

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

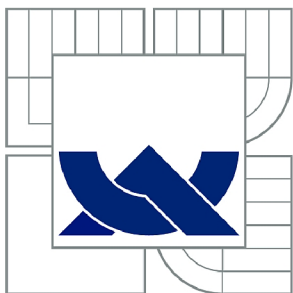
BRÁNA PRO ODESÍLÁNÍ SMS ZPRÁV

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

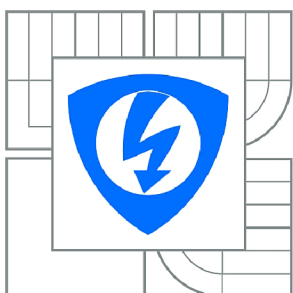
MAREK BIELCZYK

BRNO 2014



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

BRÁNA PRO ODESÍLÁNÍ SMS ZPRÁV

GATEWAY FOR SENDING OF SMS MESSAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

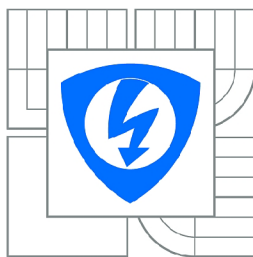
MAREK BIELCZYK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN KOUTNÝ, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Marek Bielczyk

ID: 145971

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Brána pro odesílání SMS zpráv

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte webový systém pro hromadné odesílání sms zpráv. Součástí systému bude jednoduchá administrace odeslaných zpráv, hierarchicky realizovaný uživatelský přístup a systém pro automatizované odesílání kontrolních SMS kódu a jejich zpětná validace. Systém zprovozněte v některém z dostupných PHP frameworku a realizujte aplikační rozhraní pro vzdálené užívání systému.

DOPORUČENÁ LITERATURA:

[1] LECKY-THOMPSON, Ed a Steven D NOWICKI. PHP 6: programujeme profesionálně. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně. ISBN 978-80-251-3127-5.

Termín zadání: 10.2.2014

Termín odevzdání: 4.6.2014

Vedoucí práce: Ing. Martin Koutný, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zaměřuje na kompletní problematiku SMS bran. První část práce je zaměřena na průzkum stávajících trendů a požadavků. Dle našich poznatků jsme navrhli vlastní projekt. Ten obsahuje hardwarovou část tedy volba GSM modulu, serveru a kompletní softwarovou část, která obsahovala práci s AT příkazy, php framework nette a MySQL. Naše řešení je možné aplikovat a přizpůsobit požadavkům uživatele. Projekt, který jsme vytvořili v rámci naší práce je multiplatformní.

KLÍČOVÁ SLOVA

SMS brána, textové zprávy, API, AT příkazy, odesílání, doručení, software, hardware, php, hypertextový procesor, Nette framework

ABSTRACT

The thesis is about SMS gateways. First part is focuses on the trends and requirements. After our experience we design own project. This section contains the hardware GSM module, a complete server software which included working with AT commands php framework Nette and MySQL. Our solutions can be applied and adapted by the user requirements. The project, which was developed within our work is ready for multiplatform using.

KEYWORDS

SMS gateway, text messages, send, receive, API, AT commands, delivery reports, software, hardware php, hypertext processor, Nette framework

BIELCZYK, Marek *BRÁNA PRO ODESÍLÁNÍ SMS ZPRÁV*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 40 s. Vedoucí práce byl Ing. Martin Koutný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „BRÁNA PRO ODESÍLÁNÍ SMS ZPRÁV“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Martinovy Koutnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	10
1 Analýza, návrh a výběr	11
1.1 Analýza stávajících řešení a trendů	11
1.1.1 Analýza jednotlivých projektů:	11
1.1.2 Shrnutí	13
1.2 Návrh projektu	14
1.3 Volba GSM modemu	17
1.3.1 Mezi srovnávanými byly tyto modely:	17
1.3.2 Zvolený model	18
1.4 Volba softwaru	18
1.4.1 Mezi srovnávanými byl tento software:	18
1.4.2 Zvolený software	19
2 Výsledky studentské práce	20
2.1 Schéma projektu	20
2.1.1 Model	20
2.1.2 Templates	20
2.1.3 Components	22
2.1.4 Presentery	22
2.1.5 Modul JSON RPC 2	24
2.1.6 Externí programy	25
2.1.7 Bootstrap	25
2.2 Databáze	25
2.3 Projekt	27
2.3.1 Funkce webové aplikace	27
2.3.2 Jak funguje API	28
2.3.3 Jak funguje odesílání zprávy	29
2.3.4 Práce s uloženými zprávami a funkčnost doručky	31
2.3.5 Jak funguje správa uživatelů	32
2.3.6 Jak funguje vyhledávání	34
2.3.7 Jak funguje adresář	35
2.3.8 Jak se vykresluje statistika	35
3 Závěr	37
Literatura	38

Seznam symbolů, veličin a zkratk	39
Seznam příloh	40
A Otestování aplikace	40
A.1 Server bez modemu	40
A.2 Server s modemem	40

SEZNAM OBRÁZKŮ

1.1	Blokové schéma.	14
2.1	Schéma projektu.	21
2.2	Schéma komunikace API.	24
2.3	Schéma databáze.	26
2.4	Schéma API.	29
2.5	Schéma odesílání zprávy.	30
2.6	Schéma vypisu uložených zpráv a spouštění doručky.	32
2.7	Schéma správy uživatelů.	33
2.8	Schéma vyhledávací funkce v smsPresenteru.	34
2.9	Schéma adresáře.	36

ÚVOD

V komunikaci je velmi důležitá rychlost a aktuálnost informace. Jedna z komunikačních metod je textová komunikace pomocí SMS zpráv. Využívá se často k autorizaci například v bankovníctví nebo v dalších webových službách. Práce se věnuje problematice SMS bran. Nejprve jsme provedli důkladnou analýzu situace a trendu v této oblasti, protože na trhu je k dispozici mnoho komerčních řešení. Některá jsou velice kvalitně zpracována jak z hlediska technického tak uživatelského (user experience) UX. Současně jsme při průzkumu aktuálních trendů objevili, že spousta provozovatelů, využívá k funkčnosti aplikační rozhraní (dále jen API). Klíčová je i volba správného GSM modemu, který by měl podporovat ovládání přes AT příkazy. Požadavkem z hardwarového pohledu byla, aby GSM modul komunikoval přes COM rozhraní. Tímto požadavkem jsme si výrazně omezili výběr. Rozhraní COM mají GSM moduly, které jsou často označovány jako tzv. průmyslové.

Dalším krokem je důležitá softwarová realizace. V softwarové části jsme se snažili, aby byli použit software s otevřeným zdrojovým kódem. Tudíž i samotný operační systém na serveru který obstarává komunikaci s GSM modulem je linuxový a další prvky jsme realizovali pomocí neustále rozvíjejícího se programovacího jazyka PHP, nyní ve verzi PHP5. Pro zjednodušení práce využijeme rozšíření známé jako Nette framework a MySQL databázi. Uživatelské rozhraní bude vytvořeno ve webových technologiích a tímto jsme docílili toho, že naše SMS brána bude multiplatformní.

Na trhu není ucelené řešení, které by nabídlo komplexní řešení jak z pohledu hardwaru i softwaru. Proto jsem se o tuto problematiku začal intenzivně zajímat a věnovat se jí. Cílem práce je vytvořit SMS bránu, která bude finančně nenáročná a zároveň bude velmi jednoduchá, kompatibilní s jednoduchým intuitivním uživatelským rozhraním.

1 ANALÝZA, NÁVRH A VÝBĚR

Tato část se věnuje celé problematice v technické rovině. Bylo vytvořeno blokové schéma, které rozebírá jednotlivé prvky, jenž jsou v očích uživatele brány jako jednotný celek. Toto schéma obsahuje základní klíčové prvky a kroky, díky kterým je zajištěna funkčnost SMS brány. V blokovém schéma je názorně ukázáno jak vše vzájemně komunikuje.

1.1 Analýza stávajících řešení a trendů

Na internetu jsou k nalezení projekty, které umožňují uživatelům posílat ať už jednu nebo více sms zpráv prostřednictvím GSM sítě. Přitom nám většina poskytovatelů těchto služeb nabízí různé funkce navíc. Ty placené bývají technologicky pokročilejší a nabízí ke službě spoustu funkcí, i možnost využít jejich služeb na jiném webu. Oproti tomu neplacené weby mívají spoustu chyb, nemají ošetřený vstup a často jsou až přehlceny reklamami. Několik takových různých projektů jsem sám vyzkoušel a zhodnotil.

1.1.1 Analýza jednotlivých projektů:

1. sms.sluzba.cz

Klady

- Pěkně zpracovaný web
- Přehledné menu na horní straně stránky
- 20,- kredit za registraci
- Garance spolehlivosti i při odesílání do zahraničí
- Informační texty lze schovat
- Možnost odeslat zprávu až druhý den
- Na pravé straně zprávy je umístěn adresář
- Do adresáře lze kontakty importovat
- Kontakty lze dělit do skupin
- Délka zprávy až 459 znaků

Zápory

- Špatně fungující výběr kontaktů z adresáře
- Výpis chyb - vrací jen jedno hlášení
- Na příchozí zprávy nelze odpovědět rovněž je nelze preposílat na jiná čísla
- Ve špičce doba doručení sms byla 19,65 sekund

- Ze zprávy je automaticky odstraněná diakritika
- Odeslané zprávy jsou v přehledné tabulce a lze je filtrovat
- Opětné odeslání již odeslané zprávy
- Exportování zpráv do excelu (.xls)
- Příjem příchozích zpráv
- Odesílání zpráv pomocí e-mailu
- Mimo špičku rychlé doručení zprávy

2. smsmanager.cz

Klady

- Pěkně zpracovaný web
- Přehledné menu na horní straně stránky
- Na první stránce je adresář, který lze filtrovat nebo přisazovat štítky ke kontaktům
- Štítky si uživatel nadefinuje sám
- Kontakty lze importovat pomocí připraveného pole
- Ve zprávě je možné odesílat zprávu na více čísel současně
- Výběrem štítku vybereme celou skupinu kontaktů
- Napsanou zprávu lze odeslat kdykoliv dle výběru data a času
- Zprávy jsou rozděleny do tří tarifů
- Po odeslání zprávy lze provést další akci
- Přijímané zprávy podle klíčového slova jsou přiřazeny k uživateli k provedení vybrané akce
- Přijaté zprávy chodí v pořádku a jsou přisazené k poslední odeslané
- Přijaté zprávy lze filtrovat
- Statistiky zobrazí uživateli počet odeslaných zpráv za určitou dobu

Zápory

- Nutnost ověření e-mail adresy
- 10,- kredit za vložení a potvrzení čísla
- Nepříliš funkční import kontaktů
- Do čísla příjemce je možné vložit písmena, což stránku vyhodnotí chybou o nedostatku kreditu
- Při výběru staršího data pro odeslání stránky zprávu přijmou a čekají až vybraná doba nastane
- První dva tarify pro zprávu jsou za stejnou cenu
- Zkoušená akce po odeslání zprávy nefungovala
- Ve špičce odeslání zprávy trvá až 9,14 sekund

- Historie ukazuje odeslané zprávy spolu s doručenkami
- Doručení zprávy mimo špičku během tří sekund

3. smsmidelet.com

Web vypadá na pohled pěkně, po registraci docela dlouhá doba než dorazila registrační sms. Po dokončení registrace jsem se pokusil přihlásit, ze stránek přišla odpověď „Došlo k chybě při přihlášení do systému!“. Tento stav se za dva týdny nezměnil. Při pokusu o novou registraci stránky odpověděly, že číslo je již přiřazeno, po zaregistrování jiného čísla, a pokusu se přihlásit se na nový účet opět došlo k chybě.

4. smsmania.cz

Klady

- Na úvodní straně formulář pro zprávu
- Číslo příjemce ošetřené proti znakům a omezené na 9 číslic
- V adresáři lze filtrovat, upravovat a mazat kontakty
- Odeslané zprávy lze filtrovat
- Stránky nabízejí zablokování nebo odblokování telefonního čísla
- Zprávy posílá zdarma
- Stránky jsou i v mobilní verzi

Zápory

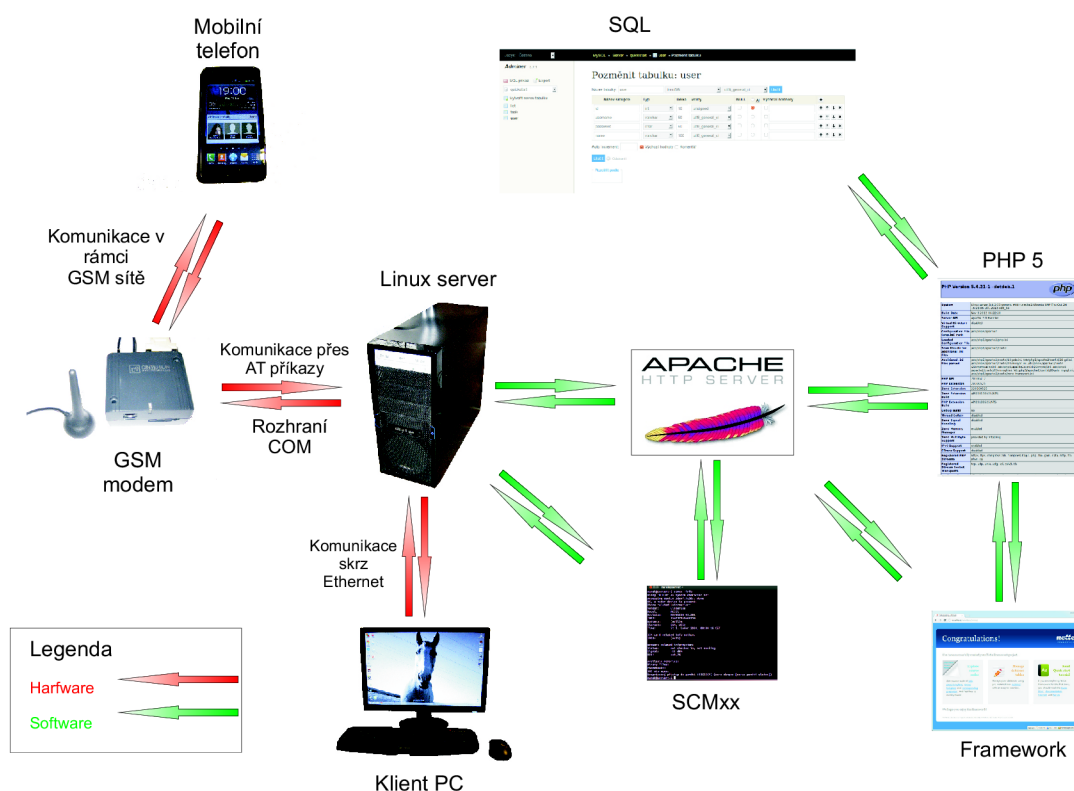
- Stránky jsou nepřehledné přepřávané reklamou
- Odesílá pouze na území České republiky
- Velikost zprávy je omezená na 160 znaků
- Zpráva není ošetřena proti interpunkci, což způsobuje chyby ve zprávě

1.1.2 Shrnutí

Mezi analyzovanými projekty se často vyskytovaly různé typy chyb, u neplacených projektů byl problém už se základní funkcí pro odeslání, kdy webová aplikace neodstranila interpunkci a uživateli tak dorazí zpráva nečitelná a nesrozumitelná. To se u placených webu neobjevilo, avšak jejich rozličné funkce obsahovaly chyby. U prvního analyzovaného webu se objevily problémy s výběrem kontaktů z adresáře. Dále byly všechny funkce ošetřeny pouze jedním chybovým hlášením, což může být matoucí. Druhý web trpěl na neošetřené naplánování zprávy, kdy webová aplikace přijala uběhlé datum. Funkce pro provedení činnosti po odeslání se taktéž neprovedla. Jiný projekt měl problémy například k zadanému číslu byl vložen znak, web odpověděl chybovým hlášením o nedostatku kreditu. Jinak analyzované projekty dispo-

valy rozličnými možnostmi a funkcemi, mezi které nejčastěji patří filtrování, adresář a historie odeslaných zpráv. Mezi velmi zajímavé funkce určitě patří odesílání jedné zprávy na více čísel, vytváření skupin čísel, na které chceme zprávu posílat. Možnost zprávu odeslat později, nebo ji později upravit a následně odeslat. Tyto funkce jsou však na každém projektu jinak koncipovány a některé z nich budeme hledat marně. Proto v této práci tyto funkce sjednotím.

1.2 Návrh projektu



Obr. 1.1: Blokové schéma.

Dle schéma je komunikace vyčleněna do několika kroků:

- Při startu webového serveru Apache2, jsou načteny knihovny PHP5 a Nette Framework
- Klientské PC vyvolá komunikaci pomocí webového prohlížeče odesláním požadavku na webový server Apache2, běžícím na linuxovém serveru.
- Přijatá data jsou následně zpracována serverem Apache2.
- Apache2 si ověří přístupové údaje v databázi MySQL.

- Apache2 spustí program SCMxx, ten pomocí AT příkazů odešle data do GSM modemu.
- Apache2 pak vrátí zpracovaná data uživateli, zapíše nové hodnoty do databáze a GSM modem odešle SMS příjemci
- Tímto je celý proces ukončen.

Telefon – zařízení, na které chceme zprávu poslat. Jedná se o jakýkoliv dnes použitelný mobilní telefon připojený do jedné ze sítí mobilních operátorů. Od tohoto přístroje požadujeme dekódování zprávy, zobrazení a následnou informaci o úspěšném doručení zprávy prostřednictvím doručenky.

GSM modem - zařízení, které spojuje náš aparát s přístroji pomocí GSM sítě určitého operátora. Výběr GSM modemu je velmi důležitý, každý nám nabídne jiné funkce a možnosti jak připojení, tak komunikace. V našem případě vybíráme zařízení, které nám umožní svobodně si vybrat, jaký ovládací program chceme, k tomu jsou nejvhodnější průmyslové GSM modemy, protože levné USB klíčenky lze ovládat většinou jen pomocí dodávaného softwaru. Nakonec jsme z velkého množství vybrali modely uvedené v další kapitole. Nejlepší volbou byl GSM modem od výrobce Siemens, a to díky otevřenosti a snadné komunikaci pomocí AT příkazů, model MC 55i, který se připojuje k PC přes rozhraní COM.

Server – počítač, na kterém běží Operační systém LINUX Ubuntu12.04 LTS. Linux jsme vybrali díky otevřenosti, snadnému nastavení komunikace s portem a snadné instalaci programů v podobě balíčků. Především však pro lepší zabezpečení. Verze Ubuntu je uživatelsky přívětivější a patří k jednomu z nejrozšířenějších operačních systémů, má velkou podporu ať už softwaru, tak i ze strany uživatelů. Číslo 12.04 LTS jsme zvolili, protože se jedná o verzi s dlouhodobou podporou (5 let) a neobsahuje žádné testovací balíčky, díky čemuž se může spolehnout na větší stabilitu.

SMCxx – terminálová aplikace určená především pro OS Linux, sloužící ke komunikaci s GSM modemem. Tato aplikace využívá AT instrukce, které jsou schovány mezi jednoduché příkazy. Takovýchto programů existuje velmi mnoho a vybrat mezi nimi, ten nejvhodnější není vůbec snadné. Do první skupiny tedy zařadíme ty určené pro systém Windows, tyto programy většinou obsahují i vlastní weby disponujícími svými webovými emulátory. Druhá skupina je tedy spíše linuxová (ale většinu z nich je možné nainstalovat i na Windows), aplikace tohoto typu jsou převážně ovladatelné přes terminál (nebo příkazový řádek), kdy po instalaci zadáme název programu a příkazy, které chceme provést. Pro většinu těchto aplikací existuje i jejich grafická aplikace, pro naši práci však stačí možnost zadávat příkazy dle potřeby, proto tuto grafickou část nevyužijeme. Při výběru jsme uvažovali i námi vybraný GSM modem. Nejvhodnějším programem pro naši práci je SCMxx s podporou pro výrobky firmy

Siemens.

Apache 2 – je softwarový webový server s otevřeným kódem pro různé platformy. Dnes již právě apache dodává prohlížečům většinu internetových stránek na celém světě. Mezi jeho přednosti patří především to, že je výkonný, spolehlivý, má ohromné konfigurační možnosti, modul PHP umožňuje dynamické změny obsahu v závislosti na návštěvníkovi a hlavně je zdarma a vyvíjí ho mezinárodní komunita vývojářů. Apache se řídí konfiguračními soubory, mezi nejdůležitější patří soubor httpd.conf, po změně tohoto souboru je nutný restart tohoto programu. Mezi další užitečné funkce patří výpis pokusů o připojení a chybové hlášení to najdeme v souborech access.log a error.log.

PHP5 – hypertextový procesor (původně „Personal Home Page“), jedná se skriptovací programovací jazyk, který funguje pouze na serveru a jeho výstup je již čistě v html. Instaluje se jako rozšíření k webovému emulátoru, např. Apache. Podporuje objektové programování a umožňuje nám vytvářet velké a krásné weby díky své dynamičnosti. Jedná se o jeden z nejpoužívanějších skriptovacích jazyků používaných pro webové stránky. Podporuje spoustu knihoven a celé řady internetových protokolů. Vývoj tohoto jazyka započal již v roce 1994, tehdy byl napsaný v jazyku C a měl zajišťovat běh úloh jako zobrazení životopisu nebo zaznamenávání návštěvnosti stránek. Od roku 1995 již verze PHP2 dovolila pracovat s databázemi, což umožnilo tvorbu prvních jednoduchých dynamických webových aplikací. Vývoj neustále pokračuje a nyní už máme PHP verze 5.

SQL (Structured Query Language) – pro práci s databází je založen na relační algebře a relací množin. Základem je tabulkové uspořádání dat. Každá množina dat je uložena v určité tabulce a vztahy mezi tabulkami jsou určeny relační vazbou. Relační vazba má několik různých typů např (0:1 – kdy první tabulka může mít jen jednu vazbu na záznam v druhé tabulce, obdobně: 1:1 kdy první tabulka musí obsahovat záznam vázaný na jen jeden záznam v tabulce druhé, dále jsou typy 0:N, 1:N, M:N – kdy druhá tabulka může nebo musí obsahovat více než jeden záznam vázaný na první tabulku. Dále se jazyk SQL dělí na dvě hlavní větve DDL a DML. Kdy DDL (Data Definition Language) pomocí příkazů definuje jednotlivé objekty v databázi (tabulky, procedury, view apod.). DML (Data Modification Language), kdy pomocí příkazů, definujeme práci s databázovými objekty (výběr dat, modifikaci dat, mazání dat apod.). K jeho použití potřebujeme jeden z databázových systémů např (MySQL, Oracle, PostgreSQL, Firebird, IBM DB/2 aj.) Z nich má však největší zastoupení MySQL, které dnes běží skoro na každém webovém serveru a je hojně v tomto podporováno v php, které z něj vytváří normální text na webových stránkách.

[1]

Framework – je to softwarová struktura, která slouží jako podpora při progra-

mování a vývoji softwarových projektů. My se budeme bavit o PHP frameworkcích, tím myslíme to, že se bavíme o sadě scriptů v PHP. To znamená, že se nejedná o žádné kompilované knihovny, které bychom do našeho kódu přidávali, naopak právě frameworky můžou nahradit nějakou extension z PHP. Můžeme tak vyžrát na nekvalitní hostingy, které nám nenabízejí to, co potřebujeme. Trochu oříškem, však zůstává jaký PHP framework zvolit. Prvním kritériem je podpora verze PHP, dalším důležitým faktorem při výběru je to, k čemu framework potřebujeme a k čemu byl určen. Nejlépe je volit ty známější, které mají dobrou velkou komunitu, u kterých vývoj za týden neskončí. Mezi ně patří např. Zend Framework, Nette Framework, Symfony, CakePHP, Codegniter. Z nich jsme vybrali Nette Framework, který má českou dokumentaci, aktivní komunitu v ČR, ladící nástroje a není tak těžkopádný. [2]

Klient – počítač připojený do sítě, obsahující webový prohlížeč a využívající službu, kterou mu poskytuje server.

1.3 Volba GSM modemu

GSM modemy byly vybírány podle několika kritérií. Jedním z nejdůležitějších byla otevřenost AT příkazů a dostupné dokumentace výrobce, protože bude použit jiný vyhovující software, dalším kritériem byly možnosti připojení k počítači neboli počet a různorodost vstupů zde byly pouze možnosti USB, COM a nebo USB i COM, přičemž port COM nám umožnil větší kontrolu nad modemem, dále nás zajímalo pásmo na, kterém modem dokaže vysílat a přijímat a to hlavně pásmo 900 / 1800 MHz, které se používá v České republice. Posledním kritériem, ale také velice důležitým, byly názory a doporučení uživatelů, kteří již GSM modem používají.

1.3.1 Mezi srovnávanými byly tyto modely:

1. **Maestro 100evo Advanced** [4]
 - Quad Band 850 / 900 / 1800 / 1900 MHz
 - 1 Digitální a 1 Analogový vstup
 - Otevřený pro AT příkazy
 - Možnost vzdálené konfigurace.
 - Cena 5112,46 Kč
2. **GSM/GPRS RS232 Modem Dual Band** [5]
 - Dual Band 900 / 1800MHz
 - 1 Analogový vstup
 - Otevřený pro AT příkazy
 - Cena 1 755,92 Kč

3. **Cinterion (Siemens) MC55i** [6]
 - Quad Band 850 / 900 / 1800 / 1900 MHz
 - 1 Analogový vstup
 - Otevřený pro AT příkazy
 - Cena 2850,76 Kč
4. **AirLink Fastrack Xtend Modem** [7]
 - Quad Band 850 / 900 / 1800 / 1900 MHz
 - 1 Digitalní vstup
 - Otevřený pro AT příkazy
 - Plně programovatelný ARM9 procesor pomocí C/C++ a Lua.
 - Cena 4 754,11 Kč

1.3.2 Zvolený model

Po vyhodnocení kritérií a doporučení různých uživatelů na fórech byl zvolen GSM modem Cinterion MC55i. Tento model se připojuje přes rozhraní COM, dokumentace je dostupná jak na stránkách výrobce, tak i na stránce prodejce. Je podporován většinou softwarových aplikací.

1.4 Volba softwaru

K řízení GSM modemu se používají AT příkazy, jejich masové rozšíření však způsobilo, že lze využít aplikace, které prací s těmito příkazy zjednodušují. Tyto aplikace jsou buď ovladatelné z příkazového řádku nebo z grafického prostředí. Z rozsáhlého množství se vybíralo podle kritérií jako je podpora vybraného GSM modemu, podpora operačního systému a flexibilita.

1.4.1 Mezi srovnávanými byl tento software:

1. **SMS server tools** [8]
 - Ovládání pomocí terminálu
 - Posílá SMS/WAP/MMS
 - Licence: Otevřený software
2. **SCMxx** [9]
 - Ovládání pomocí terminálu
 - Posílá SMS
 - Synchronizace se zařízením (datum, kalendář aj.)
 - Podpora systému Siemens
 - Licence: Otevřený software

3. **Gammu SMSD** [10]

- Ovládání pomocí terminálu nebo s rozšířením o grafické prostředí
- Posílá SMS
- Synchronizace se zařízením (kalendář, úkoly aj.)
- Podpora hlavně systému Nokia
- Licence: Otevřený software

4. **Diafaan SMS server** [11]

- Ovládání pomocí grafického prostředí
- Posílá SMS
- Obsahuje webovou aplikaci
- Pouze pro OS Windows
- Licence: Shareware

1.4.2 **Zvolený software**

Protože server běží na operačním systému Linux, přizpůsobil se k tomu výběr aplikace. Nakonec byla nejvhodnější volbou aplikace SCMxx, která je naprogramovaná převážně pro výrobky od společnosti Siemens. Ovládání aplikace se provádí v příkazovém řádku pomocí intuitivně volených příkazů. Aplikaci nebylo potřeba nijak konfigurovat po připojení modemu k portu COM si aplikace modem sama načetla.

2 VÝSLEDKY STUDENTSKÉ PRÁCE

2.1 Schéma projektu

2.1.1 Model

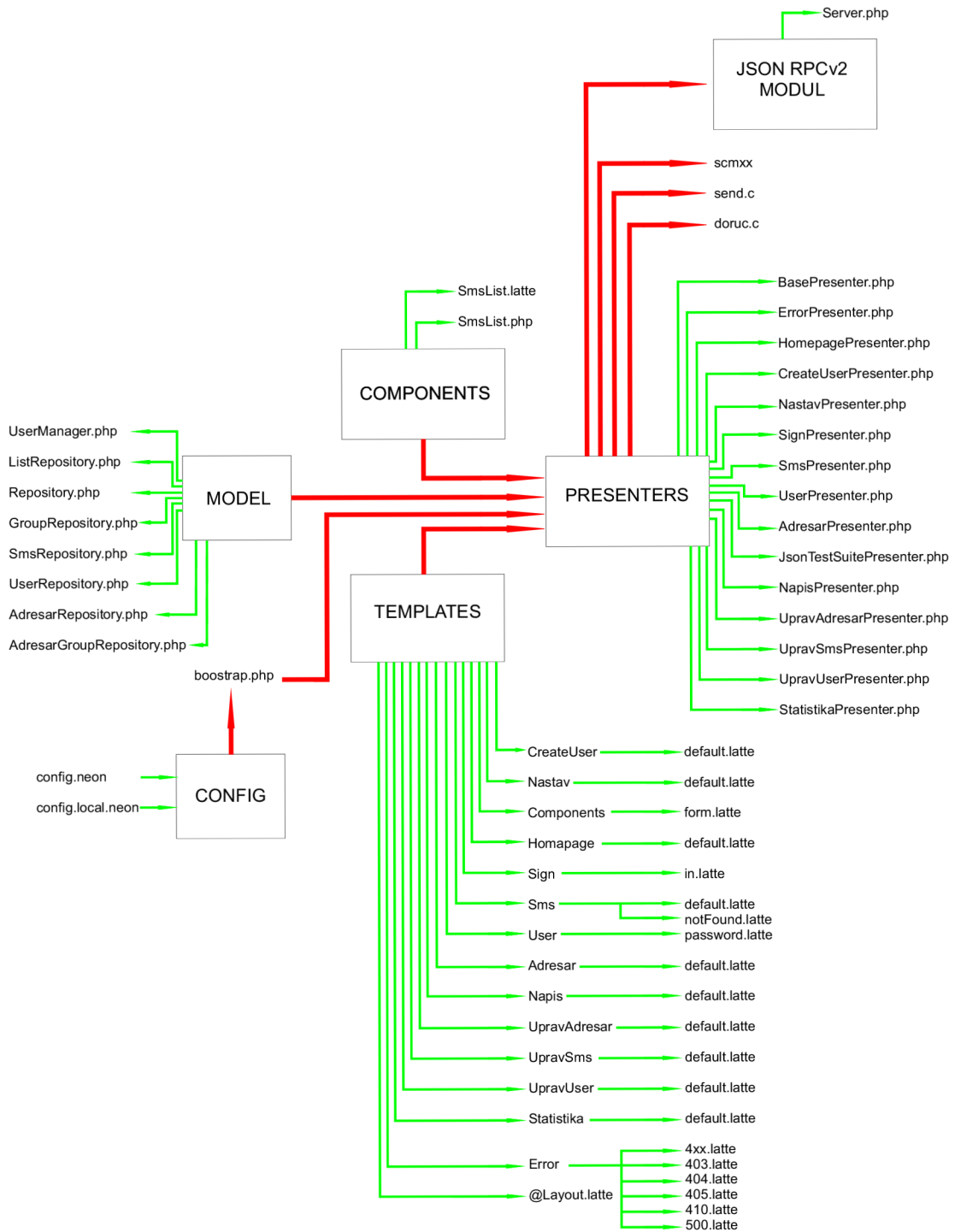
Datová a funkční vrstva celé aplikace, která se stará o ukládání dat a aplikační logiku. Jakoukoliv událost uživatele (např. přihlášení) představuje akci modelu. Ten má pevně dané rozhraní, pomocí kterého s ním ostatní části aplikace komunikují, sám o svém okolí nic neví.

- **UserManager.php** - Stará se o přihlašování uživatelů. Kontroluje, šifruje heslo a odpovídá chybovými hlášeními, vytváří nové uživatele. Před verzí nette 2.1 se jmenoval „Authenticator.php“.
- **ListRepository.php** - vypisuje a filtruje zprávy spadající pod daný list (např. Odeslat, Koncept).
- **Repository** - provádí operace nad databázovou tabulkou tj. vrací objekt jí reprezentující a vrací všechny řádky z tabulky nebo vrací řádky podle filtru.
- **SmsRepository** - vrací seznam nedoručených sms zpráv, a to buď celý nebo filtrovaný podle uživatele, vytváří, upravuje a odesílá sms zprávy, pak si načítá z výstupu programu send.c číslo message_reference.
- **UserRepository** - vyhledává uživatele podle jména nebo id. Mění hodnotu kreditu a stav odeslaných zpráv. Maže a upravuje uživatele a počítá statistiku.
- **GroupRepository** - vypisuje a filtruje záznamy uživatelů spadající pod danou skupinu a vytvoří nové skupiny.
- **AdresarRepository** - obsahuje funkce na vytvoření, smazání, upravu kontaktu a vyhledávání podle ID.
- **AdresarGroupRepository** - má funkce na vytvoření skupiny, přesunu kontaktu do dané skupiny a vyhledávání s filtrem v kontaktech.

2.1.2 Templates

Šablony představují v Nette pohled (View). Starají se o samotné vykreslení výsledku požadavku uživatele.

- **CreateUser -> default.latte** - vykreslí formuláře pro vytvoření nového uživatele a výběrové menu pro skupinu a vypíše seznam aktuálních uživatelů.
- **Nastav -> default.latte** - vykreslí nastavovací stránku GSM modemu pro administrátora.
- **Components -> form.latte** - vypíše odeslané a uložené sms zprávy všech uživatelů, používané pro kontrolu zpráv.



Obr. 2.1: Schéma projektu.

- **Homepage** -> **default.latte** - vykreslí nadpis, dále dva formuláře jeden pro číslo a druhý pro zprávu, výběrové menu a v levém sloupci adresář.
- **Sign** - > **in.latte** - vykreslí štítek pro (label) uživatelské jméno, heslo, checkbox a tlačítko přihlásit se a formuláře pro zadání jména, hesla a k zaškrtnutí tlačítka.
- **Sms** -> **default.latte** - vykreslí jednotlivé tabulky (jako je Odeslat) a vykreslí formulářový výběr pro smazání, odeslání nebo úpravy zprávy. Soubor `notFound.latte` slouží k výpisu chyby při pokusu otevřít neexistující tabulky.
- **User** -> **password.latte** - vykreslí formulář pro změnu hesla.
- **Adresar** -> **default.latte** - vykreslí formulář pro vytvoření kontaktu, vytvoření skupiny pro kontakty a vypis kontaktů s rolovacím menu, kde jsou možnosti ke smazání nebo úpravě kontaktu a přiřazení do jiné skupiny.
- **Napis** -> **default.latte** - složí k vykreslení formuláře pro odeslání zprávy správci.
- **UpravAdresar** -> **default.latte** - slouží k vykreslení formuláře pro úpravu kontaktů.
- **UpravSms** -> **default.latte** - slouží k vykreslení formuláře pro úpravu zpráv.
- **UpravUser** -> **default.latte** - vykreslí formulář pro úpravu uživatele.
- **Statistika** -> **default.latte** - slouží k vykreslení grafu statistiky odeslaných, koncepcních zpráv k poměru celkem zpráv.
- **Error** - složka obsahuje jednotlivé šablony pro vykreslování různých html chyb.
- **@layout.latte** - základní šablona, společná pro více presenterů. Vykresluje hlavičku a menu.

2.1.3 Components

Jsou základní kámen znovu použitelnosti kódu, neboli představují vykreslený objekt (např. formuláře, ankety) a je možné je do stránky opakovaně vkládat.

- **SmsList.php** - obsahuje funkci na kontrolu doručení, a funkci k vykreslení šablony.
- **SmsList.latte** - vykresluje tabulky pro šablonu sms.

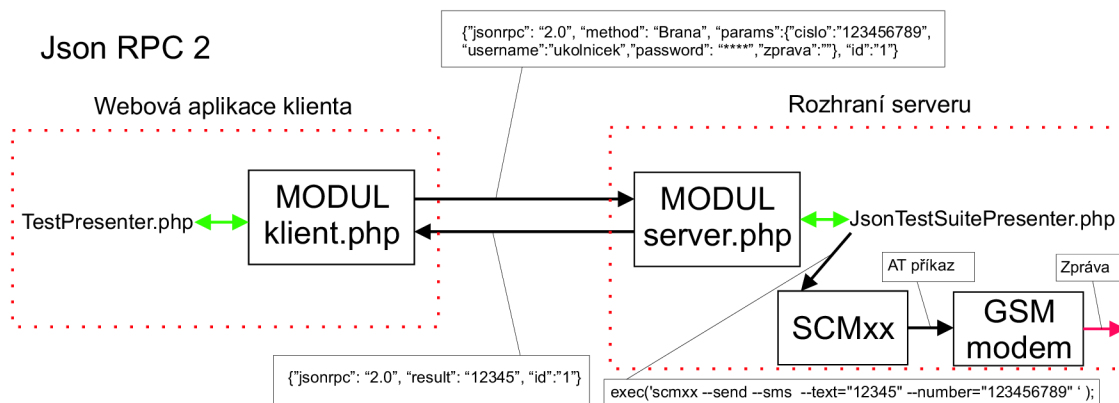
2.1.4 Presentery

Mají za úkol spojit Model a Template dohromady. Nejprve na základě požadavku od uživatele vyvolají příslušnou aplikační logiku (např. zobrazení tabulky Odeslat) a pak požádá šablonu o vykreslení výsledku.

- **BasePresenter.php** - základna pro všechny presentery - zpracovává data a zobrazí uživateli v menu všechny tabulky
- **ErrorPresenter.php** - zobrazuje uživateli chybová hlášení, rozhoduje se na základě výsledků debuggeru.
- **HomepagePresenter.php** - složí jako první stránka, zobrazuje formuláře k zadání nové zprávy, načte data a posílá je ke zpracování modelem. Pokud není uživatel přihlášen přesměruje jej na SignPresenter.
- **NastavPresenter.php** - slouží administrátorovi jako konfigurační stránka GSM modemu, ostatní uživatelé jí nevidí a nemají do ní přístup.
- **SignPresenter.php** - vyzývá uživatele ke zadání uživatelského jména a hesla, nechá ho zkontrolovat odpoví a přesměruje uživatele na homepagePresenter, a přijme žádost o odhlášení uživatele
- **SmsPresenter.php** - zobrazuje jednotlivé tabulky, je ošetřený, když nenajde vyhodí chybu „notFound“. Volá metodu pro načtení doručenek. Umožňuje mazat zprávy. V případě úpravy zprávy přesměrovává na „UpravSmsPresenter.php“.
- **UserPresenter.php** - umožňuje uživateli změnit si heslo, nastavená podmínka pro minimální délku hesla a shodnost nových hesel, potom informuje o úspěšnosti nebo neúspěšnosti změny hesla.
- **CreateUserPresenter.php** - umožňuje skupině administrátorů vytvářet nové a mazat stávající uživatele a přiřazovat je do dané skupiny, při úpravě přesměrovává na „UpravUserPresenter.php“.
- **AdresarPresenter.php** - umožňuje vytvářet a mazat kontakty do adresáře, vytvářet skupiny pro kontakty a vypisovat a filtrovat stávající kontakty v adresáři v případě změny volá „UpravAdresarPresenter.php“.
- **JsonTestSuitePresenter.php** - presenter pro API, aktivovaný příchozím připojením, obsahuje třídy, které slouží jako funkce serveru, přijímá data a na nich se rozhoduje co pošle přes modem a vrátí hodnotu klientovy, spolupracuje s „Server.php“.
- **NapisPresenter.php** - obsahuje funkci na odeslání mail zprávy pro správce.
- **UpravAdresarPresenter.php** - obsahuje funkce na úpravu již vytvořeného kontaktu.
- **UpravSmsPresenter.php** - obsahuje funkci na úpravu již vytvořené sms zprávy.
- **UpravUserPresenter.php** - obsahuje funkci na úpravu již vytvořené uživatele.
- **StatistikaPresenter.php** - slouží pro statistiku

2.1.5 Modul JSON RPC 2

Modul JSON RPC v2 byl zvolen hlavně pro možnost komunikovat pomocí tohoto protokolu i s jinými programovacími jazyky. Komunikace pomocí tohoto protokolu začíná klientem, který vytvoří instanci na server obsahující adresu serveru. A následně odešle objekt obsahující verzi protokolu, zvolenou metodu serveru a data, která očekává vykonávající presenter. Ten data zpracuje a vyhodnotí. Nejdříve si ověří existenci uživatelského jména a platného hesla. Pak se podívá jestli proměnná zpráva obsahuje data, když ne je vytvořen náhodný kód, který je odeslán na mobilní telefon a zpět klientovi jako odpověď, klient pak snadno může ověřit telefonní číslo. Pokud je v proměnné zpráva nějaký text je tento text odeslán na mobilní telefon a klientovy přijde odpověď o úspěšném či neúspěšném odeslání. Dále pak se přihlášenému klientovy odečte kredit a přičte se záznam o odeslání zprávy. Krom tohoto funkčního bezchybného modelu je vykonávající presenter schopen informovat při jakékoliv chybě, kdy například na serveru nemá klient dostatečný kredit nebo přístupové údaje účtu nebyly správně zadány.



Obr. 2.2: Schéma komunikace API.

2.1.6 Externí programy

- **scmxx** - program k odesílání a přijímání zpráv,
- **send.c** - program napsán v jazyku c, slouží k prohledání výstupu po scmxx, hledá v nich číslo `message_reference`, díky kterému je možné dohledat doručení. Jeho výstupem je toto číslo, jenž následně uloženo v databázi. Spouští se při volání metody, kdy se vytváří nová textová zpráva.
- **doruc.c** - program, který prohledává taktéž výstup z programu scmxx, a opět v nich hledá číslo `message_reference`, avšak s tím rozdílem, že prohledává přijaté zprávy a po nalezení čísla upraví hodnotu „done“ u zpráv v databázi, stránky si pak snadno přečtou, že zpráva byla doručena. Volá se při vykreslování tabulek.

2.1.7 Bootstrap

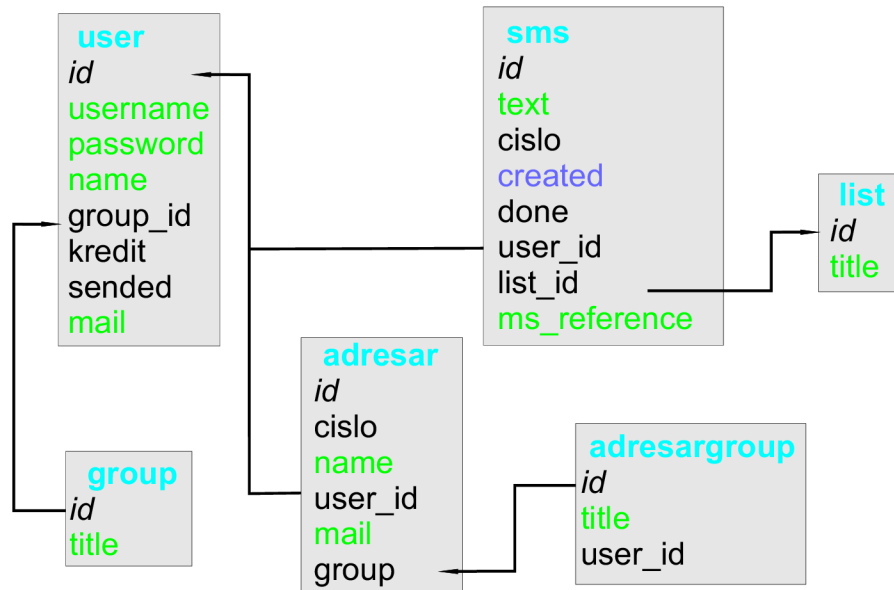
bootstrap.php - má za úkol nejdříve načíst Nette Framework. Aktivuje debugger a looger ve striktním režimu, debugger je mocný nástroj, který pomáhá vývojářům odhalovat chyby v kódu. Pokračujeme třídou „Configurator“, která vytvoří tzv. systémový DI kontejner podle konfiguračního souboru „config.neon“. Nejprve se však aktivuje autoloading, který zajistí automatické načítání všech souborů se zdrojovými kódy. Následně nastaví parametry aplikace, např. `error-presenter`, který se v produkčním prostředí stará o zobrazování hlášek při vyskytu chyby. Jakoukoliv další úlohu přidáme do události „onStartup“ např. připojení k databázi (což je obecně lepší nechat na DI kontejneru). A aplikaci spustí.

2.2 Databáze

V databázi smsbrana je šest tabulek.

Tabulka první se nazývá **user**, a jak z názvu vypovídá stará se o zpracování uživatelů, obsahuje id a název účtu, které musí být jedinečné, protože jinak by docházelo ke konfliktům a nebylo by možné určit ke kterému uživateli zprávy patří, další kolonkou je heslo, které je chráněno před útoky zašifrování, šifrování provádí přímo `authenticator.php` v modelu a pokud uživatel své heslo zapomene neexistuje způsob jak toto heslo znovu přechíst a musí se resetovat na defaultní, předposlední částí je jméno a příjmení uživatele účtu, to se zobrazuje v pravém horním rohu tohoto webového projektu, posledním sloupcem tabulky je `group_id`, podle kterého se určují práva, jaká uživatel má.

Další tabulka se jmenuje **sms** a ukládá všechny zprávy, které uživatelé napíšíou, každá zpráva je uložena v tabulce s několika dalšími parametry. První sloupeček je



Obr. 2.3: Schéma databáze.

id zprávy, toto číslo si tabulka generuje sama a je jedinečné, další sloupec obsahuje text sms zprávy, v tuto chvíli již je zpráva zbavená diakritiky. Následuje sloupec pro číslo, sem se ukládá číslo na které je zpráva odesílána. Za číslem následuje datum a čas odeslání, to je opět vytvořené ještě stránkami, při vytváření zprávy, databáze je schopná přijmout tento údaj jen ve správném formátu, který vypadá takto 0000-00-00 00:00:00. Naneštěstí je Nette framework natolik synchronizován s MySQL, že formát řešit nemusíme. Po datu a času následuje sloupeček done, který může obsahovat pouze hodnoty 0 a 1, ty vyjadřují jestli byla zpráva doručena (1) či nikoliv (0), tuto hodnotu upravuje program doruc.c. Další sloupec je user_id, ten má za úkol vědět, kterému uživateli vytvořená zpráva patří, tento údaj si stránky samy zjistí a pošlou k uložení do tabulky. Následuje sloupec list_id, který si ukládá číslo id listu, pro který byl vytvořen. Tuto hodnotu vybírá uživatel při tvorbě nové zprávy. Nakonec je tu sloupec ms_reference, do tohoto sloupce se zapisuje hodnota výstupu programu send.c a pomocí této hodnoty je schopen program doruc.c najít zprávu, která byla v pořádku doručena. Tabulka tyto data poskytuje pak stránkám, které si pomocí SQL příkazů nechají data vyfiltrovat a vyfiltrovaná data pak vypisují.

Po ní následuje tabulka **list**, ta obsahuje jména tabulek, mezi které jsou zprávy rozdělené podle jejich id čísla.

Tabulka **group**, ta obsahuje názvy různých skupin oprávnění, které rozděluje podle jejich group_id čísla.

Tabulka **adresar**, zde jsou uloženy záznamy kontaktů, které si sám uživatel uloží. Krom čísla a jména kontaktu může uživatel doplnit mail na danou osobu a skupinu do které bude uživatel uložen. Ostatní pole jsou vyplňována automaticky a podle přihlášeného uživatele.

Poslední tabulka je tabulka **adresargroup**, která obsahuje názvy různých skupin adresáře, krom hodnoty jména tabulky se uloží i, který uživatel danou skupinu vytvořil tím je docíleno, že každý uživatel vidí pouze defaultní a své vytvořené skupiny adresáře.

Velmi důležitou částí jsou vazby mezi tabulkami jinak řečené „**cizí klíče**.“ Cizí klíče spojují tabulky pomocí hodnot, které mají tabulky společné. V tomto projektu se jedná o spojení 1 ku n. První takovou vazbou je vazba mezi tabulkami user a sms, kde jsou tyto dvě tabulky propojené tak aby bylo možné záznamy v tabulce sms filtrovat podle uživatele, čehož se v projektu využívá a proto každý uživatel, kromě administrátora, viděl pouze své uložené a odeslané zprávy. Druhý cizí klíč spojuje tabulky list a sms, kdy jsou zprávy jsou rozdělené podle jejich list_id na odeslané a koncept a vypsané na levé straně v menu. To umožňuje filtrovat zprávy pro každého uživatele tak aby přehledně mohl sledovat své již odeslané zprávy a zprávy, které se chystá teprve poslat. Dalším cizím klíčem je spojení user a group, kde je vidět jaká práva daný uživatel má, tohoto se využívá například při vytváření nového uživatele nebo kontrolování odchozích zpráv z brány. Tabulky adresar a adresargroup spojuje cizí klíč tak aby bylo vidět, ve které skupině se daný kontakt z adresáře nachází, dále jsou propojeny cizím klíčem s tabulkou user, pomocí toho je docíleno, že kontakty a dané skupiny vidí jen ten uživatel co je vytvořil.

2.3 Projekt

2.3.1 Funkce webové aplikace

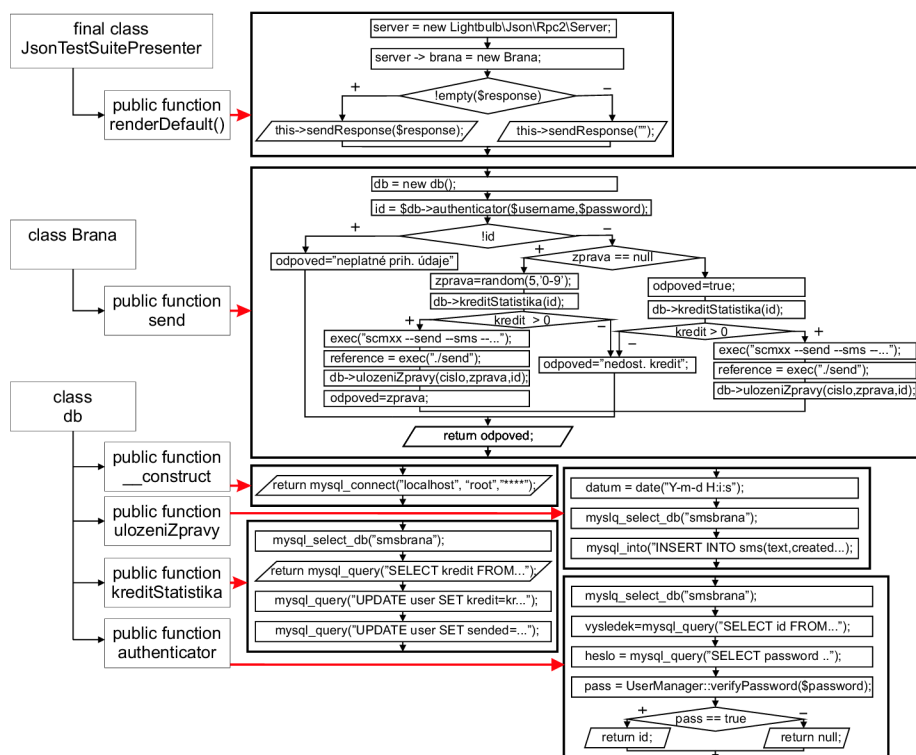
- Ochrana uživatelů před útoky pomocí zašifrovaných hesel pro přihlašování
- Uživatelé si mohou svá hesla změnit, kdy je minimální délka nového hesla 6 znaků
- Všechny formuláře jsou chráněny před útoky typu Cross-Site Request Forgery (CSRF) - tento typ útoku spočívá v tom, že přimějeme uživatele navštívit stránku, která skrytě vykoná útok na webovou aplikaci, kde je uživatel zrovna přihlášen, proti útoku se lze bránit generováním a ověřováním autorizačního tokenu.
- Odeslání zprávy je ošetřeno, že do telefonního čísla je možné vkládat pouze číslice a znak „+“, zpráva je omezena na 160 znaků (délka jedné zprávy) a byla odstraněna diakritika, při které docházelo k selhání při odesílání.

- Server odeslané zprávy ukládá a vypisuje pouze zprávy daného uživatele, jedinou výjimkou jsou správci webu a tzv. „kontrolori“, kteří mohou sledovat zprávy všech uživatelů.
- Administrátoři mohou nastavovat parametry modemu, k dispozici je zadání kódu PIN k odemčení modemu a synchronizace času modemu se serverem.
- Administrátoři vytvářejí, upravují nebo mažou uživatele, přidělují jim skupiny oprávnění a mohou resetovat hesla.
- Všichni uživatelé mohou využívat Adresář, do kterého mohou ukládat své kontakty, dělit je do jimi vytvořených skupin. Vypis kontaktů z adresáře je dostupný i při vytváření nové zprávy.
- Uživatelé mohou sledovat svoji aktivitu pomocí přehledného grafu.
- Jiné aplikace mohou využívat server díky vytvořenému API, kde jsou funkce pro ověření telefonního čísla nebo pro odeslání sms zprávy.

2.3.2 Jak funguje API

Spouštění Api probíhá pomocí zavolání presenteru. Ten defaultně spustí inicializaci serveru, vytvoří třídu serveru „brana“. Server zkontroluje odpověď a pokud je odpověď prázdná tak se upraví. Po příjmu dat od klienta jsou data přeměrována na třídu brana a dál na funkci „send“, která se předtím inicializovala. Tato třída vytvoří novou inicializaci na třídu „db“, která jí umožní komunikovat s databází. Po-té se zavolá funkce z třídy db a to „authenticator“, kde vstupními daty jsou uživatelské jméno a heslo. Tato funkce se připojí k databázi „smsbrana“ v dalším kroku vyhledá uživatele v databázi a pokud ho najde vrátí jeho id, po nalezení uživatele se uloží do proměnné „password“ jeho heslo a potom se zavolá statická funkce ze souboru „userManager.php“ v modelu. Tato funkce se nazývá verifyPassword, po ověření hesla navrátí funkce „true“ nebo „false“. Pokud je heslo v pořádku vrátí hodnotu id uživatele, pokud heslo nesouhlasí vrátí null. Po ukončení této funkce se ověří návratová hodnota pokud je „false“ dojde k ukončení a server vrátí klientovi odpověď s chybovým hlášením o nesprávném jméně nebo hesle. Pokud funkce vrátila „true“ třída pokračuje ve vykonávání funkce dál. V dalším kroku dojde k ověření obsahu proměnné „zprava“, pokud se její hodnota rovná „null“ funkce vytvoří náhodný číselný pětimístný kód. Pak se z třídy db zavolá funkce „kreditStatistika“. Když se funkce připojí k databázi ale upraví hodnoty v databázi nejprve u hodnoty „kredit“, která slouží k určení stavu kreditu a omezení uživatele. Dále pak hodnotu „sended“, která slouží pro statistiku, a funkce vrátí hodnotu kreditu a ukončí se. V hlavní funkci dojde k ověření stavu kreditu a při splnění podmínky dojde k odeslání zprávy přes příkazy exec, který spouští externí programy, v našem případě program „scmxx“ s parametry číslo, na které se odesílá zpráva, text zprávy, který v tomto

případě představuje pětimístný kód a po odeslání dojde k uložení zprávy pomocí funkce z třídy db pro uložení se nejdříve vygeneruje aktuální datum a čas, které se uloží do proměnné datum. Potom se připojí na databázi a vloží data do databáze z parametrů a z proměnné datum. Dále se vrátíme k vykonávání hlavní funkce, kdy se přesunou data z proměnné zprava do proměnné odpoved. Pokud proměnná zpráva obsahuje nějaká data, Uloží se do proměnné odpoved hodnota true a zbytek pokračuje nápodobně jako v předchozím případě, jen místo kódu se ve zprávě se nachází řetězec, vyplněný uživatelem. Následně jsou data z proměnné odeslat odeslána zpět na server, který je pošle zpět klientovi.

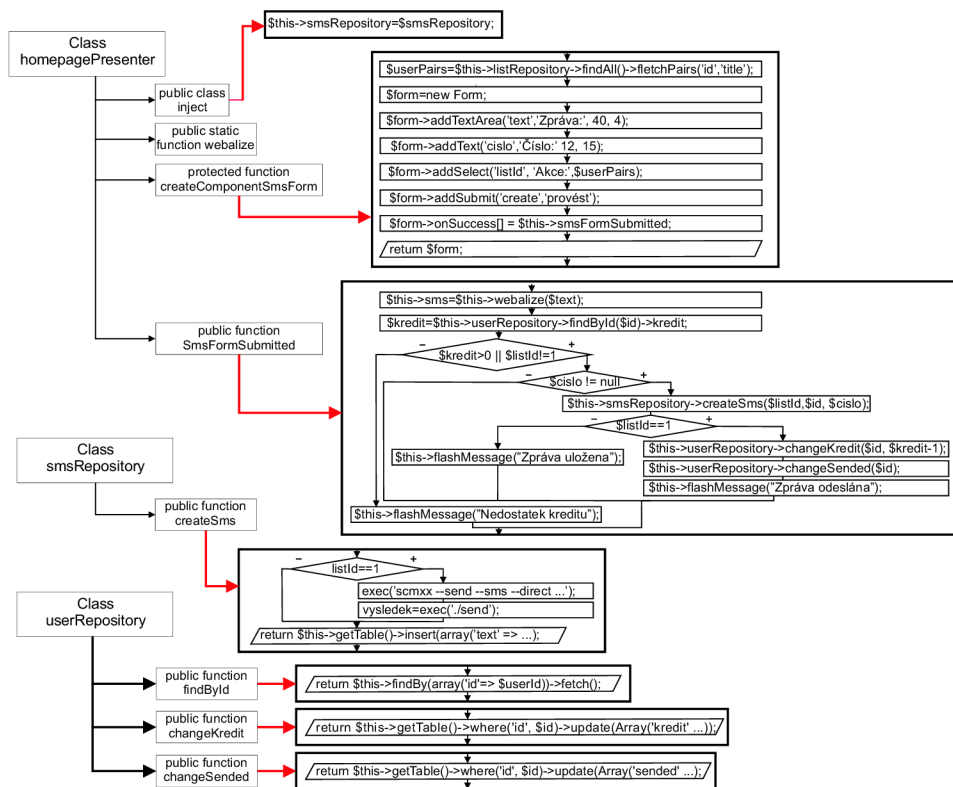


Obr. 2.4: Schéma API.

2.3.3 Jak funguje odesílání zprávy

Odeslání nové zprávy probíhá z homepagePresenteru, ten na začátku převezme načtená data z basePresenteru a „nainjektuje“ (načte) smsRepository, ve kterém se nachází funkce pro odeslání a uložení zprávy do databáze. Potom pomocí šablony (template) se načtou formuláře obsažené ve funkci „createComponentSmsForm“. Funkce si nejprve vytvoří skupinu „userPairs“, ve které načte databázové tabulky „odeslane“ a „koncept“ spolu s jejich id. Funkce dále inicializuje třídu Form, která slouží k vytvoření formulářů. Každé formulářové okno potom vytvoří podle zadaného

názvu (např. form->addText vytvoří textové políčko). Každý formulářový prvek pak obsahuje své jméno a titulek, který může zobrazit šablona, čísla za těmito názvy určují počáteční velikost formulářového prvku. Každý formulářový prvek může být navíc ošetřen pravidly vstupu (např. pro číslo možnost vkládat pouze číslice). Při nesplnění pravidla stránky vykreslí chybové hlášení. Odeslání se provede formulářovým prvkem „addSubmit“, po ověření, že jsou vstupní data v pořádku načtena, je spuštěna funkce „smsFormSubmitted“ a funkce nakonec vrátí vyplněné hodnoty.



Obr. 2.5: Schéma odesílání zprávy.

Následující funkce „smsFormSubmitted“ převezme data a v prvním kroku, prožene text zprávy přes funkci „webalize“, která odstraní diakritiku a další znaky, jenž by způsobovaly problémy při odesílání zprávy. Tato funkce je dafaultně v Nette knihovnách, v projektu lehce upravena. Po odstranění kritických znaků, funkce dál prověří stav kreditu uživatele a pokud je menší jak 1 a zároveň se má zpráva odesílat vyhodí chybové hlášení o nedostatku kreditu, pokud je kreditu dostatek nebo zpráva se pouze ukládá pokračuje program k podmínce, kdy prověří proměnnou „cislo“ (zde může být null, protože může být vybrán pouze kontakt z adresáře). V případě, že proměnná cislo obsahuje data je odeslána na funkci „createSms“ v smsRepositorech. Tato funkce se nejprve podívá, do které skupiny zpráva patří a pokud je skupina listId rovna 1, funkce spustí program „scmxx“ s parametry číslo a zpráva. Po jeho

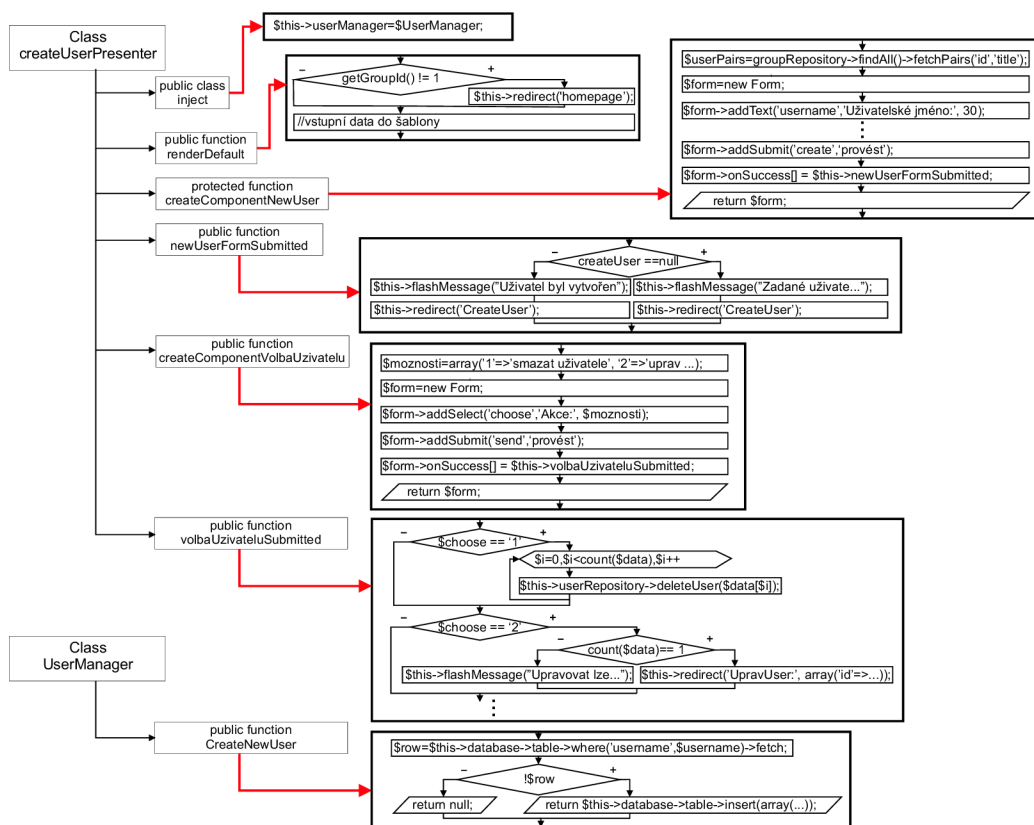
skončení funkce spustí druhý program „send“, ten projde výstup programu scmxx a navrátí hodnotu „ms_reference“, která slouží k určení doručení zprávy. Nakonec funkce zapíše zprávu do databáze a skončí. Dál pokračuje původní funkce, která si taky sama ověří kam zpráva přijde a pokud má listId rovné jedna, tak spustí další 2 funkce. První z nich upraví hodnotu kreditu, dříve načteného o mínus jedna a funkce „changeKredit“ jej změní v databázi, stejně tak funguje funkce „changeSended“, které stačí zadat id hodnotu uživatele a funkce automaticky navýší stav odeslaných zpráv o jedničku. Hlavní funkce nakonec zobrazí zprávu o úspěšném odeslání zprávy, v případě, že zpráva je ukládána jako koncept (v případě, že se listId != 1), je zobrazena hláška o uložení zprávy.

Pozn. Schéma neobsahuje princip použití adresáře, principiálně je totožný, akorát jsou všechny funkce zabaleny v cyklu for, který proběhne tolikrát kolik kontaktů z adresáře je vybráno.

2.3.4 Práce s uloženými zprávami a funkčnost doručení

Každá odeslaná zpráva je uložena do databáze. S uloženými zprávami může uživatel dále pracovat, a to smazat, upravit, vyhledávat a znovu odeslat. Stejná pravidla platí i pro zprávy typu „koncept“, které jsou uloženy ale nebyly odeslány. Každá odeslaná zpráva je po doručení potvrzena v tomto seznamu. Celý proces začíná v smsPresenteru, kdy se na počátku načte smsRepository, potom se pomocí funkce „actionDefault“ ověří platnost dané tabulky a to tak, že se vyhledá ve všech záznamech v tabulce list id vykreslované stránky a uloží se do globální proměnné list, pokud takový záznam v tabulce není, webová aplikace vygeneruje chybové hlášení o nenalezení stránky. Po vygenerování stránky funkci „renderDefault“ se provede spuštění další funkce „markDone“ v smsRepositorych. Tato funkce spustí program SCMxx s parametry ke zjištění nepřečtených zpráv, výstup programu je uložen do souboru. Další program „doruc“ otevře vygenerovaný soubor a načte z něj informace o doručených zprávách, po nalezení správného čísla ms_reference program upraví hodnoty v databázi na jedničku, což způsobí, že při vykreslování se u daných zpráv objeví stav „doručeno“. Tím je dokončeno vykreslení stránky. Avšak až teď může uživatel ovlivnit své uložené zprávy a to díky vytvořenému formuláři (v presenteru se jedná o funkci „createComponentVolbaZprav“). Kdy každá zpráva při generaci dostane svůj „checkbox“ s hodnotou id zprávy. Uživatel pak může označit kolik zpráv bude ovlivněno. Potom si v selectBoxu vybere jakou operaci se zprávami chce provést a stiskne tlačítko „Provést“. Tím se přepneme do funkce „volbaZpravSubmitted“, tato funkce si nejdříve stáhne informace o označených zprávách a na základě výběru provede danou operaci. Operace mazat a odeslat znovu lze provést více zpráv najednou ale upravovat zprávu lze pouze jednu. Proto ve volbě jedna i ve volbě

(do nette 2.1 Authenticator). Následuje funkce renderDefault, která obsahuje ověření práv uživatele, které pokud uživatel nesplňuje je odkázán zpět na domovskou stránku (homepage), po ověření dochází k předání dat do šablony, která na jejich základě vygeneruje seznam uživatelů. Dále šablona vykreslí dva různé formuláře. První z nich vychází z funkce „createComponentNewUser“, která slouží pro vytvoření nového uživatele. Funkce nejprve vyhledá všechny skupiny oprávnění a vytvoří pole dat id a title. Potom následuje samotné vytvoření instance formuláře a jednotlivé prvky, které chceme na stránce vykreslit. Pokud jsou formuláře správně vyplněny funkce se ukončí a přejde na funkci „newUserFormSubmitted“, kdy se pomocí podmínky if kontroluje zdárné vytvoření, funkce pro vytvoření „createNewUser“ se nachází ve zmíněném UserManageru, ten nejprve ověří zda není uživatelské jméno již v seznamu uživatelů, pokud ne zašifruje heslo a vloží informace o uživateli do databáze, pokud dojde k chybě při vkládání nebo uživatel již existuje funkce vrátí hodnotu null, má za následek vypsání hlášky o tom, že uživatel s takovým jménem již existuje a stránka se znovu vygeneruje v opačné případě stránka odpoví, že uživatel je vytvořen a opět se vygeneruje pro možnost vložit další záznam.

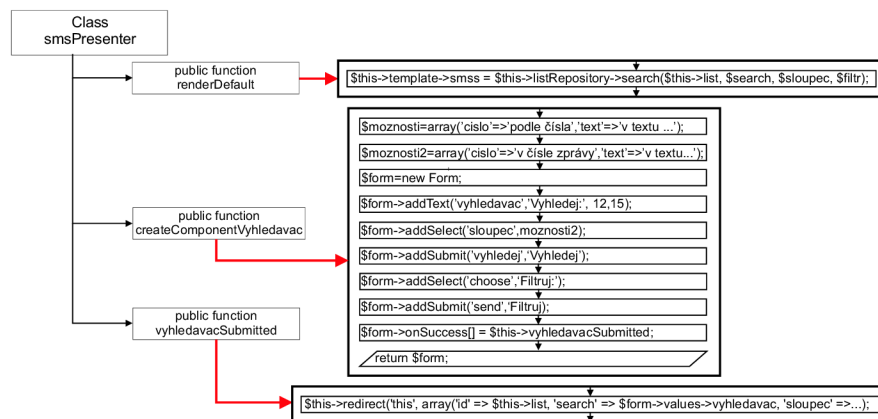


Obr. 2.7: Schéma správy uživatelů.

Druhý formulář slouží k výběru více uživatelů a operací s nimi, kdy je možnost je mazat, upravit, restartovat heslo a dobít kredit o určité částce. Po výběru uživatelů a operace s nimi se přepneme do funkce „volbaUzivateluSubmitted“. Zde bych chtěl upozornit, že tato metoda je velice podobná jejímu ekvivalentu pro úpravu zpráv, proto je zde schéma zjednodušené. Opět je tu pravidlo, kdy kromě úpravy uživatele kterého je možné upravovat pouze jednoho v danou chvíli, je možné ovlivnit najednou více uživatelů. Funkce pro mazání je totožná s funkcí pro mazání zpráv, jen je zde více kroků protože všechny data vytvořené daným uživatelem jsou na něm závislé, takže je nejprve nutno smazat tyto záznamy před samotným smazáním uživatele. Funkce pro restart hesla vygeneruje nové náhodné heslo danému uživateli. Funkce k dobytí kreditu je obdobná jako ve funkci odeslat zprávu, zde je akorát místo odečítání jedné přičtena vybraná hodnota kreditu.

2.3.6 Jak funguje vyhledávání

Vyhledávání usnadňuje uživatelům práci s webovou aplikací, vyhledávací funkce je možné najít ve zprávách, ve správě uživatelů a adresáři. Jejich koncept je velmi stejný a proto blokové schéma vychází jen z vyhledávače ve zprávách.



Obr. 2.8: Schéma vyhledávací funkce v smsPresenteru.

Celé to funguje na principu úprav výpisu zpráv při generování dat do šablony, tudíž je hlavním pachatelem funkce renderDefault, ve které máme měněná pravidla

uvedena ve funkci s defaultními hodnotami, jedna se například o případ, kdy se generuje stránka poprvé. Pokud je v některé proměnné při načítání stránky hodnota, stránka si vezme tuto hodnotu a ta ovlivní funkci, která slouží k vypsání zpráv. Aby bylo možné tyto hodnoty ovlivňovat je na každé stránce s vyhledávací funkcí formulář, ve kterém si uživatel může nastavit, co vyhledávat, kde to vyhledávat a jak výsledky poskládat. Možnosti vyhledávat a přeskládat výsledky nejsou na sobě závislé a tak je možné použít jen jednu z nich nebo obě dvě. Tento formulář vzniká ve funkci „createComponentVyhledavac“, nejprve se generují možnosti, jednou pro vyhledávač a jednou pro přeskládání výsledků. Inicializuje se třída Form pro vytvoření skupiny prvků, a přidávají se jednotlivé prvky. Dva submity jsou pouze kvůli vizuální stránce, oba spínají všechny prvky formuláře. Po úspěšném zadání se opět přesuneme na funkci „vyhledavacSubmitted“. Tato funkce obsahuje jeden řádek, který způsobí nové vygenerování stránky se zadanými novými parametry, které ovlivní výpis zpráv. Pro správu uživatelů výpis uživatelů a pro adresář výpis kontaktů.

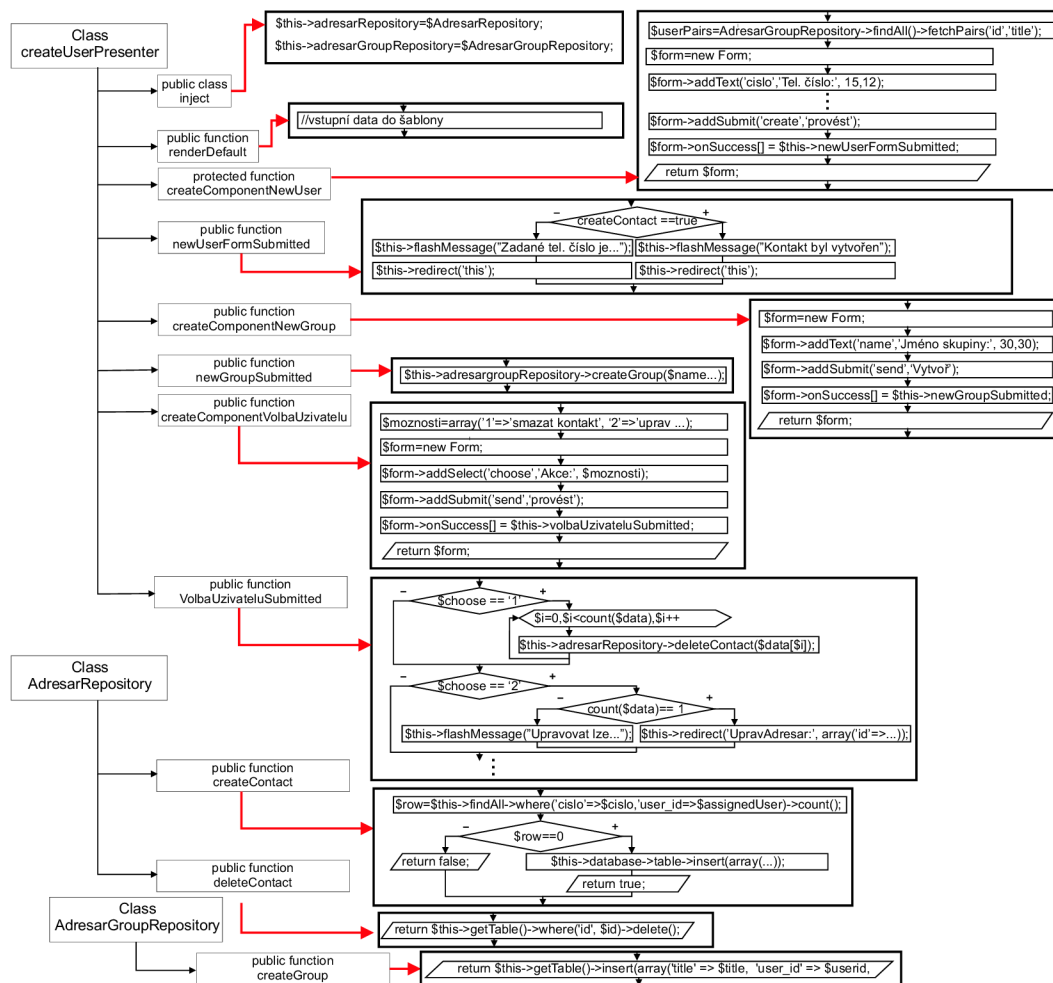
2.3.7 Jak funguje adresář

Adresář je dostupný všem užívajícím uživatelům, uživatele mohou do adresáře ukládat své kontakty, přiřazovat je do jimi předem vytvořených skupin, mazat a přesouvat i několik kontaktů najednou. Hlavní výhodou adresáře je, že kontakty jsou dostupné při psaní nové zprávy a tak je možné vybrat několik příjemců. Kontakty v adresáři i skupiny kontaktů vidí pouze jejich majitelé, jedinou společnou skupinou je „nezarazeno“ což je defaultní skupina pro všechny uživatele.

Funkcionalita celého adresáře je velmi podobná správě uživatelů. S tím rozdílem, že je potřeba načíst funkce z adresar a z adresarGroup nacházejících se v modelu. Vytváření skupin není nic jiného než další formulář, který po odeslání je zapsán do databáze. A po opětovném načtení stránky jej lze již používat.

2.3.8 Jak se vykresluje statistika

Statistika je generovaný graf z hodnot sended a z celkového počtu zpráv daného uživatele, kdy hodnotu koncept dopočítáváme snadno z předchozích dvou hodnot (celkový počet zpráv - sended = koncept). Samotný graf se generujeme pomocí Flot což je jQuery knihovna. Tato knihovna je relativně jednoduchá. Funguje to tak, že se v těle stránky vytvoří „script“ který se ošetří podmínkou zda-li je uživatel přihlášen. Pokud by se toto nedělalo stránka by se pokusila získat data od anonymního uživatele což by vedlo k pádu celé webové aplikace. Jakmile je uživatel přihlášen vygenerují se data pro graf a nastaví se parametry grafu. Samotné vykreslení se



Obr. 2.9: Schéma adresáře.

spustí ve chvíli, kdy uživatel otevře statistikaPresenter, který má v šabloně příkaz pro vykreslení grafu. [12]

3 ZÁVĚR

V této práci se řešili možnosti využití dynamických webových stránek s propojením GSM modemu. V první fázi bylo třeba najít takový GSM modem, který by toto spojení umožnil. Po zhodnocení obrovské nabídky různých modelů a výrobců se v rámci této práce několik z nich vybralo a z tohoto výběru se nakonec zvolil Cinterion MC55i. Po zjištění nepřehledného množství programů pro komunikaci s GSM modemy a složitěho vytváření sms zpráv pomocí AT příkazů jsme se rozhodli jeden z těchto programů použít. Po jejich opětovném zhodnocení se několik z nich vybralo, nejlepší volbou dopadl program scmxx pro jeho jednoduchost a snadnou konfiguraci s modemem. Po úspěšném odeslání testovací sms zprávy jsem začal s konfigurací svého serveru. Kdy bylo potřeba na server nahrát tzv. LAMP server, což je zkratka čtyř slov Linux, Apache2, MySQL a PHP. Než se však začalo na projektu pracovat, prošly se si již existující webové stránky provozující sms bránu. Po jejich shlédnutí a vyzkoušení se v rámci tohoto projektu vytvořilo jejich zhodnocení. Po té jsem se již pustil do tohoto projektu. Prvním úspěchem byla obyčejná stránka s dvěma textovými poli a tlačítkem odeslat. Pak jsem se zabýval frameworkem a objektové programování, kdy se pomocí dokumentace nakonec zdařilo vytvořit stránku, do které již šlo ukládat zprávy a zpětně je načítat, s postupem času stránka nabývala na schopnostech první bylo přihlašování a odhlašování uživatelů a takřka posledním úspěchem filtrování zpráv pro každého uživatele a zobrazení všech zpráv administrátorovi. Na konci této práce vznikly dva programy, díky kterým plně fungují doručení.

V další fázi práce se nejdříve aktualizovala Nette knihovna na verzi 2.1, což umožnilo používání dynamických formulářů. Po přechodu se postavilo API rozhraní na bázi protokolu Json RPC2, jenž zaručuje multiplatformnost mezi ostatními programovacími jazyky, kdy API může fungovat jako ověřovač telefonních čísel nebo lze přes něj posílat sms zprávy. Vylepšen byl výpis uložených zpráv, kdy lze ve zprávách hledat nebo je přeskládat podle uživatelem chtěných parametrů. Zprávy nyní lze i mazat, upravovat a znovu odeslat. Došlo i na rozšíření možnosti správy uživatele, kdy kromě vytváření uživatele je možné uživatele upravovat a mazat, navíc dochází k přehlednému výpisu uživatelů s možností mezi uživateli vyhledávat nebo je přeskládat. Stejně funkce dostal i nově vybudovaný adresář, ve kterém si uživatelé mohou přehledně ukládat kontakty do různých skupin. Hlavní výhodou adresáře je možnost výběru kontaktů při psaní nové zprávy. Dalším takovým velkým plusem pro uživatele je přehledný graf, ve kterém mohou vyčíst kolik zpráv už odeslali. Poslední velkou změnou byla změna designu pomocí kaskádových stylů k větší přehlednosti na webové aplikaci.

LITERATURA

- [1] DUBEN, S. *Základy jazyka SQL a databází - I. díl* [online]. 2007, poslední aktualizace 29. 1. 2010 [cit. 20. 11. 2013]. Dostupné z URL: <<http://duben.org/zaklady-sql/zaklady-jazyka-sql-a-databazi-i-dil>>.
- [2] Škrášek, J. *PHP frameworky* [online]. 2008, poslední aktualizace 21. 2. 2008 [cit. 20. 11. 2013]. Dostupné z URL: <<http://programujte.com/clanek/2008022000-php-frameworky/>>.
- [3] *Nette Framework dokumentace* [online]. 2008, poslední aktualizace 2013 [cit. 11. 12. 2013]. Dostupné z URL: <<http://doc.nette.org/cs/>>.
- [4] *RS Components - Maestro 100evo Advanced*[online]. [cit. 24. 11. 2013] Dostupné z URL: <<http://cz.rs-online.com/web/p/modemy-gsm-a-gprs/7424472/>>.
- [5] *RS Components - GSM/GPRS RS232 Modem Dual Band*[online]. [cit. 24. 11. 2013] Dostupné z URL: <<http://cz.rs-online.com/web/p/modemy-gsm-a-gprs/7620086/>>.
- [6] *Púhy.cz - MC55i Cinterion (Siemens) terminál*[online]. 2013[cit. 24. 11. 2013] Dostupné z URL: <<http://www.puhy.cz/mc55i-cinterion-siemens-terminal-sada-1189.html>>.
- [7] *RS Components - AirLink Fastrack Xtend Modem*[online]. [cit. 24. 11. 2013] Dostupné z URL: <<http://cz.rs-online.com/web/p/modemy-gsm-a-gprs/7165249/>>.
- [8] Kasvi, K. *SMS tools 3*[online]. poslední aktualizace 21. 10. 2012 [cit. 24. 11. 2013] Dostupné z URL: <<http://smstools3.kekekasvi.com/>>.
- [9] Shevchenko, A. Sattler, H. *Sourceforge - SCMxx*[online]. 2013, poslední aktualizace 17. 4. 2013 [cit. 24. 11. 2013] Dostupné z URL: <<http://sourceforge.net/projects/scmxx/>>.
- [10] Čihař, M. Wiacek M. *[GW]ammu*[online]. 2003, poslední aktualizace 2013 [cit. 24. 11. 2013] Dostupné z URL: <<http://wammu.eu/>>.
- [11] *Diafaan communication software*[online]. 1992, poslední aktualizace 2013 [cit. 24. 11. 2013] Dostupné z URL: <<http://www.diafaan.com/>>.
- [12] *Dynamicky generované interaktivní grafy na vašem webu* [online]. [cit. 11. 5. 2014]. Dostupné z URL: <<http://www.tovarna.cz/cz/aktuality/282-dynamicky-generovane-interaktivni-grafy-na-vasem-webu/>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

GSM Globální Systém pro Mobilní komunikaci – Groupe Spécial Mobile

USB Univerzální sériová sběrnice – Universal Serial Bus

COM sériový port – RS232

PHP hypertextový procesor – Hypertext Preprocessor

SQL Strukturovaný dotazovací jazyk – Structured Query Language

CSRF Cross-Site Request Forgery

API Aplikační rozhraní - Application Programming Interface

SEZNAM PŘÍLOH

A	Otestování aplikace	40
A.1	Server bez modemu	40
A.2	Server s modemem	40

A OTESTOVÁNÍ APLIKACE

Pro vyzkoušení běží aplikace na dvou na sobě nezávislých serverů. První z nich běží neustále ale není k němu připojen modem. Druhý z nich běží jen omezeně (je na noc vypínán), ale umožňuje připojení modemu a instalaci podpůrných programů. K připojení k obou serverům stačí jakýkoliv moderní webový prohlížeč.

A.1 Server bez modemu

Jedná se o jednotku NAS s upraveným operačním systémem. Nemá port COM a jiné porty lze užívat jen velmi omezeně. Operačním systémem neumožňuje instalaci jiných než výrobcem ověřených programů.

Dostupné z URL:

`<http://213.211.48.202/smsbrana/sandbox/www/>`.

Dostupné účty pro přihlášení:

Uživatelské jméno:	Heslo:	Oprávnění:
admin	matrox	administrátor
Tester	tester	kontrolor
john	johndoe	uživatel

A.2 Server s modemem

Jedná se o virtualizovaný linux na desktopovém počítači. Počítač má port COM ke kterému je připojen modem a lze z něj odesílat zprávy a přijímat doručky.

Dostupné z URL:

`<http://213.211.48.202:81/smsbrana/sandbox/www/>`.

Dostupné účty pro přihlášení:

Uživatelské jméno:	Heslo:	Oprávnění:
admin	matrox	administrátor
Tester	tester	kontrolor
john	johndoe	uživatel