

Európsky polytechnický inštitút, Kunovice

Odbor: Elektronické počítače

Prijímacie konanie pre SŠ

(Bakalárska práca)

Autor: Peter Medo

Vedúci práce: Ing. Oldřich Kratochvíl, Dr.h.c.

Kunovice Apríl 2007



Evropský polytechnický institut, s.r.o.

Osvobození 699, 686 04 Kunovice
☎ a fax: 572549018, 548035, e-mail: epi@vos.cz
<http://www.vos.cz/epi>

Student(ka)
Peter Medo
Jakubská 8
831 06 Bratislava

VÁŠ DOPIS ZNAČKY / ZE DNE

NAŠE ZNAČKA
BP_EP05/06

ZODP.VEDOUCÍ/VYŘIZUJE
Ing. Dušek/Čápková

KUNOVICE
1.6.2006

Zadání bakalářské práce

Vážený studente, vážená studentko,

jako téma Vaší bakalářské práce ve studiu oboru Elektronické počítače Vám zadávám

Přijímací řízení pro SŠ

- Osnova:
1. Definice problému, který má být vyřešený
 2. Analýza projektu přijímacího projektu
 3. Automatické přeposílání emailu zaregistrování přihlášky
 4. Zařazení uchazečů do tříd a skupin
 5. Nástroj pro tvrdé opravy údajů
 6. Tvorba tiskových sestav

Bakalářská práce bude zpracována pro: Evropský polytechnický institut, s.r.o.

Tento dokument je součástí Vaší bakalářské práce.

S pozdravem

Ing. Oldřich Kratochvíl / Dr.h.c.
rektor

Prehlasujem, že som bakalársku prácu vypracoval samostatne pod vedením

Ing. Oldřicha Kratochvíla, Dr.h.c. a uviedol v zozname literatúry všetky použité literárne
a odborné zdroje.

Bratislava Apríl 2007

Ďakujem pánovi Ing. Oldřichovi Kratochvílovi, Dr.h.c. za veľmi prospešnú teoretickú pomoc, ktorú mi poskytol pri spracovaní mojej bakalárskej práce.

Bratislava, Apríl 2007

Peter Medo

Obsah

Obsah	5
Úvod	6
1 Teória programových systémov	7
1.1 Databázy.....	7
1.2 MySQL	8
1.3 Úvod do PHP.....	9
1.3.1 PHP a formuláre	11
1.4 Apache	12
1.5 Ajax.....	13
1.6 HTML	15
1.7 CSS – kaskádové štýly.....	16
1.7.1 Dedičnosť	17
1.7.2 Definovanie štýlu pomocou CSS.....	17
2 Analýza problému, ktorý má byť vyriešený	18
2.1 Analýza požiadaviek na aplikáciu	18
2.2 Riešenie analýzy aplikácie	19
2.3 Analýza softvérového vybavenia servera	21
3 Štruktúra riešenia aplikácie	22
3.1 Štruktúra databáze	22
3.1.1 Tabuľky s údajmi študentov.....	22
3.1.2 Tabuľky s údajmi číselníkov	25
3.1.3 Tabuľky administrovania systému.....	25
3.2 Prihláška a formulár.....	26
3.2.1 Prihláška (formulár) na internete	27
3.2.2 Druhá strana prihlášky	30
3.3 Portál SSOŠ.....	31
3.3.1 Prihlásenie na portál.....	31
3.3.2 Nový študent.....	33
3.3.3 Browse - prezeranie	34
3.3.4 Hľadať	35
3.3.5 Export.....	36
3.3.6 Administrácia	37
3.3.7 Help - pomoc	39
3.3.8 Bezpečnosť intranetu	39
3.4 Implementácia systému.....	40
Záver	41
Resumé	42
Použitá literatúra	43
Knižné publikácie:.....	43
Internetové zdroje:.....	43
Prílohy	43

Úvod

V dnešnom svete je veľmi dôležitá komunikácia. Najprogressívnejším masovým prostriedkom na komunikáciu sa v posledných rokoch stal internet. Internet vznikol v roku 1969 ako sieť Ministerstva obrany USA ARPANET. Do úzadia sa dostala tradičná pošta a miesto posielania listov sa posielajú cez internet emaily. Ďalšou výhodou internetu je vyplňovanie formulárov. Formulár si stiahne klient internetu a pošle mailom alebo ho môže priamo vyplniť vo webovom prehliadači na internete. Výhodou internetu je úspora času, zjednodušenie procesov, flexibilita a dostupnosť všetkým užívateľom.

Cieľom bakalárskej práce je navrhnúť systém, ktorý bude slúžiť na on-line prihlasovanie študentov, ktorý sa hlásia na strednú školu na Európskom polytechnickom inštitúte, s.r.o. Aplikácia má uľahčiť evidenciu nových študentov cez internet a ďalšiu editáciu pri postupe prijímacieho konania na strednú školu. Prijímacie konanie v systéme začína vyplnením formulára. Neskôr prijímacie konanie pokračuje spracovaním údajov a ich editáciou v prípade zlého spracovania. Editovaním myslím vytvorenie alebo zmenu údajov v databáze. Prijímacie konanie je navrhnuté v PHP5, HTML a MySQL databáza.

Projekt je rozdelený na tri kapitoly, ktoré sú rozvetvené na podkapitoly. V úvodnej kapitole je opísaná teória systémov a programov, ktoré sú použité v projekte. Druhá kapitola sa zaoberá analýze projektu a v tretej je vypracovanie projektu. Hlavným systémom projektu boli databázové systémy. Systém relačnej databázy MySQL v projekte zaisťuje uchovanie a zatriedenie údajov s formulára, ktorý študent vyplní cez internet. Spracované výstupy s formulára sú ďalej softvérovo spracované v databáze MySQL a dajú sa administratívne meniť. Pod pojmom administrácia sa rozumie zásah oprávnených pracovníkov strednej školy do databázy informácií, ktoré zadali uchádzači na školu do formulára na internete. V bakalárskej práci som použil programovacie jazyky: HTML, PHP5, CSS-kaskádové štýly, JavaScript a Ajax technológiu. Na vypracovanie bakalárskej práce bol použitý balík programov Xampp, ktorý obsahuje relačnú databázu MySQL, PHP5 a Apache webserver. Na editovanie zdrojového kódu programu som použil PSPad editor. Prijímacie konanie pre SŠ je spracované tak, aby požiadavky vyplývajúce z práce na študijnom oddelení boli uspokojivé a odbremenili pracovníkov od zdlhavej administratívnej práce.

1 Teória programových systémov

V tejto kapitole sa čitateľ oboznámi s teóriou programovacích systémov, ktoré sú použité v mojej bakalárskej práci. V celej teórii programových systémov je najdôležitejšie softvérové riešenie. Softvérové riešenie bakalárskej práce spracúva a zobrazuje informácie, ako sú dáta, obrázky alebo odkazy cez webový prehliadač. Ako softvérový voľne šíriteľný balík som použil Xampp. Je to webserver balík, ktorý dovoľuje lokálne vytvárať dynamické webové aplikácie. V bakalárskej práci na databázové riešenie projektu som použil relačnú databázu MySQL. Jadrom programových webových jazykov v bakalárskej práci je kódový programovací jazyk PHP5. Ďalšími jazykmi a technológiami, ktoré sú použité v projekte sú HTML (hyper text markup language), CSS (Cascading Style Sheets) a Ajax technológia (Asynchronous JavaScript and XML). Na editovanie textu softvéru som použil editor PSPaD. Editor PSPaD má výhodu oproti editovaniu textu v poznámkovom bloku zvýraznením polí v použitom zdrojovom kóde.

1.1 Databázy

V súčasnom svete sme svedkami, že internet a databázové informačné systémy sú spojené do jedného uceleného systému. Internetový prehliadač slúži, ako strana medzi klientom internetu a serverom internetu. Internetový server spolu s databázovou vrstvou slúžia ako zdroj, výmena dát a informácií prúdiacich do a z vnútra škôl, podnikov a firiem alebo z iných inštitúcií. Databázová vrstva je základom každej webovej databázovej aplikácie. V každom systéme treba vybrať vhodný databázový systém. Ja som si vybral MySQL relačný databázový systém. Tento systém podporuje široko používaný voľne šíriteľný (Open Source) viacúčelový skriptovací jazyk PHP5. PHP5 jazyk je vhodný hlavne pre vývoj webu a môže byť súčasťou HTML kódu. Pre tieto výhody som aplikáciu naprogramoval v skriptovacom jazyku PHP5.

V súčasnosti PHP5 podporuje okrem databázy MySQL aj nasledujúce rozhrania prostredníctvom natívneho protokolu:

Adabas D	Ingres	Oracle (OCI7 a OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (iba na čítanie)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Tabuľka č.1: Zoznam databáz podporujúci PHP5

PHP má rozšírenie abstrakcie DBX databázy, ktorá nám umožňuje transparentne použiť akúkoľvek databázu podporovanú týmto rozšírením. Dodatočne PHP podporuje ODBC, štandard Open Database Connection, takže sa môžete pripojiť na akúkoľvek inú databázu podporujúcu tento svetový štandard. Výber databázy je nevyhnutná časť mojej aplikácie. Relačná databáza MySQL je v tomto prípade na mieste. Štruktúra databázy je tvorená pomocou tabuliek. Tabuľky sú usporiadané podľa informácií, ktoré sú uložené v databáze. Databázu v aplikácii je možné rozdeliť do troch skupín tabuliek. Skupina tabuliek je rozdelená podľa typu údajov, ktoré sú tam uložené. Sú to: tabuľky z údajmi študentov, tabuľky z údajmi z číselníkov a tabuľky potrebné pre administráciu systému.

1.2 MySQL

V tejto podkapitole som sa venoval teórii relačnej databázy MySQL, ktorú som použil v mojej bakalárskej práci. MySQL je otvoreným zdrojom. To znamená, že zdrojový kód programu je voľne dostupný a je poskytovaný zdarma. Úlohou relačnej databázy MySQL je ukladanie, vyhľadávanie a správa dát. MySQL databáza je systém, ktorý je možné spojiť s PHP skriptovacím jazykom a Apache technológiou. Tieto technológie sprístupňujú a zobrazujú dáta v internetovom prehliadači v čitateľnej forme. Pomocou servera, ktorý spracúva dotazy v štruktúrovanom dopytovanom jazyku (Structured Query Language –

SQL). Server je navrhnutý na spracúvanie veľmi zložitých dotazov. Databáza MySQL je uložená na serveri a je obľúbená najmä vo webových aplikáciách. Je veľmi rýchla, má dobrú podporu štandardov jazyka SQL. Nevýhodou má v základnom móde, kde jej chýbajú niektoré dôležité funkcie, ako napr.: transakcie, vnorené výbery (selekt), uložené procedúry a pohľady (view). Preto je databáza nevhodná pre použitie v komplikovaných aplikáciách, ale nám postačí. MySQL je systém správy relačných databáz (RDBMS, relational database management system – relačný systém riadený bázou dát). Umožňuje teda ukladať samostatné informácie do tabuliek do oblastí súvisiacich informácií. Vďaka tomu ponúka maximálnu efektivitu a rýchlosť. V nerelačných databázových systémoch sú všetky informácie v jednej oblasti. Nevýhodou sú potom zložitejšie operácie s informáciami v tejto nerelačnej databáze. Je to najmä pri triedení, ale aj pri bežnom vyčleňovaní z databázy. V MySQL sa každá tabuľka skladá zo samostatných stĺpcov a polí, ktoré zastupujú jednotlivé časti uložených informácií. V stĺpci sa potom nachádzajú jednotlivé informácie ako sú meno študenta, priezvisko študenta alebo rok narodenia. V poli sa nachádzajú jednotlivé typy informácií napríklad text, dátum, čísla. Jednotlivé tabuľky môžu byť prepojované podľa spoločného identifikačného kľúča.

Medzi základné funkcie MySQL patria :

- možnosť využitia väčšieho množstva procesov prostredníctvom vlákna jadra
- možnosť nezávislého spúšťania na hostiteľskej platforme
- veľa druhov stĺpcov pokrývajúce všetky dátové typy
- skripta funkcií pre matematické výpočty a triedenie
- príkazy, ktoré umožňujú prístup správcov k informáciám v databáze
- systém overovania užívateľov na základe hesla
- možnosť definície 32 indexov pre každú tabuľku; táto funkcia bola preskúšaná na 60 000 tabuliek a 5 000 000 000 riadkoch
- medzinárodné hlásenie chýb použité vo veľa rôznych štátoch

1.3 Úvod do PHP

Na začiatku by bolo vhodné najprv sa stručne zoznámiť s PHP históriou, ak sa to tak vôbec dá povedať. PHP sa zrodilo niekedy v roku 1994 a vzniklo ako program v Perle, ktorý slúžil na evidenciu prístupov k webovým stránkam pána Rasmusa Lerdorfa, ktorý bol jeho prapôvodcom. Nakoľko však Perl ako interpretovaný jazyk priveľmi zaťažoval webový server, autor svoj softvér prepísal do jazyka C. Postupne ho používalo čoraz viac ľudí, stával sa viac a viac obľúbenejším a autor bol doslova donútený vydať k svojmu dielu

dokumentáciu. Systém bol neskôr rozšírený o požiadavky jeho používateľov a bol uvoľnený pod názvom Personal Home Page Tools. Po zabudovaní SQL modulu, ktorý umožňoval PHP spolupracovať s databázami a vytvárať formuláre sa názov zmenil na Personal Home Page Form Interpreter PHP/FI. Tento systém vznikol ako kompilát prvotného PHP a programu, ktorý sprístupňoval výsledky databázových dopytov na webe (Form Interpreter). Tak vznikol PHP/FI 2.0. V roku 1998 bola vydaná verzia PHP 3.0. Oproti svojim predchodcom sa celý systém zrýchlil a rozšíril o veľa rôznych funkcií. V roku 1999 bola prístupná verzia PHP 4.0 vyvíjaná pod kódovým názvom ZEND. Táto verzia je až 8-10 krát rýchlejšia ako PHP 3.0 a disponuje rôznymi chuťovkami v štýle jazyka C, resp. C++. PHP verzia 4 dosiahla skutočne masové rozšírenie. V lete 2004 po roku beta testovania bola uvoľnená prvá finálna verzia PHP5. Nová verzia jazyka prináša veľmi podstatné zmeny, avšak bez väčších dopadov na spätnú kompatibilitu. Predovšetkým bol kompletne prepracovaný objektový model jazyka, a triedy. V PHP5 jazyku objekty už nie sú iba výsledkom, ale integrálnou súčasťou jazyka, ponúkajúca skoro rovnaké možnosti porovnateľné s objektovo orientovanými jazykmi (režimy prístupu, konštruktory a deštruktory, volanie odkazom, rozhranie atd.). Medzi ďalšie dôležité zmeny určite patrí zavedenie mechanizmu výnimiek.

Platforma PHP je dnes jedna z najrozšírenejších technológií generovania dynamických webových stránok na strane servera. PHP (Personal Home Page) je široko používaný voľne šíriteľný (Open Source viacúčelový skriptovací jazyk, ktorý je vhodný hlavne pre vývoj webových aplikácií a môže byť vnorený do HTML.

Príklad PHP vnoreného do HTML:

```
<html>
<head>
  <title>Príklad</title>
</head>
<body>
  <?php
  echo "Ahoj, ja som PHP skript!";
  ?>
</body>
</html>
```

Tento script vypíše vo webovom prehliadači: "Ahoj, ja som PHP skript!". Aplikácia prijímacieho konania bola navrhnutá v PHP5 jazyku. Rád by som niečo o PHP napísal, než predstavím zaujímavosti tohto jazyka. PHP je hypertextový preprocesor, ktorý na servery interpretuje stránky HTML s vlastnými príkazmi pred ich odoslaním ku klientovi (obvykle je ním webový prehliadač). To znamená, že PHP umožňuje vkladať vlastné skripty (krátke úseky kódu, ale i celé programy) priamo do hypertextových stránok. To nie je nič

neobvyklého. Do hypertextových stránok je možné vkladať tiež napríklad jazyk JavaScript a CSS - kaskádové štýly. Existuje tu však niekoľko podstatných rozdielov. V prvom rade je PHP interpretované na servery, zatiaľ čo JavaScript je jazyk interpretovaný len klientom. Oba spôsoby sa v niektorých ohľadoch veľmi podstatne líšia. V mojej aplikácii som použil všetky uvedené programovacie jazyky.

1.3.1 PHP a formuláre

Jazyk HTML obsahuje podporu formulárov, ktoré slúžia pre načítavanie dát od používateľa. Formulár vyzerá ako dialógový box v okne prehliadača, môže obsahovať bunky, editovacie boxy, listboxy, zaškrťavacie a prepínacie bunky a iné komponenty. Script získané dáta z formulára spracuje a výsledok odošle na okno prehliadača.

V jazyku HTML slúži pre vloženie formulára do stránky tag **FORM**. Má dva atribúty **ACTION** a **METHOD**. Atribút **ACTION** určí URL adresu skriptu, ktorý slúži pre spracovanie dát získaných z daného formulára. Atribút **METHOD** určí, akým spôsobom budú dáta z formulára predané serveru, buď metódou **GET**, ktorá pripojí prenášané dáta za URL obslužného skriptu alebo metódou **POST**, ktorá zakomponuje prenášané dáta do tela HTTP.

Základ formulára by mal dodržiavať nasledujúcu syntax:

```
<FORM ACTION="URL skriptu" METHOD="GET|POST">
```

Telo formulára:

```
<form method="post" action="aplicat_2.php5" >  
telo formulára  
</form>
```

Skript použitý v mojej aplikácii v súbore **aplicat_index.php5**.

Vo formulároch je možné používať rôzne elementy, napríklad: TEXTAREA, INPUT, SELECT, LABEL, FIELDSET, BUTTON.

Príklad použitia elementu INPUT:

```
<FORM ACTION = "tut_5_pr1_spracuj.php" METHOD=POST>  
<TABLE FRAME=BOX RULES=NONE BORDER=1 CELLPADDING=4>  
<TR><TD> Meno: <TD><INPUT TYPE=TEXT NAME=Meno SIZE=30>  
<TR><TD> Priezvisko: <TD><INPUT TYPE=TEXT NAME=Priezvisko SIZE=30>  
<TR><TD> E-mail: <TD><INPUT TYPE=TEXT NAME=Email SIZE=20>  
</TABLE>  
<B> Súhlasím s odoslaním tohto dotazníka: </B><BR><BR>  
<INPUT TYPE=SUBMIT NAME="Potvrdenie" VALUE="Ano">  
<INPUT TYPE=SUBMIT NAME="Potvrdenie" VALUE="Nie"><BR>  
</FORM>
```

Súbor pracujúci v .PHP môže vyzeráť nasledovne:

```

<HTML><HEAD><TITLE> FORMULAR 1 </TITLE></HEAD>
<BODY>
<?
  echo $Meno;
  if ($Potvrdenie=="Ano")
    echo "Ďakujem za údaje";
  else
    echo 'No nič...';
?>
</BODY></HTML>

```

Po kliknutí na ANO sa objaví vo výstupe našej aplikácie: Ďakujem za údaje.

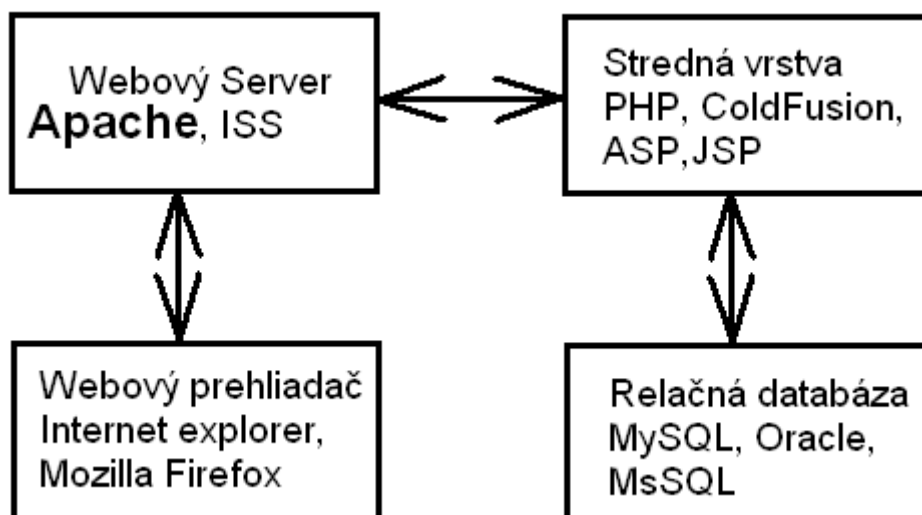
V aplikácii je použitá metóda POST, nakoľko je to bezpečnejšie. V obslužnom skripte máme k dispozícii premenné s rovnakým názvom ako obsah atribútu NAME v zdrojovom formulári. Hodnota premennej odpovedá textu uvedenému v atribúte VALUE. V poliach \$HTTP_GET_VARS a \$HTTP_POST_VARS máme k dispozícii všetky premenné odoslané metódami POST alebo GET, pričom nemusíme poznať mená vstupných polí.

Prvou časťou aplikácie je formulár, ktorý tvorí vstupné rozhranie internetovej a webovej aplikácie. Podľa jeho vzhľadu a formy potom užívateľ hodnotí často celý server, prípadne autora. Je jednoduché vytvoriť neprehľadný a neprívetivý formulár, ktorý je nutné po nesprávnom či nedostatočnom vyplnení kompletne vyplňať znova, čo užívateľa privádza často do exitovaného stavu. Tu už je veľmi blízko k tomu, že užívateľ opustí vaše webové stránky. Preto je nutné dodržiavať určité pravidlá, ktoré umožnia vytvárať prehľadné formuláre.

Snažil som sa používať kaskádové štýly, združovať rovnaké informácie do blokov, zreteľne označovať povinné a nepovinné vstupné polia. Vhodné je, aby kontrola formulára prebehla ešte na strane klienta predtým, ako dôjde k prenosu informácií (paketov). Na to sú vhodné skripty orientované na strane klienta (JavaScript, VB Script), niektoré prehliadače však s nimi majú problémy. Pokiaľ užívateľ na niečo zabudne, formulár požiada len o doplnenie chýbajúcej informácie, nebude nútiť užívateľa vyplňovať všetko znova.

1.4 Apache

Apache sa správa ako webový server. Spracúva PHP script na serveri. Hlavnou jeho úlohou je spravovať požiadavky, ktoré vyslali užívatelia s webových prehliadačov. Apache ďalej zobrazuje výsledky pomocou kódu vo vyžiadanom súbore. Na obrázku č.1 je Apache ako súčasť webového servera. Apache spolupracuje v sústave programov s ostatnými časťami informačného systému ako sú napríklad PHP jazyk alebo Mozilla Firefox webový prehliadač.



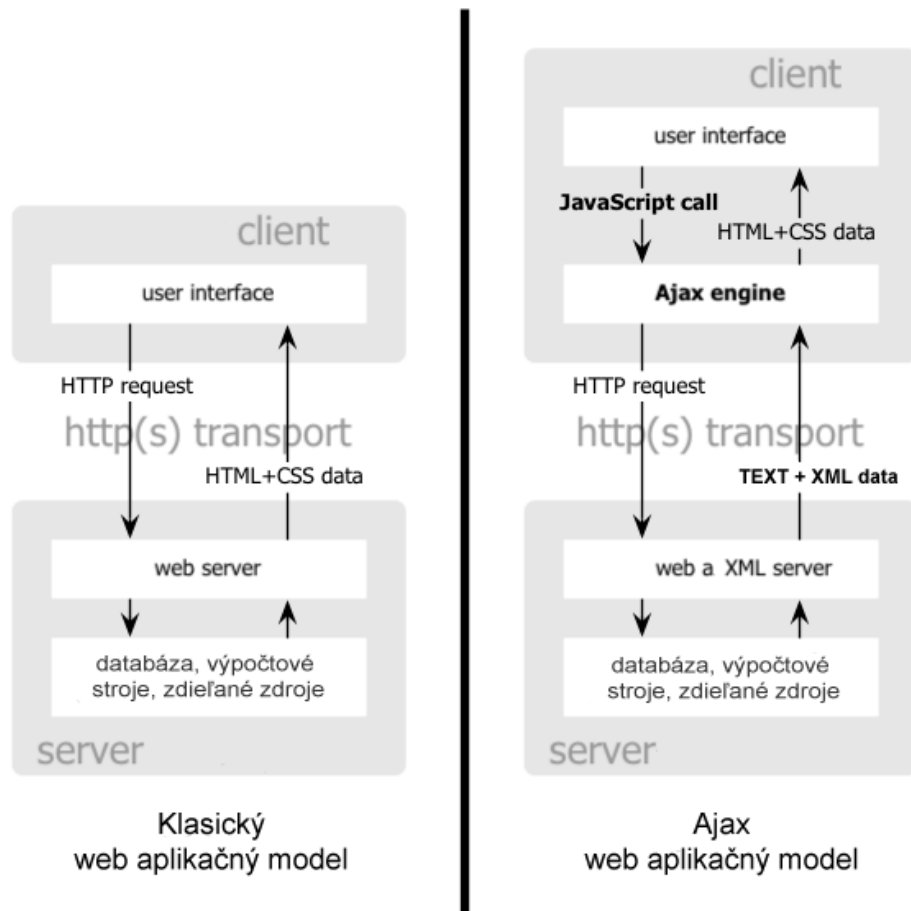
Obrázok č.1: Apache v sústave aplikácií informačného systému

1.5 Ajax

Programový názov Ajax vznikol ako skratka z výrazu „Asynchronous JavaScript and XML“. Samotný Ajax nie je technológia v pravom slova zmysle, skôr ide o technológiu tvorby webových aplikácií s využitím kombinácie (X)HTML, CSS, JavaScript, DOM, XML.

Ajax je to technológia dynamických webových aplikácií, ktorá pracuje asynchrónne. Tradične sa spracúva celá webová stránka v prehliadači. Ajax toto nerobí, pretože umožňuje možnosť komunikovať zo serverom asynchrónne. Oslobodzuje tým užívateľa od cyklu požiadavka a následná odpoveď. Užívateľ na začiatku určí, ktorú informáciu chce editovať, potom čaká na načítanie editovaného formulára v novom okne, vyplní vstupné polia a stlačí tlačidlo potvrdiť. V interaktívnej stránke si používateľ zvolí časť, ktorú chce editovať, automaticky sa mu otvorí editovací formulár v pôvodnej stránke a po potvrdení sa vykonané zmeny asynchrónne pošlú na spracovanie. Výsledkom je zobrazenie spravovanej informácie v pôvodnom okne. Ajax je možné použiť pri navzájom prepojených poliach. Príkladom je výber mesta bydliska. Podľa toho aký si vyberiem kraj podľa toho sa asynchrónne načítajú mestá k vybranému kraju. V Ajaxe je možné využiť automatické ukladanie pri nejakom editovanom prvku. Toto je možné využiť pri prerušení spojenia. Vyplnený formulár vo webovom prehliadači sa tak nestratí a je vybraný z pamäte. Ajax je využívaný pri on-line komunikácii medzi dvoma alebo viacerými používateľmi. Pri každom novom príspevku užívateľa sa mení len malá časť zobrazenej informácie. On-line komunikácia založená na statickom webe, sa pri každom novom príspevku musí načítať celá predošlá komunikácia. Vyžaduje to potom väčšiu veľkosť prenesených dát, ako použitie asynchrónneho načítania

dát. Na obrázku č.2 sú zobrazené tradičné modely aplikácií v porovnaní s Ajaxovým modelom aplikácie.



Obrázok č.2: Tradičný model aplikácií (vľavo) v porovnaní s Ajaxovým modelom (vpravo)

1.6 HTML

V skratke HTML - HyperText Markup Language v preklade znamená nad text značkového jazyka.

Prvú definíciu HTML 0.9 vytvoril Tim Berners-Lee v roku 1991. Prvá verzia umožňovala vkladať do textu obrázky, hypertextové odkazy, vytvoriť niekoľko logických úrovní a niekoľko druhov zvýraznenia. HTML 1.0 táto verzia neexistuje. Dôvodom sú neformálne štandardy, ktoré sa na začiatku zrodu HTML používali. Prvá oficiálna verzia bola označená číslom 2.0, ale schválená bola až 22. septembra 1995 ako štandard (špecifikovaná v RFC 1866). Neskôr HTML 3.0 štandard vytvorilo novozaložené W3C v marci 1995. Obsahovalo veľa nových možností ako podpora tabuliek, obtekanie textu, a zobrazovanie matematických výrazov. V tom čase ho nepodporovali webové prehliadače. Ďalšia navrhovaná verzia, HTML 3.2, väčšinu nových prvkov vynesla a namiesto toho prijala prvky, ktoré medzičasom vytvorili webové prehliadače Netscape a Mosaic. Plánovaná podpora matematiky z verzie HTML 3.0 sa neskôr stala samostatným štandardom MathML. Ďalšou a poslednou verziou je HTML 4.0. Prijala mnohé prvky a vlastnosti z jednotlivých webových prehliadačov, ale zároveň sa pokúšala vyčistiť štandard takým spôsobom, že niektoré prvky označila za zastarané. Ďalšie verzie jazyka HTML už nikto nevytvorí, lebo pokračovateľom tohto jazyka je XHTML, založený na XML. XHTML je rozšíriteľný hypertextový značkový jazyk (Extensible Hypertext Markup Language). Dokument HTML má začínať definíciou typu a potom sa skladá s hlavičky <head> a tela<body>.

Príklad definície typu dokumentu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Na vytváranie HTML a XHTML kódu používame programy na editovanie kódu. HTML a XHTML sú obyčajné textové dokumenty. Obsahujú vlastný text a značky, ktoré textu priradujú určitý význam. K písaniu webových stránok netreba žiadny špeciálny softvér. K napísaniu webovej stránky nám postačí editor poznámkový blok (Notepad). Na jednoduchšie písanie kódu používame rôzne editory. Rozdeľujeme ich na:

- štruktúrované editory
- WYSIWYG editory

V štruktúrovaných editoroch sa pracuje priamo so zdrojovým kódom dokumentu. Priebežný vzhľad je možné kedykoľvek skontrolovať prepnutím do webového prehliadača.

Príkladom editorov kódu sú napr. Doom, HTML-Kit, Vim, Emacs, 1stPage, Golden HTML Editor, HomeSite a PSPaD. Pre užívateľa Linuxu je určený napríklad Quanta editor. Tieto editory nám uľahčujú prácu pri vizuálnom odlíšení vlastného textu od HTML kódu farebným rozlíšením. Je umožnené jednoduchou formou zadávať jednotlivé tagy a veľa iných funkcií zlepšujúcich prácu. Pri písaní zdrojového kódu aplikácie pre strednú školu som použil PSPad štruktúrovaný editor.

Vo WYSING editore už nepracujeme s kódom, ale so vzhľadom stránky. Práca s nimi je pohodlnejšia, ale má to svoju nevýhodu. Výsledný HTML kód býva neusporiadaný a príliš dlhý.

1.7 CSS – kaskádové štýly

CSS - kaskádové štýly znamenajú - Cascading Style Sheets. Vznikli ako súhrn metód pre úpravu vzhľadu stránok. Prvý návrh normy bol zverejnený v roku 1994. Neskôr v roku 1994 bola vydaná špecifikácia CSS 1, v roku 1998 CSS 2 a teraz sa pracuje na verzii CSS 3.

Kaskádové štýly používame na formátovanie obsahu HTML, XHTML a XML dokumentov. CSS oproti HTML formátovaniu rozširuje svoje schopnosti. V HTML sa definuje ako má určitý element vyzerať jednotlivo. V CSS sa definujú štýly, ktoré hovoria o vzhľade elementov pre celý dokument. Štýly umožňujú určiť ako bude jednotlivý element vyzerať. Napríklad všetky nadpisy - elementy úrovne 3 budú červené. Podobne pomocou štýlov môžeme určiť odlišné formátovanie pre napríklad jediný výskyt určitého elementu. Tým sa zbavíme veľkého množstva kódu a tým pádom sa stane kód oveľa prehľadnejší. Je to možné využiť pri zmene farby, druhu alebo veľkosti písma všetkých odsekov dokumentu. Je to oveľa rýchlejší spôsob ako meniť jednotlivo každý atribút v každom elemente HTML kódu. Jeden štýl môžeme použiť ľahko pre ľubovoľné množstvo stránok.

Kód sa skladá z pravidiel pre jednotlivé elementy, ktoré sú formátované. Každé takéto pravidlo má dve časti, selektor (názov elementu, pre ktorý má toto pravidlo platiť) a deklaráciu (čo pre element má platiť). V deklarácii určujeme vlastnosť a jej hodnotu, deklarácia je uzavretá do zložených zátvoriek. Celý zápis vyzerá takto:

```
Selektor {vlastnosť: hodnota_vlastnosti}  
h1 {color: green}
```

Selektorom, teda elementom, ktorý formátujeme je tu h1 (nadpis 1. úrovne). Deklaráciou je {color: green} a táto určuje, že vlastnosť color (farba) bude mať hodnotu green (zelená). V celku to znamená, že, všetky nadpisy 1. úrovne v dokumente budú mať zelenú farbu.

1.7.1 Dedičnosť

Väčšina vlastností sa dedí. To znamená, že element, ktorý nemá vlastnosť definovanú tak ju dedí po nadradenom elemente. Týka sa to hlavne vlastností písma – farby, veľkosti, štýlu atď.

Ak chceme definovať nejakú vlastnosť, ktorú budú mať všetky elementy spoločné a neskôr vytvárať výnimky, definujeme ju pre element body (telo). Pre ukladanie poznámok do zdrojového kódu sú určené komentáre. Tvoria sa pomocou /* a */. Medzi hviezdikami je umiestnený komentár. Komentáre nie sú zobrazované na výslednom zobrazení webovej stránky.

1.7.2 Definovanie štýlu pomocou CSS

V CSS môžeme štýly definovať externe alebo vo vnútri dokumentu. V externom súbore môžeme mať zadefinované rôzne štýly. Tieto štýly je potom výhodné využívať pre rôzne dokumenty. Výhoda spočíva, že nemusíme v každom dokumente definovať kompletný štýl. Stačí len keď zadefinujeme externý dokument, v ktorom sa nachádzajú vyžadované štýly. Definíciu píšeme do hlavičky medzi tagy <head> a </head> v tagu link alebo style.

```
<link rel="stylesheet" type="text/css" href="styl.css" />  
<style type="text/css">@import "styl.css";</style>
```

Pokiaľ sa štýl nachádza na inom web serveri, tak sa používa tento zápis:

```
<style type="text/css">@import url(http://www.nazovstranky.sk/styl.css);</style>
```

Definovanie štýlu vo vnútri dokumentu robíme nasledovne:

```
<style type="text/css">body {color: blue}</style>
```

2 Analýza problému, ktorý má byť vyriešený

Súčasťou analýzy bolo spracovanie požiadaviek zo študijného oddelenia na vytvorenie aplikácie na prijímanie žiakov na strednú školu. Podľa požiadaviek študijného oddelenia prebehla analýza, aký softvérový systém očakávajú a akú aplikáciu prijímania žiakov na strednú školu treba vytvoriť.

2.1 Analýza požiadaviek na aplikáciu

Požiadavky na systém som spracoval od administratívnych pracovníkov na osobných stretnutiach. Na stretnutiach sa zúčastňovali nielen, osoby ktorých sa týkal problém prijímania žiakov na stredné školy ale aj osoby s podobným projektom prijímania žiakov na odbornú školu. Na osobných stretnutiach sme analyzovali a diskutovali ako by mala aplikácia fungovať a čo všetko by sa dalo vytvoriť a čo má všetko obsahovať.

Administratívny pracovníci mali takéto požiadavky na systém:

- formulár musí obsahovať všetky informácie z papierovej prihlášky
- vo formulári budú použité číselníky ministerstva školstva
- formulár musí automaticky kontrolovať niektoré v ňom zadané informácie
- systém má umožňovať tvrdé opravy v informáciách o študentoch
- systém má umožňovať vytvárať nových študentov
- systém má umožňovať exportovanie informácií o študentoch
- systém má generovať automaticky maily pri zmenách stavov študenta
- systém má generovať automaticky tlačové výstupy

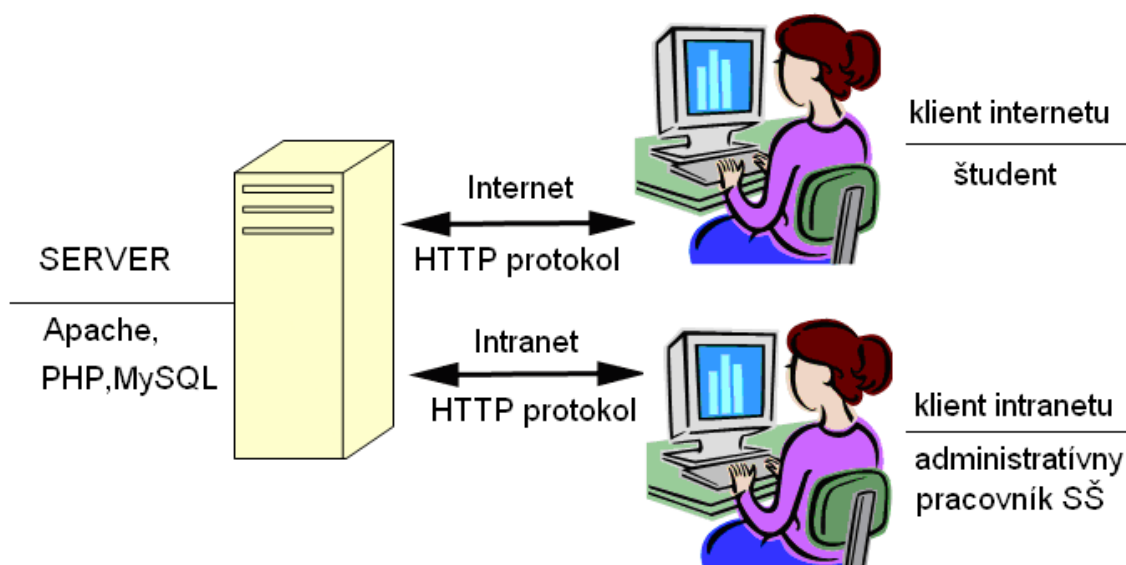
Porovnávali sme ju s existujúcou aplikáciou, ktorá sa používa na EPI. Podobnú aplikáciu prijímania žiakov vytvoril Jaroslav Lanči v roku 2004. Jaroslavov Lančiho projekt je písaný priamočiaro a to komplikuje výraznejšie zásahy v jednoduchej forme do aplikácie. Výhodou aplikácie je úspešnosť a jej doterajšieho používanie na prijímanie žiakov.

2.2 Riešenie analýzy aplikácie

Problém bol v administrácii prijímacieho konania pre strednú školu. V škole vznikol časový a administratívny problém s prijímaním nových študentov. Na tento cieľ bol vybraný projekt, ktorý toto zjednodušuje a sprehl'adňuje. Úlohou bolo spraviť aplikáciu, ktorá uľahčí administráciu prijímacieho konania na strednej škole. Riešením analýzy bolo rozdelenie aplikácie na dve základné časti:

- internetová časť
- intranetová časť

Internetová časť zaisťuje študentom on-line administratívne vyplnenie formulára na strednú školu a následné spracovanie ich osobných údajov do databázy. Formulár slúži ako prihláška na strednú školu. Súčasťou formulára sú číselníky, ktoré zjednodušujú jeho vyplňanie. Číselníky vyberajú vopred zadané informácie uložené v databáze. V číselníkoch je zoznam štátov, krajov, obcí, národností, školských predmetov, ktoré vydáva štatistický úrad. Aplikácia zadané údaje študentov z formulára spracúva do databázy. Pri nekorektnom vyplnení je vyplňajúca osoba (klient, študent) upozornený na nesprávne zadané údaje vo formulári. Po vyplnení a odoslaní formulára je možné informácie o študentovi administrovať a robiť v nich zmeny. Zmeny a rôzne iné funkcie umožňuje aplikácia v druhej časti. Aplikácia v druhej administratívnej časti umožňuje okrem zmeny údajov o študentoch aj pridanie a úpravu stavov, pridanie textov k jednotlivým stavom, vyhľadávanie študentov, štatistiku, vytváranie textov, ktoré slúžia na tlač oznámení a posielaní mailov.



Obrázok č.3: Model rozdelenia aplikácie na dve časti prístupu

Na obrázku č.3. je server, ktorý spracúva informácie od klientov internetu (študentov) a klientov intranetu (administratívny pracovník SŠ).

Klient internetu (študent) zadáva do formulára (prihlášky) tie isté informácie ako do papierovej prihlášky. Týmto sa docieli uľahčenie ukladaniu informácií do databázy administratívnymi pracovníkmi v elektronickej forme. Sú to nasledovné informácie:

- typ prihlášky a odbor na aký sa študent hlási
- osobné informácie
- informácie o narodení
- informácie o aktuálnom bydlisku
- informácie o rodičoch
- informácie o predchádzajúcej základnej škole
- informácie o prospechu na základnej škole
- žiadosť o ubytovanie

Klient intranetu (administratívny pracovník SŠ) má možnosť prehliadať a spracúvať rozsiahle informácie, ktoré mu poskytuje druhá časť aplikácie :

- vytváranie a zmenu informácií zadaných v prihlasovacom formulári
- prehliadanie študentov
- vyhľadávanie študentov
- štatistika
- administrácia textov, odborov, typov štúdia, zostáv a stavov študenta

- pomôcka pri správe aplikácie

Stavmi študenta sa myslí, akými procesmi prechádza študent pri prijímacom konaní na strednú školu.

Študent prechádza týmito stavmi prijímacieho konania:

- prihlásenie cez internet
- odoslanie papierovej prihlášky
- pozvánka na prijímacie konanie
- účasť na prijímacom konaní
- výsledky prijímacieho konania
- rozhodnutie o prijatí
- zadelenie do tried
- nastúpenie na štúdiu
- nenastúpenie na štúdium

2.3 Analýza softvérového vybavenia servera

Kapitola analýza softvérového vybavenia servera, je venovaná opísaniu programov potrebných k funkčnosti aplikácie. Na obrázku č.7. je znázornený jeden server pod ktorým musí fungovať množstvo programov, aby aplikácia prijímacieho konania fungovala vo webovom prehliadači.

Server musí obsahovať tieto tri programy:

- Apache
- PHP5
- MySQL

Webový prehliadač spracúva HTML kód, ktorý generuje server. PHP5 programovací kód práve tento HTML kód vytvára a posiela po sieti klientovi na webový prehliadač.

3 Štruktúra riešenia aplikácie

Aplikácia sa skladá s programu a databázy. Štruktúra riešenie aplikácie je rozdelená do troch častí:

- štruktúra databázy
- prihláška a formulár
- portál SSOŠ

Program je vytvorený pomocou webových programovacích jazykov PHP5, CSS - kaskádové štýly, HTML a technológie Ajax. Do aplikácie som vybral relačnú databázu MySQL. Okrem výberu databázy bolo treba navrhnuť štruktúru databázy, ktorá uchováva informácie s internetovej časti aplikácie (formulár) a intranetovej časti (administratívnej časti). Štruktúra databázy je tvorená pomocou tabuliek. Tabuľky sú rozdelené podľa informácií, ktoré sú v databáze uložené.

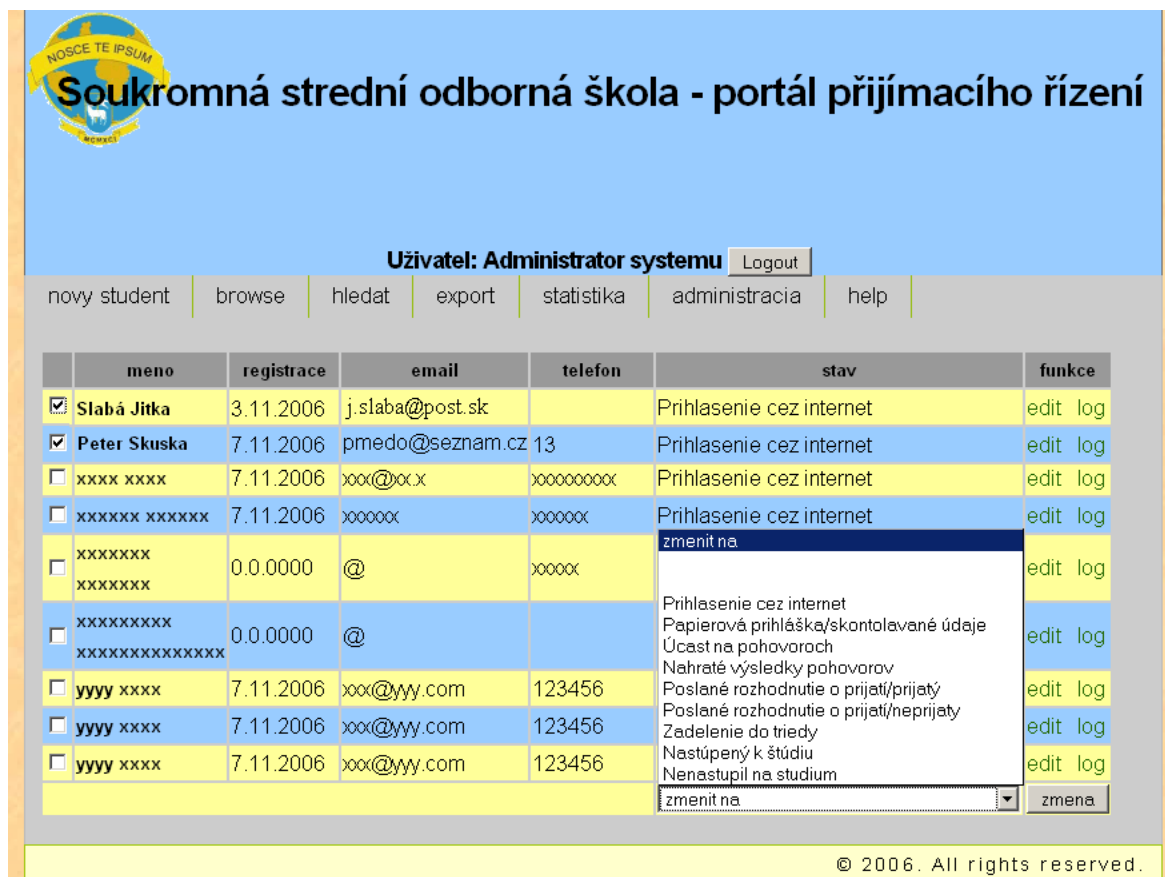
3.1 Štruktúra databáze

Databáza je rozdelená do troch skupín tabuliek. Skupiny tabuliek sú rozdelené podľa typu údajov, ktoré sú v databáze uložené: tabuľky s údajmi študentov, tabuľky s údajmi z číselníkov a tabuľky potrebné pre administráciu systému.

3.1.1 Tabuľky s údajmi študentov

V tabuľkách s údajmi o študentovi sa nachádzajú osobné údaje. Sú to meno a priezvisko študenta, mailové a telefonické kontakty, dátum narodenia, bydlisko a podobne. Do tabuliek sa tieto informácie ukladajú pomocou internetu alebo pomocou administratívneho rozhrania. Na internete vyplní študent alebo rodič prihlášku a tú potom vloží systém s údajmi o študentovi do tabuliek v MySQL relačnej databáze. V prílohe sa nachádza prihláška na strednú školu, ktorá je prístupná verejnosti cez internet. Má dve strany. Prvá je zameraná na osobné údaje študenta a druhá na prospech na základnej škole. Všetky tieto údaje sa ukladajú do relačnej databázy MySQL a spolupracujú s administratívnym rozhraním. Administratívne rozhranie má na starosti oddelenie strednej školy. Oprávnení pracovníci strednej školy môžu zasahovať do už do vytvorených údajov o študentovi, alebo môžu aj zaradiť nového študenta do databázy. Údaje o študentoch sú usporiadané a uložené v tabuľkách. Tabuľky sú navrhnuté podľa ich obsahu v relačnej databáze MySQL. V databáze je vytvorená tabuľka s údajmi, ktoré sa najviac používajú. Vodiacim znakom je v tabuľke **id** študenta. Ďalšími časťami modulu sú meno, mail, telefón, dátum registrácie a

stav študenta. Stav študenta vyjadruje, v akej časti prijímacieho konania sa študent nachádza. Začína prihlásením cez internet až nastúpenie alebo nenastúpenie na štúdium. Tabuľka s údajmi, ktoré sa najviac používajú v databáze má názov **ssos_student**. Na obrázku č.4. je znázornená administratívna časť aplikácie, na ktorom sú údaje z tabuľky načítavané do webového prehliadača.



	meno	registrace	email	telefon	stav	funkce
<input checked="" type="checkbox"/>	Slabá Jitka	3.11.2006	j.slaba@post.sk		Prihlásenie cez internet	edit log
<input checked="" type="checkbox"/>	Peter Skuska	7.11.2006	pmedo@seznam.cz	13	Prihlásenie cez internet	edit log
<input type="checkbox"/>	xxxx xxxx	7.11.2006	xxx@xx.x	xxxxxxxxxx	Prihlásenie cez internet	edit log
<input type="checkbox"/>	xxxxxx xxxxxx	7.11.2006	xxxxxx	xxxxxx	Prihlásenie cez internet	edit log
<input type="checkbox"/>	xxxxxxx xxxxxxx	0.0.0000	@	xxxxx	změnit na	edit log
<input type="checkbox"/>	xxxxxxxxxx xxxxxxxxxxxxxxxx	0.0.0000	@		Prihlásenie cez internet Papierová prihláška/skontolované údaje Účast na pohovoroch Nahraté výsledky pohovorov	edit log
<input type="checkbox"/>	yyyy xxxx	7.11.2006	xxx@yyy.com	123456	Poslané rozhodnutie o prijatí/prijatý Poslané rozhodnutie o prijatí/neprijatý	edit log
<input type="checkbox"/>	yyyy xxxx	7.11.2006	xxx@yyy.com	123456	Zadelenie do triedy	edit log
<input type="checkbox"/>	yyyy xxxx	7.11.2006	xxx@yyy.com	123456	Nastúpený k štúdiu Nenastúpil na štúdium	edit log

Obrázok č.4: Prehliadanie informácií o študentoch a stavy študentov

Tabuľky o študentovi majú jeden vodiaci kľúč ku každému študentovi . Je to **id** študenta, ktoré je jedinečné u každého študenta a je automaticky pridelené pomocou funkcie, ktorou disponuje MySQL server, **function insert_id()**, ktorá vráti identifikačné číslo posledného vloženého záznamu. Toto číslo sa spracuje a prideli novému študentovi číslo.

V tabuľke **ssos_student_person** sú osobné údaje študenta. Sú to id študenta, ulica, mesto, okres bydliska, štát, číslo občianskeho preukazu alebo pasu, národnosť a PSČ.

V tabuľke **ssos_student_2** sú študentové údaje o zdraví a záľubách. Okrem toho obsahuje aj informácie o súrodencoch a o otcovi študenta. Adresa otca sa zadáva do tabuľky iba vtedy, keď sa nezhoduje so študentovou adresou. Až má otec email a telefónne číslo, tak ho tam študent vypíše.

V tabuľke **ssos_student_3** sú informácie o matke študenta. Podobne ako otec, tak v tabuľke sú matkine údaje. Email a telefónne čísla rodičov sú dôležité na komunikáciu medzi školou, rodičom a študentom. Netreba zabúdať na to že každá tabuľka s informáciami o študentovi obsahuje vodiaci kľúč, ktorý má pridelený každý jednotlivý študent.

V tabuľke **ssos_student_birth** sú študentove údaje o narodení. Sú to najmä dátum narodenia, rodné číslo, mesto narodenia, kraj, štát, štátne občianstvo a pohlavie.

Tieto informácie sú potrebné pre spracovanie výkazov a štatistiky.

Informácie o vyštudovanej základnej škole študenta sú v tabuľke **ssos_student_school**.

Sú to informácie o názve základnej školy, ulici, PSČ, štáte, kraji a okrese mesta. V tejto tabuľke sa okrem informácií o základnej škole nachádzajú údaje o študentových pokarhaniach a pochvalách na základnej škole a o type štúdia na strenej škole, ktoré si zvolil na Európskom polytechnickom inštitúte, s.r.o. V tabuľke je aj záznam či študent má záujem o ubytovanie.

V tabuľke **ssos_grade** sa nachádzajú informácie o prospechu študenta na základnej škole. Tabuľka obsahuje predmety a známky študenta. Znamky študentov sú v piatich stĺpcoch, ktoré sú rozdelené na prvý a druhý polrok. Jeden stĺpec reprezentuje jeden ročník na základnej škole. Tabuľka obsahuje tak ako aj iné vodiaci znak id študenta. Kvôli štatistikám je identifikačné číslo predmetu rovnaké s identifikačným číslom predmetu v číselníku, ktorý vydáva štatistický úrad. Toto je posledná opísaná tabuľka, v ktorej sa nachádzajú informácie o študentovi.

3.1.2 Tabuľky s údajmi číselníkov

V týchto tabuľkách sú uložené údaje, ktoré vydal štatistický úrad. Sú to názvy a identifikačné čísla školských predmetov , štátov a okresov. Úrady môžu od školy vyžadovať výkazy. Sú to štatistiky, ktoré sú spracované pomocou číselníkov. Číselníky sú kódované v znakovnej sade UTF-8. Toto kódovanie je používané v celom systéme. Názvy tabuliek číselníkov v databáze sú tvorené z dvoch častí. Prvá je **epi_code_** a druhá časť je skratka číselníka (predmet, štát, obec, národnosť). Do tabuliek som číselníky vložil pomocou funkcie import zo súborov typu csv.

V tabuľke **epi_code_natonality** – sú informácie o národnostiach. Sú tu id čísla národností a mená národností.

V tabuľke **epi_code_state** sú informácie o štátoch. Sú tu id štátu a mená štátov.

V tabuľke **epi_code_shire** sú informácie o čísle a mene kraja.

V tabuľke **epi_code_discrit** sú informácie o čísle a mene okresu z číselníka. Obsahuje aj číslo, v ktorom kraji sa nachádza.

V tabuľke **epi_code_town** sú informácie o čísle, mene mesta a PSČ . Obsahuje aj číslo kraja a okresu, v ktorom sa mesto nachádza.

V tabuľke **epi_code_subject** je inportovaný číselník o školských predmetoch. V tabuľke sú polia o id čísla predmetu jeho dlhého a krátkeho mena.

3.1.3 Tabuľky administrovania systému

Tabuľky administrovania systému sú dôležité na chod administrovania prijímacieho riadenia. V tabuľkách sú údaje od prihlasovacích mien užívateľov, o zatriedovaní študentov do tried, až po informácie o aktivitách akcií (maily , tlačové zostavy).

V tabuľke **ssos_user** sú hodnoty o užívateľoch administrovaného systému. V tabuľke sú informácie o čísle užívateľa (id_user), prihlasovacie meno (user_login), heslo pre vstup do systému (user_login) a jeho meno (user_name). Informácie v tejto tabuľke sú dôležité kvôli bezpečnosti celého systému. Osoba, ktorá nebude mať oprávnenie v tabuľke, nemôže vykonávať žiadne operácie so systémom a ani si nemôže pozerať údaje o študentovi.

V tabuľke **ssos_class** sú informácie o zatriedení študentov do tried na základe odboru štúdia. Skladá sa z troch častí. Identifikačné číslo študenta (id_student), meno triedy do ktorej študent patrí, identifikačné číslo odboru.

V tabuľke **ssos_commission** sú informácie o komisii pre prijímacie pohovory. Tabuľka pozostáva z mena člena komisie (name_membre), číslo kola (round) prijímacích pohovorov (Denní1.kolo poh., Denní2.kolo poh. Distančný ročník, nultý ročník) a dátum skúšky. Z tejto tabuľky sú načítavané mená kôl štúdia do formulára na internetovej stránke školy.

V tabuľke **ssos_field_study** sú informácie o odboroch štúdia, ktoré škola ponúka. V tabuľke je číslo typu odboru a jeho meno. Mená odborov štúdia sú načítavané do prvej časti prihlášky (formulára) na internete a sú dostupné aj v intranetovej časti.

V tabuľke **ssos_rank_exam** sú hodnoty o výsledkoch prijímacích skúšok. Obsahuje údaje: identifikačné číslo študenta (id_student), identifikačné číslo odboru štúdia študenta (id_field), miesto a výsledky prijímacích pohovorov.

Do tabuľky **ssos_type_form** sú uložené formy štúdií, ktoré škola poskytuje. V tabuľke sú informácie o identifikačnom čísle formy štúdia (id_type_form) a o mene formy štúdia (name_form), na ktoré sa študent hlási.

V tabuľke **ssos_action** sú informácie o aktuálnom stave študenta. Stav študenta je informácia o tom, v akom štádiu je študent v prijímacom konaní. Stav študenta nám hovorí aj v akom štádiu sú spracované jeho výsledky z prijímacieho konania. Študent môže byť napríklad v týchto stavoch: prihlásenie cez internet, papierová prihláška skontrolovaná, účasť na pohovoroch, nahraté výsledky pohovorov, posledné rozhodnutie o prijatí/prijatý, posledné rozhodnutie o prijatí/neprijatý, zadelený do triedy, nastúpený k štúdiu alebo nenastúpil na štúdium.

V tabuľke **ssos_action_text** sú informácie identifikačného čísla akcie, identifikačného čísla e-mailu, a stave či je daný text aktívny, alebo nie je. Je dôležité z hľadiska histórie uchovávať texty a z hľadiska bezchybného a rýchleho chodu systému. Keď je daný text neaktívny tak sa nevyužíva, ale je len uložený v databáze. Je to posledná tabuľka administratívnej časti tabuliek.

3.2 Prihláška a formulár

Prihláška na SŠ je formulár, ktorý musí študent vyplniť, keď sa hlási na strednú školu. Do prihlášky v databáze SŠ sa ukladajú údaje o študentovi cez dve časti aplikácie. Sú to: internetová a intranetová časť aplikácie. Internetovú tvorí úvodný formulár (prihláška), do ktorého môžu vstupovať študenti a používatelia internetu bez hesla. Je vytvorený väčšinou v HTML kóde a sú v ňom odkazy na CSS- kaskádový štýl a PHP5. CSS- kaskádový štýl je

to programovací jazyk, ktorý sa používa na internete a zapisuje sa priamo do HTML kódu. Klient dostáva zo servera dáta v HTML kóde a CSS- kaskádovom štýle, ktoré spracuje vo webovom prehliadači.

3.2.1 Prihláška (formulár) na internete

V prihláške sú použité programy webových aplikácií: PHP5, CSS - kaskádové štýly, HTML a technológia Ajax. Väčšina prihlášky je písaná v programovacom jazyku PHP5 a v súbore **aplicat_index.php5**. Horná časť prihlášky, logá a adresa školy sú vytvorené v HTML kóde s odkazom na súbor **header.html**. Sú tam vložené dva obrázky, logo školy a telefón.

Vložené sú pomocou HTML kódu a príkazov `img` a `src`:

```


```

Na prihláške v hornej časti je adresa školy a mail. Po kliknutí na mail sa zavolá na predvolený mailový program napr. Outlook expres. Automaticky v mailovom programe sa vypíše mailová adresa školy. Odkaz na mailový program na vypísanie nového mailu sa programuje pomocou príkazov `"a href"` a `"mailto"`:

```
<a href="mailto:ssos@vos.cz" title="Napište nám email">ssos@vos.cz</a>
```

Toto bolo v hornej časti prihlášky v súbore **header.html**, ale vrátim sa k popisovaniu prihlášky **aplicat_idex.php5**. Červenú hviezdičku som vytvoril pomocou horného indexu párového tagu s HTML `[*]`. Študent si zvolí typ prihlášky pomocou **selectu** z tabuľky **type_study** a v súbore **aplicat_idex.php5** sa mu vyroluje výber, z ktorého si vyberie daný typ štúdia. Môže si vybrať (denné štúdium 1 kolo, denné štúdium 2 kolo, dištančné štúdium alebo nultý ročník). Príkaz **document.write** nám vypíše jednotlivé štúdia na monitor:

```
<select name="type_aplicat" id="type_aplicat">
<option value="NOT">(zvoľte typ)
<script type="text/javascript"> var id = new Array();id[0] = "Denné štúdium,1.ko" ;id[2] = "Denné
štúdium, 2.ko";id[3] = "Distančné štúdium";id[4] = "Nultý ročník"; </script><script
type="text/javascript">
for (key in id)
{ document.write("<option value=" + key + ">" + id[key] + "</option>"); }
</script>
</select>
```

Políčko priezvisko a meno je vytvorené HTML tagom `input`, ktorý nám vytvorí okno na vpisovanie priezviska a mena. Odkazuje sa pomocou PHP5 príkazom `require` na súbor **name.html**. Bunka je nastavené na veľkosť 25 pomocou príkazu `size`.

Zdrojový kód vytvorenia buniek na vpisovanie priezviska a mena:

```
<td width="80">Priezvisko</td>
<td><input name="s_name" id="s_name" value="" size="25"></td>
<td width="80"><label for="meno">Meno:</label></td>
<td><input name="p_name" id="p_name" value="" size="25"></td>
```

Adresa študenta je naprogramovaná v PHP5 a odkazuje sa na súbor **student_person1.php5**. Pomocou input sa vkladá adresa študenta. Ulica bude v tabuľke pod **name address_street** a **id address_street**.

Poštové smerové číslo bude v tabuľke v stĺpci **address_post**. Štát uloží vybranú hodnotu do **address_state**. Pomocou HTML výberu (select) sa vygenerujú štáty podľa číselníka "js_array('state')", z ktorých si študent vyberie.

Zdrojový kód výberu štátu:

```
<select name="address_state" id="address_state">
<?php
js_array('state');
?>
</select>
```

Kraj je vyberaný pomocou technológie Ajax z hodnôt v číselníku. Vyberanie som robil pomocou HTML výberu (select) a cez PHP5 som vyberal hodnoty z číselníka pomocou **js_array('shire')**.

Zdrojový kód výberu kraja:

```
<?php
js_array('shire');
?>
case 'shire':
$tbls = array("epi_code_shire");
$cols = array("id_shire AS id", "name_shire AS name");
$order = "name_shire ASC";
```

Okres som vytvoril HTML výberom (select) premennou "address_district".

Kraj som vytvoril HTML výberom (select) premennou "address_town".

Telefón vyplní študent pomocou HTML výberu (select) do premennej "phone".

Pomocou premennej title zobrazí webový prehliadač správu pre študenta. Vo formulári je predvolená nasledovná informácia: "Ak telefón nemáte, nechajte toto políčko prázdne".

Mail je naprogramovaný v PHP5 funkciou žiadosti (require) "email.html". V adresári application a súbore email.html je vstup mailu tvorený cez input a keď študent nemá mail, tak cez premennú title sa v prehliadači objaví informácia "Ak emailovú adresu nemáte, nechajte zavináč (@) alebo prázdne pole".

V súbore applicat_index.php5 je v PHP5 žiadosť (require) na tabuľku "student_person2.php5", ktorá vygeneruje tieto vstupné polia: číslo občianskeho "id_card",

rodné číslo "id_person", narodený dňa "birt_date", štát narodenia "birt_state", kraj narodenia "birth_shire", okres narodenia "birth_district", mesto narodenia "birth_town", štátne občianstvo "nationality", pohlavie "sex". Táto časť sa vygeneruje v PHP5 žiadosťou "require". Súbor aplicat_index.php5 vstúpi do adresára application a **student_person2.php5**. Súbor **student_person2.php5** vygeneruje tabuľku, ktorú študent môže vyplniť. Niektoré vstupné polia sa nevyplňajú, ale len sa vyznačujú určité možnosti.

Možnosti sa načítavajú s číselníkovi. Napríklad štát narodenia: "birth_state" načíta možnosti z databázy pomocou súboru js_array.php5. V súbore js_array.php5 nájde úlohu, ktorá vygeneruje štáty. Štáty sú uložené v databáze v tabuľke epi_code_state. Mená štátov sa vkladajú do tabuľky podľa číselníkov, ktoré vydáva ministerstvo.

Zdrojový kód výberu štátu vo formulári:

```
<?php
require "application/student_person2.php5";
?>
student_person2.php5
<td>Štát narodenia:</td>
<td width="">
<select name="birth_state" id="birth_state">
<?php // vytvorí html select zo štátmi podľa číselníku
js_array('state');
?>
</select>
</td>
js_array.php5
case 'state':
$tbls = array("epi_code_state");
$cols = array("id_state AS id", "name_state AS name");
$order = "name_state ASC";
```

Takto podobne je to aj v ďalších bunkách a poliach: kraj narodenia, okres narodenia, miesto narodenia, štátne občianstvo. Pole pohlavie je robené jednoduchým spôsobom, HTML výberom (option).

V nasledujúcej časti prihlášky je tabuľka zdravotný stav študenta. V súbore aplicat_index.php5 je PHP5 žiadosť (require) na tabuľku "school.php5". Tabuľka prijme súbor health.html. Súbor má v sebe tabuľku, do ktorej študent napíše svoj zdravotný stav. Údaje budú mať meno health. Pomôckou pre vypísanie študentovi je tittle . V tittle sa nachádza: v prípade plného zdravia napíšte len "zdravý". Toto zobratí prehliadač, keď sa na pole dostane užívateľ prehliadača myšou.

Po vyplnení poľa o zdravotnom stave, študent vyplní vstupné polia základnej školy. V súbore aplicat_index.php5 je PHP5 žiadosť (require) na súbor "school.php5", ktorý vygeneruje tabuľku. Súbor "school.php5" má v sebe tabuľku, ktorá vygeneruje tieto vstupné

polia: Názov základnej školy "school_name" , ulica "school_street" , PSČ: "school_post", štát "school_state", kraj "school_shire", okres "school_district" a mesto "school_town". Niektoré bunky prihlášky študent vyplňa a niektoré vyznačuje s vyrolovaných možností. Štát, kraj, okres a mesto sú polia ktoré, keď sa vyrolujú, študent vyznačí jeden s vygenerovaných riadkov. Generovanie údajov do riadkov nám robí technológia Ajax. Každý riadok stĺpca má svoje údaje uložené v databáze. Údaje v databáze sú dáta s číselníkov, ktoré vydáva ministerstvo. V číselníkoch sú dáta názvov štátov, krajov, okresov, miest a školských predmetov. Podobne sú vytvorené aj ostatné bunky, ktoré študent vyplňa. Sú to tabuľky o otcovi a matke. Ďalej vyplňa užívateľ záujmy študenta a svojich súrodencov.

Po vyplnení prvej strany prihlášky užívateľ vyznačí bunku "ďalej" a dostane sa na druhú stranu prihlášky vo webovom prehliadači.

3.2.2 Druhá strana prihlášky

Na druhej strane vyplňa užívateľ prihlášku o prospechu na základnej škole. Zdrojový kód druhej strany je v adresári formulár a súbore **aplicat_2.php5**. Zdrojový kód hlavičky druhej strany prihlášky je v súbore **header.html**.

Hlavička je tvorená logom a adresou školy a rovnako naprogramovaná , ako prvá strana prihlášky. Užívateľ zadá v prihláške do buniek známky zo základnej školy, priestupky a pochvaly, IZO základnej školy a vyznačí bunku či má záujem o ubytovanie. Po vyplnení druhej strany prihlášky a označení bunky "odoslať prihlášku" sa odošlú všetky údaje, ktoré užívateľ vyplnil na oboch stranách prihlášky.

3.3 Portál SSOŠ

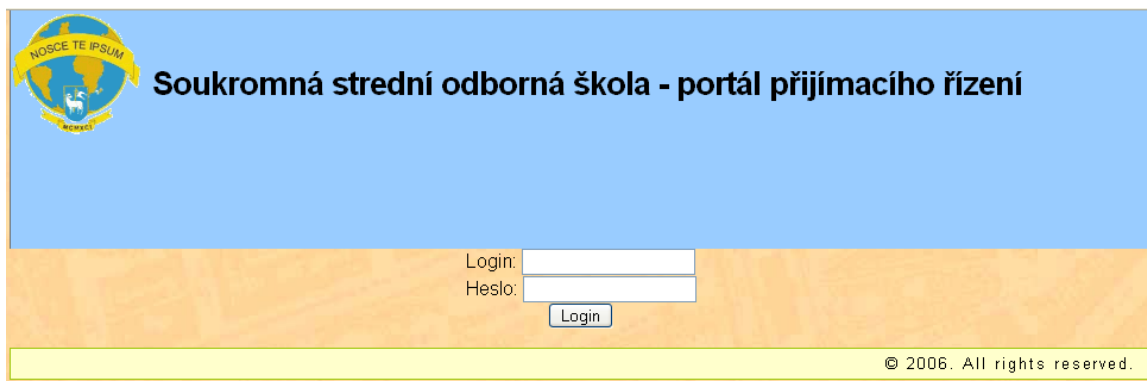
Portál je druhou časťou aplikácie. Je to interná vnútroškolská aplikácia spustiteľná vo webovom prehliadači. Je vytvorená pre administráciu prijímacieho konania nových študentov pre zamestnancov na strednej škole. Vstup do aplikácie je určený len cez školské počítače po zadaní mena a hesla. Po vstupe do internej časti, čiže na portálu má administratívny pracovník všetky práva na zmenu všetkých údajov týkajúcich sa prijímacieho konania na strednú školu. Intranetový portál SSOŠ je druhou časťou aplikácie. Intranetový portál som rozdelil na osem základných modulov:

- prihlásenie na portál
- nový študent
- browse – prezeranie
- hľadať
- export
- štatistika
- administrácia
- help – pomoc

3.3.1 Prihlásenie na portál

Prihlásenie do interného modulu prijímacieho riadenia je prihlásenie na portál strednej školy. Portál je určený výhradne pre pracovníkov Európskeho polytechnického inštitútu s.r.o. Portál spolupracuje s databázou informácií o študentoch a s rôznymi ďalšími dokumentami. K informáciám sa dostaneme cez databázu. Pripojenie k databáze vyvoláme funkciou **MySQL_Connect()**. Parametrom je meno počítača, na ktorom beží relačná databáza MySQL. V prípade, že pristupujeme k databáze, ktorá vyžaduje login heslo, zadáme ich ako ďalšie parametre. Výsledkom funkcie je zdrojový kód, ktorý je použitý v aplikácii v nasledujúcich riadkoch:

```
private function connect() {
    $this->conn = mysql_connect("localhost", "root", "");
    if (!$this->conn) {
        $this->err("", "mysql_connect", "connection error");
    } else {
        $db = mysql_select_db("ja");
        if (!$db) {
            $this->err("", "mysql_select_db", "db error");
        }
    }
}
```



Obrázok č.5: Prihlasovací modul na portál SSOŠ

Na portál SSOŠ sa dostanú užívatelia cez intranetový modul po zadaní prihlasovacieho mena a hesla v prihlasovacom okne zobrazeného na obrázku č5. Po skontrolovaní mena a hesla sa pripojí užívateľ na databázu, kde potom môže administrovať informácie na portáli. Zabezpečuje to funkcia, ktorá overí správnosť mena a hesla užívateľa, ktorý sa chce pripojiť na portál. Pri vytváraní hesla je použité kryptovanie MD5 (Message-Digest algorithm5). MD5 je široko používaná kryptografická hash funkcia z roku 1991.

V tabuľke **ssos_user** sú hodnoty o užívateľoch administrovaného systému. V tabuľke sú informácie o čísle užívateľa (`id_user`), prihlasovacie meno (`user_name`), heslo pre vstup do systému (`user_pwd`). Tieto údaje sú veľmi dôležité, lebo bez nich by sa neprihlásili pracovníci na portál. Sú dôležité aj z hľadiska bezpečnosti systému.

3.3.2 Nový študent

V tomto module aplikácie zadávajú a editujú administratívny pracovníci strednej školy prihlasovacie údaje študentov na školu. Po prihlásení na portál SSOŠ a vyznačení bunky **nový študent** je umožnené vytvoriť nového študenta. Pracovník vyplnení bunky modul osobné údaje od mena študenta až po mesto narodenia. Potom vyznačí bunku **uložiť** a pokračuje vyplňaním modulu bydlisko študenta. Pracovník pokračuje vyplňaním údajov o rodičoch, o základnej škole, o prospechu na základnej škole a nakoniec vyplní poslednú časť SSOS. V bunke SSOS vypíše užívateľ portálu typ prihlášky, odbor na aký sa hlási a či žiada študent o ubytovanie. Na obrázku č.6 je jeden zo šiestich modulov, v ktorom sa editujú osobné údaje študenta.

Soukromná střední odborná škola - portál přijímacího řízení

Uživatel: Administrator systemu

novy student | browse | hledat | export | statistika | administracia | help

osobne udaje | bydlisko | rodice | zakladni skola | znamky | SSOS

Jméno: Peter
Příjmení: Bakoš
Pohlaví: Muž Žena
e-mail: ja@ja.sk
Telefon: +420123321
Číslo OP/pasu: 123321
Občanství: Nejistěna
Rodné číslo: 0

Obrázok č.6: Pridanie nového študenta cez intranetový portál

3.3.3 Browse - prezeranie

Na intranetovom portáli SSOŠ po vyznačení bunky browse – prezeranie sa objavia najviac používané informácie o študentoch (obrázok č.4). V tomto module aplikácie administratívny pracovník mení stavy študenta v prihlasovacom konaní a jeho osobné údaje. Údaje v module browse sú vyberané z tabuľky s najčastejšie používanými informáciami.

Najčastejšie používané informácie o študentovi sú:

- meno
- dátum registrácie
- email
- telefón
- stav študenta

Administratívny pracovník môže zmeniť stav študenta cez bunku **zmena**. Stav nám hovorí o tom, v akej časti prijímacieho konania sa študent nachádza. Študent môže byť v stave: prihlásenie cez internet, papierová prihláška skontrolovaná, účasť na pohovoroch, nahrať výsledky pohovorov, poslané rozhodnutie o prijatí/prijatý, poslané rozhodnutie o prijatí/neprijatý, zadelenie do triedy, nastúpil na štúdium a nenastúpil na štúdium. Tieto vymenované stavy sú preddefinované a oprávnený pracovník strednej školy môže ich editovať. Druhou časťou editácie je zmena v osobných údajoch študenta. Zmeny v osobných údajoch je možné editovať po vyznačení bunky edit. Modul browse – prehliadanie sa prepne na modul nový študent, kde sa automaticky do buniek načítajú osobné dáta študenta z databázy, ktoré sa môžu následne editovať. Modul nový študent je popísaný v 3.3.2 kapitole.

3.3.4 Hľadat'

Na portáli SSOŠ po vyznačení bunky hľadat' sa objaví modul, ktorý slúži na vyhľadávanie a výber študentov s databázy. Vo vyhľadávaní môže užívateľ portálu vybrať vyhľadávanie podľa štyroch častí osobných údajov študenta:

- meno (firstname)
- priezvisko (lastname)
- telefónne číslo (phone)
- mail

Ďalej pracovník vyberie s vyhľadávacích znakov: rovná sa (=), nerovná sa (!=), obsahuje, neobsahuje databáza vypísanú informáciu v druhom stĺpci modulu. V treťom stĺpci vypíše informáciu, ktorú chce spracovať pracovník v databáze napr. Peter. Vyhľadávanie je znázornené na obrázku č.7. Po vyznačení bunky search sa zobrazia všetky nájdené mená z databázy vybraných študentov a ich najviac používané osobné údaje. Zobrazenie študentov je prepojené na modul browse – prezeranie.

NOSCE TE IPSUM

Soukromná střední odborná škola - portál přijímacího řízení

Uživatel: Administrator systemu Logout

novy student | browse | **hledat** | export | statistika | administracia | help

student search

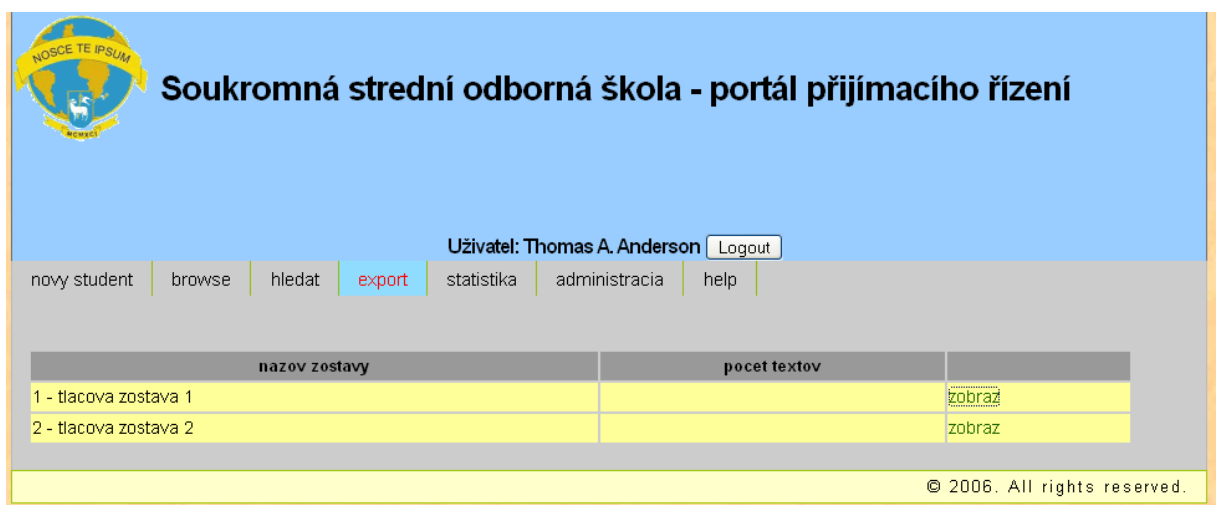
firstname	=	Peter	
lastname	=	Marek	
-	obsahuje		
-	neobsahuje		

search

Obrázok č.7: Vyhľadávací modul študentov

3.3.5 Export

Modul export na portáli pracuje s tlačovými zostavami. V tlačových zostavách sú texty. V jednej tlačovej zostave môže byť ľubovoľné množstvo textov. Tlačové zostavy nám vytvárajú výstupy z portálu. Výstupy zabezpečujú: posielanie mailov, tlač dokumentov a zobrazovanie výstupov tlačovej zostavy vo webovom prehliadači. Dokumenty môžu byť určené na posielanie pošty študentovi, školskú archíváciu alebo na výstupy pre štatistický úrad. Na obrázku č.8. je znázornený modul export ako vyzerá vo webovom prehliadači. Po vyznačení bunky zobraz nám vyexportuje databáza informácie vo webovom prehliadači k vybranej tlačovej zostave.



Soukromná střední odborná škola - portál přijímacího řízení

Uživatel: Thomas A. Anderson [Logout](#)

[novy student](#) | [browse](#) | [hledat](#) | **[export](#)** | [statistika](#) | [administracia](#) | [help](#)

nazov zostavy	pocet textov	
1 - tlačova zostava 1		zobraz
2 - tlačova zostava 2		zobraz

© 2006. All rights reserved.

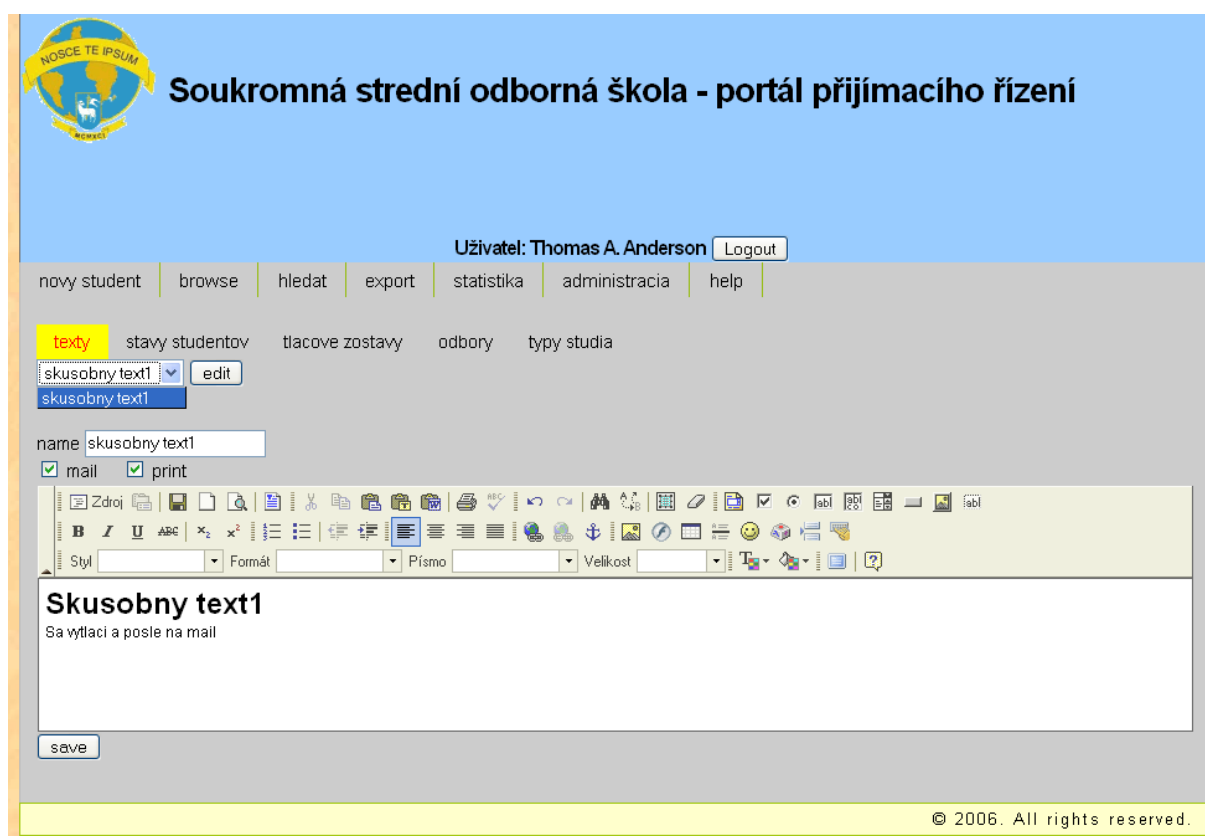
Obrázok č.8: Modul exportu tlačových zostáv

3.3.6 Administrácia

Administrácia slúži na vytváranie a zmenu údajov na intranetovom portáli. Údajmi na portáli sa myslia údaje potrebné k administrácii, prijímacieho konania.

- tlačové zostavy
- stavy študentov
- odbory
- typy štúdia

Všetky tieto údaje sú usporiadané v tabuľkách a sú uložené v databáze. Úlohou vytvoreného administratívneho modulu je ukladanie a generovanie údajov do tabuliek v databáze. Napríklad administratívny pracovník môže pridať nový text, ktorý bude slúžiť na pozvanie študentov na prednášky v škole. Tento modul je znázornený na obrázku č.9.



Obrázok č.9: Modul administrácie textov

Administrácia sa robí pomocou funkcií. Tieto sú v súbore **Admin.php5**. Názvy funkcií sú tieto: view, actionSave, action s(), text save, text(), field a type().

V module **texty** sa píše a editujú texty. Ku každému textu je možné v aplikácii vybrať mail alebo tlač (print). Pri vytváraní textov v module je vytvorená funkcia vkladania osobných údajov o študentovi z databázy pomocou znaku percento a mena poľa v tabuľke:

- priezvisko študenta: %name_last%
- meno študenta %name_first%
- email %email%
- telefón %phone%
- dátum registrovania prihlášky %regdate%

V module **stavy študentov** sa editujú druhy stavov študentov od prihlásenia cez internet až po nastúpil alebo nenastúpil na štúdium. Tento modul je z časti popísaný v mojej bakalárskej práci v časti 2.2. Riešenie analýzy aplikácie. Stav študentov môžeme pridávať a editovať v databáze. Stav študenta je určitá akcia, ku ktorej môžeme priradiť text a ten je odoslaný pri zmene stavu študenta. Môže byť odoslaný na mail alebo aj na tlačový výstup a toto nám určuje tlačová zostava. Týmto spôsobom sa automaticky posielajú maily a tlačové výstupy.

Tlačová zostava je modul, v ktorom sú uložené názvy tlačových zostáv, ktoré sme vytvorili. V tlačových zostavách je možné prideliť si texty k nami vytvorenej tlačovej zostave. Tlačové zostavy je možné meniť (editovať) na portáli cez bunku administrácia a bunku tlačové zostavy. V bunke tlačové zostavy pridelujeme alebo odoberáme k tlačovým zostavam ľubovoľný počet textov.

Ďalšou časťou administratívneho modulu aplikácie je modul editácia **odborov**. V module odbory sa nachádzajú názvy odborov, ktoré škola ponúka. V tomto module sa dajú pridávať a mazať odbory. Odbory v databáze sú v tabuľke **ssos_field_study**. V aplikácii sú preddefinované nasledovné odbory štúdia:

- Počítačové elektronické systémy
- Zahraničný obchod
- Komerčné právo
- Manažérstvo hotelov a ciest
- Bankový manažér

Typy štúdia je modul, v ktorom sa nachádza akým typom prihlášky sa študent na školu hlási. V tomto module sa editujú typy a prihlášky. V aplikácii sú preddefinované nasledovné typy prihlášky, akými sa študent môže hlásiť na školu:

- Denné štúdium, 1.kolo
- Denné štúdium, 2.kolo
- Dištančné štúdium
- Nultý ročník

3.3.7 Help - pomoc

Posledným modulom aplikácie je help. Help je pomôcka pri správe aplikácie. Help slúži na pomoc pri pracovaní s portálom strednej školy. Na portáli sú jednoduchou formou popísané funkcie intranetového portálu strednej školy. Je tu popísaná pomoc ako generovať mená, maily a telefónne kontakty z polí v relačnej databáze MySQL.

3.3.8 Bezpečnosť intranetu

Do druhej časti aplikácie prijímania žiakov na strednú školu, školského intranetu sa administratívny pracovník dostane len po zadaní mena a hesla na školskom počítači. Týmto sa zabezpečí bezpečnosť prístupu k osobným údajom študentov. Z intranetovej časti portálu sa potom odhlasuje vyznačením bunky Logout. Logout je na portály kvôli bezpečnosti informácií. Oprávnený pracovníci by mali kvôli bezpečnosti po skončení administrácie údajov na portáli prijímacieho konania strednej školy sa odhlasovať. Bezpečnosť je tu na mieste lebo databáza obsahuje osobné informácie o študentoch ako sú mená, rodné čísla, email telefónne čísla a adresy bydliska. Tieto informácie po neodhlásení by následne mohli byť zneužitú, zmenené alebo vymazané z databázy.

3.4 Implementácia systému

Úlohou implementácie bolo zaviesť funkčnú aplikáciu prijímacieho konania na strednú školu na školský server. Aplikácia bola na mieru vytvorená pre strednú školu. Pri spracovaní požiadaviek bolo nutné dávať pozor, aké programy a databázy môžu byť implementované na školský server. Po spracovaní požiadaviek a jej analýze bol vytvorený program na mojom súkromnom počítači. Prvým krokom bolo inštalovanie programu Xampp. Xampp má v sebe webserver Apache, webový programovací jazyk PHP5 a relačnú databázu MySQL. Tieto tri aplikácie navzájom spolupracujú. Po inštalácii programu Xampp bola naprogramovaná aplikácia prijímacieho konania na súkromnú strednú školu na mojom počítači. Aplikácia bola testovaná na mojom počítači a neskôr s nej bola vygenerovaná databáza, ktorá bola implementovaná na školský server do relačnej databázy MySQL. Na serveri bolo nutné nastaviť prístup do databázy, ktorý bol iný ako na mojom počítači. Na školský server bolo nutné nakopírovať zdrojové kódy týkajúce sa aplikácie prijímacieho konania. Poslednou časťou implementácie aplikácie prijímacieho konania bolo testovanie funkčnosti na školskom serveri.

Záver

Úlohou bakalárskej práce bolo vytvoriť softvérovú aplikáciu pre prijímanie žiakov na strednú školu. Aplikácia je prístupná na internete a na intranete strednej školy. Na internete je prístupný verejnosti pričom je umožnené vyplňať len prihlášku na strednú školu. Editovanie všetkých informácií je umožnené len pracovníkom školy po zadaní mena a hesla pri vstupe na portál strednej školy. V softvérovom riešení bakalárskej práci je využitá relačná databáza MySQL, webový programovací jazyk PHP5, web server Apache, technológia ajax, CSS – kaskádové štýly a HTML kód. Základom všetkých použitých softvérov bol program Xampp, ktorý podporoval všetky softvéri v aplikácií použité. Všetky vymenované softvéri a technológie sú poskytované zadarmo. Pomocou týchto revolučných softvérov som vytvoril aplikáciu, ktorá umožňuje elektronickou formou prijímanie žiakov na strednú školu. Program uľahčuje a zjednodušuje administratívnu činnosť celého prijímacieho konania. Tiež zjednodušuje ukladanie osobných údajov študentov do databázy na školskom serveri. Uľahčujú to aj číselníky vložené do programu, vytvorenie tlačových zostáv a textov. Texty vytvárajú mailové a tlačové výstupy. Softvérová aplikácia pre prijímanie žiakov na stredné školy si kládla za cieľ zlepšenie, zjednodušenie a uľahčenie administratívnej práce pre pracovníkov na strednej škole. Tento cieľ sa mi podarilo naplniť vďaka vytvorenej aplikácie.

Resumé

Program prijímacieho konania mal za úlohu uľahčiť, zlepšiť a zjednodušiť prácu pri evidencii a postupe prijímacieho konania nových študentov na školu. Program je navrhnutý tak, aby boli splnené všetky požiadavky administratívnych pracovníkov na strednej škole. Portál je navrhnutý pre pracovníkov strednej školy a pre študentov, ktorí sa hlásia na školu. Intranetový portál je prístupný všetkým pracovníkom strednej školy, ktorým je vytvorený prístup cez meno a heslo. Oprávnení pracovníci majú právo editovať všetky informácie týkajúce sa prijímacieho konania študentov na strednú školu. Internetový portál je prístupný všetkým užívateľom internetu a je vytvorený hlavne pre študentov. Študenti na internete majú právo vyplniť, len prihlášku na strednú školu čím sú zaradený do prijímacieho konania na školu a automaticky je im poslaná registrácia. Program svojou jednoduchosťou nevyžaduje, aby zamestnanci boli školení.

Moja bakalárska práca je rozdelená do troch kapitol, ktoré sú ďalej rozpracované do podkapitol. V prvej kapitole som popísal teóriu programovacích systémov, v druhej kapitole som popísal analýzu problému prijímania žiakov na strednú školu a tretia kapitola sa zaoberá štruktúrou riešenia aplikácie prijímacieho konania na strednú školu.

Použitá literatúra

Knižné publikácie:

LANČI, J. *Softwarové riešenie online prijímacího řízení EPI s.r.o.*. Bakalárska práca. Kunovice: EPI, 2004.

E.NARAMORE, J.GERNER, Y.LE SCOUARNEC, J.STOLZ, M.K.GLASS *Vytváříme webové aplikace v PHP5, MySQL a Apache*, Vydání první, Computer Press a.s., Brno 2006. ISBN 80-251-1073-7.

BRÁZA, J. *PHP 5 začínáme programovat*, Vydání první, Grada Publishing, a.s., Praha 2005. ISBN 80-247-1146

Internetové zdroje:

<http://www.php.net/history>

http://www.php.sk/php_tutor/tut_1.php

http://www.php.sk/php_tutor/tut_3.php

http://www.php.sk/php_tutor/tut_5.php

http://www.php.sk/php_tutor/tut_6.php

<http://www.linuxsoft.cz/php/Serial-PHP.pdf>

http://www.php.sk/php_tutor/tut_10-1.php

<http://php.mirror.camelnetwork.com/manual/sk/history.php>

<http://www.w3schools.com/>

<http://full.nkkr.cz/nkkr/NKKR0101/0101041.html>

<http://www.kirp.chtf.stuba.sk/~cirka/vyuka/xhtml/kap2.php>

<http://sk.wikipedia.org/wiki/XML>

<http://sk.wikipedia.org/wiki/MD5>

<http://www.pspad.com/sk/pspad.htm>

<http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/getfile.php/diplomovka%20Ajax.pdf?id=95&fid=157&type=application%2Fpdf>

Prílohy

Príloha 1: Prihláška

Príloha 2: Zdrojové kódy aplikácie