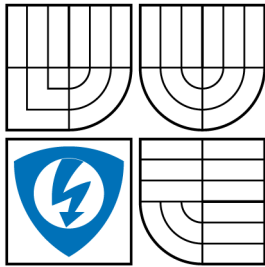


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

WEBOVÁ APLIKACE PRO SPRÁVU UČEBNY WEB APPLICATION FOR CLASSROOM ADMINISTRATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VERONIKA BARTOŇOVÁ

VEDOUcí PRÁCE
SUPERVISOR

ING. ONDŘEJ KRAJSA

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Studentka: Veronika Bartoňová

ID: 102388

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Webová aplikace pro správu učebny

POKYNY PRO VYPRACOVÁNÍ:

Podrobně se seznámte s možnostmi multiplatformního programování pro Internet. Ve Vámi vybraném vývojovém nástroji navrhnete a realizujete aplikaci pro vzdálenou správu telefonických přístrojů. Ve svém návrhu upřednostněte její možnou implementaci do webového prostředí a zabezpečte, aby bylo možné tyto přístroje jednoduchým způsobem katalogizovat. Zabezpečte tuto aplikaci proti vniknutí neautorizovanou osobou.

DOPORUČENÁ LITERATURA:

[1] KOSEK, Jiří. PHP :Tvorba interaktivních internetových aplikací. Podrobný průvodce /Praha :Grada Publishing,1998. 1. vyd. 490 s. ISBN 8071693731

[2] NARAMORE, Elizabeth. PHP5, MySQL, Apache: vytváříme webové aplikace / Vyd. 1. Brno : Computer Press, 2006. 813 s. ISBN 80-251-1073-7

[3] KOFLER, Michael; ÖGGL, Berndt .PHP 5 a MySQL 5 : průvodce webového programátora, Vyd. 1. Brno : Computer Press, 2007. 607 s., ISBN: 978-80-251-1813-9

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Ondřej Krajsa

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

ABSTRAKT

V této bakalářské práci se zabývám návrhem a realizací webového rozhraní pro vzdálenou správu telefonních přístrojů. Pro realizaci tohoto projektu bylo zvoleno vývojové prostředí jazyka PHP a databázový systém MySQL. Výsledkem celého projektu bude vytvoření aplikace, zabezpečené proti vniknutí neautorizovanou osobou, pro jeden konkrétní typ telefonního přístroje. Tato aplikace bude nejenom schopna katalogizovat přístroje, ale i provádět jejich celkovou administrativu. Tím by mělo dojít k usnadnění a zefektivnění vzdálené správy telefonních přístrojů.

KLÍČOVÁ SLOVA

PHP, MySQL, webové rozhraní, databáze, aplikace, VoIP, skript, tabulka

ABSTRACT

In this bachelor's thesis I deal with design and implementation of web-based interface for remote administration of phones. For the implementation of this project was chosen development environment language PHP and MySQL database system. The result of this project will create a application, which is secure against the intrusion of unauthorized persons, for one particular type of telephone set. This application will be able to catalog not only equipment but also to carry out all their administration. This should be facilitate and accelerate the setting of the remote VoIP phones.

KEYWORDS

PHP, MySQL, Web interface, database, application, VoIP, script, table

BARTOŇOVÁ V. *Webová aplikace pro správu učebny*. Místo: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2009. Počet stran 36 s., Počet stran příloh 6 s. příloh. Bakalářská práce. Vedoucí práce byl Ing. Ondřej Krajsa.

Poděkování

Ráda bych touto cestou poděkovala Ing. Ondřeji Krajsovi, za jeho skvělý pedagogický přístup, odbornou pomoc a cenné rady při zpracování bakalářské práce práce.

OBSAH

Úvod	12
1 Problematika webového rozhraní	13
1.1 Základní definice	13
1.1.1 Prezentací vrstva	14
1.1.2 Logická vrstva	15
1.1.3 Datová vrstva	16
1.2 Programovací jazyky pro dynamické webové aplikace a databázové systémy	17
1.2.1 Perl	17
1.2.2 PHP	18
1.2.3 AJAX	19
1.2.4 SQL	19
1.3 Zabezpečení webových aplikací	20
2 Návrh projektu a jeho realizace	22
2.1 Definice problému a přesná specifikace zadání	22
2.2 Analýza systému	22
2.3 Návrh struktury	23
2.3.1 Návrh databáze	23
2.3.2 Systém správy relační databáze a normalizace relací	25
2.4 Tvorba základního kódu a sestavení aplikace	26
2.4.1 Přihlašování	27
2.4.2 Správa uživatelů	28
2.4.3 Správa telefonních přístrojů	30
3 Závěr	32
Literatura	33
Seznam symbolů, veličin a zkratk	34
Seznam příloh	36
A První příloha	37
A.1 Vývojový diagram pro oprávnění uživatel (user)	37
A.2 Vývojový diagram pro oprávnění host (guest)	38

A.3	Datové typy v jednotlivých tabulkách	39
A.4	Mapa aplikace	41
B	Druhá příloha	42
B.1	Přiložené médium (CD)	42

SEZNAM OBRÁZKŮ

1.1	Třívrstvý model webové aplikace	13
2.1	Vývojový diagram pro oprávnění - administrator	24
2.2	Screenshot Primární klíče a relační vztahy mezi jednotlivými tabulkami	26
2.3	Screenshot Úvodní stránka	27
2.4	Screenshot HTML formulář Registrace	28
2.5	Screenshot Zápatí stránky s tlačítky určené pro správu uživatelů . . .	29
2.6	Screenshot Přehled registrovaných uživatelů s informací o jejich právech	29
2.7	Screenshot Přehled přístupových rolí	30
2.8	Screenshot Zobrazení evidence přístroje s funkčními tlačítky	30
A.1	Vývojový diagram pro oprávnění - guest	37
A.2	Vývojový diagram pro oprávnění - guest	38
A.3	Tabulka Uživatel	39
A.4	Tabulka Konfigurace	39
A.5	Tabulka Oprávnění	40
A.6	Tabulka Telefon	40
A.7	Mapa aplikace	41

ÚVOD

V této bakalářské práci se zabývám návrhem a realizací webového rozhraní pro vzdálenou správu telefonních přístrojů. Celá problematika webových rozhraní se stává v dnešní době velmi aktuálním tématem díky svému širokému uplatnění. Webová rozhraní je možné najít jak v soukromém sektoru (informační systémy řešené na zakázku) tak i na veřejné síti Internet v podobě různých internetových obchodů, blogů, freemailů nebo diskusních fór. Velký podíl na rozšíření webových aplikací má v posledních letech především rozvoj informačních technologií. Díky přenosové rychlosti, kterou v tuhle chvíli veřejná síť Internet disponuje není problém využívat Internet jako médium, které využívají nové technologie (VoIP, Streaming a jiné).

Většina webových rozhraní - aplikací, která se nachází na Internetu je založena na stejném principu - využití internetového prohlížeče jako rozhraní pro práci v daném aplikaci. Jako nejčastější vývojové prostředí se volí jazyk PHP nebo Perl. Oba jazyky jsou nezávislé na platformě, umožňují kompatibilní přístup k technologiím typu XML a dokáží spolupracovat s databázovým systémem MySQL. Největší rozdíly mezi těmito dvěma jazyky jsou v jejich syntaxi.

Pro realizaci tohoto projektu jsem si vybrala vývojové prostředí jazyka PHP z důvodu masivnější rozšířenosti a jeho jednodušší syntaxe (v porovnání s jazykem Perl). Dále využiji vlastností databázového systému MySQL, a jazyka SQL pro komunikaci s databázemi. Výsledkem celého projektu bude vytvoření aplikace, zabezpečené proti vniknutí neautorizovanou osobou, pro jeden konkrétní typ telefonního přístroje. Tato aplikace bude nejenom schopna katalogizovat přístroje, ale i provádět jejich celkovou administrativu. Tím by mělo dojít k usnadnění a zefektivnění vzdálené správy telefonních přístrojů.

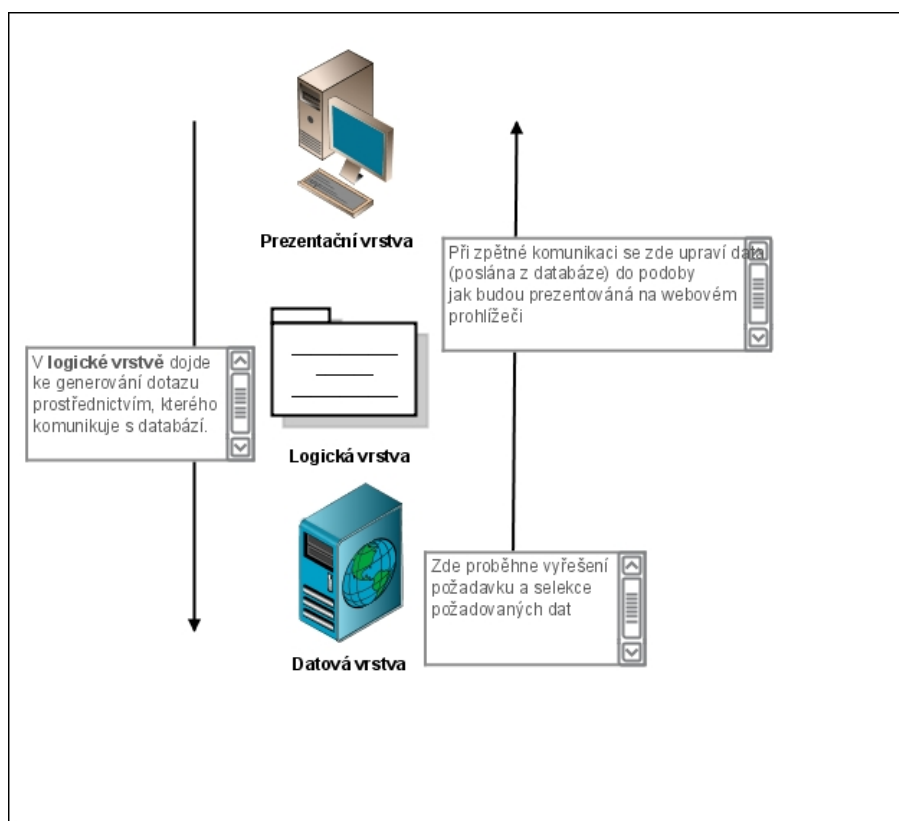
1 PROBLEMATIKA WEBOVÉHO ROZHRAŇÍ

1.1 Základní definice

Pod pojmem webové rozhraní si lze představit aplikaci, která je zpřístupněna uživatelům na webového serveru. Poskytuje uživatelům rychlejší a efektivnější přístup k požadovaným informacím pomocí webového prohlížeče, také snadnější ověřování daných údajů nebo bezpečný přístup k nim.

Webová rozhraní se nejčastěji používají v případech kdy je nutné poskytnout přístup k aplikacím, které jsou lokálně nabízeny a šířeny. Velmi často se vyskytují v podnikových nebo jiných informačních systémech z důvodu malé softwarové náročnosti tzv. není nutné šířit nebo instalovat software pro přístup k aplikacím a jejich správu. Webová rozhraní se dají najít i v jiných odlišných oblastech např. v internetových obchodech, freemailů ¹, online aukcích, weblogů či diskuzních fórech.

Struktura webového rozhraní je obvykle založena třech vrstvách: prezentační, logická a datová. Viz Obr.1.1



Obr. 1.1: Třívrstvý model webové aplikace

¹Veřejně dostupná služba v internetu, která nabízí uživatelům e-mailovou schránku zdarma.

1.1.1 Prezentační vrstva

První vrstvu představuje webový prohlížeč - počítačový program, který slouží k prohlížení WWW (World Wide Webu). Tento program umožňuje komunikovat s HTTP serverem a zpracovávat přijaté kódy (HTML, XML..). Obecně se prohlížeče dělí na textové a grafické. Textové prohlížeče jsou schopny zobrazovat pouze jednoduchý text na rozdíl od grafický prohlížečů, které zvládají zobrazovat složitější formátování textu v kombinaci s obrázky.

Grafické prohlížeče dále rozlišujeme podle jejich renderovacího jádra² používaného v prohlížeči. Pro zobrazení zvláštních součástí stránky jako jsou např. Java applety nebo Flash animace, je nutno nainstalovat speciální zásuvné modely.

Mezi nejznámější renderovací jádra patří:

- Gecko (prohlížeč: *Mozilla Firefox*) - patří do skupiny open-source³ renderovacích jáder používané produkty Mozilla pro zobrazení webových stránek. Toto jádro je psáno v programovacím jazyce C++ a chráněno trojlicencí MPL/GPL/LGPL, která umožňuje využití jádra i v jiných prohlížečích jako např. Flock, K-Meleon, Epiphany. Počáteční vývoj tohoto jádra probíhal ve firmě Netscape Communications Corporation, ale v roce 1998 jej převzala Mozilla Corporation. Tato firma rozšířila vývoj, mimo renderování webových stránek, i na vykreslování grafického rozhraní (XUL). Díky tomuto vývoji se jádro stalo mutliplatformní a je schopno být využito na různých platformách jako Microsoft Windows, Linux, Mac OS X a další.
- Trident (prohlížeč: *Windows Internet Explorer, Maxthon, Netscape Browser*) - renderovací jádro používané v prohlížečích určených pro Microsoft Windows. Toto jádro je pověstné svou slabší podporou webových standardů a slabší bezpečností. Za to, ale díky implementaci do OS Windows ho využívá řada instalovaných aplikací - nápověda systému Windows či kancelářský balík Microsoft Office. V dnešní době patří mezi nejpoužívanější renderovací jádro mezi uživateli.
- KHTML (prohlížeč: *Konqueror, Safari*) - renderovací/vykreslovací jádro psáno v jazyce C++ je chráněno LGPL a bylo vytvořeno v rámci projektu KDE⁴. Na tomto jádře je založena např. i RSS čtečka Akregator. Od roku 2002 je vlastníkem jádra firma Apple.

²Software, který použije obsah webové stránky v daném kódu a zobrazí ho do okna webového prohlížeče.

³Počítačový software s otevřeným zdrojovým kódem - uživatelům je povoleno ho prohlížet či upravovat.

⁴Desktopové prostředí pro unixové operační systémy

1.1.2 Logická vrstva

Střední vrstvu představují nástroje pro generování stránek. Základním principem generování stránek na serveru je použití HTML kódu, který je zde dynamicky generován a to na základě požadavku uživatele. Následně je tento kód odeslán do prohlížeče. V případě potřeby lze na serverech generovaných stránkách použít i klientské technologie (např. JavaScript). Celková interakce s webovou aplikací na serveru probíhá pomocí odkazů nebo HTML formulářů.

Mezi nejznámější nástroje pro dynamické generování HTML stránek patří:

- SSI (*Server Side Includes*) [1]- webový server přímo zpracovává jednoduché instrukce zapsané do HTML kódu. Server soubor s instrukcemi pozná podle přípony souboru (obvykle `.shtml`)
- CGI skripty (*Common Gateway Interface*) - externí programy, které jsou na požadavek od uživatele spuštěné WWW serverem jako samostatné procesy. Nevýhodou těchto skriptů je nízká rychlost startu - tzv. operační systém musí založit nový proces a v něm spustit požadovanou aplikaci.
- ASP (*Active Server Pages*) - skriptovací platforma vyvíjena společností Microsoft pro dynamické zpracování webových stránek na straně serveru. Webová stránka s příponou `.asp` se na serveru, před odesláním uživateli, zpracuje (tj. zašle se výsledek toho co vygeneroval interpret ASP kódu). Tato platforma pracuje s programovými jazyky VBScript a JScript.
- PHP (*Hypertext Preprocessor*) - skriptovací programovací jazyk, který byl vyvinut pro dynamické internetové aplikace. Toto téma bude rozvinuto později.
- ASP.NET - součást .NET Frameworku⁵ firmy Microsoft pro tvorbu webových aplikací a služeb. Nástupce ASP a konkurentem JSP (Java Server Pages). Výhodami této platformy jsou podpora přes 20 jazyků, bohatý výběr ovládacích prvků a knihoven tříd nebo schopnost cachovat celou stránku. Lze jej provozovat na různých platformách i webových serverech např. IIS (Windows), Apache (Windows, Linux s open source implementací .NETu Monem⁶)

⁵Nejrozšířenější platforma pro osobní počítače s operačním systémem Microsoft Windows, Microsoft Vista. Nejpoužívanější programovací jazyky pro vývoj .NET aplikací jsou C, Visual Basic .NET a Delphi.

⁶Projekt Mono vyvíjí firma Novell s cílem vytvoření sady nástrojů kompatibilních s prostředím .NET

1.1.3 Datová vrstva

Třetí vrstvu tvoří databázový systém. Tyto systémy jsou schopné shromáždit velké množství specifických informací, různým způsobem je editovat, ukládat a rovněž udržovat je na jediném centrálním místě. Databázový systém je skládá z několika částí:

1. Systém řízení dat (SŘBD)- jedná se o softwarové vybavení, které zajišťuje spolupráci s databází, neboli tvoří rozhraní mezi aplikačními programy⁷ a uloženými daty. Toto softwarové vybavení je schopno zpracovat několik datových modelů, využívá správy klíčů⁸ a některých jazyků vyšší úrovně pro manipulaci a definici dat (SQL, QBE), zajišťuje integritu dat a autentizaci uživatelů. Mezi další její výhody patří možnost uložení procedur, využití triggerů, robustnost a zotavitelnost po chybách bez ztráty dat. SŘBD využívají například: Oracle, DB2, Firebird, Microsoft Access, Microsoft SQL Server, MySQL a další.
2. Databáze - čili *báze dat*⁹, v mnoha literaturách je databáze popisována jako skupina informací uspořádaných podle určitých pravidel tak, aby následná práce s nimi byla co nejjednodušší, což je lehce zavádějící. Ve své podstatě se pojmem databáze označuje pouze skladiště dat, ve kterých jsou data uložena a zpracována nezávisle na aplikačních programech. Z hlediska způsobu ukládání dat a vztahů mezi nimi rozdělujeme databáze do základních typů:
 - Hierarchická databáze
 - Síťová databáze
 - Relační databáze
 - Objektová databáze
 - Objektově-relační databáze

V dnešní době je věnována velká pozornost především databázovým systémům typu Client - Server (C/S). Tyto systémy zvyšují výkonnost databáze tím, že oddělují SŘBD od databázové aplikace (v minulosti byla většina databázových systému navržena na programu, který spojoval SŘBD a databázovou aplikaci). V praxi to vypadá takto: databázová aplikace je provozována na jedné nebo několika pracovních stanicích (Pc) a komunikuje prostřednictvím sítě s jedním nebo více SŘBD běžícími na jiných počítačích. [8]

⁷Programy, které umožňují vybírat, prohlížet, aktualizovat informace uložené prostřednictvím SŘBD a z databáze.

⁸Jedná se o definování unikátních a primárních klíčů či implementaci cizích klíčů.

⁹Myšleno - uložená data.

Struktura systému typu C/S se dělí na následující části:

- Klientská část - tuto část má k dispozici samotný uživatel, kterému je umožněno pracovat s daty (spuštění, vkládání dat, kladení dotazů a požadavku)
- Serverová část - tato část má na starosti údržbu dat (konzistenci, zálohování replikace), kontrolu a zachování integrity (opravy, změny, rušení), provádění příkazů, které vydá klientská část aplikace (např. výpočet nebo selekce)

Výhodou tohoto systému je menší zatížení sítě (redukce množství posílaných dat), naopak za nevýhodu byla označena pořizovací cena serveru a databázového serveru.

1.2 Programovací jazyky pro dynamické webové aplikace a databázové systémy

Díky prudkému rozvoji skriptovacích jazyků určených pro použití na Internetu již není problém vybrat si z velké nabídky programovacích jazyků. Pro tvorbu webové aplikace jsou především, díky svému OOP konceptu, určeny ASP, Python, JSP, PHP a Perl. Právě poslední dva zmíněné jazyky používají programátoři nejčastěji pro tvorbu webových aplikací.

Existuje ale i řada systémů, které se přímo specializují na tvorbu webových aplikací (editory). Užití takovýchto systémů vede často ke snížení chyb v kódu aplikace především díky větší jednoduchosti a přehlednosti kódu.

1.2.1 Perl

Perl (*Practical Extraction and Reporting Language*) patří do skupiny interpretačních programových jazyků, které se využívají na World Wide Webu v oblasti zpracování souborů. Tento jazyk se používá při tvorbě velmi složité a rozsáhlé aplikace podporující práci s grafickým uživatelským rozhraním nebo práci s databází. Velmi často je možné se s ním setkat i v síťové komunikaci.

Autorem tohoto jazyka je označován Larry Wall, který vytvořil roku 1986 jazyk pro zpracování textu závislého na dálkově řízeném počítači. První verze (Perl 1.0) byla vydána prostřednictvím diskuzní skupiny alt.comp.sources na Usenetu v roce 1987. Největší rozšíření dosáhl Perl ve verzi 5 (aktuální verze je 5.10.0), kde tvůrci vylepšili datové struktury a umožnili v tomto jazyku objektově programovat, také

se tento jazyk stal nástrojem pro tvorbu CGI skripty. V poslední době získal Perl oblibu mimo jiné i v bioinformatice¹⁰ [9]

Jazyk Perl má určitou podobnost s přirozeným jazykem (angličtinou) a tím je poměrně snadno čitelný i pro člověka, který se programováním nezabývá. Syntaxe Perlu není přesně definovaná jako u ostatních programovacích jazyků. Proměnné se v tomto jazyku nemusí deklarovat přímo, také zde není možnost definovat vlastní datové typy a neexistuje typová kontrola. Perl nerozlišuje mezi celými a racionálními čísly, bere všechny jako čísla racionální - tedy jako jeden datový typ.

Původně byl Perl napsán pro platformu Unix, ale dnes běží i na jiných systémech (Windows, MS DOS, Novell. .). Perl je vhodný i pro práci s databázemi, jeho databázové integrační rozhraní (DBI) podporuje množství databází včetně Oracle, Sybase, PostgreSQL, MySQL. [9]

1.2.2 PHP

PHP (původně *Personal Home Page* nyní *Hypertext Preprocessor*) je skriptovací programovací jazyk, který byl vyvinut pro dynamické webové aplikace. Jazyk PHP lze zařadit do struktury jazyka HTML, XHTML, či WML, která se vyznačuje schopností tvorby internetových rozhraní či konzolových aplikací.

Jako tvůrce PHP se označuje Rasmus Lerdorf, který vytvořil v roce 1994 jednoduchý systém pro webové stránky. Sada jeho použitých skriptů byla ještě téhož roku vydána pod názvem Personal Home Page Tools, zkráceně PHP. Velké obliby se tento jazyk dočkal především ve své druhé verzi, obohacené především o snadnou manipulaci s daty z formulářů - PHP/FI 2.0 FI (*Form Interpreter*). Od roku 1998 byla k dispozici verze 3.0, oproti předchozím podstatně zrychlená a obohacená o podporu mnoha databázových systémů, objektů, cookies¹¹ atd. Tou dobou už nešlo mluvit o PHP jako o nástroji pro vylepšení webových stránek, ale jako o komplexním programovém jazyku pro dynamické aplikace. Možná právě proto nachází PHP velké využití na zpravodajských serverů či v archívech softwaru. V roce 2000 se objevila verze 4.0, která dosáhla skutečně masového rozšíření. K novinkám této verze patřilo jádro Zend a kromě zvýšení výkonu se změny týkaly samostatné činnosti PHP na serveru, zlepšila se architektura PHP, jeho šířitelnost a stabilita. Jazyk začal podporovat FTP a používat protokol http, jazyky Java, XML. V roce 2004 byla vydána zatím poslední verze 5 (aktuální verze je 5.2.7), obsahující velmi podstatné změny, především byl kompletně přepracován objektový model jazyka,

¹⁰Bioinformatika - vědní disciplína, zabývající se metodami shromažďování, analýzy a vizualizací rozsáhlých biologických dat.

¹¹Cookies - informace uložené v textovém dokumentu, slouží k rozlišování uživatelů.

třídy a objekty jsou nedílnou součástí jazyka, nabízející téměř možnosti srovnatelné s objektově orientovanými jazyky. [2]

PHP kód je tvořen příkazy, které vykonávají určitou funkci a krok za krokem tak provádějí algoritmus k danému výsledku, což umožňuje vkládat daný kód přímo do zdrojového kódu stránek HTML. Struktura jazyka tvoří bloky, které obsahují souvislé příkazy.

1.2.3 AJAX

Jedna z velmi zajímavých technologií, která byla v posledních letech vyvinuta se jmenuje AJAX (*Asynchronous JavaScript and XML*). O jeho velkou popularitu a rozšíření se postarala především společnost Google a její služby (nejdříve Gmail posléze Google Maps¹²). AJAX byl vytvořen pro vývoj dynamických webových aplikací, který dokáže měnit obsah stránek bez nutnosti zpětného načítání a překreslení celé stránky při každé operaci.

Tyto aplikace jsou vyvíjeny s pomocí těchto technologií:

- HTML (XHTML) a CSS - pro publikování informací
- DOM a JavaScript - zobrazování a dynamické změny publikovaných informací
- XMLHttpRequest - pro výměnu dat s webovým serverem

Většina uživatelských požadavků v této aplikaci probíhá na pozadí a po splnění požadavku se překreslí na stránce uživatele jen ty oblasti, se kterými uživatel pracuje naopak u klasické aplikace se s každým požadavkem musí uživateli poslat celý kód stránky. Další výhodou AJAXu je jeho nenáročnost na datové přenosy. Jeho velkou nevýhodou se jeví použití moderních webových prohlížečů.

1.2.4 SQL

Jazyk SQL (*Structured Query Language*) je dotazovací jazyk standartizovaný pro práci s daty v relačních databázích. K jeho nejzajímavějším vlastnostem patří především jeho deklarativnost - lze ho vložit do jiného programovacího jazyka, který je již procedurální. Se samotným SQL jazykem lze pracovat pouze po připojení terminálem na SQL server a přes zadávanými příkazy tohoto jazyka.

Syntaxi jazyka SQL tvoří položky (příkazy), které jsou pilířem jakéhokoliv skriptu. První syntax udává obecnou představu o jaký typ skriptu se bude jednat. Mezi základní příkazy patří SELECT a INSERT nebo CREATE. Jazyk SQL vyžaduje skript ukončit středníkem jako většina programovacích jazyků.

¹²Google Maps - bezplatná mapovací aplikace od společnosti Google, která nabízí vysoké-rozlišení satelitních snímků, často slouží jako navigace nebo jako plánovač cyklistických tras.

SQL se využívá pro komunikaci v multiplatformní databázi MySQL. Pro MySQL je charakteristická snadná implementace na operační systém počítače (Linux, MS Windows), velká výkonnost a především to, se jedná o volně šířitelný software. Kombinace MySQL, PHP a Apache jako základní software webového serveru se stala velmi oblíbenou mezi uživateli. [4]

1.3 Zabezpečení webových aplikací

Webové aplikace pracují nebo sdílejí často velmi důvěrná data o lidech, kteří aplikace používají. Ať už to jsou jména uživatelů nebo jejich emailové adresy, zneužití těchto informací nebývá příjemné, proto by samotná bezpečnost těchto aplikací neměla být podceňována. Je nutné si uvědomit, že bez dostatečného zabezpečení jsou důvěrná data vystavována riziku nejen zneužití osobních informací uživatelů třetími osobami, ale také znevažení samotné aplikace.

Každý systém i každá aplikace může čelit útoku na mnoha úrovních: od operačního systému serveru až po prohlížeč uživatele. Nejčastější je však napadena během interakcí webové aplikace s prohlížečem. Mezi nejznámější útoky, které jsou vedeny proti webovým aplikacím patří: [11]

1. **Cross Site Scripting (XSS)** - metoda narušení webových stránek, využívá bezpečnostní chyby ve skriptech, díky kterým dokáže do stránek injektuje (umístit) vlastní kód a tím poškodit vzhled nebo funkčnost stránky. Nejúčinnější obranou na straně serveru je použít odfiltrování nebezpečných HTML entit a na straně uživatele vypnout JavaScript.
2. **SQL Injection** - metoda, napadající databázovou vrstvu, která ukrývá způsob podvržení vstupních dat (dat proměnných odesílaných na server), tak aby v konečném důsledku byl pozměněn výsledek SQL dotazu. Nejčastěji je útok prováděn přes neošetřený formulář, manipulací s URL nebo i podstrčením cookies. Nejlepší obranou je na straně aplikace používání při skriptování co nejméně informací o struktuře databáze a tabulek a na straně databáze vhodným nastavením přístupujícím uživatelům.
3. **Podstrčení proměnných** - specifický druh útoku, týkající se především jazyka PHP, který využívá bezpečnostní chyby díky používání direktivy `register_globals` (tato direktiva automaticky převádí všechny hodnoty získané ze superglobálních proměnných - např. GET a POST data, cookies na globální proměnné pod určitým názvem. Velmi účinná obrana je vypnutí direktivy (pokud je ta možnost) a nebo ošetřit důsledně inicializace proměnných na jejich výchozí hodnoty.

4. **Upload souborů** - tato metoda vychází z předpokladu, že většina aplikací umožňuje upload souborů, ale málo která kontroluje typ uploadovaného souboru. Pokud se uploaduje jiný obsah než aplikace čeká, může dojít ke spuštění libovolného kódu.
5. **Session Hijacking** - metoda útoku při kterém je ukradena SID¹³ oprávněnému uživateli, poté se přihlásí - nalogují na účet poškozeného. Velmi často se stávají i administrátoři terčem těchto útoků.
6. **Session Fixation** - jedná se o útok kdy útočník klientovi přednastaví určitou hodnotu Session ID (podstrčením odkazu, v emailu apod.) a jakmile klient zareaguje - přihlásí se, tak tuto Session ID použije pro vlastní účely. Obrana proti tomuto typu útoku je jednoduchá - stačí vytvořit funkci, která způsobí změnu Session pro každého klienta.

Nemělo by se zapomínat i na zabezpečení samotných hesel uživatelů, pod kterými se přihlašují do databáze/aplikace. Jednou ze závažných chyb je ukládat tyto hesla bez jakéhokoliv kódování tak jak je uživatelé zadali. V tomto případě hrozí velké riziko hromadného získání uživatelských údajů. Je vhodné místo celého hesla ukládat pouze jeho hash¹⁴zadaného řetězce s hashem uloženým. Hashovací funkce převedou jakýkoliv řetězec na hash pevné délky, ze kterého se původní řetězec nedá v rozumném čase zjistit zpátky.

Mezi nejznámější hashové funkce patří *sha1 (Secure Hash Algorithm)* nebo *md5 (Message-Digest algorithm)*. V dnešní době je lepší použít hashovací funkci *sha1*, protože funkce *md5* již není považována za bezpečnou, byly objeveny bezpečnostní trhliny (nalezení kolize - dvou různých zpráv vedoucích na stejnou hash). [12]

Další alternativou je tzv. **soleňý hash**. Funkce je podobná klasickým hash funkcím rozdíl je ale v přidání náhodného řetězce k heslu. V praxi se to poté projevuje takhle: uživatel zadá heslo, k němu se připojí náhodný řetězec tzv. *sůl*. Ze spojeného řetězce se vygeneruje hash a uloží do databáze. Při kontrole hesla se z databáze načte *sůl*, připojí se k heslu, které uživatel zadal, vygeneruje hash a porovná s heslem v databázi.

¹³SID = Session ID, identifikační číslo. Používá ho server k rozlišení uživatelů, dále se používá pro zabezpečení přístupu do členské sekce, emailové schránky apod. Session se uchovávají v cookies.

¹⁴Hash - tiger = hašovací funkce neboli metoda pro převod vstupních dat do malého čísla, které vytváří jeho otisk, slouží ke kontrole integrity dat, je důležitou součástí kryptografických systémů pro digitální podpisy.

2 NÁVRH PROJEKTU A JEHO REALIZACE

2.1 Definice problému a přesná specifikace zadání

Bylo zadáno vytvořit aplikaci - webové rozhraní, které bude schopno vzdálené správy telefonních přístrojů. Dále musí být tato aplikace schopná jednoduché katalogizace a zabezpečena proti vniknutí neautorizovanou osobou.

Ze specifikovaného zadání vyplývá, že pro splnění tohoto zadání bude nutné navrhnout a vytvořit aplikaci, jejímž základem se stane databáze, která bude sloužit jako centrální úložiště dat. Dále bude důležité navrhnout skripty v jazyce SQL pro práci s daty (vkládání, mazání, editace, import či export), a zkombinovat je se skripty v jazyce PHP a HTML, které zobrazí výsledky příkazů do webového prohlížeče. Pohyb v celé aplikaci bude omezen přístupovými rolemi. Tyto role se rozdělí na základě pokynů administrátora. Přístup do aplikace mají ti uživatelé, kteří vyplnili registrační formulář a obdrželi přidělené přístupové role. Jejich práce a samotná komunikace s databází je prováděna pomocí HTML formulářů.

Celá aplikace funguje na serveru formou spolupráce Apache, PHP a MySQL není nijak hardwarově nenáročná.

2.2 Analýza systému

Pro správnou funkčnost celé aplikace je nezbytné ujasnit si výslednou podobu (tj. přesné definování příkazů, které je schopna aplikace zpracovat). Pro lepší orientaci v aplikaci byl sestaven seznam aplikačních pravidel¹:

- Pracovat v daném systému může pouze zaregistrovaný uživatel.
- K registraci slouží vstupní formulář, po vyplnění je možné se přihlásit do systému.
- Registrovaný uživatel má možnost editovat svůj profil a po případě si i změnit přístupové heslo.
- Před všemi úkony se musí uživatel přihlásit do systému, není myšleno pro každý jeho úkon v systému.
- V systému existují tři typy přístupových práv: administrator(admin), uživatel(user), host(guest). Každá role má omezené funkce.

¹Seznam aplikačních pravidel - seznam požadavků na systém - co by výsledná aplikace měla obsahovat, jaké požadavky by měla umět splnit

- Admin na základě registrace je schopen přidělit přístupové právo uživateli.
- Uživatel je schopen vyhledat v záznamech informace podle daných parametrů.
- Uživatel může editovat a exportovat parametry telefonních přístrojů v dané databázi.
- Uživatel může editovat nastavení telefonního přístroje přímo v jeho konfiguračním prostředí.
- Aplikace dovoluje uživateli importovat nastavení z konfiguračního rozhraní VoIP přístroje, nastavení však musí být pouze ve formátu `.txt`.
- Aplikace dovoluje uživateli exportovat nastavení VoIP telefonu vytvořené v této aplikaci, nastavení je uloženo ve formátu `.txt`.
- Informace zadané do HTML formulářů budou ověřeny pomocí skriptu.

2.3 Návrh struktury

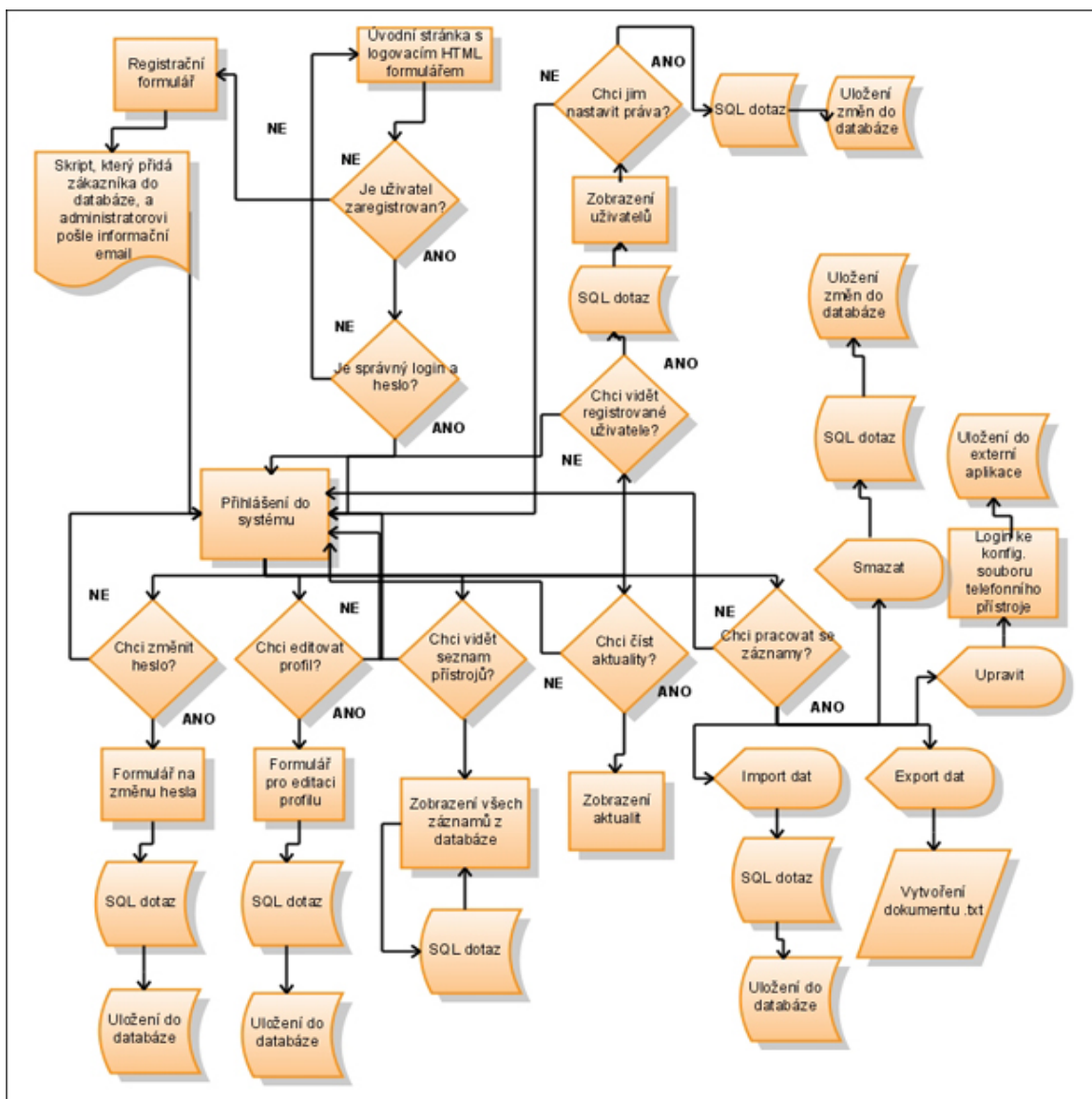
Po dokončení specifikace požadavků na aplikaci je třeba začít s návrhem datové struktury a tvorbě hypertextového modelu aplikace. Datová struktura vychází z analýzy uživatelských požadavků na aplikaci a jako výhodné se jeví její zobrazení do postupného vývojového diagramu. Na Obr.2.1 je zobrazen vývojový diagram pro přístupovou roli administrátor, ostatní vývojové diagramy pro roli user (Obr.A.1) a vývojový diagram pro roli guest (Obr.A.2) jsou k dispozici v příloze.

Z vývojového diagramu jasně vyplývá, že bez registrace či samotného nalogování nebude uživateli umožněna práce v aplikaci ani samotný přístup do aplikace. Teprve v případě úspěšného nalogování do aplikace se uživateli zobrazí menu, které ho odkáže na danou problematiku. Navigační menu je pro každou přístupovou roli specifické. Téma přístupové role v aplikaci je rozebráno na následujících stránkách.

2.3.1 Návrh databáze

Při návrhu samotné databáze se doporučuje zohlednit, zda má být pružná vůči novým požadavkům či nikoliv. V tomto daném případě je databáze navrhována a vytvořena tak, aby byla otevřená vůči novým požadavkům a zároveň neztrácela na výkonu - myšleno s ohledem na rychlost zpracování příkazu. [12]

Většina dotazů, která je kladena na databázi je optimalizována a navrhována tak aby tuto databázi zbytečně nezatěžovala. Jelikož výhledově není přesně definované



Obr. 2.1: Vývojový diagram pro oprávnění - administrator

kolik dat by měla databáze zvladát a spravovat nebylo nutné řešit otázku kešování² či defragmentaci dat.

Celá databáze je situována do 4 základních tabulek (Uživatel, Telefon, Oprávnění, Konfigurace). Do tabulky Uživatel budou zaznamenána data a údaje o uživateli, pracujících s touto aplikací, do tabulky Telefon se budou ukládat data o daném telefonním přístroji, tabulka Oprávnění slouží k zaznamenání přístupových rolí uživatele (zda se jedná o uživatele či o administrátora), poslední jmenovaná tabulka Konfigurace slouží ke sledování konfigurací - editací parametrů telefonních přístrojů.

Obecně se dá databáze rozdělit i na aplikační objekty (informace, které se chtějí v databázi uchovávat):

1. Registrovaní uživatelé
2. Telefonní přístroje

2.3.2 Systém správy relační databáze a normalizace relací

RDBMS neboli systém správy relační databáze usnadňuje a umožňuje snadno kombinovat hodnoty záznamů různých tabulek a tím násobí množství dotazů, které lze vytvářet nad danou databází. Aby relační model fungoval, je nutné tabulky zabezpečit spojenými záznamy³ (musí být relačně propojeny). Všechny operace nad relační databází vychází z matematické teorie relační algebry⁴, která např. vyvozuje, že tabulka nesmí obsahovat totožné řádky.

Pro správnost fungování relačního modelu je nutné brát na vědomí i vztahy mezi relacemi a zohlednit normalizaci⁵. V relačním modelu je možné stanovit tři druhy vazeb:

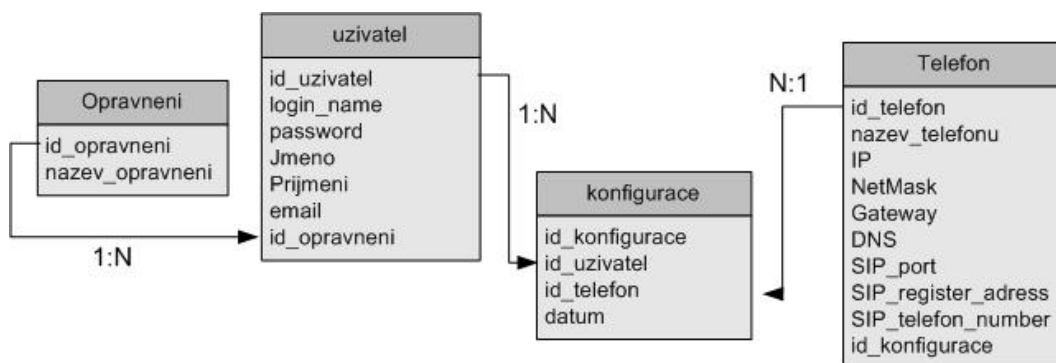
- 1:1 - jedné položce v tabulce odpovídá jedna položka v tabulce druhé
- 1:N - jedné položce v tabulce odpovídá N položek v tabulce druhé
- M:N - M položkám v tabulce odpovídá N položek v tabulce druhé

²Z anglického *cache* - označení pro vyrovnávací pamět. Cílem kešování je snížení zátěže hardware a zvýšení rychlosti odezvy

³Spojené záznamy představují dvě pole, která jsou uložena v různých tabulkách, ale mají stejnou hodnotu klíčů

⁴Relační algebra - využívá se pro práci v relační databázi, kde umožňuje zpracovávání dat v tabulkách. Mezi její základní operace patří: projekce, selekce a propojení. [13]

⁵Normalizace - sada pravidel, určující postup při transformaci struktury entit a relací ER modelu na strukturu fyzického uspořádání tabulek a relací v databázi.



Obr. 2.2: Screenshot Primární klíče a relační vztahy mezi jednotlivými tabulkami

V této databázi jsou použity pouze vztahy 1:N a to hned ve třech případech.

- Oprávnění - Uživatel (Jedno oprávnění může mít více uživatelů, tj. je možné aby v této aplikaci existovalo více uživatelů s oprávněním user)
- Uživatel - Konfigurace (Jeden uživatel může vytvořit více konfigurací, tj. aplikace dovoluje vytvořit uživateli vytvoření více konfigurací rozdílných jmen a parametrů)
- Konfigurace - Telefon (Jeden telefon může mít více konfigurací, tj. uživatel může jeden telefonní přístroj využít pro více konfigurací)

Po určení relačních vztahu je nutné dále zabezpečit jejich funkčnost, tedy modifikovat jednotlivé tabulky tak, aby splňovaly základní pravidla relačních vztahů:

1. Jedinečnost polí - každé pole by mělo představovat jedinečný typ informace.
2. Primární klíče - každá tabulka musí mít jednoznačný identifikátor, který je vytvořen z jednoho nebo více polí. Toto pole nesmí obsahovat hodnotu NULL.
3. Funkcionální závislosti - pro každou jedinečnou hodnotu primárního klíče se musí hodnoty v datových sloupcích týkat předmětu tabulky a musí tento předmět popisovat.
4. Nezávislost polí - data v libovolných polí se nesmí navzájem ovlivňovat.

2.4 Tvorba základního kódu a sestavení aplikace

Celý aplikace je rozdělena do třech velkých oblastí - přihlašování uživatelů, správa telefonních přístrojů a správa uživatelů. Všechny vytvořené skripty pro tuto aplikaci

byly naprogramovány pomocí programovacích jazyků SQL, PHP a HTML. V určitých částech byl použit i Javascript. Tyto skripty jsou k dispozici na přiloženém CD, které je součástí přílohy.

Pro lepší orientaci v aplikaci byla vytvořena i mapa aplikace, která je také součástí přílohy A.7.

2.4.1 Přihlašování

Tato sekce řeší otázku přihlašování uživatelů do systému. Pro přihlášení je nutné znát tzv. uživatelské jméno (login name) a heslo. Celé přihlášení do systému funguje



The screenshot shows a web page with a navigation bar at the top containing two links: [Registrace](#) and [Login](#). To the right of these links is an icon of a computer monitor and keyboard. Below the navigation bar is a text prompt: "Zadejte požadované údaje do tohoto formuláře:". This is followed by a form with two input fields: "Uživatelské jméno:" and "Heslo:". Below the "Heslo:" field is a button labeled "Přihlásit". At the bottom of the page, there are two links: [\[Aktuality\]](#) and [\[Přihlásit\]](#). Below these links is the text "Datum poslední aktualizace: 17/05/2009". In the bottom left corner, there is a section for the date and time: "Datum a čas:", "pátek", "22.5.2009", and "19:41". In the bottom right corner, there is a section for the author: "Autor: [Veronika Bartoňová](#)" and "[FEKT VUT Brno](#)".

Obr. 2.3: Screenshot Úvodní stránka

na principu srovnání údajů, které se nachází v HTML formulářích (viz Obr.2.3) a údajích zadaných v databázi na webovém serveru. Pokud nastane shoda systém povolí uživateli vstup do aplikace a zobrazí mu základní nabídku příkazů. V případě nesouhlasu se vrátí uživateli chybová hláška. O bezpečnost hesel se v databázi postarala kódovací funkce md5, která hesla před uložením zakódovala do jednosměrného textového řetězce.

Může však nastat situace, kdy uživatel dosud nezaregistrován bude chtít pracovat v této aplikaci. Pro takové případy byla vytvořena na vstupní stránce sekce pro registraci nového uživatele (Registrace - viz. Obr.2.4). Po vyplnění daných HTML formulářů kde je nutné zadat příjmení a jméno uživatele, jeho uživatelské jméno (login name) a heslo, které je nutné před odesláním všech informací potvrdit. Skript, který byl vytvořen pro vkládání nového uživatele nejdříve ověří zda vyplněný HTML formulář je správně vyplněný (zda email obsahuje znak @ apod.) a poté uloží nového uživatele do databáze. Automaticky s uložením se vygeneruje a odešle email, určený pro administrátora, který oznamuje založení nového uživatele do aplikace a požaduje této osobě přidělit přístupovou roli. Registrovaný uživatel bude mít právo hodnoty nula, jež bude změněno až po přidělení role administrátorem.

[Registrace](#) [Login](#)

Zadejte příslušné údaje do následujícího formuláře:

Jméno:

Příjmení:

Email:

Uživatelské jméno

Heslo:

Potvrďte heslo:

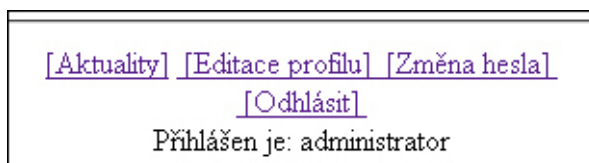
Pro práci v této aplikaci je nutné se zaregistrovat.

Obr. 2.4: Screenshot HTML formulář Registrace

2.4.2 Správa uživatelů

V této sekce se upravují data související s uživatelem aplikace jako jsou příjmení, jméno, uživatelské jméno (login name), emailová adresa a přístupová hesla. Každý zaregistrovaný uživatel má možnost editace příjmení, jména a emailu (viz Obr.2.5).

Také je schopen nahradit stávající přístupové heslo za nové.⁶



Obr. 2.5: Screenshot Zápatí stránky s tlačítka určené pro správu uživatelů

Jak již bylo zmíněno v této aplikaci postavení jednotlivých uživatelů není stejné. Jsou vytvořeny tři přístupové role (viz Obr.2.7):

- Admin - role s největší prioritou, bez omezení. Role admin spravuje celou databázi, vytváří, edituje, importuje, exportuje a maže záznamy, nastavuje přístupové role pro ostatní uživatele (viz Obr.2.6)nebo záznamy o registrovaných uživatelích maže.
- User (uživatel) - tato role má také omezené možnosti s ohledem na základní menu. Role user je schopna editace, importu, exportu nebo mazání stávajících dat.

Příjmení	Jméno	Datum registrace	Kontaktní email	Oprávnění	
Zajíc	Karel	2009-04-21 20:25:39	zajic@seznam.cz	2	Změna práv Smazat
Pavel	Petr	2009-04-21 20:27:21	pavel.petr@seznam.cz	0	Změna práv Smazat
milanek	milan	2009-05-05 13:10:41	milan@milan.cz	3	Změna práv Smazat
iva	iva	2009-05-05 13:47:33	iva@iva.cz	0	Změna práv Smazat
administrator	administrator	2009-05-05 14:35:14	admin@admin.cz	2	Změna práv Smazat
monika	iva	2009-05-13 00:17:30	monika@monika.cz	4	Změna práv Smazat
marta	marta	2009-05-13 00:19:38	marta@marta.cz	0	Změna práv Smazat
masa	masa,	2009-05-13 17:22:38	masa@masa.cz	0	Změna práv Smazat
lili	lili	2009-05-14 19:37:58	lili@seznam.cz	0	Změna práv Smazat

Obr. 2.6: Screenshot Přehled registrovaných uživatelů s informací o jejich právech

- Guest (host) - role s nejnižší prioritou, má omezené přístupy k dalším funkcím, které tato aplikace nabízí. Role je vytvořena pro uživatele hledající pouze informace k okamžitému použití, uživatel obsazen v této roli nemá možnost vkládání nových informací nebo editace stávajících.

⁶Editace a nahrazení hesla se provádí na základě přepsání stávající informace v databázi u nahrazení hesla je to komplikovanější s ohledem na potvrzení nového hesla.

Správa uživatelů je založena na SQL dotazech s parametry INSERT(vkládání dat), UPDATE(aktualiza dat), DELETE (mazání dat), SELECT(vyběr dat).

Obr. 2.7: Screenshot Přehled přístupových rolí

2.4.3 Správa telefonních přístrojů

Správa telefonů řeší evidenci telefonních přístrojů, jejichž data nebo informace o nich jsou uložena v databázi. Veškerá evidence nebo správa probíhá pomocí HTML formulářů.

[Výpis přístrojů](#) [Přidání přístroje](#) [Zobrazit registrované uživatele](#) [Zobrazit seznam přístrojů](#)

Název telefonu	Netmask	Gateway	DNS	SIP port	SIP registr. adresa	SIP telefonní číslo	Defaultní protokol	Alter DNS	
VOIP PHONE V1.7.211.223	192.168.1.178	255.255.255.0	192.168.1.1	202.96.134.133	5060	147229	815	2	Export Import Upravit Smazat
VoIP v2	192.168.1.179	255.255.255.0	192.168.1.1	202.96.134.133	5060	147229	659	4	Export Import Upravit Smazat

[\[Aktuality\]](#) [\[Editace profilu\]](#) [\[Změna hesla\]](#)
[\[Odhlásit\]](#)
 Přihlášen je: administrator

Obr. 2.8: Screenshot Zobrazení evidence přístroje s funkčními tlačítky

Mezi evidenci přístrojů lze zařadit:

- Vkládání nových přístrojů - nové hodnoty, vložené do formulářů, prochází ověřovacím skriptem (zda jsou informace správně zadané). Na základě tohoto ověření následuje jejich vložení do databáze.
- Editace přístrojů - pracuje s hodnotami, které jsou uloženy v konfigurační aplikaci daného přístroje. Nutné je nejdříve se nalogovat do konfigurační aplikace a poté odeslat požadavek na změnu. Pokud se tento postup odehraje v jiném pořadí editace se nezdaří.
- Mazání přístrojů - upravuje mazací skript, který vybere požadovaný přístroj (položku ve výpisu přístrojů), označí ji a následně smaže. Pokud data budou smazána, nelze je znovu použít.
- Export dat - je funkce vytvořena za účelem vyexportování (uložení) konfigurace přístroje. Uživatel má možnost editovat libovolný přístroj a poté toto nastavení uložit do externího souboru. Všechna nastavení jsou ukládána do textového souboru s příponou `.txt`.
- Import dat - umožňuje nahrát data uložená v externím souboru do databáze. Funkce je navržena tak aby byla schopna nahrát pouze textové soubory ve formátu `.txt`, kde následně obsah souboru rozkrájí na textové řetězce a uloží do dané tabulky v databázi.

3 ZÁVĚR

V bakalářské práci jsem se zaměřila na problematiku webových rozhraní, její návrh, vývoj a zabezpečení jak po stránce teoretické tak i stránce praktické. Mou snahou bylo z rozebrané teorie vytvořit, ve zvoleném prostředí, dané webového rozhraní pro jeden konkrétní typ telefonního přístroje, které bude splňovat zadané požadavky.

Celý projekt byl realizován na softwaru XAMPP, PHPMyAdmin a tvorba kódu v jazycích PHP, SQL, HTML a JavaScript probíhala v editoru PSPad. Vycházela jsem z předpokladu nutnosti registrace a logování, aby bylo zamezeno přístupu neautorizovaným uživatelům do aplikace, a i odlišení jednotlivých uživatelů od sebe pomocí přístupových práv. S ohledem na velikost aplikace jsem vytvořila tři přístupové role (administrator, user a guest), s tím že role administratora je klíčová.

Aplikace byla navrhována tak aby bylo možné v budoucnu její rozšíření o další typy telefonních přístrojů. V tuto chvíli je rozdělena pouze do dvou částí - Správa samotných uživatelů a Správa telefonních přístrojů. V obou částech byl brán ohled na uživatelskou přívětivost a nenáročnost práce v aplikaci (veškeré příkazy se provádí pomocí HTML formulářů). Dále je možné v obou částech provádět celkovou administrativu dat (vkládání, editace, mazání, různé způsoby katalogizace). V sekci Správa telefonních přístrojů je tento výčet doplněn o export vytvořeného nastavení přístroje do textového souboru s příponou .txt, dále import nastavení z externího konfiguračního programu telefonního přístroje, a přímé propojení s konfiguračním programem daného telefonního přístroje. Sekce Správa uživatelů je rozšířena o možnost nastavování přístupových rolí.

Hlavním přínosem bakalářské práce je možnost zjednodušení práce v podobě zkrácení času při náročném nastavování VoIP telefonů. Před vytvořením této aplikace se museli složitě nastavovat jeden přístroj po druhém. Moje aplikace má tento problém pomoci vyřešit a umožnit nastavení více přístrojů najednou.

S ohledem na zadané požadavky, které byly stanoveny se mi podařilo vytvořit aplikaci, která je nejenom optimalizovaná ale i lehce spravovatelná. Po krátkém seznámení není problém pro uživatele ji začít plně využívat.

LITERATURA

- [1] KOŠEK, J. HP: Tvorba interaktivních internetových aplikací. Podrobný průvodce Praha : Grada Publishing,1998. 1. vyd. 490 s. ISBN 8071693731
- [2] NARAMORE, E. PHP5, MySQL, Apache: vytváříme webové aplikace Vyd. 1. Brno : Computer Press, 2006. 813 s. ISBN 80-251-1073-7
- [3] KOFLER, M. PHP 5 a MySQL 5 : průvodce webového programátora Vyd. 1. Brno : Computer Press, 2007. 607 s., ISBN: 978-80-251-1813-9
- [4] BRÁZDA, J. PHP 5 začínáme programovat Praha:Grada Publishing,2005. 244 s., ISBN: 80-247-1146-X
- [5] JUN, J. MySQL manuál [online] 2005 [cit. 2008-11-14]. Dostupný z URL: <<http://mm.gene.cz>>
- [6] SKŘIVAN, J. Databáze a jazyk SQL. Interval [online] 2000 [cit. 2008-11-20]. Dostupný z URL: <<http://interval.cz/clanky/databaze-a-jazyk-sql/>>
- [7] DELISLE, M. PhpMyAdmin,efektivní správa MySQL Brno:Zoner Press,2004. 264 s., ISBN: 80-86815-09-9
- [8] DOSTÁL, P. ; KŘÍŽ, J. Databázové systémy Brno : Akademické nakladatelství CERM, s.r.o., 2006. 110 s., ISBN: 80-214-3064-8
- [9] The Perl Directory [online] 2008 [cit. 2008-11-18]. Dostupný z URL: <<http://www.perl.org/>>
- [10] FERSCHMANN, P. Bezpečnost na webu - přehled webové aplikace Zdrojak : root [online] 2008 [cit. 2008-12-01]. Dostupný z URL: <<http://zdrojak.root.cz/clanky/prehled-utoku-na-webove-aplikace/>>
- [11] VRÁNA, J. Ukládání hesel [online] 2005 [cit. 2008-12-02]. Dostupný z URL: <<http://php.vrana.cz/ukladani-hesel.php>>
- [12] MAKOWSKI, L. MySQL 16 - tvorba databázového systému. Zive computer [online] 2002 [cit. 2008-12-10]. Dostupný z URL: <<http://www.zive.cz/Clanky/MySQL-16—tvorba-databazoveho-systemu/sc-3-a-108361/default.aspx>>
- [13] ŠIROKÝ, J. Informační systémy v dopravě a spojích - Relaceční algebra [online] 2007 [cit. 2009-21-05]. Dostupný z URL: <http://homen.vsb.cz/s1i95/ISVDAS/IS/IS_relace.htm>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

- AJAX Technologie pro vývoj webových aplikací – Asynchronous JavaScript and XML
- ASP Skriptovací platforma společnosti Microsoft určená pro dynamické zpracování webových stránek na straně serveru – Active Server Pages
- ASP.NET Nástupce ASP, součást .NET Frameworku firmy Microsoft pro tvorbu webových aplikací a služeb
- C++ Objektově orientovaný programovací jazyk
- CGI skripty Skript popisující, jak bude server zpracovávat požadavky klienta – Common Gateway Interface
- CSS Kaskádové styly pro tvorbu HTML stránek – Cascading style sheets
- DBI Databázové rozhraní pro Perl – Database Interface for Perl
- DOM Objektově orientovaná reprezentace XML nebo HTML dokumentu – Document object model
- FTP Protokol aplikační vrstvy určený pro přenos souborů mezi počítači – File transfer protocol
- GPL Licence pro svobodný software – General Public License
- HTML Jazyk určený pro tvorbu webových stránek – HyperText Markup Language
- HTTP Internetový protokol pro výměnu dokumentů ve formátu HTML – Hypertext Transfer Protocol
- KDE Desktopové prostředí pro Linux a další unixové operační systémy – K Desktop Environment
- LGPL Licence pro svobodný software zaměřená především na knihovny – Lesser General Public License
- MD5 Rodina hashovacích funkcí, která vytváří ze vstupních dat výstup (otisk) fixní délky – Message-Digest algorithm
- MPL Licence pro svobodný software – Mozilla Public License
- MySQL Databázový systém, která komunikuje pomocí jazyka SQL

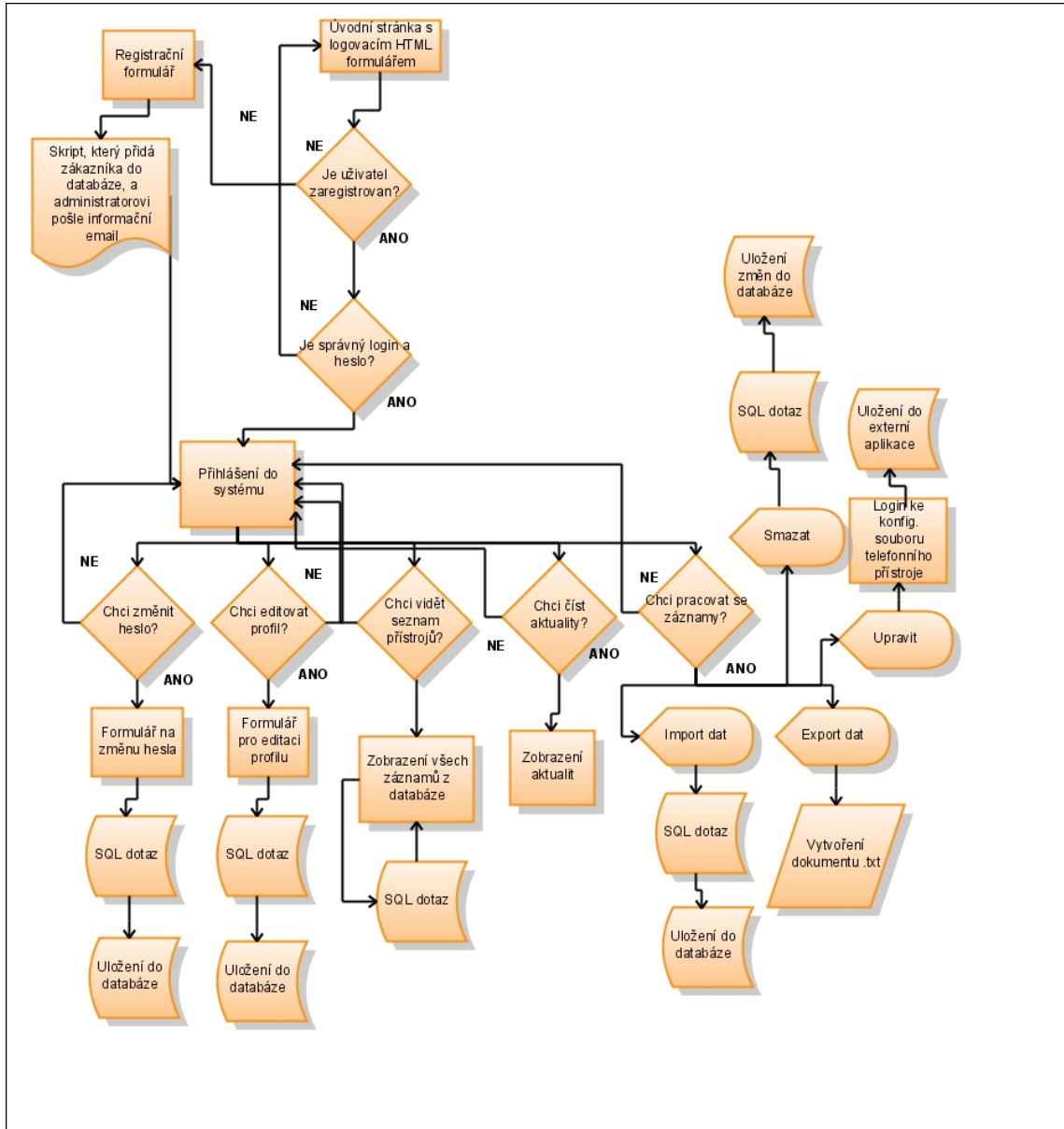
- ODBC Standardizované rozhraní pro přístup k databázovým systémům – Open database connectivity
- PHP Programovací jazyk pro tvorbu dynamických stránek – Hypertext Preprocessor
- QBE Dotazovací jazyk využívaný především v relačních databázích – Query by Example
- RDBMS Systém správy relační databáze – Relation database management system
- RSS Technologie umožňuje uživatelům Internetu čtení novinek na webových stránkách – Really Simple Syndication
- SID Identifikační číslo – Session ID
- SHA1 Hashovací funkce, která vytváří ze vstupních dat výstup (otisk) fixní délky – Secure Hash Algorithm
- SQL Dotazovací jazyk využívaný především v relačních databázích – Structured Query Language
- SŘBD nebo i DBMS Systém řízení báze dat stručně databázový systém – Database management system
- SSI Technologie pro vytváření dynamických webových stránek – Server Side Includes
- URL Řetězec znaků, který slouží ke specifikaci umístění zdrojů informací – Uniform Resource Locator
- VoIP Technologie, umožňující přenos digitalizovaného hlasu v těle paketů rodiny protokolů UDP/TCP/IP – Voice over Internet Protocol
- WWW Označení pro aplikace, které komunikují mezi sebou pomocí http protokolu – World Wide Web
- XML Jazyk určený pro výměnu dat mezi aplikacemi a pro publikování dokumentů – Extensible Markup Language
- XSS Metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech – Cross-site scripting
- XUL Formát pro tvorbu multiplatformního grafického rozhraní, který je používán v produktech Mozilla – XML User Interface Language

SEZNAM PŘÍLOH

A První příloha	37
A.1 Vývojový diagram pro oprávnění uživatel (user)	37
A.2 Vývojový diagram pro oprávnění host (guest)	38
A.3 Datové typy v jednotlivých tabulkách	39
A.4 Mapa aplikace	41
B Druhá příloha	42
B.1 Příložené médium (CD)	42

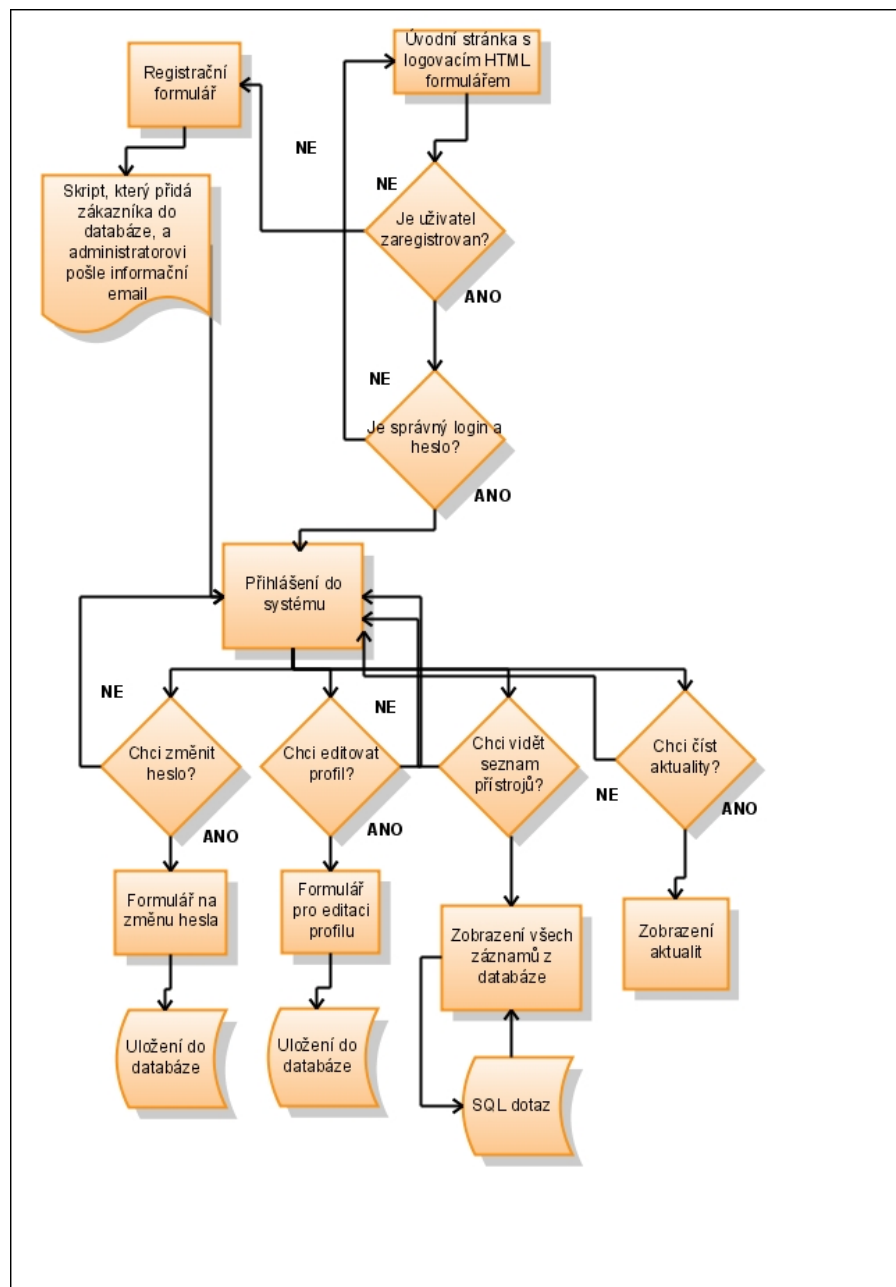
A PRVNÍ PŘÍLOHA

A.1 Vývojový diagram pro oprávnění uživatel (user)



Obr. A.1: Vývojový diagram pro oprávnění - guest

A.2 Vývojový diagram pro oprávnění host (guest)



Obr. A.2: Vývojový diagram pro oprávnění - guest

A.3 Datové typy v jednotlivých tabulkách

Sloupec	Typ
<u>id_uzivatel</u>	tinyint(4)
Prijmeni	varchar(30)
Jmeno	varchar(15)
login_name	varchar(30)
password1	varchar(60)
email	varchar(60)
datum	datetime
id_opravneni	tinyint(4)

Obr. A.3: Tabulka Uživatel

Sloupec	Typ
<u>id_konfigurace</u>	tinyint(4)
id_telefon	int(11)
id_uzivatel	int(11)
datum	datetime

Obr. A.4: Tabulka Konfigurace

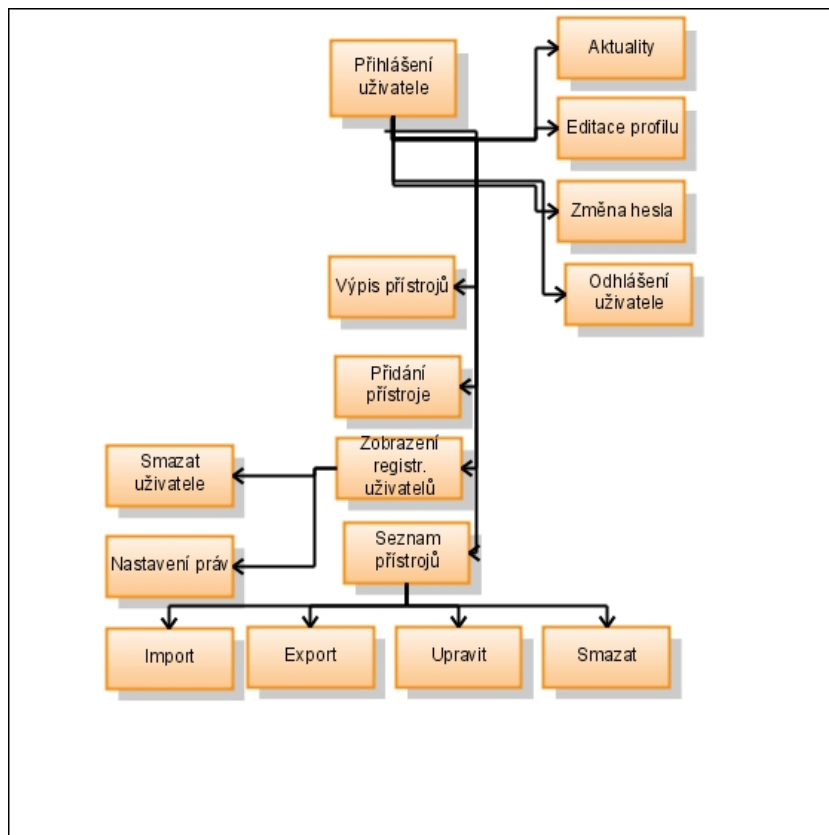
Sloupec	Typ
<u>id_opravneni</u>	tinyint(4)
nazev_opravneni	enum('admin', 'user', 'guest')

Obr. A.5: Tabulka Oprávnění

Sloupec	Typ
<u>id_telefon</u>	tinyint(4)
nazev_telefonu	varchar(30)
IP	varchar(15)
NetMask	varchar(15)
GateWay	varchar(15)
DNS	varchar(15)
SIP_port	int(11)
SIP_registr_adress	int(11)
SIP_telefon_number	int(11)
id_konfigurace	int(11)
Default_protocol	int(11)
alter_DNS	varchar(20)
DHCP_mode	tinyint(30)
DHCP_dns	tinyint(30)
Domain_name	varchar(30)
Host_name	varchar(30)
Pppoe_mode	tinyint(30)
HTL_Start_port	int(30)
HTL_Port_Number	int(30)
SNTP_server	varchar(30)
Enable_Sntp	int(30)

Obr. A.6: Tabulka Telefon

A.4 Mapa aplikace



Obr. A.7: Mapa aplikace

B DRUHÁ PŘÍLOHA

B.1 Přiložené médium (CD)

Na tomto CD jsou uloženy veškeré vytvořené skripty. Pro spuštění této aplikace je nutné mít nainstalovaný webový server, nebo alespoň mít zajištěný hosting, kde se bude tato aplikace spouštět, a MySQL, kam se nahraje vytvořená databáze, která je také uložena na tomto CD.

Důležité pokyny:

- Soubor, ve kterém je vyexportovaná databáze nese název `Bakalarska_prace_databaze.txt`
- Celá aplikace se spouští pomocí `index.php`
- Ve složce šablony jsou uloženy záhlaví a zápatí pro zobrazení horního a dolního menu použitého v aplikaci
- Přístupová hesla administrátora jsou: `login name: admin, heslo: admin`
- Přístupová hesla k konfiguračnímu souboru jsou tyto: `login name: root, heslo: (nevyplňuje se)`