

# Webový systém pro evidenci skladu a výroby

Jaromír Polášek

---

Bakalářská práce  
2015



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2014/2015

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jaromír Polášek**

Osobní číslo: **A12598**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Forma studia: **kombinovaná**

Téma práce: **Webový systém pro evidenci skladu a výroby**

Téma anglicky: **A Web System for Warehousing and Production Records**

Zásady pro vypracování:

1. Seznamte se s problematikou organizace a plánování výroby.
2. Nastudujte PHP, framework Nette a případně další nezbytné technologie pro realizaci webového systému.
3. Vypracujte koncept webového systému a jeho funkcí.
4. Naprogramujte webový systém pro evidenci skladu a výroby pro malou firmu.
5. Vytvořte Uživatelský a Administrační manuál pro práci se systémem.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. GILMORE, W. **Velká kniha PHP 5 & MySQL: kompendium znalostí pro začátečníky i profesionály**. Vyd. 1. Brno: Zoner Press, 2005, 711 s. ISBN 808681520x.
2. CHAFFER, Jonathan. **Mistrovství v jQuery: kompletní průvodce vývojáře**. 1. vyd. Brno: Computer Press, 2013, 384 s. ISBN 978-80-251-4103-8.
3. CASTRO, Elizabeth a Bruce HYSLOP. **HTML5 a CSS3: názorný průvodce tvorbou WWW stránek**. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 9788025137338.
4. ODELL, Den. **JavaScript: průvodce programováním ajaxových aplikací**. Vyd. 1. Brno: Computer Press, 2010, 368 s. ISBN 9788025127339.
5. MAKOVEC, Jaromír. **Organizace a plánování výroby**. Vyd. 1. Praha: Vysoká škola ekonomická, Fakulta podnikohospodářská, 1993, 273 s. ISBN 8070791713.
6. **Nette Foundation: Dokumentace [online]**. 2.2. 2014 [cit. 2015-01-26]. Dostupné z:<http://doc.nette.org/en/2.2/>

Vedoucí bakalářské práce:

**doc. Ing. Martin Sysel, Ph.D.**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

**6. března 2015**

Termín odevzdání bakalářské práce:

**22. května 2015**

Ve Zlíně dne 6. března 2015



doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



L.S.



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně



.....  
podpis autora

## **ABSTRAKT**

Cílem bakalářské práce je vytvořit webový informační systém pro evidenci skladu výrobků a materiálů potřebných pro provoz malé firmy. Systém má být schopen sledovat a evidovat pohyby zásob, jejich stav z hlediska jakosti, trvanlivosti a dostupnosti. Dále musí poskytovat srozumitelný přehled o stavu zakázek, jejich průběhu a shromažďovat informace využitelné pro optimalizaci výrobního procesu. Součástí projektu bude i uživatelský a administrativní manuál s informacemi pro snadné používání systému.

Klíčová slova: Nette Framework, informační systém, výroba, sklad, evidence

## **ABSTRACT**

The aim of this bachelor thesis is to create a web information system to register the storage of products and materials needed for conducting a small businesses. The system should be able to track and record stock movements and their state in terms of quality, durability and availability. Also it has to provide a clear overview of the status of orders, their process and to collect the information usable for proces optimization. The project will include a user guide and an administrative manual containing the information for easy use of the system.

Keywords: Nette Framework, information system, production, store, register

Tímto chci poděkovat vedoucímu mé bakalářské práce panu Ing. Martinu Syslovi, Ph.D. za odborné vedení, cenné rady a konstruktivní připomínky.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 INFORMACE VE VÝROBĚ</b> .....	<b>12</b>
1.1    INFORMAČNÍ SYSTÉM .....	12
1.2    VÝZNAM INFORMAČNÍ TECHNOLOGIE V ORGANIZACI VÝROBY .....	12
1.2.1    Výrobní logistika.....	12
1.2.2    Manipulace s materiálem .....	12
1.3    SYSTÉMY ERP.....	13
1.3.1    Technologické a provozní principy ERP .....	13
1.3.2    Řízení přístupu k funkcím a informacím ERP .....	13
<b>2 TECHNOLOGIE POUŽITÉ PRO VÝVOJ</b> .....	<b>14</b>
2.1    PHP.....	14
2.2    FRAMEWORK NETTE .....	14
2.2.1    MVC aplikace a presentery .....	14
2.2.2    Životní cyklus presenteru .....	15
2.2.3    Komponenty a ovládací prvky .....	16
2.2.4    Databáze .....	16
2.2.5    Přihlašování a oprávnění uživatelů .....	16
2.2.6    Formuláře .....	17
2.3    MYSQL .....	18
2.3.1    Hierarchická data v databázi .....	18
2.4    HTML A CSS.....	19
2.5    JAVASCRIPT .....	19
2.6    JQUERY .....	19
2.6.1    jQuery user interface .....	20
2.6.2    jQuery DataTables .....	21
2.6.3    jQuery TreeViewJS .....	21
<b>PRAKTICKÁ ČÁST</b> .....	<b>22</b>
<b>3 POŽADAVKY NA WEBOVÝ SYSTÉM</b> .....	<b>23</b>
3.1    VŠEOBECNÉ POŽADAVKY .....	23
3.2    OBCHOD .....	23
3.3    VÝROBA .....	23
3.4    SKLAD.....	24
<b>4 NÁVRH WEBOVÉHO SYSTÉMU</b> .....	<b>26</b>
4.1    DATABÁZE .....	26
4.2    NÁVRH JÁDRA.....	26
4.2.1    Přístupová práva.....	26
4.2.2    Navigace.....	27
4.2.3    Operace s daty .....	27
4.3    NÁVRHY MODULŮ .....	27
4.3.1    Moduly Partners a Product.....	28
4.3.2    Modul Sklad .....	28

4.3.3	Modul Prodej.....	28
<b>5</b>	<b>JÁDRO SYSTÉMU .....</b>	<b>30</b>
5.1	DATABÁZE .....	30
5.1.1	Skupina base.....	30
5.1.2	Skupina table .....	30
5.1.3	Skupina users .....	31
5.1.4	Skupina acl .....	31
5.2	OPERACE S DATY.....	32
5.3	AUTENTIZACE A AUTORIZACE .....	32
5.4	ZÁKLADNÍ KOMPONENTY .....	33
5.4.1	Datagrid.....	33
5.4.2	Strom .....	33
5.4.3	Formulář.....	34
5.4.4	Navigace.....	34
5.5	PRESENTERY .....	34
5.5.1	BasePresenter .....	34
5.5.2	SecurePresenter .....	34
5.5.3	ComponentPresenter .....	35
5.5.4	RenderPresenter .....	35
5.5.5	ActionPresenter .....	35
5.5.6	LoginPresenter .....	36
<b>6</b>	<b>MODULY SYSTÉMU .....</b>	<b>37</b>
6.1	PARTNEŘI A VÝROBKY .....	37
6.2	SKLAD.....	38
6.3	PRODEJ.....	39
6.4	BUDOUCÍ VÝVOJ A ROZŠÍŘENÍ .....	40
	<b>ZÁVĚR .....</b>	<b>42</b>
	<b>SEZNAM POUŽITÉ LITERATURY (GENEROVANÝ) .....</b>	<b>44</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>46</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>47</b>
	<b>SEZNAM TABULEK.....</b>	<b>48</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>49</b>

## ÚVOD

Základem efektivitu podniku jsou informace a jejich dostupnost. Řada malých podniků tento fakt opomíjí a při strmém nebo nárazovém nárůstu zakázek není schopna plnit své závazky partnerům a včas dodávat své produkty i přesto, že nedošlo k naplnění nebo překročení výrobních kapacit. Řadoví i vedoucí pracovníci se rozhodují na základě neúplných nebo špatně dostupných informací a domněnek.

Trh s programy a aplikacemi dnes disponuje velkou řadou komerčních i svobodných projektů, které se zaměřují speciálně na tuto problematiku a vyvíjejí rozsáhlé a robustní systémy pro kompletní evidenci a řízení. Tyto systémy splňují všechna kritéria pro spolehlivý a výkonný informačně analytický a plánovací software. Z manažerského pohledu podávají ucelený a přesný obraz o stavu společnosti jak po ekonomické stránce, tak po stránce její efektivitu. Pro svůj provoz ovšem vyžadují odpovídající počet pracovníků, který s informacemi v systému pracuje a udržuje je v aktualizovaném stavu. Jakýkoliv systém, který neobsahuje aktuální data, je ze své podstaty, kterou je podávat informace, nepoužitelný.

Firmy a podniky, které se specializují na specifický nebo typově omezený sortiment, mají tu výhodu, že valná většina dat v informačním systému je statická. Postupy a parametry výrobků se mění postupně, ve chvílích, kdy jsou inovovány technologie nebo se změni požadavky trhu. Většina v současné době vyvíjených systémů je primárně zaměřena právě na tento druh podniků, kterých je většina a které pracují podle podobného schématu. Vývoj jednostranně zaměřeného informačního systému není tak nákladný jako univerzálního s velkým množstvím doplňků.

Společnosti zabývající se primárně zakázkovou výrobou, kde není sortiment jednoznačný nebo se může přizpůsobovat představám či požadavkům zákazníka, většinou po špatných zkušenostech s komerčními a svobodnými systémy, kdy byl informační systém spíše přítěží, neboť jeho provoz snižoval výrobní kapacity a zdroje, dávají přednost jednodušším nástrojům nebo aplikacím, které mohou v omezeném množství suplovat informační systém.

Další možností je zadat vývoj systému specializované firmě, zabývající se vývojem podobných aplikací, která disponuje týmem profesionálních programátorů a je schopná dodat řešení tzv. šité na míru, samozřejmě za této práci odpovídající cenu. Poslední možností, která ale není vždy dostupná, je snížit požadavky a vytvořit jednoduchý systém pokrývající základní potřeby provozu svépomocí. Což je úkolem této práce.

Při návrhu tohoto systému je nutné klást důraz na jednoduchost a modularitu nebo možnost rozšíření systému, aby byl schopen pružně reagovat na ekonomické a provozní změny bez složitých zásahů do vlastního kódu.

## **I. TEORETICKÁ ČÁST**

## 1 INFORMACE VE VÝROBĚ

V současné době jsou podstatnou součástí výroby informace. Zajištění rozvoje výrobních postupů a zvyšování efektivity výroby vyžaduje vytváření vhodných předpokladů, mezi kterými mají velmi výrazné postavení předpoklady informační. Využívání rozsáhlých a aktualizovaných databází je cesta ke zvyšování kvality výroby. [1]

### 1.1 Informační systém

*Účelem informačního systému je zajištění správných informací na správném místě, ve správný čas [2 str. 15]*

Informační systémy mohou být v širším rámci vnímány jako informace zapsané a zpracované nejčastěji relačními databázemi, které omezují účast člověka pomocí automatizace některých činností a podpoře jeho rozhodování, dále uložené na klasických nosičích a také informace dosud nezaznamenané. V posledním případě se může jednat o informace v paměti zaměstnanců. [3]

### 1.2 Význam informační technologie v organizaci výroby

Současné výrobní technologie jsou z velké části závislé na informacích. Mimořádný význam informační technologie spočívá ve schopnosti dodat informaci od místa vzniku do místa využití poznatků. Úroveň výroby závisí na dostupnosti informací a pružné komunikaci. [1]

#### 1.2.1 Výrobní logistika

Předmětem logistiky je studium materiálových toků, před nastavující pohyb materiálu od dodavatelů přes výrobní podniky až k odběratelům. Jejím obsahem je organizace, plánování, řízení a kontrola všech činností souvisejících s materiálovými toky. Zabezpečení těchto činností je možné pouze na základě získání, zpracování a přenosu informací o těchto činnostech, které svým pohybem mezi jednotlivými subjekty tohoto procesu vytváří informační toky. [1]

#### 1.2.2 Manipulace s materiálem

Manipulace s materiálem se skládá z jednotlivých manipulačních operací. Spojením několika operací vznikne dílčí proces, který zajišťuje určitou část výrobního procesu. Tyto ma-

nipulace lze rozdělit: vnější a mezi objektová doprava, vnitropodniková manipulace, skladové hospodářství a obalové hospodářství [1]

### 1.3 Systémy ERP

ERP je aplikační software, který umožňuje koordinaci podnikových zdrojů a aktivit. Je zdrojem informací o produktech a materiálech. Integruje většinu podnikových procesů a umožňuje sdílet společná data v rámci celého podniku. Může být ale chápán i jako parametrizovaný software, který podniku umožňuje spravovat jeho hlavní procesy. [3]

V závislosti na svém provedení a pozici v informačním systému je ERP zdrojem dat i pro ostatní typy aplikací. Obsahuje základní typy produktů pro prezentaci na www stránkách, databázi zákazníků pro CRM aplikace a číselníky prakticky pro všechny typy aplikací. [4]

Přední dodavatelé ERP systémů vyvíjejí komplikované softwarové balíky, které pokrývají potřeby podniků z mnoha oborů. Přestože se jedná o přednost, přináší sebou řadu problémů a to obtížná upravitelnost na konkrétní podnik, vysoké náklady na školení, nízké využití celkové funkcionality a v neposlední řadě vysoké náklady na počítačové zdroje. Jejich integrita navíc způsobuje vysokou složitost i v případě jednoduché operace. Vhodnou cestou zřejmě bude specializace podle hospodářských sektorů. [2]

#### 1.3.1 Technologické a provozní principy ERP

Systémy ERP sdílejí společná data ať už vlastní nebo externí. Důsledkem toho jsou následující principy:

- Transakce v jednom modulu může generovat akci v modulu jiném.
- Transakce jsou navzájem konzistentní.
- Lze zpětně dohledat následky a příčiny jednotlivých transakcí. [4]

#### 1.3.2 Řízení přístupu k funkcím a informacím ERP

ERP musí být schopen zajistit bezpečný přístup k informacím všem oprávněným uživatelům, kteří mají různé potřeby, co se týče úrovně detailů a způsobu jejich vizualizace. Dále mají uživatelé různá oprávnění pro manipulaci s daty. Liší se i v tom, kdo může určitá data pouze číst a kdo má právo je upravovat. V neposlední řadě je také efektivita práce s takovým systémem. Tedy ERP musí být postaven na efektivním uživatelském rozhraní. [4]

## 2 TECHNOLOGIE POUŽITÉ PRO VÝVOJ

Pro vývoj webu je nezbytné nastudovat a využít některé technologie.

### 2.1 PHP

*PHP je rekurzivní zkratka anglického názvu „PHP: Hypertext Preprocesor,“ jež označuje skriptovací jazyk otevřeným zdrojovým kódem, který byl navržen pro tvorbu interaktivních webových stránek [5 str. 315]*

PHP5 je objektově orientovaný skriptovací jazyk, provádějící se na straně serveru. Od verze 5 prodělal velkou změnu v podobě přepracování objektového modelu a značně se vylepšila podpora objektově orientovaných metod a technik. [6]

Od verze 5.1 navíc disponuje abstraktní vrstvou PDO (PHP Data Object), napsanou v jazyce C a zakomponovanou do PHP, která potřebný ovladač načítá až při běhu, což mu dává jistou flexibilitu při změně databáze. [7]

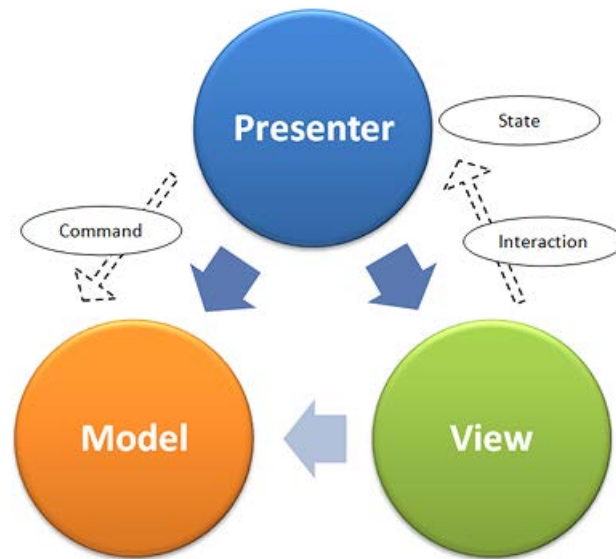
### 2.2 Framework Nette

Nette Framework (dále jen Nette) je open source framework pro tvorbu webových aplikací ve skriptovacím programovacím jazyku PHP 5, mezi jehož přednosti patří propracované nástroje jako systém šablon, ladící nástroje, efektivní databázová vrstva, zabezpečení před zranitelnostmi, podpora HTML5, AJAX nebo SEO a v neposlední řadě dokumentace v českém jazyce. [8]

#### 2.2.1 MVC aplikace a presentery

Architektura MVC se dělí na 3 části Model, View a Controller. Základním smyslem je, aby každá část byla upravovatelná samostatně. Model představuje datovou strukturu, View uživatelské rozhraní a Controller obsahuje aplikační logiku. [9]

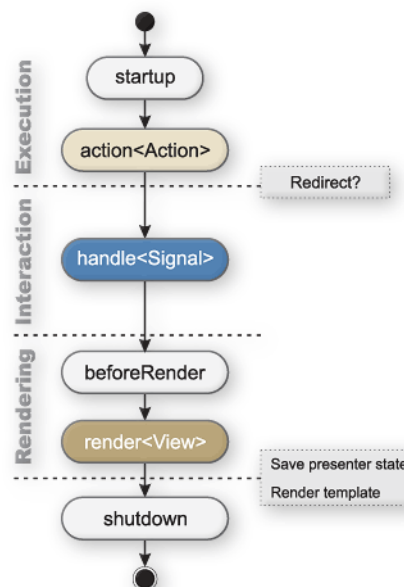
Nette využívá architekturu MVP. Model zde zastupuje celou vrstvu datového základu a aplikační logiky. Spravuje si vnitřní stav a navenek se prezentuje jako pevně dané rozhraní. Voláním metod modelu se aktualizuje nebo mění jeho stav. View je vrstva, která zobrazuje výstup. Controller se v Nette nazývá Presenter. Jedná se o řadič, který zpracovává požadavky uživatele a volá Model. Následně zavolá View, který zobrazí výstup. U složitějších aplikací lze prvky rozčlenit do podadresářů, nazvaných moduly, které mohou obsahovat další sub moduly. [8]



Obr. 1: Architektura MVP v Nette Framework [10]

### 2.2.2 Životní cyklus presenteru

Akce presenteru má za úkol, kromě volání metody *render<akce>*, také provedení dalších metod.



Obr. 2: Životní cyklus presenteru [8]

Ihned po vytvoření se zavolá metoda *startup()*, která může inicializovat proměnné nebo ověřit oprávnění. Následně se volá *action<Action>()*. Tato metoda se provádí před *meto-*

*render<View>()* a slouží k provedení akce, která nic nevykreslí. Například přesměrování. Metoda pro zpracování signálů nebo subrequestů zejména pro komponenty se nazývá *handle<Signál>()*. Přímě před renderem se volá *beforeRender()* a po renderu je zavolán *shutdown()*, který ukončí životní cyklus presenteru. [8]

### 2.2.3 Komponenty a ovládací prvky

Komponenta v Nette je renderovatelný objekt, kterých v rámci jedné aplikace může existovat neomezený počet. Jedná se o implementaci znovu použitelnosti kódu. Obsahuje továrnu na šablonu a lze ji renderovat. Každá komponenta má dva parametry konstruktoru. Jméno a rodiče. Vytvořením komponenty pod jiným jménem lze komponentu na stránce zobrazit několikrát. Rodič komponenty může být jakýkoliv jiný objekt, který implementuje rozhraní *Nette\ComponentModel\IContainer*. [8]

### 2.2.4 Databáze

Databázová vrstva Nette je zastoupena Třídou *Nette\Database\Connection*.

Ta vytváří objekt na PDO a reprezentuje připojení k databázi. Vrstva si sama vytváří vlastní ovladač pro konkrétní databázový server. Plná podpora je pro databáze MySQL, PostgreSQL, Sqlite3 a MS SQL (PDO\_SQLSRV). Pro Oracle, Sqlite2, ODBC a MS SQL (PDO\_DBLIB) je podpora částečná, bez Třídy *Nette\Database\Table*. Všechna připojení jsou vytvářena jako „lazy“ (ke skutečnému navázání připojení dojde, až když je databáze vyžadována). [8]

### 2.2.5 Přihlašování a oprávnění uživatelů

Autentizací se rozumí ověření identity uživatele. V Nette je zajištěno službou *user*. Tato služba obsahuje metody *login()* pro přihlášení uživatele, *isLoggedIn()* pro ověření, zda je uživatel přihlášen a *logout()* pro jeho odhlášení. Dále jsou zde pomocné metody *getLogoutReason()*, která vrací způsob odhlášení (neaktivita, zavření prohlížeče, manuální odhlášení) a *setExpiration()* nastavující pravidla pro automatické odhlášení. [8]

Pro vlastní přihlášení je nutné vytvořit objekt s metodou *authenticate()*, který provede vlastní přihlášení a v případě neúspěchu vrací *Nette\Security\AuthenticationException*. V opačném případě vrací objekt *Nette\Security\Identity*. [11]

Autorizace ověřuje, zda má přihlášený uživatel práva k požadované akci. Předpokladem autorizace je úspěšná autentizace. O možnostech uživatele rozhoduje členství v přiděle-

ných rolích, jejichž smyslem je nezávislost práv na konkrétním uživateli a jejich dědičnost. Práva se ověřují pro zdroj, kterým je myšlen konkrétní prvek webu a operaci, kterou s prvkem uživatel zamýšlí provést. [8]

Služba *user* vyžaduje registraci autorizačního handleru pomocí metody *setAuthorizationHandler()*. Tato operace zpřístupní ve službě *user* metodu *isAllowed()*, ta provádí vlastní ověření práv na základě parametrů, kterými jsou zdroj a požadovaná operace. [11]

### 2.2.6 Formuláře

Nette klade velký důraz na bezpečnost a transparentnost formulářů. Mimo jiné chrání aplikaci před Cross Site Scripting (XSS) a Cross-Site Request Forgery (CSRF), ověřuje Validitu UTF-8 kódování a filtruje kontrolní znaky. Zajišťuje validaci odesílaných dat na straně serveru i klienta a minimalizuje možnost vzniku chyb. Obsahuje většinu standartních formulářových prvků s předdefinovanými validačními pravidly, které jsou automaticky generovány jak na straně serveru, tak klienta a umožňuje vytvářet vlastní prvky nebo jen validační pravidla. [8]

Vykreslené formuláře splňují základní pravidla přístupnosti. Všechny popisky jsou označeny tagem `<label>` a prvkem formuláře. Jsou opraveny některé chyby prohlížečů a podobu výstupního HTML kódu lze kompletně změnit. Dotazem na metodu *isSubmitted()* lze ověřit, zda byl formulář již odeslán nebo se jedná o jeho vykreslení uživateli. V případě, že validace zjistí nesprávné vyplnění, je formulář neustále předkládán s výpisem chyb až do té doby, než je vyplněn správně. [10]

Obr. 3: Struktura a popis prvků formuláře  
Nette [8]

## 2.3 MySQL

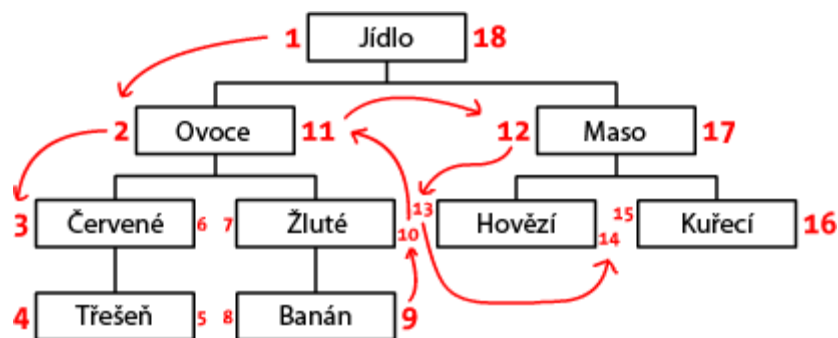
MySQL je relační databázový server disponující třemi mechanismy pro správu dat, (HEAP, InnoDB a MyISAM) fulltextovým indexováním a vyhledáváním, a od verze 5 kompletní podporu cizích klíčů. Pro komunikaci s ním se využívá jazyka SQL. [7]

V současné době MySQL vlastní společnost Oracle. Obchodní politika této firmy vzbuzuje obavy, zda bude databáze i nadále dostupná pod svobodnou licenci. Na základě těchto obav vznikl nový projekt u společnosti The Monty Program s názvem MariaDB, který vychází z MySQL a snaží se pokračovat v jejím duchu. Prvním z velkých projektů, který přešel z MySQL na MariaDB je Wikipedie. [12]

### 2.3.1 Hierarchická data v databázi

Přestože jsou k dispozici v některých ohledech pokročilé objektové databáze, zůstávají nejpoužívanějším typem relační databázové systémy. Jednou z jejich nevýhod je nedostatečná podpora dat s hierarchickou, neboli stromovou, strukturou. Proto je nutné tato data transformovat do takové podoby, v jaké s nimi dokáže pracovat aplikace. Jednou z efektivních metod je Modified Preorder Tree Traversal Algoritmus (MPTT). Tato metoda využívá ohodnocení uzlů stromu dvěma hodnotami [13]

Princip metody spočívá v přiřazení čísel označených jako levé a pravé. Kořenovému uzlu se přiřadí levá hodnota 1. Jeho prvnímu potomku pak hodnota 2. Tímto způsobem strom v podstatě obejdeme. Nejlépe celý postup vystihuje následující obrázek: [14]



Obr. 4: Princip číselného ohodnocování stromu [14]

## 2.4 HTML a CSS

HTML (Hypertext Markup Language) je značkovací jazyk, který je základním stavebním kamenem webových stránek. Jeho základem je dokument složený z elementů. HTML určuje jeho strukturu. Naproti tomu CSS (Cascading Style Sheet) specifikuje vzhled dokumentu. Jejich vývojem se v současné době zabývá společenství W3C (World Wide Web Consortium). HTML verze 5 vyvíjí skupina WHATWG. W3C pouze zapojuje jejich výsledky do své pracovní verze [5]

## 2.5 JavaScript

JavaScript je jazyk pro tvorbu RIA aplikací. Umožňuje přidávat statickým webovým stránkám dynamické prvky. Jazyk je interpretován webovým prohlížečem a vkládá se přímo do HTML dokumentu. Disponuje mimo jiné dynamickou změnou vlastností jazyka CSS a možností úpravy elementů nebo jejich obsahu. Jazyk nebyl navržen jako objektově orientovaný, podporuje však některé principy objektového návrhu. [15]

## 2.6 jQuery

jQuery je knihovna funkcí v jazyce JavaScript, která se snaží maximálně využívat možnosti tohoto jazyka a zároveň zjednodušit vytváření interaktivních stránek. K využívání je nutné znát alespoň základy jazyka JavaScript, protože jQuery používá stejnou syntaxi. [16]

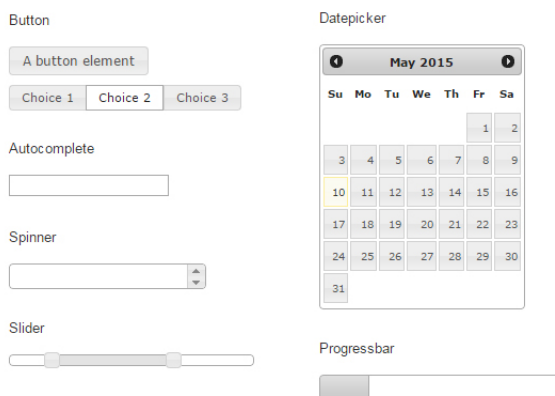
Knihovna jQuery poskytuje víceúčelovou abstraktní vrstvu pro vytváření skriptů na webu. Lze ji tedy použít při řešení projektů, které vyžadují skriptování na straně klienta. Její základní funkce slouží k provádění následujících operací:

- Přístup k elementům dokumentu.
- Úprava vzhledu webové stránky pomocí CSS stylů.
- Měnit obsah dokumentu.
- Reagovat na akce uživatele.
- Animovat změny v dokumentu.
- Načítat data ze serveru bez nutnosti obnovení stránky (Ajax).
- Zjednodušuje běžné činnosti při programování v jazyce JavaScript

jQuery umožňuje více akcí na jednom řádku a aplikuje na většinu svých metod techniku řetězení, čímž volaná metoda objektu vrací samotný objekt, se kterým se dá dále pracovat. Jednou z dalších výhod je snadná rozšiřitelnost pomocí zásuvných modulů [17]

### 2.6.1 jQuery user interface

jQuery user interfaces (jQueryUI) je balíček interakcí, efektů, kontrolních prvků a témat využívající funkcí knihovny jQuery. Jednou z jeho předností je možnost nadefinovat které prvky budeme potřebovat, upravit jejich vzhled v ThemeRolleru, stáhnout jej a následně nalinkovat do vlastního webu. Vygenerovaný vzhled má jedinečný identifikátor, takže při změně jednoho prvku nemusíme celý balíček vytvářet znovu. [18]



Obr. 5: Některé prvky jQuery UI [18]

## 2.6.2 jQuery DataTables

Show  entries Search:

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	\$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	\$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011/12/12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011/12/06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012/03/29	\$433,060

Showing 1 to 10 of 57 entries Previous  2 3 4 5 6 Next

Obr. 6 Datatables - Zásuvný modul pro jQuery [19]

DataTables je zásuvný modul pro knihovnu jQuery v jazyce JavaScript. Jedná se o vysoce flexibilní nástroj založený na metodě progressive enhancement, což znamená, že rozšiřuje možnosti stránky, ale samotná stránka může fungovat i bez tohoto rozšíření. Účelem tohoto modulu je rozšířit možnosti HTML tabulky o interaktivní funkce jako je filtrování, stránkování a třídění dat. Má implementovanou podporu pro překlad, umí zpracovat data z různých zdrojů (DOM, JavaScriptu, Ajax), a její vzhled lze upravovat pomocí CSS stylů. [19]

## 2.6.3 jQuery TreeViewJS

TreeviewJS je jednoduchý a snadno upravitelný zásuvný modul pro základní manipulaci se stromovými daty. Jeho předností je snadná upravitelnost a jednoduchý kód. Stejně jako předchozí zásuvný modul, pouze usnadňuje práci s daty, v tomto případě ve formě stromu. [20]



Obr. 7 TreeViewJS – Zásuvný modul pro jQuery [20]

## **II. PRAKTICKÁ ČÁST**

### 3 POŽADAVKY NA WEBOVÝ SYSTÉM

Návrhu webového systému pro evidenci skladu a výroby musí předcházet shrnutí požadavků, jak by měl takový systém vypadat, jak se chovat a co obsahovat.

#### 3.1 Všeobecné požadavky

Systém musí pokrýt potřeby ale i schopnosti celé hierarchie podniku. Při návrhu je nutné zohledňovat, že systém má v první řadě nahradit dosavadní komunikační principy v podniku, kterými jsou v převážné většině tužka a papír. V některých případech e-mail nebo telefon. Tyto dosavadní technologie ve většině případů pouze přesouvají problém z jedné osoby na jinou. Například, pokud se obchodní zástupce telefonicky zeptá, zda je zakázka již připravena, dotazovaný pracovník musí zakázku fyzicky vyhledat a ověřit její stav. Buď verbálním dotazem na pracovníka, který zakázku zhotovuje, kontrolou skladu hotových zakázek nebo nahlédnutím do průvodní dokumentace k zakázce, pokud ji má v dané chvíli k dispozici. Základem systému by v první řadě měla být schopnost elektronicky uchovávat tyto informace, popř. průvodní dokumentaci a umožnit k ní přístup ze všech k tomuto účelu určených počítačů bez nutnosti instalace jakéhokoliv dodatečného software.

#### 3.2 Obchod

Pracovníci obchodu a obchodní zástupci musí být obeznámeni se stavem jejich zakázek, aby byli případně schopni informovat klienta o prodlevách, nebo inicializovat vyzvednutí, popř. doručení. Systém musí být schopen evidovat tuto informaci jako stav zakázky a udržovat ji jako klíčovou pro všechny části podniku.

Dále musejí být schopni efektivně zadávat jak standartní, tak speciální zakázky a mít dostatečný prostor pro vyjádření požadavků klienta. Je nutné vycházet z předpokladu, že provoz, pro který je systém určen, se zabývá zakázkovou výrobou, kde u jednotlivých zakázek není mnoho společných parametrů.

#### 3.3 Výroba

Primárním úkolem pracovníka ve výrobě je zhotovovat zakázky. Tato činnost vyžaduje přesné a jednoznačné instrukce. Pracovník výroby musí být obeznámen s aktuálními i budoucími úkoly a mezními termíny jejich dokončení, aby byl schopen efektivně rozvrhnout vymezený čas i prostředky. K tomu mu má dopomoci nadřízený pracovník, který ze své

pozice musí být schopen upravit pořadí zakázek tak, aby odpovídaly aktuálním potřebám a dostupným zdrojům. Naproti tomu nemůže trávit velkou část této doby získáváním informací z nevhodně navrženého systému. Měl by mít právo označit svůj úkol za dokončený, čímž se zakázka automaticky přesune do nového stavu.

### 3.4 Sklad

System musí být schopen evidovat veškeré pohyby zásob a aktuální stav skladů. Pověřený pracovník musí mít s jeho pomocí přehled nejen o skutečném stavu, ale i o pohybu budoucím, nebo naopak musí být schopen dohledat historii každého výrobku nebo materiálu od jeho zdroje k cíli. Nezbytné jsou i funkce pro naskladnění a vyskladnění materiálu, popř. výrobků. Pro systém bude jak materiál, tak výrobek jednotně označen jako produkt, protože každý materiál je produktem nějaké práce, i když se nachází mimo sledovaný provoz. Některé výrobky se mohou opět stát materiálem, protože slouží jako polotovary pro další výrobu a jsou vyráběny do zásoby, nejčastěji při menším vytížení provozu. Cílem těchto produktů tedy nebude expedice, ale opět sklad materiálu.

Tok materiálů a výrobků v systému musí jednoznačně určovat, kde byl produkt pořízen, popřípadě jaké jiné produkty byly spotřebovány na jeho vytvoření. K tomuto účelu je nezbytné vytvořit logickou strukturu fyzických a virtuálních skladů. Fyzickými sklady jsou myšleny úložiště, ve kterých se sklad v provozu nachází. Virtuálním skladem může být zdrojové nebo cílové umístění produktu. Tato struktura musí být obrazem provozu, který má představovat.

Při nákupu materiálu bude v systému vytvořen pohyb produktu. Nakoupený materiál bude navýšen o požadované množství ve virtuálním skladu nazvaném *dodavatel* a následně stejným způsobem přičten ve fyzickém skladu. Tuto operaci můžeme nazvat naskladnění. Množství ve virtuálním skladu *dodavatel* a skladu fyzickém, se o toto množství navýší. Karta výrobku v tuto chvíli musí ukazovat aktuální množství ve fyzickém skladě. Virtuální sklad *dodavatel* si uchovává informaci o pořízeném množství.

Jedná-li se o materiál, u kterého vznikl požadavek na spotřebu za účelem vytvoření nového produktu, bude stejným způsobem jako v předchozím odstavci, dalším pohybem převeden do virtuálního skladu nazvaného *spotřeba*. Tento virtuální sklad bude tedy v reálném čase zobrazovat veškerý spotřebovaný materiál. Fyzický sklad, který ukazuje skutečné množství, se při této operaci musí o odebrané množství snížit. Ekvivalentně k tomuto skladu

bude existovat další virtuální sklad nazvaný *výroba*. Tento sklad bude zdrojem všech vyrobených produktů v provozu. Vyroběním produktu je myšleno jeho přesunutí buď do skladu fyzického, nebo virtuálního. Fyzické sklady evidují vždy aktuální množství, které se ve skladu nachází. Oproti tomu virtuální sklady evidují množství, které sklady prochází a každé další přičítají.

Zvláštní postavení v tomto řetězci mají virtuální sklady nazvané *Zmetky* a *Inventura*. Virtuální sklad *zmetky* slouží k vyřazení produktu ze skladu jeho znehodnocením. Oproti tomu *Inventura* je virtuální sklad, pomocí kterého můžeme dorovnat sklad fyzický, tak aby odpovídal jeho skutečnému stavu. Virtuální Sklad *Inventura* poté zobrazuje zjištěný rozdíl.

## 4 NÁVRH WEBOVÉHO SYSTÉMU

Celkový koncept systému je modulární. Nette nám v tomto případě vychází vstříc, protože podpora modulů a odpovídající adresářové struktury jsou jeho součástí. Hlavní částí systému bude jádro, které zajišťuje navigaci, autorizaci, autentizaci a manipulaci s daty. Automatizované a pro různé oblasti specifické operace zajistí konkrétní moduly. Tento koncept umožní budoucí rozvoj systému a přidávání funkcí s minimálními zásahy do ostatních komponent.

### 4.1 Databáze

Jako databázová vrstva byla zvolena relační databáze MySQL. Z důvodů její cenové dostupnosti a snahy o sjednocení různých databází, které jsou v podniku aktuálně nasazeny (současný systém OpenERP běží na databázi PostgreSQL, ale systém evidence docházky na MySQL). Jako minimální verze byla stanovena verze 5.6, pro podporu více sloupců typu timestep (časový otisk) v jedné tabulce, které jsou nezbytné pro správné fungování tříd pro sledování zásahů do databáze.

### 4.2 Návrh Jádra

Základní operace s daty nejen v relačních databázích jsou číst, vkládat, upravovat, a mazat, což odpovídá příkazům jazyka SQL SELECT, INSERT, UPDATE a DELETE. Všechny tyto operace bude zajišťovat jádro systému na základě autorizace dle uživatelských rolí. Systém bude obsahovat komponenty pro zobrazení a upravování dat, které jim bude předávat pro tuto činnost navržená služba.

#### 4.2.1 Přístupová práva

Každému uživateli bude přidělena role. Přístup bude řízen jak na úrovni položek menu, tak na jednotlivých položkách každého formuláře odpovídajícího strukturou datům v databázi. Na úrovni menu budou pouze dva stavy: zobrazit/skrýt. Naproti tomu každá položka formuláře bude mít 4 možnosti přístupu: číst, vytvořit, upravit a smazat. Pokud nebude nastaven přístup pro čtení, nebude se uživateli daná položka zobrazovat. Pokud bude mít práva pro čtení, ale ne pro zápis, nebo úpravy, zobrazí se mu položka jako needitovatelný text. K tomuto účelu má Nette předpřipravenou knihovnou *Nette\Security\Permission*, která je navržena k tomuto účelu.

## 4.2.2 Navigace

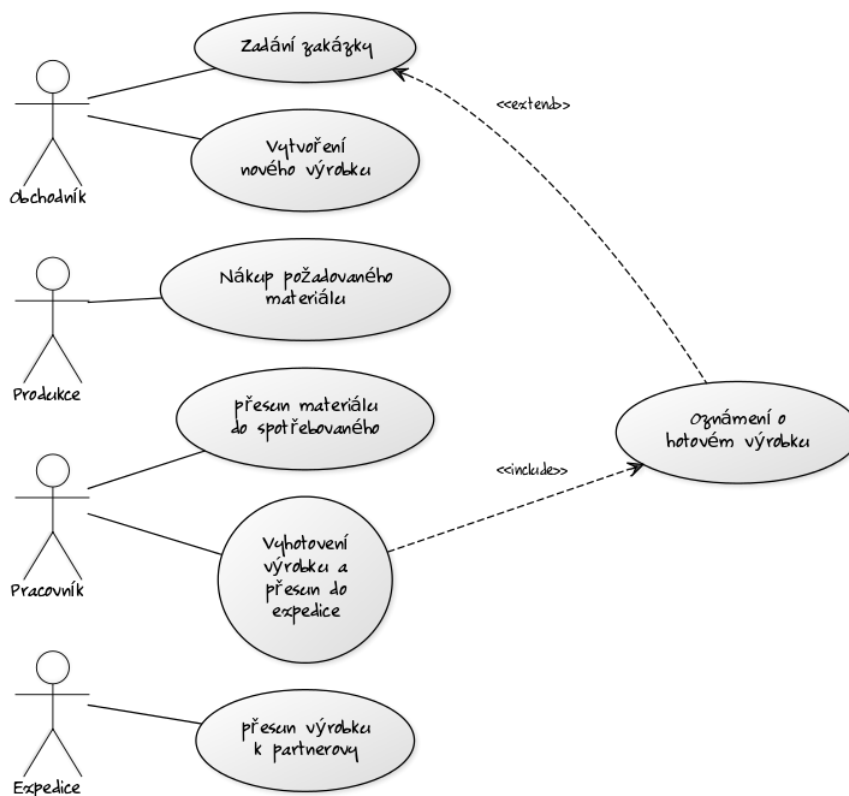
Navigaci bude zajišťovat samostatná komponenta. Položky bude získávat z databáze a o její implementaci se postará některý ze zděděných presenterů.

## 4.2.3 Operace s daty

Přístup k datům bude probíhat přes objekt, který bude data zpracovávat a obsluhovat relace. K tomuto objektu budou přistupovat komponenty, jejich vlastnost opakovaného vykreslení do stránky umožní vytvářet složitější struktury. Např. seznam rolí ve formuláři uživatele, nebo položky objednávky.

## 4.3 Návrhy modulů

Moduly budou provádět konkrétní operace, pro které budou navrženy. Komunikace mezi nimi bude řešena buď formou přesměrování na jiný modul s možností uchování tzv. backlinku, nebo přes databázi. Budou moci také zapisovat do nevlastních tabulek, pokud uživatel bude disponovat patřičnými oprávněními. Základní rozdělení na jednotlivé moduly bude vycházet z uživatelského pohledu na jednotlivé procesy.



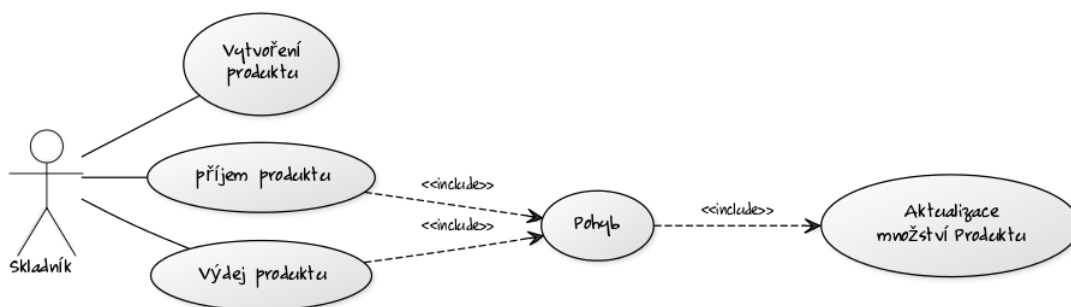
Obr. 8: Use case Diagram procesu výroby

### 4.3.1 Moduly Partners a Product

Moduly budou jedny ze základních částí potřebných pro práci ostatních modulů. *Partner* bude evidovat databázi všech obchodních partnerů. Data budou mít stromový formát, aby bylo možné pracovat i s hierarchickou strukturou partnerů. V podstatě se jedná o přiřazení kontaktů k firmě. Naproti tomu modul *Product* bude uchovávat informace o všech materiálech a produktech a také dodavatelích za pomoci modulu *Partner*.

### 4.3.2 Modul Sklad

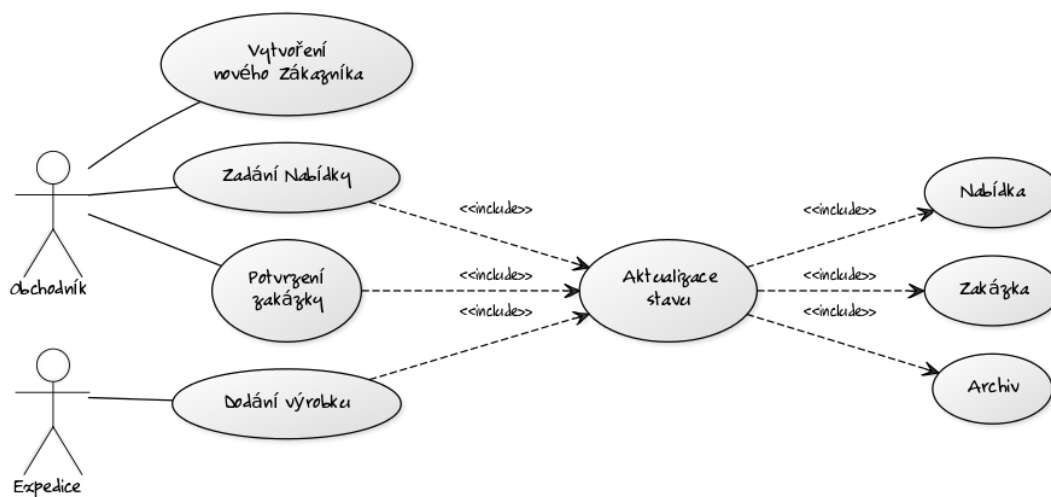
Modul bude evidovat všechny pohyby materiálu a vyhodnocovat jejich stavy. Bude přijímat a vydávat zboží. Sklady budou rozděleny na fyzické a virtuální. Struktura skladů bude hierarchická a bude ji možné upravit podle aktuálního stavu a požadavků. Mezi další funkce bude patřit možnost inventury, odpisů zásob.



Obr. 9: Use case Diagram modulu sklad

### 4.3.3 Modul Prodej

Modul bude evidovat objednávky v systému. Objednávka bude obsahovat základní informace o klientovy pomocí modulu *Partner* a také informaci o svém aktuálním stavu. Jednotlivé položky budou zaznamenány jako řádky objednávky s odkazy na moduly *Product* pro určení výrobku a *Stock* pro interní sledování zásilky.



Obr. 10: Use case diagram modulu prodej

## 5 JÁDRO SYSTÉMU

Tato kapitola se věnuje implementaci jádra a jeho propojení s databází.

### 5.1 Databáze

Tabulky jádra tvoří čtyři logické skupiny.

#### 5.1.1 Skupina base

Navigace v systému, předávání parametrů

*base\_source* - Tabulka výchozích akcí systému. Hlavní rozcestník pro další operace. Určuje který presenter bude pracovat se kterou tabulkou. Slouží k vytváření navigace v menu, generování rout a vytvoření formuláře. Všechny potřebné prvky buď obsahuje, nebo na ně odkazuje

*base\_menu* - Strom navigace. Obsahuje názvy položek v menu a odkazy na řádky v tabulce *base\_source*, kde se nacházejí informace o akci, která se má provést.

*base\_presenter* - Seznam všech dostupných presenterů, které jsou cílem akcí uložených v tabulce *base\_source*.

*base\_module* - Seznam všech modulů systému.

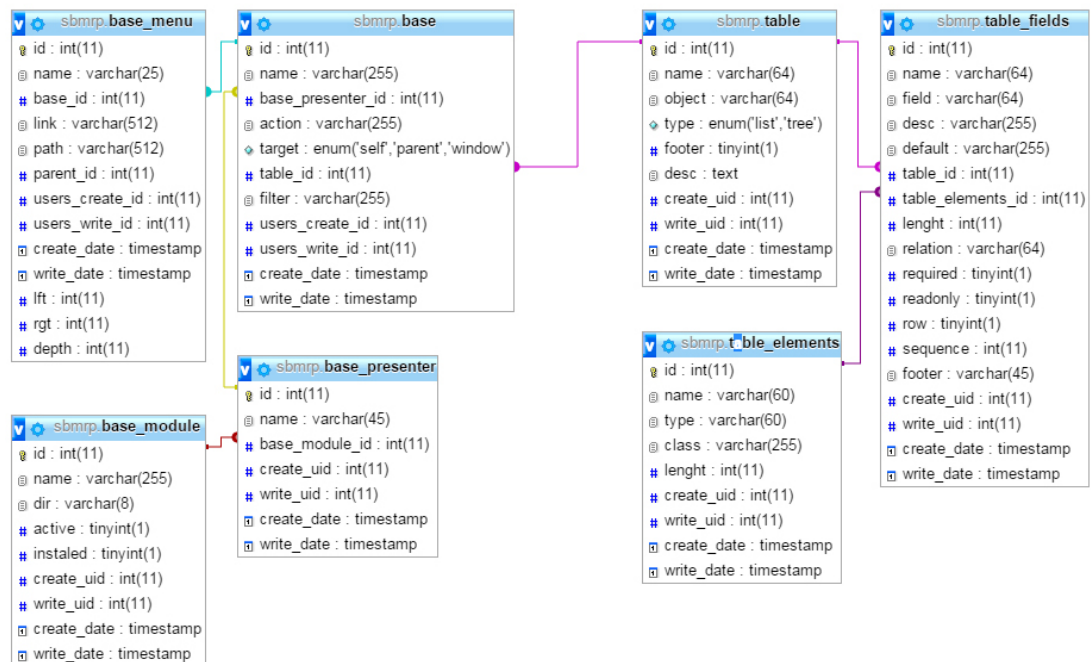
#### 5.1.2 Skupina table

Struktura dat v databázi

*table* - seznam všech tabulek obsahující data, se kterými se bude pracovat. *base\_source*

*table\_fields* - seznam všech polí, se kterými se bude v dané tabulce pracovat. Slouží ke generování formuláře, popisu sloupců tabulek.

*table\_elements* – seznam všech dostupných prvků formuláře, potřebných pro editaci polí

Obr. 11: Struktura tabulek skupin *base* a *table*.

### 5.1.3 Skupina users

Uživatelé, jejich role a předvolby.

*user* - Seznam uživatelů.

*user\_roles* - Seznam rolí uživatelů.

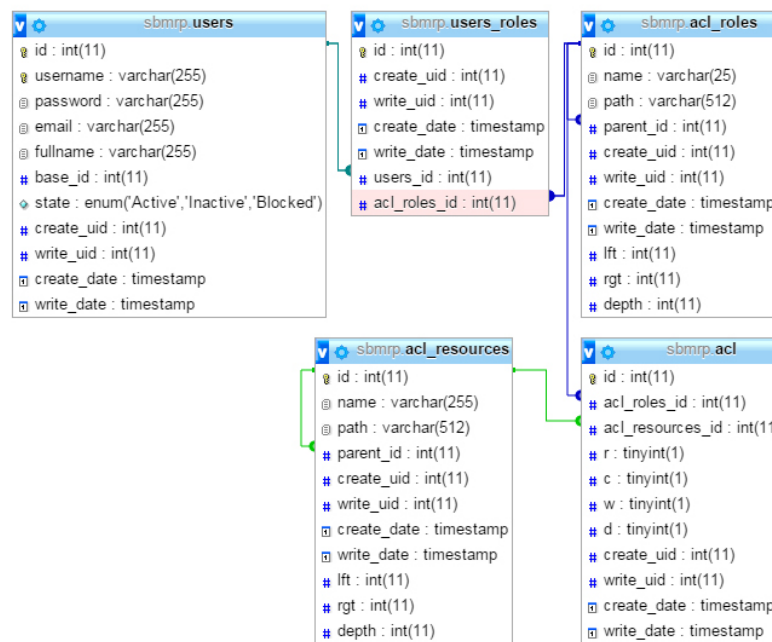
### 5.1.4 Skupina acl

Řízení přístupu ke zdrojům.

*acl* - seznam oprávnění ke čtení, vkládání, upravování a mazání záznamů v databázi (r, c, w, d).

*acl\_resources* - strom všech zdrojů v systému, ke kterým lze přistupovat.

*acl\_roles* - strom uživatelských rolí.

Obr. 12: Struktura tabulek skupin *acl* a *users*.

## 5.2 Operace s daty

Přístup k datům zabezpečuje služba *Action*. Tato služba je navržena tak, aby na základě přístupových práv předala komponentě, nebo presenteru požadovaná data. Má přístup i k informacím o jejich struktuře, které jsou nutné pro správné zobrazení a pojmenování.

Struktura je rozdělena na *seznamy* a *stromy*. Data v seznamech tvoří lineární řádkové záznamy. Získávají se vytvořením objektu *Table*, který obsahuje metody umožňující základní operace s daty. Stromy mají specifický způsob manipulace s daty. Využívají principů MPTT. Data se získávají vytvořením objektu *Tree*. Oba objekty implementují stejný interface *IData*, který zajišťuje stejné metody pro základní operace.

Údržbu a kontrolu struktury dat, popřípadě jejich opravu si objekty provádějí samostatně pomocí rozšiřujících knihoven.

## 5.3 Autentizace a autorizace

Přihlašování uživatelů do systému zabezpečuje služba *Authenticator*. Její implementace je jednodušší, než dokumentace uvádí, v podstatě ji stačí zaregistrovat do konfiguračního souboru *config.neon*, nebo jeho lokálních klonů. Služba si sama vytváří objekty třídy *table* se seznamy *users* a *users\_roles*. Následně ověří platnost zadaných přihlašovacích údajů a přiřadí role a další nezbytné informace o právě přihlášeném uživateli.

O autorizaci se stará služba *Autorizator* (taktéž zaregistrovaná v konfiguračním souboru), která je zastoupena objektem *Acl*. Tento objekt obsahuje metodu *isAllowed()*, která ověří zadaná práva k požadovanému zdroji. Objekt *Acl* vytváří instance tříd *Resources*, *Roles* a *Permission*, které jsou vlastní implementací třídy *Nette\Security\Permission* obohacené o načítání dat z databáze a upravenou logikou.

Úprava logiky spočívala hlavně v dědičnosti zdrojů, které v implementaci Nette provádějí totéž jako role, čímž například v menu neumožní zobrazení položek, ke kterým má uživatel přístup, pokud nemá přístup do položky nadřazené a naopak. Pokud má přístup do položky nadřazené má přístup i do všech položek jí podřízených. Tato logika je pro správné fungování systému nepřijatelná. Nově vytvořená a implementovaná logika přiděluje práva postupně od nejnižších a nadřazeným rolím se dědí práva automaticky. Má-li uživatel přístup třeba jen k jedinému údaji v tabulce, musí mít systém k tabulce přístup. Bezpečnostní politika v tomto případě říká, že nemá-li uživatel ke zdroji přístup, nemá k nim přístup ani modul, nebo právě používaná komponenta. Služba *Acl* při neautorizovaném pokusu o přístup vyhazuje výjimku *Nette\Application\ForbiddenRequestException* a systém zobrazí kód 403.

## 5.4 Základní komponenty

### 5.4.1 Datagrid

O vykreslení obsahu tabulky se stará komponenta *TableControl*. Tato komponenta vytváří tzv. datagrid. Jedná se o jednoduchou tabulku s popisy sloupců v záhlaví a popřípadě zápatí, pokud je definováno. Kliknutím na požadovaný záznam je uživatel přesměrován na editační formulář. O ovládání na straně klienta se stará zásuvný modul pro jQuery nazvaný *DataTables*. Obohacuje tabulku o ovládací prvky jako je třídění dat nebo filtrování. Komponenta získává data pomocí služby *Action*.

### 5.4.2 Strom

Komponenta *TreeControl* zobrazuje záznamy se stromovou strukturou jako rozbalovací seznam, Jedná se o nástroj pro přehlednější zobrazení dat. Kliknutím na název uzlu je uživatel přesměrován na editační formulář. Na straně klienta ji ovládána zásuvným modulem knihovny jQuery s názvem *TreeViewJS*. Tento modul u uzlů vykresluje rozbalovací značky a skrývá nebo zobrazuje pod uzly.

### 5.4.3 Formulář

Komponenta *FormControl* vytváří pomocí knihovny *Nette\Form* a struktury získané ze služby *Action* požadované tabulky editovatelný formulář. Při prvním zobrazení jsou pole formuláře uzamčená (pouze ke čtení) a relace fungují jako odkazy. Po kliknutí na tlačítko edit se pole, ke kterým má uživatel přístup pro úpravy odemknou, a uživatel může provést potřebné změny. Tlačítko edit po této operaci nahradí uložit. Před odesláním formuláře se provede validace obsahu prvků a až potom se celý formulář odešle na server, kde jej komponenta předá pomocí abstraktní metody aktuálnímu presenteru, který s daty provede požadované operace a vrátí je zpět. Poté služba *Action*, provede zápis takto zpracovaných dat do databáze.

### 5.4.4 Navigace

Navigaci v systému provádí persistentní komponenta *MenuControl*. Jejím výstupem je odrážkový seznam navigačního menu ohraničený tagy `<nav>`. Vytváří si vlastní instanci třídy *Table* a k datům přistupuje jejím prostřednictvím bez služby *Action*. Obsahuje model, který ji umožňuje kontrolovat a spravovat jednotlivé položky menu podle nainstalovaných a aktivních modulů a práv aktuálně přihlášeného uživatele. K vytvoření odkazů využívá metod služby *Router*.

## 5.5 Presentery

### 5.5.1 BasePresenter

Základní třída, kterou dědí každý presenter. Dědí *Nette\Application\UI\Presenter* kde hlavně rozšiřuje metody *formatLayoutTemplateFiles()* a *formatTemplateFiles()* o možnost defaultních šablon. Obsahuje persistentní proměnou *source*, která určuje požadovanou akci.

### 5.5.2 SecurePresenter

Přes metodu *startup()*, kontroluje autentizaci a autorizaci uživatele. V případě neautentizovaného přístupu provede přesměrování na přihlášení. Dále také injektuje službu *action* pro předání základních informací viewu.

### 5.5.3 ComponentPresenter

Obsahuje metody pro vytvoření základních komponent jádra systému (menu, tabulka, strom, formulář...).

### 5.5.4 RenderPresenter

Obsahuje metody `render()` pro čtyři základní zobrazení.

Akce	Parametr	Potřebná práva	Komentář
Default	source	r	výchozí zobrazení, seznam dat (tabulka, strom).
Show	source, id	r	Zobrazení vybraného záznamu
Edit	source, id	w	Formulář pro editaci vybraného záznamu
add	source	c	Formulář pro přidání záznamu
delete	source, id	d	Smazání záznamu

Tab. 1.: Základní metody `render()`, jejich parametry a potřebná práva.

### 5.5.5 ActionPresenter

Výchozí presenter pro zobrazení dat. Jedná se o univerzální presenter pro zobrazení a práci s daty bez možnosti jejich zpracování. Presenter je registrován v modulu *base* a je určen k vytváření sekcí systému, které nevyžadují žádné operace s daty. Je také výchozí šablonou pro vytváření modulů. Obsahuje dvě metody, které se volají před a po úspěšném zápisu do databáze.

```

/**
 * Prove se před zápisem do databáze
 * @param array $values
 * @return array
 */
protected function beforeProcess($values){
//Kód vykonaný předo uložením záznamu
return $values;
}

/**
 * provede se po úspěšném zápisu do databáze
 * @param integer $id Id právě uloženého záznamu
 */
protected function afterProcess($id){
if (is_int($id)) {
//Kód vykonaný po uložení záznamu
}
}

/**
 * Pokud je nastavován máster, je nutné zabezpečit, aby byl jen jeden
 * @param array $values
 * @return array
 */
protected function beforeProcess($values) {
$this->ready = $this->move->prepareMove($values);
return $values;
}

/**
 * V případě, že se podaří uložit, systém zruší starý hlavní sklad
 * @param type $id
 */
protected function afterProcess($id) {
if (is_int($id) && $this->ready === TRUE) {
$this->move->processMove($id);
}
}

```

Obr. 13: Metody `beforeProcess()` a `afterProcess()` v presenteru *Action* a jejich implementace v presenteru *Move*.

### 5.5.6 LoginPresenter

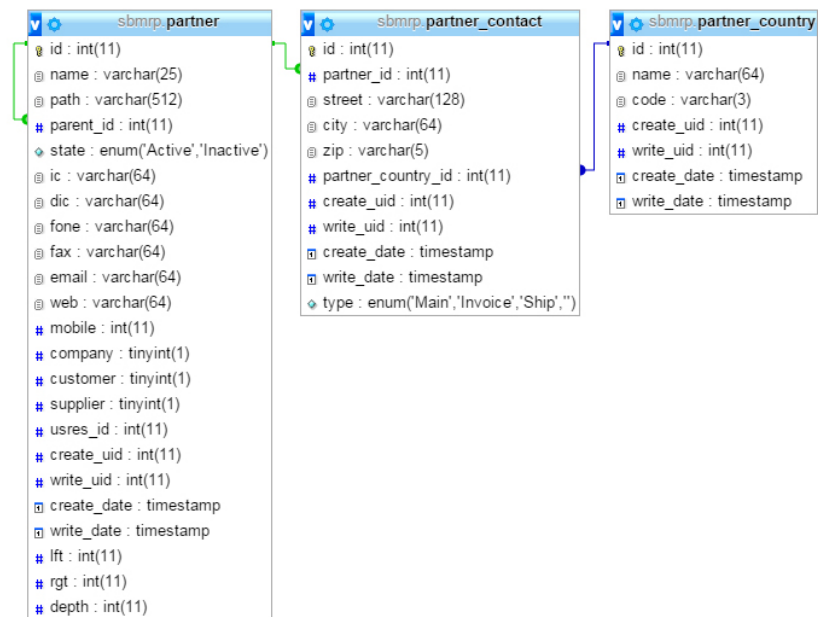
Tento presenter dědí pouze BasePresenter. Má vlastní layout šablonu i soubor se styly. Náhodný návštěvník má k dispozici pouze minimum kódu nutného k zobrazení přihlašovací stránky.

## 6 MODULY SYSTÉMU

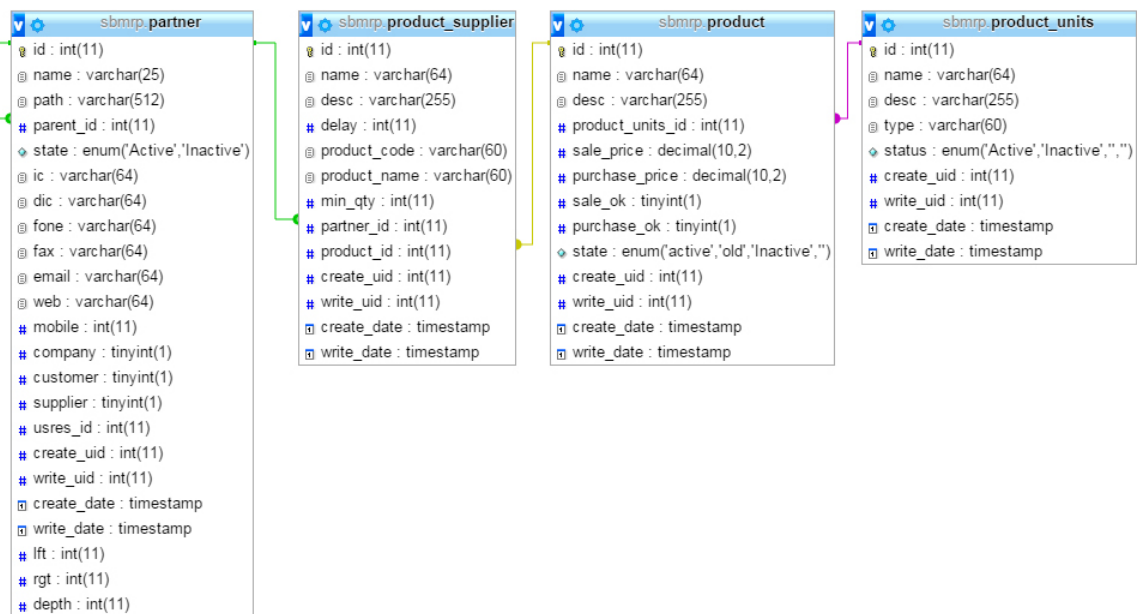
Tato kapitola se jmenuje základním modulům v systému.

### 6.1 Partneři a výrobky

Modul *Partner* i *Product* obsahují stejně nazvané presentery. Pro svou činnost nevyžadují žádné přidavné modely ani pohledy. Aktuálně využívají pouze metod a služeb jádra. Celá jejich struktura je nadefinována v databázi.



Obr. 14: Struktura tabulek modulu *Partner*



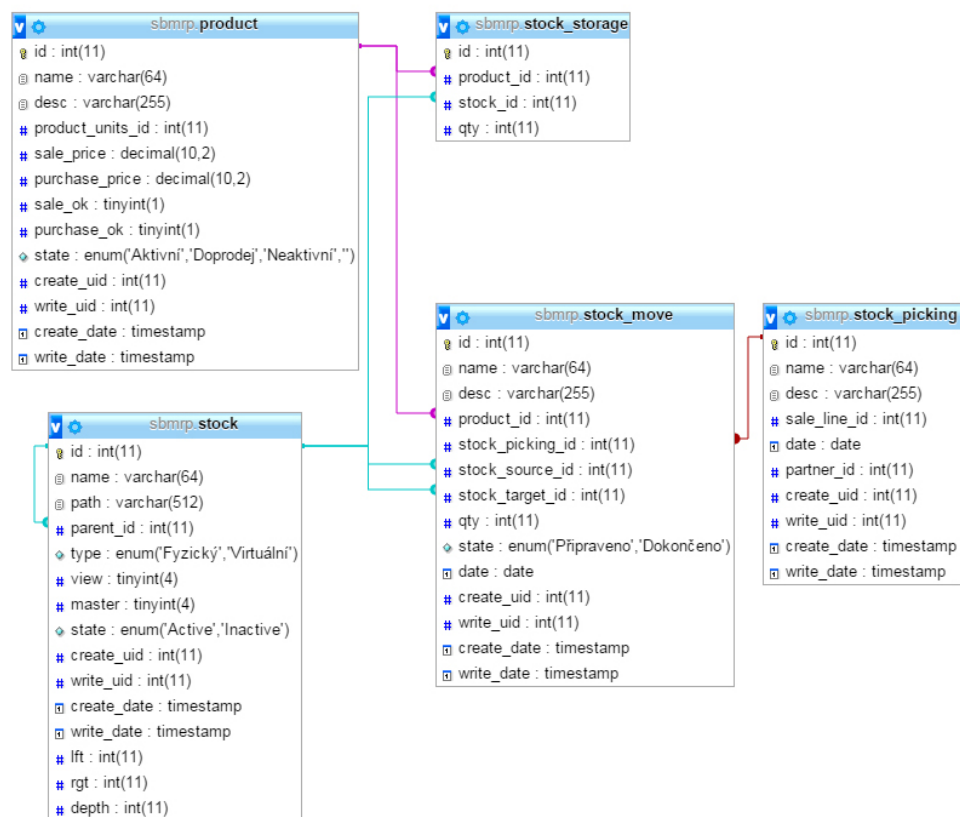
Obr. 15: Struktura tabulek modulu *Product* s vazbou na tabulku *Partner*

Data v tabulkách *partner\_contact* a *partner\_country* u modulu *Partner* se upravují pomocí výchozího presenteru *Action*. Odpovídající implementace je i pro data v tabulkách *product\_supplier* a *product\_units*.

## 6.2 Sklad

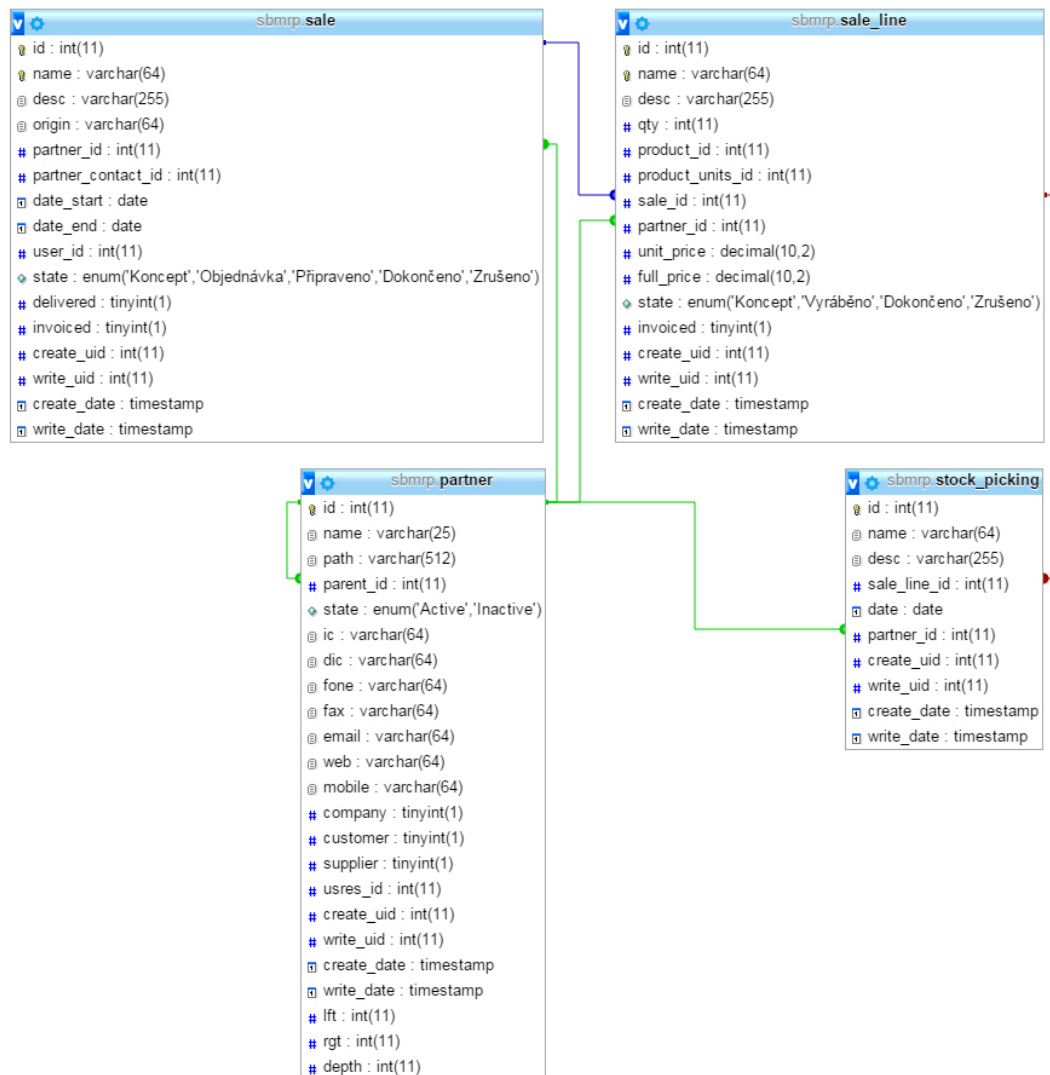
Modul *Store* obsahuje presentery *Move*, *Stock* a *Store*. Součástí modulu je i model *Move*, jehož úkolem je aktualizovat množství v jednotlivých skladech. Je implementován jako závislost u presenterů *Move* a *Store*. Vytvoření akce u těchto presenterů generuje aktualizaci stavu, jak bylo původně navrženo. Oba presentery k vygenerování akce využívá metody *beforeProces()* a *afterProcess()*, které jsou uvedeny výše.

Vytvořením pohybu v tabulce *stock\_move*, produktu obsaženého v tabulce *Product*, ze zdrojového skladu *stock\_source\_id* do cílového skladu *stock\_target\_id*, se provede pomocí modelu *Move*, aktualizace množství v obou skladech.

Obr. 16: Struktura tabulek modulu *Stock* s vazbou na tabulku *product*.

### 6.3 Prodej

Modul *Sale* obsahuje stejnojmenný presenter a model. Úkolem presenteru je jednoduché vynásobení zadaných kusů cenou za kus. Model sale naproti tomu vytváří čísla pro objednávky, generuje proces v tabulce *stock\_picking*, který slouží, jako datový kontejner pro záznamy výroby a zpětně upravuje stav zakázky v závislosti na změnách v něm provedených. *stock\_picking* si neuchovává informaci o svém stavu. První položka, která je do něj vložená je požadovaný výrobek(dále může následovat spotřebovaný, nebo zničený materiál a pod).



Obr. 17: Struktura tabulek modulu *Sale* s vazbou na tabulky *partner* a *sock\_picking*.

Stav řádku objednávky se určuje ze stavu tohoto přesunu. Stav celé objednávky je se následně určí na základě stavu jednotlivých řádků objednávky. Jsou-li všechny procesy (řádky) označeny jako hotové, je celá objednávka označena jako připravená. Je-li doručená a zaplacená, je označena za dokončenou.

## 6.4 Budoucí vývoj a rozšíření

System poskytuje velmi dobrý základ pro další vývoj a rozšíření. Plány na další moduly jako například evidence docházky už jsou v době psaní této bakalářské práce ve stádiu konceptů a brzy se přistoupí k jejich implementaci.

Použité technologie PHP a Nette, umožňují správu více databází na různých platformách. Tato vlastnost byla při vytváření databázového rozhraní implementována zatím, jako experimentální a bude s největší pravděpodobností rozšířena. Nové moduly pak budou moci pracovat i s externími daty v provozu, jako je například elektronický docházkový systém, nebo tvorba podkladů pro účetní systém.

## ZÁVĚR

Cílem této práce bylo vytvořit jednoduchý, ale silný nástroj pro provoz malé firmy. Po nastudování potřebných technologií byla naprogramována webová aplikace, která splnila původní očekávání a v některých ohledech překonala původní předpoklady.

Možnost ukládání struktury formulářů přímo do databáze umožnila vytvářet se stejným kódem různé struktury a obsahy jak ve formulářích, tak v jednotlivých komponentách. Příkladem jsou moduly *Partner* a *Product*, na které nebyl k vytvoření použit žádný dodatečný kód. Vznikly v administračním rozhraní pomocí nástrojů, kterými systém disponuje, a jsou uloženy v databázi. Tato skutečnost dává správci systému možnost pracovat pouze s prostředky, které má v aplikaci k dispozici, vytvářet nové vlastní struktury a reagovat tak na aktuální požadavky firmy.

V modulu *Sale* je pracovník prodeje schopen vytvářet objednávky, s dynamicky proměnným počtem řádků, které nesou její označení a pořadové číslo položky. Tento způsob zajistí přehlednou evidenci a případnou dohledatelnost. Systém sám na jejich základě vytváří úkoly pro výrobu s termíny dokončení. Tyto úkoly v sobě slučují všechny podstatné informace jako například spotřebovaný nebo zničený materiál v podobě pohybů materiálu. Zpětně při označení výrobku jako dokončený, systém aktualizuje stav položky nebo celé objednávky, pokud jsou všechny výrobky dokončeny.

Modul *Store* je vytvořen tak aby kopíroval logickou strukturu jednotlivých částí provozu a sledoval pohyb materiálu od dodavatele k místu spotřeby a výrobek od místa vzniku k zákazníkovi. Kterákoliv část struktury dokáže zobrazit, které výrobky se v ní nacházejí nebo nacházely a jaký je jejich aktuální stav. Stav skladu označený jako hlavní se navíc zapisuje i do karty výrobku, takže u každé položky inventáře je možné zobrazit její aktuální množství.

Systém také umožňuje jednoduše opravit eventuální chyby pracovníků pouhou změnou stavu položky, což je u systémů se složitou strukturou většinou akce vyžadující několik po sobě jdoucích operací. Vrátit dokončený výrobek zpět do procesu výroby, například z důvodů reklamace lze pouze změnou jeho stavu. Systém zpětně aktualizuje všechna potřebná pole a již dokončený úkol se opět zobrazí ve výrobě.

Každá akce, kterou v systému uživatel vytvoří nebo upraví, je zaznamenána v podobě jména a času provedení a vypisuje se pod každým zobrazením formuláře. Tyto informace systém eviduje zcela automaticky a neumožňuje jejich změnu.

V současné době je systém připraven ke zkušebnímu nasazení do reálného provozu. Vedení se k jeho provedení a funkcím staví kladně a slibují si od jeho zavedení zjednodušení a zrychlení interních evidenčních činností.

**SEZNAM POUŽITÉ LITERATURY**

- [1] MAKOVEC, Jaromír. *Organizace a plánování výroby*. Vyd. 1. Praha: Vysoká škola ekonomická, Fakulta podnikohospodářská, 1993, 273 s. ISBN 8070791713.
- [2] BRUCKNER, Tomáš. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012, 357 s. Management v informační společnosti. ISBN 978-80-247-4153-6.
- [3] BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012, 323 s. Management v informační společnosti. ISBN 978-80-247-4307-3.
- [4] GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika*. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009, 496 s. Expert (Grada). ISBN 978-80-247-2615-1.
- [5] CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 9788025137338.
- [6] SCHLOSSNAGLE, George. *Pokročilé programování v PHP 5*. Brno: Zoner Press, 2004, 640 s. ISBN 80-868-1514-5.
- [7] GILMORE, W. *Velká kniha PHP 5 & MySQL: kompendium znalostí pro začátečníky i profesionály*. Vyd. 1. Brno: Zoner Press, 2005, 711 s. ISBN 808681520x.
- [8] *Nette Foundation: Dokumentace* [online]. 2.2. 2014 [cit. 2015-01-26]. Dostupné z: <http://doc.nette.org/en/2.2/>
- [9] BERNARD, Borek. *Úvod do architektury MVC. Zdroják: o tvorbě webových stránek a aplikací* [online]. 2009 [cit. 2015-04-26]. Dostupné z: <http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>
- [10] GRUDL, David. *Začínáme s Nette Framework. Zdroják: o tvorbě webových stránek a aplikací* [online]. 2009 [cit. 2015-04-26]. Dostupné z: <http://www.zdrojak.cz/serialy/zaciname-s-nette-framework/>
- [11] MARCANÍK, Tomáš. *Trojrozměrná správa přístupových práv pro webové systémy*. Zlín, 2010. Dostupné z: <http://digilib.k.utb.cz/handle/10563/12336>. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Ing. Tomáš Dulík, Ph. D.

- [12] ŠTRAUCH, Adam. 2012. Databáze MariaDB válcuje MySQL. *Root.cz: Informace nejen ze světa linuxu* [online]. [cit. 2015-05-10]. ISSN ISSN 1212-8309. Dostupné z: <http://www.root.cz/clanky/databaze-mariadb-valcuje-mysql/>
- [13] ZELENKA, Petr. Metody ukládání stromových dat v relačních databázích. *Interval.cz*. [Online] 2005. [Citace: 27. 04 2015.] <https://www.interval.cz/clanky/metody-ukladani-stromovych-dat-v-relacnich-databazich/>.
- [14] PTÁČEK, Pavel. Ukládáme hierarchická data v databázi. *Zdroják: o tvorbě webových stránek a aplikací* [online]. 2012 [cit. 2015-04-27]. Dostupné z: <http://www.zdrojak.cz/serialy/ukladam/>
- [15] ODELL, Den. *JavaScript: průvodce programováním ajaxových aplikací*. Vyd. 1. Brno: Computer Press, 2010, 368 s. ISBN 9788025127339.
- [16] JQUERY FOUADATION. *JQuery Learning Center* [online]. 2015 [cit. 2015-04-27]. Dostupné z: <http://learn.jquery.com/>
- [17] CHAFFER, Jonathan. *Mistrovství v jQuery: kompletní průvodce vývojáře*. 1. vyd. Brno: Computer Press, 2013, 384 s. ISBN 978-80-251-4103-8.
- [18] JQUERY FOUADATION. *Jquery UI* [online]. 2015 [cit. 2015-04-27]. Dostupné z: <http://jqueryui.com>
- [19] SPRYMEDIA UI DEVELOPMENT. *DataTables: Table plug-in for jQuery* [online]. 2007 [cit. 2015-04-27]. Dostupné z: <https://www.datatables.net/>
- [20] SAMARJET. TreeViewJS. In: *GitHub* [online]. 2015 [cit. 2015-04-27]. Dostupné z: <https://github.com/samarjeet27/treeViewjs>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

- MPTT Modified Preorder Tree Traversal (Traversování kolem stromu)
- ACL Access control list (seznam pro řízení přístupu)
- PHP PHP: Hypertext Preprocessor
- RIA Rich Internet Application (webové aplikace)
- CSS Cascading Style Sheets (kaskádové styly)
- API Application Programming Interface (rozhraní pro programování aplikací)

**SEZNAM OBRÁZKŮ**

Obr. 1: Architektura MVP v Nette Framework [10] .....	15
Obr. 2: Životní cyklus presenteru [8].....	15
Obr. 3: Struktura a popis prvků formuláře Nette [8] .....	18
Obr. 4: Princip číselného ohodnocování stromu [14].....	19
Obr. 5: Některé prvky jQuery UI [18] .....	20
Obr. 6 Datatables - Zásuvný modul pro jQuery [19].....	21
Obr. 7 TreeViewJS – Zásuvný modul pro jQuery [20] .....	21
Obr. 8: Use case Diagram procesu výroby .....	27
Obr. 9: Use case Diagram modulu sklad .....	28
Obr. 10: Use case diagram modulu prodej .....	29
Obr. 11: Struktura tabulek skupin <i>base</i> a <i>table</i> .....	31
Obr. 12: Struktura tabulek skupin <i>acl</i> a <i>users</i> .....	32
Obr. 13: Metody <i>beforeProcess()</i> a <i>afterProcess()</i> v presenteru <i>Action</i> a jejich implementace v presenteru <i>Move</i> . .....	35
Obr. 14: Struktura tabulek modulu <i>Partner</i> .....	37
Obr. 15: Struktura tabulek modulu <i>Product</i> s vazbou na tabulku <i>Partner</i> .....	38
Obr. 16: Struktura tabulek modulu <i>Stock</i> s vazbou na tabulku <i>product</i> . .....	39
Obr. 17: Struktura tabulek modulu <i>Sale</i> s vazbou na tabulky <i>partner</i> a <i>sock_picking</i> . .....	40

**SEZNAM TABULEK**

Tab. 1.: Základní metody <i>render()</i> , jejich parametry a potřebná práva.....	35
---	----

## SEZNAM PŘÍLOH

- P I Aktivní moduly serveru Apache
- P II Uživatelský manuál (11. stran)
- P III Administrační manuál (10 stran)
- P IV API dokumentace Systému (Příloha dostupná na přiloženém DVD)
- P V Obsah přiloženého DVD

## PŘÍLOHA P I: AKTIVNÍ MODULY SERVERU APACHE

Moduly Apache	
✓ access_compat_module	lbmethod_bytraffic_module
✓ actions_module	lbmethod_heartbeat_module
✓ alias_module	ldap_module
✓ allowmethods_module	logio_module
✓ asis_module	✓ log_config_module
✓ auth_basic_module	log_debug_module
auth_digest_module	log_forensic_module
authn_anon_module	lua_module
✓ authn_core_module	✓ mime_module
authn_dbd_module	mime_magic_module
authn_dbm_module	✓ negotiation_module
✓ authn_file_module	proxy_module
authn_socache_module	proxy_ajp_module
authnz_ldap_module	proxy_balancer_module
✓ authz_core_module	proxy_connect_module
authz_dbd_module	proxy_express_module
authz_dbm_module	proxy_fcgi_module
✓ authz_groupfile_module	proxy_ftp_module
✓ authz_host_module	proxy_html_module
authz_owner_module	proxy_http_module
✓ authz_user_module	proxy_scgi_module
✓ autoindex_module	ratelimit_module
buffer_module	reflector_module
cache_module	remoteip_module
cache_disk_module	request_module
cern_meta_module	reqtimeout_module
✓ cgi_module	✓ rewrite_module
charset_lite_module	sed_module
data_module	session_module
dav_module	session_cookie_module
dav_fs_module	session_crypto_module
dav_lock_module	session_dbd_module
dbd_module	✓ setenvif_module
deflate_module	slotmem_plain_module
✓ dir_module	slotmem_shm_module
dumpio_module	socache_dbm_module
✓ env_module	socache_memcache_module
expires_module	socache_shmcb_module
ext_filter_module	speling_module
file_cache_module	ssl_module
filter_module	status_module
headers_module	substitute_module
heartbeat_module	unique_id_module
heartmonitor_module	userdir_module
ident_module	usertrack_module
imagemap_module	version_module
✓ include_module	vhost_alias_module
info_module	watchdog_module
✓ isapi_module	xml2enc_module
lbmethod_bybusyness_module	✓ php5_module

## PŘÍLOHA P II: UŽIVATELSKÝ MANUÁL

### Obsah Přílohy

<b>1</b>	<b>PŘEDMLUVA</b> .....	<b>52</b>
<b>2</b>	<b>PRVNÍ KROKY</b> .....	<b>52</b>
<b>2.1</b>	<b>SPUŠTĚNÍ SYSTÉMU</b> .....	<b>52</b>
<b>2.2</b>	<b>PŘIHLÁŠENÍ DO SYSTÉMU</b> .....	<b>52</b>
<b>2.3</b>	<b>BAREVNÉ ZNAČENÍ</b> .....	<b>53</b>
<b>2.4</b>	<b>KONTEXTOVÁ NÁPOVĚDA</b> .....	<b>53</b>
<b>3</b>	<b>ZÁKLADNÍ OVLÁDÁNÍ</b> .....	<b>53</b>
<b>3.1</b>	<b>NAVIGACE</b> .....	<b>54</b>
<b>3.2</b>	<b>AKCE UŽIVATELE</b> .....	<b>54</b>
<b>3.3</b>	<b>SEZNAM A STROM</b> .....	<b>55</b>
<b>3.4</b>	<b>FORMULÁŘ A JEHO PRVKY</b> .....	<b>56</b>
3.4.1	TEXTOVÉ POLE .....	57
3.4.2	ZAŠKRTÁVACÍ POLÍČKO.....	57
3.4.3	RÁDIOVÝ PŘEPÍNAČ.....	57
3.4.4	VÝBĚR.....	57
<b>4</b>	<b>MODULY</b> .....	<b>57</b>
<b>4.1</b>	<b>PARTNEŘI</b> .....	<b>57</b>
<b>4.2</b>	<b>PRODUKTY</b> .....	<b>58</b>
<b>4.3</b>	<b>PRODEJ</b> .....	<b>59</b>
<b>4.4</b>	<b>SKLAD</b> .....	<b>60</b>

## 1 PŘEDMLUVA

Tento dokument slouží k seznámení se základními operacemi s webovým systémem pro evidenci skladu a výroby. Účelem tohoto manuálu je seznámit uživatele se základními funkcemi a možnostmi systému. Manuál se nejprve zaměří na základní operace s daty a v dalších proběhne postupné seznámení s jednotlivými moduly.

Při procházení tohoto manuálu může čtenář nabýt dojmu, že se některé kapitoly a části opakují, nebo jsou si velice podobné. Je to dáno jednotnou strukturou systému, která zajišťuje, že jsou všechny operace ve všech částech systému stejné a snahou o ucelený pohled na každou prováděnou operaci. Manuál se snaží dodat všechny potřebné informace, i pokud jsou čteny jen vybrané kapitoly, o které má čtenář zájem.

Je třeba si uvědomit, že uživatel nemusí mít do všech popisovaných částí přístup. Nezobrazuje-li se vám některá sekce, tlačítko, nebo vám nejde upravit některá položka, je pravděpodobné, že na danou operaci nemáte dostatečná práva. Pokud jste přesvědčeni, že tato práva mít máte, kontaktujte vašeho administrátora.

## 2 PRVNÍ KROKY

### 2.1 Spuštění systému

Přístup do systému se získá otevřením okna prohlížeče a zadáním webové adresy. Adresu i přihlašovací údaje je potřeba si vyžádat od místního správce systému, popř. nadřízeného. Přihlášení do systému

### 2.2 Přihlášení do systému

Po načtení systému se provede přesměrování na přihlašovací stránku. Tato stránka obsahuje dvě editovatelná textová pole pro uživatelské jméno a heslo, zaškrtačací tlačítko pro zapamatování si uživatele a potvrzovací tlačítko pro odeslání. Přihlašovací údaje, nebo zapomenuté heslo je nutné vyžádat od správce systému. Po vyplnění uživatelského jména (1) a hesla (2) lze kliknout na tlačítko přihlásit (4) a v případě správných údajů bude uživatel přesměrován na předvolenou výchozí stránku.

Pokud jsou vloženy chybné údaje, systém nedovolí přihlášení a zobrazí upozornění na chybu. Z bezpečnostních důvodů systém neupozorňuje, zda bylo chybně vloženo heslo, nebo přihlašovací jméno.



Volba zapamatovat si uživatele (3), zachová přihlášení po dobu tří dnů. V opačném případě přihlášení vyprší po dvaceti minutách.

### 2.3 Barevné značení

Standardně v systému převládá šedá barva, která svými odstíny rozlišuje jednotlivé prvky stránky. Z tohoto monolitického barevného schématu vystupují dvě kontrastní barvy, které slouží k zvýraznění stavu prvků. Jedná se o červenou a zelenou.

Červená barva označuje aktivní prvky. Tímto jsou myšleny stisknuté tlačítka, zvolené volby, nebo vnitřní stavy různých procesů v systému. V kontextu formuláře jsou touto barvou zvýrazněny povinné položky. Slouží také k oddělení logických prvků stránky.

Zelená barva naproti tomu označuje aktivní prvek. Tedy prvek, se kterým se právě pracuje, nebo na něj ukazuje kurzor myši.

### 2.4 Kontextová nápověda

Najetím myši na vybraný prvek se objeví text s nápovědou. Jedná se o stručné vysvětlení, k čemu daný prvek slouží.

## 3 ZÁKLADNÍ OVLÁDÁNÍ

System disponuje čtyřmi základními komponentami pro ovládání, které mají různé zastoupení v jednotlivých modulech a sekcích.

### 3.1 Navigace

Tato komponenta se nachází v každém zobrazení systému. Nachází se vždy v horní části stránky. Navigace slouží k přesunu mezi jednotlivými moduly a sekcemi systému. V základním nastavení obsahuje položky prodej, sklad, Nákup, nastavení a úplně vpravo se nachází jméno právě přihlášeného uživatele, pokud jej má v systému vyplněné. V opačném případě se tam zobrazuje přihlašovací jméno. Počet položek i jejich názvy se mohou v závislosti na uživateli lišit. Manuál popisuje výchozí nepozměněný vzor. Vybráním položky v navigaci se rozbalí kontextové menu s dalšími položkami. Výběrem položky se provede přesměrování do požadované části systému.

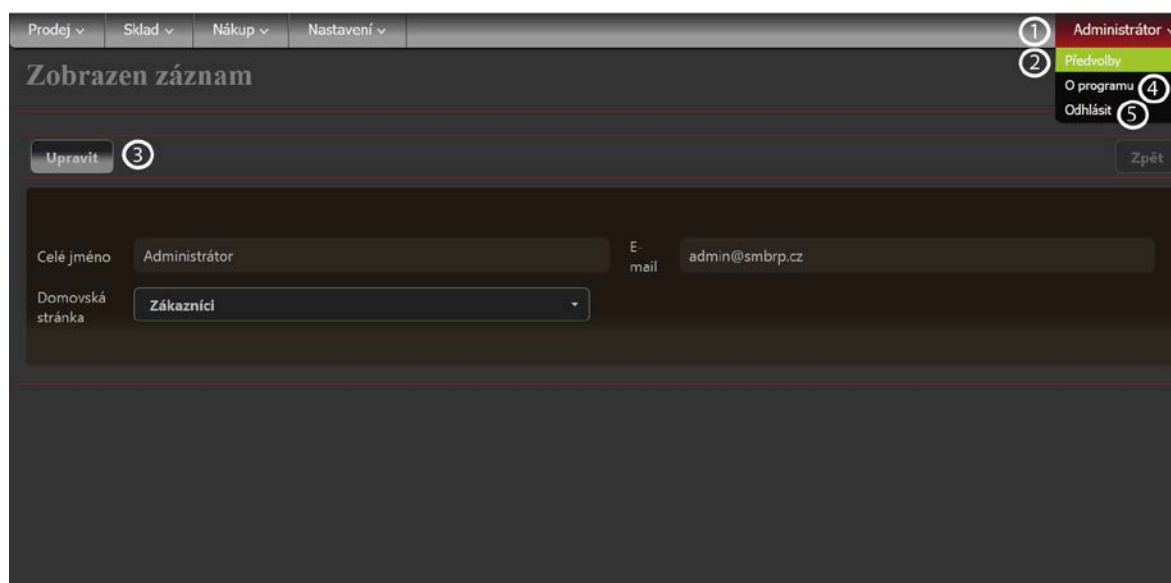
Navigace je rozdělena na dvě části. V levé sekci se nachází operační část pro práci se systémem a v pravé akce spojené s uživatelem. Menu se ovládá kliknutím na jméno požadované skupiny, čímž se zpřístupní další volby navigace.

### 3.2 Akce uživatele

Při rozkliknutí uživatelského jména (1) se vysune rozbalovací menu se třemi položkami.

První položka označená jako předvolby (2) umožňuje změnit výchozí předvolby uživatele, jako jsou výchozí stránka, jméno a další údaje, které chcete sdílet s ostatními uživateli v systému. Pro upravení hodnot je nutné kliknout na tlačítko upravit (3). Pro ukončení editace slouží tlačítka uložit, nebo pokud nechceme provést změny tak zrušit.

Položka s názvem O programu (4) zobrazí informace o programu, o jeho verzi a kontakt na správce.



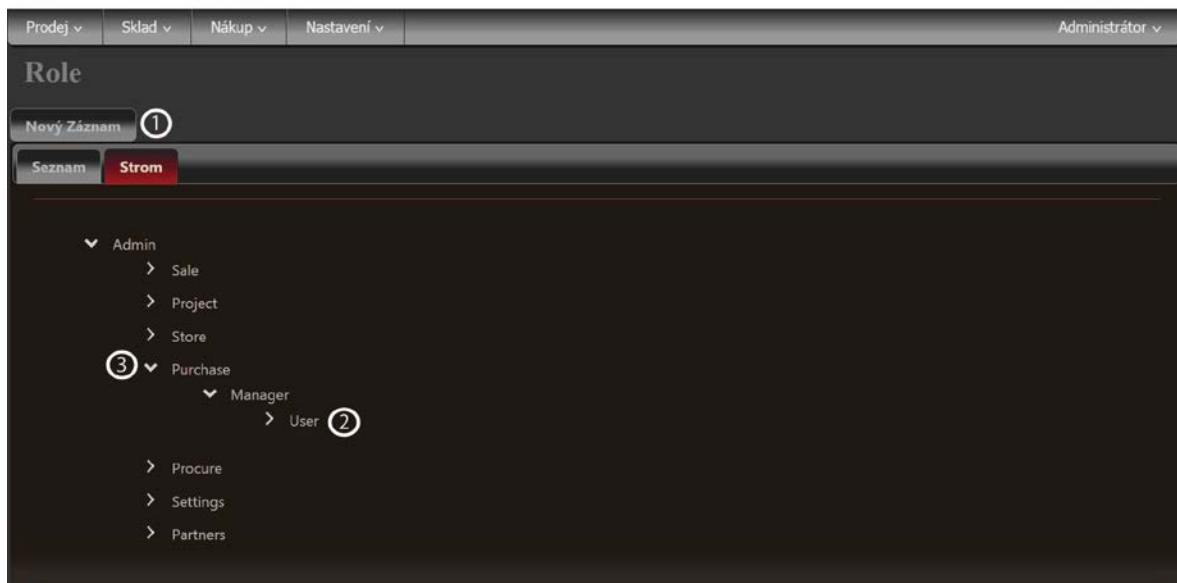
Poslední položka odhlásit (5), při volbě odhlásí uživatele ze systému a přesměruje jej zpět na přihlašovací obrazovku.

### 3.3 Seznam a strom

Seznam je tvořen tabulkou s popisovými sloupci v záhlaví a volitelně se součty položek v zápatí, pokud sloupec obsahuje číselné hodnoty. Tabulka má pouze informační charakter. K úpravě dat v ní obsazených slouží editační formulář. Nad tabulkou se nachází tlačítko přidat (1) pro vytvoření nového záznamu. Po rozkliknutí se provede přesměrování na prázdný formulář. Kliknutím na řádek tabulky (2) se se zobrazí všechny dostupné údaje z vybraného řádku s tlačítkem upravit k umožnění jejich editace a tlačítkem Smazat pro smazání záznamu. Tlačítko zpět slouží pro návrat do tabulky. Tabulka dále disponuje ovládacími prvky pro setřídění seznamu vedle názvů sloupců a filtrem pro omezení výpisu (5).

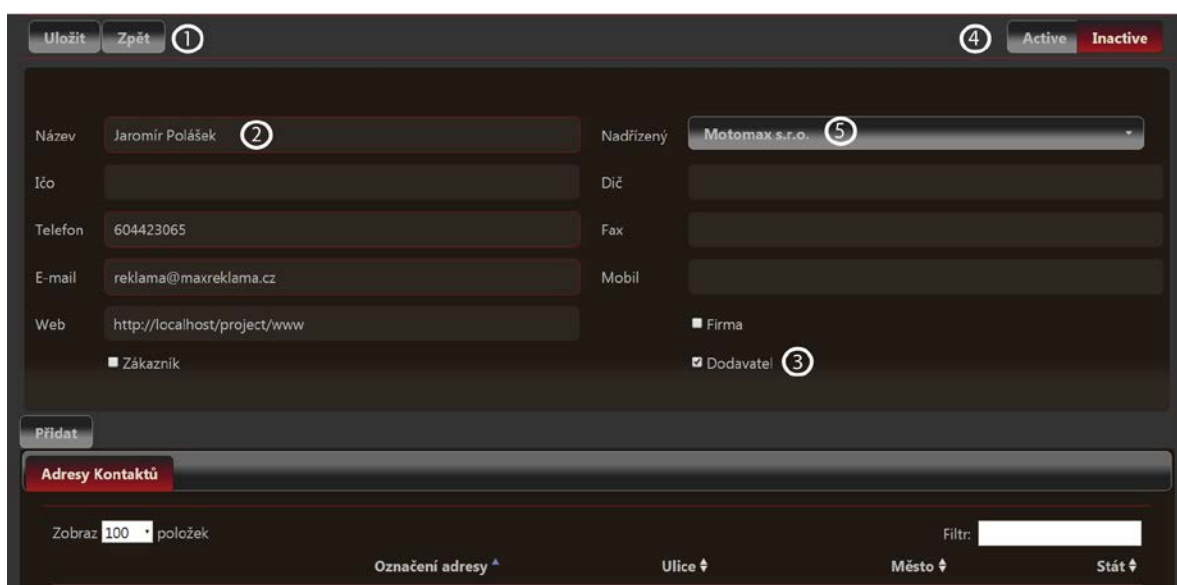
Název	Stav	Ičo	Telefon	E-mail
Jaromír Polášek	Neaktivní		604423065	reklama@maxreklama.cz
Motomax s.r.o.	Aktivní	27669971	0	info@motomax.cz

V závislosti na druhu zobrazovaných dat může být k dispozici také zobrazení ve formě stromu. Toto zobrazení navíc disponuje ovládacími prvky pro rozbalení a sbalení větví stromu (3).



### 3.4 Formulář a jeho prvky

Formulář slouží k editaci záznamu v systému. Nachází se v něm několik druhů editačních polí, které budou detailně rozebrány v následujícím textu. Ukončit editaci formuláře lze dvěma způsoby. Uložení zapsaných, nebo pozměněných hodnot tlačítkem uložit, čímž se nová, nebo pozměněná data uloží do systému, nebo kliknutím na tlačítko zrušit (1), které ukončí editaci bez zápisu dat do systému. V obou případech se provede přesměrování na výchozí zobrazení, kterým je ve většině případů seznam, nebo návrat na předchozí stránku, pokud jsme přišli z jiné sekce.



### 3.4.1 Textové pole

Toto editační pole slouží k zadání textu pole do formuláře. Je omezeno svojí délkou (defaultně 64 znaků). Jedná se o základní pole pro zadávání dat do systému (2).

### 3.4.2 Zaškrťovací políčko

Tzv. checkbox je jeden z nejjednodušších prvků formuláře, který může nabývat dvou hodnot. Zaškrtnuté pole znamená ano, nezaškrtnuté pole znamená ne (3).

### 3.4.3 Rádiový přepínač

Rádio list je seznam zaškrťovacích políček, který umožňuje vybrat jednu volbu z nabízených možností (4).

### 3.4.4 Výběr

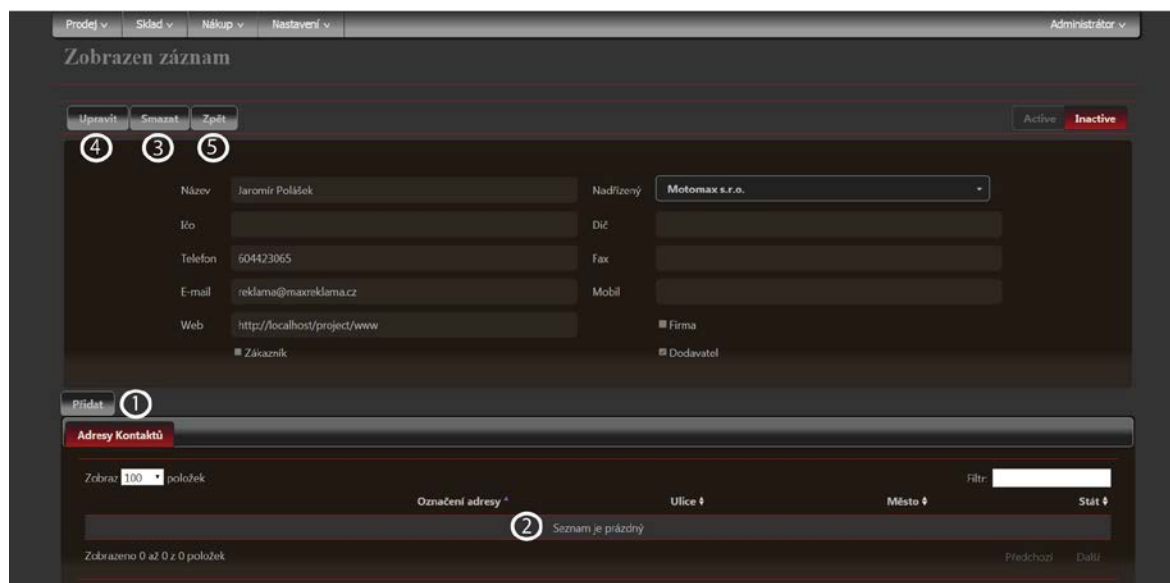
Select box umožňuje vybrat jednu z možností založenou na předchozím záznamu stejného nebo jiného uživatele. V principu umožňuje vybrat například pro zakázku zákazníka, nebo produkt pro řádek objednávky (5).

## 4 MODULY

### 4.1 Partneri

Seznamy partnerů se nachází v menu a jsou rozděleni do sekcí Prodej pod položkou zákazníci a v sekci nákup pod položkou dodavatelé. Vybráním jedné z těchto voleb se systém přesměruje do seznamu zákazníků. Přidat nový kontakt lze kliknutím na tlačítko přidat, které provede přesměrování na prázdný editační formulář. Vyplněním formuláře a jeho odesláním tlačítkem uložit se do systému přidá nový zákazník.

Kliknutím na již existujícího zákazníka se zobrazí všechny dostupné údaje o zákazníkovi. Pod těmito údaji se nachází tabulka s adresami. Kliknutím na tlačítko přidat (1), lze vytvořit novou adresu. Zobrazit adresu lze kliknutím na řádek s adresou (2). V tomto náhledu lze také zákazníka vymazat kliknutím na příslušné tlačítko (3).

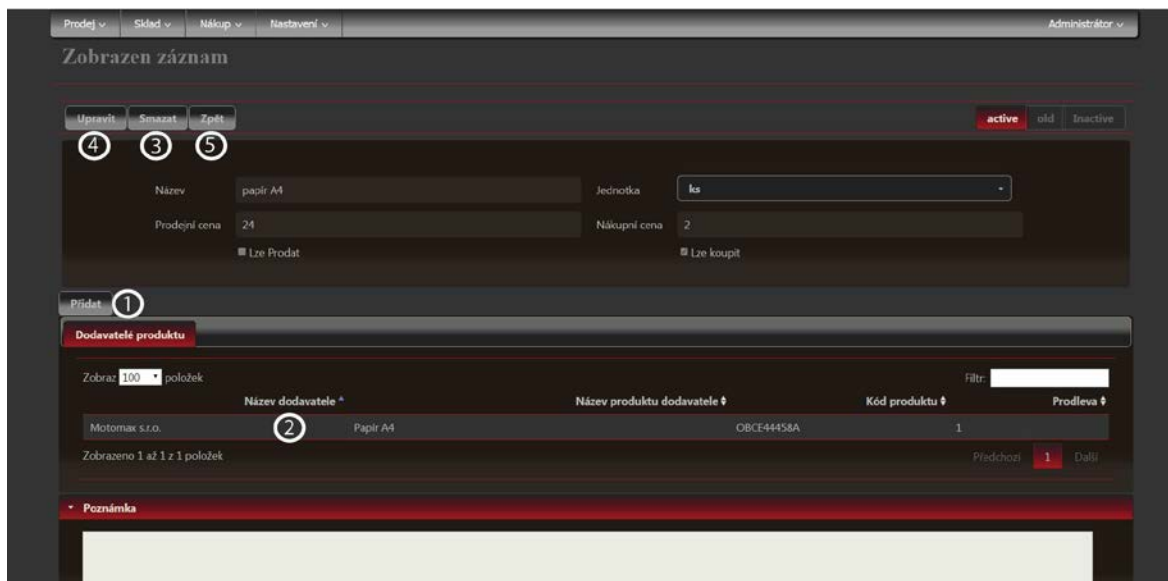


Upravit údaje o zákazníkovi lze kliknutím na tlačítko upravit (4) a k navrácení se zpět so seznamu slouží tlačítko zpět (5).

## 4.2 Produkty

Seznamy produktů se nachází v menu v sekci sklad pod položkou produkty. Vybráním této volby se systém přesměruje do seznamu Produktů. Přidat nový produkt lze kliknutím na tlačítko přidat, které provede přesměrování na prázdný editační formulář. Vyplněním formuláře a jeho odesláním tlačítkem uložit se do systému přidá nový produkt.

Kliknutím na již existující produkt se zobrazí všechny dostupné informace o požadovaném produktu. Pod těmito údaji se nachází tabulka s dodavateli. Kliknutím na tlačítko přidat (1), lze přidat produktu nového dodavatele. Zobrazit jej lze kliknutím na řádek s dodavatelem (2). V tomto náhledu lze také produkt vymazat kliknutím na příslušné tlačítko (3). Smazat produkt je možné pouze v případě, že není v systému použit. V opačném případě jej musíte deaktivovat.



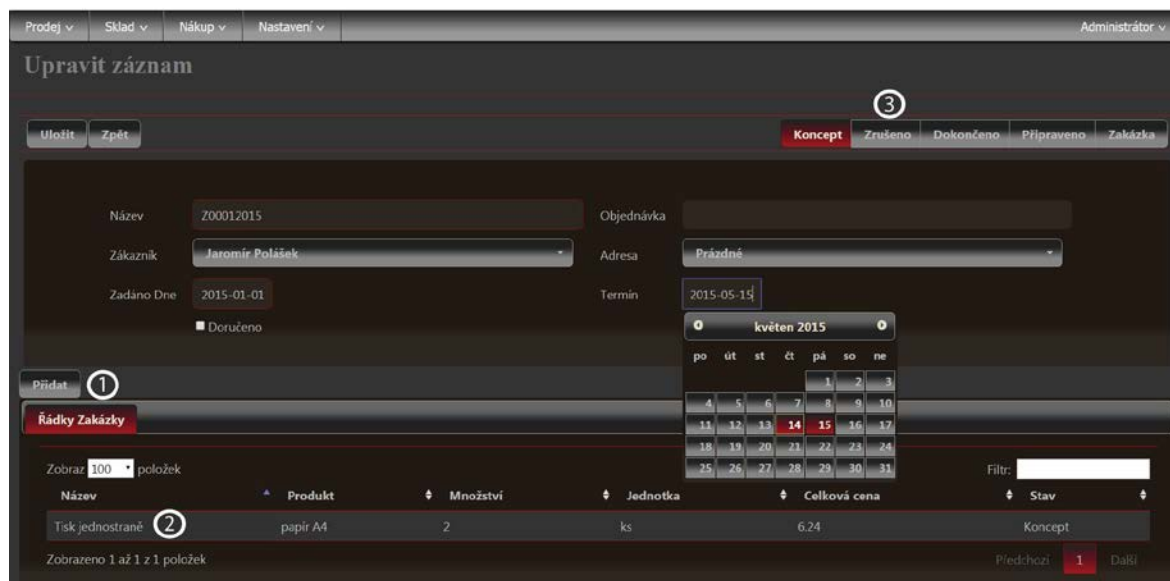
Upravit údaje o produktu lze kliknutím na tlačítko upravit (4) a k navrácení se zpět so seznamu slouží tlačítko zpět (5).

### 4.3 Prodej

Seznamy Zakázek se nacházejí v menu v sekci prodej pod položkami nabídky, zakázky expedice a archiv. V položce nabídky se nacházejí všechny objednávky, které nejsou schválené, nebo je čekají dodatečné úpravy. Slouží jako koncept k zakázkám. V položce zakázky se nacházejí všechny aktuálně prováděné zakázky a v položce archiv, se nacházejí dokončené zakázky.

Vybráním jedné z těchto voleb se systém přesměruje do seznamu Zakázek nebo nabídek. Přidat novou nabídku lze kliknutím na tlačítko přidat, které provede přesměrování na prázdný editační formulář. Vyplněním formuláře a jeho odesláním tlačítkem uložit se do systému přidá nová zakázka.

Kliknutím na zakázku se zobrazí informace. Pod těmito informacemi se nachází tabulka pro řádky objednávky.



Kliknutím na tlačítko přidat (1), lze přidat nový řádek objednávky. Zobrazit jej lze kliknutím na příslušný řádek (2). V tomto náhledu lze také zakázku vymazat kliknutím na příslušné tlačítko. Kvůli evidenci se však doporučuje zakázku zrušit (3), čímž se přesune do archivu a označí se jako zrušená.

#### 4.4 Sklad

Struktura skladu se nachází v menu v sekci sklad. Tato sekce obsahuje položky struktura a pohyby. Příjem zboží do skladu se provádí v sekci sklad položka pohyby přesunutím požadovaného počtu zboží ze skladu dodavatel, do skladu firmy. Výdej v sekci prodej položka expedice.

Struktura skladu obsahuje dva základní druhy skladů, fyzický a virtuální. Fyzickými sklady jsou chápány sklady, ve kterých jsou produkty fyzicky dohledatelné. Žádné výrobky v nich nevznikají ani nezanikají. Mohou se pouze přesouvat dovnitř a ven. Naproti tomu virtuální sklady slouží pouze jako evidence zdroje nebo cíle. Patří mezi ně sklad zákazník, dodavatel, Zmetky, nebo výroba. Jednotlivé přesuny jsou v systému označovány jako pohyby a soubor pohybů se nazývá balík. Balík, v systém také označen jako proces je soubor pohybů zboží, výrobků, nebo materiálu, které mají společný původ. Jakákoli objednávka, respektive její řádky, mají vlastní balík, do kterého je nutné vkládat všechny operace s výrobky, nebo materiálem, které jsou v rámci této objednávky, nebo její položky provedeny.

Nejjednodušším příkladem balíku pohybů je inventura. Inventura obsahuje seznam pohybů z pevného skladu do virtuálního a zpět, čímž se upravují počty výrobků na skutečnou úroveň. Obsah virtuálního skladu poté ukazuje inventurní manko.

**PŘÍLOHA P III: ADMINISTRACNÍ MANUÁL**

<b>1</b>	<b>PŘEDMLUVA</b> .....	<b>63</b>
<b>2</b>	<b>INSTALACE SYSTÉMU</b> .....	<b>63</b>
<b>2.1</b>	<b>POŽADAVKY NA SERVER</b> .....	<b>63</b>
<b>2.2</b>	<b>INSTALACE</b> .....	<b>63</b>
<b>2.3</b>	<b>ZNÁMÉ PROBLÉMY</b> .....	<b>64</b>
<b>2.4</b>	<b>SPUŠTĚNÍ A PRVNÍ PŘIHLÁŠENÍ DO SYSTÉMU</b> .....	<b>65</b>
<b>3</b>	<b>MOŽNOSTI NASTAVENÍ SYSTÉMU</b> .....	<b>65</b>
<b>3.1</b>	<b>SPRÁVA UŽIVATELŮ</b> .....	<b>65</b>
<b>3.2</b>	<b>SPRÁVA ROLÍ</b> .....	<b>65</b>
<b>3.3</b>	<b>SPRÁVA PŘÍSTUPOVÝCH PRÁV</b> .....	<b>66</b>
<b>3.4</b>	<b>SPRÁVA TABULEK</b> .....	<b>66</b>
<b>3.5</b>	<b>SPRÁVA POLÍ</b> .....	<b>67</b>
<b>3.5.1</b>	<b>SPECIÁLNÍ ELEMENTY A JEJICH APLIKACE</b> .....	<b>67</b>
<b>3.5.2</b>	<b>SPRÁVA AKCÍ</b> .....	<b>68</b>
<b>3.5.3</b>	<b>POLE AKCE</b> .....	<b>68</b>
<b>3.5.4</b>	<b>POLE FILTR</b> .....	<b>69</b>
<b>3.6</b>	<b>SPRÁVA MENU</b> .....	<b>69</b>
<b>4</b>	<b>VYTVÁŘENÍ VLASTNÍCH STRUKTUR</b> .....	<b>70</b>
<b>4.1</b>	<b>VYTVOŘENÍ TABULKY</b> .....	<b>70</b>
<b>4.2</b>	<b>VYTVOŘENÍ POLÍ</b> .....	<b>70</b>
<b>4.3</b>	<b>VYTVOŘENÍ AKCE</b> .....	<b>70</b>
<b>4.4</b>	<b>VYTVOŘENÍ POLOŽKY MENU</b> .....	<b>71</b>
<b>4.5</b>	<b>ZÁVĚREČNÁ NASTAVENÍ</b> .....	<b>71</b>

## 1 PŘEDMLUVA

Tento dokument slouží k seznámení s možnostmi nastavení webového systému pro evidenci skladu a výroby. Účelem tohoto manuálu je seznámit správce se základním nastavením a možnostmi úprav funkcí systému tak, aby práce s ním na různých úrovních co nejvíce odpovídala představám zadavatele a umožnila jeho maximální využití.

Manuál vychází z informací obsažených v uživatelském manuálu a k jeho plnému využití je vhodné si ho nejprve prostudovat. Účelem tohoto manuálu není naučit správce základním operacím v systému. Znalost těchto postupů je dostatečně popsána v manuálu uživatelském. Předpokladem je určitá znalost dané problematiky a orientace ve strukturách a principech databáze a řízení přístupu. Tento manuál neslouží jako učebnice ale jako popis metod s jakými systém pracuje a jak k nim přistupovat.

Systém disponuje mechanismy, které umožňují zásahy do struktury dat v databázi a řízení přístupu k nim. Jakýkoliv neodborný zásah může mít za následek znemožnění fungování systému. Je proto nezbytné ke všem operacím zde popsaným přistupovat obezřetně a dodržovat uvedené pravidla.

## 2 INSTALACE SYSTÉMU

### 2.1 Požadavky na server

Systém ke svému provozu vyžaduje webový server Apache, verze 2.4.4 a PHP verzi minimálně 5.4. Pro provoz databáze je potřeba MySQL server verze 5.6.

Na jiných verzích nebyl systém testován a není zaručena jeho plná funkčnost a kompatibilita.

Pro správnou funkci může být nezbytné mít v serveru Apache aktivovaný modul `rewrite_module`.

### 2.2 Instalace

V příloženém DVD se nachází ve složce *vivid* předinstalovaná verze ve formě virtuálního stroje s operačním systémem Ubuntu. Uživatel této předinstalované verze má jméno *student* a heslo *opsys*.

Prvním krokem instalace systému je nakopírování zdrojových souborů ze složky *system* do kořenového adresáře vlastního serveru.

Následně je nutné pomocí některého z nástrojů pro práci s databázovým serverem Mysql provést import struktury a základních dat ze složky *database*. Tento soubor vytvoří uživatele a databázi s názvem *sberp*. A přístupovým heslem, které je uvedeno v souboru *config.local.neon*, který se po instalaci nachází ve složce *app\config*. Jeho struktura vypadá následovně:

database:

data:

dsn: 'mysql:host=127.0.0.1;dbname=sberp'

user: sberp

password:

info:

dsn: 'mysql:host=127.0.0.1;dbname=information\_schema'

user: sberp

password:

options:

lazy: yes

*dsn*: V obou sekcích *data* i *info* je potřeba vyplnit položky *mysql:host*, a *password*, pokud si jej přejete změnit. Změnu je třeba poté udělat i u uživatele *sberp* v databázi MySQL. Informační schéma by mělo zůstat nezměněno.

*Mysql:host* definuje připojení k MySQL serveru

*dbname* definuje název databáze, výchozí je *sberp*. Importovaný soubor *sberp.sql* vytváří databázi právě tohoto jména.

*User* je jméno uživatele, který má práva přistupovat k zvolené databázi a *password* je heslo tohoto uživatele.

### 2.3 Známé problémy

Pozn. Po instalaci je ve složce *www/checker* soubor *checker.php*. Který vám může pomoci s nastavením serveru.

Pro použití aliasů u serveru apache je nutné v souboru *.htaccess* ve složce *www*, na řádku 11 odkomentovat `# RewriteBase /` a za lomítkem uvést cestu k složce *www*, kde `/` je kořen aliasu. (např. `/erp/www`).

## 2.4 Spuštění a první přihlášení do systému

Přístup do systému se získá otevřením okna prohlížeče a zadáním přidělené webové adresy. V případě lokální instalace je to `http://localhost/www`, pokud jste systém neinstalovaly do zanořené adresářové struktury.

Při spuštění se systém dotáže na jméno a heslo. Pokud se jedná o první přihlášení, je výchozí jméno *admin* a heslo *root1234*. Výchozí heslo se obecně doporučuje ihned změnit.

## 3 MOŽNOSTI NASTAVENÍ SYSTÉMU

Systém obecně disponuje nástroji pro správu uživatelů, jejich přístupových práv, formulářů, a výpisů. Všechny tyto nástroje se nacházejí v menu v sekci Nastavení.

### 3.1 Správa uživatelů

Správa uživatelů se nachází v sekci nastavení pod položkou uživatelé. Uživatele lze přidat kliknutím na tlačítko tohoto jména a vyplněním formuláře. Každému novému uživateli vytvoří heslo administrátor, které si uživatel poté může změnit. Pro aktivaci uživatele je potřeba změnit jeho stav na aktivní. V opačném případě se nový uživatel nebude moci přihlásit. Po uložení nového uživatele a následném kliknutí na řádek v tabulce s jeho jménem lze uživateli přidělit role.

Uživatel může být členem jedné nebo více rolí v závislosti na jeho plánovaném určení. Role lze spravovat v následující položce menu.

### 3.2 Správa rolí

Pod uživateli se v menu nachází položka role. Tato položka obsahuje seznam všech rolí v systému. V záložce strom je zobrazena jejich hierarchie pro lepší přehled o jejich dědičnosti. Tento nástroj umožňuje vytvořit strukturu odpovídající hierarchii podniku.

Při odstraňování rolí je nutné brát v úvahu stromovou strukturu. Systém neumožní odstranit roli, na kterou jsou navázané jiné role. Nejprve je musíte smazat, nebo přesunout pod jinou roli.

Speciální postavení v systému mají role Admin, která garantuje všechna práva a User, která se automaticky přidělí každému uživateli. Role User garantuje přístup k nastavením uživatele. Pokud by se je i přes různá omezení podařilo nedopatřením smazat, systém si je znova vytvoří.

### 3.3 Správa Přístupových práv

V tomto nástroji lze každé roli přidělit zdroj a oprávnění. Ve výchozím stavu garantuje systém roli admin všechna práva. Odebírat tyto práva administrátorovy se z praktických důvodů nedoporučuje. Systém je vázán na tuto roli a v případě změny ve zdrojích jí automaticky zajistí všechna práva. Pokud by byla tato role z nějakého důvodu ze systému odstraněna, mohlo by dojít k znepřístupnění minimálně nových sekcí. V praxi by to znamenalo, že by nezůstal nikdo, kdo by práva delegoval.

Zdrojem jsou zde myšleny tabulky a buňky v databázi a také položky menu, které se řídí zvlášť. Tyto zdroje si systém automaticky aktualizuje při provedení změny ve struktuře zobrazení dat, jak bude uvedeno dále.

Logika řízení přístupu v systému je založena na dvou větách: „Má-li uživatel oprávnění pro operaci minimálně s jednou buňkou tabulky, má přístup k tabulce. Přístup k tabulce však negarantuje přístup k buňce“. Ekvivalentně k navigaci. Má-li uživatel přístup k položce uživatelé, má přístup i do sekce nastavení. Neplatí to však obráceně. Garantování přístupu k sekci nastavení, neumožní přístup do položky uživatelé. Při vytváření oprávnění je potřeba brát tyto pravidla na vědomí.

Oprávnění je jediný způsob jak roli umožnit provést operaci se zdrojem. Obecně zná systém čtyři druhy oprávnění, které jsou ekvivalentem operací číst, vytvořit, upravit a smazat.

### 3.4 Správa tabulek

Položka tabulky v sekci nastavení umožňuje specifikovat zdroj dat v databázi a dát mu popis. Ke každé takto zaznamenané tabulce dokáže systém na základě polí, popsaných níže vygenerovat formulář nebo seznam. Každý záznam musí odrážet skutečnou strukturu dat.

Klíčovými poli jsou zde *objekt*, který udává název zdroje a *typ*, který uvádí, zda se jedná o jednoduchá, nebo hierarchická data. Objekt je navíc použit pro práci s daty a k určení zdroje oprávnění.

### 3.5 Správa polí

Položka pole v sekci nastavení slouží k identifikaci buňky v tabulce. Nese informaci o jejím formátu, způsobu editace vazbě a zobrazení.

- *Název* zde určuje označení pole ve výpisu nebo formuláři.
- *Pole* obsahuje skutečný název buňky v databázi. Název buňky musí v databázi skutečně existovat, jinak systém položku ignoruje.
- *Popis* všeobecně určuje, co se zobrazí v *tooltipu*, kromě buňky typu *enum*, kde obsahuje možné hodnoty.
- *Tabulka* odkazuje na název objektu v sekci Tabulek, se kterým je pole spojeno.
- *Element* určuje, jaký prvek má být použit pro zobrazení a editaci pole.
- *Relace* odkazuje na pole v jiné tabulce (bude popsáno dále).
- *Povinné* určuje, zda je položka povinná.
- *Pouze čtení* zamkne pole pro úpravy ze strany uživatele, data však mohou být měněna modulem.
- *Seznam* zajistí zobrazení pole v seznamu.
- *Pořadí* určuje pořadí položky ve formuláři nebo jako sloupce v seznamu.
- *Zápatí* určuje, zda má být v seznamu uveden součet hodnot. Tato funkce funguje pouze, je-li zápatí současně aktivováno v sekci správa tabulek.

Odeslání formuláře bez ohledu na změny v něm uvedené generuje akci, která aktualizuje seznam zdrojů. Tato operace sebou může v určitých případech nést jistý výkonnostní dopad. Určení zdrojů je z hlediska bezpečnosti pro systém klíčové, proto nemůže být tato operace svěřena uživateli.

#### 3.5.1 Speciální elementy a jejich aplikace

*Selektor* je speciálním prvkem formuláře, který při své definici vyžaduje specifický přístup. Slouží k definici vazeb mezi tabulkami. Položka pole zde musí být pojmenována podle názvu cílové tabulky, plus musí být na konci uvedeno *\_id*. Například *base\_id* odkazuje na konkrétní index *id* v tabulce *base*. Následně je v poli *Relace* uvedeno obsah které buň-

ky, nebo buněk má systém zobrazit. Pokud chcete i možnost hodnoty NULL je nutné za výpis napsat /NULL.

příklad : *adres,zip,country/NULL* zobrazí obsah zadaných polí a jako první možnost ve výběru bude *Prázdné*.

*Radiolist* slouží k reprezentaci buňky typu *enum*. V poli *Relace* musí být uvedeny všechny jeho možnosti oddělené čárkou (např. *Active,Inactive*).

### 3.5.2 Správa Akcí

Položka akce v sekci nastavení je srdcem systému. Doslova je zde uvedeno kdo co s čím. Obsahuje nástroje k určení jaká akce v jakém modulu s jakou tabulkou a pod-tabulkou se má použít. Obsah tohoto seznamu systém navíc používá ke generování cest pro navigaci a aplikaci filtrů.

- *Název* je zde jakýkoliv zvolený název akce. Měl by odrážet její účel
- *Presenter* určuje jaký presenter se má k provedení akce využít. Některé presentery obsahují modely pro speciální operace. Pokud chcete pouze upravovat data v databázi, zvolte presenter *Action*.
- *Akce* je výchozí zobrazení (viz. níže)
- *Objekt* odkazuje na název tabulky, která bude k akci použita
- *Potomek* odkazuje na název pod akce. Podrobný popis bude v kapitole Vytváření vlastních struktur.
- *Filtr* jak název napovídá, filtruje data podle zde zadané hodnoty.

### 3.5.3 Pole Akce

Presenter *Action* podporuje pět výchozích akcí presenteru. Jejich názvy a potřebné parametry jsou uvedeny v tabulce. Parametr *\$source* se doplňuje automaticky. Je jím id akce, která se právě provádí. Parametr *id* je potřeba uvést v odkazu. Jedná se o id řádku dat v požadované tabulce. Obecně by všechny výchozí akce měly odkazovat na default a sub akce, tedy potomci, na *show*, ale není to podmínkou. Pokud delegujete hlavní akci jako *show*, nebo *edit*, nedostane se jí parametru *Id*, neboť systém nebude vědět, kterou položku seznamu má editovat.

Akce	Parametr	Potřebná práva	Komentář
Default	source	r	výchozí zobrazení, seznam dat (tabulka, strom).
Show	source, id	r	Zobrazení vybraného záznamu
Edit	source, id	w	Formulář pro editaci vybraného záznamu
add	source	c	Formulář pro přidání záznamu
delete	source, id	d	Smazání záznamu

### 3.5.4 Pole Filtr

Pole filtr slouží k filtrování záznamů. Jedná se o jednoduchý řetězec založený na logice a jedné proměnné. Tou je proměnná `%user%`, která reprezentuje přihlášeného uživatele.

Chceme-li zobrazit pouze záznamy, které uživatel sám vytvořil, použijeme proměnnou `%user%`, název sloupce v databázi a otazník:

```
create_uid = ?,%user%
```

`create_uid` zde reprezentuje sloupec, ve kterém je uloženo id uživatele, který záznam vytvořil. V případě, že by nás zajímaly výpisy, které někdo upravil, použijeme `write_uid`.

Chceme-li zobrazit zakázky ve stavu konceptu a zakázky, použijeme klíčové slovo OR a za řárky uvedeme dvě hodnoty, filtr vrátí data odpovídající jedné z podmínek:

```
state = ? OR state = ?,Koncept,Zakázka
```

Podobně lze použít i klíčové slovo AND, které vrátí pouze výsledky odpovídající oběma podmínkám.

## 3.6 Správa Menu

Položka menu v sekci nastavení reprezentuje strom navigace. Pro přehlednost je zde i zobrazení jako stromu. Každá položka, která nemá pouze charakter skupiny, nebo pohledu obsahuje odkaz na akci, která se má vykonat. Při prohlížení má správce k dispozici také náhled vygenerovaného odkazu.

## 4 VYTVÁŘENÍ VLASTNÍCH STRUKTUR

System je schopen s pomocí dříve uvedených nástrojů vytvořit vlastní struktury a akce. Omezením je, že tyto nové struktury musí odrážet skutečné struktury v databázi. Bez znalostí principu relačních databází, nebo bez přístupu do ní mohou vést následující kroky k nestabilitě systému. V následujícím návodu si projdeme postup při vytváření sekce s vlastními zakázkami, kterou pojmenujeme *Mé zakázky*.

### 4.1 Vytvoření tabulky

Pokud nechceme měnit strukturu formuláře a výpisu, není tento ani následující krok nutný. Stejného výsledku lze dosáhnout i bez nich. Postup chce hlavně ukázat jak vytvořit vlastní pohled a možnosti, které z toho vyplývají

V Menu v sekci nastavení vyberte volbu Tabulky. Klikněte na tlačítko přidat a do pole Název zadejte *Mé zakázky*, do pole objekt *sale* a do pole typ zadejte *list*. Záznam uložte.

### 4.2 Vytvoření polí

V menu v sekci nastavení vyberte volbu Pole. Zde je vhodné si odfiltrovat všechna pole, která patří k tabulce *zakázky* a opsat si, nebo vytisknout hodnoty. Celý krok spočívá ve vytvoření vlastních polí identických s poli patřící tabulce *zakázky*. Pokud jste tabulku nevytvářely, tento krok můžete přeskočit. Vytvářená pole lze pojmenovat, jakkoliv. Je třeba dbát na přesné opsání hodnot jednotlivých polí. Každé vytvořené pole je nutné uložit.

### 4.3 Vytvoření Akce

V menu v sekci nastavení vyberte volbu akce. Do pole Název zadejte *Mé Zakázky*, jako presenter uveďte *sale*, aby byla zachována funkčnost modulu, akce bude *default* a objektem bude buď námi vytvořená tabulka *Mé zakázky*, pokud jsme postupovali podle bodu 4.1 a 4.2, nebo tabulka *Zakázky*. Jako podseznam uveďte *Řádek objednávky*, aby bylo možné upravovat i položky objednávky, nebo si podle tohoto postupu vytvořte vlastního podle bodu 4.1.

Do Pole filtr zadejte následující: `create_uid = ?,%user%`. Tímto se omezí výpis pouze na zakázky vytvořené právě přihlášeným uživatelem.

#### 4.4 Vytvoření položky Menu

Nakonec je nutné vytvořit v menu položku, která bude na akci odkazovat. V menu v sekci nastavení vyberte volbu Menu. Klikněte na tlačítko přidat a do pole Položka Menu zadejte *Mé zakázky*, jako nadřazenou položku vyberte *Prodej* a jako akci opět *Mé zakázky*.

#### 4.5 Závěrečná nastavení

Tímto jste se seznámili s možnostmi nastavení a vytvořily vlastní sekci, kde obchodník uvidí pouze své zakázky. Systém během těchto operací vygeneroval zdroje pro oprávnění a garantoval přístupy roli admin. Aby se nová sekce stala přístupná i jiným rolím, je nutné jim v sekci Přístupová práva doplnit patřičná oprávnění. Vzhledem k tomu, že se oprávnění vztahují na tabulky a jejich pole, pokud jste přeskočili body 4.1 a 4.2 sdílí nová sekce oprávnění se sekci Zakázky podle pravidla stejná tabulka (pole), stejná úroveň práv. Pro přidělení jiné úrovně práv jiné roli, je nutné vytvořit nové tabulky a pole.

## **P IV API DOKUMENTACE SYSTÉMU**

(Příloha dostupná na přiloženém DVD)

## P V OBSAH PŘILOŽENÉHO DVD

Přiložené DVD obsahuje adresářovou strukturu, která obsahuje tyto složky:

vivid	Předinstalovaná verze virtuálního stroje pro emulátor VMware
system	Složka se zdrojovými soubory systému.
database	Složka s exportem struktury databáze a základních dat
api	Složka s API dokumentací systému ve formátu HTML (příloha P IV).
tool	aplikace Wamp, pro instalaci serveru Apache 2.4.9, databáze MySql 5.6.17 a PHP 5.5.12.
document	elektronická verze bakalářské práce ve formátech PDF a Ms WORD.