

2009

Grafický a programový návrh  
informačního systému

Autor: Bc. Miroslav Pavlíček

3.1.2009

Vedoucí DP: Ing. Pavel Šimek, Ph.D.

# Čestné prohlášení o samostatném vypracování DP

---

Prohlašuji, že jsem diplomovou práci na téma:

**" *Grafický a programový návrh informačního systému* "**

vypracoval samostatně, pod odborným dohledem vedoucího diplomové práce a s použitím pramenů uvedených v příloženém seznamu použité literatury.

V Libčicích nad Vltavou dne 30. března 2009

.....  
Podpis řešitele

# Poděkování

---

Chtěl bych touto cestou poděkovat panu Ing. Pavlu Šimkovi Ph.D., za podporu a odborné vedení v průběhu tvorby této diplomové práce.

# Grafický a programový návrh informačního systému

## Graphical and Program Design of Information system

### Souhrn

Tato diplomové práce se věnuje problematice informačních systémů ve zdravotnictví. Zabývá se návrhem, vytvořením, testováním a implementací informačního systému. Tento informační systém je využit k udržování informací souvisejícími s agendou objednávek, smluv a dalších potřeb Dopravního zdravotnictví a.s. Diplomová práce se zabývá grafickým a ergonomickým návrhem a možnostmi využití moderních technologií při udržení minimálních nákladů na zprovoznění a údržbu. Práce obsahuje kompletní zdrojový kód funkční aplikace.

### Summary

This diploma work is showing problems of information system in medical care. This work has been considered of project, program, test and implementation of information system. This information system will be use for storing information about orders, contracts and others demands of Dopravni zdravotnictvi a.s. The diplomas work also showing graphical and ergonomic design and the work using modern technologies and stay on low cost for using and servicing. The diploma work is containing complete source code of application.

### Klíčová slova

Webdesign, internet, prohlížeč, stránky, informační systém, objednávky, PHP, CSS, databáze

### Key words

WebDesign, internet, browser, pages, information system, PHP, CSS, pictures, analysis

# 1 Obsah

Souhrn.....	4
Summary.....	4
Klíčová slova.....	4
Key words .....	4
1 Obsah.....	5
1 Úvod.....	9
2 Cíl práce a metodika .....	11
3 Hardwarová a softwarová problematika.....	13
3.1 Použité nástroje a prostředky.....	13
3.1.1 Apache server .....	13
3.1.2 PHP skriptovací jazyk .....	13
3.2 Programovací nástroje.....	14
3.2.1 phpMyAdmin .....	14
3.2.2 MySQL Workbench .....	14
3.3 Hardware .....	15
3.3.1 Webový server.....	15
3.3.2 Databázový server .....	15
3.4 Tvorba informačního systému .....	16
3.5 Programování.....	16
3.5.1 Třída main.....	16
3.6 Bezpečnost informačního systému.....	17
3.6.1 Uživatelské role .....	18
3.7 Přihlašování do informačního systému .....	19
3.8 Generování čísla objednávky .....	20

3.9	Import položek.....	21
4	Nároky na informační systém.....	22
4.1	Funkce systému .....	22
4.2	Další možnost systému .....	22
4.2.1	Evidence příchozích faktur .....	22
4.2.2	Evidence pošty.....	23
4.2.3	Digitalizace smluv .....	23
4.3	Časová náročnost vytvoření a implementace.....	23
4.3.1	Seznam úkolů, při tvorbě informačního systému. ....	24
5	Analýza potřeb uživatelů .....	25
5.1	Dopravní zdravotnictví a.s. ....	25
5.1.1	Nemocnice s poliklinikou Praha, Italská .....	25
5.1.2	Poliklinika Plzeň .....	25
5.1.3	Poliklinika Nymburk.....	26
5.1.4	Poliklinika Česká Třebová .....	26
5.1.5	Poliklinika Olomouc .....	26
5.1.6	Poliklinik Ostrava .....	26
5.2	Objednávání.....	26
5.3	Původní způsob objednávání.....	26
5.4	Nový způsob objednávání.....	27
5.4.1	Číslo objednávky.....	27
5.4.2	Schválení objednávky .....	28
5.4.3	Aktivace objednávky.....	28
6	Návrh řešení .....	29
6.1	Klíčové vlastnosti návrhu .....	29

6.1.1	Náklady na pořízení .....	29
6.1.2	Maximální jednoduchost ovládání .....	29
6.1.3	Možnost rozšíření a navázání na další systémy.....	29
6.1.4	Rychlost realizace a čas spuštění.....	29
6.2	Pořízení informačního systému .....	30
6.2.1	Nákup nového systému .....	30
6.2.2	Helios .....	30
6.2.3	Helios závěr.....	33
6.2.4	SAP .....	33
6.3	Vlastní tvorba.....	35
6.4	Grafický návrh informačního systému.....	35
6.4.1	Prvotní návrh .....	36
6.4.2	Finální podoba .....	37
6.5	Návrh jednotlivých modulů .....	38
6.5.1	Vytvoření objednávky.....	38
6.5.2	Seznam objednávek.....	40
6.5.3	Schvalování objednávek .....	42
6.5.4	Aktivace objednávek.....	43
6.5.5	Sklad.....	45
6.5.6	Přidat nového uživatele .....	49
6.6	Instalace systému u uživatelů .....	50
7	Testování systému .....	51
7.1	Alfa - testování .....	51
7.1.1	Průběh testování .....	51
7.2	Beta – testování .....	51

7.2.1	Školení .....	52
7.2.2	Problémy při beta-testování.....	52
7.3	Testování v internetových prohlížečích .....	52
7.3.1	Internet explorer 8 a 7.....	53
7.3.2	Opera 10 a Firefox 3 .....	54
7.3.3	Google Chrome 2.....	55
7.4	Testování rychlosti.....	55
8	Závěr .....	57
9	Seznam literatury.....	58
10	Přílohy.....	60
10.1	Seznam příloh.....	60
10.2	Seznam obrázků .....	60
10.3	Příloha A – Schéma databáze – vnitřní systém .....	62
10.4	Příloha B – Ganttův diagram – výstup MS Project .....	63
10.5	Příloha C – Zdrojové kódy aplikace .....	64



# 1 Úvod

Vytvoření tohoto informačního systému bylo vedeno potřebou zdokonalit a zpřehlednit systém vytváření objednávek na lékařský a další spotřební materiál v nemocnici s poliklinikou v Praze, Italské. Vytvořený systém bude využíván nejen nemocnicí v Praze, ale i v dalších pěti poliklinikách po celé České republice. Celý tento komplex nemocničních zařízení je sdružen pod akciovou společností Dopravní zdravotnictví a.s. Informační systém byl navrhnout a celý vytvořen z důvodu nenalezení vhodného na trhu dostupného řešení, které by komplexně pokrývalo všechny funkční potřeby v nemocnici a celém Dopravním zdravotnictví a.s. Z volně dostupných produktů v českém jazyce nebyl nalezen ani jeden plně využitelný. V komerční sféře dostupných produktů jsou nástroje dostupné, ale pokrytí problematiky je jen částečné a vytváření dalších modulů pro plné pokrytí je velice nákladné.

Systém vytvořený vlastními silami plně pokrývá všechny potřeby v Dopravním zařízení a.s. Volba využití volných a volně licencovaných technologií umožňuje snížit náklady na zavedení a provoz na minimální úroveň. Informační systém je možné rychle a bez problémů upravit dle aktuální potřeby a jeho další vývoj je možné směřovat přesně na potřeby celého komplexu Dopravního zařízení a.s.

Informační systém byl vytvořen od základu jako nový systém a pro jeho vývoj byly využity moderní technologie a objektově orientované programování, které umožní nadále bez větších komplikací rozvíjet celý systém a upravovat jeho funkce. Jako technologická základna byl využit webový programovací jazyk, který je velice srozumitelný na programování a jeho využití je velice široké. Součástí předkládané Diplomové práce jsou kompletní zdrojové kódy celého informačního systému.

Jako aplikační základna byl pro informační systém použit webový server, který umožňuje velice rychlé nasazení na velkém množství zařízení, snadnou údržbu běžící aplikace a rychlou distribuci nových funkcí. Při přípravě grafické podoby, byly využity poznatky z tvorby internetových stránek, aby bylo docíleno maximální pohodlí uživatele a komfort při ovládání celé aplikace.

Dále byla snaha vytvořit webovou aplikaci plně srovnatelnou s klasickou desktopovou aplikací a to jak v komfortu práce a vizuálním prostředí tak i při rychlosti zpracování informací, při využití co nejjednodušších funkcí s minimálním počtem aplikací na straně klienta.

## 2 Cíl práce a metodika

Cílem předkládané diplomové práce je analyzovat problematiku objednávání lékařského a spotřebního materiálu v Nemocnici s poliklinikou v Praze, Italská a potažmo celém Dopravním zdravotnictví a.s. Navrhovaný informační systém by měl umožňovat jednoduché a přehledné objednávání pro jeho uživatele a poskytovat výstupy pro manažery, kteří tak budou mít přehled o objednávání nejen jednotlivých lékařů, ale i dalších pracovníků a celých oddělení. Informační systém umožní perfektní řízení procesu objednávání, schvalování objednávek a jejich vyřízení. Informační systém současně zrychlí celý proces objednávání a zmenší počet chyb uživatelé při tvorbě objednávky. Informační systém poskytne co největší uživatelský komfort a příjemné uživatelské prostředí jak pro jednotlivé uživatele, tak pro vrcholový management. Další funkcí informačního systému je centralizace a sjednocení celého systému objednávání. Navržený informační systém by měl být maximálně přístupný a jeho prostředí by nemělo být omezeno žádným operačním systémem ani použitým zařízením. V neposlední řadě je cílem návrhu systému ušetření maximálního množství nákladů a jeho snadná a levná implementace a rozšiřitelnost.

V prvním kroku je nutné analyzovat celý stávající „papírový“ objednávkový systém v Dopravním zdravotnictví a připravit postup převedení původního objednávkového systému do podoby elektronické verze. Celý nový systém je sestaven tak, aby přesně odpovídal požadavkům všech jeho uživatelů a vedení. Vlastní informační systém je vybudován na co nejdostupnější programové základně. Při tvorbě informačního systému je zvažena možnost využít open source<sup>1</sup> řešení, případně využití nějakých komerčně dostupných aplikací. Současně je nutné vzít v úvahu, že výstupy by taktéž měly být přístupné maximálnímu množství uživatelů. Jako ideální se jeví využití HTML kódu ve spojení s libovolného internetového prohlížeče. Pro maximální kompatibilitu se všemi internetovými prohlížeči by měly být splněny standardy W3C(1) konsorcia.

Vlastní realizace změny celého informačního systému se skládá z několika dílčích částí. V první části jsou posbírána a dále analyzována dostupná data a informace pro vytvoření dostatečně velké informační základny pro vlastní návrh aplikace. V další části je příprava informačního návrhu z dostupných informací a navržení a sestavení datové základny pro

---

<sup>1</sup> Open source – programy s volně šiřitelným kódem (12)

provoz informačního systému. Po navržení datové základny je navrženo základní programové schéma a z něj vytvořena základna vlastního programu. Při vytváření základního programu je proveden test datové základny a funkční aplikace. Spolu s vyhodnocením prvního testování je upraven základní programový návrh vytvořena druhá testovací verze pro testování grafické podoby GUI.<sup>2</sup> Druhé kolo testů bude uzavřeno finální verzí informačního systému, která bude předložena ke schválení managementu společnosti. Následně, po schválení systému, bude provedena finální implementace a zavedení uživatelů.

---

<sup>2</sup> GUI – graphic user interface (grafické uživatelské rozhraní)

## 3 Hardwarová a softwarová problematika

V této části je probrána problematika hardwaru, kde bude informační systém provozován a na jaké softwarové základně bude nainstalován. Při vytváření je čerpáno z existujících zdrojů a existujícího hardwaru.

### 3.1 Použité nástroje a prostředky

Webový server Apache (<http://httpd.apache.org/>), skriptovací jazyk PHP 5.2 (<http://php.net/>), databázový server MySQL 5 (6a) (<http://mysql.com/>).

#### 3.1.1 Apache server

Apache HTTP Server je softwarový webový server<sup>3</sup> s otevřeným kódem pro Linux, BSD, Microsoft Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek. (2)

Pro potřeby informačního systému byl webový server nastaven tak, aby dokázal zpracovat .htaccess soubor a modul mod\_rewrite, který se stará o přesměrování na webovém serveru.

#### 3.1.2 PHP skriptovací jazyk

PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, „PHP: Hypertextový preprocesor“, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což lze využít při tvorbě webových aplikací. PHP je možné použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou většinou prováděny na straně serveru, k uživateli je přenášén až výsledek jejich činnosti (interpret PHP skriptu je možné volat pomocí příkazové řádky). Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez větších úprav na mnoha různých operačních systémech. Podporuje mnoho knihoven pro různé účely - např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. MySQL, ODBC, Oracle,

---

<sup>3</sup> Počítač, který je odpovědný za vyřizování požadavků HTTP od klientů - programů zvaných webový prohlížeč. Vyřízením požadavků se rozumí odeslání webové stránky. Webové stránky jsou obvykle dokumenty HTML.

PostgreSQL, MSSQL) a dále podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP...)

Pomocí technologie PHP je naprogramovaná Wikipedie.(3)

## 3.2 Programovací nástroje

Pro vlastní naprogramování je využít volně dostupné vývojové prostředí NetBeans 6.5 ve verzi pro PHP. (<http://www.netbeans.org/>) Pro vytvoření a kompletní zprávu databáze jsou využity open source nástroje

- phpMyAdmin ([http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)),
- MySQL QueryBrowser (<http://mysql.com>)
- MySQL Workbench (<http://dev.mysql.com/downloads/workbench/5.0.html>)

Pro vytvoření grafické podoby byla využita trial verze programu Adobe Photoshop CS 4 (<http://www.adobe.com/products/photoshop/family/?promoid=BPDEK>).

### 3.2.1 phpMyAdmin

phpMyAdmin je nástroj napsaný v jazyce PHP umožňující jednoduchou správu obsahu databáze MySQL prostřednictvím webového rozhraní. V současné době umožňuje vytvářet/rušit databáze, vytvářet/upravovat/rušit tabulky, provádět SQL příkazy a spravovat klíče. Jedná se o jeden z nejpoužívanějších nástrojů pro správu databáze. (4)

### 3.2.2 MySQL Workbench

MySQL Workbench je nová generace aplikace pro vizuální návrh databáze, které může být využito pro efektivní a dokonalý návrh, zprávu a dokumentaci databázových schémat.

Tato aplikace je dostupná jak v komerční tak i v open source verzi. Komunitní verze (OSS) je přístupná pod GNU.

Tato aplikace funguje jak pod Windows, tak jako beta pod Linux a MacOS.

Verze Windows potřebuje .NET 2.0 framework pro vlastní funkci.

### 3.3 Hardware

Pro provoz informačního systému je zapotřebí server, který bude provozovat službu internetového serveru a dále server, který bude provozovat databázový server. Pro snížení zátěže a zvýšení výkonu aplikace je rozdělen aplikační a databázový server. Tyto servery jsou napojeny do páteřní sítě v Nemocnici s poliklinikou Praha Italská. Připojení do vnitřní sítě je realizováno pomocí 1 Gbps UTP kabelu. Všechna zařízení dopravního zdravotnictví jsou napojena do vlastní VPN WAN sítě. Uživatelé z ostatních nemocničních zařízení se připojují k serverům v pražské nemocnici.

#### 3.3.1 Webový server

Pro provoz webového serveru slouží počítač s procesorem Intel Pentium 4, taktovaný na frekvenci 3,2 GHz. Kapacita operační paměti je 2 GB DDR II. Jako operační systém je použit Microsoft Windows XP SP3. Do počítačové sítě je napojen pomocí 1 Gbps UTP kabelu a napojen přímo na páteřní rozvod v nemocnici v Praze.

Na vlastním webovém serveru jsou spuštěny dvě verze informačního systému. Jedna pro testování nových funkcí a školení nových zaměstnanců a druhá verze s ostrými daty pro vlastní provoz informačního systému.

#### 3.3.2 Databázový server

Pro databázový server byl využit počítač s procesorem CORE 2 DUO a frekvencí 2,66 GHz. Velikost operační paměti je 4 GB DDR II na frekvenci 800 MHz. Jako operační systém jsou využity Microsoft Windows XP SP 3. Na páteřní síť nemocnice v Praze je napojen 1 Gbps UTP kabelem a je umístěn přímo v centrálním uzlu sítě.

## 3.4 Tvorba informačního systému

Při vytváření informačního systému bylo zapotřebí sledovat jak uživatelské pohodlí, tak bezpečnost při vstupování nových informací do systému a s tím současně i bezpečnost celého systému, který pracuje s některými důvěrnými daty uživatelů. Mezi těmito informacemi jsou například údaje o jménu nebo osobním čísle, pod kterým je uživatel identifikován v celém Dopravním zdravotnictví a.s.

Systém pro svou práci využívá Open source databázi MySQL. Pro interní komunikaci je využit PDO<sup>4</sup> modul MySQL. Jako výchozí systém pro kódování je zvoleno kódování UTF-8. Formát tabulek byl vybrán INODB, který umožňuje dobré řízení transakcí, jež jsou pro vlastní funkci nepostradatelné. (5)

## 3.5 Programování

Aplikace je programována v PHP jazyce a celý systém byl od začátku programován objektově. Využívá maximum možností PHP v objektově orientovaném stylu programování. Hlavním důvodem pro volbu tohoto stylu byla jeho snadná rozšiřitelnost a možnost využívat již dříve naprogramované objekty. (3)

### 3.5.1 Třída main

Nejdůležitější třídou v celém projektu je třída **Main**, jejíž funkce jsou děděny do všech ostatních tříd. Tato třída se stará o komunikaci s databází, má v sobě integrované základní metody pro vykreslování formulářů a kontrolní metody pro SQL příkazy.

```
function db() {  
    $dbh = new PDO("mysql:host=$this->server;dbname=$this->database;charset=utf-8", $this->default_name, $this->default_pass,  
        array(PDO::ATTR_PERSISTENT => true));  
    $dbh->exec("SET CHARACTER SET utf8");  
    return($dbh);  
}
```

Obrázek 1 - Metoda db - komunikace s databází

Obsahuje metody pro zjištění a kontrolu IP adresy s možností pevně definovaných rozsahů přístupů adres. Ve vlastní funkci je vytvořen konstruktor, který kdykoliv je zavolán (tudiž pokaždé, kdy je nějaká stránka vygenerována) a který pomocí dalších funkcí PHP zjišťuje, zda se jedná o zabezpečenou komunikaci. V případě, že by komunikace nebyla zabezpečena, přesměruje stránky na zabezpečenou komunikaci.

---

<sup>4</sup> PHP Data Objects



```

function __construct() {
    if ($_SERVER["HTTPS"] != "on")
    {
        $adresa = $_SERVER["SERVER_NAME"];
        echo "Přesměrovávám na zabezpečenou stránku... pokud se přesměrování nepovede klikněte <a href=\"https://$adresa\">zde</a> ";
        header("Location: https://$adresa");
    }
}

```

**Obrázek 2 - Konstruktor pro přesměrování na zabezpečené stránky**

Tento celý proces má tu výhodu, že zde není nutnost jakéhokoliv zásahu a také odpadá potřeba vymýšlet jakákoliv nastavení v případě zprovoznování aplikace.

### 3.6 Bezpečnost informačního systému

Informační systém pracuje s citlivými daty. Jedná se o data typu jména nebo osobního čísla. Vzhledem k tomu, že celý informační systém pracuje na bázi webové aplikace, je zapotřebí kontrolovat uživatele, co v systému dělá a také, jaké moduly systému využívá. Dále je nutné zabezpečit komunikaci mezi serverem a jednotlivými klientskými prohlížeči. Pro zabezpečení vlastní komunikace je využit SSL protokol. (6)

Secure Sockets Layer, SSL (doslova vrstva bezpečných socketů) je protokol, resp. vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizací komunikujících stran.

Ustavení SSL spojení funguje na principu asymetrické šifry, kdy každá z komunikujících stran má dvojici šifrovacích klíčů - veřejný a soukromý. Veřejný klíč je možné zveřejnit, a pokud tímto klíčem kdokoliv zašifruje nějakou zprávu, je zajištěno, že ji bude moci rozšifrovat jen majitel použitého veřejného klíče svým soukromým klíčem.

- Ustavení SSL spojení (SSL handshake, tedy potřásání rukou) pak probíhá následovně:
- Klient pošle serveru požadavek na SSL spojení, spolu s různými doplňujícími informacemi (verze SSL, nastavení šifrování atd.).
- Server pošle klientovi odpověď na jeho požadavek, která obsahuje stejný typ informací a především certifikát serveru.
- Dle přijatého certifikátu si klient ověří autentičnost serveru. Certifikát současně obsahuje také veřejný klíč serveru.

- Na základě dosud obdržených informací vygeneruje klient základ šifrovacího klíče, kterým se bude šifrovat následná komunikace. Ten zašifruje veřejným klíčem serveru a pošle mu jej.
- Server použije svůj soukromý klíč k rozšifrování základu šifrovacího klíče. Z tohoto základu vygenerují jak server, tak klient hlavní šifrovací klíč.
- Klient a server si navzájem potvrdí, že od teď bude jejich komunikace šifrovaná tímto klíčem. Fáze handshake tímto končí.
- Je ustaveno zabezpečené spojení šifrované vygenerovaným šifrovacím klíčem.
- Aplikace od tohoto okamžiku dál komunikují přes šifrované spojení. Například POST požadavek na server se do této doby neodešle.

### 3.6.1 Uživatelské role

Další položkou zabezpečení jsou uživatelské role. Informační systém má několik rolí, které umožní uživatelům pracovat jen s některými přesně vymezenými danými oblastmi informačního systému. (7)

#### 3.6.1.1 Uživatel

Primární role, kterou jsou všichni uživatelé systému, je role uživatele. Tato role obsahuje základní práva pro práci v systému. Jedná se zejména o práva **vytvoření objednávky za nákladová střediska, která spadají do přímé gesce uživatele**. Umožní vytvořit objednávku za přiřazená nákladová střediska. Tato základní práva mají sice všichni uživatelé, ale ve speciálních případech je možné toto právo pro konkrétního uživatele odepřít.

#### 3.6.1.2 Aktivátor

Role, kdy má uživatel možnost kromě vytvoření objednávky, také vytvořenou objednávku aktivovat, tj. odeslat ji dodavateli. Práva, která tato role obsahuje, jsou tedy jednak **vytvoření objednávky za svá nákladová střediska** a dále **aktivace a vyřízení objednávky za tato střediska**. Tito uživatelé jsou většinou pracovníci skladů. Tito mají navíc práva, která umožňují tzv. **Plnění a schvalování skladu**, tj. právo, které umožňuje přidávat nové položky do zboží a také upravovat již zadané položky pomocí speciálního modulu.

#### 3.6.1.3 Vedoucí / manažer

Tato role je primárně určena pro uživatele s právy na schvalování objednávek. Obsahuje v sobě sice roli uživatele, ale ve většině případů již ne roli aktivátora. Uživatelé informačního systému s touto rolí mají práva na **vytvoření objednávky za svá nákladová**

**střediska** a také právo **schválit objednávku za svá nákladová střediska**. Těmito uživateli jsou především vrcholový management a dále vedoucí jednotlivých oddělení.

#### **3.6.1.4 Ředitel**

Tato role obsahuje roli vedoucí / manažer, ale navíc rozšířenou o možnost schvalování bez limitní částky. Tato role je málo využívaná, jelikož ostatní manažeři mají dostatečně vysoké částky na schvalování a tato role není potřeba využít.

#### **3.6.1.5 Administrátor**

Role s nejvyšším množstvím funkcí. Obsahuje roli aktivátora rozšířenou o možnost **Přidání uživatelů** a **přidělování práv uživatelům, práce s nákladovými středisky**. Tato role nemá právo schvalování.

#### **3.6.1.6 Superadmin**

Role, kterou má tvůrce tohoto systému. Tato role umožňuje použít všechny funkce, ale pouze na speciálním (testovacím) nákladovém středisku, aby nedošlo k ohrožení reálných dat.

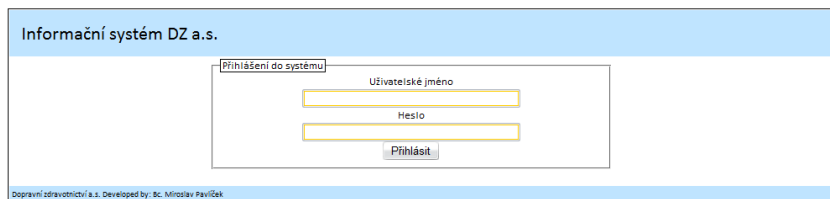
### **3.7 Přihlašování do informačního systému**

V rámci bezpečnosti bylo vytvořeno přihlašování do informačního systému za pomoci kontroly uživatelského jména a hesla, to znamená, že při spuštění informačního systému je na uživateli vyžadováno přihlašovací jméno a heslo. Pokud toto jméno a heslo není v příslušné databázi nalezeno, systém odmítne uživatele vpustit.

Ve vlastní databázi je heslo uloženo v hashované podobě, takže ani při případné chybě nemůže být toto heslo odhaleno. Heslo je zašifrováno za pomoci hashovacího<sup>5</sup> algoritmu SHA-1. Tento algoritmus byl využit z toho důvodu, že hojně využívaný hashovací algoritmus MD-5 je již prolomen.

---

<sup>5</sup> Jednosměrné zakódování



Obrázek 3 - přihlašovací okno

Při vlastním přihlášení je nejdříve zkontrolováno jméno. Pokud existuje, je zadané heslo zakódováno a výsledný otisk je kontrolován s otiskem v databázi.

### 3.8 Generování čísla objednávky

System je nucen generovat číslo objednávky dle předem dohodnutého standardu. Zde se jedná se o dvě čísla identifikující rok, dále číslo nákladového střediska a nakonec stoupající řada čísel od jedničky. System z vlastního principu využívá dvě databázové tabulky pro vytvoření objednávky. Jedna tabulka ukládá rozpracované objednávky a druhá objednávky již hotové. System prochází obě tabulky a hledá největší číslo, které je v obou tabulkách uloženo. Je nutné prověřit data v obou tabulkách, aby nedošlo ke shodě objednávkových čísel mezi rozpracovanými a hotovými objednávkami.

```

] function genInterniCislo() {
    $st = new Strediska;
    $data = $st->getRow("$_SESSION[stredisko]");
    $kod = $data[0][2];
    //echo $kod;
    $za = new Zarizeni;
    // $kod = $za->getColKod($kod);
    $rok = date("Y");
    $cislo = $this->getRowLastId();
    $cislo = substr("$cislo",-6,6);
    $temp = new tempObjednavky;
    $tempcislo = $temp->getLastId();
    $tempcislo = substr("$tempcislo",-6,6);
    if ($tempcislo > $cislo)
    $cislo = $tempcislo;
    if (!$cislo)
    $cislo = "0";

    $cislo = $cislo+1;

    $cislo = str_pad("$cislo",6,"0",STR_PAD_LEFT);
    //echo "$rok$kod$cislo";
    return("$rok$kod$cislo");
}

] function getRowLastId(){
    $sql = $this->prp("SELECT internicislo FROM objednavky ORDER BY idobjednavky DESC LIMIT 1");
    $sql->execute();
    $this->sqlchyba($sql);
    return($sql->fetchColumn());
}

```

Obrázek 4 - Kód pro generování objednávkového čísla

System využívá dvě metody a to metodu pro získání čísla z tabulky objednávek a získání čísla z tabulky rozpracovaných (temporally) objednávek. Tato čísla porovná a k většímu

z nich přičte jedničku. Celé číslo pak doplní o patřičný počet nul a přidá k němu číslo nákladového střediska a aktuální rok.

### 3.9 Import položek

Dalším problémem k řešení byl import jednotlivých položek zboží do informačního systému. Vzhledem k nedostatečné znalosti lékařského materiálu byly do katalogu informačního systému naimportovány celé katalogy dodavatelů. Tím vznikl problém duplicit u některých položek zboží a tudíž i zbytečné importace některých položek.

Na druhou stranu vznikl opačný nedostatek - některé z položek importovány nebyly, neboť dodavatel nebyl schopen dodat elektronickou verzi svého katalogu. Ruční přepisování papírových katalogů nemá téměř cenu z důvodu značné nákladovosti tohoto procesu. S vyřešením této záležitosti pomohli sami zaměstnanci, kteří dodali seznamy komodit, které nejvíce objednávají a pomohli tak doplnit seznamy o chybějící potřebné položky. Jako velmi přínosná se ukázala spolupráce s pracovníci skladu, která díky dlouholeté praxi se starým systémem věděla, co se objednává a jakému dodavateli se při objednávání konkrétních položek dává přednost.

Dalším nedostatkem bylo nejednotné značení objednávaného zboží, rozdílnost popisů a dodávané množství v balení. Nejvíce odlišností mezi dodavateli má v tomto směru množství. Některý dodavatel uvádí např. cenu za kus, jiný za sto kusů a tak dříve docházelo k chaotickému, problematickému a neekonomickému způsobu objednávání. Položky, které byly tímto nedostatkem dotčeny, byly buď upraveny, nebo na ně bylo upozorněno. V tuto chvíli v systému běží automatický proces, který hlásí případný problém při objednávání množství.

Další, co bylo nutné při zavádění systému vyřešit, bylo objednávání náplní do tiskáren a kopírek. Málokterý uživatel si je jistý typem náplně a tiskárny. Tato záležitost byla podchycena úpravou poznámek u jednotlivých náplní, kde byly doplněny identifikátory jednotlivých tiskáren, tzn. pro jakou tiskárnu náplň vlastně je a také do jaké tiskárny se příslušná náplň hodí. Současně byl přidán i popis toho, co uživatel na tiskárně může vidět napsáno a tím pádem byly maximálně eliminovány chyby při objednávání odpovídajícího spotřebního materiálu.

## **4 Nároky na informační systém**

Vzhledem k tomu, že celý informační systém bude využívat více jak 500 uživatelů a bude funkční po celé České Republice, je velmi důležitá okamžitá synchronizace a minimální zátěž uživatelského systému. Další podmínkou je, aby se uživatelé v systému jednoduše orientovali a systém reagoval na jejich požadavky bez zpoždění. Je také nutné, aby uživatelé měli přehled o tom, co v systému dělali. Současně je třeba, aby i příslušní manažeři měli možnost v tomto systému sledovat aktuální stav objednávek a měli tak současně kontrolu nejen běhu a objemu objednávání, ale i dostupné ekonomické údaje.

Celý informační systém by měl být platformě nezávislý a neměl by požadovat instalaci dalších programů nebo podpůrných komponent.

### **4.1 Funkce systému**

Celý informační systém bude sledovat kompletní proces objednání, od vytvoření objednávky přes její schválení až po realizaci. Toto je primární funkce a systém byl pro tuto funkci prvotně naprogramován.

### **4.2 Další možnost systému**

Systém je doplněn o podporu digitalizace. Je to nástroj, který umožní nahrávat do systému potřebné soubory a pomocí uživatelských rolí je distribuovat oprávněným uživatelům. Při plném využití této podpory je možné odbourat více než 80% nákladů na kopírování a distribuci.

#### **4.2.1 Evidence příchozích faktur**

Podklad pro vytvoření objednávky je skvělým základem pro systém evidence příchozích faktur, kdy systém dokáže spárovat přijatou fakturu s objednávkou, či objednávkami. Tato funkce je použitelná pro evidenci všech příchozích faktur, a umožňuje sdílet digitalizované faktury po celém Dopravním zdravotnictví a.s., bez nutnosti tyto faktury kopírovat a rozesílat.

### **4.2.2 Evidence pošty**

Stejně jako příchozí faktury je možné evidovat i příchozí a odchozí poštu. Systém umožňuje provádět její digitalizaci a pomocí integrovaného emailového rozhraní rozesílat digitální podobu na emaily.

### **4.2.3 Digitalizace smluv**

Systém podpory digitalizace je následně možné použít také pro archivaci a distribuci smluv. Tento systém, spojený se systémem pravomocí a přidělování uživatelských rolí, může distribuovat informace o smlouvách nebo i smlouvy samotné.

## **4.3 Časová náročnost vytvoření a implementace**

Pro zobrazení vývoje celé aplikace byla autorem zvolena práce MS Project a jeho interpretace ganttova diagramu, který přehledně zobrazí, jaké práce byly na systému vytvářeny a jak dlouho tyto práce trvaly. Dále je zde vidět odlišná náročnost programování jednotlivých modulů. Z celkového pohledu zabralo poměrně značnou část doby vkládání nových dat do již hotového informačního systému.

### 4.3.1 Seznam úkolů, při tvorbě informačního systému.

V této tabulce je seznam všech úkolů, které byly provedeny při budování informačního systému. Na prvním řádku je zobrazena celková doba práce. Zadání na vytvoření tohoto informačního systému padlo 1. 9. 2008 a celý systém byl spuštěn do plného provozu 26. 2. 2009.

Tabulka 1- Seznam úkolů

Informační systém	129 dní	1. 9. 08	26. 2. 09
Analýza potřeb informačního systému v DZ a.s.	6 dní	1. 9. 08	8. 9. 08
Příprava a návrh projektu	5 dní	9. 9. 08	15. 9. 08
Konzultace s managementem a účtárnou	3 dní	16. 9. 08	18. 9. 08
Úprava projektu dle připomínek	1 dní	19. 9. 08	19. 9. 08
Vytvoření grafického návrhu	1 dní	16. 9. 08	16. 9. 08
Návrh vlastní databáze informačního systému	3 dní	22. 9. 08	24. 9. 08
Seznam modulů	3 dní	25. 9. 08	29. 9. 08
Hlavní modul	7 dní	30. 9. 08	8. 10. 08
Modul zabezpečení	11 dní	9. 10. 08	23. 10. 08
Modul vytváření objednávek	13 dní	24. 10. 08	11. 11. 08
Modul schvalování objednávek	6 dní	12. 11. 08	19. 11. 08
Modul aktivace objednávek	12 dní	20. 11. 08	5. 12. 08
Testování celého systému s testovacími daty	4 dní	8. 12. 08	11. 12. 08
Implementace na server	2 dní	12. 12. 08	15. 12. 08
Testování běhu na serveru	4 dní	16. 12. 08	19. 12. 08
Plnění reálnými daty	20 dní	22. 12. 08	16. 1. 09
Beta - testování	14 dní	19. 1. 09	5. 2. 09
Školení uživatelů	4 dní	19. 1. 09	22. 1. 09
Zařazení do ostrého provozu	1 dní	6. 2. 09	6. 2. 09
Zpracování dokumentace a návodu	14 dní	9. 2. 09	26. 2. 09



Celé budování systému včetně hlavních úkolů je přehledně zobrazeno na výstupech z programu MS Project. Celý ganttův diagram je uveden v příloze B této diplomové práce.

## 5 Analýza potřeb uživatelů

### 5.1 Dopravní zdravotnictví a.s.

Dopravní zdravotnictví a.s. je akciová společnost 100% ve vlastnictví firmy AGEL a.s. Tato síť pronajala od Českých Draž síť nemocnic a poliklinik, které byly sdruženy jako část Českých Draž pod hlavičkou Železniční zdraví. Dopravní zdravotnictví a.s. má od Českých Draž pronajata tato zařízení po celé České republice.

- Nemocnici s poliklinikou Praha Italská (<http://www.dopravnizdravotnictvi.cz/zn-praha/obsah/default.aspx>)
- Poliklinika Plzeň (<http://www.dopravnizdravotnictvi.cz/zp-plzen/obsah/default.aspx>)
- Poliklinika Nymburk (<http://www.dopravnizdravotnictvi.cz/zp-nymburk/obsah/default.aspx>)
- Poliklinika Česká Třebová (<http://www.dopravnizdravotnictvi.cz/zp-ceskatrebova/obsah/default.aspx>)
- Poliklinika Olomouc a její detašované pracoviště Poliklinika Přerov (<http://www.dopravnizdravotnictvi.cz/zp-olomouc/>)
- Poliklinika Ostrava (<http://www.dopravnizdravotnictvi.cz/zp-ostrava/>)

#### 5.1.1 Nemocnice s poliklinikou Praha, Italská

Jediná nemocnice v Dopravním zdravotnictví a.s. Nemocnice zaměstnává zhruba 140 pracovníků. Z tohoto počtu bude využívat Informační systém zhruba 120 zaměstnanců a to zejména lékaři a sestry. Každý z těchto uživatelů je vybaven počítačem.

#### 5.1.2 Poliklinika Plzeň

Největší ze zařízení v Dopravním zdravotnictví a.s. Je zde zhruba 200 zaměstnanců. Každý uživatel je vybaven počítačem.

### **5.1.3 Poliklinika Nymburk**

Nejmenší ze všech zařízení v Dopravním zdravotnictví a.s. V poliklinice je 30 zaměstnanců, kteří budou tento informační systém využívat.

### **5.1.4 Poliklinika Česká Třebová**

Poliklinika má zhruba 40 zaměstnanců, využívat jej budou všichni.

### **5.1.5 Poliklinika Olomouc**

Největší moravská poliklinika, díky přidružení Polikliniky Přerov. Je zde zaměstnáno 110 zaměstnanců.

### **5.1.6 Poliklinik Ostrava**

Poslední z poliklinik, která má zhruba 60 zaměstnanců. V tomto zařízení nejsou využívány plnohodnotné počítače, ale pouze počítačové terminály.

## **5.2 Objednávání**

Prvním požadavkem bylo vytvořit systém pro kontrolu a zpřehlednění objednávek. Tyto objednávky slouží pro nákup materiálu a zařízení, nutných pro funkci jednotlivých oddělení a lékařských ordinací. Každý příslušný zaměstnanec má své odpovídající nákladové středisko, na které jsou tyto náklady účtovány. Systém by měl objednávatí potřebných komodit urychlit, zpřehlednit a hlavně umožnit kontrolu celé agendy.

## **5.3 Původní způsob objednávání**

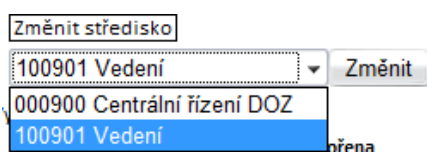
Původní "informační systém" vycházel z lety prověřeného způsobu vytváření objednávek pomocí papírových objednávkových formulářů. Tyto formuláře musel každý vyplnit, potvrdit pomocí razítka a podpisu. Tento formulář byl následně předán nadřízenému, který měl oprávnění danou objednávku schválit. Takto schválená objednávka byla předána dalšímu vedoucímu pracovníkovi, který ji potvrdil. Potvrzená a schválená objednávka byla opět předána dál a to pověřenému pracovníkovi ke konečnému kroku, k vlastnímu objednání. Po dodání bylo zboží rozvezeno na příslušná pracoviště, která si zboží objednala. Dotyční podepsali po doručení dodávkový list (předávací protokol), který byl následně předán do účtárny.

Vzhledem k tomu, že forma úhrady takto dodaného zboží je vždy fakturou. Faktury chodí od dodavatelů později, byla tedy nutná další operativní evidence těchto dokladů. Dodavatelská faktura byla při příchodu zapsána do knihy příchozích faktur a následně byla ručně spárována se souvisejícími objednávkami a dodacími listy.

Celý tento „systém“ byl nejen velice pomalý, vyžadující značné množství administrativní práce, ale stávalo se, že zdržení objednávky, v některých případech tzv. založení nebo případná ztráta potřebných souvisejících dokladů, zkomplikovaly celý proces a zdržely i následné účtování. Sledování běhu objednávky a to jak procesně, tak i z hlediska ekonomického bylo nejen velmi pracné, ale mnohdy i velmi komplikované.

## 5.4 Nový způsob objednávání

Zrychlení, přehlednost, jednoduchost, kontrola, sdílení dat a snížení administrativní náročnosti bylo hlavním cílem. Každý zaměstnanec má jednoznačně přiděleno nákladové středisko, na které může objednávat produkty potřebné k výkonu svojí činnosti. Pro



Obrázek 5 - Možnosti volby středisek

přihlášení do informačního systému je tento systém nastaven tak, že uživateli je automaticky přiděleno jeho středisko a ten na něj objednává. Někteří

uživatelé, kteří pracují na více místech a mají proto více nákladových středisek si mohou vybrat, za které nákladové středisko budou objednávat. (Obr. 7) Změna střediska s sebou nese vytvoření nového čísla objednávky, které je přiřazeno k aktuální objednávce uživatele.

### 5.4.1 Číslo objednávky

Pro objednávání je v systému generováno jednoznačné číslo objednávky. Toto číslo je jedinečné a je sestaveno tak aby se z něj dalo určit kdy (v kterém roce) byla objednávka udělána, pro jaké nákladové středisko, kdo je objednavatelem a jaké má pořadové číslo.

09	100901	000011
Rok vytvoření objednávky	Nákladové středisko	Pořadové číslo objednávky

Obrázek 6 - Formát čísla objednávky

### 5.4.2 Schválení objednávky

Po vytvoření objednávky jde objednávka do schvalovacího procesu. Tento proces zařizují vedoucí jednotlivých oddělení nebo jejich nadřízení. Každý z vedoucích má stanovenou maximální schvalovací částku. Objednávka, nepřesahující tento limit jde přímo do procesu aktivace (objednání). Pokud je cena objednávky vyšší než maximálně stanovený limit, pokračuje ke schválení k další osobě, která má oprávnění na vyšší limit, tzn., že dotyčný pouze objednávku v systému potvrdí, tj. odsouhlasí, že objednané zboží a jeho množství je v pořádku a předá objednávku dál ke schválení odpovídající osobě nebo uživateli s manažerskými schvalovacími právy. Manažerská schvalovací práva umožňují schválit všechny objednávky bez ohledu na výši limitu. Schválení objednávky se objeví v systému jako Změna stavu v uživatelské objednávce.

### 5.4.3 Aktivace objednávky

Pokud je objednávka schválena, je zboží, které je třeba objednat, zobrazeno v aktivačním modulu a uživatelé s právy na aktivaci (reálným objednááním) mohou toto zboží objednat přímo u dodavatelů. Forma objednání může být různá, od e-shopu až po faxování objednávek. Forma objednání je zcela na dohodě mezi dodavatelem a odběratelem. Po odeslání objednávky k dodavateli je do informačního systému tento stav nastaven. Uživatel vidí, že jeho objednávka je již odeslána k dodavateli.

Při doručení a převzetí objednávky uživatel s právy aktivátora zadá do systému, které zboží přišlo, aby objednavatel věděl, že si může dodané zboží jít vyzvednout, popřípadě aby dostal informaci o dosud nedodaných produktech.

Pokud je zboží dodáno, má aktivátor možnost vytvořit tiskovou sestavu, kde jsou zobrazeny všechny důležité informace o objednávce. Tuto sestavu při předání zboží nechá podepsat a tím je potvrzeno, že příslušné zboží bylo předáno. Sestava je následně předána do účtárny.

## 6 Návrh řešení

### 6.1 Klíčové vlastnosti návrhu

Při posuzování možností pro úpravu celého systému objednávání a další důležitých informací potřebných pro digitalizaci maximálního počtu procesů, muselo být počítáno se čtyřmi následujícími kritérii. Předmětná kritéria jsou seřazena podle důležitosti.

- Finance – náklady na pořízení
- Maximální jednoduchost v ovládání a obsluze
- Možnost rozšíření a navázání na další systémy
- Rychlost realizace a čas spuštění

#### 6.1.1 Náklady na pořízení

Nejdůležitější podmínkou byla minimalizace nákladů spojená s pořízením informačního systému. Náklady by současně s vlastním nasazením systému neměly přesahovat náklady na jeho nákup nebo vybudování.

#### 6.1.2 Maximální jednoduchost ovládání

S informačním systémem bude pracovat několik set lidí. Každý z těchto uživatelů má rozdílné zkušenosti a znalosti s využíváním výpočetní techniky. Je nutné vzít také v potaz, že většina uživatelů budou lékaři pracovníci, pro které by neměla být tato činnost zbytečně časově náročná .

#### 6.1.3 Možnost rozšíření a navázání na další systémy

S postupem digitalizace je také nutné počítat s tím, že se systém bude rozvíjet, doplňovat potřebnými funkcemi a měl by současně umožnit zprostředkování komunikace mezi jednotlivými systémy.

#### 6.1.4 Rychlost realizace a čas spuštění

Posledním požadavkem bylo, aby byl projekt spuštěn v reálné době a jeho implementace nezabrala příliš mnoho času. Dále je nutné vzít také v potaz, že tento systém bude využíván každý den.

## 6.2 Pořízení informačního systému

Možnosti pořízení informačního systému byly dvě. Jednou z možností bylo využít a přizpůsobit již existující komerční řešení. Druhá varianta znamenala vytvořit informační systém přesně na míru potřebám Dopravního zdravotnictví a.s. Tato varianta měla dvě možnosti realizace. Jednak za pomoci vnitřních zdrojů nebo nechat připravit systém nějakou vývojovou firmu.

### 6.2.1 Nákup nového systému

Jako komerční řešení byly uvažovány tyto dva produkty

- Helios
- SAP

Výhodou je šířka využití možnosti systému již prověřený provoz, možnosti rozšíření, technické zázemí a minimální zatížení pracovníků IT.

Nevýhodami jsou v tomto případě veliká mohutnost systému, velký počet nevyužitých modulů a obrovské pořizovací náklady.

### 6.2.2 Helios

Informační systém HELIOS vyvíjí firma LCS International, a.s. se sídlem v Praze.

Společnost LCS je jedním ze tří nejvýznamnějších producentů podnikových aplikací (ERP) na českém trhu. Od května 2007 je členem nadnárodní skupiny ASSECO. Zabývá se vývojem, implementací a podporou specializovaných informačních systémů pro výrobu, služby a obchod. Produkty společnosti LCS – podnikové informační systémy Helios Red, Helios Orange a Helios Green – pokrývají nejširší spektrum potřeb podniků. Všechny systémy jsou doplněné širokou nabídkou služeb a partnerských programů. Společnost LCS zaměstnává 245 pracovníků v Česku, na Slovensku a v SRN. (8)

### 6.2.2.1 Helios Green

Určen pro střední a velké firmy a nadnárodní korporace. Je postaven na platformě Microsoft .NET. Jeho základní a specializované moduly tvoří neustále širokou nabídku tzv. branžových řešení.



Obrázek 7 - Helios Green schéma modulů

### 6.2.2.2 Helios Orange

Určen středně velkým a menším firmám. Vedle Ekonomické agendy zahrnuje i řízení obchodu a služeb, řízení procesů ve výrobních firmách, problematiku Dopravy a spedice, Cel, Zemědělství, CRM a další oblasti.



Obrázek 8 - Helios Orange seznam modulů

### 6.2.2.3 Helios Red

Ekonomický systém pro zpracování podnikové agendy podnikatelů, malých a středních firem využívajících účetnictví nebo daňovou evidenci.



Obrázek 9 - Helios Red seznam modulů

Při detailním zkoumání možností informačního systému Helios tak byl vybrán na testování nejsilnějšího z informačních systémů a to Helios Green.

### 6.2.2.4 Požadavky na informační systém Helios Green

#### 6.2.2.4.1 Databázový server

- Windows 2003 Server (verze podle potřebné konfigurace HW), s nainstalovaným aktuálním service packem.
- MS SQL 2005 Server (verze podle potřebné konfigurace HW), s nainstalovaným aktuálním service packem



Aplikační server (v případě Helios Green na platformě Microsoft .NET)

- Windows 2003 Server (verze podle potřebné konfigurace HW), s nainstalovaným aktuálním service packem
- IIS verze 6.0 s podporou ASP .NET a .NET Framework 2.0

#### 6.2.2.4.2 PC uživatele

- Windows XP Professional, s nainstalovaným aktuálním service packem
- Office XP nebo Office 2003 s nainstalovaným aktuálním service packem

#### 6.2.2.5 Výkonové požadavky na systém Helios Green

Typicky SQL server - pro > 50 uživatelů a součtu velikosti všech databází > 15GB doporučujeme server, ve kterém je možné osadit 4 procesory (P4 min 2GHz) a 8 GB RAM

#### 6.2.3 Helios závěr

Systém Helios má velké spektrum modulů. Tento systém je možné využít pro většinu funkcí, které jsou v Dopravním zdravotnictví a.s. potřebné. Bohužel, konkrétní modul pro interní objednávky by bylo nutné nechat dodělat a upravit dle požadavků Dopravního zdravotnictví a.s.

Při počtu zhruba 600 uživatelů by tento systém vyžadoval aplikační server velmi vysokého výkonu a i ostatní počítačové stanice by potřebovaly vyšší výkon a to by znamenalo značné náklady na upgrade stanic a datové infrastruktury, jelikož všechny stanice by musely být napojeny na velmi rychlou datovou linku. Tento systém nemá nijak vysoké pořizovací náklady, ale nutnost zaučení a školení uživatelů by tyto náklady rapidně zvýšilo.

#### 6.2.4 SAP

SAP R/3 je client/server aplikace využívající třívrstvý model. Prezentační vrstva nebo klient komunikují s uživatelem. V aplikační vrstvě je uložena business logika a databázová vrstva zaznamenává a ukládá všechna data systému včetně transakčních a konfiguračních dat.

Funkčnost systému SAP R/3 je programována vlastním proprietárním jazykem ABAP (Advanced Business Application Programming). ABAP, neboli ABAP/4, je jazykem čtvrté generace (4GL) umožňujícím vytvářet jednoduché, ale výkonné programy. R/3 obsahuje také kompletní vývojové prostředí, které umožňuje vývojářům modifikovat existující

programový kód SAPu nebo vytvářet vlastní funkčnost, od reportů až po transakční systémy, s využitím SAP frameworku. ABAP komunikuje s databází pomocí SQL dotazů, které umožňují vybírat, měnit a mazat data. Dále umožňuje vytvářet grafická uživatelská rozhraní a middleware pro integraci s jinými systémy.

Nastavení systému při zavádění je velmi složité, protože SAP R/3 je v každé společnosti nastaven jinak. Proto si společnosti najímají SAP konzultanty, kteří přizpůsobují systém potřebám dané společnosti. Některé společnosti zavádějí jen některé moduly, jiné společnosti zavádějí vše (cena licence není na počtu modulů závislá).(9)

#### **6.2.4.1 Cena systému SAP**

Jedná se o modul pro 100 uživatelů, který ale je potřeba rozšířit pro dalších 500 uživatelů.

- Licence: 6,868,700 CZK
- Služby: 3,800,000 CZK
- Servery: 180,000 CZK
- Celkem: 10,848,700 CZK

Takto vysoké náklady by se navýšily o nutnost posílení infrastruktury a klientských počítačů. Celkové náklady jsou velmi vysoké, a ačkoliv by se v systému již nemuselo dělat příliš mnoho změn, jeho nasazení by bylo velice drahé.

### **6.3 Vlastní tvorba**

Poslední možností jak vytvořit systém, je naprogramovat jej dle aktuálních potřeb žadatele. Zpracování tohoto požadavku a řešení je předmětem této diplomové práce. Systém je navržen přesně na míru Dopravního zdravotnictví a.s. a je možné jej dle potřeb firmy rychle upravit, přidat potřebné funkce, popřípadě opravit eventuelní chyby operativně bez časových prodlev a vysokých nákladů.(10)

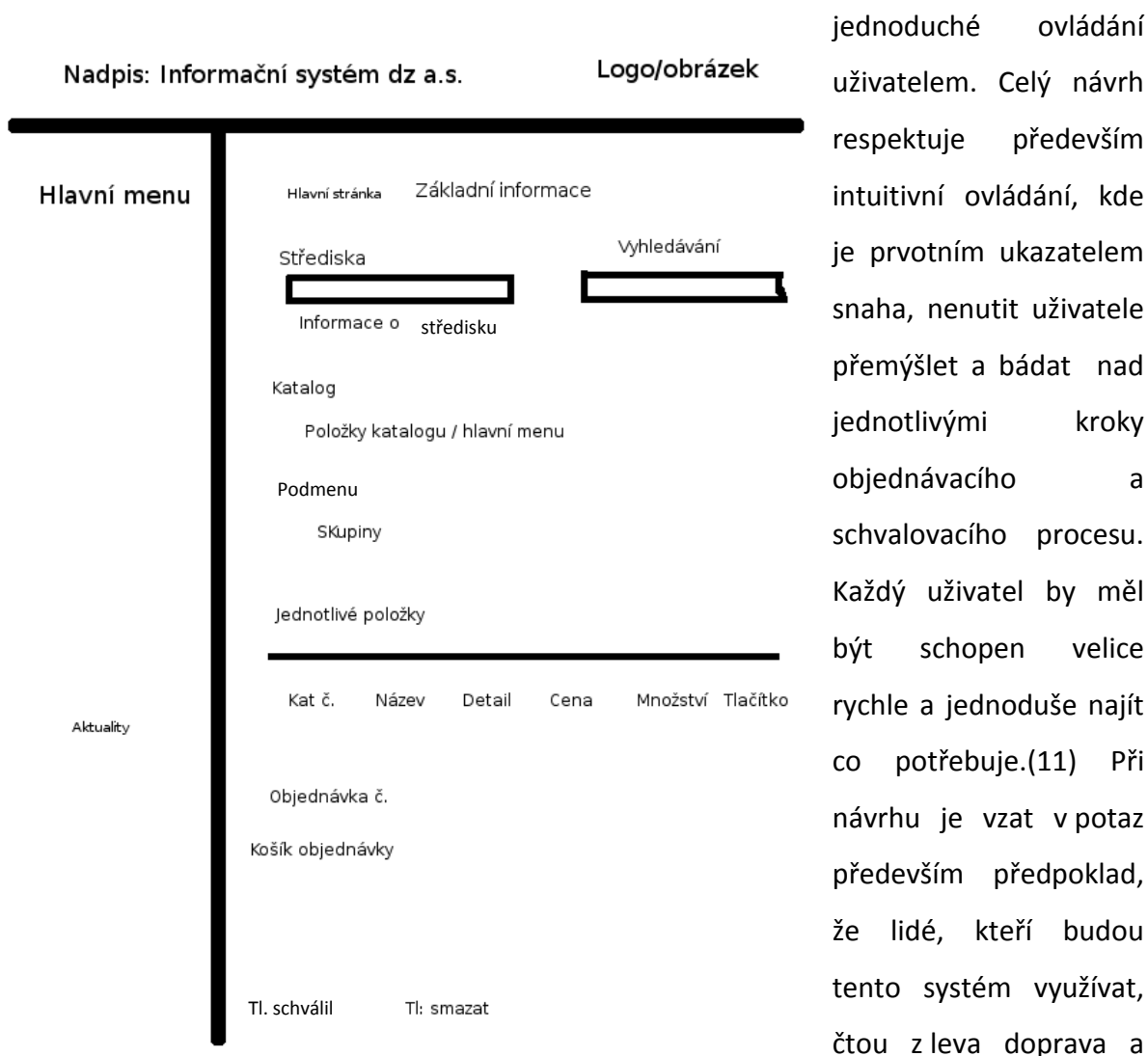
### **6.4 Grafický návrh informačního systému**

Základní vzhled celého informačního systému je jednotný. Je to dán použitou technologií a stylem komunikace. Ve specifických případech je možné vzhled upravit a přizpůsobit požadavkům jednotlivých uživatelů. Celkový grafický vzhled je navržen pro využití co největším počtem zařízení a monitorů. Technologie webových stránek umožňuje systém využít téměř na jakémkoliv zařízení, včetně mobilních telefonů a PDA zařízeních.

Při vlastní realizaci návrhu bylo maximálně využito kaskádové stylování (CSS), které umožnilo maximálně přizpůsobit grafickou stránku aplikace.

### 6.4.1 Prvotní návrh

Prvotní návrh vychází z dvousloupcového layoutu<sup>6</sup>, který je přizpůsoben pro maximálně



jednoduché ovládání uživatelem. Celý návrh respektuje především intuitivní ovládání, kde je prvotním ukazatelem snaha, nenutit uživatele přemýšlet a bádát nad jednotlivými kroky objednávacího a schvalovacího procesu. Každý uživatel by měl být schopen velice rychle a jednoduše najít co potřebuje.(11) Při návrhu je vzat v potaz především předpoklad, že lidé, kteří budou tento systém využívat, čtou zleva doprava a

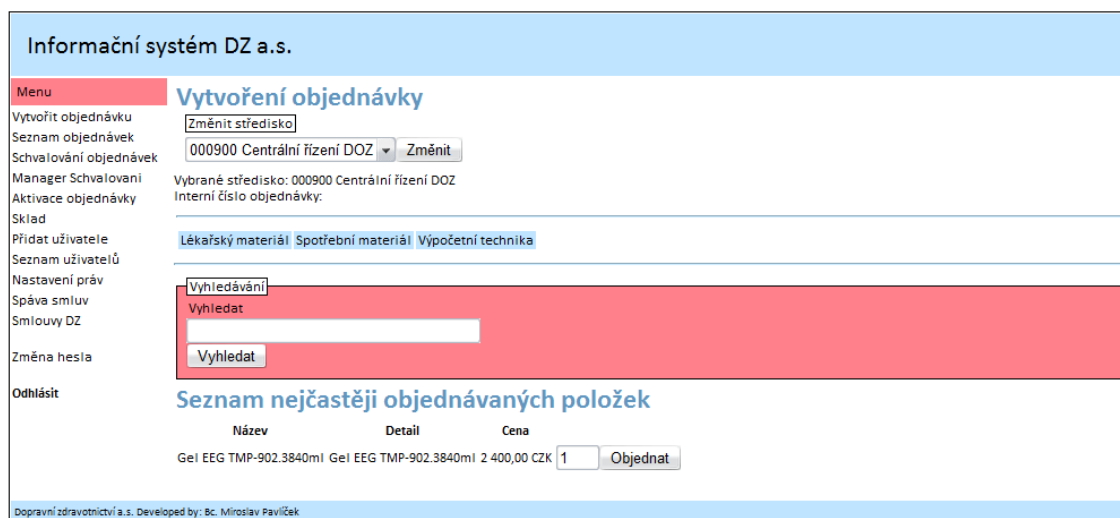
Obrázek 10 - Prvotní návrh grafické podoby

jsou zvyklí na určitý standard. To znamená, že ovládací menu bude vlevo, tj. tam, kde ho uživatelé běžně očekávají. Značným problémem bylo pak samostatné umístění vyhledávacího pole, které bylo nutné přesunout z původního místa vedle volby střediska (obrázek: Prvotní návrh grafické podoby), až pod výběr z katalogu. V tomto případě se jedná o zpříjemnění používání, protože uživatelé většinou neví, co přesně hledají nebo užívají zažité „lidové“ názvy pro věci, které potřebují, ale tyto názvy v systému vyhledávání nejsou obsaženy a vytvářet pro ně vlastní systém a jejich ukládání je neefektivní.

<sup>6</sup> Layout – vzhled, podoba stránky

## 6.4.2 Finální podoba

Po celou dobu vytváření systému byla maximální snaha držet se původního grafického návrhu informačního systému. Byly sice upraveny některé položky, ale hlavní idea zůstala plně zachována. Grafický návrh respektuje maximální přehlednost bez nutnosti dlouhého „bádání“ nad jednotlivými položkami.



Obrázek 11 - Finální grafický návrh

Barevnost byla zvolena z hlediska maximální přehlednosti a vysokého kontrastu. Velikost písma v základu 12px a bezpatkový font Calibri (Arial v případě, že font v systému není obsažen). Každá položka je vizuálně graficky oddělena, ale počet barev byl zvolen minimální.

## 6.5 Návrh jednotlivých modulů

### 6.5.1 Vytvoření objednávky

Vlastní katalog je rozdělen do tří částí. Jedná se o hlavní skupiny, které jsou pevně definovány jako lékařský materiál, spotřební materiál a výpočetní technika.

**Lékařský materiál** **Spotřební materiál** **Výpočetní technika**

Obrázek 12 - Hlavní skupiny

Po kliknutí na zvolenou hlavní skupinu se zobrazí nadskupiny

Čepelky	Dezinfekce kůže	Dezinfekce plochy a nástroje	Firma GAMA
Fotomateriál	Gely	INFÚZNÍ TECHNIKA	Jehly
Kanyly	Kuchyňské vybavení	Lukasterik	Materiál na odpady
Nádoby na kontaminovaný odpad	Oblečení,obuv	Obvazový materiál,náplasti,kompresy,vata	Operace
Ostatní	Ostatní -nezařazené	Papírové výrobky	Péče o tělo
Pervin/Perlan	Pomůcky vyšetřovací	Přizové výrobky	Rukavice
Šití	Speciální mycí přípravky	Speciální mycí přípravky	Stříkačky
Teploměry	Výrobky Bochemie	Výrobky Dahlhaussen	Výrobky Hartmann
Výrobky Schulke&Mayr	Zvláštní dezinfekce a příslušenství		

Obrázek 13 - Nadskupiny (Lékařský materiál)

a následně i jednotlivé skupiny otevřené nadskupiny.

Obturátory pro kanyly s křídélky	Obturátory pro kanyly s křídélky a portem	Obturátory pro rovné kanyly
Ostatní kanyly	SURFLASH - nitrožilní kanyla - rovná - polyuretan	SURFLO - nitrožilní kanyla - rovná - teflon
SURFLO - W nitrožilní kanyla s křídélky - teflon	SURFLO - W nitrožilní kanyla s křídélky a injekčním portem - teflon	VERSATUS - nitrožilní kanyla - polyuretan
VERSATUS - nitrožilní kanyla s křídélky a injekčním portem - polyuretan	VERSATUS - nitrožilní kanyla s křídélky- polyuretan	

Obrázek 14 - Jednotlivé skupiny (Kanyly)

Po výběru skupiny se již zobrazí vlastní seznam položek. Každá položka v katalogu má svůj název, popis, informaci o dodávaném množství a pro vlastní objednávání se využívá klasický e-shopový objednávací mechanismus, kde si uživatel vyplní počet položek, které chce objednat a klikne na tlačítko objednat, kterým provede vložení patřičného množství zboží do objednávky. Položek je na jedné stránce zobrazeno maximálně 20 a přechod mezi stránkami je vytvořen pomocí stránkovacích tlačítek.

## Seznam položek ve skupině

Kat. č.	Název	Detail	Cena		
TER1SROF1464	i.v.kanyla rovní 14G	i.v.kanyla rovní 14G	13,55 CZK	<input type="text" value="1"/>	Objednat
TER1SROF1651	i.v.kanyla rovní 16G	i.v.kanyla rovní 16G	13,55 CZK	<input type="text" value="1"/>	Objednat
TER1SROF1832	i.v.kanyla rovní 18G	i.v.kanyla rovní 18G	13,55 CZK	<input type="text" value="1"/>	Objednat
TER1SROF1851	i.v.kanyla rovní 18G	i.v.kanyla rovní 18G	13,55 CZK	<input type="text" value="1"/>	Objednat
TER1SROF2032	i.v.kanyla rovní 20G	i.v.kanyla rovní 20G	13,55 CZK	<input type="text" value="1"/>	Objednat
TER1SROF2051	i.v.kanyla rovní 20G	i.v.kanyla rovní 20G	13,55 CZK	<input type="text" value="1"/>	Objednat
TER1SROF2225	i.v.kanyla rovní 22G	i.v.kanyla rovní 22G	13,55 CZK	<input type="text" value="1"/>	Objednat
TER1SROF2419	i.v.kanyla rovní 24G	i.v.kanyla rovní 24G	13,55 CZK	<input type="text" value="1"/>	Objednat

Obrázek 15 - Seznam položek

Vlastní objednávací košík (objednávka) obsahuje seznam objednaných položek (položky nejsou závislé na dodavateli). Je zde zobrazen název, objednané množství, cena za kus a celková cena zboží. K objednávce je možné přidat patřičný komentář (poznámku), která může objednávku upřesnit, nebo lze do této poznámky přidat i další potřebné informace, jako např. i název zboží, které v systému chybí. Dále je možné vymazat položky, které v objednávce nechceme.

Sk.č.	Název	Jednotky	Množství	Cena	Celková cena	
TER1SROF2032	i.v.kanyla rovní 20G	ks	10	13,55	135,50	<a href="#">Smazat</a>
0.75356931588945	denní pracovní záznam v a.s.	ks	1	0,42	0,42	<a href="#">Smazat</a>

Poznámka k objednávce

Obrázek 16 - Objednávkový košík

Jako poslední jsou pak tlačítka k odeslání objednávky ke schválení a také k vymazání celé objednávky. (číslo objednávky je recyklováno).

## 6.5.2 Seznam objednávek

V tomto okně je možné zobrazit stav objednávek, které byly vytvořeny, které čekají na zpracování, jsou objednány, doručeny nebo zamítnuty. Pokud není objednávka schválena, je možné v ní měnit množství nebo odebírat položky. Nové položky již do objednávky přidat nelze.

### Seznam objednávek

Rozpracované objednávky Hotové objednávky

Změnit středisko

000900 Centrální řízení DOZ Změnit

Vybrané středisko: 000900 Centrální řízení DOZ

Interní číslo	Stav	Vytvořena	
<a href="#">09000900000010</a>	Objednáno	2009-04-07 18:50:41	Detail
<a href="#">09000900000009</a>	Doručeno	2009-04-07 18:50:35	Převzít Detail
<a href="#">09000900000008</a>	Přijato ke schválení	2009-04-07 18:50:30	Detail

Obrázek 17 - Seznam objednávek

U každé objednávky je možné kliknutím na interní číslo zobrazit položky, které jsou v objednávce vloženy. Zde je pak možné tyto položky měnit (obr. 8).

### Seznam objednávek

Rozpracované objednávky Hotové objednávky

#### Položky v objednávce

Bc. Miroslav Pavlíček (apofis)  
Číslo objednávky: 09000900000008

Číslo	Název	Množství	Jednotky	Cena	Celková cena		
0.49051277478273	<a href="#">dx2450 MT Athlon X2-4450B</a>	1	Změnit	ks	9 020 CZK	9 020 CZK	Odstranit

Poznámka

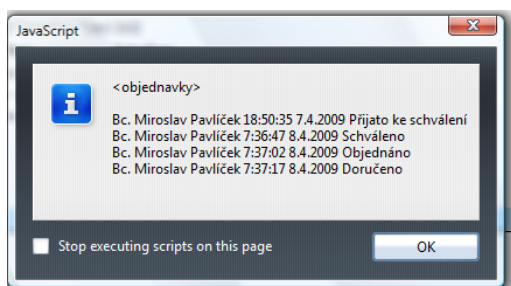
Přidat poznámku

Obrázek 18 - Úprava objednávky

Je také možné přidat k objednávce poznámku. Při kliknutí na název výrobku se zobrazí jeho popis a dodavatel.

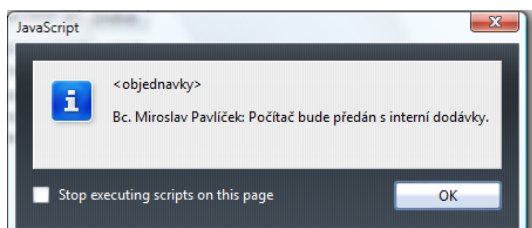


V seznamu objednávek (pak i v dalších modulech) jsou tlačítka pro zjištění detailů o stavu objednávky, kde je zobrazeno, jaké procesy již proběhly a kdo je za tyto procesy odpovědný.



**Obrázek 19 - Seznam procesů objednávky**

Další informací jsou vložené poznámky (tlačítko poznámky), kde jsou zobrazeny všechny poznámky přidáné k objednávce.



**Obrázek 20 - Poznámky k objednávce**

Jako další je funkce na elektronické (bezpapírové) převzetí objednávky. Takto převzaté objednávky se objeví ve výstupech pro účetnictví a není nutné je archivovat v papírové podobě.

V této stránce (modulu) je možné vidět seznam již dříve vytvořených objednávek, které byly objednány nebo i zrušeny. Objednávka se v tomto seznamu objeví, až v momentě, když je předána nebo když je při schvalování zamítnuta.

### 6.5.3 Schvalování objednávek

Modul schvalování slouží vedoucím jednotlivých oddělení a manažerům pro změnu stavu objednávky. Při těchto procesech jsou uživatelé limitováni částkou, kterou mohou schválit.

[Seznam na schválení](#) [Schválené objednávky](#)

**Schvalovací limit je 50 000 CZK, větší částky pouze odsouhlasíte**

Uživatel	Středisko	Datum	Interní číslo	Cena	Stav				
Bc. Miroslav Pavlíček	000900 (Centrální řízení DOZ)	7. 4. 2009	<a href="#">09000900000008</a>	9 020,00	Přijato ke schválení	<a href="#">Schválit</a>	<a href="#">Zamítnout</a>	<a href="#">Detail</a>	<a href="#">Poznámka</a>

**Obrázek 21 - Seznam objednávek na schválení**

V seznamu je zobrazen uživatel, který objednávku vytvořil a nákladové středisko, na které je objednávka směřována. Současně je vidět, kdy byla objednávka provedena a jaké je její interní objednávkové číslo. Přičemž je při kliknutí na číslo možné změnit množství nebo vyřadit jednotlivé položky. Dále je zde vidět celková cena objednávky a aktuální stav objednávky. Jednotlivá tlačítka pak slouží k operacím s celou objednávkou. Objedávka se může schválit nebo zamítnout. Při schválení se objednávka přesune do modulu aktivace. Při případném zamítnutí je objednávka zrušena a je přesunuta do zrušených (pasivních) objednávek. Kliknutím na tlačítko detail je zobrazeno, kdo objednávce změnil status a na jaký status byla objednávka změněna. Poznámka pak zobrazí detailní informace k objednávce a případné komentáře objednatele nebo schvalovatele.

## 6.5.4 Aktivace objednávek

Uživatel s rolí aktivátora má práva na odeslání objednávky dodavateli. Uživatel si vybere dodavatele, na kterého bude tvořit reálnou objednávku.

### 6.5.4.1 Nové objednávky

V této části se zobrazí seznam zboží, které bylo schváleno pro objednání u vybraného dodavatele.

---

[Nové objednávky](#) [Vytvořené objednávky](#) [Dodané objednávky](#)

**Seznam objednavek pro aktivaci**

**Dodavatelé**

Změnit dodavatele

MARTEK MEDICAL a.s.

Dopravní zdravotnictví

Objednávka

Číslo	Název	Množství	Cena	Číslo objednávky	Středisko	Objednal
0.49051277478273	dx2450 MT Athlon X2-44508	1	<input type="text" value="9020.00"/>	09000900000008	000900	Bc. Miroslav Pavlíček

Obrázek 22 - Aktivace objednávky

Zde je uvedeno katalogové číslo pro snadnější vyhledání položky v katalogu dodavatele, dále název v případě změny položky za jinou, nebo pro kontrolu při objednávání. Současně je zde zobrazeno množství a cena za jeden kus. Tuto cenu je možné upravit dle aktuálního ceníku a díky této funkci se udržjí v systému aktuální ceny jednotlivého zboží. Změny cen se provádí v databázi sklad a u právě objednávaných položek. Pro kontrolu je zde také vypsáno číslo objednávky, středisko a kdo ze střediska předmětnou objednávku vytvořil.

### 6.5.4.2 Vytvořené objednávky

Tato stránka modulu aktivace zobrazuje zboží, které je objednáno (prošlo novou objednávkou) a mělo by být doručeno.

Nové objednávky Vytvořené objednávky **Dodané objednávky**

Odeslané objednávky

Dodavatelé

Dodavatelé

MARTEK MEDICAL a.s. Změnit

Objednané zboží

Dodáno	Číslo	Název	Množství	Cena	Číslo objednávky
<input type="checkbox"/>	0.49051277478273	dx2450 MT Athlon X2-4450B	1	9020.00	09000900000010

Potvrdit odeslání objednávky

Obrázek 23 - Vytvořené objednávky

Zde aktivátor může vybrat, které zboží je dodáno a pokud je v pořádku dodáno, tak potvrdit, že již přišlo. Pro kontrolu je zde znovu uvedeno katalogové číslo, název, množství, cena a číslo objednávky. Po zaškrtnutí pole, že je zboží dodáno, se tato změna automaticky projeví v seznamu dodávek a zboží je připraveno k vyzvednutí.

### 6.5.4.3 Dodané objednávky

Tato stránka modulu aktivace objednávky se stará o výtisk předávacích protokolů pro jednotlivé uživatele, kteří objednávali. Protokoly se tisknou najednou, avšak bez okolní grafiky, zůstávají pouze tabulky. V jednotlivých tabulkách jsou uvedeny informace o tom, kdo co objednal, za jaké středisko, kdy byla objednávka vytvořena a kdy byl vygenerován vlastní protokol a kdo protokol vytvořil. Dále zde číslo objednávky pro účtárnu. Ve vlastním obsahu je katalogové číslo produktu a jeho název. Dále je zde uvedena cena, za kterou bylo zboží objednáno, jaké množství a v jakých jednotkách je udáno. Jako poslední položka je celková cena za řádek. Na konci protokolu je místo pro podpis přebírajícího. Ten svým podpisem potvrdí, že zboží bylo předáno.

[Nové objednávky](#) [Vytvořené objednávky](#) [Dodané objednávky](#)

**Vyberte osobu, od které chcete vytisknout protokol**

Dodané zboží					
Uživatel: Bc. Miroslav Pavlíček <input type="button" value="Vybrat"/>					
Jméno:	Bc. Miroslav Pavlíček	Středisko	100901 (Vedení)	Vytvořena:	13.04.2009 01:50:57
Vytvořil:	Bc. Miroslav Pavlíček			Vytvořeno:	13.04.2009 01:43:39
Číslo objednávky:	09100901000006				
Katalogové číslo	Název	Cena	Množství	Jednotky	Suma
0.49051277478273	dx2450 MT Athlon X2-4450B	9020	1	ks	9020
Převzal:					

Jméno:	Bc. Miroslav Pavlíček	Středisko	000900 (Centrální řízení DOZ)	Vytvořena:	18.04.2009 06:50:35
Vytvořil:	Bc. Miroslav Pavlíček			Vytvořeno:	13.04.2009 01:43:39
Číslo objednávky:	09000900000009				
Katalogové číslo	Název	Cena	Množství	Jednotky	Suma
0.49051277478273	dx2450 MT Athlon X2-4450B	9020	1	ks	9020
Převzal:					

Jméno:	Bc. Miroslav Pavlíček	Středisko	000900 (Centrální řízení DOZ)	Vytvořena:	18.04.2009 06:50:41
Vytvořil:	Bc. Miroslav Pavlíček			Vytvořeno:	13.04.2009 01:43:39
Číslo objednávky:	09000900000010				
Katalogové číslo	Název	Cena	Množství	Jednotky	Suma
0.49051277478273	dx2450 MT Athlon X2-4450B	9020	1	ks	9020
Převzal:					

Schválení
<input type="button" value="Hromadné přijetí"/>

Obrázek 24 - Dodané objednávky

## 6.5.5 Sklad

Tento modul slouží pro doplňování materiálu do skladu a pro úpravu skladových zásob. Skládá se z několika stránek, každá pro vlastní úpravu jednotlivých položek.

### 6.5.5.1 Přidat položku do skladu

Tato stránka se stará o import nových položek do skladu informačního systému. Je vhodný pro zadávání jednotlivých položek. Tento systém umí vložit věci pouze do Nejnižší skupiny. Ostatní skupiny jsou pouze třídící.

Jako parametr je zvolen dodavatel zboží, zboží bude následně objednáno u tohoto dodavatele, dále skupina, do které zboží patří a v které se objeví. Tyto skupiny jsou seřazeny podle novosti, kdy naposledy přidaná skupina je na posledním místě.

Dále pak název zboží, kde by měl být uveden krátký, ale výstižný popis. Další položkou jsou jednotky, kde se uvede, v jakých jednotkách je zboží dodáváno. Následně se vyplní cena bez DPH a dole do posledního políčka pak sazba DPH pro danou položku.

### Sklad - číselník

[Přidat položku do skladu](#) [Schválit skladové zboží](#) [Skladový seznam](#) [Přidat skupinu zboží](#) [Seznam skupin](#) [Nový dodavatel](#) [Seznam dodavatelů](#)

Dodavatel

Medikoplast Address: <http://objednavky/index.php?menu=sklad&pridat=ano>

Skupina

Rukavice vyšetřovací latexové nesterilní pudrované

Název zboží

Jednotky

Cena

Katalogové číslo

Popis

Ihned zobrazit

Sazba DPH

19

Vložit do skladu

#### Obrázek 25 - Přidání položky do skladu

Cena bez DPH se vypisuje proto, že někteří lékaři mají krácené DPH a tak cena vstupuje do tohoto výpisu zkrácena o část původního DPH.

Další důležitou informací je katalogové číslo. Je to číslo, pod jakým můžeme zboží najít v katalogu dodavatele nebo i výrobce. Poslední položkou je popis, kdy se mohou vyplnit další potřebné informace o produktu.

Políčko „ihned zobrazit“ určuje, zda se zboží má objevit ihned v systému. V tuto chvíli je inaktivní.

### 6.5.5.2 Skladový seznam

Tato stránka umožňuje zobrazit veškeré zboží vybraného dodavatele a umožní změnu umístění zboží jiného dodavatele, ceny katalogového čísla a jména a to současně pro několik kusů zboží najednou. Generace této stránky může trvat i několik minut.

Sklad - číselník

Přidat položku do skladu Schválit skladové zboží Skladový seznam Přidat skupinu zboží Seznam skupin Nový dodavatel Seznam dodavatelů

Medikoplast Změnit Vpis může trvat i několik minut!

Jméno	Cena	Katalog	Skupina	Dodavatel
HP Ink Cart Photo No. 348 pro DJ 5740,6540, C9369EE	425.000	0.9645753	Inkousty	MARTEK MEDICAL a.s.
HP Ink Cart Black No. 338 pro DJ 5740,6540, C8766EE	44.000	0.7649351	inkousty	MARTEK MEDICAL a.s.
HP Ink Cart Colour No. 343 pro DJ 5740,6540, C8766EE	44.000	0.9309496	inkousty	MARTEK MEDICAL a.s.
ARMOR páska pro EPSON LQ / LX-200, 300, 500, 570, 580, 880	44.000	0.3599427	Jehličkové	MARTEK MEDICAL a.s.
DIMM DDR 1GB, 400MHz, A-DATA, Bulk	526.000	0.0068649	Operační paměti	MARTEK MEDICAL a.s.
DIMM DDR2 1GB, 800MHz A-DATA, Bulk	526.000	0.9544964	Operační paměti	MARTEK MEDICAL a.s.
PC - Office	3113.000	0.7518875	Neznačkové	MARTEK MEDICAL a.s.
ACER MT LCD V193WAb 19\	3113.000	0.8959483	Monitory	MARTEK MEDICAL a.s.
HP LaserJet M2727nf MFP	12188.000	0.2240789	Laserové tiskárny	MARTEK MEDICAL a.s.
LG OS DVD-ROM Drive DH16NS10 SATA Bare Black	287.000	0.4325499	Mechaniky	MARTEK MEDICAL a.s.
dx2450 MT Athlon X2-4450B	9020.000	0.4905127	Hewlett - Packard	MARTEK MEDICAL a.s.
EPSON LX-300+II, A4, 9 jehel, 377 zn/s, 1+4 kopii, USB 1.1, LP	4146.000	0.1549141	Jehličkové	MARTEK MEDICAL a.s.
Linksys GigaSwitch SD2005, 5x10/100/1000, desktop	1258.000	0.3030323	Switche	MARTEK MEDICAL a.s.
LG OS DVD±R±RW/RAM Drive	1028.000	0.0504193	Mechaniky	MARTEK MEDICAL a.s.
HP LaserJet P1006 - (16str/min, A4, USB)	2814.000	0.3429997	Laserové tiskárny	MARTEK MEDICAL a.s.

Upravit zboží

Obrázek 26 - Změna skladu

Po úpravě potřebných položek je možné celou úpravu nahrát do databáze. Změny se projeví jen na nově zadávaných objednávkách.

### 6.5.5.3 Přidat skupiny zboží

Tato položka umožňuje přidávat skupiny zboží. Vyberete si nadskupinu a vložíte informace o podskupině.

Sklad - číselník

Přidat položku do skladu Schválit skladové zboží Skladový seznam Přidat skupinu zboží Seznam skupin Nový dodavatel Seznam dodavatelů

Název skupiny

Popis skupiny

Vložit do databáze

Obrázek 27 - Přidání skupiny zboží

K této položce patří i funkce seznam skupin, která vypíše všechny skupiny zboží, které jsou aktuálně uloženy v systému.

#### 6.5.5.4 Nový dodavatel

Poslední položka, která umožňuje přidat nového dodavatele zboží.

##### Sklad - číselník

[Přidat položku do skladu](#) [Schválit skladové zboží](#) [Skladový seznam](#) [Přidat skupinu zboží](#) [Seznam skupin](#) [Nový dodavatel](#) [Seznam dodavatelů](#)

Jméno dodavatele

Ulice a číslo

Město

PSČ

IČ

DIČ

Kontaktní osoba

Telefon

Email

#### Obrázek 28 - Nový dodavatel

Pro interní potřebu objednávání stačí vyplnit jméno dodavatele. Systém ale také současně umožňuje po dohodě s dodavatelem zasílání buď emailových, nebo přímo datových objednávek. V této souvislosti je systém vybaven i dalšími informacemi o adrese dodavatele, jako jsou např. ulice a číslo popisné, město, dále poštovní směrovací číslo, identifikační číslo, daňové číslo, kontaktní osoba a především telefon a email. Také k této skupině se váže vlastní seznam dodavatelů.



## 6.5.6 Přidat nového uživatele

Tato nabídka umožňuje přidávat nové uživatele do systému. Pro uživatele je nutné vyplnit několik důležitých informací, jako jsou uživatelské jméno a heslo, jméno a příjmení, email nebo telefon. Pokud osoba má mít schvalovací pravomoc, je nutné vložit částku, do jejíž výše bude moci uživatel schvalovat objednávky. Jako další identifikace slouží také osobní číslo.

Přidání nového uživatele

Uživatelské jméno

Heslo

Heslo znova

Jméno a příjmení

Email

Telefon

Limit CZK (pouze lidé, kteří mají právo schvalovat)

Osobní číslo

Středisko

000900 - Centrální řízení DOZ (25000000.00)

100901 - Vedení (1000000000.00)

100902 - Technická správa ()

100904 - Výkonová účtárna (100000000.00)

Oddělení

DZ (Dopravní zdravotnictví)

L (Lékař)

LAB (Laboratoř)

LEK (Lékárna)

Práva

vytvoření objednávky za své nákladové středisko

vytvoření objednávky za jakékoliv nákladové středisko

schválit objednávku za své nákladové středisko

schválit objednávku za všechna nákladová střediska

aktivace a vyřízení objednávky za své středisko

aktivace a vyřízení objednávky za všechna střediska

Plnění a schvalování skladu

Plnění dočasného skladu

Práce s nákladovými středisky

Práce s uživatelem

Přidělování práv

Vkládat smlouvy

Prohlížet smlouvy

Vložit uživatele

Obrázek 29 - Přidání uživatele formulář

Pro vlastní práci je nutné zařadit uživatele do odpovídajících jednotlivých nákladových středisek. Je možné zvolit od jednoho do všech. Jako další povinný údaj je třeba zadat do jakého oddělení daný uživatel patří. Toto je důležité hlavně z hlediska DPH, kdy lékaři mají určitý odpočet na dani na vstupu.

Poslední částí je přidělení práv. Zaškrtnají se vybrané možnosti, a tímto se uživateli automaticky přiřadí role daného uživatele.

K tomuto seznamu se přidává ještě seznam všech uživatelů, který umožňuje uživateli s právy pro práci s uživatelem měnit heslo, popřípadě uživatele smazat.

## **6.6 Instalace systému u uživatelů**

Již z povahy vlastní aplikace vyplývá, že instalace informačního systému se skládala pouze z vytvoření zástupce na plochu a přidání položky informačního systému mezi oblíbené. Další z potřebných věcí bylo aktualizování maximálního počtu internetových prohlížečů. Některé počítače bohužel obsahovaly ještě Internet Explorer 5, který již nevyhovuje aktuální aplikaci. U těchto počítačů bylo nutné nejdříve nainstalovat nový prohlížeč FireFox 2 a ten byl zamaskován jako Internet Explorer aby uživatel měl co nejméně problémů s přechodem na nový internetový prohlížeč. Poté již nebyl problém s vlastním zprovozněním aplikace.

## 7 Testování systému

Předmětem této části je okruh testování, plnění daty a zavádění systému mezi uživatele.

### 7.1 Alfa - testování

Každý modul systému byl ihned po kompletním vytvoření a naimplementování do informačního systému otestován na zkušebním vzorku dat. Při tomto alfa-testování bylo odstraněno zhruba 90% programovacích chyb a 40% chyb funkčních.

Náročnost testování vzrůstala s počtem přidávaných modulů, které mezi sebou komunikovaly. Při každém testu byla ověřena také stabilita databáze a její operativní zatížení stejně, jako zatížení vlastního webového serveru tak, aby se případně odstranil maximální počet chyb a rychlost systému byla zachována.

#### 7.1.1 Průběh testování

Po vytvoření modulu byl otestován sled metod v jeho třídách a bylo dostatečně prověřeno zabezpečení proti zneužití. Každý modul obsahuje interní kontrolu chyb vkládání a spouštění funkcí. Stejně tak je kontrolována i komunikace s databází, kde se jedná hlavně o obranu proti sql-injection napadení a konzistence vrácených dat.

Po této fázi testování bylo provedeno otestování v kompletním systému, včetně předávání dat a maximalizace zrychlení chodu systému. Zároveň byl využit vzorek dat pro otestování funkčnosti, a pokud se jednalo o modul s možností vkládat, tak i stav vkládaných informací.

### 7.2 Beta - testování

Po naprogramování celého informačního systému, byl tento nahrán na aplikační a databázový server v Nemocnici s poliklinikou Praha, Italská. Při tomto testu bylo nutné zjistit problémy a možné komplikace v chodu systému při plném provozu. Vlastní aplikace byla zprovozněna hlavně na počítačích sester, které se dle počátečního průzkumu starají o největší procento objednávek.

Během tohoto testování bylo odstraněno více jak 90% procent funkčních chyb a 95% všech chyb programových. Testování probíhalo 14 dní a celý systém byl plynule převeden do plného provozu.

### 7.2.1 Školení

V době testování na reálných datech byl veškerý personál Nemocnice s poliklinikou v Praze Italské proškolen v ovládání tohoto informačního systému.

Problémy při uvedení systému do praxe

Vlastní školení probíhalo skupinově a bylo velmi jednoduché díky intuitivnímu a snadnému ovládání celého informačního systému, který kopíruje původní proces objednávání, tak že vlastní ovládání nebylo problémem ani pro osoby vyššího věku.

### 7.2.2 Problémy při beta-testování

Hlavně starším pracovníkům a zaměstnancům s dioptrickými brýlemi nevyhovovalo horizontální členění nadskupin a nepřehledné rozbalování jednotlivých skupin, dále velikost písma a barva podkladu.

Tento problém byl vyřešen změnou fontu a formátu písma. Jednotlivé katalogové skupiny byly usazeny do horizontální matice o velikosti 4xn. Pro některé zaměstnance byl informační systém graficky a funkčně upraven s ohledem na jejich specifické žádosti, například ohledně velikosti písma a grafiky.

## 7.3 Testování v internetových prohlížečích


Jelikož jako kritérium bylo vytvořit systém funkční na maximálním počtu zařízení, proběhlo testování na několika internetových prohlížečích.

Informační systém byl testován na

- Internet Explorer 7,8
- Opera 10
- Firefox 3
- Chrome 2

## 7.3.1 Internet explorer 8 a 7

Informační systém DZ a.s.



**Menu** **Vytvoření objednávky**

Vytvořit objednávku  
Seznam objednávek  
Schvalování objednávek  
Manager Schvalování  
Aktivace objednávek  
Sklad  
Přidat uživatele  
Seznam uživatelů  
Nastavení práv  
Změna hesla  
Odhlásit

Eměnit středisko  
100901 Vedení

Vybrané středisko: 100901 Vedení  
Interní číslo objednávky: 09100901000011

[Lékařský materiál](#) [Spotřební materiál](#) [Výpočetní technika](#)

Vyhledávání

Vyhledávání

**Seznam nejčastěji objednávaných položek**

Název	Detail	Cena		
dx2450 MT Athlon X2-4450B	2 GB RAM, 250 GB Disk, Vista B	9 020,00 CZK	1	<input type="button" value="Objednat"/>
Pytel 700x1100.80mi.černý.R (150)	Pytel 700x1100.80mi.černý.R (150)	9,00 CZK	1	<input type="button" value="Objednat"/>
čepelka skalpelová sterilní ě.11pro držák ě.3(100ks)	čepelka skalpelová sterilní ě.11pro držák ě.3(100ks)	60,00 CZK	10	<input type="button" value="Objednat"/>

Objednávka č. 09100901000011


Sk.č.	Název	Jednotky	Množství	Cena	Celková cena	
TER1SR0F2032	l.v.kanyla rovná 20G	ks	10	13,55	135,50	<input type="button" value="Smazat"/>
0.75356931588945	denní pracovní záznam v a.s.	ks	1	0,42	0,42	<input type="button" value="Smazat"/>

Poznámka k objednávce

Dopravní zdravotnictví a.s. Developed by: Bc. Miroslav Pavlíček

Obrázek 30 - Internet Explorer 8

Informační systém DZ a.s.



**Menu** **Vytvoření objednávky**

Vytvořit objednávku  
Seznam objednávek  
Schvalování objednávek  
Manager Schvalování  
Aktivace objednávek  
Sklad  
Přidat uživatele  
Seznam uživatelů  
Nastavení práv  
Změna hesla  
Odhlásit

Eměnit středisko  
100901 Vedení

Vybrané středisko: 100901 Vedení  
Interní číslo objednávky: 09100901000011

[Lékařský materiál](#) [Spotřební materiál](#) [Výpočetní technika](#)

Vyhledávání

Vyhledávání

**Seznam nejčastěji objednávaných položek**

Název	Detail	Cena		
dx2450 MT Athlon X2-4450B	2 GB RAM, 250 GB Disk, Vista B	9 020,00 CZK	1	<input type="button" value="Objednat"/>
Pytel 700x1100.80mi.černý.R (150)	Pytel 700x1100.80mi.černý.R (150)	9,00 CZK	1	<input type="button" value="Objednat"/>
čepelka skalpelová sterilní ě.11pro držák ě.3(100ks)	čepelka skalpelová sterilní ě.11pro držák ě.3(100ks)	60,00 CZK	10	<input type="button" value="Objednat"/>

Objednávka č. 09100901000011

Sk.č.	Název	Jednotky	Množství	Cena	Celková cena	
TER1SR0F2032	l.v.kanyla rovná 20G	ks	10	13,55	135,50	<input type="button" value="Smazat"/>
0.75356931588945	denní pracovní záznam v a.s.	ks	1	0,42	0,42	<input type="button" value="Smazat"/>


Poznámka k objednávce

Dopravní zdravotnictví a.s. Developed by: Bc. Miroslav Pavlíček

Obrázek 31 - Internet Explorer 7

## 7.3.2 Opera 10 a Firefox 3

Informační systém DZ a.s.



**Menu** **Vytvoření objednávky**

Vytvořit objednávku

Seznam objednávek

Schvalování objednávek

Manager Schvalování

Aktivace objednávek

Sklad

Přidat uživatele

Seznam uživatelů

Nastavení práv

Změna hesla

Odhlásit

Vybrané středisko: 100901 Vedení  
Interní číslo objednávky: 09100901000011

[Lékařský materiál](#) [Spotřební materiál](#) [Výpočetní technika](#)

**Vyhledávání**

Vyhledat

Vyhledat

**Seznam nejčastěji objednávaných položek**

Název	Detail	Cena		
dx2450 MT Athlon X2-4450B	2 GB RAM, 250 GB Disk, Vista B	9 020,00 CZK	1	<input type="button" value="Objednat"/>
Pytel 700x1100.80mi.černý.R (150)	Pytel 700x1100.80mi.černý.R (150)	9,00 CZK	1	<input type="button" value="Objednat"/>
Čepelka skalpelová sterilní č.11pro držák č.3(100ks)	Čepelka skalpelová sterilní č.11pro držák č.3(100ks)	60,00 CZK	10	<input type="button" value="Objednat"/>

Objednávka č. 09100901000011


Sk.č.	Název	Jednotky	Množství	Cena	Celková cena	
TER15ROF2032	i.v.kanýla rovná 20G	ks	10	13,55	135,50	<input type="button" value="Smazat"/>
0.75356931588945	denní pracovní záznam v a.s.	ks	1	0,42	0,42	<input type="button" value="Smazat"/>

Poznámka k objednávce

Dopravní zdravotnický a.s. Developed by: Bc. Miroslav Pavlíček

Obrázek 32 - Opera 10

Informační systém DZ a.s.



**Menu** **Vytvoření objednávky**

Vytvořit objednávku

Seznam objednávek

Schvalování objednávek

Manager Schvalování

Aktivace objednávek

Sklad

Přidat uživatele

Seznam uživatelů

Nastavení práv

Změna hesla

Odhlásit

Vybrané středisko: 100901 Vedení  
Interní číslo objednávky: 09100901000011

[Lékařský materiál](#) [Spotřební materiál](#) [Výpočetní technika](#)

**Vyhledávání**

Vyhledat

Vyhledat

**Seznam nejčastěji objednávaných položek**

Název	Detail	Cena		
dx2450 MT Athlon X2-4450B	2 GB RAM, 250 GB Disk, Vista B	9 020,00 CZK	1	<input type="button" value="Objednat"/>
Pytel 700x1100.80mi.černý.R (150)	Pytel 700x1100.80mi.černý.R (150)	9,00 CZK	1	<input type="button" value="Objednat"/>
Čepelka skalpelová sterilní č.11pro držák č.3(100ks)	Čepelka skalpelová sterilní č.11pro držák č.3(100ks)	60,00 CZK	10	<input type="button" value="Objednat"/>

Objednávka č. 09100901000011

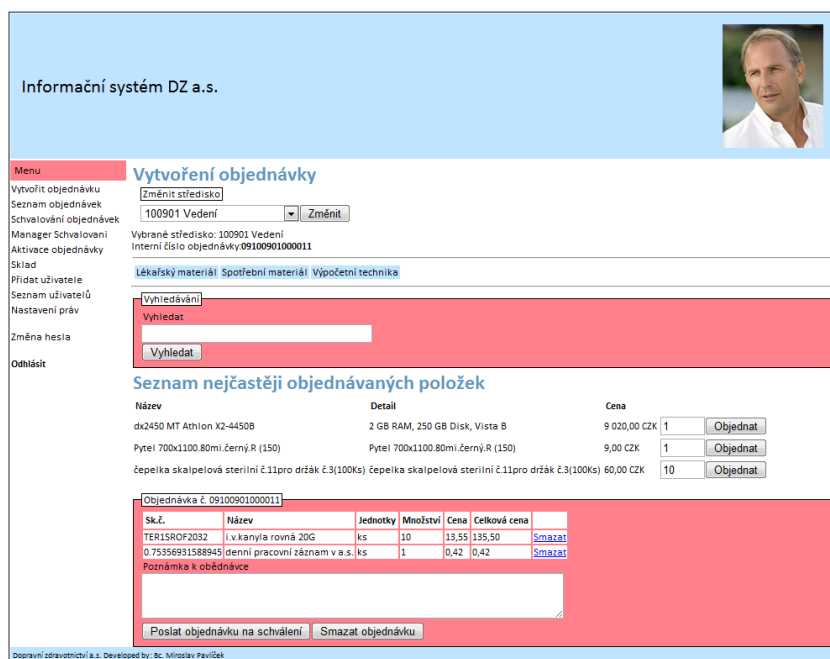
Sk.č.	Název	Jednotky	Množství	Cena	Celková cena	
TER15ROF2032	i.v.kanýla rovná 20G	ks	10	13,55	135,50	<input type="button" value="Smazat"/>
0.75356931588945	denní pracovní záznam v a.s.	ks	1	0,42	0,42	<input type="button" value="Smazat"/>

Poznámka k objednávce

Dopravní zdravotnický a.s. Developed by: Bc. Miroslav Pavlíček

Obrázek 33 - Firefox 3

### 7.3.3 Google Chrome 2



Obrázek 34 - Google Chrome 2

Testování proběhlo bez problémů, grafická podoba je v pořádku ve všech testovaných prohlížečích. Nejlepší vykreslení má Opera 10 a Google Chrome 2.

### 7.4 Testování rychlosti

Jedna z funkcí systému je hromadná úprava skladového seznamu. Generování tohoto seznamu trvá od několika vteřin po několik minut. Zde jsou uvedeny výsledky rychlostního testu. Každý prohlížeč generoval stránku několikrát a toto jsou průměrné hodnoty ze všech testů.

Test byl proveden na procesoru CORE 2 DUO 2,1 GHz a Windows Vista

**Tabulka 2 - Test rychlosti prohlížečů**

Prohlížeč	Rychlost načtení [min]
<b>Opera 10</b>	0:45:47
<b>Internet Explorer 8</b>	1:09:50
<b>Firefox 3</b>	1:40:00
<b>Google Chrome 1.0</b>	0:19:14

Výsledky testu jsou docela zajímavé. Stránka po vygenerování má necelých 15 MB. Doba vlastního vykreslení je dána tím, že se pro každou položku vytváří seznam skupin a dodavatelů. Při 2083 položkách od dodavatele je pro každou z těchto položek vygenerována 223 položek skupin a 4 položky dodavatelů. Značná velikost souboru je dána velkým počtem HTML značek.



## 8 Závěr

V celé diplomové práci se tvůrce snažil navrhnout a vytvořit ucelený informační systém schopný plného nasazení na požadovaných aplikacích.

Systém umožňuje jednodušší správu a zpracování ekonomické agendy zastřešující provoz v zařízeních Dopravního zdravotnictví a.s. Informační systém byl vytvořen přesně dle požadavků vedení společnosti. Celý objednávkový proces byl urychlen, jedná se hlavně o zrychlení zadání, kdy nejsou již potřeba vypisovat ručně objednávky, ale stačí je přidat do objednávkového košíku. Další zrychlení je dáno okamžitou reakcí (schválení) vytvořené objednávky. Celý proces od objednání, až po odeslání reálné objednávky nezabere víc než 3 minuty.

Celý informační systém byl vytvořen s cílem minimalizovat náklady na pořízení a zavedení a přitom zabezpečit plnou funkčnost. Jelikož celý systém byl vyvíjen na OpenSource platformě PHP a pro svoje potřeby využívá taktéž OpenSource databázi MySQL jsou náklady na vytvoření funkčního prostředí pro běh aplikace minimální. Náklady na aplikační prostředí jsou nulové. Určité náklady jsou v ceně hardwaru, ale systém je schopen fungovat již na existujících zařízeních.

Velikou výhodou je, že systém na bázi web server-client, nevyžaduje na uživatelských stanicích žádný speciální program a pro jeho provozování stačí dnešní internetové prohlížeče. Systém web server-client také umožňuje velice rychlé šíření aktualizací a úprav aplikace, díky které se snižují prodlevy, a uživatel vždy pracuje na nejaktuálnější verzi aplikace.

V této souvislosti nejsou kladeny ani žádné velké požadavky na jednotlivé uživatele. K tomu přispívá i to, že systém byl navržen dle původního způsobu objednávání a evidování informací a uživatelé se v informačním systému snadno orientují. Objektově orientovaný styl programování usnadňuje přidávání nových a úpravy stávajících funkcí.

## 9 Seznam literatury

1. **Koncorcium, W3C.** W3C HTML. *W3C*. [Online] W3C, 26. 9 2006. [Citace: 4. 3 2007.] <http://www.w3.org/MarkUp/#xhtml1>.
2. **The Apache Software Foundation.** The Apache Software Foundation. *The Apache Software Foundation*. [Online] The Apache Software Foundation, 2009. [Citace: 19. Březen 2009.] <http://apache.org/>.
3. **The PHP Group.** PHP: Hypertext Preprocessor. *PHP: Hypertext Preprocessor*. [Online] The PHP Group, 2. Duben 2009. [Citace: 4. Duben 2009.] <http://php.net/>.
4. **team, phpMyAdmin devel.** phpMyAdmin. *phpMyAdmin*. [Online] phpMyAdmin devel team, 16. Únor 2009. [Citace: 12. Březen 2009.] [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php).
5. **Sun Microsystems, Inc.** MySQL. *MySQL*. [Online] Sun Microsystems, Inc., 3. Leden 2009. [Citace: 4. Březen 2009.] <http://www.mysql.com/>.
6. **Kolektiv autorů.** *PHP Programujeme profesionálně*. místo neznámé : Computer Press, 2001. 8072263102.
7. **Michael Howard, David LeBlanc.** *Bezpečný kód - Techniky a strategie tvorby bezpečných webových aplikací*. Computer Press : autor neznámý, 2008. 978-80-251-2050-7.
8. **LCS.** Helios - podnikové informační systémy. *Helios - podnikové informační systémy*. [Online] LCS, 29. 03 2009. [Citace: 29. 03 2009.] <http://www.helios.eu/>.
9. **Wikipedia.org.** SAP R/3. *Wikipedia.org*. [Online] Wikipedia.org, 10. Leden 2009. [Citace: 16. Březen 2009.] [http://cs.wikipedia.org/wiki/SAP\\_R/3](http://cs.wikipedia.org/wiki/SAP_R/3).
10. **Meruňka, Vojtěch.** *Datové modelování*. místo neznámé : Alfa Publishing s.r.o., 2006. 80-86851-54-0.
11. **Krug, Stebe.** *Webdesign Nenuťte uživatele přemýšlet*. Brno : Cumputer Press, 2006. 80-251-1291-8.

12. **Wikipedia.org.** Open source. *wikipedia.org*. [Online] Wikimedia Foundation, Inc, 11. duben 2008. [Citace: 13. červen 2008.] [http://en.wikipedia.org/wiki/Open\\_source](http://en.wikipedia.org/wiki/Open_source).

## 10 Přílohy

### 10.1 Seznam příloh

Příloha – **A** – ERD schéma databáze

Příloha – **B** – Ganttův diagram

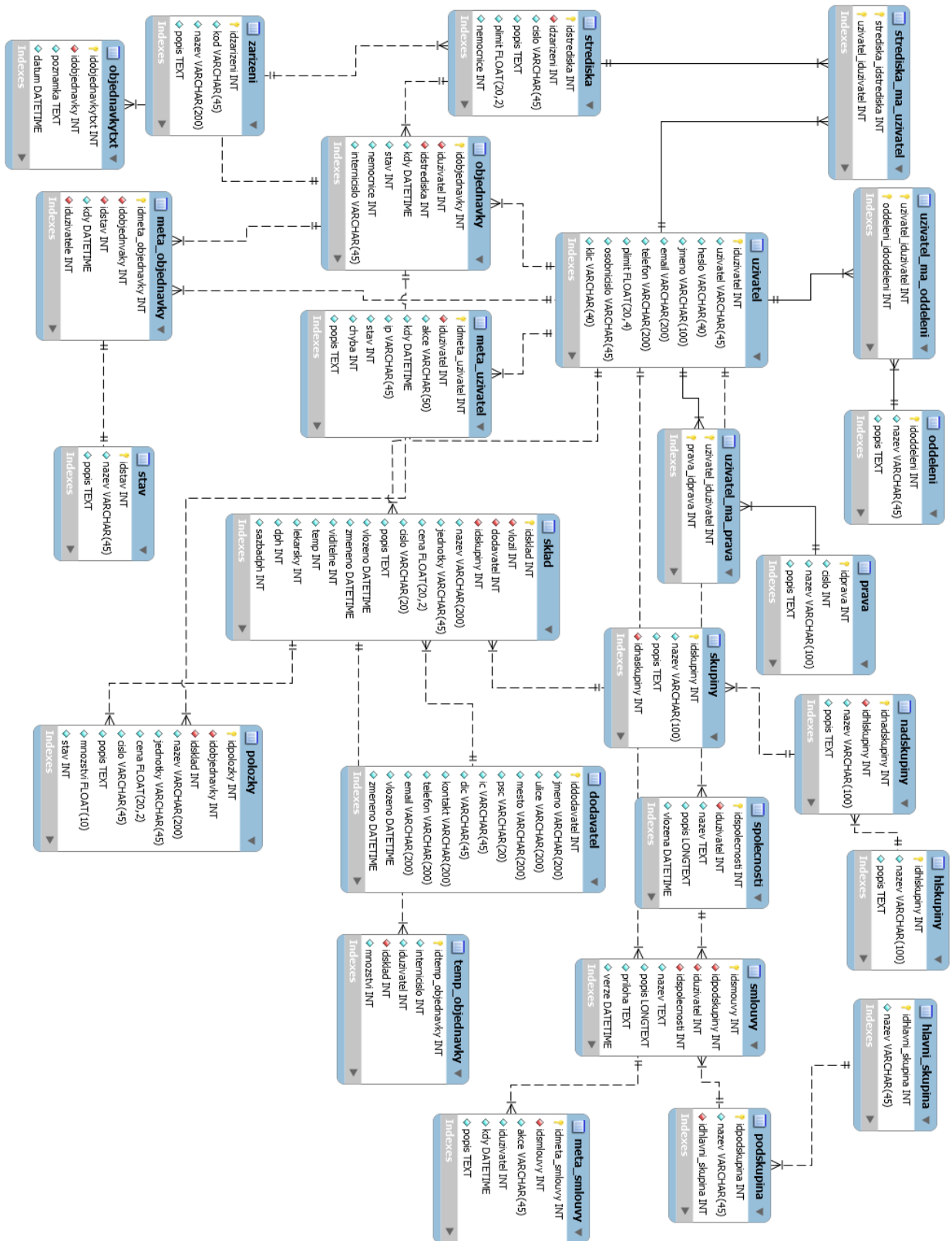
Příloha – **C** – Zdrojové kódy pro informační systém

### 10.2 Seznam obrázků

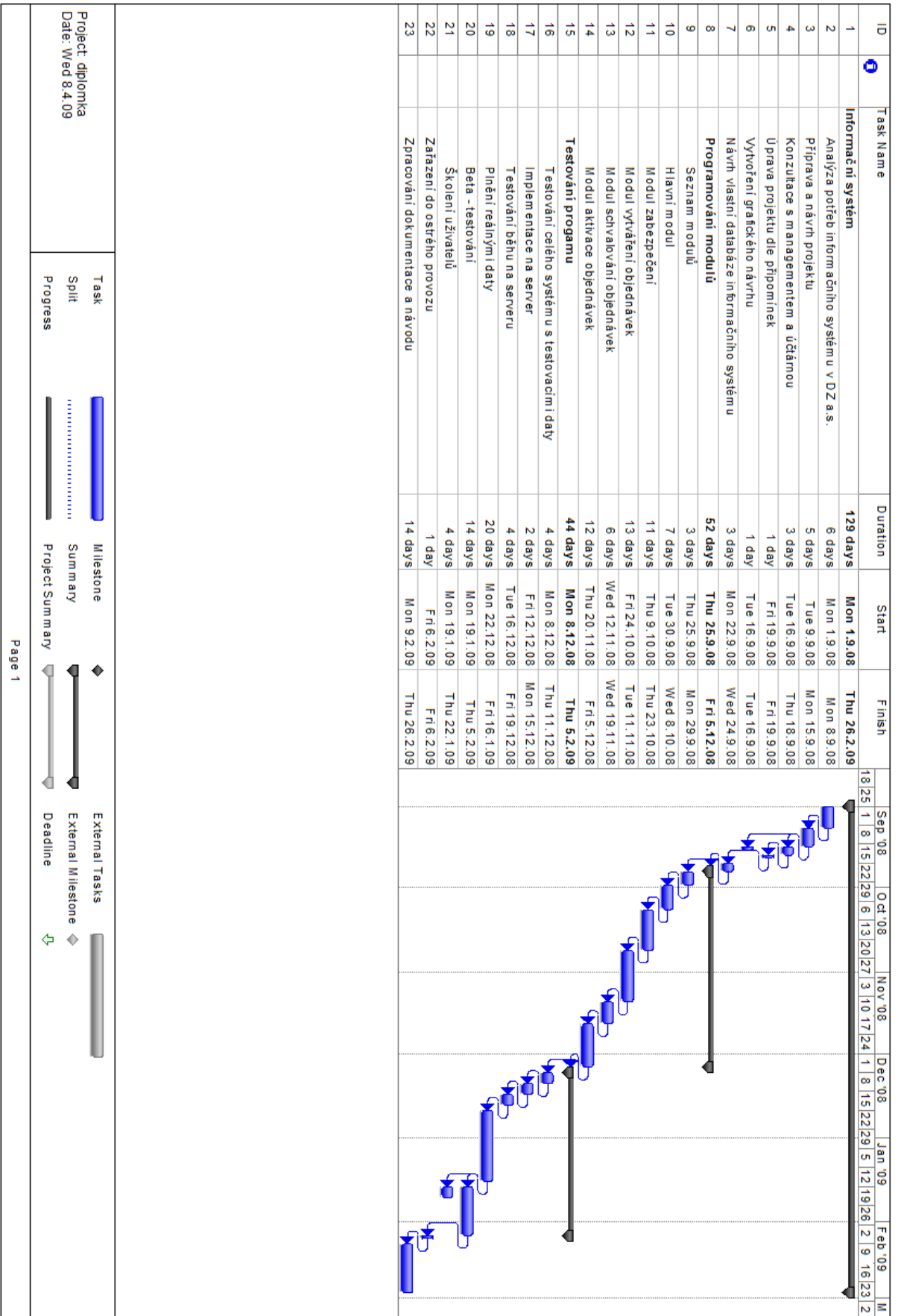
Obrázek 1 - Metoda dtb - komunikace s databází.....	16
Obrázek 2 - Konstruktor pro přesměrování na zabezpečené stránky.....	17
Obrázek 3 - přihlašovací okno .....	20
Obrázek 4 - Kód pro generování objednávkového čísla .....	20
Obrázek 6 - Formát čísla objednávky .....	27
Obrázek 5 - Možnosti volby středisek .....	27
Obrázek 7 - Helios Green schéma modulů .....	31
Obrázek 8 - Helios Orange seznam modulů .....	31
Obrázek 9 - Helios Red seznam modulů .....	32
Obrázek 10 - Prvotní návrh grafické podoby.....	36
Obrázek 11 - Finální grafický návrh .....	37
Obrázek 12 - Hlavní skupiny .....	38
Obrázek 13 - Nadskupiny (Lékařský materiál).....	38
Obrázek 14 - Jednotlivé skupiny (Kanyly).....	38
Obrázek 15 - Seznam položek.....	39
Obrázek 16 - Objednávkový košík .....	39
Obrázek 17 - Seznam objednávek .....	40
Obrázek 18 - Úprava objednávky .....	40
Obrázek 19 - Seznam procesů objednávky.....	41
Obrázek 20 - Poznámky k objednávce.....	41
Obrázek 21 - Seznam objednávek na schválení .....	42

Obrázek 22 - Aktivace objednávky .....	43
Obrázek 23 - Vytvořené objednávky .....	44
Obrázek 24 - Dodané objednávky .....	45
Obrázek 25 - Přidání položky do skladu.....	46
Obrázek 26 - Změna skladu .....	47
Obrázek 27 - Přidání skupiny zboží .....	47
Obrázek 28 - Nový dodavatel .....	48
Obrázek 29 - Přidání uživatele formulář.....	49
Obrázek 30 - Internet Explorer 8 .....	53
Obrázek 31 - Internet Explorer 7 .....	53
Obrázek 32 - Opera 10.....	54
Obrázek 33 - Firefox 3.....	54
Obrázek 34 - Google Chrome 2 .....	55

## 10.3 Příloha A – Schéma databáze – vnitřní systém



## 10.4 Příloha B – Ganttův diagram – výstup MS Project



## 10.5 Příloha C - Zdrojové kódy aplikace

### dodavatel.php

```
<?php
class Dodavatel extends Main{

    var $iddodavatel;
    var $jmeno;
    var $ulice;
    var $mesto;
    var $psc;
    var $ic;
    var $dic;
    var $kontakt;
    var $telefon;
    var $email;
    var $vlozeno;
    var $zmeneno;

    function addRow()
    {
        $sql = $this->prp("INSERT INTO dodavatel VALUES (NULL,?, ?, ?, ?, ?, ?, ?, ?, ?, NOW(), NOW())");
        $sql->execute(array("$this->jmeno", "$this->ulice", "$this->mesto", "$this->psc", "$this->ic", "$this->dic", "$this->kontakt", "$this->telefon", "$this->email"));
        $this->sqlchyba($sql);
    }

    function check(){
        $this->jmeno = $_POST[jmeno];
        $this->ulice = $_POST[ulice];
        $this->mesto = $_POST[mesto];
        $this->psc = $_POST[psc];
        $this->dic = $_POST[dic];
        $this->ic = $_POST[ic];
        $this->kontakt = $_POST[kontakt];
        $this->telefon = $_POST[telefon];
        $this->email = $_POST[email];

        if (strlen($this->jmeno)<1)
            throw new Exception ("Nebylo vyplněno jméno",1001);
        if (strlen($this->ulice)<1)
            throw new Exception ("Nebyla vyplněna ulice",1002);
        if (strlen($this->psc)<1)
            throw new Exception ("Nebylo vyplněno PSČ",1003);
        if (strlen($this->dic)<1)
            throw new Exception ("Nebylo vyplněno DIČ",1004);
        if (strlen($this->ic)<1)
            throw new Exception ("Nebylo vyplněno IČ",1005);
        if (strlen($this->kontakt)<1)
            throw new Exception ("Nebyla vplněna kontaktní osoba",1006);
        if (strlen($this->telefon)<1)
            throw new Exception ("Nebyl vyplněn telefon",1007);
        if (filter_var($this->email,FILTER_VALIDATE_EMAIL)==false)
            throw new Exception ("Email je neplatný",1008);
    }

    function getList(){
        $sql=$this->prp("SELECT iddodavatel,jmeno,ulice,mesto,psc,ic,dic,kontakt,telefon,email,vlozeno,zmeneno FROM dodavatel");
        $sql->execute(array());
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

    function getDataCount(){
        $sql = $this->prp("SELECT count(iduzivatel) FROM uzivatel");
        $sql->execute();

        return($sql->fetchColumn());
    }

    function getListByName($jmeno=NULL){
        if ($_GET[limit]<1)
            $i= 0;
        else
            $i = $_GET[limit];

        if ($jmeno)
        {
            $filtr = "WHERE jmeno LIKE '%$jmeno%'";
        }
        else
        {
            $limit = "LIMIT $i,30";
        }

        $sql=$this->prp("SELECT iddodavatel,jmeno,ulice,mesto,psc,ic,dic,kontakt,telefon,email,vlozeno,zmeneno FROM dodavatel $filtr $limit");
        $sql->execute(array());
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }
}
?>
```



## Gui.php

```
<?php
class GUI extends Main {

    function show(){
    $data = new Uzivatel;
    echo "<div id=\"main\">";
    // echo "<h1>Systém v provozu prosím nezádávejte zkušební objednávky</h1>";
    echo "<div class=\"hlavicka\">";
    $kostner = array(665,570,552,642,653,1);

    echo "<span class=\"nadpis\">";

    //echo $_SESSION[logid];
    echo "Informační systém DZ a.s.</span>";
    if (in_array($_SESSION[logid],$kostner))
    {
        echo "<span class=\"obr\"><img src=\"img/\".$_SESSION[logid].\".jpg\" alt=\"pic\" width=\"120px\"/></span>";
    }
    echo "</div>";

    if ($_SESSION[logid]>0 AND strlen($_SESSION[hash])==40)
    {
        echo "<div class=\"pravý\">";
        echo "<div class=\"txt\">";
        if ($_GET[menu]=="")
        {
            $string = new String;
            $string->statsF();
        }
        if ($_GET[warning]==1){
            echo "<h2>Upozornění</h2>";
            echo "<div class=\"upozorneni\">Máte nedokončenou objednávku č. $_SESSION[cislo]. Pokud chcete založit novou předchozí dokončete nebo smažte</div>";
        }

    }

    elseif ($_GET[changePASS]==true)
    {
        if (strlen($_POST[heslo])>0)
        {
            $heslo = sha1($_POST[heslo]);
            $sql = $data->prp("UPDATE uzivatel SET heslo=? WHERE iduzivatel=?");
            $sql->execute(array("$heslo",$_SESSION[logid]));
            $data->sqlchyba($sql);
            echo "Heslo bylo změněno";
        }
        else
        {
            echo "<form action=\"index.php?changePASS=true&go=true\" method=\"post\">";
            $data->input("Nové heslo:",heslo,NULL);
            $data->button("Změnit heslo");
        }
    }

    if ($_GET[menu]){
    try{
    $data->testprava();
    }
    catch (Exception $e){
    $data->chyba($e);
    }

    if (!$e)
    {
    if ($_GET[menu]=="pridatuzivatele")
    {
    if ($_GET[novy] == "ano")
    {
    try {
    $data->zkontrolovat();
    }
    catch (Exception $e)
    {
    $data->chyba($e);
    }
    if (!$e)
    $data->addRow();
    $data->addPrava();
    $data->addStrediska();
    $data->addOddeleni();
    $data->uzivatelResult();
    }
    }

    $this->pridatuzivatele();
    }
    elseif ($_GET[menu]=="seznamy")
    {

    }
    elseif ($_GET[menu]=="smlouvy")
    {
        $hs = new hlavniSkupina;
```

```

$ps = new podskupina;
$ss = new spolecnosti;
$sm = new Smlouvy;
echo "<h1>Smlouvy administrace</h1>";
$this->smlouvyM();
if ($ _GET[pridat]==ano){
    if ($ _GET[co]=="hlavniskupinu")
        $this->hlavniskupinaF();
    elseif ($ _GET[co]=="podskupinu")
        $this->podskupinaF();
    elseif ($ _GET[co]=="stranu")
        $this->smluvniF();
    elseif ($ _GET[co]=="smlouvu")
        $this->smlouvaF();
    elseif ($ _GET[funkce]=="pridat")
    {
        try {
            $hs->checkAdd();
        }
        catch (Exception $e){
            $this->chyba($e);
        }
        if (!$e)
            $hs->addRow();
    }
    elseif ($ _GET[funkce]=="pridatpod")
    {
        try {
            $ps->checkAdd();
        }
        catch (Exception $e){
            $this->chyba($e);
        }
        if (!$e)
            $ps->addRow();
    }
    elseif ($ _GET[funkce]=="stranu")
    {
        try{
            $ss->checkAdd();
        }
        catch (Exception $e){
            $this->chyba($e);
        }
        if (!$e)
            $ss->addRow();
    }
    elseif ($ _GET[funkce]=="smlouvu")
    {
        try{
            $sm->check();
        }
        catch (Exception $e){
            $this->chyba($e);
        }
        if (!$e)
            $sm->addRow();
        else
            $this->smlouvaF();
    }
}
elseif ($ _GET[editovat]=="ano")
{
    if ($ _GET[co]=="smlouvu")
        $this->editovatsmlF();
}

}
elseif ($ _GET[menu]=="seznamobjednavek")
{
    if ($ _GET[smazat]==true)
    {
        $po = new Polozky;
        $po->deleteBySklad("$ _GET[id]","$ _GET[sklad]");
    }
    if ($ _GET[edit]==true)
    {
        $po = new Polozky;
        $po->editBySklad("$ _GET[id]","$ _GET[sklad]");
    }
    if ($ _GET[change]=="true")
    {
        $_SESSION[stredisko]="$ _POST[stredisko]";
        $obj = new Objednavky;
        $cislo = $obj->genInterniCislo();
        $_SESSION[cislo]=$cislo;
    }
}

echo "<h1>Seznam objednávek</h1>";
$this->seznamObjednavekM();
if (is_numeric($ _GET[prevzit]) AND $ _GET[prevzit]>0)
$this->prevzitF();
elseif ($ _GET[p]==true)
{
    $this->prevzalF();
}

```

```

$this->seznamObjednavekF();

if ($_GET[change]=="true")
{
    $_SESSION[stredisko]=$_POST[stredisko];
}
}
elseif ($_GET[menu]=="vytvoritobjednavku")
{
    if ($_GET[change]=="true")
    {
        $_SESSION[stredisko]=$_POST[stredisko];
        $obj = new Objednavky;
        $cislo = $obj->genInterniCislo();
        $_SESSION[cislo]=$cislo;
    }
    if ($_GET[smazat]=="true")
    {
        $tmp = new tempObjednavky;
        $tmp->delTemp();
        $_SESSION[cislo]=NULL;
    }
    elseif ($_GET[smazat]=="temp")
    {
        $tmp = new tempObjednavky;
        $tmp->delTempRow($_GET[temp]);
    }

    echo "<h1>Vytvoření objednávky</h1>";
    $obj = new Objednavky;

    if ($_GET[novy]!="ano")
    $this->vytvoreniObjednavkyF();

    if ($_GET[novy]=="ano"){
        $id= $obj->zarad();
        if (strlen($_POST[poznamka])>1)
        {
            $obt = new Objednavkytxt;

            $uzi = new Uzivatel;

            $jmeno = $uzi->getNameById($_SESSION[logid]);
            $obt->setvariables(null, $id, "$jmeno[0]: $_POST[poznamka] \\\n");
            $obt->insert();
        }
        $tmp = new TempObjednavky;
        $tmp->delTemp();
        $_SESSION[cislo]=NULL;
        $mo = new Meta_objednavky;
        $mo->setvar("$id","1",NULL,$_SESSION[logid]);
        $mo->insert();
    }
}
elseif ($_GET[menu]=="seznamuzivatelu")
{
    if ($_GET[akce]=="smazat")
    {
        $data->deleteUser($_GET[uzivatel]);
    }
    elseif ($_GET[akce]=="heslo")
    {
        $this->noveHesloF();
        if ($_GET[zmena]=="ano")
        {
            try {
                $data->editPassword($_GET[uzivatel]);
            }
            catch (Exception $e)
            {
                $data->chyba($e);
            }
        }
    }
    else
    {
        $uziv = $data->getListByName($_POST[jmeno]);
    }

    echo "<form action=\"index.php?menu=seznamuzivatelu\" method=\"post\">";
    $data->input("Filtrovat jméno","jmeno",NULL);
    $data->button("Filtrovat");
    echo "</form>";

    echo "<table><tr><th>Číslo</th><th>Uživatel</th><th>Jmeno</th><th>Limit</th><th>Osobni cislo</th><th></th></tr>";
    foreach ($uziv AS $info){
        echo
        "<tr><td>$info[0]</td><td>".substr($info[1],0,16)."</td><td>".substr($info[2],0,64)."</td><td>".number_format($info[5],2,'
        ',' ')."</td><td>$info[6]</td>
        <td><img src=\"/pic/edit.png\" alt=\"Editovat\" width=\"32px\" title=\"Editovat\"/>
        <a href=\"index.php?menu=seznamuzivatelu&akce=smazat&uzivatel=$info[0]\">
        <img src=\"/pic/delete.png\" alt=\"Smazat\" width=\"32px\" title=\"Smazat\"/></a>
        <a href=\"index.php?menu=seznamuzivatelu&akce=heslo&uzivatel=$info[0]\">
        <img src=\"/pic/pass.png\" width=\"32px\" alt=\"změnit heslo\" title=\"změnit heslo\" /></a>
        </td></tr>";
    }
}

```



```

        $obt = new Objednavkytxt;
        $uzi = new Uzivatel;

        $jmeno = $uzi->getNameById($_SESSION[logid]);
        $poznold = $obt->get($_GET[cislo]);

        //echo $jmeno[0];
        //echo $poznold[0];
        //echo $_GET[cislo];

        $poznamka = $obt->prepare($_POST[poznamka]);

        //echo "$poznamka";
        $obt = new Objednavkytxt;
        $poz = $obt->get($_GET[cislo]);
        if (count($poz)>1)
        $obt->update("$_GET[cislo]", $poznold[0]."$jmeno[0]".": ". "$poznamka."\\n");
        else
        {
            $obt->setvariables(NULL, "$_GET[cislo]", "$jmeno[0]".": ". "$poznamka."\\n", NULL);
            $obt->insert();
        }

        try {
            $ob->checkShow();
        }
        catch (Exception $e) {
            $this->chyba($e);
        }
        if (!$e)
            $this->objednavkyDetail();
    }

    if ($_GET[schvalit]=="ok")
    {
        try {
            $ob->checkChangeStatus();
        }
        catch (Exception $e) {
            $this->chyba($e);
        }
        if (!$e)
        {
            $ob->changeStatus();
            $mo = new Meta_objednavky;
            $mo->setvar("$_GET[cislo]", "$_GET[stav]", NULL, $_SESSION[logid]);
            $mo->insert();
        }
        $this->schvalitM();
        if ($_GET[seznam]=="old")
            $old = true;
        else
            $old = false;

        $this->schvalitF($old);

    }
    elseif ($_GET[menu]=="sklad")
    {
        $this->menuSkladM();

        if ($_GET["seznam"]=="ano")
        {
            if ($_GET["edit"]==true)
            {
                $data = $_POST[pole];
                $data = explode(";", "$data");
                $sk = new Sklad;
                foreach ($data AS $info)
                {
                    $nazev = "nazev".$info;
                    $cena = "cena".$info;
                    $skupina = "skupina".$info;
                    $katalog = "katalog".$info;
                    $dodavatel = "dodavatel".$info;
                }
                //echo $skupina;
                //echo $katalog;
                $nazev = $_POST[$nazev];
                $cena = $_POST[$cena];
                $skupina = $_POST[$skupina];
                $katalog = $_POST[$katalog];
                $dodavatel = $_POST[$dodavatel];
                //echo $dodavatel;
                $cislo = $info;
                //echo $skupina;
                //echo $katalog;
                $sk->idskupiny = $skupina;
                $sk->cena = $cena;
                $sk->cislo = $katalog;
                $sk->idsklad = $cislo;
                $sk->nazev = $nazev;
                $sk->dodavatel = $dodavatel;
            }
        }
    }

```

```

        $sk->update();

    }
    echo "Sklad byl aktualizován";
}
$sklad = new Sklad;
$data = $sklad->getListByDodavatel("$_POST[dodavatel]");

$do = new Dodavatel;
$seznam = $do->getList();
echo "<form action='index.php?menu=sklad&seznam=ano&change=dodavatel' method='post'>";
echo "<select name='dodavatel'>";
foreach ($seznam AS $dodavatel)
{
    echo "<option value='\$dodavatel[iddodavatel]'>\$dodavatel[jmeno]</option>";
}
echo "</select>";
$this->button("Změnit");
echo "Výpis může trvat i několik minut!";
echo "</form>";
echo "<table>";
echo "<form action='index.php?menu=sklad&seznam=ano&edit=true' method='post'>";
echo "<tr><th>Jméno</th><th>Cena</th><th>Katalog</th><th>Skupina</th><th>Dodavatel</th></tr>";
$pole = array();
$sk = new Skupiny;
$do = new Dodavatel;
$dodavatel = $do->getList();
$skupiny = $sk->getList();
foreach ($data AS $info)
{
    echo "<tr><td class='male'>";

    echo "<input type='text' name='navez'.".$info[idsklad]."\\" value='\$info[0]' size='60' /></td>";

    echo "<td><input type='text' name='cena'.".$info[idsklad]."\\" value='\$info[3]' size='8' /></td>";

    echo "<td><input type='text' name='katalog'.".$info[idsklad]."\\" value='\$info[4]' size='8' /></td>";

    echo "<td>";
    echo "<select name='skupina'.".$info[idsklad]."\\">";
    $selected = "";
    foreach ($skupiny AS $skup)
    {
        if ($skup[idskupiny]==$info[idskupiny])
            $selected = "selected";
        else
            $selected = "";
        echo "<option value='\$skup[idskupiny]' \$selected>\$skup[nazev]</option>";
    }
    echo "</select>";
    echo "</td>";

    echo "<td>";
    echo "<select name='dodavatel'.".$info[idsklad]."\\">";
    $xselected = "";
    foreach ($dodavatel AS $dod)
    {
        if ($dod[iddodavatel]==$info[dodavatel])
            $xselected = "selected";
        else
            $xselected = "";
        echo "<option value='\$dod[iddodavatel]' \$xselected>\$dod[jmeno]</option>";
    }
    echo "</select>";
    echo "</td>";
    echo "</tr>";

    $pole[] = $info[idsklad];
}
echo "</table>";

$pole = implode(";", $pole);
echo "<input type='hidden' value='\$pole' name='pole'>";
$this->button("Upravit zboží");
echo "</form>";
}
if ($_GET[pridat]=="ano")
{
    if ($_GET[add]=="ano")
    {
        $sklad = new Sklad;
        try
        {
            $sklad->check();
        }
        catch (Exception $e)
        {
            $this->chyba($e);
        }
        if (!$e){
            $sklad->addRow();
            $this->oznameni("Zboží bylo vloženo");
        }
    }
}

```

```

        $this->novySkladF();
    }

    if ($_GET[skupina]=="list")
    {
        $sk = new Skupiny;
        if ($_GET[funkce] == "smazat")
        {
            $sk->deleteRow($_GET[id]);
        }

        $data = $sk->getList();
        echo "<table><tr><th>Číslo skupiny</th><th>Název skupiny</th><th></th>";
        foreach($data AS $info)
        {
            echo "<tr><td>$info[0]</td><td><span title=\"\$info[2]\">$info[1]</span></td><td>Editovat
<a href=\"index.php?menu=sklad&skupina=list&funkce=smazat&id=$info[0]\">Smazat</a></td></tr>";
        }
        echo "</table>";
    }

    if ($_GET[skupina]=="nova")
    {
        $sk = new Skupiny;
        $this->novaSkupinaF();
    }

    if ($_GET[vloz]=="novy")
    {
        try {
            $sk->check();
        }
        catch (exception $e)
        {
            $sk->chyba($e);
        }
        if (!$e)
            $sk->addRow();
    }

    if ($_GET[dodavatel]=="list")
    {
        $dod = new Dodavatel;
        $seznam = $dod->getListByName("$_POST[jmeno]");

        echo "<form action=\"index.php?menu=sklad&dodavatel=list\" method=\"post\">";
        $data->input("Filtrovat jméno", "jmeno", NULL);
        $data->button("Filtrovat");
        echo "</form>";

        echo "<table><tr><th>Číslo</th><th>Jmeno</th><th>Kontaktní osoba</th><th>Email</th><th>Telefon</th><th></th></tr>";
        foreach ($seznam AS $info) {
            echo "<tr><td>$info[0]</td><td>". $info[1]. "</td><td>$info[7]<td>". $info[9]. "</td><td>$info[8]</td>
<td><img src=\"/pic/edit.png\" alt=\"Editovat\" width=\"32px\" title=\"Editovat\"/>
<a href=\"index.php?menu=skladu&dodavatel=list&akce=smazat&uzivatel=$info[0]\">
<img src=\"/pic/delete.png\" alt=\"Smazat\" width=\"32px\" title=\"Smazat\"/></a>
</td></tr>";
        }
        echo "</table>";
        $radku = $dod->getDataCount();

        $stranek = ceil($radku/30);
        for ($i=0;$i<$stranek;$i++)
        {
            echo "<span class=\"stranek\"><a
href=\"index.php?menu=sklad&dodavatel=listu&limit=.". ($i*30). "\">". ($i+1). "</a></span>";
        }
    }

    if ($_GET[dodavatel]=="novy")
    {
        {
            $dod = new Dodavatel;
            if ($_GET[pridat]=="ano")
            {
                try
                {
                    $dod->check();
                }
                catch (Exception $e)
                {
                    $dod->chyba($e);
                }
                if (!$e)
                {
                    $dod->addRow();
                    echo "<div class=\"oznameni\">Dodavatel <strong>$_POST[jmeno]</strong> byl přidán</div>";
                }
            }
        }
    }

```

```

        $this->novyDodavatelF();
    }
}
}
echo "</div>";
echo "</div>";

echo "<div class=\"menu\">";
echo "<div class=\"txt\">";
echo "Menu";

echo "</div>";
$this->menu();
echo "</div>";

}
elseif ($_GET[login] == true)
{
    try {
        $data->checkuser();
    }
    catch (Exception $e)
    {
        $data->chyba($e);
    }
    if (!$e)
    {
        $data->login();
    }
}
else
{
    echo "<div class=\"login\">";
    $this->login();
    echo "</div>";
}
else
{
    echo "<div class=\"login\">";
    $this->login();
    echo "</div>";
}

echo "<hr class=\"divaider\" />";
echo "<div id=\"paticka\">";
echo "<div id=\"txt\">";
echo "Dopravní zdravotnictví a.s. Developed by: Bc. Miroslav Pavlíček";
echo "</div>";
echo "</div>";

echo "</div>";
}

function login(){
    echo "<div class=\"loginform\">";
    echo "<form action=\"index.php?login=true\" method=\"post\">";
    echo "<fieldset><legend>Přihlášení do systému</legend>";

    $this->input("Uživatelské jméno","uzivatel",NULL);
    $this->input("Heslo","heslo",NULL,"password");

    $this->button("Přihlásit");
    echo "</fieldset>";
    echo "</form>";
    echo "</div>";

}

function menu(){
    $data = new Uzivatel;

    $prava = $data->getprava();

    $odkazy = $data->odkazy();

    $menu = $data->menu();

    $seznam = array();

    foreach ($prava AS $pravo){
        $seznam = $menu["$pravo"];
        foreach ($seznam AS $odkaz)

```



```

    {
        echo $odkazy[$odkaz];
    }

}

echo "<br/>";
echo "<a href='\"index.php?changePass=true\"'>Změna hesla</a>";
echo "<br/>";
echo "<a href='\"index.php?logout=true\"'><strong>Odhlásit</strong></a>";

}

function pridatuzivatele(){
    echo "<div class='\"standardform\"'>";
    $this->pridatUzivateleF();
    echo "</div>";
}

}

?>

```

## Guiforms.php

```

<?php
class guiForms Extends GUI{
    function noveHesloF(){
        echo "<form action='\"index.php?menu=seznamuzivatelu&akce=heslo&zmena=ano&uzivatel=$_GET[uzivatel]\"'
method='\"post\"'>";
        $this->input("Nové heslo",heslo,NULL);
        $this->button("Změnit");
        echo "</form>";
    }

    function menuSkladM(){
        echo "<h1>Sklad - číselník</h1>";
        echo "<div class='\"submenu\"'>";
        echo "<a href='\"index.php?menu=sklad&pridat=ano\"'>Přidat položku do skladu</a>";
        echo "<a href='\"index.php?menu=sklad&schvalit=ano\"'>Schválit skladové zboží</a>";
        echo "<a href='\"index.php?menu=sklad&seznam=ano\"'>Skladový seznam</a>";
        echo "<a href='\"index.php?menu=sklad&skupina=nova\"'>Přidat skupinu zboží</a>";
        echo "<a href='\"index.php?menu=sklad&skupina=list\"'>Seznam skupin</a>";
        echo "<a href='\"index.php?menu=sklad&dodavatel=novy\"'>Nový dodavatel</a>";
        echo "<a href='\"index.php?menu=sklad&dodavatel=list\"'>Seznam dodavatelů</a>";
        echo "</div>";
    }

    function novySkladF(){
        echo "<form action='\"index.php?menu=sklad&pridat=ano&add=ano\"' method='\"post\"'>";
        echo "<div class='\"formpole\"'>";
        echo "<label for='\"dodavatel\"'>Dodavatel</label><select name='\"dodavatel\"' id='\"dodavatel\"'>";
        $dod = new Dodavatel;
        $data = $dod->getList();
        foreach($data AS $info)
        {
            if ($_POST[dodavatel]==$info[0])
                echo "<option value='\"$info[0]\"' selected>$info[1]</option>";
            else
                echo "<option value='\"$info[0]\"'>$info[1]</option>";
        }
        echo "</select>";
        echo "</div>";
        echo "<div class='\"formpole\"'>";
        echo "<label for='\"skupina\"'>Skupina</label><select name='\"skupina\"' id='\"skupina\"'>";
        $sk = new Skupiny;
        $data = $sk->getList();
        foreach ($data AS $info)
        {
            if ($_POST[skupina]==$info[0])
                echo "<option value='\"$info[0]\"' selected>$info[1]</option>";
            else
                echo "<option value='\"$info[0]\"'>$info[1]</option>";
        }
        echo "</select>";
        echo "</div>";
        $this->input("Název zboží",navez,$_POST[navez]);
        $this->input("Jednotky",jednotky,$_POST[jednotky]);
        $this->input("Cena",cena,$_POST[cena]);
        $this->input("Katalogové číslo",cislo,$_POST[cislo]);
        echo "<div class='\"formpole\"'><label>Popis</label><textarea name='\"popis\"' cols='\"30\"'
rows='\"5\"'>$_POST[popis]</textarea></div>";
        echo "<input type='\"checkbox\"' name='\"viditelne\"' id='\"viditelne\"' value='\"1\"' checked/><label for='\"viditelne\"'
/>Ihned zobrazit</label><br/>";
        //echo "<input type='\"checkbox\"' name='\"lekarsky\"' id='\"lekarsky\"' value='\"1\"'><label for='\"lekarsky\"' />Jedná se o
lékařský materiál</label><br/>";
        //echo "<input type='\"checkbox\"' name='\"dph\"' id='\"dph\"' value='\"1\"'><label for='\"dph\"' />Snižené DPH</label><br/>";
        if (!$_POST[sazbadph])
            $sazba = 19;
        else
            $sazba = $_POST[sazbadph];
        $this->input("Sazba DPH",sazbadph,$sazba);
        $this->button("Vložit do skladu");
        echo "</form>";
    }
}

```

```

    }
    function novaSkupinaF(){
        echo "<form action=\"index.php?menu=sklad&skupina=nova&vloz=novy\" method=\"post\"/>";
        $this->input("Název skupiny",navez,NULL);
        $this->input("Popis skupiny",popis,NULL);
        $this->button("Vlož do databáze");
        echo "</form>";
    }
    function novyDodavatelF(){
        echo "<form action=\"index.php?menu=sklad&dodavatel=novy&pridat=ano\" method=\"post\"/>";
        $this->input("Jméno dodavatele",jmeno,"$_POST[jmeno]");
        $this->input("Ulice a číslo",ulice,"$_POST[ulice]");
        $this->input("Město",mesto,"$_POST[mesto]");
        $this->input("PSČ",psc,"$_POST[psc]");
        $this->input("IČ",ic,"$_POST[ic]");
        $this->input("DIČ",dic,"$_POST[dic]");
        $this->input("Kontaktní osoba",kontakt,"$_POST[kontakt]");
        $this->input("Telefon",telefon,"$_POST[telefon]");
        $this->input("Email",email,"$_POST[email]");
        $this->button("Přidat");
        echo "</form>";
    }
    function pridatUzivateleF(){
        echo "<form action=\"index.php?menu=pridatuzivatele&novy=ano\" method=\"post\"/>";
        echo "<fieldset><legend>Přidání nového uživatele</legend>";

        $this->input("Uživatelské jméno","uzivatel,$_POST[uzivatel]");
        $this->input("Heslo","heslo",NULL,"password");
        $this->input("Heslo znova","heslo2",NULL,"password");

        $this->input("Jméno a příjmení","jmeno,$_POST[jmeno]");
        $this->input("Email","email,$_POST[email]");
        $this->input("Telefon","telefon,$_POST[telefon]");
        $this->input("Limit CZK (pouze lidé, kteří mají právo schvalovat)","limit,$_POST[limit]");
        $this->input("Osobní číslo","osobni,$_POST[osobni]");
        echo "<div class=\"formpole\">";
        echo "<label for=\"stredisko\">Středisko</label>";
        echo "<select name=\"stredisko[]\" id=\"stredisko\" multiple>";
        $std = new Strediska;
        $xstrediska = array();
        $xstrediska = $_POST[stredisko];
        $strediska = $std->getList();
        foreach ($strediska AS $data)
        {
            if (@in_array($data[0],$xstrediska))
                echo "<option value=\"$data[0]\" selected>$data[2] - $data[3] ($data[4])</option>";
            else
                echo "<option value=\"$data[0]\">$data[2] - $data[3] ($data[4])</option>";
        }
        echo "</select>";
        echo "</div>";
        echo "<div class=\"formpole\">";
        echo "<label for=\"oddeleni\">Oddělení</label>";
        echo "<select name=\"oddeleni[]\" id=\"oddeleni\" multiple>";
        $odd = new Oddeleni;
        $strediska = $odd->getList();
        $xstrediska = array();
        $xstrediska = $_POST[oddeleni];
        foreach ($strediska AS $data)
        {
            if (@in_array($data[0],$xstrediska))
                echo "<option value=\"$data[0]\" selected>$data[1] ($data[2])</option>";
            else
                echo "<option value=\"$data[0]\">$data[1] ($data[2])</option>";
        }
        echo "</select>";
        echo "</div>";

        $prv = new Prava;
        $prava = $prv->getList();
        echo "<div class=\"formpole\">Práva</div>";
        foreach ($prava AS $data)
        {
            echo "<div>";
            $xprava = array();
            $xprava = $_POST[prava];
            if (@in_array($data[0],$xprava))
                echo "<input type=\"checkbox\" name=\"prava[]\" value=\"$data[0]\" id=\"$data[0]\" checked/><label for=\"$data[0]\">$data[2]</label>";
            else
                echo "<input type=\"checkbox\" name=\"prava[]\" value=\"$data[0]\" id=\"$data[0]\"/><label for=\"$data[0]\">$data[2]</label>";
            echo "</div>";
        }

        $this->button("Vložit uživatele");

        echo "</fieldset>";
        echo "</form>";
    }
}

function vytvoreniObjednavkyF(){
    $uz = new Uzivatel();
    $stredisko = $uz -> getStredisko();
}

```

```

        $st = new Strediska;
        $ob = new Objednavky;
if (count($stredisko)>1)
{
    echo "<form action=\"index.php?menu=vytvoritobjednavku&amp;change=true\" method=\"post\">";
    echo "<fieldset class=\"noshow\"><legend>Změnit středisko</legend>";
    echo "<select name=\"stredisko\">";
    $data = $st->getListArray($stredisko);

    foreach ($data AS $info)
    {
        if ($_SESSION[stredisko]==$info[0])
            $sel = "selected = \"selected\"";
        else
            $sel = "";
        echo "<option value=\"\$info[0]\" $sel>$info[2] $info[3]</option>";
    }
    echo "</select>";
    $this->button("Změnit");
    echo "</fieldset>";
    echo "</form>";

}
else
{
    $_SESSION[stredisko] = $stredisko[0][0];
    //print_r($stredisko);
    //echo $_SESSION[stredisko];
}
$stredisko = $st->getRow($_SESSION[stredisko]);
//print_r($stredisko);
echo "<div>Vybrané středisko: ".$stredisko[0][2]. " ".$stredisko[0][3]."</div>";
echo "Interní číslo objednávky:<strong>".$_SESSION[cislo]."</strong>";
if ($_SESSION[stredisko]>0){
    if (!$_SESSION[cislo]){
        $cislo = $ob->genInterniCislo();
        $_SESSION[cislo]=$cislo;
    }
}
echo "<div class=\"skupiny\">";
$sk = new Skupiny;
$stemp = new TempObjednavky;
$spo = new Polozky;
$na = new Nadskupiny();

$hl = new Hlskupiny();
$hlavniskupina = $hl->getAll();
    echo "<div class=\"submenu\">";
    echo "<hr/>";
    echo "<table class=\"tablemenu\">";
    $k=0;
    $tr=0;
    foreach ($hlavniskupina AS $info)
    {
        $k++;
        // echo $k;
if ($k==1)
{
    echo "<tr>";
    $tr = 1;
}
    echo "<td><a href=\"index.php?menu=vytvoritobjednavku&amp;hlkupina=$info[idhlkupiny]\">$info[nazev]</a></td>";
    if ($k==3)
{
    echo "</tr>";
    $k=1;
    $tr = 0;
}
}
if ($tr==1)
echo "<tr/>";
echo "</table>";
echo "</div>";
if ($_GET[hlkupina]>0){
echo "<hr/>";

    $nadskupina = $na->getAllByHl($_GET[hlkupina]);

    echo "<div class=\"submenu\">";

    echo "<table class=\"tablemenu\">";
    $tr=0;
    $k=0;

    foreach ($nadskupina AS $info)
    {
        $k++;
if ($k==1)
{
    echo "<tr>";
    $tr = 1;
}
    echo "<td><a href=\"index.php?menu=vytvoritobjednavku&amp;nadskupina=$info[idnadskupiny]amp;hlkupina=$_GET[hlkupina]\">$info[nazev]<
/a></td>";
if ($k==4)

```



```

GET[hlskupina]& nadskupina=$_GET[nadskupina]" method="post"><p class="inline"><input type="text"
name="mnozstvi" value="1" size="3" />;
echo "<input type="hidden" name="sklad" value="\$info[0]" />";
$this->button("Objednat");
echo "</p></form></td></tr>";

}
echo "</table>";
if ($_GET[hledat]!="ok"){
$počet = $sk->getDataCount($_GET[skupina]);
$stran = ceil($počet/30);
for ($i=0;$i<$stran;$i++)
{
$limit = $i*30;
echo "<span class="lb"><a
href="index.php?menu=vytvoritobjednavku& skupina=$_GET[skupina]& limit=$limit& hlskupina=$_GET[hlskupina]& na
dskupina=$_GET[nadskupina]">".($i+1)."</a></span>";
}
}
else
{
echo "<h1>Seznam nejčastěji objednávaných položek</h1>";
$data = $po->top10($_SESSION[logid]);
echo "<table>";
echo "<tr><th>Název</th><th>Detail</th><th>Cena</th><th></th><th></th></tr>";

foreach ($data AS $info)
{
//echo "<form action="index.php?menu=vytvoritobjednavku& skupina=0& add=true& limit=$_GET[limit]"
method="post">";
echo "<tr><td>$info[nazev]</td><td><span>$info[popis]</span></td><td>
class="male">".number_format($info[cena],2,""," ")." CZK </td><td><form
action="index.php?menu=vytvoritobjednavku& skupina=0& add=true& limit=$_GET[limit]" method="post"><p
class="inline"><input type="text" name="mnozstvi" size="3" value="\$info[mnozstvi]" />";
echo "<input type="hidden" name="sklad" value="\$info[idsklad]" />";
$this->button("Objednat");
echo "</p></form></td></tr>";

}
echo "</table>";
}
echo "<div id="polozky"></div>";
echo "<br/>";
echo "<div id="polozky"></div>";
$počet = $temp->getYoutCount($_SESSION[logid],"$_SESSION[cislo]");
if ($počet > 0){
echo "<form action="index.php?menu=vytvoritobjednavku& novy=ano" method="post">";
echo "<fieldset class="colored"><legend>Objednávka č. $_SESSION[cislo]</legend>";
echo "<div id="objednavka_list"><div>";
$data = $temp->getYourList($_SESSION[logid],"$_SESSION[cislo]");

echo "<table class="table">";
echo "<tr><th>Sk.č.</th><th>Název</th><th>Jednotky</th><th>Množství</th><th>Cena</th><th>Celková
cena</th><th></th></tr>";
// rint_r($data);
foreach ($data AS $info){
echo "<tr><td>$info[cislo]</td><td>$info[nazev]</td><td>$info[jednotky]</td><td>$info[mnozstvi]</td>
<td>".number_format($info[cena],2,""," ")."</td><td>".number_format($info[mnozstvi]*$info[cena],2,""," ")."</td><td>
<a
href="index.php?menu=vytvoritobjednavku& skupina=$_GET[skupina]& limit=$_GET[limit]& smazat=temp& temp=$info[
idsklad]">Smazat</a>
</td></tr>";
}
echo "</table>";

echo "</div></div>";
echo "<div class="poznamka"><label>Poznámka k objednávce</label><br><textarea name="poznamka" rows=3
cols=60></textarea></div>";
$this->button("Poslat objednávku na schválení");
echo "<input type="submit" value="Smazat objednávku" name="delete" onclick="this.form.action =
'index.php?menu=vytvoritobjednavku& smazat=true'" />";
echo "</fieldset></form>";
}
}

function seznamObjednavekF(){
$ob = new Objednavky;

if ($_GET[zobrazit]=="seznam")
{
echo "<h2>Položky v objednávce</h2>";
try {
$ob->checkShow();
}
catch (Exception $e){
$this->chyba($e);
}
if (!$e)
$this->objednavkyDetail();
echo "<br/>";
}
elseif ($_GET[zobrazit]=="seznamno")
{

```

```

        echo "<h2>Položky v objednávce</h2>";
        $this->objednavkyDetailNoedit();
        echo "<br/>";
    }

    try {
        $ob->checkstav();
    }
    catch (Exception $e)
    {
        $ob->chyba($e);
    }
    if (!$e){
        $uz = new Uzivatel();
        $stredisko = $uz -> getStredisko();

        $st = new Strediska;

        if (count($stredisko) > 1)
        {
            echo "<form action=\"index.php?menu=seznamobjednavek&amp;change=true&amp;stav=$_GET[stav]\" method=\"post\">";
            echo "<fieldset class=\"noshow\"><legend>Změnit středisko</legend>";
            echo "<select name=\"stredisko\">";
            $data = $st->getListArray($stredisko);
            foreach ($data AS $info)
            {
                if ($SESSION[stredisko]==$info[0])
                    $sel = "selected = \"selected\"";
                else
                    $sel = "";
                echo "<option value=\"\$info[0]\" $sel>$info[2] $info[3]</option>";
            }
            echo "</select>";
            $this->button("Změnit");
            echo "</fieldset>";
            echo "</form>";
        }
        else
        {
            $data = $st->getListStred("$SESSION[stredisko]");
            //print_r($data);
            //echo "<div>Vybrané středisko: ".$data[0][cislo]." ".$data[0][popis]."</div>";
            $stredisko = $st->getRow($SESSION[stredisko]);
            //print_r($stredisko);
            echo "<div>Vybrané středisko: ".$stredisko[0][2]." ".$stredisko[0][3]."</div>";

            $data = $ob->getMyListR();
            echo "<table><tr><th>Interní číslo</th><th>Stav</th><th>Vytvořena</th><th></th></tr>";
            foreach ($data AS $info){
                echo "<tr><td>";
                if ($info[stav]==1)
                    echo "<a
href=\"index.php?menu=seznamobjednavek&amp;zobrazit=seznam&amp;cislo=$info[idobjednavky]\">$info[interncislo]</a>";
                else
                    echo "<a
href=\"index.php?menu=seznamobjednavek&amp;zobrazit=seznamno&amp;cislo=$info[idobjednavky]\">$info[interncislo]</a>";
                echo "</td><td>".$ob->stav2text($info[stav])."</td><td>$info[kdy]</td><td>";
                if ($info[stav]==5 OR $info[stav]==10 OR $info[stav]==6)
                    echo "<span class=\"lb\"><a
href=\"index.php?menu=seznamobjednavek&amp;stav=1&amp;prevzit=$info[idobjednavky]\">Převzit</a></span>";

                $mo = new Meta_objednavky;
                $informace = $mo->getListByObjednavka($info[idobjednavky]);
                $infox = "";
                foreach($informace AS $inf)
                {
                    $infox=$infox.$inf[2]." ".$inf[1]." ".$inf[0]."\n";
                }
                //echo $infox;
                echo "<span class=\"lb\"><a href=\"#\" onclick=\"alert('\$infox.\')\">Detail</a></span>";

                $obt = new Objednavkytxt;
                $poznamka = $obt->get($info[idobjednavky]);
                //echo $info[idobjednavky];
                if (count($poznamka)>1)
                    echo "<span class=\"lb\"><a href=\"#\" onclick=\"alert('".trim($poznamka[0])."')\">Poznámka</a></span>";

            }
            echo "</td></tr>";
        }
    }
    echo "</table>";
}

function seznamObjednavekM(){
    echo "<div class=\"submenu\">";
    echo "<a href=\"index.php?menu=seznamobjednavek&amp;stav=1\">Rozpracované objednávky</a>&nbsp;&nbsp;&nbsp;";
    echo "<a href=\"index.php?menu=seznamobjednavek&amp;stav=2\">Hotové objednávky</a>";
    echo "</div>";
}

function schvalitF($old = FALSE)
{
    $ob = new Objednavky;

```

```

    $uz = new Uzivatel;
    $limit = $uz->getLimit($SESSION[logid]);
    echo "<h1>Schvalovací limit je ".number_format($limit,0,""," ")." CZK, větší částky pouze odsouhlasíte</h1>";
    echo "<table>";
    echo "<tr><th>Uživatel</th><th>Středisko</th><th>Datum</th><th>Interní
číslo</th><th>Cena</th><th>Stav</th><th></th></tr>";
    $list = $ob->getMyAcceptList($old);
    echo "</table>";
}

function schvalitM(){
    echo "<div class=\"submenu\">";
    echo "<a href=\"index.php?menu=schvalit\">Seznam na schválení</a>&nbsp;&nbsp;&nbsp;";
    echo "<a href=\"index.php?menu=schvalit&amp;seznam=old\">Schválené objednávky</a>";
    echo "</div>";
}

function objednavkyDetail(){
    $obj = new Objednavky;
    $pol = new Polozky;

    $objednavka = $obj->getDetail($_GET[cislo]);
    $polozky = $pol->getGroupDetail($_GET[cislo]);

    // print_r($polozky);

    echo "<div class=\"objednavka\">";
    echo "<div class=\"objednavatel\">$objednavka[jmeno] ($objednavka[uzivatel])</div>";
    echo "<div class=\"cisloobjednavky\">Číslo objednávky: $objednavka[internicislo]</div>";
    echo "<div class=\"polozky\">";
    echo "<table>";
    echo "<tr><th>Číslo</th><th>Název</th><th>Množství</th><th>Jednotky</th><th>Cena</th><th>Celková
cena</th><th></th></tr>";
    foreach($polozky AS $info){
        // print_r($polozky);
        echo "<tr><td>$info[cislo]</td><td>";
        echo "<a href=\"#\" onclick=\"alert('\$info[jmeno].' - '.trim(\"$info[popis]\").'\\')\">".$info[nazev]."</a>";
        echo "</td><td>";
    }

    echo "<form
action=\"index.php?menu=$_GET[menu]&amp;cislo=$_GET[cislo]&amp;edit=true&amp;cislo=$_GET[cislo]&amp;objednavka=ukaz&amp;id
=$info[idobjednavky]&amp;sklad=$info[idsklad]\" method=\"post\"><p class=\"inline\"><input type=\"text\" name=\"mnozstvi\"
value=\"\".$info[mnozstvi].\"\" size=\"5\" />";
    $this->button("Změnit");

    echo "<p></form></td><td>$info[jednotky]</td><td>".number_format($info[cena],0,""," ")."
CZK</td><td>".number_format($info[sum],0,""," ")." CZK</td><td>

<span class=\"lb\"><a
href=\"index.php?menu=$_GET[menu]&amp;zobrazit=$_GET[zobrazit]&amp;cislo=$_GET[cislo]&amp;smazat=true&amp;objednavka=$_GET
[objednavka]&amp;id=$info[idobjednavky]&amp;sklad=$info[idsklad]\">Odstranit</a></span>";

    echo "</td></tr>";

}

echo "</table>";
echo "<form
action=\"index.php?menu=$_GET[menu]&amp;cislo=$_GET[cislo]&amp;poznamka=true&amp;objednavka=ukaz&amp;id=$info[idobjednavky
]\" method=\"post\">";
echo "<fieldset><legend>Poznámka</legend>";
echo "<textarea name=\"poznamka\" cols=\"40\" rows=\"3\"></textarea><br/>";
$this->button("Přidat poznámku");
echo "</fieldset>";
echo "</form>";
echo "</div>";
echo "</div>";
}

function objednavkyDetailNoedit(){
    $obj = new Objednavky;
    $pol = new Polozky;

    $objednavka = $obj->getDetail($_GET[cislo]);
    $polozky = $pol->getGroupDetail($_GET[cislo]);

    // print_r($polozky);

    echo "<div class=\"objednavka\">";
    echo "<div class=\"objednavatel\">$objednavka[jmeno] ($objednavka[uzivatel])</div>";
    echo "<div class=\"cisloobjednavky\">Číslo objednávky: $objednavka[internicislo]</div>";
    echo "<div class=\"polozky\">";
    echo "<table>";
    echo "<tr><th>Číslo</th><th>Název</th><th>Množství</th><th>Jednotky</th><th>Cena</th><th>Celková cena</th></tr>";
    foreach($polozky AS $info){
        echo "<tr><td>$info[cislo]</td><td><span title=\"\$info[jmeno] - \$info[popis]\">$info[nazev]</span></td>
<td>";
        echo "$info[mnozstvi]";
    }

    echo "</td><td>$info[jednotky]</td><td>".number_format($info[cena],0,""," ")."
CZK</td><td>".number_format($info[sum],0,""," ")." CZK</td><td>";

    echo "</td></tr>";

}

echo "</table>";
echo "</div>";
echo "</div>";

```

```

}

function objednatM(){
    echo "<div class=\"submenu\">";
    echo "<a href=\"index.php?menu=objednat\">Nové objednávky</a>&nbsp;";
    echo "<a href=\"index.php?menu=objednat&stav=aktivovane\">Vytvořené objednávky</a>&nbsp;";
    echo "<a href=\"index.php?menu=objednat&stav=dodane\">Dodané objednávky</a>";
    echo "</div>";
}

function objednatF(){
    echo "<h2>Seznam objednavek pro aktivaci</h2>";
    echo "<h3>Dodavatelé</h3>";
    $do = new Dodavatel;
    $seznam = $do->getListByName();
    echo "<form action=\"index.php?menu=objednat&dodavatel=change\" method=\"post\">";
    echo "<fieldset class=\"colored\"><legend>Změnit dodavatele</legend><select name=\"dodavatel\">";
    foreach ($seznam AS $info)
    {
        if ($info[iddodavatel]==$_SESSION[dodavatel] OR $info[iddodavatel]==$_POST[dodavatel])
            $selected = "selected=\"selected\"";
        else
            $selected = "";
        echo "<option value=\"{$info[iddodavatel]}\" $selected>{$info[jmeno]</option>";
    }
    echo "</select>";
    $this->button("Vybrat");
    echo "</fieldset>";
    echo "</form>";

    $st = new Strediska;
    $po = new Polozky;

    $skupiny = $st->getAllMyGroup($_SESSION[logid]);
    foreach ($skupiny AS $group)
    {
        $polepolozek = array();
        echo "<div class=\"skupina\">";
        echo "<div class=\"nadpis\">". $st->skupina($group[nemocnice]). "</div>";
        $strediska = $st->getAllMyBySkupiny($_SESSION[logid], $group[nemocnice]);
        $polozky = $po->getByDodavatel($_SESSION[dodavatel], $strediska);
        echo "<form action=\"index.php?menu=objednat&stav=aktivovane&update=true\" method=\"post\">";
        echo "<fieldset class=\"operaready\"><legend>Objednávka</legend>";
        echo "<table>";
        echo "<tr><th>Číslo</th><th>Název</th><th>Množství</th><th>Cena</th><th>Číslo
objednávky</th><th>Středisko</th><th>Objednal</th></tr>";
        foreach ($polozky AS $polozka)
        {
            echo "";
            $sumapolozek = $po->getByCisloZarizeni($polozka[cislo], $strediska);
            // print_r($sumapolozek);
            //if (in_array("$polozka[cislo]", $polepolozek)==false)
            //{

                $polepolozek[] = $polozka[cislo];
                $zmen = str_replace(array("."), array("_"), $polozka[cislo]);

                echo "<tr><td class=\"male\"><input type=\"hidden\" name=\"polozka[]\" value=\"{$polozka[idpolozky]}\"
/>{$polozka[cislo]</td><td class=\"male60\">{$polozka[nazev]</td><td>". $polozka[mnozstvi]. "</td><td>
<input type=\"text\" name=\"\$zmen\" value=\"{$polozka[cena]}\" size=\"8\" />";
                echo "<input type=\"hidden\" name=\"xpolozka[]\" value=\"\$zmen\" /></td>";
                echo "<td class=\"male60\">{$polozka[internicislo]</td><td class=\"male60\">{$polozka[stredisko]</td><td
class=\"male60\">{$polozka[uzivatel]</td></tr>";

                // }
            }
        }
        echo "</table>";
        echo "<br/>";
        $this->button("Potvrdit odeslání objednávky");
        echo "</fieldset>";
        echo "</form>";
        echo "<pre>";
        //print_r($polozky);
        echo "</pre>";

        echo "</div>";

    }

}

function updateF(){
    $polozky = array();
    $polozky = $_POST[polozka];
    $xpolozka = array();
    $xpolozka = $_POST[xpolozka];

    $po = new Polozky;
    $ob = new Objednavky;
}

```



```

$sk = new Sklad;

//print_r($polozky);
//print_r($xpolozka);

$xcena = array();
foreach ($xpolozka AS $data)
{
    $scena = $_POST["$data"];
    //echo "Cena".$scena;
    // echo $data;
    $data = str_replace(array("_"),array("."),$data);
    $xcena[$data]=$scena;

    foreach ($polozky AS $mdata)
    {

        $spo->changeStatus($mdata,4,$data);
        //echo "Mdaata:".$mdata;
        //echo "Sklad:".$sklad;
        //echo "Cena:".$scena;
        //echo $info;
        $objednavka = $spo->getIdObjednavky("$mdata");
        $ob->changeStatus(10,"$objednavka","$scena");

        $mo = new Meta_objednavky;
        //echo "Objednavka: $objednavka";
        $mo->setvar("$objednavka","4",NULL,$_SESSION[logid]);
        $mo->insert();
    }
}
//print_r($xcena);
foreach ($polozky AS $mdata)
{
    $sklad = $spo->getIdSklad($mdata);
    $xcislo = $sk->getById($sklad);
    $scena = $xcena[$xcislo[cislo]];
    /* echo "Mdata:$mdata<br/>";
    echo "Sklad:$sklad<br/>";
    echo "Cena:$scena<br/>";*/
    $spo->setPrice($mdata, $sklad, $scena);
}

$ob->checkAll(4,10);

echo "Objednávka uskutečněna";

}
function aktivovaneF(){
    echo "<h3>Odeslané objednávky</h3>";
    echo "<h3>Dodavatelé</h3>";

    $do = new Dodavatel;
    $st = new Strediska;
    $po = new Polozky;
    $seznam = $do->getListByName();
    echo "<form action=\"index.php?menu=objednat&addodavatel=change&stav=aktivovane\" method=\"post\">";
    echo "<fieldset class=\"colored\"><legend>Dodavatelé</legend><select name=\"dodavatel\">";
    foreach ($seznam AS $info)
    {
        if ($info[idodavatel]==$_SESSION[dodavatel] OR $info[idodavatel]==$_POST[dodavatel])
            $selected = "selected=\"selected\"";
        else
            $selected = "";
        echo "<option value=\"".$info[idodavatel].\" $selected>$info[jmeno]</option>";
    }
    echo "</select>";

    $this->button("Změnit");
    echo "</fieldset>";
    echo "</form>";

    $skupiny = $st->getAllMyGroup($_SESSION[logid]);
    echo "<form action=\"index.php?menu=objednat&addodavatel=aktivovane&addodano=true\" method=\"post\">";
    echo "<fieldset><legend>Objednané zboží</legend>";
    echo "<table>";
    echo "<tr><th>Dodáno</th><th>Číslo</th><th>Název</th><th>Množství</th><th>Cena</th><th>Číslo objednávky</th></tr>";
    foreach ($skupiny AS $group)
    {
        $polepolozek = array();

        $strediska = $st->getAllMyBySkupiny($_SESSION[logid],$group[nemocnice]);
        $polozky = $po->getByDodavatel($_SESSION[dodavatel],$strediska,4);
        //print_r($polozky);
        foreach ($polozky AS $polozka)
        {

```

```

$sumapolozek = $po->getByCislo($polozka[cislo],4);
//echo $sumapolozek;
//print_r($sumapolozek);
/* if (in_array("$polozka[cislo]", $polepolozek)==false)
{*/

    $polepolozek[] = $polozka[cislo];
    echo "<tr><td><input type=\"hidden\" name=\"polozka[]\" value=\"$polozka[idpolozky]\" />";
    echo "<input type=\"checkbox\" name=\"$polozka[idpolozky]\" value=\"1\" />";
    echo "</td><td>$polozka[cislo]</td><td>$polozka[nazev]</td><td>$. $polozka[mnozstvi].</td><td>";
echo "$polozka[cena]";
    echo "<input type=\"hidden\" name=\"xpolozka[]\" value=\"$polozka[idpolozky]\" />";
    echo "</td><td>$polozka[internicislo]</td></tr>";

    // }

}

}

    echo "</table>";
$this->button("Potvrdit odesláni objednávky");
echo "</fieldset>";
echo "</form>";

}

function dodanoF(){
    $po = new Polozky;
    $ob = new Objednavky;
    $seznam = array();
    $seznam = $_POST[polozka];
    // print_r($seznam);
    foreach ($seznam AS $polozky)
    {
        $stest = $_POST[$polozky];
        if (is_numeric($stest)==true AND $stest==1)
        {
            $po->changeStatus($polozky,5);
            $stest = 0;

            $objednavka = $po->getIdObjednavky("$polozky");

            $mo = new Meta_objednavky;
            //echo "Objednavka: $objednavka";
            $mo->setvar("$objednavka", "5", NULL, $_SESSION[logid]);
            $mo->insert();

        }

    }

    $ob->checkAll(5,11);

    echo "Položky byly dodány";
}

function dodaneF(){
    echo "<h2>Vyberte člověka pro kterého chcete vytisknout protokol</h2>";
    echo "<div class=\"pnone\">";
    echo "<form action=\"index.php?menu=objednat&stav=dodane&filter=1\" method=\"post\">";
    echo "<fieldset class=\"colored\"><legend>Dodané zboží</legend>";
    echo "<label>Uživatel:</label>";
    echo "<select name=\"uzivatel\">";
    $obj = new Objednavky;
    $uzi = new Uzivatel;
    $str = new Strediska;
    $data = $obj->getUzivatelByStav(NULL);
    //print_r($data);
    $pol = new Polozky;
    foreach ($data AS $info){
        $jmeno = $uzi->getNameById($info[0]);
        echo "<option value=\"$info[0]\">$jmeno[0]</option>";
    }

    echo "</select>";
    $this->button("Vybrat");
    echo "</fieldset>";
    echo "</form>";
    echo "</div>";

    if ($_GET[filter]==2)
    {
        // $obje = new Objednavky;
        // $polo = new Polozky;

        $moje = $obj->getMy($_POST[user]);
        foreach ($moje AS $objednavky)
        {
            $obj->changeStatus(7,$objednavky[0]);
            $mopo = $pol->getAllMy(5,$objednavky[0]);
            foreach ($mopo AS $polozky)
            {
                $pol->changeStatus($polozky[0],7);
            }
        }
    }
}

```

```

}

if ($_POST[uzivatel]>0 AND is_numeric($_POST[uzivatel])==true)
{
    $data = $obj->getMy($_POST[uzivatel]);
    // print_r($data);
    foreach ($data AS $info)
    {
        $detail = $obj->getDetail($info[idobjednavky]);
        //print_r($detail);
        $stredisko = $str->getDetail($detail[idstrediska]);

        //print_r($stredisko);
        echo "<table class='mtable'>";
        echo "<tr><td>Jméno:</td><td><strong>$detail[jmeno]</strong></td><td>Středisko</td><td>$stredisko[cislo]
($stredisko[popis])</td><td>Vytvořena:</td>";
        echo "<td>.date('H.m.Y h:i:s', strtotime($detail[kdy]))</td>";
        echo "</tr>";
        $ja = $uzi->getNameById($_SESSION[logid]);
        echo "<tr><td colspan='2'>Vytvořil:</td><td colspan='2'>$ja[jmeno]</td><td>Vytvořeno:</td><td>.date('H.m.Y
h:i:s')</td></tr>";
        echo "<tr><td colspan='2'>Číslo objednávky:</td><td colspan='4'><span
class='big'>$detail[internicislo]</span></td></tr>";
        echo "<tr><th>Katalogové
číslo</th><th>Název</th><th>Cena</th><th>Množství</th><th>Jednotky</th><th>Suma</th></tr>";
        $polozky = $pol->getIdByObjednavka($info[idobjednavky]);
        foreach ($polozky AS $pk)
        {
            //echo $pk[idpolozky];
            $mdetail = $pol->getmDetail($pk[idpolozky]);
            //print_r($mdetail);
            $mdetail = $mdetail[0];
            //print_r($mdetail);
            echo
"<tr><td>$mdetail[cislo]</td><td>$mdetail[nazev]</td><td>.round($mdetail[cena],2)</td><td>$mdetail[mnozstvi]</td><td>$m
detail[jednotky]</td><td>.round($mdetail[sum],2)</td></tr>";
        }
        echo "<tr><td colspan='5'>Převzal:</td><td colspan='5'></td></tr>";
        echo "</table>";
        echo "<br/><hr/><br/>";
    }
    echo "<div class='pnone'>";
    echo "<form action='index.php?menu=objednat&stav=dodane&filter=2' method='post'>";
    echo "<fieldset class='operaready'><legend>Schválení</legend>";
    echo "<input type='hidden' name='user' value='$_POST[uzivatel]' />";
    $this->button("Hromadné přijetí");
    echo "</fieldset>";
    echo "</form>";
    echo "</div>";
}

}

function prevzitF(){
    $po = new Polozky;
    $cislo = $_GET[prevzit];
    // $data = $po->getIdByObjednavka($cislo);
    echo "<form action='index.php?menu=seznamobjednavek&stav=1&p=true&o=$_GET[prevzit]' method='post'>";
    echo "<table>";
    echo
"<tr><th>Číslo</th><th>Název</th><th>Cena</th><th>Množství</th><th>Jednotky</th><th>Celkem</th><th>Přijal</th></tr>";
    // print_r($data);
    $data = $po->getDetail($cislo,5);
    foreach ($data AS $detail) {
        echo "<tr><td>$detail[cislo]</td><td>$detail[nazev]</td><td>$detail[cena]</td><td>$detail[mnozstvi]</td>
<td>$detail[cena]</td><td>$detail[sum]</td>
<td><input type='checkbox' name='dod$detail[idpolozky]' value=1 />$detail[internicislo]</td></tr>";
        $sumasum = $sumasum+$detail[sum];
        $xseznam = $xseznam.", ".$detail[idpolozky];
    }
    echo "</table>";
    echo "<div class='suma'>";
    echo "Celková cena: ".number_format($sumasum,2,".", " ")." CZK";
    echo "</div>";
    echo "<input type='hidden' name='seznam' value='$xseznam' />";
    $this->button("Převzít objednávku");
    echo "</form>";
}

function prevzalf(){
    $po = new Polozky;
    $ob = new Objednavky;
    $cislo = $_GET[o];
    $seznam = $_POST[seznam];
    $seznam = explode(",",$seznam);
    // print_r($seznam);
    foreach ($seznam AS $info){
        $value = "dod$info";
        //echo $value;
        $data = $_POST[$value];
        // echo $data;
        if ($data == 1)
            $po->changeStatus($info,7);
            $objednavka = $po->getIdObjednavky("$info");

            $mo = new Meta_objednavky;
            //echo "Objednavka: $objednavka";
            $mo->setvar("$objednavka", "7", NULL, $_SESSION[logid]);

```

```

    $mo->insert();
}
$ob->checkAll(7,6);
}
function mschvalitF(){
    $ob = new Objednavky;
    $uz = new Uzivatel;
    // $limit = $uz->getLimit($SESSION[logid]);
    echo "<h1>Manažerské schvalování - neexistence limitu pro schálení</h1>";
    echo "<table>";
    echo "<tr><th>Uživatel</th><th>Datum</th><th>Interní číslo</th><th>Cena</th><th></th></tr>";
    $list = $ob->getAcceptedList();
    echo "</table>";
}
function smlouvyM(){
    echo "<div class='submenu'>";
    echo "<a href='index.php?menu=smlouvy&pridat=ano&co=hlavniskupinu'>Přidat hlavní skupinu</a>";
    echo "<a href='index.php?menu=smlouvy&pridat=ano&co=podskupinu'>Přidat podskupinu </a>";
    echo "<a href='index.php?menu=smlouvy&pridat=ano&co=stranu'>Přidat smluvní stranu</a>";
    echo "<a href='index.php?menu=smlouvy&pridat=ano&co=smlouvu'>Přidat smlouvu</a>";
    echo "<a href='index.php?menu=smlouvy&editovat=ano&co=smlouvu'>Editovat smlouvu</a>";
    echo "</div>";
}
function hlavnikupinaF(){
    echo "<form action='index.php?menu=smlouvy&pridat=ano&funkce=pridat' method='post'>";
    $this->input("Název",navez,$_POST[navez]);
    $this->button("Přidat skupinu");
    echo "</form>";
}
function podskupinaF(){
    $hs = new hlavnikupina;
    echo "<form action='index.php?menu=smlouvy&pridat=ano&funkce=pridatpod' method='post'>";
    $this->input("Název",navez,$_POST[navez]);
    $data = $hs->getList();
    echo "<label>Hlavní skupina</label><br><select name='skupina'>";
    foreach ($data AS $info)
    {
        echo "<option value='\$info[0]'\> \$info[1]</option>";
    }
    echo "</select><br>";
    $this->button("Přidat skupinu");
    echo "</form>";
}
function smluvniF(){
    echo "<form action='index.php?menu=smlouvy&pridat=ano&funkce=stranu' method='post'>";
    $this->input("Název",navez,$_POST[navez]);
    echo "<label>Popis</label><br><textarea name='popis'\>$_POST[popis]</textarea><br>";
    $this->button("Přidat smluvní stranu");
    echo "</form>";
}
function smlouvaF(){
    $ps = new Podskupina;
    $ss = new Spolecnosti;
    echo "<form action='index.php?menu=smlouvy&pridat=ano&funkce=smlouvu' enctype='multipart/form-data' method='post'>";
    $data = $ps->getList();
    echo "<label>Vyberte skupinu</label><br>";
    echo "<select name='skupina'>";
    foreach ($data AS $info){
        echo "<option value='\$info[0]'\> \$info[1]</option>";
    }
    echo "</select><br>";
    $data = $ss->getList();
    echo "<label>Vyberte smluvní stranu</label><br>";
    echo "<select name='spolecnost'\>";
    foreach ($data AS $info){
        echo "<option value='\$info[0]'\> \$info[1]</option>";
    }
    echo "</select>";
    $this->input("Název smlouvy",navez,$_POST[navez]);
    echo "<label>Popis</label><br><textarea name='popis' cols='40' rows='8'\>$_POST[popis]</textarea><br>";
    $this->input("Příloha (jen PDF)",priloha,$_POST[priloha],file);
    $this->button("Přidat smlouvu");
}
}
function editovatsmlF(){
    $sm = new Smlouvy;
    echo "<table>";
    echo "<tr><th>Společnost</th><th>Název</th><th>Skupina</th><th>Smlouva</th><th>Nástroje</th></tr>";
    foreach ($sm->getList() AS $info)
    {
        echo "<tr><td>$info[spolecnost_navez]</td><td>$info[navez]</td><td>$info[skupina_navez]</td><td><a href='smlouvy/$info[priloha]'\>$info[priloha]</a></td><td>Editovat</td></tr>";
    }
    echo "</table>";
}
}
?>

```



```

$scim = array('a','c','d','e','e','i','n','r','s','t','u','u','o','-',' ','z','-
','a','c','d','e','e','i','n','r','s','t','u','u','o','z','y','Y','o','O','E','U','A','a','e','u','L','l','s',' ','-
','E','e');
$string = str_replace($co,$scim,$string);
if ($slower == true)
$string = strtolower($string);
return($string);
}

function input($label,$name,$value,$type="text",$id=""){
if (!$id)
$id = $this->ntext("$name");
if (!$name)
$name = $this->ntext("$label");
echo "<div class=\"formpole\">";
echo "label for=\"\$id\">$label</label><input type=\"\$type\" name=\"\$name\" value=\"\$value\" id=\"\$id\" size=\"40\"/>";
echo "</div>";
}

function button($value){
echo "<input type=\"submit\" name=\"b1\" value=\"\$value\" />";
}

function chyba($e){
echo "<div class=\"chyba\">";
echo "<div class=\"hlavicka\"><span class=\"text\">Chyba čísla:<span class=\"number\">".$e->
getCode()."</span></span></div>";
echo "<div class=\"pic\">";
echo "<img src=\"pic/error.png\" width=\"64px\"/>";
echo "</div>";
echo "<div class=\"text\">".$e->getMessage()."</div>";
echo "<hr class=\"divaider\"/>";
echo "</div>";
}

}

function sqlchyba($sql){
$error = array();
$error = $sql->errorInfo();

if ($error[1] > 0){
echo "<div class=\"chyba\">";
echo "<div class=\"hlavicka\"><span class=\"text\">Chyba čísla:<span class=\"number\">".$error[1]."</span></span></div>";
echo "<div class=\"text\">".$error[2]."</div>";
echo "</div>";
}
}

}

function ip()
{
if ($this->validip($_SERVER["HTTP_CLIENT_IP"])) {
return $_SERVER["HTTP_CLIENT_IP"];
}
foreach (explode(",",$_SERVER["HTTP_X_FORWARDED_FOR"]) as $ip) {
if ($this->validip(trim($ip))) {
return $ip;
}
}
if ($this->validip($_SERVER["HTTP_X_FORWARDED"])) {
return $_SERVER["HTTP_X_FORWARDED"];
} elseif ($this->validip($_SERVER["HTTP_FORWARDED_FOR"])) {
return $_SERVER["HTTP_FORWARDED_FOR"];
} elseif ($this->validip($_SERVER["HTTP_FORWARDED"])) {
return $_SERVER["HTTP_FORWARDED"];
} elseif ($this->validip($_SERVER["HTTP_X_FORWARDED"])) {
return $_SERVER["HTTP_X_FORWARDED"];
} else {
return $_SERVER["REMOTE_ADDR"];
}
}

}

function validip($ip) {
if (!empty($ip) && ip2long($ip)!=-1) {
$reserved_ips = array (
array('0.0.0.0','2.255.255.255'),
array('10.0.0.0','10.255.255.255'),
array('127.0.0.0','127.255.255.255'),
array('169.254.0.0','169.254.255.255'),
array('172.16.0.0','172.31.255.255'),
array('192.0.2.0','192.0.2.255'),
array('192.168.0.0','192.168.255.255'),
array('255.255.255.0','255.255.255.255')
);

foreach ($reserved_ips as $r) {
$min = ip2long($r[0]);
$max = ip2long($r[1]);
if ((ip2long($ip) >= $min) && (ip2long($ip) <= $max)) return false;
}
return true;
} else {
return false;
}
}
}
}

```

```
function oznameni($string){
    echo "<div class='oznameni'><div class='hlavicka'>Oznameni</div><div class='text'>".$string.</div></div>";
}
?>
```

## Metaobjednavky.php

```
<?php
```

```
class meta_objednavky extends Main {
    var $idmeta_objednavky;
    var $idobjednavky;
    var $idstav;
    var $kdy;
    var $iduzivatele;

    function setvar($idobjednavky=NULL,$idstav=NULL,$kdy=NULL,$iduzivatele=NULL,$idmeta_objednavky=NULL)
    {
        $this->idmeta_objednavky = $idmeta_objednavky;
        $this->idobjednavky = $idobjednavky;
        $this->idstav = $idstav;
        $this->kdy = $kdy;
        $this->iduzivatele = $iduzivatele;
    }

    function insert()
    {
        //echo $this->iduzivatele;
        $sql = $this->prp("INSERT INTO meta_objednavky
        VALUES (
        NULL,
        ?,
        ?,
        NOW(),
        ?)");
        $sql->execute(array("$this->idobjednavky","$this->idstav","$this->iduzivatele"));
        $this->sqlchyba($sql);
    }

    function getListByObjednavka($id)
    {
        $sql = $this->prp("SELECT st.nazev,DATE_FORMAT(mo.kdy,'%k:%i:%s %e.%c.%Y'),u.jmeno FROM stav AS st, meta_objednavky AS
        mo, uzivatel AS u
        WHERE mo.idobjednavky = ? AND u.iduzivatel = mo.iduzivatele AND mo.idstav = st.idstav ORDER BY kdy
        ");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

}
?>
```

## Metauzivatel.php

```
<?php
```

```
class meta_uzivatel Extends Main {
    var $idmeta_uzivatel;
    var $iduzivatel;
    var $akce;
    var $kdy;
    var $ip;
    var $stav;
    var $chyba;
    var $popis;

    function addrow(){
        $this->ip = $this->ip();

        $data = $this->prp("INSERT INTO meta_uzivatel VALUES (NULL,?,?,?,NOW(),?,?,?,?)");
        $data->execute(array("$this->iduzivatel","$this->akce","$this->ip","$this->stav","$this->chyba","$this->popis"));
        $this->sqlchyba($data);
    }

    function getLastInfo()
    {
        $data = $this->prp("SELECT DATE_FORMAT(kdy,'%k:%i:%s %d.%m.%Y') AS datum,ip FROM meta_uzivatel WHERE
        akce='Přihlášení' ORDER BY kdy DESC LIMIT 2");
        $data->execute();
        $this->sqlchyba($data);
        return($data->fetchAll());
    }

}
?>
```

## nadskupiny.php

```

<?php

class nadskupiny extends Main {
    var $idnadskupiny;
    var $idhlskupiny;
    var $nazev;
    var $popis;

    function getAllByHl($hlavni)
    {
        $sql = $this->prp("SELECT nazev,idnadskupiny FROM nadskupiny WHERE idhlskupiny=? ORDER BY nazev");
        $sql->execute(array("$hlavni"));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

}
?>

```

## Objednavky.php

```

class Objednavky extends Main {
    var $idobjednavky;
    var $iduzivatel;
    var $idstrediska;
    var $kdy;
    var $stav;
    var $nemocnice =0;
    var $internicislo;

    function genInterniCislo(){
        $st = new Strediska;
        $data = $st->getRow("$_SESSION[stredisko]");
        $kod = $data[0][2];
        //echo $kod;
        $za = new Zarizeni;
        // $kod = $za->getColKod($kod);
        $rok = date("y");
        $cislo = $this->getRowLastIc();
        $cislo = substr("$cislo",-6,6);
        $temp = new tempObjednavky;
        $tempcislo = $temp->getLastIc();
        $tempcislo = substr("$tempcislo",-6,6);
        if ($tempcislo > $cislo)
        $cislo = $tempcislo;
        if (!$cislo)
        $cislo = "0";

        $cislo = $cislo+1;

        $cislo = str_pad("$cislo",6,"0",STR_PAD_LEFT);
        //echo "$rod$skod$cislo";
        return("$rok$skod$cislo");
    }

    function getRowLastIc(){
        $sql = $this->prp("SELECT internicislo FROM objednavky ORDER BY idobjednavky DESC LIMIT 1");
        $sql->execute();
        $this->sqlchyba($sql);
        return($sql->fetchColumn());
    }

    function zarad(){
        $temp = new tempObjednavky;
        $data = $temp->getOrderList();
        $id = $this->newRow();
        $po = new polozky;
        // print_r($data);
        $sk = new sklad;
        foreach ($data AS $info){
            $sklad = $sk->getById("$info[idsklad]");
            // print_r($sklad);
            // echo "Zaražuji";
            $po->addRow($id, $info[idsklad], "$sklad[nazev]", "$sklad[jednotky]", "$sklad[cena]", "$sklad[cislo]",
"$sklad[popis]", "$info[mnozstvi]",1);
        }

        $this->oznameni("Objednávka byla zařazena ke schválení. Dokud nebude schválena můžete objednávku editovat");

        return($id) ;
    }

    function newRow(){
        $this->db()->beginTransaction();
        $sql = $this->prp("INSERT INTO objednavky VALUES (NULL,?,?,?,NOW(),1,0,?)");
        $sql->execute(array("$_SESSION[logid]","$_SESSION[stredisko]","$_SESSION[cislo]"));
        $this->sqlchyba($sql);
        $sql2 = $this->prp("SELECT idobjednavky FROM objednavky ORDER BY idobjednavky DESC LIMIT 1");
        $sql2->execute();
        $this->sqlchyba($sql2);
        return($sql2->fetchColumn());
        $this->db()->commit();
    }

    function getMyListR(){
        /*stavy:
        * 1 - přijato ke schválení

```



```

* 2 - potvrzeno
* 3 - schváleno
* 4 - objednáno
* 5 - doručeno
* 6 - částečné převzato
* 7 - převzato
* 8 - stornováno
* 9 - zamítnuto
* 10 - částečně objednáno
* 11 - částečně dodáno

*/
$поволено = array("525","526");
if (in_array($_SESSION[logid],$поволено))
$uzivatel = "";
else
$uzivatel = "iduzivatel=$_SESSION[logid] AND";

if ($this->stav == 1)
$sql = $this->prp("SELECT internicislo,stav,kdy,idobjednavky FROM objednavky WHERE $uzivatel idstrediska=? AND (stav<7 OR
stav>9) ORDER BY kdy DESC");
else
$sql = $this->prp("SELECT internicislo,stav,kdy,idobjednavky FROM objednavky WHERE $uzivatel idstrediska=? AND stav>6 AND
stav<10 ORDER BY kdy DESC");
$sql->execute(array("$_SESSION[stredisko]"));
$this->sqlchyba($sql);
return($sql->fetchAll());
}

function checkStav(){
$this->stav = $_GET[stav];
if (!$this->stav)
$this->stav = 1;
if (is_numeric($this->stav)==false OR $this->stav<1 OR $this->stav > 2)
throw new Exception("Stav neodpovídá realitě",9001);
}

function stav2text($stav){
if ($stav == 1)
return("Přijato ke schválení");
if ($stav == 2)
return("Potvrzeno");
if ($stav == 3)
return("Schváleno");
if ($stav == 4)
return("Objednáno");
if ($stav == 5)
return("Doručeno");
if ($stav == 6)
return("Částečné převzato");
if ($stav == 7)
return("Převzato");
if ($stav == 8)
return("Stornováno");
if ($stav == 9)
return("Zamítnuto");
if ($stav == 10)
return("Částečně objednáno");
if ($stav == 11)
return("Částečně doručeno");
if ($stav == 12)
return("Odstraněna zadavatelem");
}

function getMyAcceptList($old = FALSE){
if($old == true)
{
$vetsi = ">";
}
else
{
$vetsi = "=";
}

$uz = new Uzivatel;
$limit = $uz->getLimit($_SESSION[logid]);
$list = $uz->getStredisko();
$stre = array();
foreach ($list AS $stred)
{
$stre[] = $stred[0];
}
$stred = implode(",",$stre);
$sql = $this->prp("SELECT o.idobjednavky,u.uzivatel,DATE_FORMAT(o.kdy,'%e. %c. %Y') AS kdyl,internicislo,stav,s.cislo
AS cislo_s,s.popis AS popis_s,u.jmeno FROM objednavky AS o, Uzivatel AS u,strediska AS s
WHERE stav"."$vetsi"."1 AND o.idstrediska IN ($stred) AND o.iduzivatel = u.iduzivatel AND s.idstrediska = o.idstrediska
ORDER BY o.kdy DESC");

$sql2 = $this->prp("SELECT SUM(cena*mnozstvi) FROM polozky WHERE idobjednavky = ?");

$sql->execute();
$this->sqlchyba($sql);
$xdata = ($sql->fetchAll());

if ($old==false)
{
$sql3 = $this->prp("SELECT o.idobjednavky,u.uzivatel,DATE_FORMAT(o.kdy,'%e. %c. %Y') AS
kdyl,internicislo,stav,s.cislo AS cislo_s,s.popis AS popis_s,u.jmeno FROM objednavky AS o, uzivatel AS u,strediska AS s

```

```

WHERE stav=2 AND o.idstrediska IN ($stred) AND o.iduzivatel = u.iduzivatel AND s.idstrediska = o.idstrediska ORDER BY
o.kdy DESC");
    $sql3->execute();
    $ydata = ($sql3->fetchAll());

    $xdata = array_merge($ydata,$xdata);
}

    foreach ($xdata AS $data){
        $sql2 -> execute(array("$data[0]"));
        $this->sqlchyba($sql2);
        $cena = $sql2->fetchColumn();
        echo "<tr><td class=\"male60\">$data[7]</td><td class=\"male60\">$data[cislo_s] ($data[popis_s])</td><td
class=\"male60\">$data[2]</td><td class=\"male\"><a
href=\"index.php?menu=schvalit&amp;objednavka=ukaz&amp;cislo=$data[0]>$data[3]</a></td><td
class=\"male\">$data[2].number_format($cena,2,\".\",\" \").</td><td class=\"male60\">".$this->stav2text($data[4]).</td><td>";

if ($sold == false){
if ($limit>=$cena)
echo "<span class=\"lb\"><a
href=\"index.php?menu=schvalit&amp;schvalit=ok&amp;cislo=$data[0]&amp;stav=3\">Schválit</a></span>";
else
echo "<span class=\"lb\"><a
href=\"index.php?menu=schvalit&amp;schvalit=ok&amp;cislo=$data[0]&amp;stav=2\">Potvrdit</a></span>";
echo "<span class=\"lb\"><a
href=\"index.php?menu=schvalit&amp;schvalit=ok&amp;cislo=$data[0]&amp;stav=9\">Zamítnout</a></span>";

$mo = new Meta_objednavky;
$informace = $mo->getListByObjednavka($data[0]);
$info = "";
foreach($informace AS $inf)
{
    $info=$info.$inf[2]. " " . $inf[1]. " " . $inf[0]. "\n";
}
//echo $info;
echo "<span class=\"lb\"><a href=\"#\" onclick=\"alert('\\'. $info. '\')\">Detail</a></span>";
$obt = new Objednavkytxt;
$poznamka = $obt->get($data[0]);
if (count($poznamka)>1)
echo "<span class=\"lb\"><a href=\"#\" onclick=\"alert('".trim($poznamka[0])."')\">Poznámka</a></span>";

}
else
{
    $mo = new Meta_objednavky;
    $informace = $mo->getListByObjednavka($data[0]);
    $info = "";
    foreach($informace AS $inf)
    {
        $info=$info.$inf[2]. " " . $inf[1]. " " . $inf[0]. "\n";
    }
    //echo $info;
    echo "<span class=\"lb\"><a href=\"#\" onclick=\"alert('\\'. $info. '\')\">Detail</a></span>";

    $obt = new Objednavkytxt;
    $poznamka = $obt->get($data[0]);
    if (count($poznamka)>1)
    echo "<span class=\"lb\"><a href=\"#\" onclick=\"alert('".trim($poznamka[0])."')\">Poznámka</a></span>";
}

    echo "</td></tr>";

}
}
function checkChangeStatus(){
    $this->stav = $_GET[stav];
    $this->idobjednavky = $_GET[cislo];

    if (is_numeric($this->stav)==false OR $this->stav<1)
    throw new Exception ("Stav musí být číslo,9002);
    if (is_numeric($this->idobjednavky)==false OR $this->idobjednavky<1)
    throw new Exception ("Číslo objednávky musí být číslo,9003);

    $uz = new Uzivatel;
    $strediska = $uz->getStredisko();
    $data = array();
    foreach($strediska AS $info){
        $data[] = $info[0];
    }
    if ($_GET[hi]!=true){
        $sql = $this->prp("SELECT idstrediska FROM objednavky WHERE idobjednavky = ?");
        $sql->execute(array("$this->idobjednavky"));
        $this->sqlchyba($sql);
        $cislo = $sql->fetchColumn();

        if (in_array($cislo,$data)==false)
        throw new Exception ("Nemáte právo schvalovat tuto objednávku");

        $limit = $uz->getLimit();
        $sql2 = $this->prp("SELECT SUM(cena*mnozstvi) FROM polozky WHERE idobjednavky = ?");
        $sql2->execute(array("$this->idobjednavky"));
    }
}

```

```

        $this->sqlchyba($sql2);
        $cena = $sql2->fetchColumn();
        if ($cena > $limit AND $this->stav == 3)
        {
            throw new Exception ("Nemůžete schalovat takovou částku, byla provedan změna na potvrzení");
            $this->stav = 2;
        }
    }
}
function changeStatus($stav=NULL,$cislo=NULL) {

if (is_null($stav)==true)
{
    $this->stav = $_GET[stav];
}
else
    $this->stav = $stav;
if (is_null($cislo)==true)
{
    $this->idobjednavky = $_GET[cislo];
}
else
    $this->idobjednavky = $cislo;

    if (is_null($stav)==true AND is_null($cislo)==true){
        $po = new Polozky;
        $polozky = $po->getIdByObjednavka($this->idobjednavky);

        foreach ($polozky AS $info){
            // print_r($info);
            $po->changeStatus($info[0],"$this->stav",$info[1]);
        }

        $sql = $this->prp("UPDATE objednavky SET stav=? WHERE idobjednavky = ?");
        $sql->execute(array("$this->stav","$this->idobjednavky"));
        $this->sqlchyba($sql);
        if (is_null($stav)==true AND is_null($cislo)==true)
            $this->oznameni("Stav objednávky byl změněn na ".$this->stav2text($this->stav));
    }
}
function checkShow(){
    $this->idobjednavky = $_GET[cislo];
    $uz = new Uzivatel;
    $strediska = $uz->getStredisko();
    $data = array();
    foreach($strediska AS $info){
        $data[] = $info[0];
    }

    $sql = $this->prp("SELECT idstrediska FROM objednavky WHERE idobjednavky = ?");
    $sql->execute(array("$this->idobjednavky"));
    $this->sqlchyba($sql);
    $cislo = $sql->fetchColumn();

    if (in_array($cislo,$data)==false)
        throw new Exception ("Nemáte právo zobrazit tuto objednávku");
}
function getDetail($cislo){
    $sql = $this->prp("SELECT u.uzivatel,u.jmeno,o.idstrediska,o.kdy,o.stav,o.internicislo
        FROM objednavky AS o, uzivatel AS u
        WHERE idobjednavky = ?
        AND o.iduzivatel = u.iduzivatel
        ");
    $sql->execute(array("$cislo"));
    $this->sqlchyba($sql);
    return($sql->fetch());
}
function getMyOrderList($subjekt=NULL){
    if (is_null($subjekt)==false)
        $where = "WHERE cislo = 1";
}
function checkAll($stav=4,$sestav=null){
    // echo $stav;
    $sql = $this->prp("SELECT idobjednavky FROM objednavky WHERE stav < 7 OR stav > 9");
    $sql2 = $this->prp("SELECT stav FROM polozky WHERE idobjednavky = ? AND stav <> ?");
    $sql3 = $this->prp("SELECT stav FROM polozky WHERE idobjednavky = ?");
    $sql->execute();
    $this->sqlchyba($sql);
    $this->sqlchyba($sql2);
    $this->sqlchyba($sql3);
    foreach ($sql->fetchAll() AS $info){
        //echo $info[0];
        //echo $stav;
        $sql2->execute(array("$info[0]","$stav"));
        // $ok = true;
        $data = $sql2->fetchAll();
        //print_r($data);
        // echo count($data);
        if (count($data)==0)
            $this->changeStatus($stav,$info[0]);
        elseif(is_null($sestav) == false)
            {

```

```

$sql3->execute(array("$sinfo[0]"));
//print_r($sql3->fetchAll());
//echo "Nová";
$sold = "0";
$new = "0";
$polozky = $sql3->fetchAll();

//echo "<pre>";
//print_r($polozky);
//echo "</pre>";

//echo $xdt[0];
$ok = false;
foreach ($polozky AS $xdt)
{
    $step = false;
    if ($new<1)
    $old = $xdt[0];
    else
    $old = $new;

    $new = $xdt[0];

//echo $xdt[0];
    /*echo "$old"."-";
    echo "$new";*/

    if ($old != $new)
    $ok = true;
}
// echo "<br/>";
if ($ok == true)
{
    $this->changeStatus($stav,$sinfo[0]);
}

}

}

function getAcceptedList($old = FALSE){

    if($old == true)
    {
        $vetsi = ">";
    }
    else
    $vetsi = "=";

    $uz = new Uzivatel;
    $limit = $uz->getLimit($_SESSION[logid]);
    $list = $uz->getStredisko();
    $sql = $this->prp("SELECT o.idobjednavky,u.uzivatel,kdy,internicislo FROM objednavky AS o, uzivatel AS u
WHERE stav=2 AND o.iduzivatel = u.iduzivatel ORDER BY kdy DESC");
    $sql2 = $this->prp("SELECT SUM(cena*mnozstvi) FROM polozky WHERE idobjednavky = ?");

    $sql->execute(array("$sinfo[0]"));
    $this->sqlchyba($sql);
    $xdata = ($sql->fetchAll());

    foreach ($xdata AS $data){
        $sql2 -> execute(array("$data[0]"));
        $this->sqlchyba($sql2);
        $cena = $sql2->fetchColumn();
        echo "<tr><td>$data[1]</td><td>$data[2]</td><td><a
href='\"index.php?menu=mschvaleni&amp;objednavka=ukaz&amp;cislo=$data[0]\\\">$data[3]</a></td><td>".number_format($cena,2,"
, ").\"</td><td>";
        if ($old == FALSE){
            echo "<span class='\"lb\\\"><a
href='\"index.php?menu=mschvaleni&amp;schvalit=ok&amp;cislo=$data[0]&amp;stav=3&amp;hi=true\\\">Schválit</a></span>";
            echo "<span class='\"lb\\\"><a
href='\"index.php?menu=mschvaleni&amp;schvalit=ok&amp;cislo=$data[0]&amp;stav=9&amp;hi=true\\\">Zamítnout</a></span>";
        }
        echo "</td></tr>";
    }
}

function getUzivatelByStav($strediska,$stav=5){
    $st = new Strediska;
    $strediska = $st->getAllMy($_SESSION[logid]);
    // print_r($strediska);
    foreach ($strediska AS $data)
    {
        $stred = $data[0].".".$stred;
    }
    $stred = substr($stred,0,-1);

    $sql = $this->prp("SELECT DISTINCT iduzivatel FROM objednavky WHERE stav=? AND idstrediska IN ($stred)");
    $sql->execute(array($stav));
    $this->sqlchyba($sql);
    //print_r($sql->fetchAll());
    return($sql->fetchAll());
}

```

```

function getMy($id) {
    $sql = $this->prp("SELECT idobjednavky FROM objednavky WHERE iduzivatel=? AND stav=5");
    $sql->execute(array($id));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}
function getMyByZarizeni($uzivatele)
{
    $sql = $this->prp("SELECT idobjednavky FROM objednavky WHERE idstrediska IN ($uzivatele) AND stav=3");
    $sql->execute();
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}
}
?>

```

## Objednavkytxt.php

```

<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 * Description of objednavkytxt
 *
 * @author Mirek
 */
class objednavkytxt extends Objednavky {
    var $idobjednavkytxt;
    var $idobjednavky;
    var $poznamka;
    var $datum;

    function setvariables($idobjednavkytxt=null,$idobjednavky=null,$poznamka=null,$datum=null)
    {
        $this->idobjednavkytxt=$idobjednavkytxt;
        $this->idobjednavky = $idobjednavky;
        $poznamka = urlencode($poznamka);
        //echo $poznamka;
        $poznamka = str_replace("%0D%0A", "", $poznamka);
        //echo $poznamka;
        $poznamka = urldecode($poznamka);
        $this->poznamka = $poznamka;
        $this->datum = $datum;
    }
    function insert()
    {
        $sql = $this->prp("INSERT INTO objednavkytxt VALUES (NULL,?,?,?,NOW())");
        $sql->execute(array("$this->idobjednavky", "$this->poznamka"));
        $this->sqlchyba($sql);
    }
    function update($id,$poznamka)
    {
        $sql = $this->prp("UPDATE objednavkytxt SET poznamka=?,datum=NOW() WHERE idobjednavky=?");
        $sql->execute(array($poznamka,$id));
        $this->sqlchyba($sql);
    }
    function get($id)
    {
        $sql = $this->prp("SELECT poznamka FROM objednavkytxt WHERE idobjednavky=? LIMIT 1");
        $sql->execute(array($id));
        $this->sqlchyba($sql);
        return($sql->fetch());
    }

    function prepare($string)
    {
        $poznamka = urlencode($string);
        //echo $poznamka;
        $poznamka = str_replace("%0D%0A", "", $poznamka);
        //echo $poznamka;
        $poznamka = urldecode($poznamka);
        return($poznamka);
    }
}
?>

```

## Oddeleny.php

```

<?php
class Oddeleni extends Main {
    var $idoddeleni;
    var $nazev;
    var $popis;

    function getlist()
    {
        $sql = $this->prp("SELECT idoddeleni,nazev,popis FROM oddeleni ORDER BY nazev");
        $sql->execute();
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }
}

```

```
}  
}
```

```
?>
```

## Podskupina.php

```
<?php
```

```
class podskupina extends Main {  
    var $idpodskupina;  
    var $idhlavni_skupina;  
    var $nazev;  
  
    function addRow(){  
        $sql=$this->prp("INSERT INTO podskupina VALUES (NULL,?,?)");  
        $sql->execute(array("$this->nazev","$this->idhlavni_skupina"));  
        $this->sqlchyba($sql);  
    }  
    function checkAdd(){  
        $this->nazev = $_POST[nazev];  
        $this->idhlavni_skupina = $_POST[skupina];  
        if (strlen($this->nazev)<5)  
            throw new Exception ("Název skupiny musí být delší než 5 znaků",11001);  
        if (is_numeric($this->idhlavni_skupina)==false OR $this->idhlavni_skupina < 1)  
            throw new Exception ("Hlavní skupina nebyla vybrána",11002);  
    }  
    function getList(){  
        $sql=$this->prp("SELECT idpodskupina,nazev FROM podskupina");  
        $sql->execute();  
        $this->sqlchyba($sql);  
        return($sql->fetchAll());  
    }  
  
}
```

```
?>
```

## Polozky.php

```
<?php
```

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
/**  
 * Description of polozky  
 *  
 * @author Mirek  
 */  
class Polozky extends Main {  
    var $idpolozky;  
    var $idobjednavky;  
    var $idsklad;  
    var $nazev;  
    var $jednotky;  
    var $cena;  
    var $cislo;  
    var $popis;  
    var $mnozstvi;  
    var $stav;  
  
    function addRow($idobjednavky,$idsklad,$nazev,$jednotky,$cena,$cislo,$popis,$mnozstvi,$stav) {  
        $sql = $this->prp("INSERT INTO polozky VALUES (NULL,  
        ?,  
        ?,  
        ?,  
        ?,  
        ?,  
        ?,  
        ?,  
        ?)");  
        $sql->execute(array("$idobjednavky","$idsklad","$nazev","$jednotky","$cena","$cislo","$popis","$mnozstvi","$stav"));  
        // $sql->execute();  
        $this->sqlchyba($sql);  
    }  
  
    function getDetail($cislo,$stav=null){  
        //echo $cislo;  
        if (is_null($stav)==false)  
            $det = "AND p.stav=$stav";  
        $sql = $this->prp("SELECT p.idpolozky,p.cena,p.mnozstvi,s.nazev,s.jednotky,s.cislo,d.jmeno, (p.cena*p.mnozstvi) AS  
sum,p.popis,p.stav,p.idobjednavky  
FROM polozky AS p, dodavatel AS d, sklad AS s  
WHERE p.idsklad = s.idsklad AND  
s.dodavatel = d.iddodavatel AND  
p.idobjednavky = ? $det  
ORDER BY s.nazev  
");  
        $sql->execute(array("$cislo"));  
    }  
}
```

```

$this->sqlchyba($sql);
//print_r($sql->fetchAll());
return($sql->fetchAll());
}

function getmDetail($cislo,$stav=null){
    //echo $cislo;
    if (is_null($stav)==false)
        $det = "AND p.stav=$stav";
    $sql = $this->prp("SELECT p.idpolozky,p.cena,p.mnozstvi,s.nazev,s.jednotky,s.cislo,d.jmeno,(p.cena*p.mnozstvi) AS
sum,p.popis,p.stav,p.idobjednavky
FROM polozky AS p, dodavatel AS d, sklad AS s
WHERE p.idsklad = s.idsklad AND
s.dodavatel = d.iddodavatel AND
p.idpolozky = ? $det
ORDER BY s.nazev
");
    $sql->execute(array("$cislo"));
    $this->sqlchyba($sql);
    //print_r($sql->fetchAll());
    return($sql->fetchAll());
}

function getGroupDetail($cislo,$stav=null){
    if (is_null($stav)==false)
        $det = "AND p.stav=$stav";
    $sql = $this->prp("SELECT p.idpolozky,p.cena,SUM(p.mnozstvi) AS mnozstvi,s.nazev,s.jednotky,s.cislo,d.jmeno,
(p.cena*SUM(p.mnozstvi)) AS sum,p.popis,p.stav,s.idsklad,p.idobjednavky
FROM polozky AS p, dodavatel AS d, sklad AS s
WHERE p.idsklad = s.idsklad AND
s.dodavatel = d.iddodavatel AND
p.idobjednavky = ? $det
GROUP BY s.cislo
ORDER BY s.nazev
");
    $sql->execute(array("$cislo"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getByDodavatel($dodavatel,$stredisko,$stav=3){
    foreach ($stredisko AS $data){
        $str = $data[0].",".$str;
    }
    $str = substr("$str",0,-1);
    //echo $str;

    //echo $dodavatel;
    $sql = $this->prp("SELECT p.idsklad,p.idobjednavky,p.nazev,p.cena,p.jednotky,p.cislo,SUM(p.mnozstvi) AS
mnozstvi,o.internicislo,p.idpolozky,
st.cislo AS stredisko,u.jmeno AS uzivatel
FROM polozky AS p,sklad AS s, objednavky AS o,strediska AS st, uzivatel AS u
WHERE p.idsklad = s.idsklad AND
o.idobjednavky = p.idobjednavky AND
st.idstrediska = o.idstrediska AND
u.iduzivatel = o.iduzivatel AND
s.dodavatel = ? AND
o.idstrediska IN ($str) AND
p.stav=$stav
GROUP BY o.idstrediska,p.cislo
ORDER BY p.cislo DESC
");
    $sql->execute(array("$dodavatel"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getByCislo($cislo,$stav=3) {
    //echo $cislo;
    $sql = $this->prp("SELECT cena,SUM(mnozstvi) AS mnozstvi FROM polozky WHERE cislo=? AND stav=$stav GROUP BY cislo");
    $sql->execute(array("$cislo"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getByCisloZarizeni($cislo,$stredisko,$stav=3)
{
    foreach ($stredisko AS $data){
        $str = $data[0].",".$str;
    }
    $str = substr("$str",0,-1);

    // $sql = $this->prp("SELECT cena,SUM(mnozstvi) AS mnozstvi FROM polozky WHERE cislo=? AND stav=$stav AND idobjednavky IN
($cisla) GROUP BY cislo");

    $sql = $this->prp("SELECT p.cena,SUM(p.mnozstvi) AS mnozstvi,o.internicislo,p.idpolozky
FROM polozky AS p,sklad AS s, objednavky AS o
WHERE p.idsklad = s.idsklad AND
o.idobjednavky = p.idobjednavky AND

o.idstrediska IN ($str) AND
p.stav=$stav
ORDER BY p.cislo DESC
");
    $sql->execute(array("$cislo"));
}

```

```

$this->sqlchyba($sql);
return($sql->fetchAll());

}

function getByStav($dodavatel,$stredisko,$stav) {
    $sql = $this->prp("SELECT
    p.idsklad,p.idobjednavky,p.nazev,p.cena,p.jednotky,p.cislo,p.mnozstvi,o.internicislo,p.idpolozky
    FROM polozky AS p,sklad AS s, objednavky AS o
    WHERE p.idsklad = s.idsklad AND
    o.idobjednavky = p.idobjednavky AND
    s.dodavatel = ? AND
    o.idstrediska = ? AND

    p.stav=?
    ORDER BY p.cislo
    ");
    $sql->execute(array("$dodavatel","$stredisko","$stav"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function changeStatus($id,$stav,$data=null){
    // echo "Changing status";
    if (is_null($data)==false)
        $cislo = "AND cislo=?";

    $sql = $this->prp("UPDATE polozky SET stav=? WHERE idpolozky=? $cislo");

    if (is_null($data)==false)
        $sql->execute(array("$stav","$id","$data"));
    else
        $sql->execute(array("$stav","$id"));

    $this->sqlchyba($sql);
}

function getIdByObjednavka($objednavka){
    $sql=$this->prp("SELECT idpolozky,cislo FROM polozky WHERE idobjednavky=?");
    $sql->execute(array("$objednavka"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getIdObjednavky($info){

    $sql=$this->prp("SELECT idobjednavky FROM polozky WHERE idpolozky=?");
    $sql->execute(array("$info"));
    $this->sqlchyba($sql);
    return($sql->fetchColumn());
}

function getIdSkladu($id){
    $sql=$this->prp("SELECT idpolozky FROM polozky WHERE idsklad=?");
    $sql->execute(array("$id"));
    $this->sqlchyba($sql);
    return($sql->fetchColumn());
}

function getIdSklad($polozka)
{
    $sql=$this->prp("SELECT idsklad FROM polozky WHERE idpolozky=?");
    $sql->execute(array("$polozka"));
    $this->sqlchyba($sql);
    return($sql->fetchColumn());
}

function setPrice($polozka,$sklad,$cena){
    //echo $cena;
    $sql=$this->prp("UPDATE polozky SET cena=? WHERE idpolozky=?");
    $sql->execute(array("$cena","$polozka"));
    $this->sqlchyba($sql);
    $sql=$this->prp("UPDATE sklad SET cena=? WHERE idsklad=?");
    $sql->execute(array("$cena","$sklad"));
    $this->sqlchyba($sql);
}

function top10($id){
    $sql = $this->prp("
    SELECT p.idsklad, p.nazev, count( p.idsklad ) AS pocet, MIN( p.mnozstvi ) AS mnozstvi, MAX(p.cena) AS cena,p.popis
    FROM polozky AS p, objednavky AS o
    WHERE o.iduzivatel =?
    AND o.idobjednavky = p.idobjednavky
    GROUP BY idsklad
    ORDER BY count( idsklad ) DESC
    LIMIT 25
    ");
    $sql->execute(array("$id"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function deleteBySklad($id,$sklad){

    //echo $id;
    $sql = $this->prp("
    SELECT idpolozky
    FROM polozky

```



```

WHERE idobjednavky = ?
AND idsklad=?"
);

$sql2 = $this->prp("DELETE FROM polozky WHERE idpolozky = ?");
$sql->execute(array("$id", "$sklad"));
$this->sqlchyba($sql);
//print_r($sql->fetch());
foreach ($sql->fetchAll() AS $info)
{
    // echo "Mažu $info[0]";
    $sql2->execute(array("$info[0]"));
    $this->sqlchyba($sql2);
}

$sql = $this->prp("SELECT count(idpolozky) FROM polozky WHERE idobjednavky=?");
$sql->execute(array("$id"));
$this->sqlchyba($sql);
if ($sql->fetchColumn() < 1)
{
    $ob = new Objednavky;
    $ob->changeStatus(12, $id);
}

}
function getAllMy($stav, $objednavky)
{
    $sql = $this->prp("SELECT idpolozky FROM polozky WHERE stav=? AND idobjednavky=?");
    $sql->execute(array("$stav", "$objednavky"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

}
function editBySklad($id, $sklad)
{
    // echo $id;
    // echo $sklad;
    $sql = $this->prp("
SELECT idpolozky, SUM(mnozstvi), idobjednavky, idsklad, nabez, jednotky, cena, cislo, popis, mnozstvi, stav
FROM polozky
WHERE idobjednavky = ?
AND idsklad=?
GROUP BY idsklad"
);
    $sql->execute(array("$id", "$sklad"));
    $this->sqlchyba($sql);

    foreach ($sql->fetchAll() AS $info)
    {
        $idpolozky = $info[idpolozky];
        $mnozstvi = $_POST[mnozstvi];
        $idobjednavky = $info[idobjednavky];
        $idsklad = $info[idsklad];
        $nabez = $info[nabez];
        $jednotky = $info[jednotky];
        $cena = $info[cena];
        $cislo = $info[cislo];
        $popis = $info[popis];
        $stav = $info[stav];

        $dsq = $this->prp("DELETE FROM polozky WHERE idobjednavky=? AND idsklad=?");
        $dsq->execute(array("$id", "$sklad"));
        $this->sqlchyba($dsq);

        $isq = $this->prp("INSERT INTO polozky VALUES (NULL, ?, ?, ?, ?, ?, ?, ?)");
        $isq->execute(array("$idobjednavky", "$idsklad", "$nabez", "$jednotky", "$cena", "$cislo", "$popis", "$mnozstvi", "$stav"));
        $this->sqlchyba($isq);

        echo "Množství změněno";
    }
}
}
?>

```

## Prava.php

```
<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of sklad
 *
 * @author Mirek
 */
class Sklad extends Main {
    var $idsklad;
    var $vlozil;
    var $dodavatel;
    var $idskupiny;
    var $nazev;
    var $jednotky;
    var $cena;
    var $cislo;
    var $popis;
    var $vlozeno;
    var $zmeneno;
    var $viditelne;
    var $temp;
    var $lekarsky;
    var $dph;
    var $sazbadph;

    function addRow()
    {
        $sql=$this->prp("INSERT INTO sklad VALUES (NULL,?, ?, ?, ?, ?, ?, ?, NOW(), NOW(), ?, ?, ?, ?)");
        $sql->execute(array("$this->vlozil", "$this->dodavatel", "$this->idskupiny", "$this->nazev", "$this->jednotky",
"$this->cena", "$this->cislo", "$this->popis", "$this->viditelne", "$this->temp", "$this->lekarsky", "$this->dph", "$this-
>sazbadph"));
        $this->sqlchyba($sql);
    }

    function check()
    {
        $this->vlozil = $_SESSION[logid];
        $this->dodavatel = "$_POST[dodavatel]";
        $this->idskupiny = "$_POST[skupina]";
        $this->nazev = "$_POST[nazev]";
        $this->jednotky = "$_POST[jednotky]";
        $this->cena = "$_POST[cena]";
        $this->cislo = "$_POST[cislo]";
        $this->popis = "$_POST[popis]";
        $this->viditelne = "$_POST[viditelne]";
        $this->lekarsky = "$_POST[lekarsky]";
        $this->dph = "$_POST[dph]";
        $this->sazbadph = "$_POST[sazbadph]";

        if (is_numeric($this->dodavatel)==false OR $this->dodavatel<1)
            throw new Exception("Špatně vybraný dodavatel", 3001);
        if (is_numeric($this->idskupiny)==false OR $this->idskupiny < 1)
            throw new Exception("Neexistující dodavatel", 3002);
        if (strlen($this->nazev)<2)
            throw new Exception("Nebyl zadán název", 3003);
        if (strlen($this->jednotky)<1)
            throw new Exception("Nebyly zadány jednotky", 3004);
        $sql = $this->prp("SELECT idsklad FROM sklad WHERE dodavatel=? AND idskupiny=? AND cislo=?");
        $sql->execute(array("$this->dodavatel", "$this->idskupiny", "$this->cislo"));
        $data = $sql->fetchColumn();
        if ($data > 0)
            throw new Exception("Zadaný výrobek <strong>$this->nazev</strong> již existuje, <br/>změňte u něj jednu z následujících
informací<br/>
katlogové číslo, skupinu nebo dodavatele.", 3105);
        if (is_numeric($this->cena) == false OR $this->cena<1)
            throw new Exception("Cena musí být číslo a větší než 0", 3006);
        if (strlen($this->cislo)<2)
            throw new Exception("Musí být zadáno katalogové číslo", 3007);
        if (strlen($this->popis)<1)
            $this->popis = "Popis nebyl vyplněn";
        if (!$this->lekarsky)
            $this->lekarsky = 0;
        if (is_numeric($this->lekarsky)==false OR $this->lekarsky>1 OR $this->lekarsky<0)
            throw new Exception("Lékažský stav musí být 1 nebo 0, jedná se o neutarizovaný vstup", 3208);
        if (!$this->dph)
            $this->dph = 0;
        if (is_numeric($this->dph)==false OR $this->dph>1 OR $this->dph<0)
            throw new Exception("DPH stav musí být 1 nebo 0, jedná se o neutarizovaný vstup", 3209);
        if (!$this->temp)
            $this->temp = 0;
        if (is_numeric($this->temp)==false OR $this->temp>1 OR $this->temp<0)
            throw new Exception("TEMP stav musí být 1 nebo 0, jedná se o neutarizovaný vstup", 3210);
        if (is_numeric($this->sazbadph)==false OR $this->sazbadph<0 OR $this->sazbadph>100)
            throw new Exception("Sazba DPH neodpovídá limitům");
    }

    function getList()
    {
        $sql = $this->prp("SELECT * FROM sklad ORDER BY nazev");
    }
}
```

```

    $sql->execute();
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getListBySkupina($skupina) {
    if ($_GET[limit]<1)
    $i= 0;
    else
    $i = $_GET[limit];
    $limit = "LIMIT $i,30";
    $sql = $this->prp("SELECT idsklad,nazev,jednotky,cena,popis,cislo FROM sklad WHERE idskupiny=? AND viditelne = 1 ORDER BY
    nazev $limit");
    $sql->execute(array("$skupina"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getListByName($jmeno=NULL) {
    if ($_GET[limit]<1)
    $i= 0;
    else
    $i = $_GET[limit];

    if ($jmeno)
    {
        $filtr = "AND nazev LIKE '%$jmeno%'";
    }
    else
    {
        $limit = "LIMIT $i,30";
    }

    $sql=$this->prp("SELECT sk.nazev,s.nazev,d.jmeno,sk.cena,sk.cislo FROM sklad AS sk,skupiny AS s,dodavatel AS d WHERE
    sk.idskupiny = s.idskupiny AND sk.dodavatel = d.iddodavatel AND viditelne=1 $filtr $limit");
    $sql->execute(array());
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getDataCount($skupina=NULL) {
    if ($skupina)
    {
        $where = "WHERE idskupiny=$skupina AND viditelne = 1";
    }
    $sql = $this->prp("SELECT count(idsklad) FROM sklad $where");
    $sql->execute();
    $this->sqlchyba($sql);
    return($sql->fetchColumn());
}

function existing($id){
    $sql = $this->prp("SELECT nazev FROM sklad WHERE idsklad=?");
    $sql->execute(array("$id"));
    $this->sqlchyba($sql);
    $data = $sql->fetchColumn();
    if (strlen($data)>1)
    return(true);
    else
    return(false);
}

function getId($id){
    $sql = $this->prp("SELECT nazev,jednotky,cena,cislo,popis FROM sklad WHERE idsklad=?");
    $sql->execute(array("$id"));
    $this->sqlchyba($sql);
    return($sql->fetch());
}

function getListByHledat($co)
{
    $sql=$this->prp("SELECT idsklad,nazev,jednotky,cena,popis,cislo FROM sklad WHERE viditelne = 1 AND (nazev LIKE '%$co%' OR
    cislo LIKE '%$co%') ORDER BY nazev LIMIT 50") ;
    $sql->execute(array());
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function getListByDodavatel($dodavatel)
{
    $sql=$this->prp("SELECT sk.nazev,s.nazev,d.jmeno,sk.cena,sk.cislo,sk.idsklad,s.idskupiny,d.iddodavatel AS dodavatel
    FROM sklad AS sk,skupiny AS s,dodavatel AS d WHERE sk.idskupiny = s.idskupiny AND sk.dodavatel = d.iddodavatel AND
    d.iddodavatel=?");

    $sql->execute(array("$dodavatel"));
    $this->sqlchyba($sql);
    return($sql->fetchAll());
}

function update()
{
    //echo $this->dodavatel;

    if ($this->cislo == 0 AND $this->cena == 0)
    $zneviditelnit = ",viditelne=0";
    $sql=$this->prp("UPDATE sklad SET idskupiny=?,nazev=?,cena=?,cislo=?,dodavatel=?$zneviditelnit WHERE idsklad=?");
    $sql->execute(array("$this->idskupiny","$this->nazev","$this->cena","$this->cislo","$this->dodavatel","$this-
    >idsklad"));
}

```

```
        $this->sqlchyba($sql);
    }
}
```

?>

## Skupiny.php

```
<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of skupiny
 *
 * @author Mirek
 */
class Skupiny extends Main {
    var $idskupiny;
    var $nazev;
    var $popis;

    function addRow()
    {
        $sql = $this->prp("INSERT INTO skupiny VALUES (NULL,?,?)");
        $sql->execute(array("$this->nazev","$this->popis"));
        $this->sqlchyba($sql);
    }

    function getList(){
        $sql = $this->prp("SELECT idskupiny,nazev,popis FROM skupiny");
        $sql->execute();
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

    function getAllByNad($nadskupina)
    {
        $sql = $this->prp("SELECT idskupiny,nazev,popis FROM skupiny WHERE idnaskupiny = ? ORDER BY nazev");
        $sql->execute(array($nadskupina));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

    function check(){
        $this->nazev = $_POST[nazev];
        $this->popis = $_POST[popis];
        if (strlen($this->nazev)<3)
            throw new Exception("Nebyl vyplněn název skupiny",2001);
        $sql = $this->prp("SELECT idskupiny FROM skupiny WHERE nazev=?");
        $sql->execute(array("$this->nazev"));
        $data = $sql->fetchColumn();
        if ($data > 0)
            throw new Exception("Název skupiny <strong>$this->nazev</strong> již existuje",2002);
    }

    function deleteRow($id){
        $sql=$this->prp("DELETE FROM skupiny WHERE id=?");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
    }
}
?>
```

## Smlouvy.php

```
<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of smlouvy
 *
 * @author Mirek
 */
class smlouvy extends Main {
    var $idsmlouvy;
    var $idpodskupiny;
    var $iduzivatel;
    var $idspolecnosti;
    var $nazev;
    var $popis;
    var $priloha;
    var $verze;

    function addRow(){
        $sql = $this->prp("INSERT INTO smlouvy VALUES (NULL,?,?,?,?,?,NOW())");
        $sql->execute(array("$this->idpodskupiny","$this->iduzivatel","$this->idspolecnosti",
"$this->nazev","$this->popis","$this->priloha"));
        $this->sqlchyba($sql);
    }
}
```

```

}
function check(){
    $this->idpodskupiny = $_POST[skupina];
    $this->iduzivatel = $_SESSION[logid];
    $this->idspolecnosti = $_POST[spolecnost];
    $this->nazev = $_POST[nazev];
    $this->popis = $_POST[popis];
    $this->priloha = $_FILES[priloha][name];

    $priloha = substr($this->priloha,-3,3);

    if ($priloha!="pdf")
        throw new Exception ("Přiloha může být <strong>pouze</strong> soubor PDF",13005);
    else
    {

        $target_path = "smlouvy/";

        $name=uniqid()."pdf";
        $this->priloha = $name;
        $target_path = $target_path . $name;

        if (file_exists($target_path))
        {
            $name=uniqid()."pdf";
            $this->priloha = $name;
            $target_path = $target_path . $name;
        }

        if(move_uploaded_file($_FILES['priloha']['tmp_name'], $target_path)) {
            echo "Soubor smlouvy byl nahrán";
        } else{
            throw new Exception ("There was an error uploading the file, please try again!",13004);
        }

        if (strlen($this->priloha)<1)
            throw new Exception ("Nebyl vložen text smlouvy",13001);
        if (strlen($this->nazev)<5)
            throw new Exception ("Vyplňte název smlouvy",13002);
        if (strlen($this->popis)<5)
            throw new Exception ("Vyplňte popis smlouvy",13003);

    }

    function getList(){
        $sql = $this->prp("SELECT s.idsmouvy AS
idsmlouvy,s.idpodskupiny,s.iduzivatel,s.idspolecnosti,s.nazev,s.popis,s.priloha,s.verze,sp.nazev AS
spolecnost_nazev,sk.nazev AS skupina_nazev FROM smlouvy AS s, spolecnosti AS sp, podskupina AS sk WHERE s.idpodskupiny =
sk.idpodskupina AND s.idspolecnosti=sp.idspolecnosti ORDER BY nazev");
        $sql->execute();
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }
}
?>

```

## Spolecnosti.php

```

<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 * Description of spolecnosti
 *
 * @author Mirek
 */
class spolecnosti extends Main {
    var $idspolecnosti;
    var $iduzivatel;
    var $nazev;
    var $popis;
    var $vlozena;

    function addRow()
    {
        $sql=$this->prp("INSERT INTO spolecnosti VALUES (
NULL,
?,
?,
?,
NOW()
)");
        $sql->execute(array("$this->iduzivatel", "$this->nazev", "$this->popis"));
        $this->sqlchyba($sql);
    }
    function checkAdd(){
        $this->iduzivatel = $_SESSION[logid];
        $this->nazev = $_POST[nazev];
        $this->popis = $_POST[popis];

        if (strlen($this->nazev)<3)

```

```

        throw new Exception ("Zadejte název",13001);
        if (strlen($this->popis)<5)
            throw new Exception ("Napište popis",13002);
    }
    function getList(){
        $sql=$this->prp("SELECT idspolecnosti,nazev FROM spolecnosti ORDER BY nazev");
        $sql->execute();
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }
}
?>

```

## Strediska.php

```

<?php
class Strediska extends Main {
    var $idstrediska;
    var $idzarizeni;
    var $cislo;
    var $popis;
    var $plimit;
    var $nemocnice;

    function getlistStred($id)
    {
        $sql = $this->prp("SELECT idstrediska,idzarizeni,cislo,popis,plimit,nemocnice FROM strediska WHERE idstrediska = ? ORDER
        BY cislo");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

    function getList(){
        $sql = $this->prp("SELECT idstrediska,idzarizeni,cislo,popis,plimit,nemocnice FROM strediska ORDER BY cislo");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

    function getlistArray($idstrediska=array())
    {
        $idst = array();
        foreach ($idstrediska AS $stred)
        {
            $idst[]=$stred[0];
        }
        $idstrediska = implode(",",$idst);

        $sql = $this->prp("SELECT idstrediska,idzarizeni,cislo,popis,plimit,nemocnice FROM strediska WHERE idstrediska IN
        ($idstrediska) ORDER BY cislo");
        $sql->execute();
        $this->sqlchyba($sql);
        //print_r($sql->fetchAll());
        return($sql->fetchAll());
    }

    function getRow($id){
        $sql = $this->prp("SELECT idstrediska,idzarizeni,cislo,popis,plimit,nemocnice FROM strediska WHERE idstrediska = ?");
        $sql->execute(array($id));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

    function getMy($id){

        $sql = $this->prp("SELECT idstrediska
        FROM objednavky
        WHERE iduzivatel = ?
        GROUP BY idstrediska
        ORDER BY count(idobjednavky) DESC LIMIT 1");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
        $data = $sql->fetchColumn();
        if (!$data){
            $data = $this->getAllMy($id);
            $dat = $data[0][0];
            return($dat);
        }
        return($data);
    }

    function getAllMy($id){
        $sql=$this->prp("SELECT strediska_idstrediska
        FROM strediska_ma_uzivatel
        WHERE uzivatel_iduzivatel = ?");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
        $data = $sql->fetchAll();
        return($data);
    }

    function getAllMyBySkupiny($id,$skupina){
        //echo $id;
    }
}

```

```

        //echo $skupina;
        $sql=$this->prp("SELECT s.strediska_idstrediska
FROM strediska_ma_uzivatel AS s, strediska AS sr
WHERE s.uzivatel_iduzivatel = ? AND s.strediska_idstrediska = sr.idstrediska AND sr.nemocnice=?
");
        $sql->execute(array("$id","$skupina"));
        $this->sqlchyba($sql);
        $data = $sql->fetchAll();
        return($data);
    }

    function getAllMyGroup($id){
        $sql=$this->prp("SELECT sr.nemocnice
FROM strediska_ma_uzivatel AS s, strediska AS sr
WHERE s.uzivatel_iduzivatel = ? AND s.strediska_idstrediska = sr.idstrediska
GROUP BY sr.nemocnice");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }

    function skupina($cislo){
        if ($cislo == 0)
            return("Dopravní zdravotnictví");
        elseif ($cislo == 1)
            return("Nemocnice THP");
        elseif ($cislo == 2)
            return("Lékař");
        elseif ($cislo == 3)
            return("Lékárna");
    }

    function getDetail($cislo)
    {
        $sql = $this->prp("SELECT cislo,popis FROM strediska WHERE idstrediska=?");
        $sql->execute(array($cislo));
        $this->sqlchyba($sql);
        return($sql->fetch());
    }
}

```

?>

## String.php

```

<?php

class String extends Main {

    var $title = "";
    var $keywords = "";
    var $description = "";
    var $authors = "";

    function xmlheader()
    {

        $this->title = "Informační systém - Dopravní zdravotnictví a.s.";

        echo '<?xml version="1.0" encoding="utf-8"?>';
        echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">';
        <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs">
        <head>
        <title>'.$this->title.'</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
        <meta http-equiv="content-language" content="cs"/>
        <meta name="distribution" content="global"/>
        <meta name="Keywords" content="'.$this->keywords.'"/>
        <meta name="Description" content="'.$this->description.'"/>
        <meta name="authors" content="'.$this->authors.'"/>
        <meta name="robots" content="index, follow"/>
        <meta name="revisit-after" content="1 days"/>
        <meta name="rating" content="general"/>
        <meta name="generator" content="raven system, miracle php object base"/>

        <link href="css/object.css" rel="stylesheet" type="text/css" />

        ';
        echo '</head>';
        echo '<body>';

    }

    function xmlfooter()
    {
        echo '</body></html>';
    }

    function text($data){
        echo "Hurá";
    }

    function statsF(){
        echo "<h2>Základní informace</h2>";
        $mu = new Meta_uzivatel;
        $mu->iduzivatel = $_SESSION[logid];
        $last = $mu->getLastInfo();
        $last = $last[1];
        //print_r($last);
    }
}

```

```

echo "<div>Naposled přihlášen <strong>$last[datum]</strong> z PC </strong>$last[ip]</strong></div>";
$pr = new Prava();
$prava = $pr->getMy($_SESSION[logid]);
foreach ($prava AS $data)
{
    if ($data[0]==3)
        $this->naschvaleni();
}

}

function naschvaleni()
{
    /*
    $ob = new Objednavky;
    $old = false;
    echo "<h3>Objednávky na schválení</h3>";
    echo "<div>Počet objednávek na schválení</div>";
    //$list = $ob->getMyAcceptList($old);
    //print_r($list);
    */
}

?>

```

## Tempobjednavky.php

```
<?php
```

```

class tempObjednavky extends Main{
    var $idtemp_objednavky;
    var $internicislo;
    var $iduzivatel;
    var $idsklad;
    var $mnozstvi;

    function newRow(){
        $sql = $this->prp("INSERT INTO temp_objednavky VALUES (NULL,?,?,?,?)");
        $sql->execute(array("$this->internicislo","$this->iduzivatel","$this->idsklad","$this->mnozstvi"));
        $this->sqlchyba($sql);
    }

    function Exist($cislo){
        $sql = $this->prp("SELECT idtemp_objednavky FROM temp_objednavky WHERE internicislo=?");
        $sql -> execute(array("$cislo"));
        $this->sqlchyba($sql);
        return($sql->fetchColumn());
    }

    function getLastIc(){
        $sql = $this->prp("SELECT internicislo FROM temp_objednavky ORDER BY idtemp_objednavky DESC LIMIT 1");
        $sql->execute();
        $this->sqlchyba($sql);
        return($sql->fetchColumn());
    }

    function check(){
        $this->internicislo = $_SESSION[cislo];
        $this->iduzivatel = $_SESSION[logid];
        $this->idsklad = $_POST[sklad];
        $this->mnozstvi = $_POST[mnozstvi];

        if (is_numeric($this->mnozstvi)==false OR $this->mnozstvi<1)
            throw new Exception("Nebylo zadáno množství",5001);
        $sk = new Sklad;
        if ($sk->existing("$this->idsklad")==false)
            throw new Exception("Zboží neexistuje ve skladu. Chcete vytvořit návh?",5002);

        $exist = $this->exist($this->internicislo);
        if ($exist > 0)
        {
            $sql = $this->prp("SELECT iduzivatel FROM temp_objednavky WHERE internicislo=? LIMIT 1");
            $sql->execute(array("$this->internicislo"));
            $data = $sql->fetchColumn();
            if ($data != $_SESSION[logid])
            {
                $ob = new Objednavky;
                $obj = new Objednavky;
                $cislo = $obj->genInternicislo();
                $_SESSION[cislo]=$cislo;
                $this->internicislo = $cislo;
            }
        }
    }

    function getYourList($id,$internicislo){
        $sql = $this->prp("SELECT t.idtemp_objednavky,t.idsklad,SUM(t.mnozstvi) AS
mnozstvi,t.internicislo,s.nazev,s.jednotky,s.cena,s.cislo,s.popis FROM temp_objednavky AS t, sklad AS s WHERE
s.idsklad=t.idsklad AND iduzivatel=? AND internicislo=? GROUP BY t.idsklad");
        $sql->execute(array("$id","$internicislo"));
        $this->sqlchyba($sql);
        return($sql->fetchAll());
    }
}

```





```

{
    $jmeno = sha1("$this->uzivatel"."$this->jmeno");
    return($jmeno);
}

function addPrava() {
    $data = $this->prp("INSERT INTO uzivatel_ma_prava VALUES (?,?)");
    $prava = array();
    $prava = $_POST[prava];
    foreach ($prava AS $xdata)
    {
        $data->execute(array("$this->iduzivatel", "$xdata"));
    }
}

function addStrediska() {
    $data = $this->prp("INSERT INTO strediska_ma_uzivatel VALUES (?,?)");
    $prava = array();
    $prava = $_POST[stredisko];
    foreach ($prava AS $xdata)
    {
        $data->execute(array("$xdata", "$this->iduzivatel"));
    }
}

function addOddeleni() {
    $data = $this->prp("INSERT INTO uzivatel_ma_oddeleni VALUES (?,?)");
    $prava = array();
    $prava = $_POST[oddeleni];
    foreach ($prava AS $xdata)
    {
        $data->execute(array("$this->iduzivatel", "$xdata"));
    }
}

function checkuser()
{
    $this->uzivatel = "$_POST[uzivatel]";
    $this->heslo = sha1("$_POST[heslo]");
    if (strlen($this->uzivatel)<2)
    throw new Exception("Nebylo zadáno správně jméno",7);
    if (strlen($_POST[heslo]<2)
    throw new Exception("Nebylo zadáno správně heslo",8);
    $jmeno = $this->prp("SELECT iduzivatel FROM uzivatel WHERE uzivatel=? LIMIT 1");
    $jmeno->execute(array("$this->uzivatel"));
    $this->iduzivatel = $jmeno->fetchColumn();
    $this->sqlchyba($jmeno);
    if ($this->iduzivatel < 1)
    throw new Exception("Zadaný uživatel neexistuje",9);
    $jmeno = $this->prp("SELECT iduzivatel FROM uzivatel WHERE heslo=? AND uzivatel=? LIMIT 1");
    $jmeno->execute(array("$this->heslo", "$this->uzivatel"));
    $data = $jmeno->fetchColumn();
    $this->sqlchyba($jmeno);
    if ($data < 1){
        $meta = new meta_uzivatel;
        $meta->iduzivatel = $this->iduzivatel;
        $meta->akce = "Přihlašování";
        $meta->stav = "10";
        $meta->chyba = "1";
        $meta->popis = "Bylo zadáno špatné heslo";
        $meta->addrow();

        throw new Exception("Heslo nebylo vyplněno správně",10);
    }
}

function login(){
    $meta = new meta_uzivatel;
    $meta->iduzivatel = $this->iduzivatel;
    $meta->akce = "Přihlašování";
    $meta->stav = "0";
    $meta->chyba = "0";
    $meta->popis = "Uživatel byl přihlášen";
    $meta->addrow();

    $_SESSION[logid] = $this->iduzivatel;
    $_SESSION[hash] = sha1(uniqid());

    $temp = new tempObjednavky;
    $cislo = $temp->getOldTemp("$_SESSION[logid]");
    if ($cislo){
        $_SESSION[cislo]=$cislo;
        $warning = 1;
    }
    $st = new Strediska;
    $_SESSION[stredisko] = $st->getMy($this->iduzivatel);
    header("Location: index.php?warning=$warning");
}

function getprava(){
    $sql = $this->prp("SELECT prava_idprava FROM uzivatel_ma_prava WHERE uzivatel_iduzivatel = ?");
    $sql->execute(array("$_SESSION[logid]"));
    $prava = $sql->fetchAll();
    foreach($prava AS $pravo){

```

```

$this->prava[] = $pravo[0];
}
return($this->prava);
}

function testprava(){
$prava = array();
$prava = $this->getprava();

$odkazy = $this->odkazy();
$menu = $this->menu();
$odkazu = array();

foreach($odkazy AS $index=>$string)
{
    $rovnice = strpos($string,"=");
    $string = substr("$string",$rovnice+1,-1);
    $rovnice = strpos($string,"=");
    $string = substr("$string",$rovnice+1,-1);
    $rovnice = strpos("$string","\");
    $string = substr("$string",0,$rovnice);
    $odkaz["$index"]=$string;
}

$odkaz = array_flip($odkaz);
//print_r($odkaz);

$getmenu = $_GET[menu];
$testuj = $odkaz[$getmenu];

foreach ($menu AS $mdata){
if(in_array($testuj,$mdata))
{
    $pouzitelny = true;
    break(0);
}
else
    $pouzitelny = false;
}

echo $pouzitelny;
if ($pouzitelny == false)
throw new Exception("Přístup do této funkce není povolen",11);
}

function odkazy(){
$odkazy = array();
$odkazy[1] = "<a href='\"index.php?menu=pridatuzivatele'\">Přidat uživatele</a>";
$odkazy[2] = "<a href='\"index.php?menu=seznamuzivatelu'\">Seznam uživatelů</a>";
$odkazy[3] = "<a href='\"index.php?menu=nastaveniprav'\">Nastavení práv</a>";
$odkazy[4] = "<a href='\"index.php?menu=vytvoritobjednavku'\">Vytvořit objednávku</a>";
$odkazy[5] = "<a href='\"index.php?menu=sklad'\">Sklad</a>";
$odkazy[6] = "<a href='\"index.php?menu=seznamobjednavek'\">Seznam objednávek</a>";
$odkazy[7] = "<a href='\"index.php?menu=schvalit'\">Schvalování objednávek</a>";
$odkazy[8] = "<a href='\"index.php?menu=objednat'\">Aktivace objednávky</a>";
$odkazy[9] = "<a href='\"index.php?menu=mschvaleni'\">Manager Schvalování</a>";
$odkazy[10] = "<a href='\"index.php?menu=smlouvy'\">Spáva smluv</a>";
$odkazy[11] = "<a href='\"index.php?menu=seznamy'\">Smlouvy DZ</a>";
return($odkazy);
}

function menu(){
$menu = array();
$menu[2] = array();
$menu[3] = array(7);
$menu[4] = array(9);
$menu[5] = array(8);
$menu[6] = array();
$menu[7] = array(5);
$menu[8] = array();
$menu[9] = array();
$menu[10] = array(1,2);
$menu[11] = array(3);
$menu[1] = array(4,6);
$menu[12] = array(10);
$menu[13] = array(11);
return($menu);
}

function zkontrolovat(){
$this->uzivatel = $_POST[uzivatel];
$this->heslo = $_POST[heslo];
$heslo2 = $_POST[heslo2];
$this->email = $_POST[email];
$this->telefon = $_POST[telefon];
$this->limit = $_POST[limit];
$this->osobnicislo = $_POST[osobni];
$this->jmeno = $_POST[jmeno];

$stredisko = array();

```

```

$stredisko = $_POST[stredisko];
$oddeleni = array();
$oddeleni = $_POST[oddeleni];
$prava = array();
$prava = $_POST[prava];

if ($this->heslo != $heslo2)
throw new Exception("Hesla se neshodují",12);
if (strlen($this->heslo)<3)
throw new Exception("Heslo je příliš krátké, min. 3 znaky",13);
if (strlen($this->uzivatel)<3)
throw new Exception("Uživatelské jméno je příliš krátké, min 3 znaky",14);
if (strlen($this->email)<3 AND strlen($this->telefon)<3)
throw new Exception("Musí být zadán email nebo telefon",15);
if (strlen($this->osobnicislo)<1)
throw new Exception("Špatně zadané osobní číslo",16);

$jmeno = $this->prp("SELECT iduzivatel FROM uzivatel WHERE uzivatel=? LIMIT 1");
$jmeno->execute(array("$this->uzivatel"));
$this->iduzivatel = $jmeno->fetchColumn();
$this->sqlchyba($jmeno);
if ($this->iduzivatel > 0)
throw new Exception("Zadaný uživatel již existuje",17);

if (count($stredisko)<1)
throw new Exception("Nebylo vybráno žádné středisko",18);
if (count($oddeleni)<1)
throw new Exception("Nebylo vybráno žádné oddělení",19);
if (count($prava)<1)
throw new Exception("Uživatel nemá nastavena práva",20);

$this->heslo = sha1($_POST[heslo]);
}
function uzivatelResult(){
echo "<div class=\"submenu\">";
echo "<div class=\"text\">Nový uživatel s uživatelským jménem $this->uzivatel byl vytvořen</div>";
echo "</div>";

echo "<div class=\"karticka\">";
echo "<div class=\"jmeno\">Vytvořené jméno pro: <strong><em>$this->jmeno</em></strong></div>";
echo "<div class=\"uzivatel\">Uživatelské jméno <strong>$this->uzivatel</strong></div>";
echo "<div class=\"heslo\">Uživatelské heslo <strong>$_POST[heslo]</strong></div>";
echo "</div>";
}

function getList(){
$sql=$this->prp("SELECT iduzivatel,uzivatel,jmeno,email,telefon,plimit,osobnicislo FROM uzivatel");
$sql->execute(array());
$this->sqlchyba($sql);
return($sql->fetchAll());
}

function deleteUser($id){
$sql = $this->prp("DELETE FROM uzivatel_ma_oddeleni WHERE uzivatel_iduzivatel = ?");
$sql->execute(array("$id"));
$this->sqlchyba($sql);
$sql = $this->prp("DELETE FROM uzivatel_ma_prava WHERE uzivatel_iduzivatel = ?");
$sql->execute(array("$id"));
$this->sqlchyba($sql);
$sql = $this->prp("DELETE FROM strediska_ma_uzivatel WHERE uzivatel_iduzivatel = ?");
$sql->execute(array("$id"));
$this->sqlchyba($sql);
$this->deleterow("$id");

$meta = new meta_uzivatel;
$meta->iduzivatel = $_SESSION[logid];
$meta->akce = "Smazání uživatele";
$meta->chyba = "0";
$meta->popis = "Smazání uživatele";
$meta->stav = "0";
$meta->addrow();

echo "<div class=\"oznameni\">Uživatel byl smazán</div>";

}

function deleteRow($id){
$sql = $this->prp("DELETE FROM uzivatel WHERE iduzivatel = ?");
$sql->execute(array("$id"));
$this->sqlchyba($sql);
}

function getListByName($jmeno=NULL){
if ($_GET[limit]<1)
$i= 0;

if ($jmeno)
{
$filtr = "WHERE jmeno LIKE '%$jmeno%'";
}
else
{
$limit = "LIMIT $i,30";
}

$sql=$this->prp("SELECT iduzivatel,uzivatel,jmeno,email,telefon,plimit,osobnicislo FROM uzivatel $filtr $limit");
$sql->execute(array());
}

```

```

$this->sqlchyba($sql);
return($sql->fetchAll());
}
function getDataCount(){
    $sql = $this->prp("SELECT count(iduzivatel) FROM uzivatel");
    $sql->execute();

    return($sql->fetchColumn());
}
function editPassword($id){
    $delka = strlen("$_POST[heslo]");
    if ($delka < 3)
    {
        throw new Exception("Heslo je příliš krátké",21);
    }
    else
    {
        $pass = sha1("$_POST[heslo]");
        $sql=$this->prp("UPDATE uzivatel SET heslo = ? WHERE iduzivatel = ?");
        $sql->execute(array("$pass","$id"));
        $this->sqlchyba($sql);
        $meta = new meta_uzivatel;
        $meta->iduzivatel = $_SESSION[logid];
        $meta->akce = "Změna hesla";
        $meta->chyba = "0";
        $meta->popis ="Změna hesla";
        $meta->stav = "0";
        $meta->addrow();
        echo "<div class=\"oznameni\">Heslo bylo změněno</div>";
    }
}
function getStredisko(){
    $sql = $this->prp("SELECT strediska_idstrediska FROM strediska_ma_uzivatel WHERE uzivatel_iduzivatel = ?");
    $sql->execute(array("$_SESSION[logid]"));
    $this->sqlchyba($sql);
    $data = $sql->fetchAll();
    return($data);
}
function getLimit(){
    $sql = $this->prp("SELECT plimit
FROM uzivatel
WHERE iduzivatel=?
");
    $sql->execute(array("$_SESSION[logid]"));
    $this->sqlchyba($sql);
    return($sql->fetchColumn());
}
function getNameById($id){
    $sql = $this->prp("SELECT jmeno FROM uzivatel WHERE iduzivatel=?");
    $sql->execute(array("$id"));
    $this->sqlchyba($sql);
    return($sql->fetch());
}
}
}
?>

```

## Zarizeni.php

```

<?php
class Zarizeni extends Main {
    var $idzarizeni;
    var $kod;
    var $nazev;
    var $popis;

    function getColKod($id){
        $sql = $this->prp("SELECT kod FROM zarizeni WHERE idzarizeni=?");
        $sql->execute(array("$id"));
        $this->sqlchyba($sql);
        return($sql->fetchColumn());
    }
}
?>

```

## Index.php

```

<?php
@session_start();
ob_start();
header("Content-type: text/html; charset=utf-8");

function __autoload($jmeno_tridy){
    $jmeno_tridy = strtolower($jmeno_tridy);
    require ("class/".$jmeno_tridy.'.php');
}

```

```

$str = new String;
$gui = new guiForms;

if ($_GET[logout]==true)
{
    session_destroy();
    Header("Location: index.php");
}

$gui->show();

$data = ob_get_contents();
ob_end_clean();
$str->xmlheader();
echo $data;
$str->xmlfooter();

```

?>

## Object.css

```

html {
text-align: center;
padding: 20px 0;
font-family: Calibri,Arial,Helvetica,sans-serif;
}
#main {
width: 990px;
border: 1px solid black;
margin-left: auto;
margin-right: auto;
text-align: left;
font-size: 12px;
}
#main .hlavicka {
padding: 0.5em;
font-size: 180%;
margin: 1px;

border-top-style: solid;
border-bottom-style: solid;
border-left-style: solid;
border-right-style: solid;
border-top-color: #BFE4FF;
border-bottom-color: #BFE4FF;
border-left-color: #BFE4FF;
border-right-color: #BFE4FF;
background-color: #BFE4FF;
}

.formpole label {
margin: 0.2em;
display: block;
}
.nadpis {
vertical-align:middle;
}
.obr {
margin-left: 600px;
}
.obr img {
vertical-align: middle;
}
.loginform {
width: 480px;
margin-left:240px;
text-align: center;
}

.pravy {
width: 845px;
float:right;
}

.menu {
width: 140px;
float: left;
padding: 0;
border: 0;
}

.menu .txt {

```

```
padding: 5px;
background-color: #FF808A;
margin: 1px;
}

.pravy .txt {
}

.menu a {
display:block;
padding: 2px;
text-decoration: none;
color: black;
}

.menu a:hover {
color: gray;
font-size: 14px;
padding: 2px;
}

.noscreen {
display: none;
}

.chyba {
margin: 0.2em;
background: #E6ECFF;
border: 1px solid #E6ECFF;
width: 50%;
}

.male {

font-size: 100%;
}

.male60
{
font-size: 90%;
}

.chyba .pic {
float: left;
padding: 0.2em;
width: 50px;
height: 50px;
}

.chyba .hlavicka {
padding: 0.2em;
font-size: 110%;
border: 1px solid #BFCFFF;
background: #BFCFFF;
}

.chyba .hlavicka .text {
margin-right: 5px;
}

.chyba .hlavicka .number {
font-weight: bold;
}

.chyba .text {
text-align: center;
}

.divaider {
width: 0;
clear: both;
border-color: white;
border: 0;
}

form,input,button {
margin: 0px 0px 0px 0px;
}

label {
margin: 2px;
}

#paticka {
border: 1px solid #BFE4FF;
background-color: #BFE4FF;
padding: 2px;
margin: 1px;
font-size: 80%;
}

#paticka .txt {
text-align:center;
}

h1 {
color: #6096BF;
margin: 1px;
padding: 1px;
}
```

```

}
.submenu {
margin: 2px;
/*width: 200px;*/
}
.tablemenu td{
background-color: #BFE4FF;
text-align: center;
}
hr {
color: #BFE4FF;
}
.submenu a {
font-size: 100%;
text-decoration: none;
background-color: #BFE4FF;
padding-left: 2px;
padding-right: 2px;
padding-top: 2px;
color: black;

padding: 1px;
width: 100%;
min-width: 100%;
}
.submenu a:hover {
background-color: #FF808A;
}
.info table{
text-align:center;
font-size: 80%;
}
.form {

}
.zmensit_popis{
}

.big {
font-size: 120%;
font-weight: bold;
}
.big2{
font-size: 160%;
}
.mtable td,th{
background-color: white;
padding: 3px;
}
.mtable {
background-color:black;
}
.zmensit_pismo {
font-size:80%;
}

.lb {
margin: 0px;
margin-right: 2px;
}

.lb a {
border: 2px outset #D4D4D4;
padding: 1px;
color:black;
background-color: #F2F2F2;
text-decoration: none;
font-size: 12px;
}
.lb a:active {
border: 2px inset;
}

.noshow{
border: 0px white;
}
.inline {
margin: 0;
padding: 0;
display:inline;
}
.colored {
background-color:#FF808A;
border: 1px solid black;
}
.operaready {

```



```
        border: 1px solid black;
    }
    fieldset legend
    {
        border: 1px solid black;
        background-color: white;
    }

    .table td {
        background-color:white;
    }

    @media print {
    #main .hlavicka {
        display:none;
    }
    .submenu {
        display:none;
    }
    .menu {
        display: none;
    }
    .pravy {
        padding-top: 10px;
        width: 950px;
    }
    h2 {
        display: none;
    }
    .pnone {
        display: none;
    }
    .mtable td,th{
        border: 1px solid black;
    }

    .mtable {
        background-color:black;
    }
    .mtable tr {
        border: 1px solid black;
    }
    #main {
        border: 0px solid white;
    }
    }
```

## Objednavky.sql

```
SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `objednavky` DEFAULT CHARACTER SET utf8 COLLATE utf8_bin ;
USE `objednavky`;

-----
-- Table `objednavky`.`uzivatel`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`uzivatel` (
  `iduzivatel` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `uzivatel` VARCHAR(45) NOT NULL ,
  `heslo` VARCHAR(40) NOT NULL ,
  `jmeno` VARCHAR(100) NOT NULL ,
  `email` VARCHAR(200) NULL ,
  `telefon` VARCHAR(200) NULL ,
  `plimit` FLOAT(20,4) NOT NULL ,
  `osobnicislo` VARCHAR(45) NULL ,
  `klic` VARCHAR(40) NOT NULL ,
  PRIMARY KEY (`iduzivatel`) )
ENGINE = InnoDB;

-----
-- Table `objednavky`.`prava`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`prava` (
  `idprava` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `cislo` INT NOT NULL ,
  `nazev` VARCHAR(100) NOT NULL ,
  `popis` TEXT NULL ,
  PRIMARY KEY (`idprava`) )
ENGINE = InnoDB;

-----
-- Data for table `objednavky`.`prava`
-----
SET AUTOCOMMIT=0;
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 1, 'vytvoření objednávky za své nákladové středisko', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 2, 'vytvoření objednávky za jakékoliv nákladové středisko', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 3, 'schválit objednávku za své nákladové středisko', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 4, 'schválit objednávku za všechna nákladová střediska', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 5, 'aktivace a vyřízení objednávky za své středisko', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 6, 'aktivace a vyřízení objednávky za všechna střediska', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 7, 'Plnění a schvalování skladu', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 8, 'Plnění dočasného skladu', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 9, 'Práce s nákladovými středisky', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 10, 'Práce s uživatelem', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 11, 'Přidělování práv', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 12, 'Vkládat smlouvy', NULL);
INSERT INTO `prava` (`idprava`, `cislo`, `nazev`, `popis`) VALUES (NULL, 13, 'Prohlížet smlouvy', NULL);

COMMIT;

-----
-- Table `objednavky`.`uzivatel_ma_prava`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`uzivatel_ma_prava` (
  `uzivatel_iduzivatel` INT UNSIGNED NOT NULL ,
  `prava_idprava` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`uzivatel_iduzivatel`, `prava_idprava`),
  CONSTRAINT `fk_uzivatel_ma_prava_uzivatel`
    FOREIGN KEY (`uzivatel_iduzivatel`)
      REFERENCES `objednavky`.`uzivatel` (`iduzivatel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_uzivatel_ma_prava_prava`
    FOREIGN KEY (`prava_idprava`)
      REFERENCES `objednavky`.`prava` (`idprava`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE INDEX fk_uzivatel_ma_prava_uzivatel ON `objednavky`.`uzivatel_ma_prava` (`uzivatel_iduzivatel` ASC) ;

CREATE INDEX fk_uzivatel_ma_prava_prava ON `objednavky`.`uzivatel_ma_prava` (`prava_idprava` ASC) ;

-----
-- Table `objednavky`.`zarizeni`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`zarizeni` (
  `idzarizeni` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `kod` VARCHAR(45) NOT NULL ,
  `nazev` VARCHAR(200) NOT NULL ,
  `popis` TEXT NULL ,
  PRIMARY KEY (`idzarizeni`) )
ENGINE = InnoDB;
```

```

-----
-- Table `objednavky`.`strediska`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`strediska` (
  `idstrediska` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `idzarizeni` INT UNSIGNED NOT NULL ,
  `cislo` VARCHAR(45) NOT NULL ,
  `popis` TEXT NULL ,
  `plimit` FLOAT(20,2) NULL ,
  `nemocnice` INT NULL DEFAULT 0 ,
  PRIMARY KEY (`idstrediska`) ,
  CONSTRAINT `fk_strediska_zarizeni`
    FOREIGN KEY (`idzarizeni`)
    REFERENCES `objednavky`.`zarizeni` (`idzarizeni`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_strediska_zarizeni ON `objednavky`.`strediska` (`idzarizeni` ASC) ;

-----
-- Table `objednavky`.`strediska_ma_uzivatel`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`strediska_ma_uzivatel` (
  `strediska_idstrediska` INT UNSIGNED NOT NULL ,
  `uzivatel_iduzivatel` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`strediska_idstrediska`, `uzivatel_iduzivatel`) ,
  CONSTRAINT `fk_strediska_ma_uzivatel_strediska`
    FOREIGN KEY (`strediska_idstrediska`)
    REFERENCES `objednavky`.`strediska` (`idstrediska`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_strediska_ma_uzivatel_uzivatel`
    FOREIGN KEY (`uzivatel_iduzivatel`)
    REFERENCES `objednavky`.`uzivatel` (`iduzivatel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE INDEX fk_strediska_ma_uzivatel_strediska ON `objednavky`.`strediska_ma_uzivatel` (`strediska_idstrediska` ASC) ;
CREATE INDEX fk_strediska_ma_uzivatel_uzivatel ON `objednavky`.`strediska_ma_uzivatel` (`uzivatel_iduzivatel` ASC) ;

-----
-- Table `objednavky`.`dodavatel`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`dodavatel` (
  `iddodavatel` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `jmeno` VARCHAR(200) NOT NULL ,
  `ulice` VARCHAR(200) NOT NULL ,
  `mesto` VARCHAR(200) NOT NULL ,
  `psc` VARCHAR(20) NOT NULL ,
  `ic` VARCHAR(45) NULL ,
  `dic` VARCHAR(45) NULL ,
  `kontakt` VARCHAR(200) NOT NULL ,
  `telefon` VARCHAR(200) NOT NULL ,
  `email` VARCHAR(200) NULL ,
  `vlozeno` DATETIME NOT NULL ,
  `zmeneno` DATETIME NOT NULL ,
  PRIMARY KEY (`iddodavatel`) )
ENGINE = InnoDB;

-----
-- Table `objednavky`.`hlskupiny`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`hlskupiny` (
  `idhlskupiny` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `nazev` VARCHAR(100) NOT NULL ,
  `popis` TEXT NULL ,
  PRIMARY KEY (`idhlskupiny`) )
ENGINE = InnoDB;

-----
-- Table `objednavky`.`nadskupiny`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`nadskupiny` (
  `idnadskupiny` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `idhlskupiny` INT UNSIGNED NOT NULL ,
  `nazev` VARCHAR(100) NOT NULL ,
  `popis` TEXT NOT NULL ,
  PRIMARY KEY (`idnadskupiny`) ,
  CONSTRAINT `fk_nadskupiny_hlskupiny`
    FOREIGN KEY (`idhlskupiny`)
    REFERENCES `objednavky`.`hlskupiny` (`idhlskupiny`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_nadskupiny_hlskupiny ON `objednavky`.`nadskupiny` (`idhlskupiny` ASC) ;

-----
-- Table `objednavky`.`skupiny`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`skupiny` (
  `idskupiny` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `nazev` VARCHAR(100) NOT NULL ,

```

```

`popis` TEXT NULL ,
`idnaskupiny` INT UNSIGNED NOT NULL ,
PRIMARY KEY (`idskupiny`),
CONSTRAINT `fk_skupiny_nadskupiny`
  FOREIGN KEY (`idnaskupiny`)
  REFERENCES `objednavky`.`nadskupiny` (`idnadskupiny`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_skupiny_nadskupiny ON `objednavky`.`skupiny` (`idnaskupiny` ASC);

-----
-- Table `objednavky`.`sklad`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`sklad` (
  `idsklad` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `vlozil` INT UNSIGNED NOT NULL ,
  `dodavatel` INT UNSIGNED NOT NULL ,
  `idskupiny` INT UNSIGNED NOT NULL ,
  `nazev` VARCHAR(200) NOT NULL ,
  `jednotky` VARCHAR(45) NOT NULL ,
  `cena` FLOAT(20,2) NOT NULL ,
  `cislo` VARCHAR(20) NOT NULL ,
  `popis` TEXT NULL ,
  `vlozeno` DATETIME NOT NULL ,
  `zmeneno` DATETIME NOT NULL ,
  `viditelne` INT NOT NULL DEFAULT 1 ,
  `temp` INT NOT NULL DEFAULT 0 ,
  `lekarsky` INT NOT NULL DEFAULT 0 ,
  `dph` INT NOT NULL DEFAULT 0 ,
  `sazbadph` INT NOT NULL ,
  PRIMARY KEY (`idsklad`),
  CONSTRAINT `fk_sklad_dodavatel`
    FOREIGN KEY (`dodavatel`)
    REFERENCES `objednavky`.`dodavatel` (`iddodavatel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_sklad_uzivatel`
    FOREIGN KEY (`vlozil`)
    REFERENCES `objednavky`.`uzivatel` (`iduzivatel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_sklad_skupiny`
    FOREIGN KEY (`idskupiny`)
    REFERENCES `objednavky`.`skupiny` (`idskupiny`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_sklad_dodavatel ON `objednavky`.`sklad` (`dodavatel` ASC);
CREATE INDEX fk_sklad_uzivatel ON `objednavky`.`sklad` (`vlozil` ASC);
CREATE INDEX fk_sklad_skupiny ON `objednavky`.`sklad` (`idskupiny` ASC);

-----
-- Table `objednavky`.`oddeleni`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`oddeleni` (
  `idoddeleni` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `nazev` VARCHAR(45) NOT NULL ,
  `popis` TEXT NULL ,
  PRIMARY KEY (`idoddeleni`))
ENGINE = InnoDB;

-----
-- Data for table `objednavky`.`oddeleni`
-----
SET AUTOCOMMIT=0;
INSERT INTO `oddeleni` (`idoddeleni`, `nazev`, `popis`) VALUES (NULL, 'DZ', 'Dopravní zdravotnictví');
INSERT INTO `oddeleni` (`idoddeleni`, `nazev`, `popis`) VALUES (NULL, 'L', 'Lékař');
INSERT INTO `oddeleni` (`idoddeleni`, `nazev`, `popis`) VALUES (NULL, 'LAB', 'Laboratoř');
INSERT INTO `oddeleni` (`idoddeleni`, `nazev`, `popis`) VALUES (NULL, 'OIS', 'Oddělení informačních systémů');
INSERT INTO `oddeleni` (`idoddeleni`, `nazev`, `popis`) VALUES (NULL, 'PTO', 'Obslužný personál');

COMMIT;

-----
-- Table `objednavky`.`uzivatel_ma_oddeleni`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`uzivatel_ma_oddeleni` (
  `uzivatel_iduzivatel` INT UNSIGNED NOT NULL ,
  `oddeleni_idoddeleni` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`uzivatel_iduzivatel`, `oddeleni_idoddeleni`),
  CONSTRAINT `fk_uzivatel_ma_oddeleni_uzivatel`
    FOREIGN KEY (`uzivatel_iduzivatel`)
    REFERENCES `objednavky`.`uzivatel` (`iduzivatel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_uzivatel_ma_oddeleni_oddeleni`
    FOREIGN KEY (`oddeleni_idoddeleni`)
    REFERENCES `objednavky`.`oddeleni` (`idoddeleni`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE INDEX fk_uzivatel_ma_oddeleni_uzivatel ON `objednavky`.`uzivatel_ma_oddeleni` (`uzivatel_iduzivatel` ASC);

```

```
CREATE INDEX fk_uzivatel_ma_oddeleni_oddeleni ON `objednavky`.`uzivatel_ma_oddeleni` (`oddeleni_idoddeleni` ASC) ;
```

```
-----  
-- Table `objednavky`.`objednavky`  
-----
```

```
CREATE TABLE IF NOT EXISTS `objednavky`.`objednavky` (  
  `idobjednavky` INT UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `iduzivatel` INT UNSIGNED NOT NULL ,  
  `idstrediska` INT UNSIGNED NOT NULL ,  
  `kdy` DATETIME NOT NULL ,  
  `stav` INT NOT NULL ,  
  `nemocnice` INT NOT NULL ,  
  `internicislo` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`idobjednavky`) ,  
  CONSTRAINT `fk_objednavky_uzivatel`  
    FOREIGN KEY (`iduzivatel` )  
    REFERENCES `objednavky`.`uzivatel` (`iduzivatel` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION ,  
  CONSTRAINT `fk_objednavky_strediska`  
    FOREIGN KEY (`idstrediska` )  
    REFERENCES `objednavky`.`strediska` (`idstrediska` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION )  
ENGINE = InnoDB;
```

```
CREATE INDEX fk_objednavky_uzivatel ON `objednavky`.`objednavky` (`iduzivatel` ASC) ;
```

```
CREATE INDEX fk_objednavky_strediska ON `objednavky`.`objednavky` (`idstrediska` ASC) ;
```

```
-----  
-- Table `objednavky`.`polozky`  
-----
```

```
CREATE TABLE IF NOT EXISTS `objednavky`.`polozky` (  
  `idpolozky` INT UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `idobjednavky` INT UNSIGNED NOT NULL ,  
  `idsklad` INT UNSIGNED NOT NULL ,  
  `nazev` VARCHAR(200) NOT NULL ,  
  `jednotky` VARCHAR(45) NOT NULL ,  
  `cena` FLOAT(20,2) NOT NULL ,  
  `cislo` VARCHAR(45) NOT NULL ,  
  `popis` TEXT NULL ,  
  `mnozstvi` FLOAT(10) NOT NULL ,  
  `stav` INT NOT NULL ,  
  PRIMARY KEY (`idpolozky`) ,  
  CONSTRAINT `fk_polozky_objednavky`  
    FOREIGN KEY (`idobjednavky` )  
    REFERENCES `objednavky`.`objednavky` (`idobjednavky` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION ,  
  CONSTRAINT `fk_polozky_sklad`  
    FOREIGN KEY (`idsklad` )  
    REFERENCES `objednavky`.`sklad` (`idsklad` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION )  
ENGINE = InnoDB;
```

```
CREATE INDEX fk_polozky_objednavky ON `objednavky`.`polozky` (`idobjednavky` ASC) ;
```

```
CREATE INDEX fk_polozky_sklad ON `objednavky`.`polozky` (`idsklad` ASC) ;
```

```
-----  
-- Table `objednavky`.`stav`  
-----
```

```
CREATE TABLE IF NOT EXISTS `objednavky`.`stav` (  
  `idstav` INT UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `nazev` VARCHAR(45) NOT NULL ,  
  `popis` TEXT NULL ,  
  PRIMARY KEY (`idstav` ) )  
ENGINE = InnoDB;
```

```
-----  
-- Table `objednavky`.`meta_objednavky`  
-----
```

```
CREATE TABLE IF NOT EXISTS `objednavky`.`meta_objednavky` (  
  `idmeta_objednavky` INT UNSIGNED NOT NULL AUTO_INCREMENT ,  
  `idobjednavky` INT UNSIGNED NOT NULL ,  
  `idstav` INT UNSIGNED NOT NULL ,  
  `kdy` DATETIME NULL ,  
  `iduzivatele` INT UNSIGNED NOT NULL ,  
  PRIMARY KEY (`idmeta_objednavky`) ,  
  CONSTRAINT `fk_meta_objednavky_objednavky`  
    FOREIGN KEY (`idobjednavky` )  
    REFERENCES `objednavky`.`objednavky` (`idobjednavky` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION ,  
  CONSTRAINT `fk_meta_objednavky_uzivatel`  
    FOREIGN KEY (`iduzivatele` )  
    REFERENCES `objednavky`.`uzivatel` (`iduzivatel` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION ,  
  CONSTRAINT `fk_meta_objednavky_stav`  
    FOREIGN KEY (`idstav` )  
    REFERENCES `objednavky`.`stav` (`idstav` )  
    ON DELETE NO ACTION )
```

```

        ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_meta_objednavky_objednavky ON `objednavky`.`meta_objednavky` (`idobjednavky` ASC) ;

CREATE INDEX fk_meta_objednavky_uzivatel ON `objednavky`.`meta_objednavky` (`iduzivatele` ASC) ;

CREATE INDEX fk_meta_objednavky_stav ON `objednavky`.`meta_objednavky` (`idstav` ASC) ;

-----
-- Table `objednavky`.`meta_uzivatel`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`meta_uzivatel` (
  `idmeta_uzivatel` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `iduzivatel` INT UNSIGNED NOT NULL ,
  `akce` VARCHAR(50) NOT NULL ,
  `kdy` DATETIME NOT NULL ,
  `ip` VARCHAR(45) NOT NULL ,
  `stav` INT UNSIGNED NOT NULL ,
  `chyba` INT UNSIGNED NOT NULL ,
  `popis` TEXT NOT NULL ,
  PRIMARY KEY (`idmeta_uzivatel`),
  CONSTRAINT `fk_meta_uzivatel_uzivatel`
    FOREIGN KEY (`iduzivatel`)
      REFERENCES `objednavky`.`uzivatel` (`iduzivatel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_meta_uzivatel_uzivatel ON `objednavky`.`meta_uzivatel` (`iduzivatel` ASC) ;

-----
-- Table `objednavky`.`temp_objednavky`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`temp_objednavky` (
  `idtemp_objednavky` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `internicislo` INT UNSIGNED NOT NULL ,
  `iduzivatel` INT UNSIGNED NOT NULL ,
  `idsklad` INT UNSIGNED NOT NULL ,
  `mnozstvi` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`idtemp_objednavky`),
  CONSTRAINT `fk_temp_objednavky_sklad`
    FOREIGN KEY (`idsklad`)
      REFERENCES `objednavky`.`sklad` (`idsklad`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_temp_objednavky_sklad ON `objednavky`.`temp_objednavky` (`idsklad` ASC) ;

-----
-- Table `objednavky`.`spolecnosti`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`spolecnosti` (
  `idspolecnosti` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `iduzivatel` INT UNSIGNED NOT NULL ,
  `nazev` TEXT NOT NULL ,
  `popis` LONGTEXT NOT NULL ,
  `vlozena` DATETIME NOT NULL ,
  PRIMARY KEY (`idspolecnosti`),
  CONSTRAINT `fk_spolecnosti_uzivatel`
    FOREIGN KEY (`iduzivatel`)
      REFERENCES `objednavky`.`uzivatel` (`iduzivatel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_spolecnosti_uzivatel ON `objednavky`.`spolecnosti` (`iduzivatel` ASC) ;

-----
-- Table `objednavky`.`hlavni_skupina`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`hlavni_skupina` (
  `idhlavni_skupina` INT UNSIGNED NOT NULL ,
  `nazev` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`idhlavni_skupina`) )
ENGINE = InnoDB;

-----
-- Table `objednavky`.`podskupina`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`podskupina` (
  `idpodskupina` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `nazev` VARCHAR(45) NOT NULL ,
  `idhlavni_skupina` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`idpodskupina`),
  CONSTRAINT `fk_podskupina_hlavni_skupina`
    FOREIGN KEY (`idhlavni_skupina`)
      REFERENCES `objednavky`.`hlavni_skupina` (`idhlavni_skupina`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_podskupina_hlavni_skupina ON `objednavky`.`podskupina` (`idhlavni_skupina` ASC) ;

```

```

-----
-- Table `objednavky`.`smlouvy`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`smlouvy` (
  `idsmouvy` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `idpodskupiny` INT UNSIGNED NOT NULL ,
  `iduzivatel` INT UNSIGNED NOT NULL ,
  `idspolecnosti` INT UNSIGNED NOT NULL ,
  `nazev` TEXT NOT NULL ,
  `popis` LONGTEXT NOT NULL ,
  `priloha` TEXT NOT NULL ,
  `verze` DATETIME NOT NULL ,
  PRIMARY KEY (`idsmouvy`) ,
  CONSTRAINT `fk_s mouvy_uzivatel`
    FOREIGN KEY (`iduzivatel`)
      REFERENCES `objednavky`.`uzivatel` (`iduzivatel`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_s mouvy_spolecnosti`
    FOREIGN KEY (`idspolecnosti`)
      REFERENCES `objednavky`.`spolecnosti` (`idspolecnosti`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_s mouvy_podskupina`
    FOREIGN KEY (`idpodskupiny`)
      REFERENCES `objednavky`.`podskupina` (`idpodskupina`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_s mouvy_uzivatel ON `objednavky`.`smlouvy` (`iduzivatel` ASC) ;

CREATE INDEX fk_s mouvy_spolecnosti ON `objednavky`.`smlouvy` (`idspolecnosti` ASC) ;

CREATE INDEX fk_s mouvy_podskupina ON `objednavky`.`smlouvy` (`idpodskupiny` ASC) ;

-----
-- Table `objednavky`.`meta_smlouvy`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`meta_smlouvy` (
  `idmeta_smlouvy` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `idsmlouvy` INT UNSIGNED NOT NULL ,
  `akce` VARCHAR(45) NOT NULL ,
  `iduzivatel` INT UNSIGNED NOT NULL ,
  `kdy` DATETIME NOT NULL ,
  `popis` TEXT NULL ,
  PRIMARY KEY (`idmeta_smlouvy`) ,
  CONSTRAINT `fk_meta_smlouvy_s mouvy`
    FOREIGN KEY (`idsmlouvy`)
      REFERENCES `objednavky`.`smlouvy` (`idsmouvy`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_meta_smlouvy_s mouvy ON `objednavky`.`meta_smlouvy` (`idsmlouvy` ASC) ;

-----
-- Table `objednavky`.`objednavkytxt`
-----
CREATE TABLE IF NOT EXISTS `objednavky`.`objednavkytxt` (
  `idobjednavkytxt` INT NOT NULL AUTO_INCREMENT ,
  `idobjednavky` INT UNSIGNED NOT NULL ,
  `poznamka` TEXT NOT NULL ,
  `datum` DATETIME NOT NULL ,
  PRIMARY KEY (`idobjednavkytxt`) ,
  CONSTRAINT `fk_objednavkytxt_objednavky`
    FOREIGN KEY (`idobjednavky`)
      REFERENCES `objednavky`.`objednavky` (`idobjednavky`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX fk_objednavkytxt_objednavky ON `objednavky`.`objednavkytxt` (`idobjednavky` ASC) ;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```