořadové číslo
PHP – Personal Home Page
SQL – Structured Query Language
SSL – Secure Sockets Layer
SW – Software
SWIFT – Society for Worldwide Interbank Financial Telecommunication
TLS – Transport Layer Security
TPS – Transaction Processing System
UML – Unified Modeling Language
UTF – Unicode Transformation Format
XML – Extensible Markup Language
XPath – XML Path Language

1 Úvod

V dnešní době se informační systémy využívají v drtivé většině firem. Mají velkou řadu výhod a funkcí, které firma využívá a bez nich by byl chod při nejmenším velmi složitý a zdlouhavý. Proč ale využívat informační systém na webu? Má to své výhody, jednou z nich je bezpochyby dostupnost systému z kteréhokoliv místa na světě. Spousta světových firem totiž nemá jen jednu pobočku, výrobu a nejsou jen v jedné zemi. Proto je dostupnost informačního systému velmi důležitá a internet je bezpochyby nejlepší místo, kam tento systém umístit. Další výhoda se ukázala v posledních letech, i přestože se tak dalo využívat v dřívější době a tou je tzv. homeoffice. V době pandemie spousta lidí pracovala z domova a byli odkázáni využívat internet jako prostředek, kterým svou práci mohli provádět.

V mé práci jsem se zaměřil na potravinářskou firmu, která jistě nejen tyto výhody využívá. V takové firmě je důležité, aby chod systému byl stabilní. Další vlastnosti, které systém musí splňovat jsou, aby zaměstnanci měli své vlastní přihlašovací údaje, aby bylo viděno, kdo jaký položky upravoval, přidával nebo mazal. Každému uživateli se udělují práva (síla práv pak odráží možnosti zobrazení určitých záložek a jejich následnou úpravu). Proto tento systém není jen pro řadové zaměstnance, ale i pro hlavní pracovníky firmy. V systému také nesmí chybět položky, jako jsou: pohledávky, objednávky, účetnictví, sklad, závazky a administrativa.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této bakalářské práce je navržení informačního systému potravinářské firmy s využitím objektového jazyka PHP. Systém bude pracovat rovněž s databází MYSQL a poběží na webovém serveru Apache. Všichni uživatelé budou pracovat se stejnou databází. Systém je generován v podobě HTML stránek na webu. Uživatel bude obsluhovat systém v prohlížečovém prostředí na webových stránkách.

2.2 Metodika

První část bakalářské práce obsahuje literární rešerši, která rozebírá jednotlivé pojmy související s BP. Bude rozdělená do tří částí, informační systém, potravinářství a samotné technologie k vytvoření informačního systému.

Než se začne vytvářet informační systém, je potřeba provést analýzu potřeb potravinářské firmy. Na základě výsledků bude vytvořen UML diagram s třídami, které budou obsaženy v databázi MySQL. Každá z tříd bude mít svůj vlastní primární klíč a budou navzájem propojené vazebními klíči, které budou odkazovat na tyto primární klíče (jeden vazební klíč odkazuje na jeden primární). Na jeden primární klíč může být napojeno více vazebních klíčů a zároveň tabulka může obsahovat více těchto vazebních klíčů. Poté co bude naplněna databáze potřebnými atributy bude databáze připravena k chodu. Poté se může přejít k samotné implementaci informačního systému.

Pro chod stránek je zvolený webový server Apache. Systém se bude vytvářet v jazyku PHP, kde funkce budou zajišťovat generaci HTML stránek z XML formulářů. Neboli bude použito tzv. trio Apache, PHP a MySQL. Zároveň budou používány CSS styly pro úpravu vzhledu stránek. Při vytváření systému budou také používány Java scripty spolu s JQuery.

Vývojové prostředí bude zvolený pro PHP Eclipse a pro databázi MySQL HeidiSQL. Systém bude mít i zabezpečení formou loginu a práv pro jednotlivé uživatele. Výsledná aplikace bude obsluhována v prohlížečovém prostředí přes webové stránky.

3 Teoretická východiska

3.1 Informační systém firmy

3.1.1 Změna ve využívání počítačové techniky

Hodně významných změn IS a IT proběhlo ve firmách v průběhu produktivního života nynější generace padesátiletých a šedesátiletých. Ti budou v nejbližší době odcházet do důchodu. Počítačová technika v průběhu několika desítek let změnila. Hlavní změna byla zmenšení počítačů. Ze speciálních klimatizovaných místností na tyhle velké stroje, se počítače dostali do kanceláří, kde běžně plní svůj účel. Počítače v průběhu doby, také změnily svůj primární účel, který byl zpočátku výpočetní úkol. Jejich úkol začínal být i jiný, a to být informační bránou k informacím dostupných ve firmě (i v rámci vnějších informačních zdrojů). Nyní technika postoupila, tak že se využívají mobilní telefony. [1]

Obrázek 1: Změny v ISP [1]

	50. léta	60. až 70. léta	80. léta	90. léta
Hlavní technické prostředky IT	sálové počítače	mainframy	osobní počítače počítačové sítě	přenositelná k internetu připojitelná zařízení
Hlavní oblasti využití prostředků IS	vědecko-technické výpočty	hromadné zpracování dat	kancelářský SW podpora inženýrských prací	komunikační nástroj podpora rozhodování

Tab. II.1 Hlavní změny ve využití IS/IT

Pracovní terminály, které se využívaly k zadávání dat a práci, byly nahrazeny v osmdesátých letech pro uživatele lepší osobní počítače. Ty daly stavební kámen „uživatel“ strany komunikace se „serverem“, který pracoval s vlastní aplikací používanou pro firemní informační systém a potřebná data. [1]

Značnou část úkolů od začátku na osobních počítačích byly také i tzv. „office“ programy příhodné pro texty, zhotovení tabulek, grafů nebo různých prezentací. Dále osobní počítač mohli začít sloužit kvůli potencionálu nabízeného multimediálními aplikacemi, které podporují vzdělávání a zároveň daly možnost využití svého volného času například formou četby, sledováním videí, popřípadě hraním her. [1]

3.1.2 Druhy informačních systémů

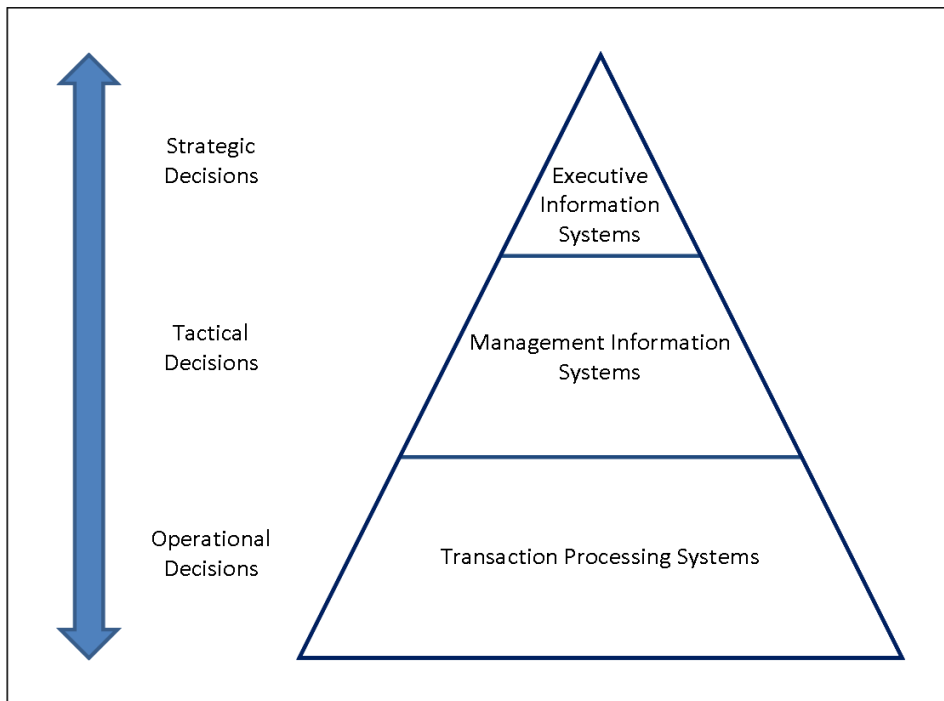
V počátcích výpočetní techniky byl pokaždé, když byl potřeba informační systém, „na míru“ - postaven jako jednorázové řešení konkrétního problému. Brzy však vyšlo najevo, že mnoho problémů, které informační systémy vytyčily k řešení, sdílely určité charakteristiky. V důsledku toho se lidé pokusili zkusit vybudovat jediný systém, který by vyřešil celou řadu podobných problémů. Brzy si však uvědomili, že k tomu je nejprve nutné definovat, jak a kde bude informační systém používán a proč je potřeba. Tehdy začalo hledat způsob, jak přesně klasifikovat informační systémy. [2]

Různé typy informačního systému, které lze najít, jsou identifikovány prostřednictvím procesu klasifikace. Klasifikace je jednoduše metoda, kterou lze věci kategorizovat nebo klasifikovat společně, aby s nimi bylo možné zacházet, jako by to byla jedna jednotka. Klasifikace věcí v přírodním světě, jako jsou rostliny nebo zvířata, má dlouhou historii, nicméně informační systémy nejsou součástí „přirozeného“ světa; jsou vytvářeny a získávány člověkem k řešení konkrétních úkolů a problémů. Klasifikace informačních systémů do různých typů je užitečnou technikou pro návrh systémů a diskusi o jejich aplikaci; není to však pevná definice, která se řídí nějakým přirozeným zákonem. „Typ“ nebo kategorie informačního systému je jednoduše koncept, abstrakce, který byl vytvořen jako způsob, jak zjednodušit složitý problém prostřednictvím identifikace oblastí shodnosti mezi různými věcmi. Jeden z nejstarších a nejpoužívanějších systémů pro klasifikaci informačních systémů je známý jako pyramidový model; toto je podrobněji popsáno níže. [2]

V závislosti na tom, jak vytváříte klasifikaci, můžete najít téměř libovolný počet různých typů informačního systému. Je však důležité si uvědomit, že existují různé druhy systémů nalezených v organizacích, které se zabývají konkrétními problémy a úkoly, které se v organizacích vyskytují. Většina pokusů o klasifikaci informačních systémů do různých typů se tedy spoléhá na způsob, jakým jsou úkoly a odpovědnosti v rámci organizace rozděleny. Protože většina organizací je hierarchických, způsob, jakým jsou různé třídy informačních systémů kategorizovány, má tendenci následovat hierarchii. To je často popisováno jako „pyramidový model“, protože způsob, jakým jsou systémy uspořádány, odráží povahu úkolů nalezených na různých úrovních organizace. [2]

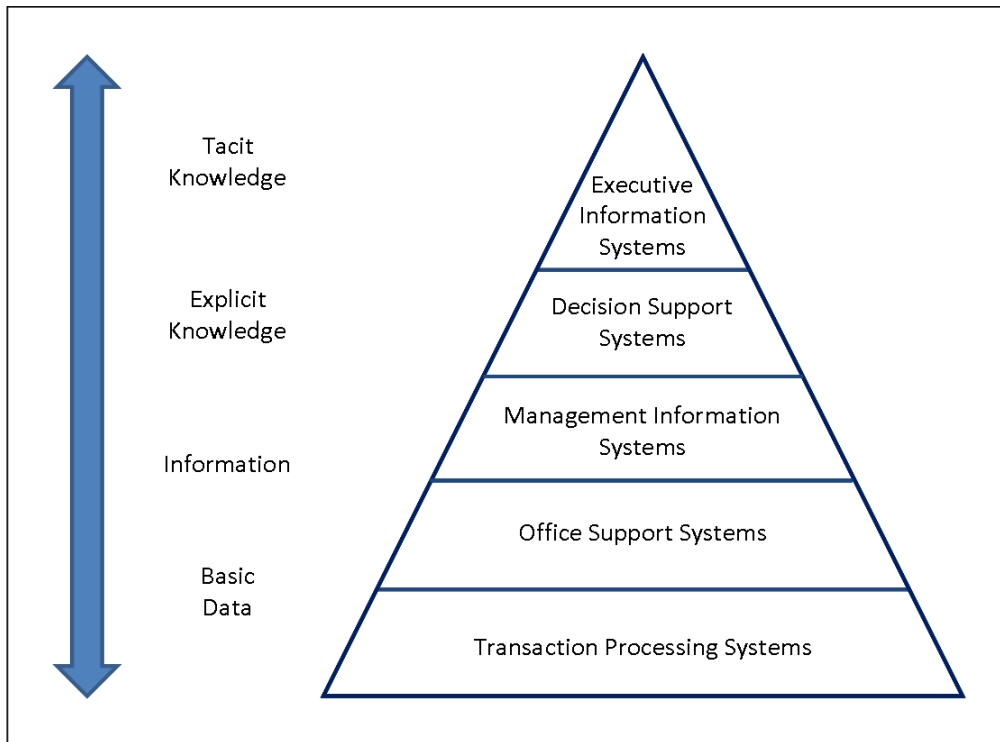
Jedná se například o tříúrovňový pyramidový model založený na typu rozhodnutí přijímaných na různých úrovních v organizaci. [2]

Obrázek 2: Tříúrovňový pyramidový model [2]



Podobně změnou našich kritérií na různé typy dat / informací / znalostí, které jsou zpracovávány na různých úrovních v organizaci, můžeme vytvořit model na pěti úrovních. [2]

Obrázek 3 - Pyramidový model s pěti úrovněmi [2]



I když existuje několik různých verzí pyramidového modelu, nejběžnější je pravděpodobně čtyř úrovnový model založený na lidech, kteří systémy používají. Založení klasifikace na osobách, které používají informační systém, znamená, že více či méně automaticky je zohledněno mnoho dalších charakteristik, jako je povaha úkolu a informační požadavky. [2]

Srovnání různých druhů informačních systémů:

1. Transaction Processing Systems

Transaction Processing System jsou systémy na provozní úrovni ve spodní části pyramidy. Obvykle jsou obsluhovány přímo dělníky nebo pracovníky frontové linky, kteří poskytují klíčová data potřebná pro podporu řízení provozu. Tato data se obvykle získávají automatizovaným nebo poloautomatickým sledováním činností na nízké úrovni a základních transakcí. TPS jsou nakonec o něco více než jednoduché systémy pro zpracování dat. [2]

2. Management Information Systems

Z historických důvodů se mnoho různých typů informačních systémů, které se nacházejí v komerčních organizacích, označuje jako „Management Information Systems“. V rámci našeho pyramidového modelu jsou však Management Information Systems systémy na úrovni managementu, které používají střední manažeři k zajištění hladkého chodu organizace v krátkodobém až střednědobém horizontu. Vysoce strukturované informace poskytované těmito systémy umožňují manažerům vyhodnotit výkon organizace porovnáním proudu s předchozími výstupy. [2]

3. Decision Support Systems

Systém podpory rozhodování lze chápat jako systém založený na znalostech, používaný vrcholovými manažery, který usnadňuje vytváření znalostí a umožňuje jejich integraci do organizace. Tyto systémy se často používají k analýze stávajících strukturovaných informací a umožňují manažerům promítat potenciální dopady jejich rozhodnutí do budoucnosti. Takové systémy jsou obvykle interaktivní a používají se k řešení špatně strukturovaných problémů. Nabízejí přístup k databázím, analytickým nástrojům, umožňují simulace typu „co kdyby“ a mohou podporovat výměnu informací v rámci organizace. [2]

4. Executive Information Systems

Výkonné informační systémy jsou strategické informační systémy, které se nacházejí v horní části pyramidy. Pomáhají vedoucím pracovníkům a vrcholovým manažerům analyzovat prostředí, ve kterém organizace působí, identifikovat dlouhodobé trendy a plánovat vhodný postup. Informace v těchto systémech jsou často slabě strukturované a pocházejí z interních i externích zdrojů. Výkonný informační systém je navržen tak, aby byl provozován přímo vedoucími pracovníky bez potřeby zprostředkovatelů a aby byl snadno přizpůsoben preferencím jednotlivce, který je používá.[2]

3.1.3 Základní klasifikace informačních systémů

Každá firma má několik organizačních úrovní, které vyžadují určitý způsob zpracování informací nebo specifický druh informací. Nejvíce se rozlišují přitom strategická, řídicí, znalostní a provozní úroveň. Ani jedna z těchto úrovní samotná nedokáže poskytnout veškeré informace, které management potřebuje k řízení. Také ale žádná z těchto úrovní není samostatnou jednotkou, která by reflektovala reálnou potřebu nasazení samostatného informačního systému (SW aplikace). Tudiž také často používaná klasifikace, která dělá rozdíl mezi provozními, znalostními, řídicími a strategickými informačními systémy, reflektuje zásadně teoretický náhled na chod podniku. Jejím účelem je charakterizovat hodnotu automatizovaného zpracování informací pro zaměstnance na určitých organizačních úrovních, to znamená že, tento pohled má svůj význam. Avšak není moc správné pojednávat v téhle souvislosti přímo o informačních systémech, zbytečně se tím komplikuje už tak těžké pochopení spousta navzájem se překrývajícím pojmům. [1]

Provozní úroveň:

Vyžaduje zpracování informací, které souvisí s rutinní podnikovou agendou. To jsou realizace výrobních zakázek, nákupu a tržby, přijímání plateb a mzdy apod. Informační systémy na provozní úrovni reagují na splnění každodenních úkolů a sledují tok transakcí přes celou firmu.

Odpovídají na otázky typu: Je na skladě dostatek materiálu pro zakázku? Proběhla konečná transakce s hlavním zákazníkem? Byly výrobky dopraveny na místo určení?

Z příkladů otázek je vidět, že informační systémy na provozní úrovni mají za účel poskytnout přesné, aktuální a lehce dostupné informace. Klasickým uživatelem těchto informací je účetní, provozní zaměstnanec nebo operátor dispečinku pracující se zákaznickou stanicí poskytující rozhraní firemního informačního systému. [1]

Znalostní úroveň:

Obsahuje nejen uživatelské aplikace firemního informačního systému, ale i nástroje osobní informatiky, jako jsou kancelářské aplikace, software, který slouží pro týmovou práci atd. Tyhle aplikace prohlubují znalosti o organizaci a řídí převážně chod dokumentů.[1]

Odpovídají na otázky typu: Jak dopadly poslední schůzky s dodavateli? Jaký je aktuální stav hospodaření firmy? Jak jsou spokojeni zákazníci s našimi produkty v došlé korespondenci? Informace, které se posílají těmito aplikacemi, vytváří potenciální znalosti. Spolu s nimi se utváří zkušenosti zaměstnanců z provozu firmy. Klasickými uživateli aplikací na znalostní úrovni jsou manažeři a techničtí či hospodářští zaměstnanci na všech úrovních. [1]

Řídící úroveň:

Potřebuje informace nutné k děláni administrativních úkolů a podpoře rozhodování, hlavně u středního i vrcholového managementu. Informační systém, který se používá na řídicí úrovni, odpovídá na zásadní otázku: Funguje vše tak, jak má? Odpovědi sděluje formou tzv. reportingu. To je generování výstupních sestav, které obsahují veškeré výsledky z požadované oblasti. Podpora strukturovaného rozhodování skrze hlášení probíhá hlavně v pravidelných časových úsecích. Například lze reportovat ekonomické výsledky z obchodní činnosti.

Důležitou součástí řídicí úrovně je také potřeba reportů pro nevšední rozhodování, u kterého požadavky uživatele nebývají tak zřejmé.

Klasickým příkladem jsou analýzy typu: Co se stane, když? Manažeři pak mají odpovědi na otázky: Jak bychom měli rozvrhnout množství produkce, pokud bychom chtěli mít prodej o 30% vyšší do konce fiskálního období? Za jak dlouho se nám navrátí investice do výrobního zařízení, jestliže se termíny dodávek zpozdí o 3 měsíce? [1]

Strategická úroveň:

Informační systémy ve strategické oblasti bývají vrcholovému managementu užitečný k rozpoznání dlouhodobých trendů (Jak uvnitř, tak mimo organizaci). Jejich podstatou je pomoci odhalit očekávané změny a určit, jestli a jak je firma na změny schopna reagovat. Ke klasickým otázkám, na které dávají informační systémy na této úrovni odpovědi, jsou:

Jak se nejčastěji pracuje s náklady na produkci a jak by to mohlo fungovat v naší organizaci? Jaký výkon podnikových procesů bude potřeba pro odběratele a dodavatele za další dva roky? [1]

Informace pro řídicí i strategické analýzy často mají původ nejen z provozního systému, ale i z externích zdrojů. [1]

K praxi má blízko technologický pohled na firemní informační systémy. Jedná se o rozdělení na základě vrstev, kterými jsou data transformována na informace, které uživatel snáze pochopí. [1]

Možná se zdá, že technologický význam informačního systému není již v současné době tak významný, má však mnoho technických vlivů, které determinují automatizované zpracování dat daným informačním systémem.[1]

3.2 Potravinářství

3.2.1 Marketingový informační systém

Velmi důležitou částí marketingu je sběr a analýza informací a následné jejich využití. Aby firma totiž uspokojila své cíle musí nejdříve uspokojit své zákazníky, a to kvalitněji a rychleji než její konkurenti.

Firma, aby zjistila, co zákazníci zajímá, buduje svou databázi prodeje, zásob, objednávek atd. Tento soubor databází, informačních zdrojů, nástrojů analýzy a prezentace výsledků se nazývá Marketingový informační systém. [3]

3.2.2 Interní databáze:

Přírozenou součástí MIS je interní databáze. Někdy je firma musí mít ze zákona, někdy zase firma je má zřízený v rámci činnosti firmy. Mezi interní databáze patří:

Objednávky – Objednávka je pro firmu nejdůležitější v získávání informací o zákazníkovi. Podle objednávky firma může zjistit obor činnosti, sezónnost, prioritu ve výběru produktů, naléhavost a další. Firma navíc získává informace jako bydliště nebo typ platby. K objednávce se mohou přibalit různé propagační letáky, dárky atd.

Skladové zásoby – Jelikož firma zná už zvyky svých zákazníků, může firma efektivně řídit své zásoby výrobků a materiálů. V ideálním světě by byla zásoba na skladě rovna nule. Jelikož ale je těžké provozovat tento styl, je zapotřebí mít aspoň minimální pojistné zásoby. Firma rozděluje výrobky na tři skupiny. První je malá skupina, která vydělává převážnou část peněz. Druhá skupina je poměrně významná, která má slušný vliv na tržbu. Třetí skupina je velká, u které je minimální vliv na tržbu. Firma proto se snaží zaměřit na první skupinu a tím podřizuje svůj propagační marketing, a i své zásoby, které u této skupiny jsou větší.

Pohledávky – Pohledávky se evidují podle splatnosti. Na základě toho firma reaguje na poptávku zákazníka.

Závazky – Firma by měla ve svém zájmu hradit své závazky do zajednaného času. Proto je vhodné si dělat evidenci, aby firma měla povědomí o finanční stabilitě dodavatele.

Účetnictví – Evidence hospodářských aktivit, kdy výstupem firmy jsou především výkazy účetní závěrky. Nakonec je utvořena finanční analýza o plnění cílů podniku.[3]

3.3 TECHNOLOGIE PRO TVORBU IS ve webovém prostředí

Na začátku, než se začne tvořit webová aplikace, je třeba se rozhodnout, jak stránky vytvořit. Pro tvorbu stránek existuje několik základních možností:

- On-line webové editory
- Redakční systémy
- Wysiwyg editory
- Strukturní editory

Všechny tyto možnosti mají své výhody a nevýhody. Hlavním faktorem pro rozhodování je znalost (neznalost) jazyků pro tvorbu webových stránek. V praktické části využívám strukturní editor. V následujících kapitolách budou popsány jednotlivé technologie pro tvorbu webové stránky informačního systému.

3.3.1 Apache

Důležitým bodem k úspěšným webovým stránkám je, aby byly přístupné z jakéhokoliv místa, kde je internet. K tomuto slouží obecně webové servery, které nejen uchovávají stránky, ale také i zpracovávají a odesílají data koncovým uživatelům. Tyto servery se nachází na hardwaru, který je napojen na internet. Zajišťuje uživatelům obsluhu v podobě přijímání dotazů v HTTP (ty zpracuje a odešle odpověď). Odpověď bývá často v podobě HTML dokumentu, není to ale podmínka, existuje více formátů (textové nebo multimediální). Mezi nejpoužívanější webový servery je Apache HTTP server [4]. První veřejná verze tohoto serveru byl vydána již v roce 1995 a jedná se o open-source software, který je zdarma. V roce 1996 se provedlo kompletní přepsání kódu a od té doby patří mezi celosvětově populární webové servery. I přesto, že není server nejrychlejší, tak disponuje stabilitou, spolehlivostí a podporou velkého množství platforem. Jeho obliba mezi uživateli s postupujícími roky pomalu klesá s narůstající konkurencí, který sází na rychlost a nízkou náročnost na paměť [5]. Apache spolu s PHP a MySQL tvoří trio, které vytváří dynamický web. Tato spolupráce spočívá v tom, že Apache zpracovává data, která byla vyžádána PHP dotazem z databáze MySQL. Často jsou k dostání spolu, jako balíček na vytváření webových stránek. Jedna z možností je i vzdálený webhosting, ale kvůli delší odezvě se tato možnost moc nepoužívá. K vytvoření vlastního serveru vedou dvě cesty, první je stáhnutí a nainstalování jednotlivých programů a následné nastavování, aby spolu

spolupracovali, nebo druhá cesta použitím některých balíčků, které vše nainstalují a nastavení je pohodlné v grafickém rozhraní. [6]

3.3.2 HTML

První technologie, která je potřeba pro tvorbu stránek je HTML. Jazyk je používám k tvorbě webových stránek v systému World Wide Web. Obsah HTML dokumentů tvoří zdrojový kód, který obsahuje HTML značky, symboly a text, ten se pak přeloží do grafického vizuálu stránky. Zdrojový kód je možný zobrazit v kterémkoliv textovém editoru, či integrovaném editoru v prohlížeči [7] [8].

Zdrojový kód je psán textem, různými symboly a HTML značkami (tagy). Značky se píšou v ostrých závorkách $\langle \rangle$ dělí se na dva druhy, a to párové a nepárové. V případě párových značek je jejich obsah proveden mezi počátkem a koncem tagu. Tyto značky utváří příkazy, které slouží k úpravě webových stránek ať už po vzhledové nebo textové části, či jiné manipulaci s dalšími prvky. Tagy mají také tu vlastnost, že se mohou vzájemně ovlivňovat a vkládat mezi sebe, tím pak se utváří hierarchický kód. Ukázka kódu: [9]

```
<html>
<head>
  <title>Název </title>
</head>

<body>
  Hlavní část kódu
</body>
</html>
```

Tento jazyk prochází stálým vývojem. Byl inspirací pro jiné jazyky. V posledních letech se rozvíjí HTML5, která přináší novinky v podobě nových možnostech pro práci s multimédií a javascriptovu API3. [9]

3.3.3 Java script

Jedná se o kompaktní, objektově orientovaný skriptovací jazyk. Jeho úlohou je zakomponovat drobné programy do webového prostředí. Kromě toho JS částečně vyplňuje mezery, které HTML vytváří. [10]

JS je pro webové prostředí velmi důležitý, neboť dokáže zachytit a zpracovat veškeré události ve webovém prohlížeči. Například se skript může spustit poté, co uživatel klikne na tlačítko nebo než se načte webová stránka. Pomocí JS je také možné testovat korektnost vstupů dříve, než je server obdrží. V případě, že se jedná o chybné vstupy, může JS ušetřit čas a snížit zatížení sítě. [10]

JS rozšiřuje webové stránky o několik podstatných věcí:

1. Mohou se pomocí něho v omezené míře vytvářet HTML dokumenty.
2. Vstupy se mohou testovat přímo v prohlížeči dříve, než dorazí zpět na server.
3. JS provádí interaktivní komunikaci s uživatelem a lze vytvářet vstupy, na které uživatel ihned reaguje.
4. Pokud se odstraní formuláře dokumentu HTML může se v rámci webových stránek vytvářet uživatelské rozhraní. To platí i pro dialogových oken. [10]

Příklad jednoduchého JS:

```
<body>
```

```
Toto je normální text stránky.<br>
```

```
<script>
```

```
document.write("A toto napsal JavaScript");
```

```
</script>
```

```
</body>
```

3.3.4 JQuery

jQuery je rychlá, malá a na funkce bohatá JavaScriptová knihovna. Díky snadno použitelnému rozhraní API, které funguje ve velkém množství prohlížečů, jsou věci, jako je procházení a manipulace s dokumenty HTML, zpracování událostí, animace a Ajax, mnohem jednodušší. [11]

jQuery je zároveň JavaScriptový framework, který umožňuje lehce vyhledávat elementy DOMu, upravovat je, i vytvářet nové. Vyhledávání je možné za pomoci CSS anebo XPath. Vytváření nových HTML elementů je také jednoduché. jQuery sám vytvoří patřičnou strukturu DOMu ze zadaného HTML kódu. Framework jQuery oplývá dalšími možnostmi jako jsou schopnost pracovat s událostmi, nabízení pokročilých funkcí pro práci s poli, či podpora Ajaxu a animací. jQuery svou všestranností a rozšířeností změnil způsob vytváření JavaScriptů. [12]

3.3.5 CSS

CSS bylo vytvořeno koncem roku 1996. V češtině značený jako kaskádové styly, mají za úlohu určovat styl dokumentu. To může znamenat například rozvržení stránky, barvy, velikost písma atd. Kaskádové jsou, protože se mohou na sebe vrstvit, ale vždy platí jen ta poslední. CSS přináší webovým stránkám styly a interagují s prvky HTML. Jako příklad odstavec v HTML vypadá následovně: [13]

```
<p> Odstavec </p>
```

Pokud bychom chtěli, aby se odstavec zobrazoval růžově použil by se kód CSS:

```
p { color:pink; }
```

Část kódu CSS určující, který prvek HTML má být modifikován, se nazývá selektor. V tomto případě selektorem je „p“ a je zapsán vlevo před závorkou. Obsah mezi závorkami se nazývá deklarace, ta určuje vlastnosti a hodnoty, které se provedou na

selektoru. Vlastnosti jsou například barva nebo velikost písma. Hodnoty jsou nastavení či číselná hodnota pro tyto vlastnosti např. růžová nebo 1.

Existují tři způsoby, jak přidat CSS kód do HTML: externí, interní nebo vložený. Stejně jako u HTML je kód CSS editován v textovém editoru. [13]

Externí:

Styly se ukládají zvlášť do jednoho souboru, následně se do záhlaví .html odkazuje na tento soubor šablon. [13]

```
<head>
<link rel = "stylesheet" type = "text / css" href = mojestyly.css ">
</head>
```

Interní:

Pokyny CSS jsou zapsané přímo do záhlaví konkrétních stránek .html. To je užitečné například, když je potřeba, aby jedna stránka vypadala jinak než ty ostatní. [13]

```
<head>
<style>
Body {background-color: blue; }
P {velikost písma: 12px; barva: red; }
</style>
</head>
```

Vložené:

Jsou zapsané přímo v kódu HTML a mají vliv jen na určitý jeden prvek kódu. [13]

```
<h1 style = "font-size: 40px; color: violet;"> Podívejte se na tento nadpis!
</h1>
```

V dnešní době se používá CSS 2.1 a ve vývoji je již CSS 3. Novější verze nese s sebou zároveň problém s kompatibilitou se staršími verzemi prohlížečů. [13]

3.3.6 PHP

Jedná se o dynamický programovací skriptovací jazyk, který je vhodný pro vývoj webových aplikací. [14]

Používá se na straně serveru a slouží ke generování HTML kódu stránky, který pak server odešle do prohlížeče. Výhodou je například malá náročnost na hardware koncového uživatele a slušná bezpečnost kódu. Mezi dalšími významnými výhodami jsou jeho široká podpora a rozšířenost. [15]

PHP dokáže běžet na jakékoliv platformě a pracuje s různými databázemi. Jeho volná syntaxe pomáhá snadnému učení. Spolu s MySQL a Apache tvoří tzv. triádu. [16]

Poslední vydaná verze PHP je verze 5. Oproti předchozí verzi došlo k vylepšení jádra integrací nových prvků. PHP dnes používá statisíce vývojářů a miliony serverů obsahuje PHP, což je kolem 20 % domén na internetu. [17]

3.3.6.1 Eclipse IDE for PHP Developers

Pro implementaci IS bylo zvoleno prostředí PHP IDE poskytuje rámec integrovaného vývojového prostředí PHP pro platformu Eclipse. Je to zdarma open-source prostředí, které jsem použil při vývoji IS. Tento projekt zahrnuje vývojové komponenty nezbytné pro vývoj webových aplikací založených na PHP a usnadňuje rozšiřitelnost. Využívá existující Web Tools Project při poskytování funkcí PHP vývojářům. [18]

3.3.7 MySQL

Posledním stavebním kamenem v chodu informačního systému je databáze. Pro mou práci jsem zvolil MySQL, což je open source relační databáze typu DBMS a vychází z deklarativního programovacího jazyka SQL. Díky své rychlosti, dostupnosti a jednoduchosti je v dnešní době jeden z nejoblíbenějších systémů.

Do MySQL lze uložit různá data od textů až po obrázky, s nimiž se dá dále jednoduše pracovat. Struktura databáze je jednoduchá v podobě tabulek, které mají sloupce a řádky. Každý řádek obsahuje předem určené typy záznamů. [19]

Software má plno výhod například, snadná implementace, vysoký výkon a dá se nainstalovat na různý operační systémy. Často se používá na Linuxu. Možné je i využít open source nástroje PhpMyAdmin, který slouží ke kompletní správě systému přes webové prostředí. [20]

Co se týče řádků, tak ty zastupují jednotlivé záznamy a ve sloupcích jsou vyplněny jednotlivé hodnoty. Způsob, jakým je tvořen zápis, je ideální pro různé seznamy, ceníky a data která se zapisují pomocí tabulek. Další výhodou MySQL je, že běží jako samostatný server a k obsluze dat slouží většinou skriptování na straně serveru (v našem případě pomocí PHP). V MySQL se používají příkazy zakončené středníkem. [21]

3.3.7.1 HeidiSQL

Databáze byla spravována v HeidiSQL. Jedná se o zdarma open source software a jeho cílem je, aby byl snadný se naučit. Umožňuje prohlížet a upravovat data a struktury z počítačů s jedním z databázových systémů MariaDB, MySQL, Microsoft SQL, PostgreSQL a SQLite. HeidiSQL. [22]

Aby uživatel mohl spravovat databáze s HeidiSQL musí se přihlásit do lokálního nebo vzdáleného serveru MySQL. Software disponuje dostatečným počtem funkcí pro běžné a pokročilé databáze (tabulky a datové záznamy atd.). [23]

3.3.8 UML

Jedná se o soubor grafických notací při vývoji softwaru. V dnešní době se využívá při analýze potřeb a navrhování různých systému. Tudiž je pro návrh a vývoje informačního systému užitečným nástrojem. [24]

Dnešní systémy jsou natolik komplexní a rozvinuté, že jen nestačí rovnou psát kód. Systém se musí nejprve navrhnout a k tomu slouží i právě UML. Tento nástroj může vyřešit spoustu problémů, jako je komunikace v rámci vývojového týmu, změny v klientovi a samozřejmě v rané fázi, kdy probíhá analýza. Využitelný je i ve fázi designu, kdy se rozhoduje, jak systém naprogramovat.

Existuje několik druhů UML diagramů:

Jednou z forem je například náčrt na papíru nebo tabuli.

Dalším je UML jako plán, což jsou diagramy vytvářené v CAD nástrojích. Ty usnadňují programátorům lehčí implementaci, komunikaci v rámci týmu a orientaci v systému. Nakonec slouží diagramy jako dokumentace systému.

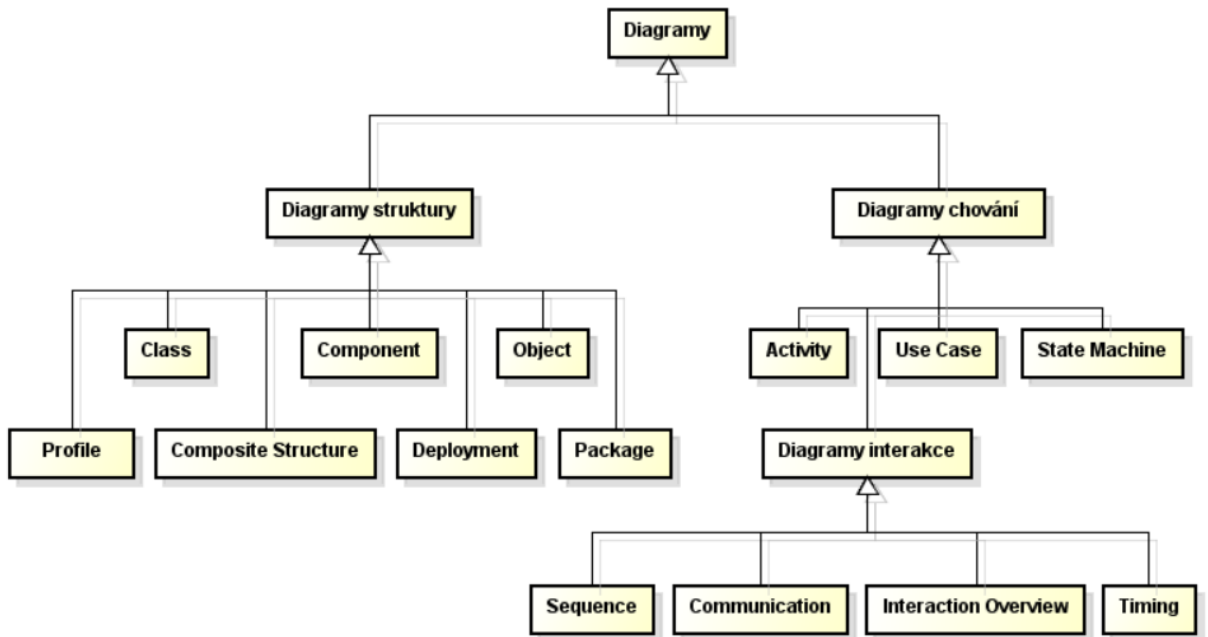
Posledním je UML jako programovací jazyk. Z detailního UML diagramu se může vygenerovat šablona kódů, pomocí který se dá začít implementovat systém. V databázi se tak využívá pro vygenerování základních skriptů. [24]

Diagramy lze ještě rozdělit na dvě skupiny:

Diagramy struktury – Popisují strukturu systému (Z čeho je systém složený).

Diagramy chování – Popisují chování systému (Jak funguje systém). [24]

Obrázek 4: Ukázka UML diagramu [24]



4 Vlastní práce

Vlastní práce spočívá v analýze potřeb firmy a následného navržení informačního systému. Poté se vytvoří databáze se samotným informačním systémem. Obsluhování IS dělá uživatel ve webovém prostředí internetového prohlížeče.

4.1 Požadavky na IS

Kromě standardních požadavků na IS jako je evidence pohledávek, závazků, skladového hospodářství a objednávkové služby by měl IS pro potravinářský podnik umožňovat řešit tyto oblasti:

- Dosledovatelnost potravin
- Jakostní údaje mouky a obilí.
- Normy spotřeb (receptury)
- Složení výrobku a nutriční údaje

4.1.1 Dosledovatelnost

Sledovatelnost potravin je podle nařízení o bezpečnosti potravin č. 178/2002/ES (čl. 18) možnost najít a vysledovat výroby, zpracování a distribuce potravinářský produkt atd. Aby byla sledovatelnost zajištěna, jsou zaváděny systémy a postupy, které umožní dodat potřebné informace příslušným orgánům.

V případě závady u produktu je pomocí těchto informací možnost dohledat příčinu. Součástí systému sledovatelnosti je podniková dokumentace o příjmu surovin a jejich využití, o distribuci produktů a označování výrobní šarže. Tento proces běží pomocí programů, který chod velmi usnadňují.

4.1.2 Jakostní údaje mouky a obilí

Z hlediska jakosti je ve mlýně sledována jakost u dodávaného obilí, jakost vyráběných druhů mouk a jakost prodávaných druhů mouk.

Pro jednotlivé položky obilí a mouky jsou předepsány standardní jakostní parametry. S těmito parametry jsou pak srovnávány dosahované jakostní parametry v nákupu a ve výrobě.

Tabulka 1: Příklad jakostních údajů

Mouka jakostní údaje	Jakostní údaje pšenice
Vlhkost (%)	Vlhkost (%)
Popel (%)	Příměsi (%)
Lepek (%)	Nečistoty (%)
Granulace	Barevné změny (%)
Číslo poklesu (s)	Obj. hmotnost (g/litr)
Očkovitost	Pokles (s)
Gluten index (%)	Lepek (%)
Maltóza (%)	Dusík (%)
N-látky (%)	Zelený (ml)
Tuk (%)	SDS test (ml)
	Alveo.energie (cm ²)
	Poměrové číslo

Zdroj: Vlastní zpracování, 2021

4.1.3 Normy spotřeb (receptury)

Struktura výrobku (normu spotřeby, receptura) se používá při zápisu příjmu vyrobeného množství finálního výrobku a zpětného výdeje do výroby spotřebovávaných materiálů.

Norma spotřeby vyráběného produktu může být vytvořená ve více úrovních. Na vrcholu normy je produkt, který je zde nazýván „rodičem“ a produkty, které se na jeho výrobu spotřebovávají, jsou nazývány „dětmi“.

4.1.4 Složení výrobku a nutriční údaje

Údaje uváděné na obalech potravin jsou pro zákazníka základním zdrojem informací, aby se mohl rozhodnout při výběru, který výrobek si koupí. Co má být na výrobku uvedeno, říkají české zákony, vyhlášky a také nařízení EU. Za informace odpovídá ta firma, pod kterou je potravina uvedená na trh

Pravidla pro informace uváděné na potravinách:

1. Nesmějí zákazníka mást v charakteristikách potraviny. Nesmí uvádět účinky nebo vlastnosti, které potravina nemá. Nesmí se vypisovat přítomnosti nebo nepřítomnosti určitých složek, látek nebo živin, pokud jsou běžně obsažené v jiných podobných potravin.
2. Musí být čitelně, srozumitelně a přesně napsané

3. Nesmí obsahovat, že potravina léčí, či mírní lidské nemoci. Výjimkou jsou minerální vody a potraviny určené pro zvláštní výživu, které mají jiné předpisy.
4. Údaje musí být uvedeny ve srozumitelném jazyku dané země.

4.2 Řešení Dosledovatelnosti potravinu

Řešení problematiky „dosledovatelnosti“ potravin se využívá šaržová evidence výrobku a surovin k výrobě potravinářského produktu.

Pro řešení šaržové evidence v IS jsou stanoveny tyto hlavní kritéria:

1. Pro zajištění dosledovatelnosti využívat primárně údaj „datum minimální trvanlivosti“, který je povinně uváděn na obalech produktů.
2. Jednoduchý proces „expedice“ zahrnující příjem objednávek.
3. Minimalizovat pracnost při práci s IS.

Způsob řízení evidence šarží výrobků je určen nastavením parametrů na skladové kartě produktu.

1. Není šaržová evidence
2. Dle ID šarže
3. FIFO dle pořadí příjmu
4. FIFO dle DMT a pořadí příjmu
5. Dle data expedice a ID šarže

4.2.1 Identifikace šarže produktu

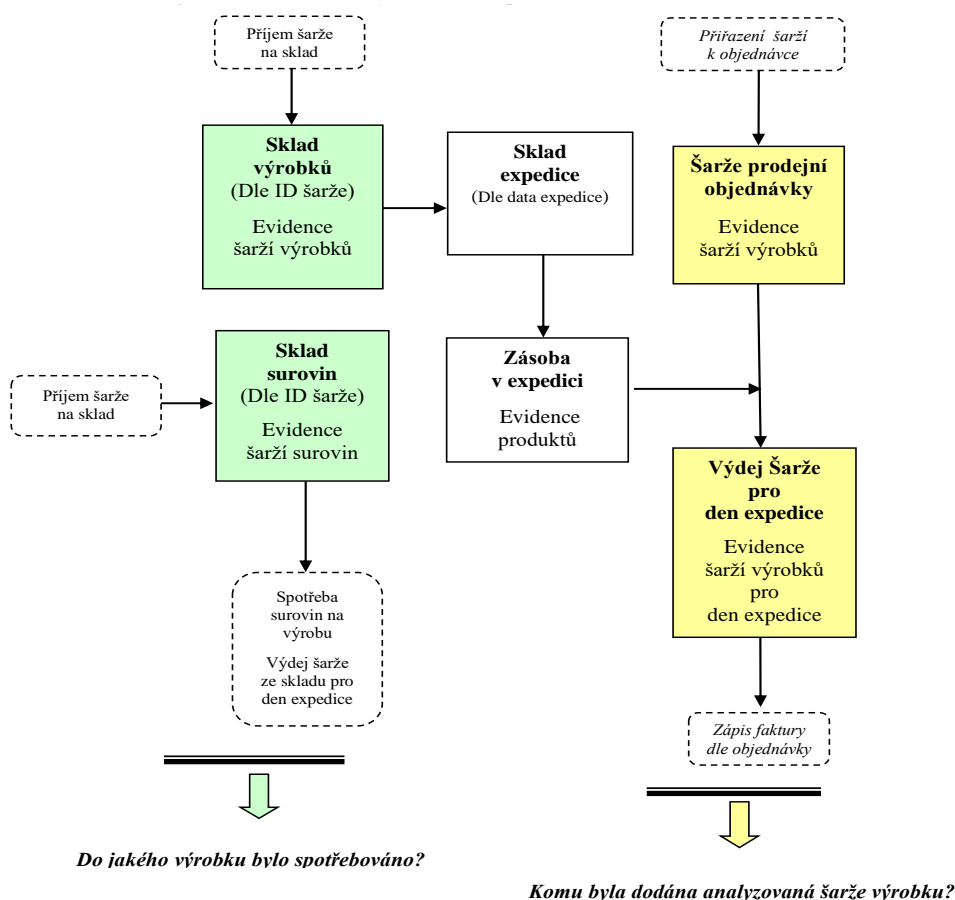
Šarže produktu je primárně identifikována údajem DMT. Evidenci šarží produktů je možno realizovat s vazbou na „umístění ve skladu“ (využití například pro volné mouky uskladněné v jednotlivých silech skladu mouk).

Zadávání identifikace šarže produktů při výdejích ze skladu uživatelem je závislé na metodě řízení šarží nastavené na skladové kartě produktu.

Šarže produktu je v IS identifikovaná těmito údaji:

- kód produktu
- výrobce produktu
- datum trvanlivosti
- šarže výrobce

Obrázek 5: Schéma fungování šarží



Zdroj: Vlastní zpracování, 2021

Tabulka 2 - Popis údajů skladové karty

Údaj	Význam údaje
Indikátor sledování šarží A/N	A = Šarže jsou sledovány N = Šarže nejsou sledovány
Metoda řízení evidence šarží	1 = Dle ID šarže 2 = FIFO dle pořadí příjmu 3 = FIFO dle DMT a pořadí příjmu
Evidence šarží dle umístění A/N	A = Šarže evidovány na umístění ve skladu N = Šarže evidovány na skladovou kartu

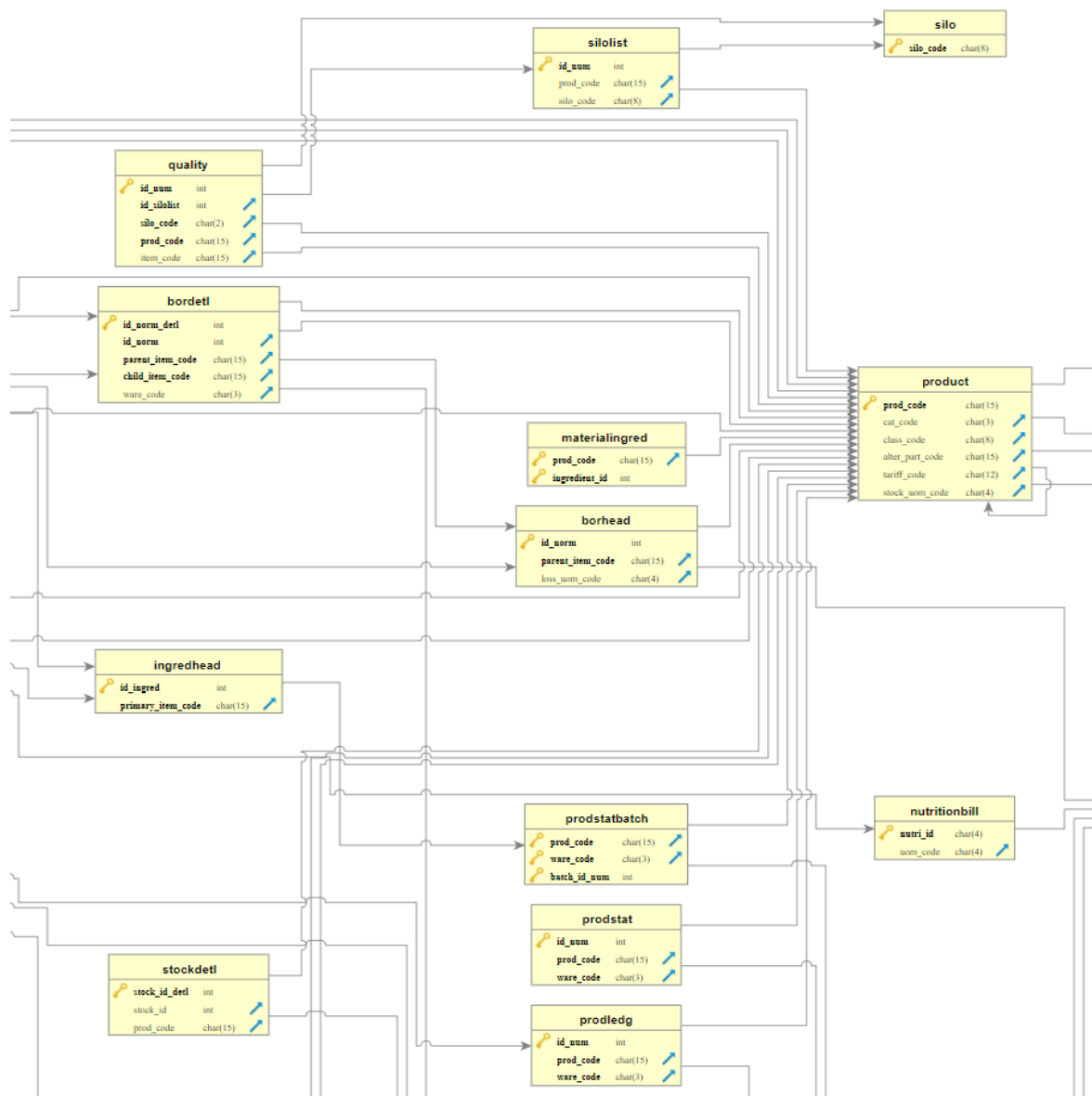
Zdroj: Vlastní zpracování, 2021**Tabulka 3 - Popis šaržových parametrů skladu**

Údaj	Význam údaje
Povolení sledování šarží ve skladu A/N	A = Šarže jsou ve skladu povoleny, tj. na skladovou kartu produktu lze nastavit parametry řízení šarží. N = Šarže jsou ve skladu zakázány, tj. na skladovou kartu produktu nebude možné nastavit parametry řízení šarží.
Identifikace expedičního skladu A/N	A = Sklad je expedičním skladem (možno použít pouze metodu řízení šarží ve skladu typu 4) N = Sklad není expedičním skladem (možno použít pouze metodu řízení šarží ve skladu typu 1,2 a 3)
Implicitní nastavení metody řízení evidence šarží	1 = Dle ID šarže 2 = FIFO dle pořadí příjmu 3 = FIFO dle DMT a pořadí příjmu
Implicitní nastavení metody umístění ve skladu	A = Šarže evidovány na umístění ve skladu N = Šarže evidovány na skladovou kartu

Zdroj: Vlastní zpracování, 2021

4.3 UML diagramy

Obrázek 6 - Referenční schéma vazeb mezi tabulkami

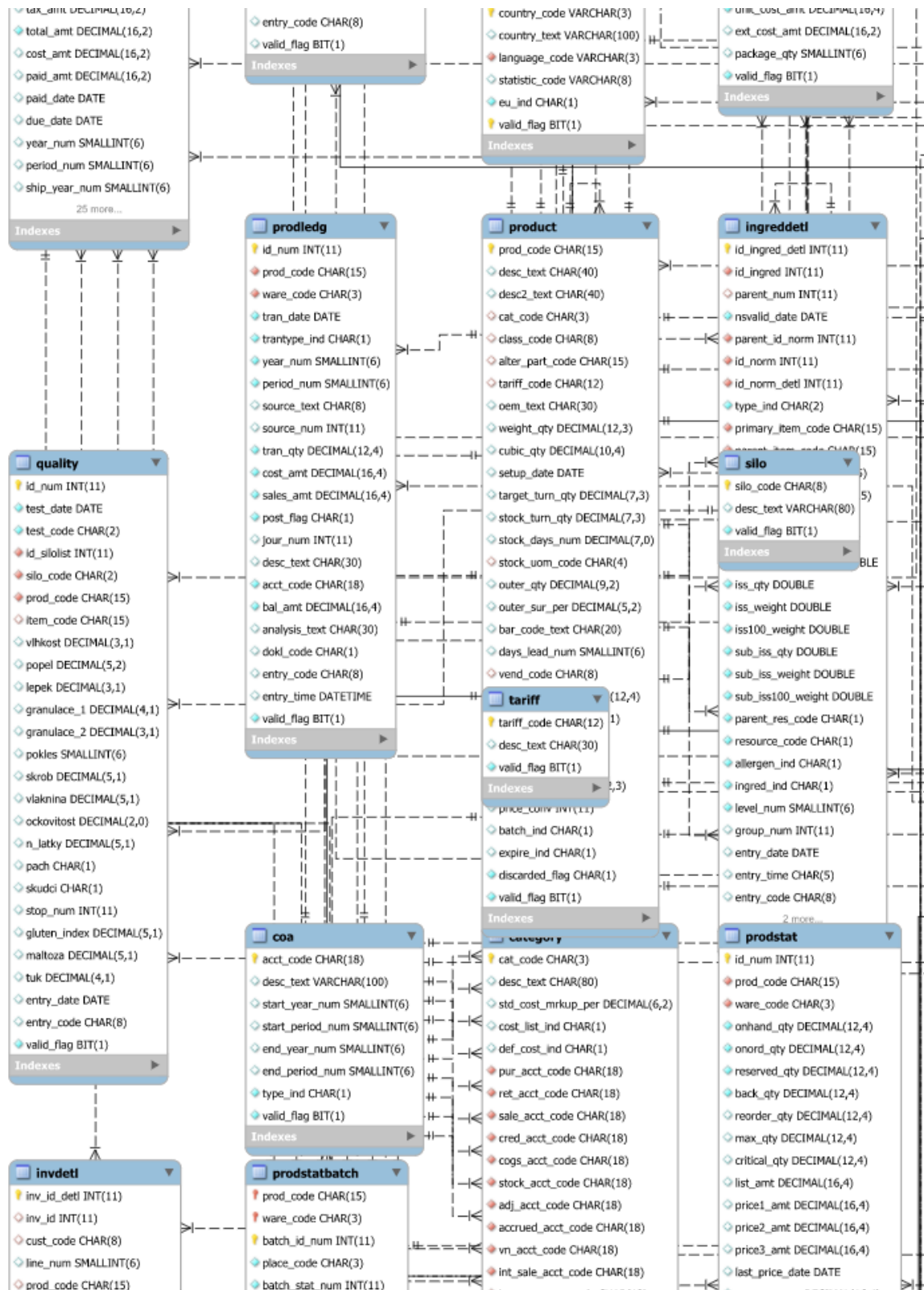


Zdroj: Vlastní zpracování, 2021

Odkaz na obrázek v plné velikosti:

http://www.perut.www3.cz/obrazky/schema_DB_perut.svg

Obrázek 7: EER diagram



Zdroj: Vlastní zpracování, 2021

Odkaz na obrázek v plné velikosti:

http://www.perut.www3.cz/obrazky/UML_IS_diagram.svg

4.4 Popis databáze

Databáze byla navržena v programu MySQL Workbench 8.0 firmou Oracle. Kvůli velkému počtu tabulek, je zde uvedeno jen několik důležitých.

Tabulka 4: Vybrané tabulky z databáze

Tabulka DB	Název sloupce	DB typ	Délka	Popis
borhead Hlavička normy spotřeby	id_norm	INT	10	Položka-RODIČ (vrcholová položka NS)
	parent_item_code	CHAR	15	Pořadové číslo normy
	type_ind	CHAR	2	Typ normy spotřeby
	price_conv	DECIMAL	12	Množství měrné jednotky, pro které je norma spotřeby vytvořena
	cost_rollup_amt	DECIMAL	16	Kalkulované náklady
	loss_qty	DECIMAL	12	Normovaná ztráta
	loss_uom_code	CHAR	4	MJ normované ztráty. Jsou povoleny pouze hodnoty % nebo MJ produktu.
	loss_ind	CHAR	1	Indikátor ztráty. Hodnoty: 0=ztráty nejsou zahrnuty v NS 1=ztráty jsou zahrnuty v NS
	start_date	DATE	10	Začátek platnosti
	end_date	DATE	10	Konec platnosti
	entry_date	DATE	10	Datum zápisu
	entry_code	CHAR	8	Uživatel, který provedl zápis
	valid_flag	BIT	1	Platnost záznamu
bordetl Řádky normy spotřeby	id_norm_detl	INT	10	ID řádky NS
	id_norm	INT	10	ID hlavičky NS
	parent_item_code	CHAR	15	Vrcholová položka normy (RODIČ)
	child_item_code	CHAR	15	Kód položky normy (DÍTĚ) Pro druh řádky I-instrukce je ve sloupci uveden implicitně kód "INSTRUKCE"
	type_ind	CHAR	2	Typ normy spotřeby
	sort_num	INT	10	Třídící číslo řádky normy v rámci RODIČE
	required_qty	DOUBLE	22	Množství položky-dítě požadované pro položku-rodíč
	resource_code	CHAR	1	Typ vztahu RODIČ-DÍTĚ Hodnoty C=komponenta F=fantom G=generická položka I=instrukce P=Podskupina S=náklady
	cost_amt	DECIMAL	16	Náklady na položku normy
	entry_date	DATE	10	Datum zápisu
	entry_code	CHAR	8	Kód uživatele
	work_place_code	CHAR	8	Kód pracoviště
	ware_code	CHAR	3	Implicitní sklad výdeje

Tabulka DB	Název sloupce	DB typ	Délka	Popis
	back_flush	CHAR	1	Zpětný výdej Zpětný výdej Hodnoty A = Provádí se automatický výdej ze skladu N = Výdej se automatický neprovádí
	start_date	DATE	10	Začátek platnosti vazby RODIČ-DÍTĚ
	end_date	DATE	10	Konec platnosti vazby RODIČ-DÍTĚ
	valid_flag	INT	10	Platnost záznamu
Customer Odběratel-základní údaje	cust_code	VARCHAR	8	Kód odběratele
	name_text	VARCHAR	100	Název firmy odběratele
	addr1_text	VARCHAR	100	1. řádka adresy odběratele
	addr2_text	VARCHAR	60	2. řádka adresy (ulice)
	city_text	VARCHAR	40	Město
	state_code	CHAR	6	Kód regionu (země, kraj, okres)
	post_code	CHAR	10	PSČ
	country_text	VARCHAR	100	Název státu
	country_code	CHAR	3	Kód státu
	type_code	CHAR	3	Typ odběratele (vazba na účetnictví)
	term_num	INT	10	Implicitní počet dní splatnosti
	tax_code	CHAR	3	Implicitní kód daně
	tax_num_text	VARCHAR	20	DIČ odběratele
	contact_text	VARCHAR	100	Kontaktní osoba u odběratele
	tele_text	VARCHAR	20	Telefon
	email_text	VARCHAR	100	Email text
	bal_amt	DECIMAL	16	Limit úvěru
	hold_flag	CHAR	1	Příznak zadrženého prodeje (A/N)
	currency_code	CHAR	3	Měna odběratele
	bank_acct_code	VARCHAR	60	Bankovní účet odběratele v text. tvaru
ico_num	CHAR	12	IČO – Identifikační číslo odběratele	
bic_code	CHAR	12	BIC/SWIFT	
valid_flag	BIT	1	Platnost záznamu	
custship Odběrní místa	cust_code	CHAR	8	Kód odběratele
	ship_code	CHAR	8	Kód odběrního místa
	name_text	CHAR	80	Název odběrního místa
	addr_text	CHAR	80	1. řádka adresy
	addr2_text	CHAR	80	2. řádka adresy (ulice)
	city_text	CHAR	80	Město
	state_text	CHAR	6	Kód regionu (okresu)
	post_code	CHAR	10	PSČ
	country_text	CHAR	80	Název státu
	contact_text	CHAR	80	Kontaktní osoba
	tele_text	CHAR	20	Tel. číslo odběrního místa
	email_text	CHAR	80	E-mail
	ware_code	CHAR	3	Impl.kód skladu prodeje
	cat_code	CHAR	3	Kategorie OM
run_text	CHAR	3		
valid_flag	BIT	1	TRUE – Platný záznam FALSE – Zrušený záznam	
ingredhead Hlavička složení	id_ingred	INT	10	Číslo výpočtu složení výrobku
	nsvalid_date	DATE	10	Datum platnosti NS, pro který se provedl

Tabulka DB	Název sloupce	DB typ	Délka	Popis
výrobku	type_ind	CHAR	2	Typ normy spotřeby
	primary_item_code	CHAR	15	Kód karty výrobku
	tot_iss_weight	DOUBLE	22	Hmotnost v gramech dle NS (MJ výr.) NEUPEČENÝ VÝROBEK. Celkové potřebné množství suroviny v gramech na jeden výrobek nebo celkové potřebné množství suroviny v Kg na 1000 MJ výrobků
	tot_iss100_weight	DOUBLE	22	Hmotnost v gramech dle NS (100 g výr.) NEUPEČENÝ VÝROBEK (100 g) Celkové požadované množství surovin v gramech na 100 g výrobku
	vypek_g	DOUBLE	22	Výpek v gramech
	vypek_per	DOUBLE	22	Výpek %
	vypek_tot	DOUBLE	22	Celkem výpek
	reduct_weight	DOUBLE	22	Po zpracování
	entry_date	DATE	10	Datum zápisu
	entry_time	CHAR	5	Čas zápisu
	entry_code	CHAR	8	Uživatel
	ingred_text	VARCHAR	3000	Text Složení
	allergen_text	VARCHAR	1000	Text Alergeny
	valid_record	BIT	1	Platnost záznamu
ingreddetl Složení výrobku řádky	id_ingred_detl	INT	10	Číslo řádky výpočtu
	id_ingred	INT	10	Číslo výpočtu složení výrobku
	parent_num	INT	10	Primární sekvenci číslo
	nsvalid_date	DATE	10	Datum platnosti výpočtu
	parent_id_norm	INT	10	Číslo nadřazené NS
	id_norm	INT	10	Číslo NS polotovaru
	id_norm_detl	INT	10	Číslo řádky NS
	type_ind	CHAR	2	Typ normy spotřeby
	primary_item_code	CHAR	15	Kód produktu, pro který bylo vypočteno složení surovin.
	parent_item_code	CHAR	15	Kód výrobku/polotovaru
	child_item_code	CHAR	15	Kód položky NS
	subst_prod_code	CHAR	15	Kód záměnné položky
	required_qty	DOUBLE	22	Požadované množství výrobku v MJ na 1000 MJ primárního výrobku.
	required_weight	DOUBLE	22	Požadované množství výrobku v KG na 1000 MJ primárního výrobku.
	required100_weight	DOUBLE	22	Požadované množství výrobku v MJ na 100 Kg primárního výrobku.
	iss_qty	DOUBLE	22	Požadované množství suroviny na 1000 MJ primárního výrobku.
	iss_weight	DOUBLE	22	Požadované množství suroviny v KG 1000 MJ primárního výrobku.
	iss100_weight	DOUBLE	22	Požadované množství suroviny na 100 KG primárního výrobku.
	sub_iss_qty	DOUBLE	22	Požadované množství záměnného produktu v MJ na 1000 MJ primárního výrobku.
	sub_iss_weight	DOUBLE	22	Požadované množství záměnného produktu v KG na 1000 MJ primárního

Tabulka DB	Název sloupce	DB typ	Délka	Popis
				výrobku.
	sub_iss100_weight	DOUBLE	22	Požadované množství zámenného produktu v KG na 100 KG primárního výrobku.
	parent_res_code	CHAR	1	Druh řádky nadřazené NS Hodnoty 0 = Vygenerovaný záznam pro primární produkt C = Komponenta D = Podskupina F = Fantom G = Zástupná položka
	resource_code	CHAR	1	Druh běžné řádky NS Hodnoty C = Komponenta F = Fantom G = Zástupná položka P = Podskupina
	allergen_ind	CHAR	1	Míra existence alergenů ve složce
	ingred_ind	CHAR	1	Typ záznamu: Hodnoty L = Složky primární suroviny P = Primární surovina S = Surovina V = Výrobek X = Ostatní Y = Ignorovaná suroviny
	level_num	SMALLINT	5	Číslo úrovně NS
	group_num	INT	10	Číslo skupiny pro zástupnou položku
	entry_date	DATE	10	Datum zápisu
	entry_time	CHAR	5	Čas zápisu
	entry_code	CHAR	8	Kód uživatele
	group_prim_ingred	INT	10	Seskupení, do které patří složka primární suroviny. Údaj je určen k seskupení údajů pro tisk složení výrobku. Nesouvisí se skupinami složek.
	valid_flag	BIT	1	Platnost záznamu
materialingred	prod_code	CHAR	15	Produkt
Složky v surovině	ingredient_id	INT	10	ID složky suroviny
	desc_text	VARCHAR	100	Název složky
	ingredient_order	SMALLINT	5	Pořadí složky v surovině
	ingredient_per	DECIMAL	5	Procento složky obsažené v surovině
	allergen_text	VARCHAR	200	Alergeny složky
	group_text	VARCHAR	50	Skupina složek např. Tuky
	norm_text	VARCHAR	200	Normovaný popis složky
	valid_flag	BIT	1	Platnost záznamu
menu	menu_id	INT	10	ID menu
Menu IS	parent_menu_id	INT	10	ID menu rodiče
	base_menu_text	CHAR	80	Název hlavního menu
	desc_text	CHAR	80	Popis menu

Tabulka DB	Název sloupce	DB typ	Délka	Popis
	security_ind	CHAR	1	Úroveň oprávnění
	sort_key	SMALLINT	5	Třídící klíč
	file	CHAR	80	Název a cesta k souboru s formulářem
	sub_menu	CHAR	1	Podmenu Y/N
	param	CHAR	80	Parametry menu
	valid_flag	BIT	1	Platnost záznamu
nutritionbill Položky nutričních údajů	nutri_id	CHAR	4	Kód nutričního údaje
	desc_text	VARCHAR	80	Název nutričního údaje
	uom_code	CHAR	4	MJ nutričního údaje
	valid_flag	BIT	1	Platný záznam
nutritionvalue Hodnoty nutričních údajů produktů	prod_code	CHAR	15	Kód produktu
	id_ingred	INT	10	Číslo výpočtu složení výrobku
	nutri_id	CHAR	4	Kód nutričního údaje
	nutri_value	DECIMAL	7	Hodnota nutričního údaje
	valid_flag	BIT	1	Platnost záznamu
prodbatch Karta šarže produktu	batch_id_num	INT	10	Identifikační číslo šarže
	prod_code	CHAR	15	Kód produktu
	supplier_code	CHAR	8	Dodavatel dodávky
	production_date	DATE	10	Datum výroby
	expire_date	DATE	10	Datum trvanlivosti
	vend_batch_code	CHAR	20	Šarže výrobce
	supplier_type	CHAR	1	Typ dodavatele: Hodnoty: E = externí V = vlastní
	mnf_batch_ind	CHAR	1	Typ výrobní šarže: Hodnoty: D=datum výroby ve tvaru RRnnn (nnn=pořadové číslo dne v roce) T = textová informace
	bar_code	CHAR	15	Čárový kód šarže
	valid_flag	BIT	1	Platnost záznamu
prodstat Skladová karta	id_num	INT	10	Unikátní klíč v rámci tabulky
	prod_code	CHAR	15	Kód produktu
	ware_code	CHAR	3	Kód skladu
	onhand_qty	DECIMAL	12	Aktuální stav zásoby na skladě
	onord_qty	DECIMAL	12	Nedodané množství z nákupních objednávek
	reserved_qty	DECIMAL	12	Rezervováno pro prodej
	back_qty	DECIMAL	12	Nekryté množství produktu
	reorder_qty	DECIMAL	12	Obvykle objednávané množství
	max_qty	DECIMAL	12	Maximální povolený stav zásoby na skladě
	critical_qty	DECIMAL	12	Kritické množství na skladě
	list_amt	DECIMAL	16	Základní ceníková cena
	price1_amt	DECIMAL	16	Cena v 1.cenové hladině
	price2_amt	DECIMAL	16	Cena v 2.cenové hladině
	price3_amt	DECIMAL	16	Cena v 3.cenové hladině
	last_price_date	DATE	10	Datum poslední změny ceníku
est_cost_amt	DECIMAL	16	Obvyklá skladová standardní cena	

Tabulka DB	Název sloupce	DB typ	Délka	Popis
	act_cost_amt	DECIMAL	16	Cena posledního příjmu v základní měně
	wgted_cost_amt	DECIMAL	16	Aktuální skladová cena
	bin1_text	CHAR	15	Umístění 1
	bin2_text	CHAR	15	Umístění 2
	bin3_text	CHAR	15	Umístění 3
	last_sale_date	DATE	10	Datum posledního výdeje
	last_receipt_date	DATE	10	Datum posledního příjmu
	stocked_flag	CHAR	1	Označení skladovaného produktu (Y/N)
	abc_ind	CHAR	1	Označení dle metody ABC
	reserve_flag	CHAR	1	Indikátor, zda je produkt rezervován
	batch_ind	CHAR	1	indikátor sledování šarží
	batch_method	CHAR	1	Metoda řízení evidence šarží
	place_flag	CHAR	1	Evidence šarží dle umístění ve skladu A/N
	valid_flag	BIT	1	
product Produkt (Výrobek, zboží, materiál atd.)	prod_code	CHAR	15	Kód produktu
	desc_text	CHAR	40	Popis produktu
	desc2_text	CHAR	40	Druhá řádka pro popis produktu
	cat_code	CHAR	3	Kategorie produktu
	class_code	CHAR	8	Skladovací třída
	alter_part_code	CHAR	15	Alternativní produkt
	tariff_code	CHAR	12	Kód celního sazebníku
	oem_text	CHAR	30	Kód produktu u dodavatele
	weight_qty	DECIMAL	12	Čistá váha produktu v kg (netto)
	cubic_qty	DECIMAL	10	Objem jednotky v metrech krychlových
	setup_date	DATE	10	Datum zapsání záznamu o produktu
	target_turn_qty	DECIMAL	7	Cílový obrat zásob
	stock_turn_qty	DECIMAL	7	Aktuální obrat zásob
	stock_days_num	DECIMAL	7	Počet dní uložení produktu ve skladu
	stock_uom_code	CHAR	4	MJ skladu
	outer_qty	DECIMAL	9	Množství v obalu (např. v kartonu apod.)
	outer_sur_per	DECIMAL	5	Přirážka za balení v procentech
	bar_code_text	CHAR	20	Označení produktu v čárovém kódu – EAN
	days_lead_num	SMALLINT	5	Průměrná dodací lhůta pro nákup. produkt
	vend_code	CHAR	8	Obvyklý dodavatel
	min_ord_qty	DECIMAL	12	Min. objednatelné množství nákup
	days_warr_num	INT	10	Záruční lhůta udána ve dnech
	size2	DECIMAL	7	Rozměr 2 v mm
	size3	DECIMAL	7	Rozměr 3 v mm
	bruto_qty	DECIMAL	12	Hrubá váha produktu v kg (brutto)
	price_conv	INT	10	Počet MJ za který je stanovena prod. cena
	batch_ind	CHAR	1	Příznak sledování šarží A/N
	expire_ind	CHAR	1	Povinnost data trvanlivosti A/N
	discarded_flag	CHAR	1	Vyřazený produkt A/N
	valid_flag	BIT	1	
quality Jakost obilí nebo mouk	id_num	INT	10	Unikátní číslo záznamu
	test_date	DATE	10	Datum měření
	test_code	CHAR	2	Kód testu Hodnoty

Tabulka DB	Název sloupce	DB typ	Délka	Popis
				O1--Obilí – po prvním nakropení O2--Obilí – po druhém nakropení JB--Jedlé balené JP--Jedlé pytlované JV--Jedlé volné KV--Krmné volné OS--Obilí – výstup ze síla PM--Obilí – produkty mletí
	id_silolist	INT	10	ID přiřazení síla k produktu
	silo_code	CHAR	2	Kód zásobníku
	prod_code	CHAR	15	Kód obilí
	item_code	CHAR	15	Kód produktu mletí z obilí. Pro O1,O2 a OS nevyplněno (NULL)
	vlhkost	DECIMAL	3	Procento vlhkosti
	popel	DECIMAL	5	Procento popele
	lepek	DECIMAL	3	Procento lepku
	granulace_1	DECIMAL	4	Číslo granulace 1
	granulace_2	DECIMAL	3	Číslo granulace 2
	pokles	SMALLINT	5	Číslo poklesu v sekundách
	skrob	DECIMAL	5	Škrob g/Kg
	vlaknina	DECIMAL	5	Vláknina g/Kg
	ockovitost	DECIMAL	2	Očkovitost
	n_latky	DECIMAL	5	N látky g/Kg
	pach	CHAR	1	Prach
	skudci	CHAR	1	Škúdcí
	stop_num	INT	10	Číslo stop zkoušky
	gluten_index	DECIMAL	5	Procento gluten indexu
	maltoza	DECIMAL	5	Procento maltózy
	tuk	DECIMAL	4	Tuk g/Kg
	entry_date	DATE	10	Datum zápisu
	entry_code	CHAR	8	Kód autora zápisu
	valid_flag	BIT	1	Platnost záznamu
stockhead Skladový doklad	stock_id	INT	10	ID záznamu
	type_stock	CHAR	1	Typ Doklad: P-Příjemka, V-Výdejka
	ref_text	CHAR	15	Prvotní doklad
	tran_date	DATE	10	Datum dokladu
	year_num	SMALLINT	5	Účetní rok
	period_num	SMALLINT	5	Účetní období
	ware_code	CHAR	3	Kód skladu, kam se přijímá
	override_code	CHAR	18	Maska účtu příjmu – středisko
	acct_code	CHAR	18	Účet příjmu
	vend_code	CHAR	8	Kód odběratele pro vratky
	entry_code	CHAR	8	Zapsal
	entry_date	DATE	10	Datum zápisu příjemky
	print_flag	CHAR	1	Příznak, zda byla příjemka vytištěna
	com1_text	CHAR	30	Poznámka 1
	com2_text	CHAR	30	Poznámka 2
	ship_date	DATE	10	Datum zdanitelného plnění
	place_code	CHAR	3	Kód umístění
	valid_flag	BIT	1	

Tabulka DB	Název sloupce	DB typ	Délka	Popis
stockdetl Řádka skladového dokladu	stock_id_detl	INT	10	ID řádky
	stock_id	INT	10	ID hlavičky
	receipt_num	INT	10	Číslo příjemky
	line_num	SMALLINT	5	Číslo řádky příjemky
	prod_code	CHAR	15	Kód produktu
	desc_text	CHAR	40	Popis produktu
	tran_qty	DECIMAL	12	Množství na řádce v měrné jednotce sklad
	cost_amt	DECIMAL	16	Jednotková cena
	tot_amt	DECIMAL	16	Celkem cena na řádce
	sales_amt	DECIMAL	16	Jednotková prodejní cena
	ship_date	DATE	10	Datum zdanitelného plnění
	valid_flag	BIT	1	
user Uživatelé systému	user_code	CHAR	8	Přihlašovací jméno uživatele – login
	name_text	CHAR	80	plné jméno
	security_ind	CHAR	1	úroveň zabezpečení / přístupová úroveň
	password_text	CHAR	80	heslo pro vstup do systému
	sign_on_date	DATE	10	Datum nastavení
	phone_number	CHAR	20	Telefonní číslo uživatele
	email	CHAR	40	Email
	mobil_number	CHAR	20	mobilní telefon
	fax_number	CHAR	20	Fax
	admin_db	CHAR	1	Administrátor databáze (A/N)
	admin_is	CHAR	1	Administrátor informačního systému (A/N)
	valid_flag	BIT	1	

Zdroj: Vlastní zpracování, 2021

4.5 Softwarová realizace IS

V této kapitole se budeme věnovat popisu realizace informačního systému. Popíšeme strukturu definice formuláře uloženého v XML souboru, dle kterého pomocí PHP funkcí se vygeneruje HTML stránka s definovanými vazbami na databázové tabulky.

4.5.1 Struktura definice formuláře v XML

Definice formuláře se tvoří v XML souborech. Podle kterých je generována HTML stránka pomocí PHP.

Tabulka 5 - Přehled tagu pro definici formuláře

Název tagu	Atributy	Typ	Povinný	Popis tagu
LAYOUT		object	Ano	Definovat rozvržení plochy pro kontejnery
	ID	string	Ano	Identifikační kód layoutu
	title	string	Ne	Titulek layoutu
	direct_fill_data	boolean	Ne	Přímé zobrazení dat y databáze bez zadání filtru

Název tagu	Atributy	Typ	Povinný	Popis tagu
	show_button	boolean	Ne	Zobrazení tlačítek na konci layoutu
HBOX		object	Ne	seskupuje několik gridu nebo tabulek položek do stejného rámečku vodorovného rozvržení uvnitř kontejneru.
	ID	string	Ano	Identifikátor virtuálního boxu
	title	string	Ne	Titulek
	show_button	boolean	Ne	Zobrazení tlačítek na konci HBOXu
	area	string	Ne	Identifikace oblasti: head – Oblast hlavičky dokladu detl – Oblast řádek dokladu
VBOX		object	Ano	Definuje oblast rozvržení pro vykreslení podřízených prvků ve vswislém směru
	ID	string	Ano	Identifikace virtuálního boxu
	title	string	Ne	Titulek virtuálního boxu
		boolean	Ne	Kontejner pro zobrazení filtru na výběr dat z databáze
	show_button	boolean	Ne	Zobrazení tlačítek na konci VBOXu
	area	string	Ne	Identifikace oblasti: head – Oblast hlavičky dokladu detl – Oblast řádek dokladu
	show_box4insert	boolean	Ne	Povolení zobrazení BOXu v režimu zápisu nových dat
	show_box4detail	boolean	Ne	Povolení zobrazení BOXu v režimu editace existujících dat
	clone_button	boolean	Ne	Zobrazení tlačítka na kopírování DB záznamu.
GRID		object	Ano	Definuje oblast rozvržení buněk do pomyslné mřížky. Můžete určit polohu formulářových polí pro zadávání dat nebo dalších interaktivních objektů, jako jsou tlačítka.
	ID	string	Ano	Identifikace pro GRID
	db_primary_key	string	Ne	Název sloupce databázového tabulky určující primární klíč pro výběr dat
	show_button	boolean	Ne	Zobrazení tlačítek na konci VBOXu
	detail_flag	boolean	Ne	Zobrazení gridu pro editaci již existujících dat
	insert_flag	boolean	Ne	Zobrazení gridu pro zápis nových údajů
	group	string	Ne	Ohraničení gridu s názvem skupiny Příklad:
	font_Size	Integer	Ne	Velikost písma v pixelech.
TABLE		object	Ne	Definuje tabulku určenou k zobrazení seznamu záznamů.
	ID	string	Ano	Identifikátor tabulky
	db_primary_key	string	Ne	Název sloupce databázového tabulky určující primární klíč pro výběr dat do tabulky
	limit_row	integer	Ne	Limit pro zobrazení řádek z DB tabulky
	height_px	integer	Ne	Výška tabulky v pixelech
	ins_row_flag	boolean	Ne	Povolení vkládat nové záznamy
	delete_row_flag	boolean	Ne	Povolení rušit záznamy

Název tagu	Atributy	Typ	Povinný	Popis tagu
	detail_flag	boolean	Ne	Povolení zobrazení detailu (dalších informací, které se nevešly do tabulky)
	font_Size	Integer	Ne	Velikost písma v pixelech.
ITEM		object	Ano	Položka Gridu
	ID	string	Ano	Identifikace položky
COLUMN		string	Ano	Sloupec tabulky
	ID	string	Ano	Identifikace sloupce
NAME		string	Ne	Název sloupce / Název položky
DB_TABLE		string	Ne	Název tabulky v databázi
DB_COLUMN		string	Ne	Název sloupce v DB tabulce
DB_REF_TABLE		string	Ne	Vazební DB tabulka
DB_REF_COLUMN		string	Ne	Sloupec vazební tabulky (primární klíč)
TYPE		string	Ne	Databázový typ. Např. INTEGER, DATE
SIZE		ineger	Ne	Velikost ve znacích pro zobrazení v HTML
UPSIFT		char	Ne	Automatické psaní velkých písmen Y/N
LABEL		string	Ne	Text vpravo od položky formuláře. Pokud je použit prefix _DBC. a za ním sloupec DB tabulky pak se přednastaví hodnota z tohoto sloupce, pokud již hodnota není vyplněna uživatelem.
POSITION		string	Ne	Pozice položky: left, right, center
DEFAUL_VALUE		string	Ne	Implicitní hodnota položky. Pokud je použit prefix _DBC. a za ním sloupec DB tabulky pak se přednastaví hodnota z tohoto sloupce, pokud již hodnota není vyplněna uživatelem. Parametrické výrazy – Význam \$TIME: Aktuální čas \$USER: Přihlašovací jméno uživatele \$TODAY: Aktuální datum \$YEST: Včerejší datum \$TOMM: Zítřejší datum \$YEAR: Aktuální rok \$PER: Aktuální měsíc \$LPYEAR Rok: minulého měsíce \$LPPER: Minulí měsíc
COMMENT		string	Ne	Komentář k položce
MODE		string	Ne	Mód Editace/Zobrazení
REQUIRED		char	Ne	Povinný údaj Y/N
READONLY		char	Ne	Jen pro čtení Y/N
COMBO		object	Ne	Vyklápěcí položka
C_ITEM		object	Ne	Položka comba
VALUE		string	Ano	Hodnota
TEXT		string	Ano	Popisný text hodnoty položky comba

Zdroj: Vlastní zpracování, 2021

[Odkaz na XML soubor učení ke generování formuláře příjemky.](#)

Ukázka XML souboru:

```
<?xml version="1.0" encoding="UTF-8"?>
<LAYOUT ID="W108" title="Přímý příjem do skladu"
  direct_fill_data="true" show_button="false">
  <VBOX ID="W108.a" show_box4insert="true" show_box4detail="true">
    <GRID ID="stockhead" db_primary_key="stock_id" show_button="true"
      detail_flag="Y" insert_flag="Y" font_size="14">
      <ITEM ID="type_stock">
        <NAME>Typ Doklad</NAME>
        <DB_TABLE>stockhead</DB_TABLE>
        <DB_COLUMN>type_stock</DB_COLUMN>
        <TYPE>char</TYPE>
        <SIZE>1</SIZE>
        <UPSIFT>N</UPSIFT>
        <POSITION>left</POSITION>
        <DEFAULT_VALUE>P</DEFAULT_VALUE>
        <LABEL>Příjemka</LABEL>
        <COMMENT>Typ Doklad: P-Příjemka, V-Výdejka</COMMENT>
        <MODE>EDIT</MODE>
        <REQUIRED>Y</REQUIRED>
        <READONLY>Y</READONLY>
        <COMBO>
          <C_ITEM checked="true">
            <VALUE>P</VALUE>
            <TEXT>Příjemka</TEXT>
          </C_ITEM>
          <C_ITEM checked="false">
            <VALUE>V</VALUE>
            <TEXT>Výdejka</TEXT>
          </C_ITEM>
        </COMBO>
      </ITEM>
      <ITEM ID="ware_code">
        <NAME>Sklad</NAME>
        <DB_TABLE>stockhead</DB_TABLE>
        <DB_COLUMN>ware_code</DB_COLUMN>
        <TYPE>char</TYPE>
        <SIZE>3</SIZE>
        <UPSIFT>N</UPSIFT>
        <POSITION>left</POSITION>
        <DEFAULT_VALUE>SUR</DEFAULT_VALUE>
        <COMMENT>Sklad</COMMENT>
        <MODE>COMBO_EDIT</MODE>
        <REQUIRED>Y</REQUIRED>
        <READONLY>N</READONLY>
        <DB_REF_TABLE>warehouse</DB_REF_TABLE>
        <DB_REF_COLUMN>ware_code</DB_REF_COLUMN>
        <COMBO db_table="warehouse">
          <C_ITEM checked="true">
            <VALUE>ware_code</VALUE>
            <TEXT>desc_text</TEXT>
          </C_ITEM>
        </COMBO>
      </ITEM>
    </GRID>
  </VBOX>
</LAYOUT>
```

4.5.2 Programová část

Z důvodu velkého rozsahu zdrojového kódu budou v této práci uvedeny pouze vybrané části jedné PHP třídy FormCore, která za pomoci dalších tříd a funkcí vytváří blok HTML formulář. Celý programový kód nebude v této práci uveden. Jednotlivé části jsou okomentovány v rámci kódu.

```
<?php
/**
 *
 * Třída je určena k vytvoření hlavního bloku formuláře
 */
class FormCore
{
    private $pdo;
    private $updateSQL;
    private $formStructure;
    private $progressText = null;
    private $tempFlag = false;
    private $tableTop;
    private $cscProgress;

    // formulář pro editaci
    private $formEditAction;
    private $keyName;
    private $noFiltred = true;

/**
 * Třída je určena k vytvoření HTML formuláře
 * @param isPDO $pdo
 * @param string $keyName například Příjemka dle struktury XML souboru
 * @param FormStructure $formStructure - Třída definice struktury formuláře
 */
    public function __construct(isPDO $pdo, $keyName, FormStructure $formStructure) {
        $this->pdo = $pdo;
        $this->formStructure = $formStructure;
        $this->keyName = $keyName;

        $sql = "create table if not exists filter_data (user char(20), form_id char(20), tabname
varchar(30), colname varchar(30), value varchar(500), dbtype char(20), exact bit, required bit,
negation bit, caseSensitive bit)";
        //      echo "<BR>".$sql;
        if ($pdo->exec($sql) === false){
            echo '<p>Selhalo vytvoření tabulky filter_data';
            echo '<p>' . $sql;
            die();
        }

        if (!isset($_SESSION["LAYOUT_ID"])){
            $_SESSION["LAYOUT_ID"] = $formStructure->getFormID();
        } else {
            if ($_SESSION["LAYOUT_ID"] != $formStructure->getFormID()){
                $_SESSION["LAYOUT_ID"] = $formStructure->getFormID();
                unset($_SESSION["_AreaKey_Table_det1_key"]);
                unset($_SESSION["_AreaKey_Table_det1_column"]);
                unset($_SESSION["_AreaKey_Table_head_key"]);
                unset($_SESSION["_AreaKey_Table_head_column"]);
                unset($_SESSION["_AreaKey_Grid_head_key"]);
                unset($_SESSION["_AreaKey_Grid_det1_key"]);
            }
        }
        $this->initForm(); //Resetuje prmoměnné session
    }
}
```

```

/**
 * Výkonná funkce, která vygeneruje HTML stránku
 */
public function run() {
    $this->initFormCore();
    $this->actionForm();

    if ($this->progressText == null){
        $this->progressText = "Probíhá načtení dat";
    }
    $this->genPage($this->progressText);
}

/**
 * Funkce na Inicializaci Formuláře.
 * Dále funkce zkoumá, zdali byla vyvolaná akce pomocí tlačítek: Kopie, Reset filtru, Filtr,
 * Insert řádky a OK při potvrzení filtru
 */
public function initFormCore() {
    if (isset($_SESSION["updateSQL"])){
        $this->updateSQL = $_SESSION["updateSQL"];
    }

    $this->noFiltred = true;
    $_SESSION["noFiltred"] = "A";
    $_SESSION["show_last_data"] = false;
    $_SESSION["insRow"] = "N";
    $_SESSION["tempFlag"] = $this->tempFlag;

    generateCancelButton();

    if ($this->tempFlag)
        generateInsertDoClButton("Zápis - " . $this->keyName);

    //Test na požadavek kopie záznamu v DB
    if (isset($_COOKIE['Copy_ID_row'])){
        if ($_COOKIE['Copy_ID_row'] != "null") {
            //Má se udělat kopie DB záznamu na kód dle zadání uživatelem.
            $idRow = $_COOKIE["Copy_ID_row"];
            $value = $_COOKIE["Copy_db_row"];
            echo '<script type="text/javascript"> document.cookie = "Copy_ID_row=null";
</script>';

            // echo "<BR>value = ". $value;
            $formCheck = new FormCheck($this->pdo, $this->keyName, $this->formStructure,
false);

            if ($formCheck->checForm()){
                $msg = cloneDBRow($this->pdo, $idRow, $value);
                if (isset($msg)) {
                    echo getErrorMsg($msg);
                }
            } else {
                $msg = $formCheck->getMsg();
                echo "<font color=red><b>".$msg."</b></font>";

                $_SESSION["show_last_data"] = false;
                $_SESSION["primaryKey"] = $value;
            }

            $this->noFiltred = false;
            $_SESSION["noFiltred"] = "N";
        }
    }

    $this->formEditAction = new PrHtmlForm("editAction", null);
    $this->formEditAction->setNoPrint();
    echo $this->formEditAction->publishStart();

    if ($this->formEditAction->wasSentButton(FormTableCreator::ACTION_CLEAR_FILTER)){
        $msg = deleteFilter($this->pdo, $this->formStructure);
        if (isset($msg)) {
            echo getErrorMsg($msg);
        }
    }
}

```

```

    } else {
        $this->noFiltred = false;
        $_SESSION["noFiltred"] = "N";
    }
}

if (($this->formStructure->getDirectFillData() ||
    $this->formEditAction->wasSentButton(FormTableCreator::ACTION_INSERT_ROW)) &&
    !$this->formEditAction->wasSentButton(FormTableCreator::ACTION_NEW_FILTER))
{
    //Data se načítají rovnou bez výběrového formuláře
    $this->noFiltred = false;
    $_SESSION["noFiltred"] = "N";
}

if ($this->formEditAction->wasSentButton(FormTableCreator::ACTION_INSERT_ROW)){
    //Vložení nového záznamu
    $_SESSION["insRow"] = "Y";
}

if ($this->formEditAction->wasSentButton(FormGridCreator::ACTION_GOTO_QUERY_OK)) {
    //Provádí se výběr dat

    $msg = createQuery($this->pdo, $this->formStructure);

    if (isset($msg)) {
        echo getErrorMsg($msg);
    } else {
        $this->noFiltred = false;
        $_SESSION["noFiltred"] = "N";
    }
}

if ($this->formEditAction->wasSentButton(FormTableCreator::ACTION_DELETE_TABLE_ROWS)) {
    $msg = deleteRows($this->pdo);
    if (isset($msg)) {
        echo getErrorMsg($msg);
    }
    $this->noFiltred = false;
    $_SESSION["noFiltred"] = "N";
}

if ($this->formEditAction->wasSentButton(FormTableCreator::ACTION_GOTO_DETIAL)) {
    setSesionPrimaryKeyFordetail();
    $this->noFiltred = false;
    $_SESSION["noFiltred"] = "N";

    $postData = new FormPostDataLib(false);
    $_SESSION["_AreaKey_Grid_".$postData->getArea()."_key"] = $postData->getIdDetail();
    if ($postData->getArea() == "head"){
        $_SESSION["_AreaKey_Table_detl_key"] = $postData->getIdDetail();
        $_SESSION["_AreaKey_Table_detl_column"] = $postData->getColumn4Detail();
        $_SESSION["_AreaKey_Grid_detl_key"] = "NULL";
    }
}
}

/**
 * Inicializace pro tisk formuláře. Funkce nastaví začátek oblasti určené k tisku a vygeneruje
 začátek form.
 */
public function initFormCore4Print() {
    echo '<input type="button" onclick="printDiv(\'printableArea\')" value="Tisk" />';
    echo '<div id="printableArea">';

    echo "\n";
    echo '<style type="text/css">';
    echo "\n";
    echo "\n";
    // potlačení objektu v tisku (class="noprint")
    echo '@media print';
}

```

```

echo '{';
echo '.noprint {display:none;}';
echo '}';
echo "\n";

// odstránkování tisku - styl pomocného div
echo '@media print';
echo '{';
echo '.page-break { display: block; page-break-before: always; }';
echo '}';
echo "\n";
echo "\n";
echo '</style>';
echo "\n\n";

$_SESSION["noFiltred"] = "N";
$_SESSION["show_last_data"] = false;
$_SESSION["insRow"] = "N";
$_SESSION["tempFlag"] = $this->tempFlag;
$_SESSION["print"] = true;
$this->noFiltred = false;
}

/**
 * Funkce vytvoří HTML kód a zobrazí jej přes echo
 * @param string $progressText - Text pro zobrazení jaké data se právě načítají
 */
public function genPage($progressText){
    // tabulka pro rozvržení stránky
    $this->tableTop = new PrHtmlTable();

    $this->csProgress = new PrProgress();
    $this->csProgress->start($progressText);

    $this->genPageImpl($progressText);

    echo $this->tableTop->publish() . $this->formEditAction->publishEnd();

    $this->csProgress->finish(1);
}

/**
 * Funkce vytvoří tabulku určenou k rozvržení stránky.
 * Volá se jen jednou před přidáváním řádek do hlavní tabulky.
 * Pak se zavolá funkce genPage4Print ta se může volat opakovaně.
 * Dokud se nepoužije funkce na zveřejnění celého HTML kódu
 *
 * @param string $progressText - Text pro zobrazení jaké data se právě načítají
 */
public function initPage4Print($progressText){
    // tabulka pro rozvržení stránky
    $this->tableTop = new PrHtmlTable();
    $this->tableTop->setPageBreakAvoid();

    $this->csProgress = new PrProgress();
    $this->csProgress->start($progressText);
}

/**
 * Funkce připravuje HTML kód. Tento kód se zveřejní pomocí funkce publishPage4Print
 */
public function genPage4Print(){
    $this->genPageImpl();
}

public function pageBreak(){
    $this->tableTop->addRow();
    $cellText = '<p><div class="page-break">';
    $cellText .= '</div>';
    $this->tableTop->addRowCell("");
    $this->tableTop->addRowCell($cellText);
}

```

```

/**
 * Funkce zveřejňuje HTML kód pomocí echa a ukončuje progressbar a ukončuje DIV určený k
 * definici oblasti určené k tisku
 */
public function publishPage4Print(){
    echo $this->tableTop->publish() . $this->formEditAction->publishEnd();

    $this->csProgress->finish(1);
    $this->finishForm();
}

/**
 * Hlavní funkce na tvorbu HTML
 */
private function genPageImpl(){
    $this->tableTop->addRow();

    $cellFormID = $this->tableTop->addRowCell($this->formStructure->getFormID());
    $cellFormID->setFontSize(8);
    $cellTitle = $this->tableTop->addRowCell($this->formStructure->getFormTitle());
    $cellTitle->setAlignCentered();
    $cellTitle->setWidth(900);
    $cellTitle->setTextBold();

    $this->tableTop->addRow();

    $formLayout = $this->formStructure->getFormLayout();

    $lastBox = null;
    $firstGrid = true;
    $_SESSION["existsFilterBox"] = false;
    foreach($formLayout->getBoxList() as $formBox) {
        $typeBox = $formBox->getTypeBox();
        if (!$formBox->isShowBox4Insert() && $_SESSION["insRow"] == "N" &&
            !$this->formEditAction->wasSentButton(FormTableCreator::ACTION_GOTO_DETIAL)) &&
            $_SESSION["print"] == false){
            //BOX se zobrazuje jen při vkládání
            continue;
        }
        if (!$formBox->isShowBox4Detail() && $_SESSION["insRow"] == "Y" &&
            $this->formEditAction->wasSentButton(FormTableCreator::ACTION_GOTO_DETIAL)){
            //BOX se zobrazuje jen při detailu
            continue;
        }

        if ($formBox->getFilter() && $_SESSION["print"] == false){ //Při tisku se ikony filtru
nezobrazují
            //Existuje BOX s filtrem
            $_SESSION["existsFilterBox"] = true;
        }

        if (!$formBox->getFilter() && $this->noFiltred){
            //Neni filtr
            continue;
        }
        if ($formBox->getFilter() && !$this->noFiltred){
            //Jen BOX typu filtr
            continue;
        }

        $this->tableTop->addRow();
        $this->tableTop->addRowCell("");
        //    }

        if ($lastBox != null && $lastBox->getTypeBox() == FormStructure::HBOX && $lastBox-
isShowButton()) {
            $tableFooter = new PrHtmlTable();
            $tableFooter->setWidthPercent(100);
            $tableFooter->addRow();
            $tableFooter->addRowCell(generateFormButtons($this->noFiltred))->setAlignCentered();

```

```

        $this->tableTop->addRow();
        $this->tableTop->addRowCell("");
        $this->tableTop->addRowCell($tableFooter->publish());
    }

    $lastBox = $formBox;

    if ($typeBox == FormStructure::VBOX){
        $boxCrt = new FormBoxCreator($this->pdo, $this->formStructure->getFormID(),
$formBox, $firstGrid);
        $this->tableTop->addRowCell($boxCrt->getHtml());//->setAlignCentered();
        $firstGrid = $boxCrt->getFirstGrid();

        if ($boxCrt->getUpdateSQL() != null ) {
            $this->updateSQL = $boxCrt->getUpdateSQL();
        }

        } else {
            $boxCrt = new FormBoxCreator($this->pdo, $this->formStructure->getFormID(),
$formBox, $firstGrid);
            $firstGrid = $boxCrt->getFirstGrid();
            $this->tableTop->addRowCell($boxCrt->getHtml());//->setAlignCentered();
            if ($boxCrt->getUpdateSQL() != null ) {
                $this->updateSQL = $boxCrt->getUpdateSQL();
            }
        }
    }

    if ($this->updateSQL != null){
        $_SESSION["updateSQL"] = $this->updateSQL;
    }

    if ($this->formStructure->isShowButton()) {
        $this->tableTop->addRow();
        $this->tableTop->addRowCell("");
        $this->tableTop->addRowCell(generateFormButtons($this->noFiltred));
    }
}
/**
 * Funkce ukončí DIV určený k vymezení tiskové oblasti
 */
private function finishForm(){
    echo '</div>'; //printableArea

    $js = '<script type="text/javascript" charset="UTF-8">
function printDiv(divName) {
    var printContents = document.getElementById(divName).innerHTML;
    var originalContents = document.body.innerHTML;
    document.body.innerHTML = printContents;
    window.print();
    document.body.innerHTML = originalContents;
}
    </script>';
    echo $js;
}

/**
 * Funkce zkoumá zdali byla vyvolaná akce pomocí tlačítka na stránce (Zapsat, Uložit, Storno
atd.)
 * @return string - Vrací případnou chybu
 */
public function actionForm(){
    if ($this->formEditAction->wasSentButton(FormGridCreator::ACTION_GOTO_STORNO)) {
        //Bylo stisknuto tlačítka STORNO
        $this->noFiltred = true;
        $_SESSION["noFiltred"] = "A";
        $_SESSION["updateSQL"] = null;
        $_SESSION["primaryKey"] = null;

        $sql = "delete from filter_data where user = :user and form_id = :form_id";
        $this->pdo->setNamedParam("user", $_SESSION["is_user"]);
    }
}

```

```

$this->pdo->setNamedParam("form_id", $this->formStructure->getFormID());
if ($this->pdo->exec($sql) === false){
    return "Chyba filtru dat";
}
}

if ($this->formEditAction->wasSentButton(FormTableCreator::ACTION_SAVE_LINES) ||
    $this->formEditAction->wasSentButton(FormGridCreator::ACTION_GOTO_EDITOR_UPD_SAVE)) {
    //Aktualizace dat
    $custCheck = new FormCheck($this->pdo, $this->keyName, $this->formStructure, false);

    if ($custCheck->checForm()){
        $msg = saveChanges($this->pdo);
        if (isset($msg)) {
            echo getErrorMsg($msg);
        }
    } else {
        $msg = $custCheck->getMsg();
        echo "<font color=red><b>". $msg. "!</b></font>";

        $_SESSION["show_last_data"] = true;
        $_SESSION["primaryKey"] = $custCheck->getPrimaryKey();
    }
    $this->noFiltred = false;
    $_SESSION["noFiltred"] = "N";
}

if ($this->formEditAction->wasSentButton(FormGridCreator::ACTION_GOTO_EDITOR_INS_SAVE)) {
    //Zápis nových dat
    $custCheck = new FormCheck($this->pdo, $this->keyName, $this->formStructure, true);
    if ($custCheck->checForm()) {
        $msg = InsertIntoDB("Zápis - " . $this->keyName, $this->pdo, false);
        if (isset($msg)) {
            echo getErrorMsg($msg);
        }
    } else {
        $msg = $custCheck->getMsg();
        echo "<font color=red><b>". $msg. "!</b></font>";
        $_SESSION["show_last_data"] = true;
        $_SESSION["primaryKey"] = null;
        $_SESSION["insRow"] = "Y";
    }
    $this->noFiltred = false;
    $_SESSION["noFiltred"] = "N";
}
}
/**
 * Funkce nastavuje text určitý k zobrazení v progressbaru
 * @param string $progressText - Popis jaké data se načítají
 */
public function setPregressText($progressText) {
    if ($progressText == null){
        $progressText = "Probíhá načtení dat";
    }
}
/**
 * Nastavení pomocného režimu, kdy se data ukládají jako neplatná (valid_flag = false)
 * @param boolean $tempFlag
 */
public function setTempFlag($tempFlag) {
    $this->tempFlag = $tempFlag;
}
/**
 * Statická funkce nastavuje pro editační políčko ($editText) regulární výraz, kterým se pak
 provádí validace zadaného údaje.
 * @param string $dbType - databázový typ: date, decimal2 - na 2 desetiny, integer, smallint
 * @param PrHtmlText $editText
 */
public static function setPattern4EditText($dbType, PrHtmlText $editText) {
    switch ($dbType) {
        case FormStructure::TYPE_DATE:
            $editText->setPatternDate();
    }
}

```



```

        break;
    case FormStructure::TYPE_DEC:
        $editText->setPatternDecimal4();
        break;
    case FormStructure::TYPE_DECIMAL:
        $editText->setPatternDecimal4();
        break;
    case FormStructure::TYPE_DEC2:
        $editText->setPatternDecimal2();
        break;
    case FormStructure::TYPE_DECIMAL2:
        $editText->setPatternDecimal2();
        break;
    case FormStructure::TYPE_INT:
        $editText->setPatternIntegerUnsign();
        break;
    case FormStructure::TYPE_INTEGER:
        $editText->setPatternIntegerUnsign();
        break;
    case FormStructure::TYPE_SMALLINT:
        $editText->setPatternIntegerUnsign();
        break;
    }
}
/**
 * Vrací třídu s formulářem
 * @return PrHtmlForm
 */
public function getFormEditAction() {
    return $this->formEditAction;
}
/**
 * Vrací TRUE pokud nebyl ještě zobrazený filtr pro zadání výběrové podmínky
 * @return boolean
 */
public function getNoFiltred() {
    return $this->noFiltred;
}
/**
 * @param boolean $noFiltred
 */
public function setNoFiltred($noFiltred) {
    $this->noFiltred = $noFiltred;
}
}
}

```

4.6 Tiskové výstupy IS

Potravinářská firma a firmy obecně tvoří tiskový výstupy ze svých IS. V mé práci není kladen důraz na tuhle část. Pokud by byla řešena, mohl by se použít např. BIRT.

4.7 Zabezpečení systému

Zabezpečení systému má za úkol, aby nepovolaný uživatel nezískal přístup do systému případně informace pro přístupu do něj. V rámci systému jsou vytvořeny práva a ty jsou aplikována na každou stránku. Každý uživatel má přidělenou úroveň pravomocí.

Jednotlivé komponenty, co zajišťují bezpečnost IS:

- Login do IS
- Šifrování (HTTPS)
- Přístupová práva k menu IS pro jednotlivé uživatele.

4.7.1 Login do IS

Pokud je vyžadována autentizace, webová aplikace vygeneruje HTML formulář s poli potřebnými pro úspěšné ověření uživatele. Formulář se odesílá výhradně http metodou POST (metoda GET předává odeslané parametry v adresním řádku, takže se ukládají do historie prohlížeče a případných logů). Po úspěšné autentizaci vygeneruje server session token, který je odeslán uživateli a uložen v prohlížeči (nejčastěji ve formě cookies).

Obrázek 8 - Ukázka přihlašovacího HTML formuláře

Login:

Heslo:

Zdroj: Vlastní zpracování, 2021

4.7.2 Šifrování

HTTPS je protokol, který zprostředkovává zabezpečenou komunikaci v počítačové síti. Jde o kombinaci dvou protokolů http a SSL nebo TLS. Jelikož se využívá primárně pro komunikaci webového prohlížeče s webovým serverem, je jeho použití více než žádoucí, aby data byla ochráněna. Jelikož HTTPS není zdarma, nemohl jsem tento protokol využít. Proto je v mé práci použit pouze protokol HTTP.

4.7.3 Přístupová práva k menu IS pro jednotlivé uživatele

Další z nástrojů zabezpečení je používání přístupových práv do jednotlivých částí systému. Na jednotlivé sekce a stránky jsou udělené nároky v podobě číselné hodnoty (čím vyšší, tím větší nárok na pravomoc uživatele). Uživatel tedy, který má nejnižší práva neuvidí některé sekce a stránky, aby nemohl nijak zasahovat do chodu IS tam, kde nemá. Např. uživatel s hodnotou pravomocí 5 uvidí všechno, co má hodnotu 5 a nižší. Pro systém je i vytvořená položka u uživatele „Admin“, která, když je zaškrtnutá, umožňuje uživateli vidět, upravovat a přidávat v rámci celého systému.

5 Výsledná aplikace

5.1 Odkaz na aplikaci IS na webu

<http://perut.www3.cz/is>

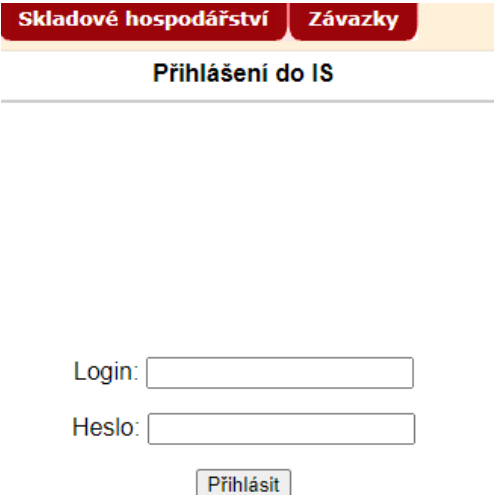
Přihlašovací údaje:

- uživatel: **guest**
- heslo: **guest**

5.2 Přihlášení do aplikace

Uživatel poté, co zadá webové stránky IS, či rozklikne odkaz uvedený výše, otevře se mu stránka s přihlášením do systému. Každý uživatel má přidělený svoje přihlašovací údaje. Poté co se zadají správné kombinace loginu a hesla, systém vyhodnotí správnost a buď uživatele pustí dál, či mu přístup bude odepřen. Každý uživatel je přihlášený pod svým účtem, který má svou hodnotu oprávnění.

Obrázek 9 - Přihlášení do aplikace



Skladové hospodářství Závazky

Přihlášení do IS

Login:











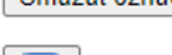




Heslo:

Přihlásit

Zdroj: Vlastní zpracování, 2021

5.4 Funkční tlačítka používané v aplikaci IS

Tabulka 6 - Přehled funkčních tlačítek v aplikaci IS

Tlačítko	Popis
<p>Obrázek 11: Funkční tlačítka v IS</p>	
1. 	1. Odhlášení přihlášeného uživatele z aplikace.
2. 	2. Storno údajů formuláře.
3. 	3. Přidání nového záznamu.
4. 	4. Tlačítko filtru pro zdání výběrové podmínky.
5. 	5. Tlačítko zrušení filtru.
6. 	6. Zobrazení detailu řádky tabulky.
7. 	7. Označení řádky tabulky ke smazání.
8. 	8. Zobrazení kalendáře pro položku formuláře typu datum.
9. 	9. Kopie databázového objektu pod novým kódem.
10. 	10. Uložení nového záznamu do DB.
11. 	11. Uložení změn v existujícím záznamu v DB.
12. 	12. Smazání řádek tabulky označených ke smazání.
13. 	13. Uložení nepotvrzených dat do databáze. Např. potvrzení skladové příjemky.
14. 	14. Potvrzení výběrové podmínky zadané ve formuláři filtru.
15. 	15. Tlačítko zpět vrací uživatele na předchozí stránku.
Zdroj 1: Vlastní zpracování, 2021	

Zdroj: Vlastní zpracování, 2021

5.5 Uživatelé

Menu slouží k zakládání a rušení uživatelů IS. Pro uživatele se nastavuje login, jeho heslo a přístupová práva k jednotlivým menu aplikace.

Obrázek 12 - Ukázka editace nebo přidání uživatele IS

The screenshot shows a web application interface for user management. At the top, there is a navigation bar with tabs: Pohledávky, Objednávky, Účetnictví, Skladové hospodářství, Závazky, and Systémové služby. The user 'Petr Ruttkay' is logged in. The main content area is titled 'Uživatelé systému' and 'DETAIL'. It contains a form with the following fields: Kód uživatele (AAA), Plné jméno (Petr Ruttkay), Úroveň zabezpečení (2), Heslo (aaa), Datum nastavení (20.07.2010), Telefonní číslo, Email (blabla@seznam.cz), Mobil, Administrátor databáze (ANO), and Administrátor informačního systému (ANO). There are buttons for 'Uložit změny' and 'Zpět'. Below the form is a section titled 'Seznam uživatelů' with a table listing users.

Kód uživatele	Plné jméno	Úroveň zab.	Mobil	Admin DB	Admin IS		
AAA	Petr Ruttkay	2		ANO	ANO		
GUEST	Anonym	1		NE	NE		

Zdroj: Vlastní zpracování, 2021

Kód uživatele – Kód uživatele pro identifikaci uživatel v IS (Povinný údaj).

Plné jméno – Jméno a příjmení uživatele

Úroveň zabezpečení – Práva uživatele na menu IS (číslo 1-9)

Heslo – Heslo uživatele pro přihlášení do IS

Datum nastavení – Datum narození uživatele,

Email – E-mail uživatele

Mobil – Telefonní číslo uživatele

Administrátor databáze – Nabývá hodnot Ano/Ne (Povinný údaj)

Administrátor informačního systému – Nabývá hodnot Ano/Ne (Povinný údaj)

5.6 Menu aplikace IS

Nabídka menu IS je uložené v databázi a lze jej konfigurovat pomocí zadávacího formuláře administrátorem systému.

Obrázek 13 - Ukázka formuláře pro zadání nebo editaci menu

Název hlavního menu	Rodič ID	Třídící klíč	Práva	Menu ID	Popis menu	Podmenu			
Skladové hospodářství		10	1	45	Produkt - zápis	Ne			<input type="checkbox"/>
Skladové hospodářství		20	1	48	Produkty - výběr	Ne			<input type="checkbox"/>
Skladové hospodářství		30	1	46	Skladová karta - zápis	Ne			<input type="checkbox"/>

Zdroj: Vlastní zpracování, 2021

Popis údajů formuláře

Menu ID – Informativní údaj generovaný automaticky (Povinný údaj).

Název hlavního menu – Název menu v hlavní nabídce (např. Pohledávky) (Povinný údaj).

Popis menu – Název menu.

Třídící klíč – Třídící číslo, podle kterého bude méně setříděno

Práva – Úroveň přístupu (1-9). Uživatel nemá právo spustit menu, která mají úroveň přístupu menší než tato hodnota.

Soubor – PHP soubor určený pro tuto funkci aplikace

Parametry – Parametry předávané metodou GET pro zobrazení stránky v internetovém prohlížeči.

Podmenu – Nabývá hodnot Ano/Ne. Pokud je nastaveno Ano lze pro toto menu vytvořit podmenu. Lze vytvořit maximálně jedno podmenu. (Povinný údaj).

5.7 Filtrování (výběr) dat v IS

Pro formuláře, které zobrazují data v tabulce lze zadat podmínky pro výběr dat. Aplikace si pamatuje posledně zadané výběrové údaje zadané uživatelem a tyto údaje po volbě menu znovu přednastaví. Původní údaje lze vyresetovat tlačítkem „Storno údajů formuláře“ viz tabulka 5 - [Přehled funkčních tlačítek v aplikaci IS](#).

Dále je možnost provést negaci výběrové podmínky u jednotlivých údajů, které nepožadují zadat exaktně konkrétní hodnotu a u znakových údajů lze zvolit rozlišování velikosti písma.

Obrázek 14 - Ukázka formuláře pro zadání výběru

W105

Výběr Produktů

Kód produktu

Popis produktu

Druhá řádka pro popis produktu

Kategorie produktu

Skladovací třída

Alternativní produkt

MJ skladu

EAN

Kód celního sazebníku

Prodejní ceny za

Příznak sledování šarží -vše-

Povinnost data trvanlivosti -vše-

Vyřazený produkt -vše-

OK Zpět

Zdroj: Vlastní zpracování, 2021

Při zadávání výběrového kritéria můžete používat následující operátory:

- = rovno
- > větší než
- < menší než
- >= větší než nebo rovno
- <= menší než nebo rovno
- <> nerovná se (lze použít také !=)
- * náhradní znak
- ? náhradní znak
- : rozsah
- | alternativa

Prvních šest operátorů se nazývá relační operátory a mohou být použity jako popis přípustných hodnot pro jednotlivá pole. Operátory jsou nazývány relační, protože popisují relaci mezi daty, která uživatel hledá.

Obvykle uživatel nezadáva znak rovnosti. Jestliže uživatel zadá hodnotu, systém předpokládá relaci "rovno". Například, jestliže uživatel zadá KG do pole MJ skladu, systém vyhledá všechny produkty, kteří mají v údaji MJ skladu vyplněn kód KG. Systém předpokládá význam:

MJ skladu = "KG"

Následující odstavce vysvětlí, jak použít relační operátory v rozdílných typech polí:

Znakové pole

Zadejte operátor a jakýkoliv znak. Například pro vyhledání produktu s MJ v KG zadáte =KG do pole *STÁT*.

Systém porovná znaky v závislosti na hodnotách v kódování v UTF8 v závislosti na nastavení databáze, jak se mají znaky porovnávat.

Použití náhradních znaků

Znaky hvězdička (*) a otazník (?) jsou náhradní znaky, které mohou být použity pouze ve znakových polích. Hvězdička nahrazuje jakoukoliv skupinu znaků (od žádného do několika znaků), zatímco otazník nahrazuje pouze jeden znak.

Abychom například našli všechny produkty, jejichž název začíná znakem M, zapíšete do pole *POPIS PRODUKTU* výraz *K**. Můžete zadat opakovaně hvězdičku pro vyhledání hodnoty, která obsahuje pouze část textového řetězce. Jestliže tedy potřebujete vyhledat produkt, v jejichž názvu se vyskytuje řetězec *ouka*, zadáte do údaje *POPIS PRODUKTU VÝRAZ *ouka** a systém vyhledá názvy jako jsou "Mouka hladká" a "Mouka hrubá".

Můžete použít náhradního znaku otazník (?), abyste našli hodnoty, které odpovídají vzoru, kde je počet znaků pevný. Můžete tedy například zadat výraz *Pa?ez*, abyste našli *NÁZVY ODBĚRATELŮ* jako jsou "Pařez" a "Parez".

Vyhledání s alternativním operátorem

Uživatel může použít alternativní operátor pro vyhledání alternativních hodnot v poli. Abychom vyhledali zákazníky, jejichž názvy jsou "Karel" nebo "Petr", zadáme do pole výraz Karel|Petr.

Vyhledání rozsahu hodnot

Znak dvojtečka (:) je operátor rozsahu. Dovolí uživateli zadat všechny hodnoty, jejichž obsah leží mezi zadanými hodnotami. Rozsah je inkluzivní.

Abychom například vyhledali všechny zákazníky, jejichž jméno leží mezi Kovář a Novák včetně, zadáme do pole výraz Kovář:Novák.

Abychom vyhledali názvy ležící v abecední řadě za jménem Petřík (včetně) a vyloučili názvy začínající malým písmenem zadáme do pole jméno a výraz Petřík:Zzzzzz.

Nezapomeňte, aby počáteční hodnota byla menší než hodnota koncová.

5.8 Zápis příjmového skladového dokladu

Formulář se používá k příjmu produktu do skladu bez vazby na jiné moduly. Jakmile je dokončen a potvrzen záznam o příjmu produktu, systém aktualizuje stav produktu na skladě a vytváří odpovídající záznam v knize skladových pohybů.

Obrázek 15: Příjemka do skladu

The screenshot shows a web application interface for creating a warehouse receipt. On the left is a navigation menu with options like 'Skl.Hospodářství', 'Produkt - zápis', and 'Přímý příjem do skladu'. The main form is titled 'Přímý příjem do skladu' and 'NOVÝ ZÁZNAM'. It includes the following fields:

- Typ Doklad: P Příjemka
- Prvotní doklad: 20211115
- Datum dokladu: 26.11.2021
- Účetní rok: 2021
- Účetní období: 11
- Sklad: SUR - Sklad obilí
- Sřídisko: 000
- Účet příjmu: 001-111051 - Přímý nákup surovin
- Poznámka 1: (empty)
- Poznámka 2: (empty)
- Datum zdanitelného plnění: 26.11.2021

The 'Řádka příjemky' section is expanded and contains:

- ID řádky: 0
- ID hlavičky: 0
- Číslo řádky příjemky: 1
- Kód produktu: O100 - Pšenice jakost A,B,C
- Popis produktu: (empty)
- Množství MJ: 5000
- Jednotková cena: 5 Kč
- Celkem cena na řádce: (empty) Kč

Zdroj: Vlastní zpracování, 2021

Popis obsluhy formuláře:

Po zadání údajů v hlavičce a řádce dokladu uživatel IS stiskne tlačítko **Zapsat** a systém provede validaci údajů a pokud jsou údaje v pořádku provede se zápis do databáze. Tyto údaje jsou v nepotvrzeném stavu a lze v tomto okamžiku formulář opustit a následně se k rozepsanému dokladu vrátit.

Po zápisu řádky se v sekci řádky dokladu zobrazí první řádka dokladu a tabulka z již zapsanými řádky viz obrázek níže. K těmto řádkám se lze vrátit stiskem tlačítka „[Detail](#)“ viz. [5.4 tlačítko číslo 6](#) a provést případnou opravu některých údajů. Dále lze řádky dokladu označit ke zrušení a následně pomocí tlačítka **Smazat označené** zrušit v DB.

Zrušení nelze provést, pokud je formulář rozepsán, protože se při rušení záznamu provádí validace zobrazených údajů.

Obrázek 16: Řádka příjemky

Řádka příjemky

ID řádky

ID hlavičky

Číslo řádky příjemky

Kód produktu

Popis produktu

Množství MJ KG

Jednotková cena Kč

Celkem cena na řádce Kč

Řádky příjemky

ID	PČ.	Produkt	Množství MJ	JC	Celkem cena	
<input type="text" value="96"/>	<input type="text" value="1"/>	O100	<input type="text" value="5000.0000"/>	<input type="text" value="5.0000"/> Kč	<input type="text" value="25000.00"/> Kč	<input type="button" value="🔍"/> <input type="button" value="✖"/> <input type="button" value="□"/>

Zdroj: Vlastní zpracování, 2021

V případě, že uživatel zapsal všechny údaje lze doklad potvrdit do databáze tlačítkem „[Zápis](#)“ viz. [5.4 tlačítko číslo 13](#), které je uvedené v pravém horním rohu formuláře.

Poznámka:

Popis údajů formuláře je uveden v tabulce Vybrané tabulky z databáze.

6 Závěr

Tato bakalářská práce byla zaměřena na přiblížení možností vývoje webových stránek se zaměřením na použitelné prostředky pro tvorbu Informační systém podniku. V teoretické části je popsán vývoj a druhy informačních systémů. Dále byl popsán potravinářský marketing. Následně byly charakterizovány jednotlivé technologie pro tvorbu IS ve webovém prostředí.

Na začátku práce byla provedena analýza požadavků firmy na IS. Poté se vytvořilo schéma ve formě UML diagramu, kde byly rozvrženy potřebné třídy s jejich vazbami a primárními klíči.

Dalším krokem bylo vytvořit databázi podle UML diagramu. Postupně se vytvářely třídy s potřebnými atributy. Každá třída má svůj primární klíč a pomocí vazebních klíčů, které odkazují na tyto primární klíče jsou tabulky navzájem propojené podle UML diagramu.

Na řadu pak přichází samotný implementování informačního systému. Pomocí formulářů uložených v XML souboru se generují přes PHP funkce HTML stránky. Každý formulář je pomocí vazeb napojen na databázové tabulky.

Výsledná aplikace se na dané internetové stránce (<http://perut.www3.cz/is>) zobrazí v prohlížeči. Zde zaměstnanec zadá přidělené přihlašovací údaje a může s aplikací dále pracovat. Pro případné přihlášení:

login: guest

heslo: guest

7 Seznam použitých zdrojů

- [1] Josef Basl, ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI. *Podnikové informační systémy: podnik v informační společnosti*. Praha: Grada, 2008. ISBN 978-80-247-2279-5.
- [2] Information Systems and Strategy Course [online]. [cit. 2021-11-26]. Dostupné z: http://www.chris-kimble.com/Courses/World_Med_MBA/Types-of-Information-System.html
- [3] VOCHOZKA, Marek a Petr MULAČ a kolektiv *Podniková ekonomie*. Praha: Grada, 2012.
- [4] The Apache HTTP Server Project [online]. The Apache Software Foundation, ©1997-2021 [cit. 2021-11-26]. Dostupné z: <https://httpd.apache.org/>
- [5] Apache HTTP Server [online]. San Francisco: Wikimedia Foundation [cit. 2021-11-26]. Dostupné z: https://cs.wikipedia.org/wiki/Apache_HTTP_Server
- [6] Péhápko: Učebnice PHP. Péhápko.cz [online]. Michal Mencl [cit. 2021-11-26]. Dostupné z: <http://pehapko.cz/sprava-serveru/instalace>
- [7] STEJSKAL, Jan. Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. Brno: Computer Press, 2004. ISBN 80-251-0167-3.
- [8] Základy HTML, jak na HTML. Jak psát web, návod na html stránky [online]. Praha: Dušan Janovský [cit. 2021-11-26]. Dostupné z: <https://www.jakpsatweb.cz/zaklady-html.html>
- [9] <https://www.zdrojak.cz/clanky/xhtml-je-mrtve-at-zije-html5-nebo-ne/>. Zdroják - o tvorbě webových stránek a aplikací [online]. Praha: Devel.cz Lab, 2021 [cit. 2021-11-26]. Dostupné z: <https://www.zdrojak.cz/clanky/xhtml-je-mrtve-at-zije-html5-nebo-ne/>
- [10] EISENMENGER, Richard. *JavaScript: kompletní kapselní průvodce*. Praha: Grada, 1999. ISBN 80-716-9383-9.
- [11] *jQuery: write less, do more* [online]. OpenJS Foundation, 2021 [cit. 2021-11-26]. Dostupné z: <https://jquery.com/>
- [12] *JavaScript s jQuery: lehký úvod* [online]. ZONER software, a.s, 2007 [cit. 2021-11-26]. Dostupné z: <https://www.interval.cz/clanky/javascript-s-jquery-lehky-uvod/>
- [13] *Co je to CSS: kaskádové styly* [online]. ARTSTER, 2020 [cit. 2021-11-26]. Dostupné z: <https://artster.cz/co-je-to-css/>

- [14] *PHP* [online]. The PHP Group, 2021 [cit. 2021-11-26]. Dostupné z: <https://www.php.net/>
- [15] *Co je to PHP a k čemu mi bude dobrý?* [online]. CoreIT, 2021 [cit. 2021-11-26]. Dostupné z: <https://www.coreit.cz/php/>
- [16] *PHP* [online]. Praha: Jan Štráfelda [cit. 2021-11-26]. Dostupné z: <https://www.strafelda.cz/php>
- [17] Kosek, J.: *PHP – Tvorba interaktivních internetových aplikací*. Praha: Grada Publishing, 1999 ISBN 80-7169-373-1
- [18] *Eclipse PHP Development Tools* [online]. Eclipse Foundation [cit. 2021-11-27]. Dostupné z: <https://www.eclipse.org/pdt/>
- [19] *Co je to databáze MySQL?* [online]. ARTIC STUDIO, 2011 [cit. 2021-11-26]. Dostupné z: <https://www.artic-studio.net/slovnicek-pojmu/databaze-mysql/>
- [20] *CO JE TO DATABÁZE MYSQL?* [online]. BEST-HOSTING.cz, 2021 [cit. 2021-11-26]. Dostupné z: <https://best-hosting.cz/cs/napoveda/co-je-to-databaze-mysql>
- [21] *MySQL: Úvod* [online]. [cit. 2021-11-26]. Dostupné z: <https://ct.upce.cz/machalik/pxwww/wp-content/cviceni/cv12/databaze/>
- [22] *HeidiSQL: What's this?* [online]. Ansgar Becker [cit. 2021-11-27]. Dostupné z: <https://www.heidisql.com/>
- [23] *Co je to HeidiSQL?* [online]. IT-Slovník.cz team., 2021 [cit. 2021-11-27]. Dostupné z: <https://it-slovník.cz/pojem/heidisql>
- [24] *Lekce 1: Úvod do UML* [online]. itnetwork.cz, 2021 [cit. 2021-11-26]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy>

8 Přílohy

Stažení přílohy souboru zip na následujícím odkazu:

<http://www.perut.www3.cz/soubory/is.zip>

Obsahuje: Schéma databáze a úplnou adresářovou strukturu IS se zdrojovými soubory PHP, CSS a JS.