

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Jiří Kunčar

### Informační systém pro jazykovou agenturu

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Miroslav Spousta

Studijní program: informatika, správa počítačových systémů

2009

Děkuji panu RNDr. Miroslavu Spoustovi za pomoc, připomínky, cenné rady a za odborné vedení bakalářské práce. Dále bych rád poděkoval firmě Primalingua s.r.o. za poskytnutí prostředků na vývoj aplikace. Speciální poděkování patří především paní majitelce Mgr. Miluši Psotové a paní RNDr. Jitce Kunčarové, která vývoj informačního systému iniciovala.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 6.8.2009

Jiří Kunčar

# Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Analýza úlohy</b>                    | <b>8</b>  |
| 1.1      | Požadavky klienta . . . . .             | 8         |
| 1.2      | Definice a upřesnění pojmů . . . . .    | 9         |
| <b>2</b> | <b>Existující implementace</b>          | <b>11</b> |
| <b>3</b> | <b>Návrh řešení</b>                     | <b>13</b> |
| <b>4</b> | <b>Použité technologie a frameworky</b> | <b>14</b> |
| 4.1      | Server . . . . .                        | 14        |
| 4.2      | Klient . . . . .                        | 16        |
| <b>5</b> | <b>Programátorská dokumentace</b>       | <b>18</b> |
| 5.1      | Databázové schéma . . . . .             | 18        |
| 5.2      | Adresářová struktura . . . . .          | 18        |
| <b>6</b> | <b>Uživatelská dokumentace</b>          | <b>19</b> |
| 6.1      | Instalace serveru . . . . .             | 19        |
| 6.2      | Umístění aplikace . . . . .             | 20        |
| 6.3      | Inicializace databáze . . . . .         | 20        |
| 6.4      | Upřesňující informace . . . . .         | 21        |
| 6.5      | Správce (hlavní manažer) . . . . .      | 21        |
| 6.6      | Dodavatel . . . . .                     | 25        |

Název práce: Informační systém pro jazykovou agenturu  
Autor: Jiří Kunčar  
Katedra (ústav): Ústav formální a aplikované lingvistiky  
Vedoucí bakalářské práce: RNDr. Miroslav Spousta  
e-mail vedoucího: Miroslav.Spousta@mff.cuni.cz

Abstrakt: Cílem práce je navrhnout a implementovat modulární informační systém pro firmu zabývající se výukou jazyků.

Součástí informačního systému bude zejména:

- modul pro nabídku a prodej kurzů pro veřejnost, individuální výuku, jazykovou výuku pro podniky, překlady a tlumočení
- modul produkty (výuka, překlady, tlumočení), výuka (individuální výuka, kurzy pro veřejnost, jazyková výuka pro podniky)
- modul pro News (hromadné rozesílání mailem), rozvrh výuky, třídní knihy, studijní materiály
- modul dodavatelé (smlouvy s lektory, překladateli, tlumočníky, měsíční výkazy odpracovaných hodin, měsíční přehledy nákladů na lektory, překladatele, tlumočníky)
- modul odběratelé (zápisy do kurzů pro veřejnost, objednávky, smlouvy, přílohy faktur za období od-do)
- modul pro testování znalostí (jednoduché testovací prostředí)
- modul pro správu IS (uživatelé, jejich práva, přehled změn).

Klíčová slova: informační systém, PHP, MySQL

Title: Information system of a language school

Author: Jiří Kunčar

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Miroslav Spousta

Supervisor's e-mail address: Miroslav.Spousta@mff.cuni.cz

Abstract: The goal of the thesis is to design and implement a modular information system for a company involved in teaching of foreign languages.

The main parts of the information system will include, particularly:

- a module for offer and sale of courses to the public and individuals, language training for businesses, translation and interpretation
- a module of products (teaching, translation and interpretation), training

- (individual training courses for the public, language classes for businesses)
- a module for News (sending bulk mail), the schedule of teaching, class books, study material
  - a module of suppliers (contracts with teachers, translators, interpreters, monthly class-sheets, monthly reports on the cost of staff translators and interpreters)
  - a module of customers (registration in courses, orders, contracts, supplements to invoices for "from-to" periods)
  - a module for testing of knowledge levels (a simple testing environment)
  - a module for managing the IS (users, their rights and a summary of changes)

Keywords: information system, PHP, MySQL

# Úvod

Informační systémy hrají v rychle se rozvíjejícím prostředí svou nepostradatelnou roli a Internet jim poskytl výbornou platformu umožňující další rozvoj v dříve nepříliš využívaném prostředí. Umožňují snadné a rychlé zpracování, vyhledávání a publikování informací, které se díky němu stávají dostupné pro širší spektrum zákazníků, dodavatelů i samotných zaměstanců.

S využíváním Internetu, ale zároveň rostou nároky na funkce, bezpečnost a dostupnost uložených informací, které jsou pro chod firem životně důležité. Toto zvyšuje požadavky na robustnost řešení včetně nároků na nástroje použité při implementaci systému.

Před zavádění informačního systému je nutné stanovit, čeho chce firma s využitím nového systému v daném čase dosáhnout. Tomu by měla pomoci důkladná analýza firemních procesů a jejich optimalizace.

*„Kdo nepozná vlastní firemní procesy, nemůže je zlepšovat.”[citát]*

Z původních jednoduchých aplikací na  
Informačný systém, Client/Server.

## Cíle práce

Cílem práce je návrh a implementace informačního systému usnadňující činnost a spolupráci více subjektů za použití vhodných opensource technologií a frameworků v použitých programovacích jazycích.

Cílem práce není vytvořit, vzhledem k náročným právním úpravám a častým změnám, komplexní účetní program, ale systém umožní vytvořit přehledy pro účetní evidenci, které mohou být zavedeny do některého již existujícího programu.

## Obsah práce

- Analýza úlohy
- Požadavky
- Existující implementace - IS na míru vs. hotová řešení
- Návrh vlastní implementace
  - bezpečnost: Sanitize,
    - \* SQL Injection, Cross Site Scripting
  - výkonnost (použití cache)
  - optimalizace GET/POST požadavků na stránku
- Programátorská dokumentace
- Uživatelská dokumentace

Poznámky k prepsání....

Ve druhé kapitole této práce je provedena analýza úlohy s ohledem na několik motivačních praktických příkladů. V této kapitole jsou mimo jiné stanoveny požadavky na řešení a je uveden přehled obdobných existujících implementací.

Ve třetí kapitole jsou stručně popsány technologie dále používané v této práci a je ukázáno jejich použití.

Čtvrtá kapitola popisuje návrh vlastní implementace s ohledem na požadavky stanovené v kapitole druhé. Jsou zde základní návrhová rozhodnutí a důvody pro volbu konkrétních řešení.

Pátá kapitola obsahuje programátorskou dokumentaci. Je zde popsáno technické řešení implementace a jsou nastíněny některé problémy, které bylo při implementaci nutné řešit.

Uživatelská dokumentace je obsažena v šesté kapitole. Popisuje uživatelské rozhraní vytvořených aplikací a ukazuje jejich použití na konkrétních příkladech.

V závěru jsou přehledně shrnuty výsledky, kterých bylo při vývoji systému dosaženo, a jsou naznačeny možnosti dalšího rozšíření.

# Kapitola 1

## Analýza úlohy

### 1.1 Požadavky klienta

Na začátku stála žádost manažerky agentury zjednodušit proces zpracování měsíčních výkazů lektorů, překladatelů a tlumočnicků tvořících podklady pro mzdy a fakturaci služeb. Tyto měsíční výkazy neměly jednotný vzor a ani nebylo jednoduše možné, bez znalostí místních poměrů přiřadit vykázanou činnost k jednotlivým produktům.

### Zavedení jednotných identifikátorů

Kvůli výše uvedeným problémům se začalo s postupným zaváděním jednotných identifikátorů závazných pro všechny zúčastněné strany. Tento krok byl ze začátku velmi těžce snášen a trvalo několik měsíců, než se tento proces tvorby ustálil a začal být všemi akceptován.

### Měsíční výkazy

Vzhledem k velkým rozdílům mezi jednotlivými lektory, překladateli a tlumočníky, bylo rozhodnuto, že se v první fázi začne s jednotným papírovým formulářem [ @todo odkaz na přílohu ]. Ten bude následně zkontrolován administrativním pracovníkem a vložen do systému, aby se předešlo chybám.



## **Podklady pro fakturaci**

Se vzrůstajícím objemem překladů a odučených hodin přestávalo být únosné ruční vytváření měsíčních přehledů pro zákazníky. Ti si přáli být informováni nejen o počtu odučených hodin, ale i o všech změnách zavedeném měsíčním rozvrhu a případném suplování. Zároveň musely být v systému zachyceny vazby mezi produktem, zákazníkem a vlastními účastníky tak, aby bylo možné stanovit výslednou cenu zakázky, která může být závislá nejen od počtu odučených hodin či přeložených stran, ale i od počtu účastníků.

## **Oddělení reálných dat více agentur**

Vzhledem ke složitým poměrům v agentuře bylo potřeba oddělit evidenci zakázek vyřizovaných manažerem pro různé agentury a fyzické osoby, jenž s agenturami úzce spolupracují.

## **Úzká provázanost s webem**

Evidence hodin, rozvrhy učeben

## **Vícejazykový systém**

Vzhledem k tomu, že je systém určen primárně pro jazykové agentury, kde se počítá s komunikací se zákazníky v jiném než českém jazyce, bylo k tomu přizpůsobit systém již od počátku.

# **1.2 Definice a upřesnění pojmů**

## **Agentura**

## **Infomační systém**

Informační systém (IS) je systém pro sběr, udržování, zpracování a poskytování informací a dat[IS].

**Produkt**

**Zakázka**

CSS

## Kapitola 2

# Existující implementace

Existující aplikace lze rozdělit na několik skupin, kde ovšem žádná nepokrývá veškeré požadavky v plném rozsahu. První se více zaměřuje na správu webového obsahu. Označované anglickou zkratkou CMS - Content Management System, před kterou se někdy přidává W označující webové systémy. Tyto systémy lze rozdělit do podskupin podle způsobu, jakým prezentují uložená data.

**Offline zpracování** Server vygeneruje statické HTML před samotnou publikací, proto takové systémy nepotřebují, aby server aplikoval šablony na data při každém požadavku. Výhody jsou zjevné pro vytížené CMS používané převážně pro čtení dat. Naopak při častých změnách dat mohou být tyto systémy pomalé nebo neaktuální, proto je nutné si jejich nasazení řádně rozmyslet. Příkladem takového systému je například Vignette CMS<sup>1</sup>.

**Online zpracování** Server generuje prezentační data až na základě požadavků klienta. Pro keše

**E-learning** ... Moodle

**ciele:**

\* strategické (plánovanie investícií...) \* taktické (vedenie, kontrola rozpočtu...) \* operatívne (každodenná rutina)

**Dôležité** sú tiež úlohy IS:

---

<sup>1</sup><http://www.vignette.com/>

\* manažérske (EIS - Executive IS) \* taktické (DSS - Decision Support System) \* vedenie (MIS - Management IS) \* expertné (KWS - Knowledge Work System) \* kancelárske (OIS - Office IS) \* operatívne o TPS - transakčné (banky, ...) o CRM - vzťahy so zákazníkmi o RIS - rezervačné systémy o CAM - konštrukčné (CAD, ...) o GIS - geografické systémy

# Kapitola 3

## Návrh řešení

Evidence produktů

Podklady pro fakturaci dodavatelům a mzdy

Finanční vyrovnaní subjektů v IS

# Kapitola 4

## Použité technologie a frameworky

Použité technologie byly částečně determinovány požadavy klienta<sup>1.1</sup>. Tím, že se mělo jednat o systém využívající výhradně open-souce technologie tak, aby nebyly zvyšovány náklady na nákup licencí.

### 4.1 Server

Tento webový informační systém využívá technologii PHP (jazyk, interpret a knihovny), která vychází se skriptovacího víceúčelového jazyka, jenž byl původně vyvinut pro tvorbu dynamických webových stránek. Z tohoto využití vznikla i zkratka z anglických slov *Personal Home Page*, které byly nahrazeny slovy *PHP: Hypertext Preprocessor* dající vznik rekurzivní zkratce<sup>1</sup>.

Výhodou použití PHP je existence interpretu pro různé operační systémy a podobnost jeho syntaxe s C, Javou.

Nevýhodou, která brzdí dalšímu rozvoji a rozšíření, je absence normy (k datu vydání BP). Jazyk je tak de facto standardizovaný interpretem a množstvím lidí<sup>2</sup>, kteří jej využívají. I když existují mnohé polemiky a živé diskuze mezi jeho zastánci a odpůrci o jeho výkonnosti, bezpečnosti a vhodnosti pro velké projekty, existují výjimky<sup>3</sup>, které tyto názory vyvrací a zároveň se podílejí na vývoji, a tak se snaží přispět k jeho větší výkonnosti a bezpečnosti.

Vývoj jazyka sebou nese i stinné stránky. Mezi ty nejpodstatnější patří úpravy API<sup>4</sup> některých vestavěných funkcí a změna syntaxe. To může zapří-

---

<sup>1</sup>[http://cs.wikipedia.org/wiki/Rekurzivní\\_zkratka](http://cs.wikipedia.org/wiki/Rekurzivní_zkratka)

<sup>2</sup>dle statistiky na <http://www.php.net/usage.php>

<sup>3</sup>Facebook, YouTube, Wikipedia (MediaWiki) a další

<sup>4</sup>anglická zkratka *Application Programming Interface*, označuje sadu funkcí, procedur

činit, že po aktualizaci interpretu jazyka, přestanou fungovat některé části nebo celá aplikace. Řešením ovšem není zůstat na několik let staré verzi, ve které mohly být objeveny chyby.

Pokud programátor nechce nebo nemůže přepisovat kód může využít služeb některého z frameworků nad daným jazykem, který se snaží tyto rozdíly ve verzích zakrýt. Další nespornou výhodou používání frameworků spočívá v jednodušším vývoji aplikací a minimalizaci rizika chyb v jinak ručně psaném jádru aplikace. Toto je zajištěno pouze pokud má kvalitní a úplnou dokumentaci a je zastřešen silnou komunitou nebo společností zajišťující jeho vývoj.

Při výběru byl proto důraz kladen hlavně na kvalitní dokumentaci, rozšiřitelnost a možnost práce s různými relačními databázemi. Frameworky, které splňují většinu požadavků jsou Zend<sup>5</sup>, Symfony a CakePHP.

## CakePHP

Nám dává příležitost věnovat se návrhu schématu. MVC - Model-View-Controller - <http://en.wikipedia.org/wiki/Model-view-controller>

Odstínění od způsobu práce odlišnými zdroji dat. mezi relačními databázemi (MySQL, PostgreSQL, Oracle, MSSQL, ...) a dokonce i definování vlastních zdrojů dat ať už z lokálních zdrojů s přímým přístupem (formátované soubory např. csv, xml) nebo za pomoci API k on-line službám (LDAP, twitter, IMAP).

Díky množství napsaného a otestovaného kódu, není potřeba psát již jednou napsané části, ale možné šetřit lidské zdroje na vývoj. Tím není myšleno prosté skládání kusů posbíraného kódu, ale smysluplného využívání dostupných knihoven a pluginů do programovacího jazyka nebo frameworku.

Díky těmto úsporám je možné se zaměřit na ergonomii dané aplikace a její možnou optimalizaci, která je ovšem limitována výkonností použitého skriptovacího jazyka. Tato omezení lze, minimalizovat udržováním částí zpracovaného zdrojového kódu v paměti a kešováním<sup>6</sup> nebo kompresí výstupu.

Tuto nepříjemnou vlastnost skriptovacích jazyků lze řešit předkompilací zdrojových kódů, nebo vhodným využíváním „keší“.

APC, Xcache, File

---

či tříd programu či knihovny, jenž mohou být využívány programátorem

<sup>5</sup><http://framework.zend.com/>

<sup>6</sup>z anglického slova *cache*, označuje vyrovnávací paměť

Další možnou optimalizací, která sice přímo nesouvisí s CakePHP, ale je v něm snadno implementovatelná, je snížení počtu požadavků na stránku. S používáním javascriptových frameworků a knihoven se snadno může stát, že počet vkládaných odkazů na skripty a kaskádové styly (CSS) se vyšplhá až k desítkám a začne se neúměrně prodlužovat doba potřebná na stažení všech potřebných částí. Zvyšuje se tak počet požadavků na server a díky režii protokolu HTTP je ve výsledku stažen větší objem dat. Tento problém úspěšně řeší projekty *jsMin*<sup>7</sup> a *CSSTidy*<sup>8</sup>.

## MySQL

### 4.2 Klient

Pro vlastní běh aplikační logiky je možné se spolehnout, že serverová část aplikace bude zpracována jednou verzí PHP interpretu v uzavřeném a otestovaném prostředí. Naproti tomu klientská část bude prezentována na odlišných operačních systémech v mnoha prohlížečích nejrůznějších verzích. Základními požadavky kladené na prohlížeč jsou:

- XHTML 1.0<sup>9</sup>
- CSS 2.1<sup>10</sup>
- JavaScript<sup>11</sup>

S XHTML a CSS prohlížeče problémy nemívají. Horší je to, ale s implementacemi Javascriptu. Toto bylo vyřešeno díky provázanosti CakePHP a javascriptového frameworku Prototype[@todo odkaz] a jeho rozšíření Script.aculo.us[@todo odkaz], který se snaží zakrýt rozdíly mezi prohlížeči.

Prototype není jediným či nejlepším frameworkem. Existují i jiné, které mají rozsáhlejší schopnosti v oblasti používání dotazovacího jazyka XPath nebo tvorby GUI<sup>12</sup>

---

<sup>7</sup><http://code.google.com/p/jsmin-php/>

<sup>8</sup><http://csstidy.sourceforge.net/>

<sup>9</sup><http://www.w3.org/TR/xhtml1/>

<sup>10</sup><http://www.w3.org/TR/CSS2/>

<sup>11</sup>JavaScript je dialektem ECMAScriptu. Minimálním implementovaným standardem by měl být ECMA-262, revize 3:<http://www.ecma-international.org/publications/standards/Ecma-262.htm>.

<sup>12</sup>z anglických slov *Graphical User Interface*, uživatelské prostředí, jenž umožňuje uživateli ovládat aplikaci pomocí grafických ovládacích prvků



## Prototype a jeho rozšíření

Prototype umožňuje

- JS - Prototype, Script.aculo.us, Livepipe

# Kapitola 5

## Programátorská dokumentace

V následující kapitole je popsáno a vysvětleno databázové schéma, základní adresářová struktura projektu, a metody použitých tříd.

### 5.1 Databázové schéma

Normalizace???

### 5.2 Adresářová struktura

# Kapitola 6

## Uživatelská dokumentace

Tato uživatelská dokumentace si klade za cíl stručně přiblížit čtenáři výhody informačního systému a jeho základními způsoby ovládání. Jednotlivých kapitolách je vysvětleno, jak má správce postupovat od úvodní instalace, přes inicializaci databáze, uložení informací o uživateli, přidání produktů až po tisk účetních podkladů.

Dodavatelům a zaměstnancům je názorně předvedeno, jak správně a včas vyplnit měsíční výkazy a jak vést evidenci docházky účastníků kurzů.

Zákazníkům je vysvětleno, jak jednoduše zjistit, kolik z objednaných služeb již bylo zapláceno a zkontrolovat průběh aktuálních kurzů či počet přeložených stran překladu.

Studenti

### 6.1 Instalace serveru

Pro běh serverové aplikace je nutné mít sprovozněný program, který umožňuje zpracování zdrojových kódů a prezentaci výstupu protokolem HTTP popřípadě HTTPS. Nejznámějším volně dostupným programem je Apache [<http://httpd.apache.org/>], který umožňuje pomocí modulů [<http://httpd.apache.org/modules/>] přidat podporu pro jazyk PHP nutný k běhu IS.

### Stažení a instalace Apache

Pokud používáte některou z moderních linuxových distribucí, zkuste nejdříve projít repozitáře [[@todo vysvětlit](#)], zda se zde nenachází již hotový balík upravený pro snadnější instalaci a konfiguraci.

Debian a jemu podobní (Ubuntu, Kubuntu, ...):

```
apt-get install apache2
```

Gentoo

```
emerge apache2
```

Pokud jste požadovaný balík nenašli či používáte jiný operační systém, můžete ze stránek projektu<sup>1</sup> vybrat odkaz vedoucí na požadovaný balík zdrojových kódů nebo předkompilovanou aplikaci pro Váš operační systém.

## Konfigurace Apache

Pro správnou funkci aplikace je potřeba doinstalovat, popřípadě pouze povolit následující moduly: *mod\_php5*, *mod\_rewrite* a *mod\_ssl*. Pro správnou funkci zabezpečeného připojení [@todo definovat zabezpečené připojení] je nutné vygenerovat certifikáty a upravit konfiguraci stránek.

## Stažení a instalace MySQL serveru

uživatelské jméno: isadb

heslo: isapass

## 6.2 Umístění aplikace

Pokud máme server správně nakonfigurovaný, zkopírujeme složku s aplikací do adresáře určeného konfigurací Apache (obvykle /var/www či C:\\Program Files\\apache2\\www). Dále je potřeba nastavit přihlašovací údaje k databázi podle 6.1 předchozí části.

## 6.3 Inicializace databáze

V souboru %CDROM%/app/config/sql/isa\_init.sql se nachází MySQL 5.0+ kompatibilní skript, který vytvoří tabulky a naplní je daty nutnými k prvnímu přihlášení administrátora.

---

<sup>1</sup><http://httpd.apache.org/download.cgi>

## 6.4 Upřesňující informace

Pro další čtení manuálu je potřeba upřesnit několik důležitých pojmů, které se budou dále vyskytovat.

**Systémová skupina** je nutná pro správné fungování IS.

Tato sekce je rozdělena podle rolí definovaných v IS.

**Správce:** pověřený uživatel s plnými právy ke všem modulům systému.

**Editor:** osoba s omezenými právy k editaci vybraných modulů.

**Dodavatelé:** zaměstnanec, brigádník či jiný subjekt vykonávající zadanou práci.

**Odběratelé:** @todo

**Poskytovatelé:** subjekty, které jsou vedeny „pod jednou střechou“ a sdílí část informací.

**Účastníci:** studenti jednotlivých kurzu.

Výše popsané role mohou být změněny či zakázány administrátorem systému. Čtenáři je doporučeno číst pouze části, jež se ho týkají.

## 6.5 Správce (hlavní manažer)

Má standardně nastavena veškerá přístupová práva ke všem modulům systému.

### Správa IS

Tento modul obsahuje moduly pro nastavení jednotlivých částí systému, správu uživatelů, skupin, práv a ostatních číselníků (daně, místnosti, kategorie produktů).

## Uživatelé (Users)

Modul UŽIVATELÉ nabízí veškeré nastavení potřebné pro definování možností při užívání systému osobami majícími vztah k IS. Dále jsou zde uchovány veškeré osobní informace důvěrného charakteru.

**Seznam uživatelů [/admin/Users/index]:** V horní části se nachází odkazy na akce související s uživateli. Filtry pro práci se seznamem uživatelů jsou umístěny nad hlavní tabulkou a v hlavičce tabulky s funkcí „našeptávače”.

**Přidání uživatele [/admin/Users/add]:** Jedinou povinnou položkou je *zobrazované jméno*, která slouží jako popisec ve všech výběrových filtrech. Ostatní položky můžete vyplnit až při jejich potřebě ve výpisu. Volba *Aktivní* slouží k aktivaci uživatelského účtu. Uživatel je schopen se přihlásit pouze pokud je jeho účet aktivní.

**Editace uživatele:** Vyberte odpovídající skupinu k editaci a pro uložení změn použijte tlačítko *Uložit*. V případě chybového hlášení zkontrolujte všechny skupiny údajů.

**Zobrazení uživatele:** @todo

**Nastavení či změna hesla:** V přehledu uživatele klikněte na odkaz Změnit heslo a vyplňte nové heslo do obou kolonek. Správce může editovat hesla všem uživatelům a měl by je o této změně informovat zabezpečeným kanálem[@todo definovat zabezpeceny kanal], aby se předešlo zneužití jejich účtu.

## Skupiny a oprávnění

V tomto modulu můžete přidávat a mazat uživatelské skupiny a oprávnění. Pro zachování správného fungování systému nemažte tyto systémové skupiny: customers, employees, providers, students a admin.

Při implicitní nastavení má každá ze systémových skupin práva definované a přiřazené oprávnění k prefixovaným akcím jednotlivých modulů. Tyto oprávnění jsou definovány tímto způsobem: *\*:customers\_\**, *\*:employees\_\**, *\*:providers\_\**, *\*:students\_\**, *\*:admin\_\** a říkají, že daná skupina může v libovolném modulu spouštět akce začínající jejich jménem. V příkladu byl použit expanzní znak „\*”, který je možné používat k nahrazení libovolného počtu libovolných znaků, a „:”, jenž určuje hranici mezi modulem a akcí

(např. oprávnění *Users:\*\_view* umožní zobrazit všechny prefixované akce *view* v modulu UŽIVATELÉ).

Každá skupina může mít přiřazeno více definovaných oprávnění a je na správci, jak s nimi bude zacházet.

## Číselníky

V této části se seznámíme se všemi číselníky, jejich funkcemi v systému a možnostmi editace. Všechny dále uvedené moduly obsahují automaticky generovaný číselný identifikátor (dále jen *id*), který slouží pro interní potřeby systému a není možné ho změnit. Dále je potřeba upozornit na fakt, že vytvořené položky v číselníku, které již byly v systému přiřazeny nějakým záznamům, nelze vymazat. Vymazání je umožněno až když je číselník u daných záznamů změněn.

**Kategorie** obsahuje název, zkratku, jednoduchý slovní popis a zaškrtačací políčko určující, zda se produkty v této kategorii považují za veřejné a má se zobrazovat jejich rozv. na webových stránkách. Účel číselníku spočívá v rozčlenění množství produktů do skupin, podle kterých lze vytvářet tiskové sestavy (přílohy faktur, rozvrhy, měsíční přehledy nákladů, ...).

**Místa** obsahují název a adresu. Slouží k odkazu na místo v událostech.

**Štítky** obsahují název a odkaz na nadřazený štítek, pro možnost tvorby hierarchické struktury článků.

## Produkty

Modul PRODUKTY je závislý na správně rozdělených uživateli do systémových skupin. Rovněž je doporučeno předvyplnit číselník kategorií produktů.

### Vytvoření nového produktu (jednoduchá verze)

Vyberte kategorii a poskytovatele. Dále vyplňte název produktu a informaci o datu zahájení a předpokládaném či žádaném datu ukončení prací. Dále si řekneme, jak se vypočítá výsledná cena (bez DPH) pro zákazníka. Od toho se bude odvíjet další vyplňování formuláře.

Stanovte si, jak budete daný produkt nabízet a od čeho se odvíjí náklady. V případě, že se náklady odvíjí od počtu participujících osob zašrtněte možnost *Počítat účastníky*. Pokud nevíte dobře, kolik hodin bude odučeno či kolik normostran bude účtováno a náklady na ně nejsou fixní, zaškrtněte možnost *Počítat množství*. Celková cena bude spočítána takto:

- jednotková cena je vynásobena počtem účastníků pokud byla odpovídající volba zaškrtnuta
- mezisoučet je vynásoben součtem odpracovaných jednotek v událostech ve výkazech přiřazených v produktu pokud byla volba *počítat množství* zaškrtnuta
- @todo možná konverze

Teď už zbývá doplnit zbývající povinné položky a to jednotku a jednotkovou cenu.

## Výkazy

### Vytvoření a přiřazení výkazu k produktu

Vytvoření a přiřazení výkazu k produktu provádějte pouze pokud je produkt, ke kterému chcete přidat výkaz, již vytvořen.

V menu klikněte na položku *Přidat výkaz* a vyplňte následný přehledný formulář. U dodavatele se rozlišuje se i typ smlouvy uvedený v závorce. Sazba a jednotka jsou rovněž povinné položky.

Jinou možností, jak přiřadit nový výkaz, je přes modul PRODUKTY (nebo přes „rychlé hledání“), kde lze snadno vyhledat daný produkt. Zobraze si detaily nalezeného produktu a ve skupině *Přiřazené výkazy* klikněte na odkaz *Přidat výkaz*.

### Přidání události k výkazu

### Tipy na urychlení práce

Zvláštní formulář na přidání produktu spolu se zákazníkem i dodavatelem umožní rychlejší zadávání většího množství nových produktů.



## 6.6 Dodavatel

zadávání výkazu,  
evidence docházky do kurzů

# Závěr

Zavedení informačního systému je běh na dlouhou trať ... příprava dodavatelů i odběratelů na změny ve způsobu vykazování práce ...

# Literatura

[1]

[IS] Wikipedia: *Informační systém*,  
[http://cs.wikipedia.org/wiki/Informační\\_systém](http://cs.wikipedia.org/wiki/Informační_systém)