

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

SMS Brána
Bc. Drahošlav Stejskal

Diplomová práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Drahoslav Stejskal**
Osobní číslo: **I11410**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Komunikační brána pro hromadné rozesílání SMS zpráv**
Zadávající katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem diplomové práce je vytvoření SMS brány, která bude umožňovat hromadné rozesílání SMS zpráv.

V úvodní části bude provedena rešerše dostupných řešení na trhu, bude zhodnocena legislativa ve vztahu k problematice spamu a rešerše dostupnosti příslušného API u různých mobilních platforem.

SMS brána bude vytvořena jako aplikace do vybrané mobilní platformy a bude disponovat serverovou částí pro centrální řízení aplikace.

Výstupem práce bude implementace navržené SMS brány s veřejně dostupným API, který bude moci být použit třetími stranami ve svých aplikacích.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

ARLOW, J., NEUSTADT, I. UML 2 a unifikovaný proces vývoje aplikací. Brno: Computer Press 2007. ISBN 978-80-251-1503-9.

Schmitt, Christofer, and Simpson, Kyle. HTML5 Cookbook. O'Reilly, 2011.

Zakas, Nicholas Z. JavaScript pro webové vývojáře. Computer Press, 2009.

Vedoucí diplomové práce:

Ing. Lukáš Čegan, Ph.D.

Katedra informačních technologií

Datum zadání diplomové práce: **31. října 2012**

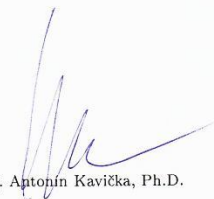
Termín odevzdání diplomové práce: **17. května 2013**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 17. 5. 2013

Bc. Drahošlav Stejskal

Poděkování

Děkuji svému vedoucímu Ing. Lukáši Čeganovi Ph.D. za vedení a myšlenku celé diplomové práce. Dále děkuji své přítelkyni a rodině za psychickou podporu a všem, kteří mi během tvorby práce přispěli jakoukoli radou.

Anotace

Diplomová práce se zabývá vytvořením vlastní SMS brány za pomoci mobilního telefonu platformy Android. Práce se skládá z mobilní aplikace a serverové části. Mobilní aplikace slouží k rozesílání zpráv do mobilní sítě. Serverová část poskytuje API rozhraní a uživatelské rozhraní SMS brány.

Klíčová slova

SMS brána, Android, SQLite, PHP, MySQL

Title

SMS gateway

Annotation

This thesis deals with creating of SMS gateway for platform Android. The work consists of a mobile application and a server part. Mobile application is used to send messages to the mobile network. The server part provides the API and user interface SMS gateway.

Keywords

SMS gateway, Android, SQLite, PHP, MySQL

Obsah

Seznam zkratk	10
Seznam obrázků	11
Seznam tabulek	12
Úvod	13
1 Dostupná řešení na trhu	14
1.1 Kubík SMS DreamCom	14
1.2 Daktela.....	16
1.3 Smsbrana.cz.....	17
1.4 Axima Direct SMS	19
1.5 Klikniavolej.cz.....	20
1.6 Droid Messenger	21
1.7 Remote SMS Sender.....	22
1.8 O2 SMS Connector.....	23
1.9 SMSC Direct	24
1.10Shrnutí stávajících řešení.....	25
2 Dostupnost API pro odesílání SMS u různých mobilních platformem	26
2.1 Android.....	26
2.2 Windows Phone.....	27
2.3 BlackBerry.....	27
2.4 Symbian OS.....	27
2.5 iOS.....	28
2.6 Hodnocení.....	28
3 Problematika spamu na mobilních telefonech	29
3.1 Prémiové textové zprávy	29
3.2 Podvodná zpráva pro zaslání kódu	29
3.2.1 Princip objednání služby s platbou přes mobil.....	29
3.2.2 Princip podvodného objednání služby.....	29
3.2.3 Jak se bránit	30
3.3 Antispamová legislativa	30
3.3.1 Informovaný souhlas	30
3.4 Spam a vymahatelnost.....	31

3.5	Trest za rozesílání spamu	31
3.6	Stalking.....	31
3.6.1	Trest za provozování stalkingu.....	31
3.6.2	Znění zákona č.40/2009 Sb. § 354 Nebezpečné pronásledování	31
3.7	Účinnost antispamového zákona	32
4	Analýza a návrh aplikace.....	33
4.1	Požadavky na aplikaci	33
4.2	Funkční požadavky.....	33
4.3	Nefunkční požadavky	33
4.4	Klady a zápory aplikace	34
4.5	Tarifů pro neomezené odesílání SMS	34
4.6	Porovnání navržené aplikace s konkurencí	35
4.7	Use Case diagram webové aplikace	36
4.8	Use Case diagram mobilní aplikace	39
4.9	Návrh tříd mobilní aplikace.....	41
4.10	Návrh modelu webové aplikace	46
4.11	Návrh databázového modelu	52
4.11.1	EER model serverové části.....	52
4.12	Význam tabulek.....	53
4.13	SQLite databáze.....	54
5	Implementace řešení.....	54
5.1	Grafické schéma systému	54
5.2	Použité technologie	55
	Android.....	56
5.2.1	SQLite.....	56
5.2.2	Eclipse	56
5.2.3	Android SDK.....	57
5.2.4	PHP.....	57
5.2.5	JavaScript	57
5.2.6	HTML.....	57
5.2.7	CSS.....	57
5.2.8	NetBeans.....	58
5.2.9	MySQL.....	58

5.2.10	JSON.....	58
5.3	Mobilní aplikace.....	59
5.3.1	Architektura aplikace pro Android.....	59
5.3.2	Založení projektu.....	59
5.3.3	Struktura adresářů.....	62
5.3.4	Balíčky aplikace.....	63
5.3.5	Obsah manifestu.....	64
5.3.6	Implementace odesílání zpráv.....	65
5.3.7	Odstranění Android limitu.....	67
5.3.8	Implementace fronty zpráv.....	67
5.3.9	Implementace tříd spravujících databázi.....	68
5.3.10	Komunikace se serverovou částí.....	69
5.3.11	Uživatelské rozhraní.....	70
5.4	Webová aplikace.....	72
5.4.1	Struktura a funkce aplikace.....	73
5.4.2	Zabezpečení.....	74
5.4.3	Odeslání zprávy.....	75
5.4.4	API.....	77
5.4.5	AJAX.....	77
5.4.6	Uživatelské rozhraní.....	81
5.5	Shrnutí.....	84
	Závěr.....	84
	Literatura.....	85
	Příloha A – Příložené CD.....	89
	Příloha B – Dokumentace API.....	89
	Dokumentace API.....	89

Seznam zkratk

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CSS	Cascading Style Sheets
CSV	Comma-separated values
GSM	Groupe Spécial Mobile (Globální Systém pro Mobilní komunikaci)
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
JS	JavaScript
JSON	JavaScript Object Notation
OS	Operating System
PHP	Hypertext Preprocessor
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
SMS	Short Message Service
SQL	Structured Query Language
UC	Use Case
UI	User Interface
UP	Unified Process
XML	Extensible Markup Language

Seznam obrázků

Obrázek 1 – Kubík SMS desktopová aplikace.....	15
Obrázek 2 – Kubík SMS mobilní aplikace.....	16
Obrázek 3 – Webové rozhraní Daktela	17
Obrázek 4 – Webové rozhraní Smsbrana.cz	18
Obrázek 5 – Aplikace pro smartphone SMSBrána.cz.....	19
Obrázek 6 – Mobilní aplikace klikniavolej.cz.....	20
Obrázek 7 – Webové rozhraní aplikace klikniavolej.cz.....	21
Obrázek 8 – Webové rozhraní Droid Messenger	22
Obrázek 9 – Mobilní aplikace Droid Messenger.....	22
Obrázek 10 – Mobilní a webové rozhraní Remote SMS Sender.....	23
Obrázek 11 – O2 SMS Connector, tabulka cen.....	24
Obrázek 12 –SMSC Direct, ceník odchozích SMS.....	25
Obrázek 13 – Use Case diagram webové aplikace.....	36
Obrázek 14 – Use Case diagram mobilní aplikace.....	39
Obrázek 15 – Diagram tříd MainActivity	41
Obrázek 16 – Diagram tříd SMSListActivity	43
Obrázek 17 – Diagram tříd SmsSender.....	44
Obrázek 18 – Diagram tříd databáze.....	45
Obrázek 19 – Activity diagram registrace a editace uživatele	46
Obrázek 20 – Activity diagram správa kontaktů.....	47
Obrázek 21 – Activity diagram správa skupin	48
Obrázek 22 – Activity diagram odeslání SMS.....	49
Obrázek 23 – Activity diagram API.....	51
Obrázek 24 – EER model.....	52
Obrázek 25 – Grafické schéma systému	55
Obrázek 26 – JSON objekt.....	58
Obrázek 27 – Založení nové Android Aplikace.....	60
Obrázek 28 – Volba ikony aplikace	61
Obrázek 29 – Volba navigace activity	61
Obrázek 30 – Struktura adresářů.....	62
Obrázek 31 – Balíčky mobilní aplikace	63
Obrázek 32 – Úvodní obrazovka aplikace	70
Obrázek 33 – Fronta zpráv	71
Obrázek 34 – Psaní SMS.....	71
Obrázek 35 – Nastavení aplikace	71
Obrázek 36 – Statistika a informace o aplikaci.....	72
Obrázek 37 – Struktura webové aplikace.....	73
Obrázek 38 – Úvodní obrazovka.....	81
Obrázek 39 –Uživatelské rozhraní pro odeslání SMS zprávy.....	82
Obrázek 40 – Přihlášení a registrace uživatele.....	82
Obrázek 41 – Adresář, vytvoření kontaktu	83
Obrázek 42 – Adresář, editace kontaktu	83

Seznam tabulek

Tabulka 1 – Zhodnocení aplikace Kubík SMS.....	15
Tabulka 2 – Zhodnocení aplikace Daktela	17
Tabulka 3 – Zhodnocení aplikace SmsBrana.cz.....	18
Tabulka 4 – Zhodnocení Axima Direct SMS	19
Tabulka 5 – Zhodnocení aplikace klikniavolej.cz.....	20
Tabulka 6 – Zhodnocení aplikace Droid Messenger	21
Tabulka 7 – Zhodnocení O2 SMS Connector	24
Tabulka 8 – Zhodnocení SMSC Direct	25
Tabulka 9 – Parametry metody sendTextMessage	26
Tabulka 10 – Zhodnocení navrhované aplikace	34
Tabulka 11 – Tarify pro neomezené zasílání SMS.....	34

Úvod

V dnešním světě jsou elektronické komunikační kanály neodmyslitelnou součástí našich životů. Mobilní telefon dnes vlastní téměř každý od dětí po seniory. Mnoho lidí si dokonce neumí představit svůj den bez tohoto přístroje. Základní funkce mobilního telefonu jsou volání a zasílání krátkých textových zpráv o délce 160 znaků, což odpovídá 140 bytům dat.

Ačkoliv je dnes zcela běžné komunikovat na telefonu prostřednictvím mobilního internetu, cena obyčejných textových zpráv si stále drží svoji cenu zhruba 1,50 Kč za jednu SMS. Mnoho lidí tuto službu využívá i přes to, že vlastní mobilní internet, který lze snadno využít k bezplatné komunikaci s druhou osobou. SMS je zkrátka jednoduchý a rychlý způsob mezilidské komunikace.

Textové zprávy mají stále vysoké využití ve firemní sféře, kde se hojně používají například pro upozornění o změnách stavu objednávky, pro marketingové účely a komunikaci se zákazníkem. V tomto případě je potřeba sáhnout po nějakém komerčním řešení, které umožňuje automatizaci odesílání SMS zpráv přes funkce SMS brány.

Účelem této práce je poskytnout nástroj, jenž umožní pohodlné odesílání hromadných zpráv přes SMS bránu za využití vlastního mobilního telefonu. V současnosti existuje mnoho komerčních řešení, které umožňují využívat rozhraní pro levnější odesílání textových zpráv. Nenarazil jsem však na žádnou aplikaci, která by umožňovala proměnit vlastní mobilní telefon v SMS bránu při zachování pokročilých funkcí profesionálních SMS brán.

Diplomová práce je rozdělena do pěti kapitol. První kapitola ukazuje současná řešení na trhu. Druhá kapitola se zabývá dostupností rozhraní pro odesílání zpráv u různých mobilních platforem. Třetí kapitola řeší problematiku spamu, která k rozesílání hromadných SMS zpráv jistě také patří. Kapitola čtvrtá se již zaměřuje na analýzu a návrh vlastní mobilní a serverové aplikace. Poslední kapitola popisuje implementaci řešení a nechá čtenáře nahlédnout do uživatelského rozhraní.

1 Dostupná řešení na trhu

Na našem trhu se vyskytuje mnoho aplikací, které nabízí odesílání SMS¹. Všichni tři čeští operátoři nabízí placené SMS brány. Tyto brány umožňují také rozesílání zpráv zdarma, to však pouze v případě odeslání do vlastní sítě. V České republice je bohužel cena SMS stále velice vysoká, pohybuje se v ceně kolem 1,50 Kč za 160 odeslaných znaků. Pro odesílání hromadných SMS je toto řešení nevhodné.

Pro hromadné rozesílání SMS, existují tzv. SMS agregátory, u kterých je cena za odeslanou SMS výrazně nižší. Taková SMS brána je napojena přímo na SMS centra mobilních operátorů. Poskytovatel služeb má sjednaný s operátorem výhodnější kontrakt než je v nabídce trhu pro běžného zákazníka.

Všichni mobilní operátoři na území České republiky v současnosti nabízejí tarif či balíček, který umožňuje neomezené rozesílání SMS za určitý paušální poplatek. Proč tedy takové nabídky nevyužít? Na trhu jsem se zatím nesešel s aplikací, která by umožňovala z telefonu zákazníka vytvořit SMS bránu pro odesílání zpráv, při zachování pokročilejších funkcí. V rešerši dostupných řešení uvedu několik porovnatelných aplikací z hlediska zadání diplomové práce.[1]

Hlavní kritéria, dle kterých jsou srovnávány aplikace pro odesílání SMS:

- Cena za odeslanou SMS.
- Možnost hromadného rozesílání textových zpráv.
- Možnost naplánovat čas odeslání zprávy.
- Dostupnost API.
- Možnost archivace odeslaných zpráv.
- Možnost využívat předdefinované telefonní kontakty či skupiny čísel.
- Možnost odeslání zprávy z počítače.

1.1 Kubík SMS DreamCom

Aplikace pro odesílání SMS z rodiny DreamCom představuje poměrně komplexní řešení pro odesílání zpráv. Textové zprávy lze rozesílat aplikací Kubík SMS DreamCom, určenou pro počítače s operačním systémem Windows.

Tato aplikace umožňuje import kontaktů, např. pomocí CSV² souboru, avšak samotné spárování telefonu s aplikací je dnes již téměř nemožné. Pokud bych chtěl aplikaci používat pro rozesílání zpráv přes připojený telefon, musí tento přístroj disponovat COM připojením a Hardwarovým GSM³ modemem. Sám autor aplikace uvádí, že aplikace

¹Short message service

²Jednoduchý souborový formát pro výměnu tabulkových dat.

³ Globální systém pro mobilní komunikaci

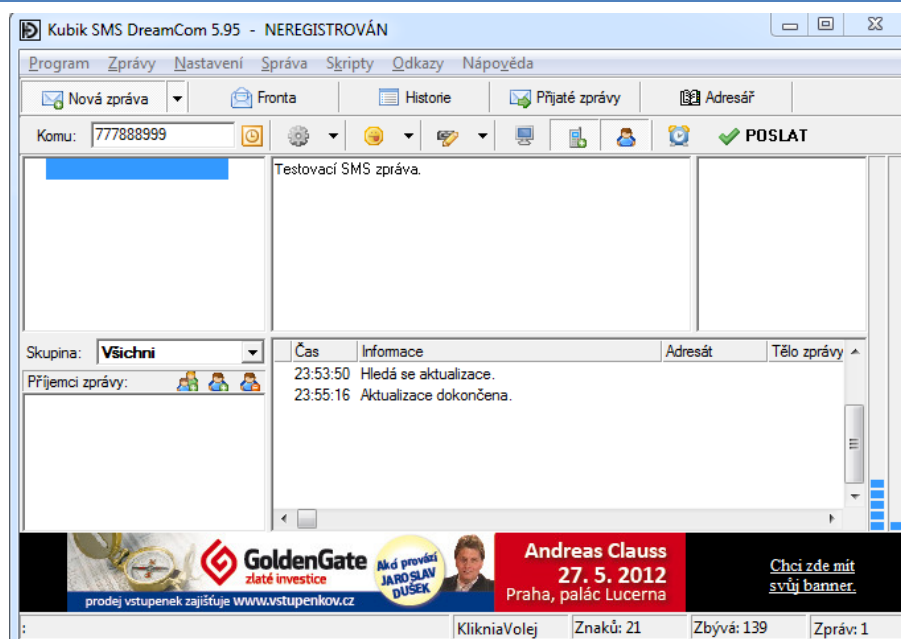
neumožňuje připojit novější telefony se systémy Symbian, Windows Mobile, iPhone a dle vlastní zkušenosti ani Android. Uživatelské rozhraní aplikace zobrazuje Obrázek 1.

Další možností aplikace je využití partnerských SMS brán. Tento způsob je možné využívat v desktopové aplikaci a také přes webové rozhraní partnerů. Cena za využití je však pro masové rozesílání zpráv stále vysoká – 0,81 Kč za jednu textovou zprávu.

Poslední možností, která přibyla v roce 2010, je aplikace Kubík SMS pro mobilní telefony běžící na platformě Android (Obrázek 2). Tato aplikace je velice jednoduchá, a umožňuje rozesílat zprávy přes bezplatné SMS brány operátorů, případně přes SMS bránu určenou pro operátora užívané SIM karty. Dle ohlasů uživatelů bohužel aplikace není zcela doladěná a objevují se v ní stále problémy s odesláním. [2]

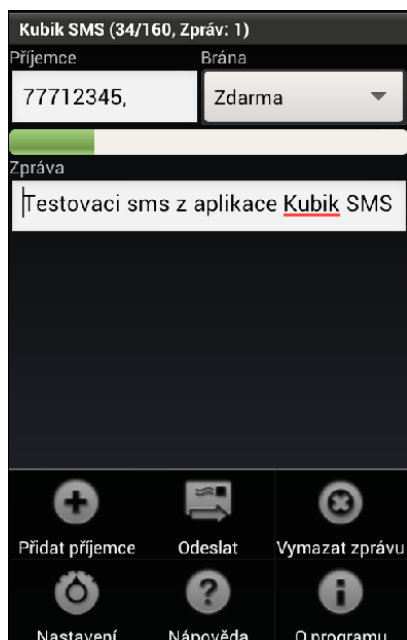
Tabulka 1 – Zhodnocení aplikace Kubík SMS

Výhody	Nevýhody
<ul style="list-style-type: none"> • Neomezeně dlouhé zprávy 	<ul style="list-style-type: none"> • Placené SMS přes Dreamcom bránu za 0,81 Kč
<ul style="list-style-type: none"> • API rozhraní 	<ul style="list-style-type: none"> • Placená verze za 300 Kč
<ul style="list-style-type: none"> • Odesílání zpráv z MS Office 	<ul style="list-style-type: none"> • Banner v bezplatné verzi
<ul style="list-style-type: none"> • Odesílání přes bezplatné SMS brány 	<ul style="list-style-type: none"> • Pouze COM připojení telefonu
<ul style="list-style-type: none"> • Odesílání přes mobilní telefon 	<ul style="list-style-type: none"> • Podpora pouze starších telefonů s HW GSM modemem
<ul style="list-style-type: none"> • Placená SMS přes bránu s číslem zákazníka (Kubík Free) 	
<ul style="list-style-type: none"> • Možnost odesílat hromadné zprávy 	
<ul style="list-style-type: none"> • Odeslání zprávy v určený čas 	
<ul style="list-style-type: none"> • Import kontaktů 	
<ul style="list-style-type: none"> • Aplikace pro Android 	



Obrázek 1 – Kubík SMS desktopová aplikace

Zdroj: Vlastní zpracování



Obrázek 2 – Kubík SMS mobilní aplikace
Zdroj: Vlastní zpracování

1.2 Daktela

Společnost Daktela s.r.o. je na českém trhu od roku 2005. Specializuje se na návrhy a následnou implementaci komplexních VoIP⁴ telefonních řešení založených na ústřednách Asterisk PBX. Firma je provozovatelem vlastní telefonní platformy pro virtuální ústředny a call centra, umožňuje zákazníkům používat telefonní systém v cloudu.

Mimo hlasových služeb nabízí také ojedinělou službu SIM hosting s podporou SMS. SIM karta zákazníka je zapojena do multiportové GSM brány. Tímto způsobem lze SIM kartu propojit s virtuálním call centrem. Hlavní funkcí, která mě zajímá, je možnost posílání a příjem SMS přes programové API⁵ rozhraní. V kombinaci s virtuální ústřednou či call centrem lze automatizovat zasílání SMS zpráv. Zpráva může být odeslána, pokud dojde k požadované události, jako je například uskutečnění objednávky.

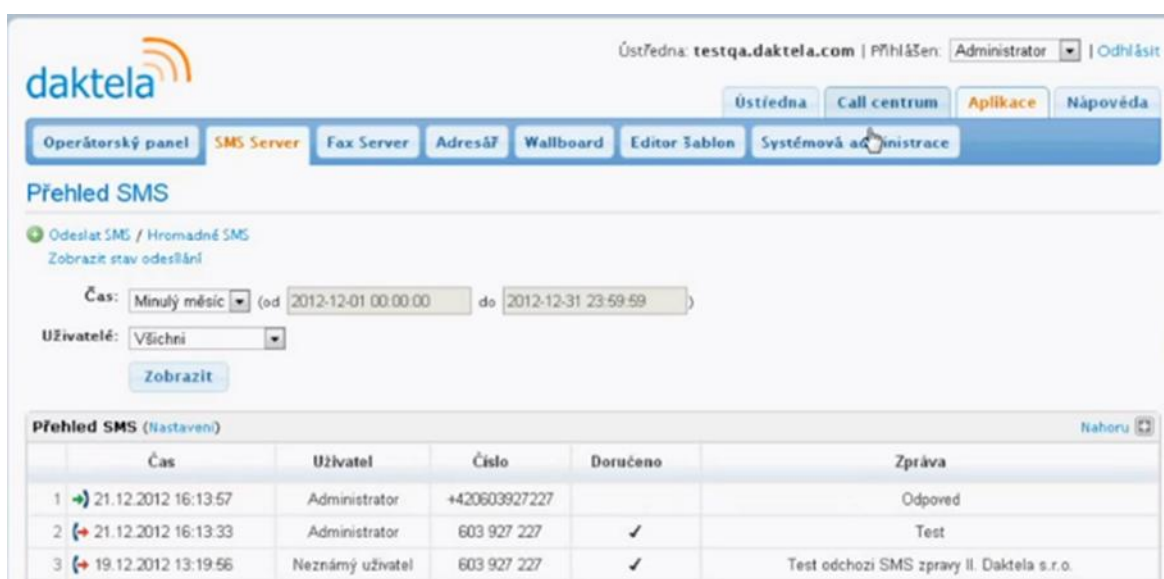
Webové rozhraní (Obrázek 3) Daktela umožňuje pohodlné odeslání SMS zprávy, a také funkcionality jako hromadné rozesílání v určený čas do skupin uživatelů. Další příjemnou vlastností je možnost exportu zpráv do formátů CSV, XLS a TXT. Aktivační poplatek za zprovoznění této služby je 1490 Kč, měsíční poplatek 750 Kč a dalších 390 Kč za API rozhraní. K tomu je nutné samozřejmě připočíst paušální poplatek na SIM kartě. [3]

⁴Voice over Internet Protocol

⁵Application Programming Interface

Tabulka 2 – Zhodnocení aplikace Daktela

Výhody	Nevýhody
<ul style="list-style-type: none"> • SIM hosting • Rozsáhlé API rozhraní • Export zpráv • Notifikace, odesílání zpráv při určené aktivitě • Hromadné odesílání zpráv, skupiny • Odeslání zprávy v určený čas 	<ul style="list-style-type: none"> • Nutnost umístění SIM u provozovatele • Poplatek za využívání API 390 Kč • Měsíční poplatek 750 Kč • Zprovoznění služby za 1490 Kč • Nevýhodné pro menší objem zpráv



Obrázek 3 – Webové rozhraní Daktela

Zdroj: Vlastní zpracování

1.3 Smsbrana.cz

Smsbrana.cz je služba od společnosti Neogenia. Brána je určena pro běžné uživatele i firemní zákazníky. Mezi funkcionality SMS brány patří notifikační SMS a také marketingové zprávy. Pomocí hromadných SMS je možno zaslat zprávy libovolnému počtu příjemců. Další využití tohoto řešení lze nalézt u platebních SMS, které mohou být využity při zpoplatnění služeb, hlasování soutěžích atd. Zprávy se zasílají pomocí webového rozhraní (Obrázek 4), které mimo základní funkcionality umožňuje vést adresáře kontaktů, vytvářet skupiny příjemců, zasílat zprávy v určený čas a také archiv přijatých a odeslaných zpráv. Příjemnou vlastností je také import kontaktů pomocí aplikace SMSbrána.cz pro smartphone. Mobilní aplikace (Obrázek 5) nabízí možnost odeslání placené SMS za cenu 0,87 Kč včetně DPH.

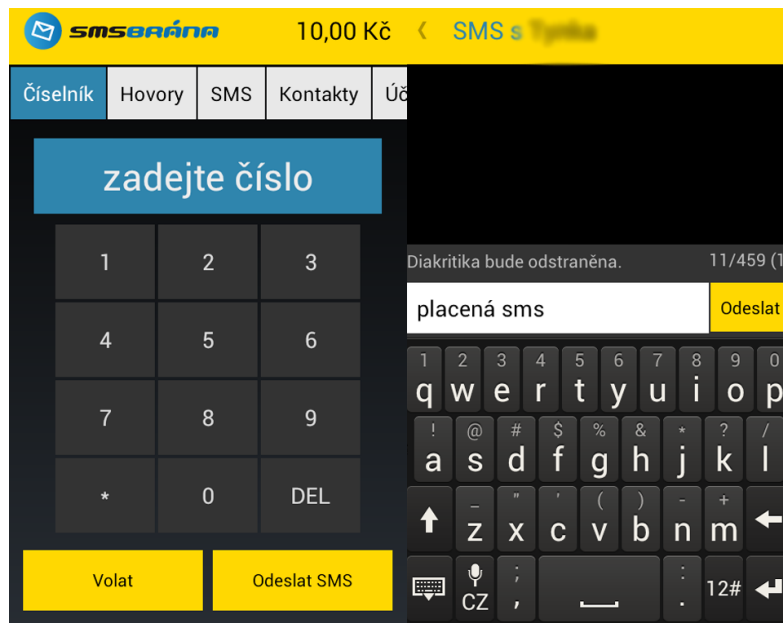
Cena za odeslanou zprávu se liší dle tarifu a dobíté částky. Nejlevnější SMS Smart má cenu od 0,72 Kč, SMS Fast od 0,78 Kč a nejlepší tarif SMS Fast+, který umožňuje odeslání SMS z vlastního čísla, za cenu 1,19 Kč za SMS. Uváděné ceny jsou bez DPH. Cenu za odeslanou SMS lze snížit až o 21% pomocí bonusů za dobíjení. Základní tarif SMS Smart má pevně přidělené číslo bez možnosti vlastního čísla. Rychlost odesílání SMS je od 6 do 30 SMS za sekundu dle tarifu. [4]

Tabulka 3 – Zhodnocení aplikace Smsbrana.cz

Výhody	Nevýhody
<ul style="list-style-type: none"> • Smartphone aplikace • Archiv zpráv 	<ul style="list-style-type: none"> • Cena až 1,19 Kč bez DPH za SMS • Odeslání SMS z cizího čísla přes smartphone aplikaci
<ul style="list-style-type: none"> • Import kontaktů • Notifikace, odesílání zpráv při určené aktivitě 	<ul style="list-style-type: none"> • Krátký text s reklamou u bezplatných zpráv • U SMS není garantováno doručení
<ul style="list-style-type: none"> • Hromadné odesílání zpráv, skupiny • Odeslání zprávy v určený čas • API pro zasílání zpráv 	



Obrázek 4 – Webové rozhraní Smsbrana.cz
Zdroj: Vlastní zpracování



Obrázek 5 – Aplikace pro smartphone SMSBrána.cz
Zdroj: Vlastní zpracování

1.4 Axima Direct SMS

Firma Axima spol s.r.o. působí na trhu komunikačních a informačních systémů od roku 1993. Služba Direct SMS představuje řešení vysokorychlostního odesílání SMS pro firmy. V případě potřeby je možné SMS zasílat rychlostí až 50 SMS za vteřinu díky přímým a zdvojeným konektivitám do SMS center. Stejně jako konkurence podporuje odpovědi a doručky. Odesílání a příjem SMS lze také integrovat do systému zákazníka pomocí API rozhraní.

Stejně jako společnost Daktela nabízí službu SIM hosting. SMS tak zákazník může odesílat z vlastní SIM umístěné v datovém centru. Axima uvádí maximální propustnost 10 SMS za minutu s paušálem 1000 Kč za měsíc. S doručovanými SMS lze dále pracovat např. přeposláním, ukládáním, statistikami atd. Nadstandardní možnosti služby jsou například: rozesílání SMS z databáze zákazníka, rozesílání SMS z xls souboru či napojení libovolného systému na službu Direct SMS. [5]

Tabulka 4 – Zhodnocení Axima Direct SMS

Výhody	Nevýhody
<ul style="list-style-type: none"> • Mnoho nadstandardních služeb 	<ul style="list-style-type: none"> • Absence aplikace na telefon
<ul style="list-style-type: none"> • Rychlé zasílání zpráv 	<ul style="list-style-type: none"> • Základní cena SMS 0,72 Kč
<ul style="list-style-type: none"> • Podpora doruček, odpovědi 	<ul style="list-style-type: none"> • SIM hosting za 1000 Kč měsíčně
<ul style="list-style-type: none"> • API rozhraní 	
<ul style="list-style-type: none"> • Hromadné odesílání zpráv, skupiny 	
<ul style="list-style-type: none"> • Sim hosting 	
<ul style="list-style-type: none"> • Dávkové a časově definované rozesílání SMS 	

1.5 Klikniavolej.cz

Tato webová stránka (Obrázek 7) se zabývá nejen výhodným voláním, ale také hromadným odesíláním SMS. Služba umožňuje funkcionality jako je seznam příjemců a šablona zprávy. Kromě klasického odesílání zpráv nabízí poskytovatel také odesílání SMS zpráv přes vlastní aplikaci (API). Propustnost SMS brány je 30 SMS zpráv za vteřinu a cena jedné zprávy je od 0,72 Kč bez DPH. U vlastního API mě zaujal způsob zabezpečené autentizace. Zabezpečená autentizace probíhá pomocí algoritmu SHA⁶ a je použito parametrů id uživatele, id zprávy a hash hesla. Výsledný hash pak vypadá takto:

```
hash = sha1(user || `:` || id || `:` || sha1(password))
```

Poskytovatel služby preferuje použití zabezpečeného spojení přes HTTPS⁷ s vlastním certifikátem a certifikátem autority. Jednou z největších předností této služby je možnost definování zdrojového telefonního čísla. Příjemce zprávy se tak nedozví, že SMS byla odeslána přes SMS bránu. Příjemnou vlastností je přítomnost doručenek zpráv. Kromě vlastního API je možné využít mobilní aplikaci pro iPhone a Android (Obrázek 6). Mobilní aplikace však postrádá nadstandartní služby, které jsou dostupné při odesílání přes vlastní API. Cena za odeslanou zprávu přes mobilní aplikaci je 1,19 Kč. Kompletní dokumentaci API je možné shlédnout v dokumentu formátu pdf na stránce: http://data.idnes.cz/klikniavolej/pdf/SMS_Gateway.pdf [6]

Tabulka 5 – Zhodnocení aplikace klikniavolej.cz

Výhody	Nevýhody
<ul style="list-style-type: none">• Propustnost 30 SMS za vteřinu	<ul style="list-style-type: none">• Základní cena za SMS 1,19 Kč
<ul style="list-style-type: none">• Propracované API	<ul style="list-style-type: none">• Nelze odeslat SMS v určený čas
<ul style="list-style-type: none">• Podpora doručenek	
<ul style="list-style-type: none">• Hromadné odesílání zpráv, šablony	
<ul style="list-style-type: none">• Vlastní telefonní seznam	
<ul style="list-style-type: none">• Mobilní aplikace pro iPhone, Android	



Obrázek 6 – Mobilní aplikace klikniavolej.cz

Zdroj: [7]

⁶Secure Hash Algorithm

⁷Hypertext Transfer Protocol Secure

1 Vaše telefonní číslo

aaa (+420 775 664 285)

[Přidat další telefonní číslo](#)

2 Volané číslo

+420

Česká republika

3 Volat

klikni a volej

🇸🇰 Volání z mobilu trvale za **1,90 Kč/min.!** 🇸🇰 Volejte na Slovensko za **4,90 Kč/min.!**

Telefonní seznam

Historie hovorů

Můj účet

SMS

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | [vše](#)

Jméno	Telefon 1	Rychlá volba 1	Telefon 2	Rychlá volba 2	
Pepa	+420 123 456 789	+420 910 999 000			Upravit smazat

📄 [Vytisknout](#)
📱 mobil
📞 pevná linka
VLOŽIT NOVÝ KONTAKT ▶

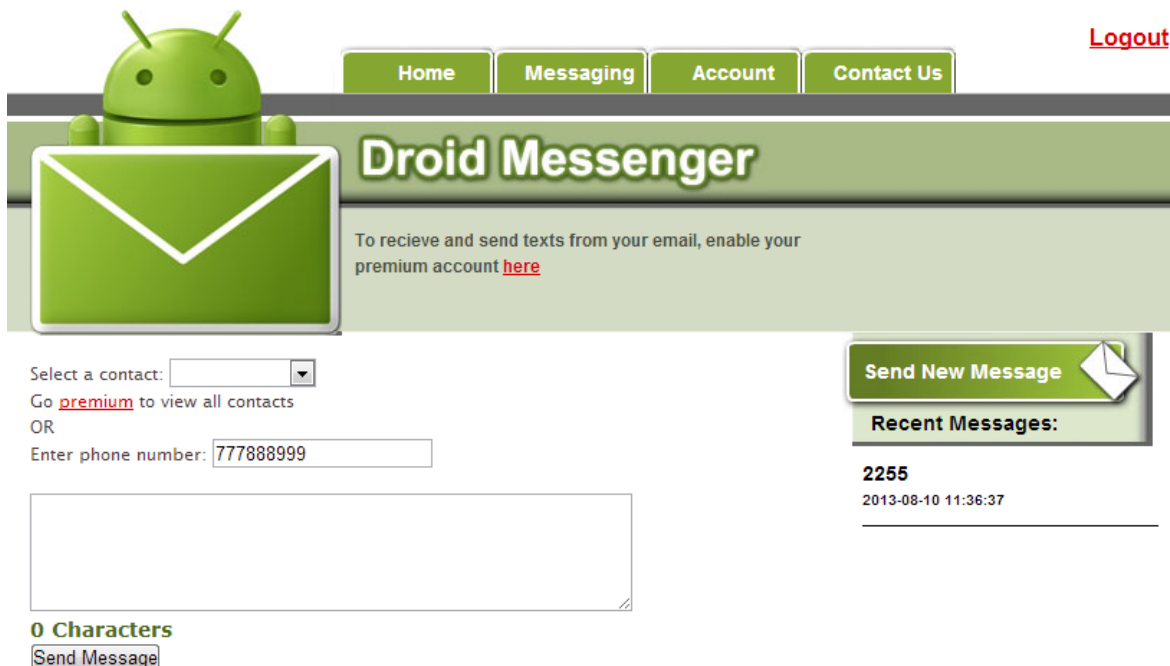
Obrázek 7 – Webové rozhraní aplikace klikniavolej.cz
Zdroj: Vlastní zpracování

1.6 Droid Messenger

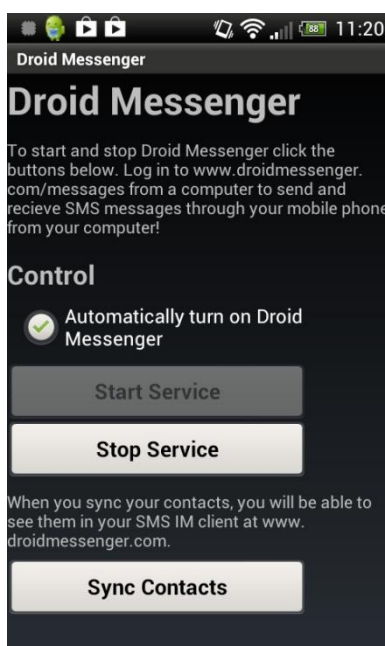
Droid Messenger je ojedinělá služba, která umožňuje propojit mobilní telefon disponující operačním systémem Android s webovým rozhraním. Aplikace pro Android (Obrázek 9) je velice jednoduchá, umožňuje pouze spuštění nebo zastavení služby a synchronizaci kontaktů. Webové rozhraní (Obrázek 8) umožňuje odeslat pohodlně SMS zprávu za cenu dle Vašeho tarifu. Na internetové stránce je také jednoduchý archiv zpráv, kde má uživatel možnost prohlížet příchozí zprávy z posledních tří dnů. V případě placeného účtu \$ 3.50 za měsíc můžete importovat kontakty z telefonu a odesílat i přijímat SMS prostřednictvím emailu. Po placené registraci je uživateli přidělen email, který slouží pro příjem a odesílání SMS zpráv. Jako předmět je uvedeno telefonní číslo příjemce, v těle emailu je potom text zprávy. Nevýhodou programu Droid Messenger je absence API a hromadného odesílání zpráv. [8]

Tabulka 6 – Zhodnocení aplikace Droid Messenger

Výhody	Nevýhody
<ul style="list-style-type: none"> Propojení telefonu s webovým rozhraním Zasílání zpráv dle vlastního tarifu Možnost odesílat a přijímat zprávy na email HTTPS komunikace 	<ul style="list-style-type: none"> Absence API Omezený archiv zpráv \$ 3.50 / měsíc za placený účet Nelze vytvářet nové kontakty ve webovém rozhraní. Nelze odesílat hromadné zprávy Není možné odeslat SMS v určený čas



Obrázek 8 – Webové rozhraní Droid Messenger
Zdroj: Vlastní zpracování



Obrázek 9 – Mobilní aplikace Droid Messenger
Zdroj: Vlastní zpracování

1.7 Remote SMS Sender

Jedná se o jednoduchou aplikaci, která stejně jako Droid Messenger umožňuje jednoduché odesílání zpráv z počítače pomocí mobilního telefonu (Obrázek 10). Pro funkčnost aplikace je potřeba mít počítač i mobilní telefon (platforma Android) ve stejné síti WiFi.

Webové rozhraní načte kontakty z mobilního telefonu a umožňuje odesílání hromadných SMS. Bohužel však chybí archivace zpráv, možnost časovaného odeslání, a vytváření kontaktů přes webové rozhraní. Veřejné API není dostupné. [9]



Obrázek 10 – Mobilní a webové rozhraní Remote SMS Sender

Zdroj: [9]

1.8 O2 SMS Connector

O2 SMS Connector je služba pro rozesílání většího množství SMS zpráv od operátora Telefónica O2 Czech Republic a.s. API rozhraní umožňuje využívat všechny funkcionality spojené s odesíláním SMS. Základní měsíční tarif garantuje rychlost odesílání 1 SMS za sekundu. Propustnost 5 SMS/sec je zpoplatněna částkou 10 000 Kč za měsíc, maximální propustnost 20 SMS/sec je oceněna na 20 000 Kč měsíčně. Cena za odeslání SMS do ostatních českých mobilních sítí je 1,50 Kč. Do zahraničí je to dvojnásobná cena. Obrázek 11 znázorňuje cenové zpoplatnění tarifů. [10]

- Technická dokumentace k rozhraní je dostupná pouze v angličtině na adrese: https://www.o2.cz/_pub/90/c9/37/43182_580775_Web_services_interface_2_1.pdf

Název tarifu	Měsíční paušál (Kč)	Počet SMS do sítě O2 zdarma v rámci paušálu	Cena za další SMS do sítě O2 (Kč)	SMS do ostatních českých mobilních sítí (Kč)	SMS do zahraničních mobilních sítí (Kč)	
SMS Connector I	350	300	301-8000 8001+	0,90 (1)	1,50	3
SMS Connector II	1500	1400	1401-12000 12001+	0,80 (0,90)	1,50	3
SMS Connector III	15000	15000	15001-50000 50001-200000 200001+	0,80 0,70 0,60	1,50	3

Obrázek 11 – O2 SMS Connector, tabulka cen
Zdroj: [10]

Tabulka 7 – Zhodnocení O2 SMS Connector

Výhody	Nevýhody
<ul style="list-style-type: none"> • Využití všech funkcionalit při zasílání SMS • Propracované API 	<ul style="list-style-type: none"> • Vysoká cena 1,50 Kč za odeslání SMS do cizí mobilní sítě • Měsíční paušál • Technická dokumentace v angličtině • Nutnost vytvoření vlastní aplikace • Zvýšení propustnosti za vysoké měsíční poplatky 10 000 – 20 000 Kč.

1.9 SMSC Direct

Se službou SMSC Direct uživatel získá přímé napojení na SMS centrum T-Mobile. Stejně jako předchozí aplikace slouží pro rozesílání velkého množství SMS. Garantovaná kapacita odeslání SMS zpráv je až 36 000 za hodinu (10 SMS/sec). Přenos zpráv je vysoce spolehlivý díky oddělenému provozu od veřejné GSM sítě. Jako provozní nástroj k této službě je určená aplikace SMS/MMS Manager, která poskytuje širokou škálu možností pro správu a zpracování SMS zpráv. Rozhraní pro implementaci do vlastního software se nazývá EMI⁸. Aktivace služby je zpoplatněna částkou 20 000 Kč. Navýšení propustnosti je zpoplatněno takto: 2 SMS/s: 45 000 Kč, 5 SMS/s: 80 000 Kč a 10 SMS/s za částku 120 000 Kč. Všechny uváděné ceny jsou bez DPH. Ceník za odchozí SMS zobrazuje Obrázek 12. [11]

⁸External Machine Interface

Pro provoz služby je zapotřebí toto vybavení:

- FrameRelay linka nebo připojení na internet.
- Hardware pro provoz aplikace zpracovávající SMS zprávy.
- SMS aplikace podporující komunikaci EMI protokolem.

Ceník odchozích SMS	
1. - 50 000. SMS v síti T-Mobile	0,80 Kč (s DPH 0,97 Kč) Prvních 20 000 SMS je zdarma.
50 001. - 300 000. SMS v síti T-mobile	0,70 Kč (s DPH 0,85 Kč)
300 001. - 1 000 000. SMS v síti T-Mobile	0,60 Kč (s DPH 0,73 Kč)
1 000 001. a další SMS v síti T-Mobile	0,50 Kč (s DPH 0,61 Kč)
1. a další SMS v ostatních mobilních sítích	1,50 Kč (s DPH 1,82 Kč)

Obrázek 12 – SMSC Direct, ceník odchozích SMS

Zdroj: [11]

Tabulka 8 – Zhodnocení SMSC Direct

Výhody	Nevýhody
<ul style="list-style-type: none"> • Využití všech funkcionalit při zaslání SMS • Propracované API (EMI) • Aplikace SMS/MMS Manager 	<ul style="list-style-type: none"> • Vysoká cena 1,50 Kč bez DPH za odeslání SMS do cizí mobilní sítě • Měsíční paušál 20 000 Kč bez DPH • Technická dokumentace v angličtině
	<ul style="list-style-type: none"> • Zvýšení propustnosti za vysoké měsíční poplatky 45 000 – 120 000 Kč bez DPH. • Komunikace s datacentrem přes složité EMI

1.10 Shrnutí stávajících řešení

Výše vypsane služby pro rozesílání SMS zpráv ukazují, že i při zakoupení výhodného tarifu je cena za odeslanou zprávu stále vysoká a pohybuje se v rozmezí 0,60 Kč až 1,50 Kč bez DPH. K tomu je nutné připočítat často velmi vysoké náklady za měsíční paušál. Nejméně výhodná jsou paradoxně řešení pro rozesílání SMS zpráv od mobilních operátorů. Současné aplikace, které zasílají zprávy přes mobilní telefon, jsou velice jednoduché. Neumožňují nutné vlastnosti pro použití ve firemní sféře – možnost hromadného odesílání

SMS zpráv, určení času odesílané zprávy, potvrzení doručení či archivaci odeslaných zpráv. V neposlední řadě aplikace, které rozesílají SMS zprávu přes telefon neposkytují vlastní API, které je možné implementovat do firemních systémů.

Z výše uvedených poznatků je patrné, že firma, která chce provozovat systém pro rozesílání SMS zpráv si musí vyhradit nemalý finanční obnos na tuto činnost, nebo řešit komunikaci přes alternativní komunikační kanály jako je např. email. Ideálním řešením by byla vlastní aplikace na mobilní telefon rozesílající SMS přes GSM síť propojená se serverovou částí poskytující data přes API. Toto řešení je obsahem praktické části této diplomové práce.

2 Dostupnost API pro odesílání SMS u různých mobilních platform

V této části se budu zabývat, jakým způsobem lze v různých mobilních platformách odesílat SMS zprávy přes GSM síť. Hlavním úkolem této části je zjištění, zda lze ze všech mobilních platform odesílat SMS zprávy bez interakce uživatele.

2.1 Android

Programovací jazyk Android používá od API verze 4 třídu `SMSManager`, která umožňuje odesílání datových a textových zpráv. Pro odesílání SMS zpráv slouží metoda **`sendTextMessage`**.

```
public final void sendTextMessage (String destinationAddress, String
scAddress, String text, PendingIntent sentIntent, PendingIntent
deliveryIntent)
```

Tabulka 9 – Parametry metody `sendTextMessage`

Parametry	Význam
<code>destinationAddress</code>	Telefonní číslo příjemce
<code>scAddress</code>	Telefonní číslo odesílatele, může být null
<code>text</code>	Text zprávy
<code>sentIntent</code>	Po odeslání zprávy nebo selhání při odeslání je odeslán Broadcast signál, který je dále zpracován, může být null
<code>deliveryIntent</code>	Po doručení zprávy je odeslán Broadcast signál, který je dále zpracován, může být null

Pro odeslání delší zprávy než 160 znaků je třeba volat metodu **`sendMultipartTextMessage`**. Zde je text zpráv rozdělen na příslušný počet SMS s maximální délkou 160 znaků a poté je vložen do pole. Broadcast signály pro odeslání a potvrzení doručení zprávy jsou také v poli. Pro každou část SMS je přiřazen signál. Textové zprávy je možné odesílat **automaticky**. [12]

2.2 Windows Phone

Odesílání SMS zpráv ze systému Windows Phone zajišťuje třída `SmsComposeTask`. Pro zpřístupnění této třídy je třeba importovat knihovnu `Microsoft.Phone.Tasks`. Odeslání SMS zprávy ze systému Windows Phone **vyžaduje potvrzení uživatele**. [13]

```
using Microsoft.Phone.Tasks;

        SmsComposeTask smsComposeTask = new SmsComposeTask();
smsComposeTask.To = „777123456“;
        smsComposeTask.Body = „Tělo zprávy“;
        smsComposeTask.Show();
```

2.3 BlackBerry

Pro odesílání SMS zpráv na platformě BlackBerry je určené rozhraní `MessageConnection`. **Automatické odesílání SMS** je možné. Jednoduché odeslání SMS zprávy lze uskutečnit například takto:[14]

```
private static final int MAX_PHONE_NUMBER_LENGTH = 32;
private String mobilenNumber = "777123456";
private String message = "Tělo zprávy";
private DatagramConnection _dc = null;
private static String _openString = "sms://";

public SendSms()
{
    try {
msgConn = (MessageConnection) Connector.open("sms://+" + mobilenumber);
        TextMessage text = (TextMessage)
msgConn.newMessage(MessageConnection.TEXT_MESSAGE);
        text.setPayloadText(message);
        msgConn.send(text);
        msgConn.close();
    }
}
```

2.4 Symbian OS

Operační systém Symbian umožňuje jednoduché odesílání SMS zpráv **bez nutnosti potvrzení** uživatelem pomocí Qt SDK⁹. Qt SDK poskytuje kompletní sadu nástrojů a knihovny C++ nezbytné pro vývoj, testování a ladění aplikací pro Symbian, MeeGoo a desktopové aplikace. [15]

- Pro odesílání SMS je třeba importovat knihovny `XQMessaging` a `XQMessage`.

```
#include "XQMessaging.h"
#include "XQMessage.h"
```

⁹ Software development kit od společnosti Nokia

- Samotné odeslání zprávy je pomocí Qt knihovny velice jednoduché:

```
void QtSMSOperation::sendSMS ()
{
    XQMessaging messaging;
    XQMessage message(QStringList("777123456"),QString("Ahoj"));

    if (messaging.send(message) == XQMessaging::NoError)
    {
        QMessageBox msgBox;
        msgBox.setText(tr("SMS was sent successfully"));
        msgBox.exec();
    }
}
```

2.5 iOS

Od iOS 4 je na platformě iOS dostupná třída MFMessageComposeViewController pro odeslání SMS zpráv. Odeslání SMS zprávy ze systému od společnosti Apple **vyžaduje potvrzení uživatele**. Uvedený kód zobrazí pouze obrazovku s předpřipraveným textem zprávy a čísla příjemců. [16]

Jednoduché použití pro odeslání zpráv je takovéto:

```
NSArray *recipients = @[@"12345678", @"72345524"];
NSString *message = [NSString stringWithFormat:@"Testovací SMS "];

MFMessageComposeViewController *messageController =
[[MFMessageComposeViewController alloc] init];
messageController.messageComposeDelegate = self;
[messageController setRecipients:recipients];
[messageController setBody:message];

[self presentViewController:messageController
animated:YES completion:nil];
```

2.6 Hodnocení

Operační systémy iOS a Windows Phone nepodporují automatické odeslání SMS zpráv. Odeslání textové zprávy musí být u těchto systémů vždy potvrzeno stisknutím tlačítka pro odeslání. Pro SMS bránu jsou tedy tyto dva operační systémy nevhodné. Ostatní systémy Android, Symbian a BlackBerry jsou pro automatické rozesílání zpráv vhodné. Nejjednodušší rozhraní pro programové odeslání SMS zpráv mají operační systémy Android a Symbian.

3 Problematika spamu na mobilních telefonech

S hromadným rozesíláním textových zpráv souvisí problematika spamu, proto považují za vhodné tuto tematiku zmínit. Spam je nevyžádané sdělení, nejčastěji reklamního charakteru. Původně se tato technika používala pro nevyžádané reklamní emaily, později se rozšířila do diskuzních fór, komentářů, nebo instant messaging programů¹⁰. S nástupem mobilních telefonů se spam rozšířil také do mobilních telefonů. Mobilní spam je nejspíše nejprotivnější verze nevyžádané pošty, protože uživatel obdrží zvukové, vizuální či vibrační upozornění, které uživatele více nutí zprávu zaregistrovat a přečíst. Dle údajů z roku 2008 mělo dojít na české mobilní telefony 1,5 miliardy spamových SMS. [17]

3.1 Prémiové textové zprávy

Spam na mobilních telefonech nemusí být pouze obtěžující reklama, ale také hrozba v podobě nevyžádaných prémiových SMS, které jsou zpoplatněny desítkami korun. Obvykle se prostřednictvím prémiových SMS objednává nějaká zpoplatněná služba, nebo zasílá soutěžní SMS. V nevyžádané prémiové zprávě je uživatel vyzván k zaslání kódu na uvedené telefonní číslo. Tím je uživateli stržena částka za službu, kterou si tím zakoupil.

3.2 Podvodná zpráva pro zaslání kódu

V poslední době je aktuální téma rozesílání nevyžádaných SMS, které vybízí uživatele k zaslání nějakého kódu na uvedené telefonní číslo. Tímto způsobem podvodníci objednávají nejrůznější služby na úkor okradeného důvěřivého člověka. Tvar takové SMS může být v například v takovéto podobě:

„Dobrý den, omylem jsem odeslal SMS s mojí registrací na Vaše číslo, mohu Vás poprosit o přeposlání SMS na číslo 777111111. Předem děkuji za pochopení, jedná se o mé osobní údaje. S pozdravem Karel Novák“.

3.2.1 Princip objednání služby s platbou přes mobil

1. Objednání služby.
2. Uvedení telefonního čísla, ze kterého bude služba placena.
3. Operátor zašle potvrzovací kód na uvedený telefon.
4. Zadání potvrzovacího kódu do online formuláře.
5. Služba je aktivována, peníze jsou odečteny z telefonu.

3.2.2 Princip podvodného objednání služby

1. Podvodník si objedná službu
2. Uvede telefonní číslo oběti, za kterého bude služba uhrazena
3. Operátor zašle potvrzovací kód oběti.
4. Podvodník si vyžádá kód od oběti pod záminkou „mylného zadání čísla“.
5. Pokud oběť zašle kód podvodníkovi, zadá jej do online formuláře.
6. Podvodník čerpá službu za peníze oběti. [18]

¹⁰Odesílání a přijímání zpráv v reálném čase.

3.2.3 Jak se bránit

Tuto podvodnou praktiku lze dokázat pouze velice obtížně. Objednávat lze takto například i kredit mobilního telefonu, a tak se škody mohou pohybovat v řádech tisíců korun. Pokud uživateli přijde SMS s požadavkem na přeposlání jakéhokoliv kódu, v žádném případě by na takovou zprávu neměl reagovat. Pokud se někdo opravdu spletl při zadávání čísla, může si nechat zaslat kód znovu na správné číslo. Pokud uživatel nechce naletět takovému útočníkovi, může si u mobilního operátora zablokovat prémiové SMS zprávy. O podvodných praktikách může uživatel vyrozumět Český telekomunikační úřad, který na základě stížností může na podvodníka podat trestní oznámení.

3.3 Antispamová legislativa

Problematiku spamu řeší Zákon o některých službách informační společnosti (č. 480/2004 Sb.), který je v platnosti od 7. září 2004. Tento zákon vyžaduje prokazatelný souhlas příjemce zprávy. Zákon byl postupně novelizován v letech 2005, 2006, 2007 a 2011. Zákon byl vytvořen dle směrnice Evropského společenství č. 2000/31/ES. Spam je definován jako obchodní sdělení, což jsou všechny formy sdělení určeného k přímé či nepřímé podpoře zboží či služeb nebo image podniku fyzické či právnické osoby. Zákon se vztahuje na elektronické prostředky a elektronické nástroje dle zákona č. 137/2006 Sb., o veřejných zakázkách § 149:

„Elektronickými nástroji se pro účely tohoto zákona rozumí programové vybavení, případně jeho součásti, které jsou spojeny se sítí nebo službou elektronických komunikací a umožňují prostřednictvím této sítě nebo služby provádění úkonů v elektronické podobě“ [19]

Elektronické nástroje tedy zahrnují nejen internetový spam, ale také další formy elektronické komunikace (SMS, telemarketing). Dozorem nad dodržováním zákona byl pověřen **Úřad pro ochranu osobních údajů**. [20]

Obchodní sdělení může prodejce zaslat když:

a) je adresátem jeho zákazník, i) který zaslání podobných sdělení v minulosti neodmítl, ii) sdělení se týká obdobného zboží či služeb

b) adresát obchodníkovi poskytl informovaný souhlas.

3.3.1 Informovaný souhlas

Informovaný souhlas nemůže prodejce získat zasláním pouhého souhlasného e-mailu od zákazníka. Tento způsob informování byl sice zákonem vyňat z definice obchodního sdělení, ale podle současné legislativy neobsahuje taková zpráva dostatek informací. Informovaný souhlas k zaslání obchodního sdělení lze získat například pomocí formuláře na webových stránkách prodejce.

3.4 Spam a vymahatelnost

Členství České republiky v EU přináší výhodu v boji proti spamu, protože je možné postihnout odesílatele nevyžádaného reklamního sdělení z jakékoliv země v EU. Většina spamu však přichází z jiných částí světa a vymáhání práva v této oblasti je potom prakticky nemožné.

3.5 Trest za rozesílání spamu

Dle zákona č. 480/2004 Sb. § 11 hrozí právnické osobě, která hromadně nebo opakovaně šíří spam **pokuta do výše 10 000 000 Kč**. Fyzická osoba může být postižena za trestný čin hromadného nebo opakovaného šíření obchodního sdělení bez souhlasu adresáta pokutou **do výše 100 000 Kč**.

3.6 Stalking

Předcházející odstavce se zabývají pouze zasíláním nevyžádaného reklamního sdělení. Co když se ale jedná o zprávy, které nesou obtěžující obsah, který není reklamního charakteru? V tom případě se jedná o provozování stalkingu. Stalking představuje **zlovolné a úmyslné pronásledování a obtěžování jiné osoby**. Od 1. Ledna 2010 se touto trestnou činností zabývá trestní zákon č. 40/2009 Sb. § 354. Tento zákon se vztahuje i na obtěžování prostřednictvím prostředků elektronické komunikace.

3.6.1 Trest za provozování stalkingu

Zatímco u rozesílání spamu hrozí „pouze“ finanční sankce do 10 000 000 Kč, u obtěžování a úmyslného pronásledování pomocí např. SMS zpráv hrozí **odnětí svobody až na tři roky**, pokud je čin spáchán vůči více lidem, dítěti nebo těhotné ženě. V ostatních případech hrozí útočníkovi **odnětí svobody na dobu až jeden rok** nebo **zákaz činnosti**.
[21]

3.6.2 Znění zákona č.40/2009 Sb. § 354 Nebezpečné pronásledování

„Kdo jiného dlouhodobě pronásleduje tím, že vyhrožuje ublížením na zdraví nebo jinou újmou jemu nebo jeho osobám blízkým, vyhledává jeho osobní blízkost nebo jej sleduje, vytrvale jej prostřednictvím prostředků elektronických komunikací, písemně nebo jinak kontaktuje, omezuje jej v jeho obvyklém způsobu života, nebo zneužije jeho osobních údajů za účelem získání osobního nebo jiného kontaktu a toto jednání je způsobilé vzbudit v něm důvodnou obavu o jeho život nebo zdraví nebo o život a zdraví osob jemu blízkých, bude potrestán odnětím svobody až na jeden rok nebo zákazem činnosti. Odnětím svobody na šest měsíců až tři roky bude pachatel potrestán, spáchá-li čin vůči dítěti nebo těhotné ženě, se zbraní, nebo nejméně dvěma osobami.“[22]

3.7 Účinnost antispamového zákona

Úřad pro ochranu osobních údajů obdržel v roce 2012 celkem 7993 stížností na spam. Ačkoliv právníkům osobám hrozí pokuty až 10 000 000 Kč, čeští spameři dostali v tomto roce pokutu v součtu pouhých **147 000 Kč**.

Může za to benevolentní zákon č. 480/2004 Sb. o některých službách informační společnosti. Ten mimo jiné říká, že nelze zakročit proti nepodnikatelskému subjektu. Obtěžující emaily od politických stran, občanských sdružení, a různých kazatelů jsou mimo pravomoc tohoto zákona.

Další klíčkovou v zákoně je možnost zasílání obchodních sdělení zákazníkovi, od kterého fyzická nebo právnická osoba získala kontakt např. při uskutečnění obchodu nebo registraci. Pokud se obchodní sdělení týkají vlastních nebo obdobných výrobků a služeb, je obchodník oprávněn zasílat spam do té doby, než příjemce vysloví nesouhlas s jejich dalším zasíláním.

Problémem je také skutečnost, že Úřad pro ochranu osobních údajů může udělit pokutu pouze právníkům subjektům sídlícím v členském státě Evropské unie a smí potrestat pouze odesílatele spamu, nikoliv toho, v jehož prospěch byl spam odeslán. V praxi si tak česká firma může najmout pro rozesílání nevyžádaných zpráv firmu z oblasti nepodléhající Evropské unii a je prakticky nepostižitelná.

Účinnosti antispamového úřadu neprospívá ani průměrná doba vyřízení žádosti, jež činí v průměru deset měsíců. Agendě nevyžádaných obchodních sdělení se věnuje pouze jeden z inspektorátů, ve kterém působí jediný inspektor. Ten má k dispozici pět dalších pracovníků. [23]

4 Analýza a návrh aplikace

V této části jsou stanoveny požadavky na mobilní a serverovou aplikaci a jsou představeny použité technologie. Při analýze bylo využito některých metodik UP¹¹. Po prvotním stanovení požadavků na aplikaci je nutné vytvořit diagramy tříd a navrhnout databázový model. Pro návrh webové aplikace jsou použity diagramy aktivit.

4.1 Požadavky na aplikaci

Základní myšlenkou je vytvoření mobilní aplikace, která bude komunikovat se serverovou částí, odkud bude načítat SMS zprávy určené k odeslání. Po načtení je zpráva odeslána z mobilního telefonu přes GSM síť. Na mobilním telefonu bude sjednán tarif pro bezplatné odesílání SMS zpráv do všech českých sítí.

Serverová část bude umožňovat registraci uživatele a pohodlné odesílání hromadných textových zpráv. U odesílaných zpráv bude možné určit čas odeslání. Bude také umožňovat vkládání a editaci kontaktů. Odeslané zprávy jsou ukládány do archivu. Základní funkcionality serverové části budou dostupné pomocí veřejně dostupného API. API umožní implementaci SMS brány do vlastních systémů a programů.

4.2 Funkční požadavky

- Uživateli je umožněno zasílat SMS zprávy přes webové rozhraní.
- Mobilní aplikace pro vybranou platformu bude komunikovat se serverovou částí.
- Mobilní aplikace bude SMS zprávy rozesílat určeným příjemcům.
- Odeslání SMS zprávy je potvrzeno do serverové části.
- Serverová část bude disponovat veřejně dostupným API, které bude poskytovat základní funkce serverové části.
- Systém bude umožňovat registraci uživatele a editaci jeho účtu.
- Serverová část bude disponovat odesíláním hromadných SMS zpráv.
- Serverová část bude umožňovat určit čas odeslání textové zprávy.
- Odeslané SMS zprávy jsou archivované a jejich odesílatel k nim má přístup.
- Uživatel je informován o tom, kdy byla SMS zpráva odeslána.
- Webové rozhraní bude poskytovat vytváření adresářů a telefonních kontaktů.

4.3 Nefunkční požadavky

- Dostupnost – aplikace musí mít dostupnost alespoň 99,5 %.
- Spolehlivost – konzistence aplikace a všech prováděných transakcí.
- Bezpečnost – komunikace přes API musí být zabezpečena autentizací a následnou autorizací uživatele. Odesílatel zprávy bude mít důkaz o doručení a příjemce si je jist odesílatelovou identitou.
- Propustnost SMS brány musí činit alespoň 10 odeslaných SMS za minutu.

¹¹Unified Process

- Paralelní připojení více uživatelů – systém musí být schopný zpracovat požadavky od více uživatelů najednou

4.4 Klady a zápory aplikace

Tabulka 10 – Zhodnocení navrhované aplikace

Výhody	Nevýhody
<ul style="list-style-type: none"> • Možnost využití vhodného tarifu pro bezplatné odesílání SMS 	<ul style="list-style-type: none"> • Mobilní telefon musí mít přístup na internet
<ul style="list-style-type: none"> • Veřejně dostupné API s dokumentací 	<ul style="list-style-type: none"> • Zprávy jsou odesílané z jednoho čísla
<ul style="list-style-type: none"> • Archivace odeslaných zpráv 	
<ul style="list-style-type: none"> • Informace o odeslání zprávy 	
<ul style="list-style-type: none"> • Možnost vytvářet vlastní kontakty a skupiny 	
<ul style="list-style-type: none"> • Časované odeslání zprávy 	
<ul style="list-style-type: none"> • Pohodlné odeslání hromadných zpráv 	
<ul style="list-style-type: none"> • Dlouhé textové zprávy 	
<ul style="list-style-type: none"> • Mobilní aplikace obsahuje jednoduchou statistiku vytižení 	

4.5 Tarify pro neomezené odesílání SMS

Předpokladem pro výhodnost odesílání zpráv přes navrženou aplikaci je neomezený tarif pro rozesílání SMS zpráv do všech sítí. Uvedu zde výčet nejvýhodnějších dostupných tarifů od tří hlavních českých operátorů dostupných k srpnu 2013.

Tabulka 11 – Tarify pro neomezené zaslání SMS

Operátor	Tarif	Cena včetně DPH
Vodafone	RED (se smlouvou)	699 Kč
Vodafone	RED (bez smlouvy)	874 Kč
O2	FREE CZ (se smlouvou)	749 Kč
O2	FREE CZ (se smlouvou do 26 let)	699 Kč
O2	FREE CZ (se smlouvou od 60 let)	699 Kč
T-mobile	S námi síť nesít' (se smlouvou)	749 Kč
T-mobile	S námi síť nesít' bez závazků	899 Kč

4.6 Porovnání navržené aplikace s konkurencí

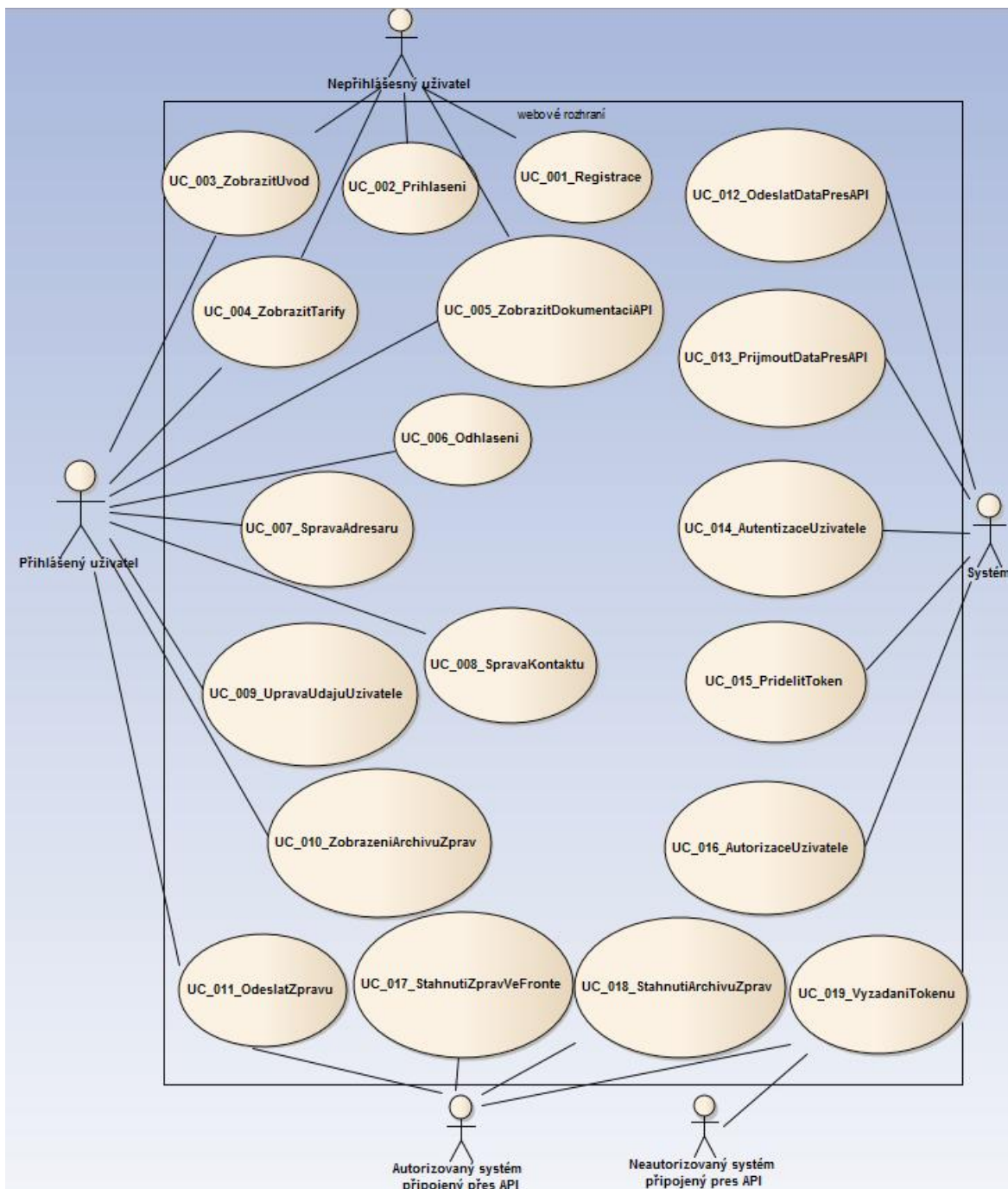
Navržená aplikace je v porovnání s výčtem stávajících řešení z kapitoly Dostupná řešení na trhu výhodná v tom, že nabízí řešení pro hromadné rozesílání textových zpráv za nejnižší možnou cenu. Je to z toho důvodu, že na telefonu rozesílajícím SMS zprávy stačí zakoupit neomezený tarif pro odesílání SMS do všech sítí a o počet odeslaných textových zpráv se uživatel nemusí starat. Aplikace počítá s tím, že ji bude využívat více uživatelů zároveň, a tak se stanou náklady zanedbatelné.

Webové rozhraní poskytuje všechny potřebné funkce, které často nenabízí ani mnohem dražší komerční řešení jako je hromadné odesílání zpráv, možnost odeslat zprávu v určený čas nebo vlastní adresář s kontakty. Samozřejmostí není u konkurence ani archiv odeslaných zpráv, které aplikace nabízí. Velice příjemnou vlastností je veřejně dostupné API, které na rozdíl od většiny konkurenčních řešení umožňuje integrovat odesílání SMS zpráv do vlastní aplikace.

Nevýhodou této aplikace může být skutečnost, že jsou zprávy odesílány z jednoho telefonního čísla. To však pro rozesílání různých sdělení a informací není až tak závažné. V případě obav ze zneužití čísla se skýtá možnost automaticky připojit na konec textové zprávy podpis, který by musel být schválen administrátorem. Předpokladem pro funkčnost aplikace je připojení mobilního telefonu k internetu, dostupný GSM signál a stále nabitá baterie. To však v případě vhodného neměnného umístění mobilního telefonu nečiní problém.

4.7 Use Case diagram webové aplikace

Na základě požadavků na aplikaci byly vytvořeny Use Case digramy webové aplikace a aplikace na mobilní platformu. Hlavním účelem je zachycení všech aktérů, kteří vstupují do systému a služeb, které systém poskytuje.



Obrázek 13 – Use Case diagram webové aplikace

Zdroj: Vlastní zpracování

Obrázek 13 zobrazuje zjednodušený Use Case diagram, ve kterém se vyskytuje celkem pět aktérů. Webové rozhraní poskytuje rozdílný obsah přihlášenému a nepřihlášenému uživateli. Stejně tak je tomu u systému, který je připojený přes API. Bez řádné autorizace pomocí tokenu mu nejsou umožněny žádné akce.

UC_001_Registrace – Nepřihlášený uživatel má možnost registrace do systému.

UC_002_Prihlaseni – Nepřihlášený uživatel se musí přihlásit do systému, aby získal výhody spojené se založením účtu. Funkcionality získané přihlášením jsou zobrazeny na obrázku výše.

UC_003_ZobrazitUvod – Jakýkoli uživatel může zobrazit úvodní stránku webové prezentace.

UC_004_ZobrazitTarify – Kterýkoli uživatel může zobrazit nabídku tarifů pro odesílání SMS zpráv.

UC_005_ZobrazitDokumentaciAPI – Libovolný uživatel může zobrazit dokumentaci k veřejně dostupnému API.

UC_006_Odhlaseeni – Přihlášený uživatel se může ze systému odhlásit.

UC_007_SpravaAdresaru – Autorizovaný uživatel smí prohlížet, vytvářet, editovat a mazat adresáře, které uchovávají kontakty.

UC_008_SpravaKontaktu – Autorizovaný uživatel smí prohlížet, vytvářet, editovat a mazat kontakty. Kontakty jsou dle uživatele přiřazovány do patřičného adresáře.

UC_009_UpravaUdajuUzivatele – Uživatel si může upravit svoje registrační údaje. Úprava kontaktů je podmíněna zadáním správného hesla.

UC_010_ZobrazeniArchivuZprav – Odeslané zprávy jsou archivovány a přihlášený uživatel si může prohlížet svůj archiv.

UC_011_OdeslatZpravu – Zasílání textových zpráv je hlavní funkcionalitou SMS brány. Uživatel si může zvolit čas odeslání a určit zprávu více příjemcům. Po odeslání je zpráva uložena do fronty zpráv čekajících na odeslání. Tato fronta je zpracovávána mobilním telefonem.

UC_012_OdeslatDataPresAPI – SMS brána poskytuje veřejně dostupné API, které poskytuje SMS zprávy uživatele čekající na odeslání a archiv odeslaných zpráv. Přes API komunikuje také mobilní telefon, který rozesílá zprávy přes GSM síť. Mobilní telefon vlastní administrátorský účet, který načte zprávy všech uživatelů čekající na odeslání.

UC_013_PrijmoutDataPresAPI – Uživatelé mají možnost odesílat SMS zprávy přes API a integrovat tak zasílání SMS do svého systému. Serverová část tyto příchozí zprávy zpracuje a uloží je do fronty zpráv čekajících na odeslání.

UC_014_AutentizaceUzivatele – Systém porovnává přihlašovací údaje uživatele s údaji uloženými v databázi. Pokud je autentizace úspěšná, následuje autorizace, ve které je uživateli přiděleno patřičné oprávnění.

UC_015_PridelitToken – Přidělování tokenu slouží k zabezpečení komunikace přes API rozhraní. Při zahájení komunikace proběhne autentizace uživatele komunikujícího přes rozhraní. Pokud je autentizace úspěšná, je komunikujícímu vrácen vygenerovaný token, který má platnost 5 minut. Tímto vygenerovaným řetězcem se společně s uživatelským jménem prokazuje po dobu platnosti tokenu. Po vypršení tokenu je připojený uživatel opět vyzván k autentizaci. Pokud je autentizace úspěšná, je uživateli přidělen nový token.

UC_016_AutorizaceUzivatele – Po úspěšné autentizaci je uživateli přiděleno patřičné oprávnění.

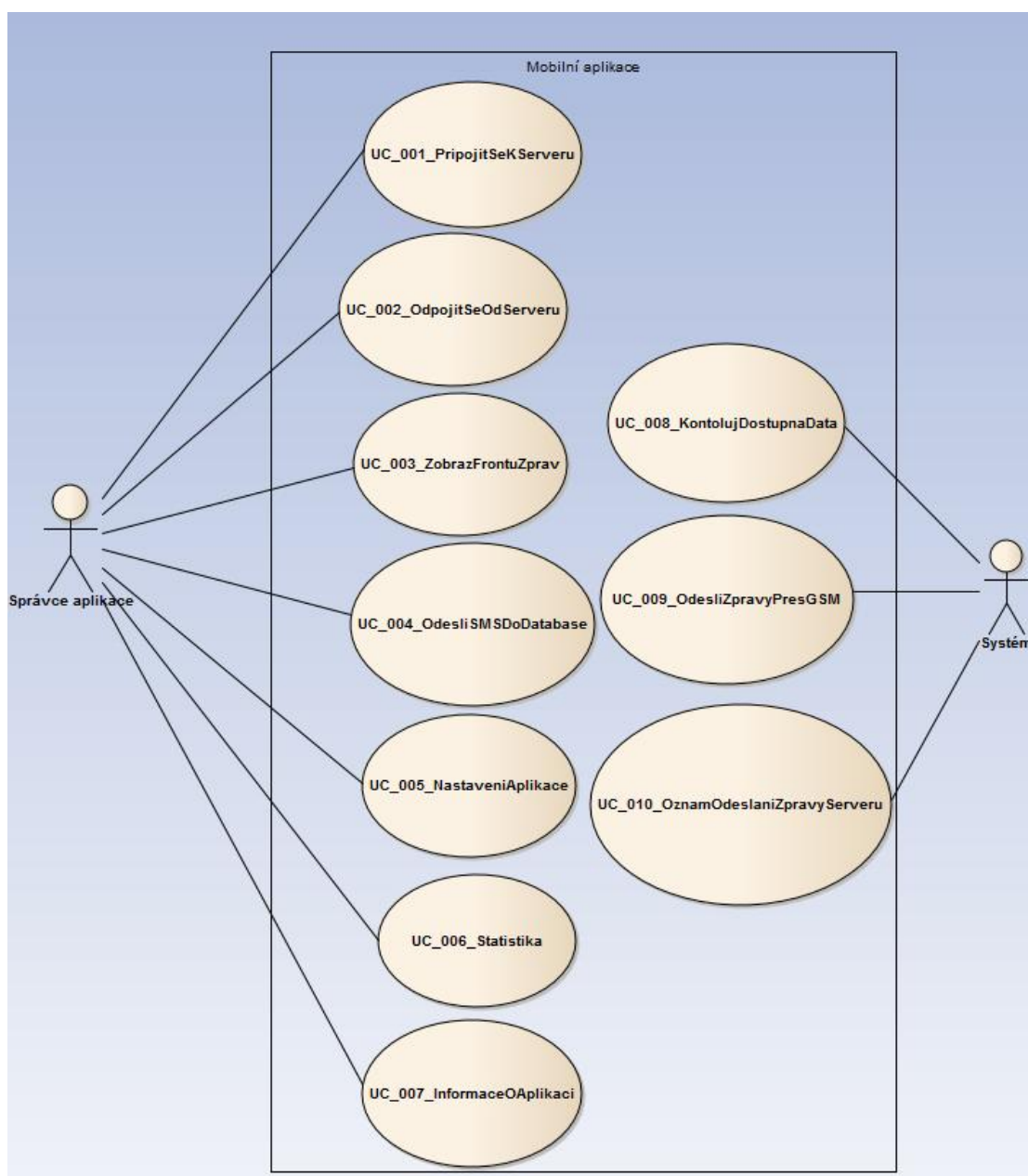
UC_017_StahnutiZpravVeFronte – Uživatel připojený přes API rozhraní si touto cestou může vyžádat své SMS zprávy čekající na odeslání.

UC_018_StahnutiArchivuZprav – Uživatel připojený přes API rozhraní si touto cestou může vyžádat své SMS zprávy čekající na odeslání.

UC_019_VyzadaniTokenu – Pokud uživatel komunikuje přes API a nemá token, nebo vypršela platnost přiděleného tokenu, požádá s přiděleným API klíčem o nový token.

4.8 Use Case diagram mobilní aplikace

Use Case diagram mobilní aplikace (Obrázek 14) je jednodušší, než je tomu u serverové části. Mobilní aplikace má pouze dva uživatele. Správce aplikace manuálně vytváří a ruší spojení se serverem. Dále má k dispozici základní funkcionality, které jsou popsány pod obrázkem. Systém po připojení serveru kontroluje frontu zpráv, která je zasílána ze serverové části. V případě úspěšného načtení a zpracování dat je zpráva odeslána určenému adresátovi přes GSM síť. Serverová část je o úspěšném odeslání vyrozuměna.



Obrázek 14 – Use Case diagram mobilní aplikace

Zdroj: Vlastní zpracování

UC_001_PripojitSeKServeru – Po spuštění mobilní aplikace je pro zahájení přeposílání zpráv ze serveru do GSM sítě nutné ručně připojit aplikaci stisknutím tlačítka.

UC_002_OdpojitSeOdServeru – Pokud je z nějakého důvodu potřeba ihned zastavit spojení aplikace se serverem, je to správci umožněno opět stisknutím tlačítka.

UC_003_ZobrazFrontuZprav – Správce má možnost zobrazit si frontu zpráv čekajících na odeslání. Tato funkce se hodí především pro kontrolu, kolik zpráv čeká na odeslání, pokud je brána vysoce zatížena nebo odpojena od serveru.

UC_004_OdesliSMSdoDatabase – Tato funkcionality umožňuje správci odeslání textové zprávy do databáze serveru. Odtud je odeslána zpět do telefonu a přeposílána GSM sítí. Výhoda v tomto odeslání zprávy spočívá v archivaci odeslané SMS v centrální databázi.

UC_005_NastaveniAplikace – Mobilní aplikace poskytuje správci také jednoduché nastavení. Je možné zvolit interval kontroly dostupnosti nových SMS ve frontě, určit adresy rozhraní a vyplnit uživatelské jméno a API klíč pro komunikaci se serverem.

UC_006_Statistika – Aplikace umožňuje správci zobrazit graf s počtem odeslaných zpráv za určené časové období.

UC_007_InformaceOAplikaci – Poskytnou správci informace o mobilní aplikaci a adresu k serverové části.

UC_008_KontrolujDostupnaData – Systém kontroluje v nastaveném intervalu frontu zpráv čekajících na odeslání. Tyto zprávy jsou nahrány do aplikace a přeposlány přes GSM síť jejich adresátům.

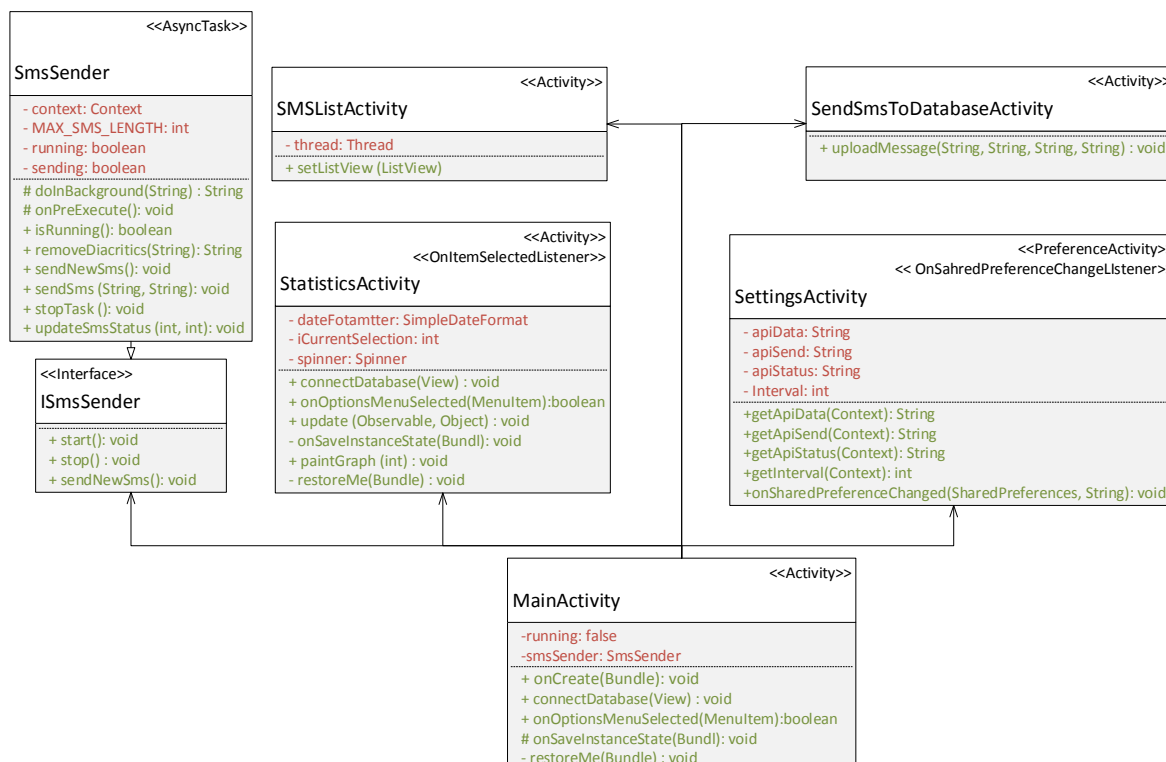
UC_009_OdesliZpravyPresGSM – Zprávy načtené ze serveru systémem odesílá příjemci přes GSM síť.

UC_010_OznamOdeslaniZpravyServeru – Poté co je zpráva úspěšně odeslána, vyšle systémem informaci do serverové části, aby mohla být SMS archivována.

4.9 Návrh tříd mobilní aplikace

Model tříd mobilní aplikace je kvůli lepší přehlednosti rozdělen na menší celky, které popisují stěžejní funkce a vlastnosti celé aplikace. Diagramy se nemusí shodovat s konečnou podobou implementace, protože v konečném návrhu jsou použity i metody, které jsou nutné a typické pro platformu Android.

První třída, se kterou přijde uživatel do styku po spuštění aplikace je `MainActivity`. Nejprve se provede inicializační metoda `onCreate`, která načte grafický layout z XML souboru. Aktivity tvoří prezentační vrstvu aplikace. Načítání grafického layoutu je stejné pro všechny aktivity, proto ho dále nebudu zmiňovat.



Obrázek 15 – Diagram tříd MainActivity

Zdroj: Vlastní zpracování

Na prvním diagramu tříd (Obrázek 15) je zobrazena hlavní třída `MainActivity`, která obsahuje hlavní metodu `connectDatabase`. Tato metoda po prvním stisku tlačítka vytvoří instanci třídy `SmsSender`, čímž se na pozadí zahájí stahování dat ze serverové části a následné rozesílání SMS zpráv. Po opětovném stisku tlačítka je rozesílání zpráv z databáze zastaveno. Hlavní třída také poskytuje hlavní menu, které je zobrazeno po stisku hardwarového tlačítka menu. Výběr položky z menu má na starosti metoda `onOptionsItemSelected`. Metody `onSaveInstanceState` a `restoreMe` jsou typické pro systém Android, a slouží k uložení a obnovení stavu aplikace při překlopení displeje. V tomto případě je to stav tlačítka a tedy i informace o tom, zda se mají rozesílat SMS zprávy.

O příjem a rozesílání textových zpráv se stará třída `SmsSender`. Její rozhraní `ISmsSender` poskytuje metody `start` a `stop` pro spuštění a ukončení načítání a rozesílání textových zpráv. Protože příjem zpráv probíhá přes síť Internet, je nutné provádět tuto komunikaci ve vláknech. To z toho důvodu, že Android nepovolí síťovou komunikaci na hlavním User Interface vláknech. K tomuto účelu je určena generická abstraktní třída `AsyncTask`[24][25], která umožňuje vykonávat asynchronní operace. Užitečnost této třídy spočívá v tom, že není potřeba ručně vytvářet vlákna a starat se o ně. Inicializace před prováděním úkonů probíhá pomocí metody `onPreExecute`. Hlavní úkony se provádějí v metodě `doInBackground`, která v určeném intervalu načítá SMS a rozesílá je do GSM sítě pomocí metody `sendNewSms`. Před odesláním SMS je text odstraněna diakritika metodou `removeDiacritics`. Po odeslání zprávy, je třeba do hlavní databáze odeslat informaci o úspěšném odeslání zprávy metodou `updateSmsStatus`.

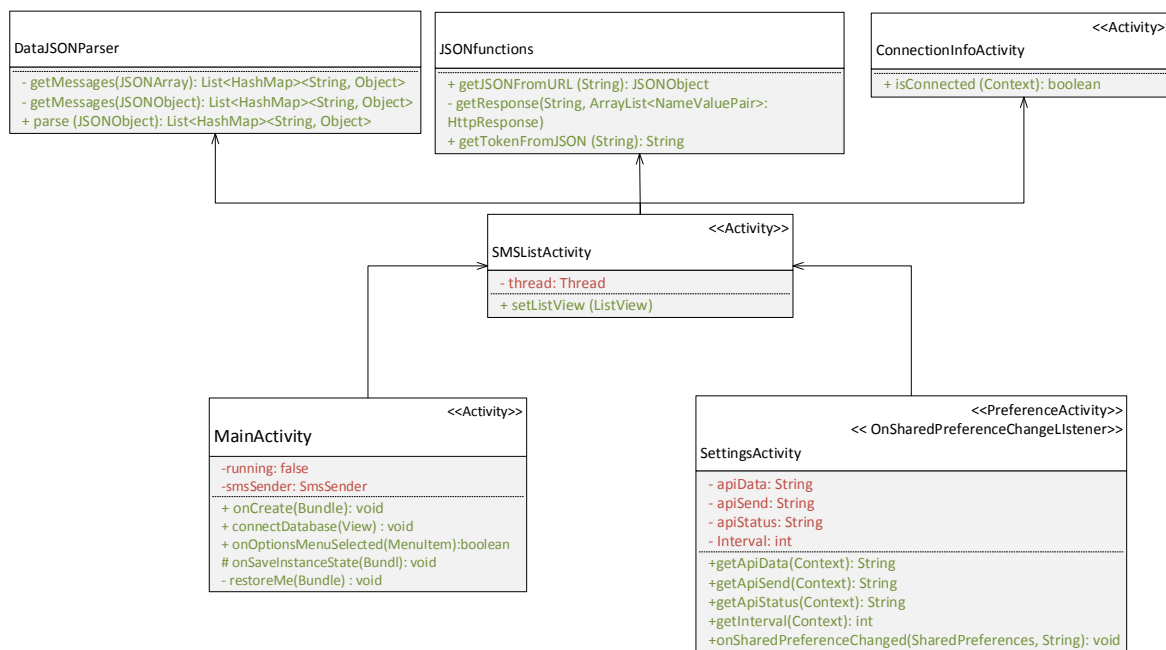
Další funkcionalitou, kterou nabízí aplikace je zobrazení fronty zpráv, čekajících na odeslání. Toto zajišťuje třída `SMSListActivity`. Pro zobrazení zpráv je použita třída `SimpleAdapter`. Protože se opět jedná o asynchronní komunikaci se serverem, zprávy musí být načítány ve vláknech.

K otestování API pro odesílání zpráv je vytvořena třída `SendSmsToDatabaseActivity`. Třída poskytuje metodu `uploadMessage`, jejímž úkolem je odeslání textové zprávy na server.

Třída `SettingsActivity` slouží k základnímu nastavení aplikace. Sdílení nastavení pro celou aplikaci zajišťuje rodičovská třída `PreferenceActivity`. Třída `SettingsActivity` nastavuje a poskytuje API adresy pro data, odesílání SMS, aktualizaci stavu zprávy a nastavení intervalu odesílání zpráv. Metoda `onSharedPreferenceChanged` slouží pro validaci vkládaných hodnot nastavení z uživatelského rozhraní. [26]

Poslední Aktivita, která je vyvolávána přímo z hlavního menu je `StatisticsActivity`. Ta poskytuje uživateli přehled o počtu odeslaných SMS. Četnost odeslaných zpráv je vykreslena do grafu metodou `paintGraph`. Samotné vykreslení obstarává externí knihovna. Statistika jsou vykreslovány pro různý časový úsek, o překreslení při výběru intervalu se stará metoda `onOptionsMenuSelected`.

Následující diagram tříd (Obrázek 16) popisuje podrobněji kooperaci třídy SMSListActivity, která načítá textové zprávy čekající na odeslání.



Obrázek 16 – Diagram tříd SMSListActivity

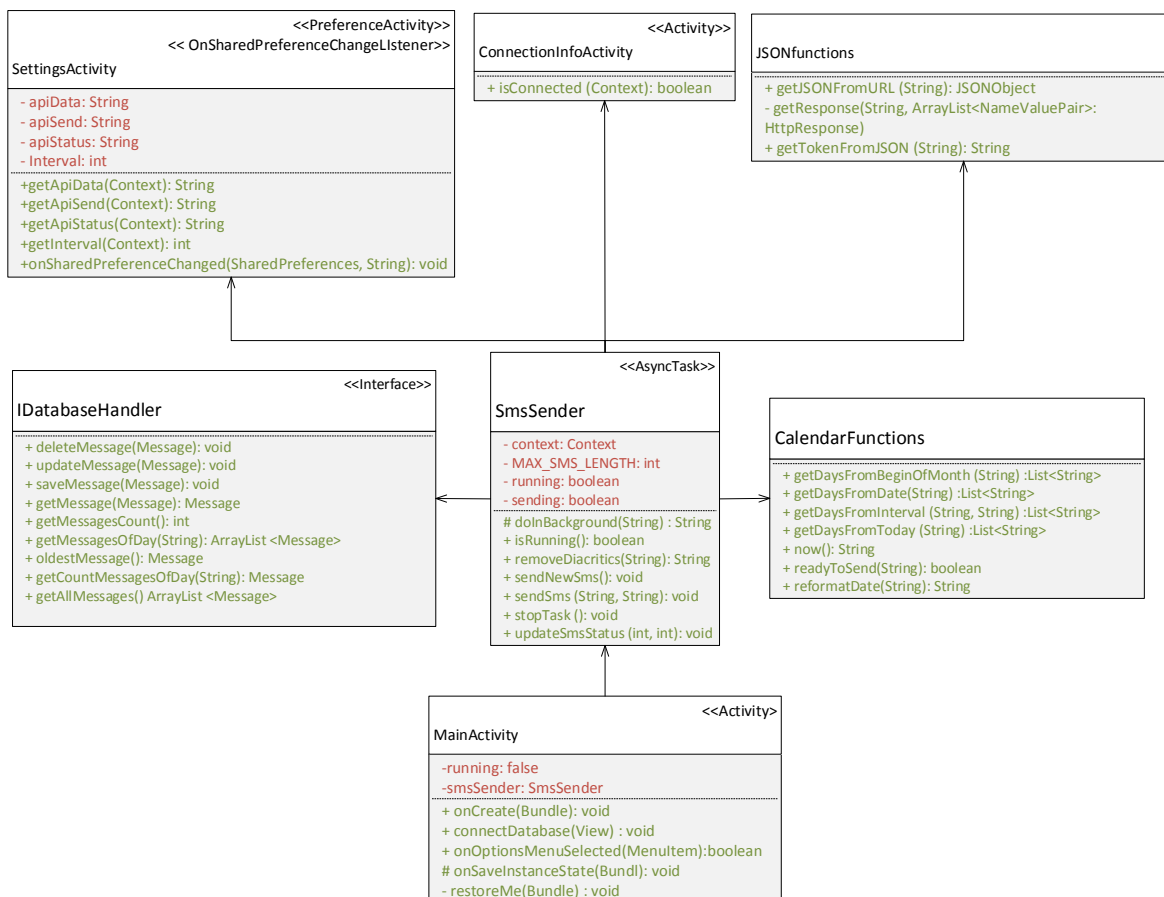
Zdroj: Vlastní zpracování

Nejprve se provede kontrola, zda je telefon připojený k síti Internet. Tato kontrola probíhá pomocí statické metody `isConnected` ze třídy `ConnectionInfoActivity`. Zprávy jsou staženy z adresy API poskytující data. Adresu poskytuje třída `SettingsActivity`.

Stažená data jsou ve formátu JSON. Jsou proto uložena jako JSON objekt díky funkci `getJSONfromURL` ze třídy `JSONfunctions`. Protože má tato třída na starosti načítání HTTP Response, je potřeba ošetřit také situaci, kdy vyprší platnost přiděleného API klíče (tokenu). Zažádání o token pro síťovou komunikaci a následné uložení obstarává metoda `getTokenFromJSON`. Rozhraní `JSONParser` poskytuje metodu `parse`. Rozhraní náleží třídě `DataJSONParser`, která z načteného pole ve tvaru JSON vytvoří List zpráv.

Srdcem celé aplikace je třída `SmsSender`. Jak je popsáno výše, třída zajišťuje stahování a odesílání SMS zpráv na pozadí pomocí metody `doInBackground`. Při rozesílání zpráv je nutné kontrolovat, zda se jedná o zprávu určenou k odeslání v určený čas. Pokud tomu tak je, zavolá se metoda `readyToSend` ze třídy `CalendarFunctions`. Metoda má za úkol zkontrolovat, zda už nastal čas odeslání zprávy. Tato metoda vrací pravdivé tvrzení, pokud čas odeslání zprávy je větší nebo roven aktuálnímu času.

Odeslané zprávy jsou ukládány do interní SQLite databáze. K tomuto slouží rozhraní `IDatabaseHandler`, které poskytuje všechny potřebné metody pro práci s daty ve formě zpráv. Do databáze jsou ukládány informace o odeslané zprávě avšak bez textu zprávy. V případě potřeby je možné text zprávy dohledat v serverové části podle id odeslané zprávy. Kromě funkce logů, slouží uložené zprávy také pro statistické účely třídy `StatisticsActivity`.

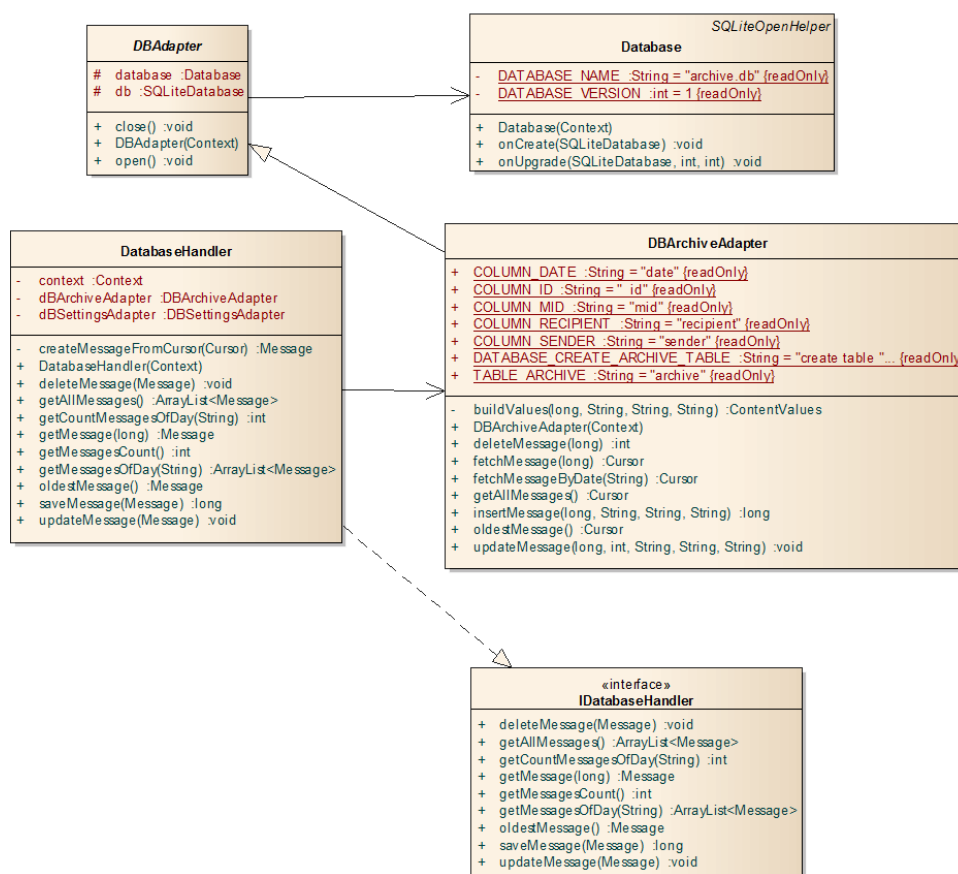


Obrázek 17 – Diagram tříd SmsSender

Zdroj: Vlastní zpracování

Kromě výše vypsaných tříd obsahuje mobilní aplikace také třídy definující datovou strukturu zpráv – `Message`. Pro uložení a poskytnutí API klíče slouží třída `Token`.

Poslední diagram tříd (Obrázek 18) popisuje způsob, jak mobilní aplikace pracuje s databází.



Obrázek 18 – Diagram tříd databáze

Zdroj: Vlastní zpracování

Třída Database dědí od SQLiteOpenHelper a slouží k nastavení názvu databáze a tabulek. Tabulky jsou vytvořeny metodou onCreate. Metoda onUpgrade je vyvolána, pokud se změní verze databáze. To nastává například při změně struktury tabulek.

Abstraktní třída DBAdapter je bázovou pro všechny třídy tabulkových adaptérů. Třída umožňuje načítání databáze, otevření databáze a také její uzavření. Třída DBArchiveAdapter obsahuje atributy a název tabulky archive, také slouží pro vykonávání SQL dotazů nad tabulkou.

DatabaseHandler má na starosti manipulaci se všemi tabulkami databáze. Typicky otevírá databázové spojení, provede SQL dotazy Adapteru tabulky a poté databázové spojení ukončí. Umožňuje také procházení vrácených dat pomocí cursoru. Nad třídou DatabaseHandler se nachází rozhraní IDatabaseHandler, pomocí kterého je databáze připojena k aplikaci.

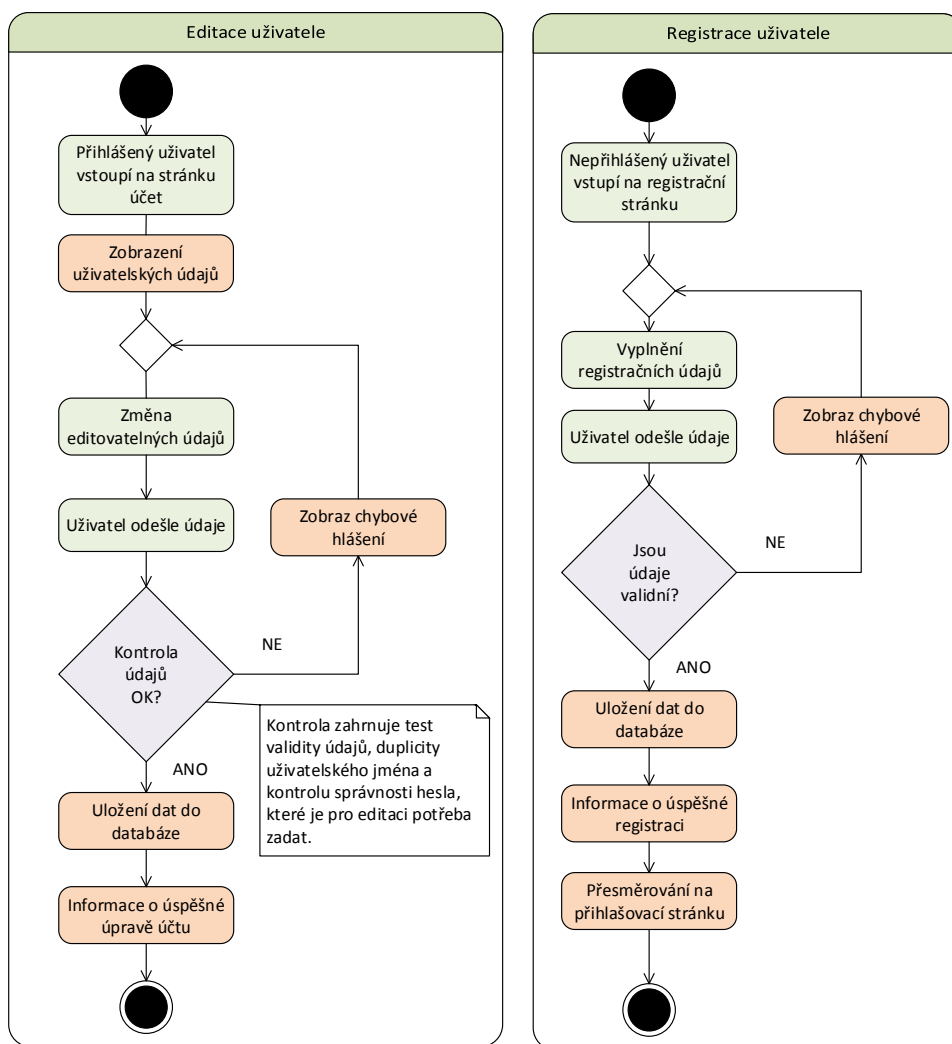
4.10 Návrh modelu webové aplikace

Protože webová aplikace není tvořena objektově orientovaným programováním, nebylo vhodné tvořit analytický model tříd. Místo toho se návrh skládá z klíčových Activity diagramů.

Registrace uživatele, editace profilu uživatele

Obrázek 19 znázorňuje registraci uživatele a editaci uživatelského účtu. Diagram aktivit vlevo znázorňuje registraci nového uživatele. Registrace je umožněna pouze nepřihlášeným uživatelům. Musí být vyplněny všechny povinné údaje a zadané hodnoty musí být validní. Není možné zadat neplatný email, telefon ve špatném formátu, krátké heslo a podobně. Po vytvoření účtu je uživatel přesměrován na přihlašovací stránku.

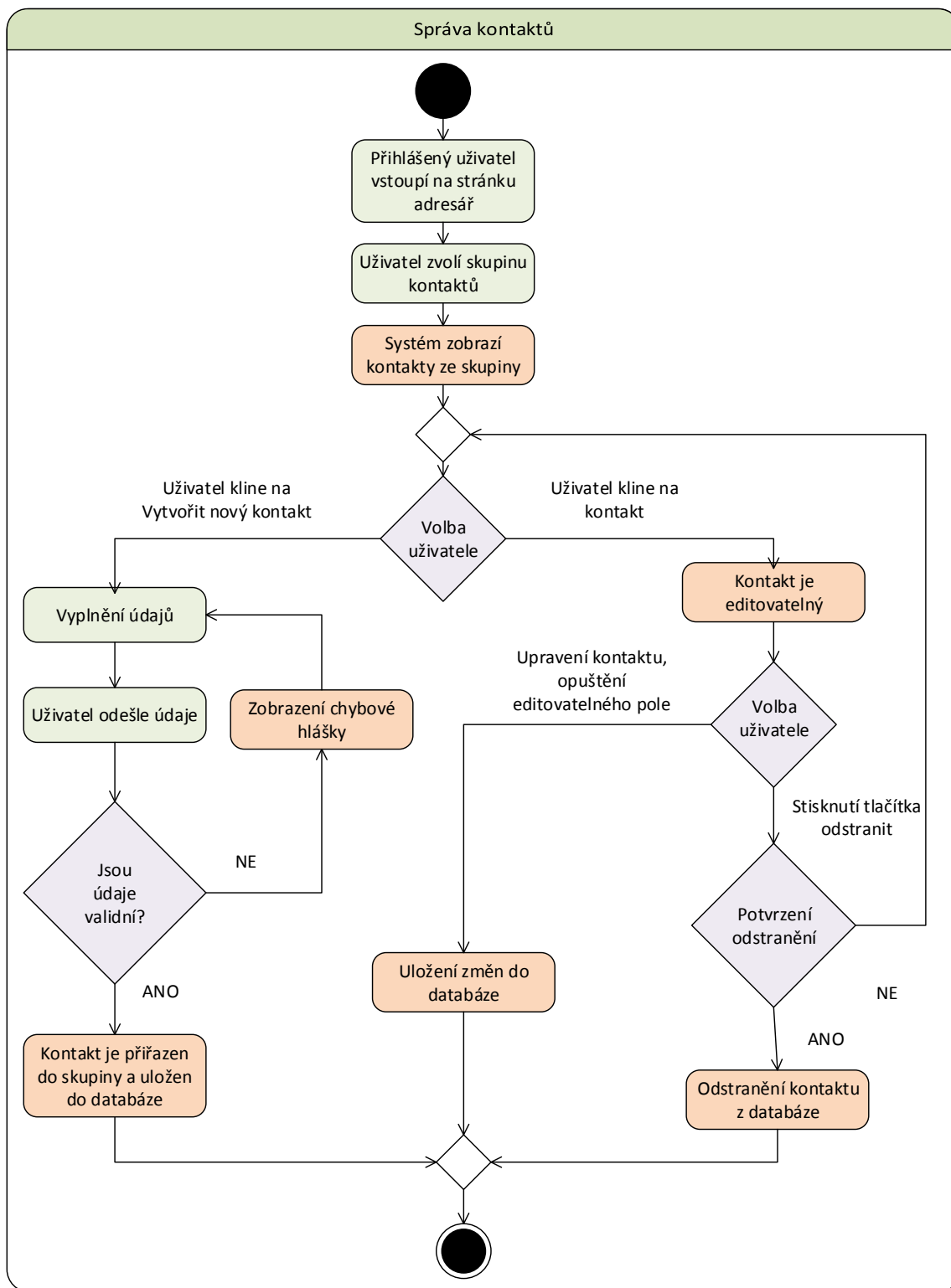
Po vytvoření účtu může uživatel zobrazit a měnit svoje osobní údaje. Průběh změny údajů je znázorněn na obrázku vpravo. Editace účtu je uživateli umožněna pouze při zadání současného hesla uživatele. Stejně jako při registraci zadávané údaje musí projít validací.



Obrázek 19 – Activity diagram registrace a editace uživatele
Zdroj: Vlastní zpracování

Správa kontaktů

Přihlášený uživatel má k dispozici adresář kontaktů, který může spravovat. Jak vyplývá ze schématu (Obrázek 20), pro správu kontaktů je nutné nejprve vybrat adresář, ve kterém tato správa bude probíhat. Kontakty lze snadno editovat, odebírat a také přidávat.

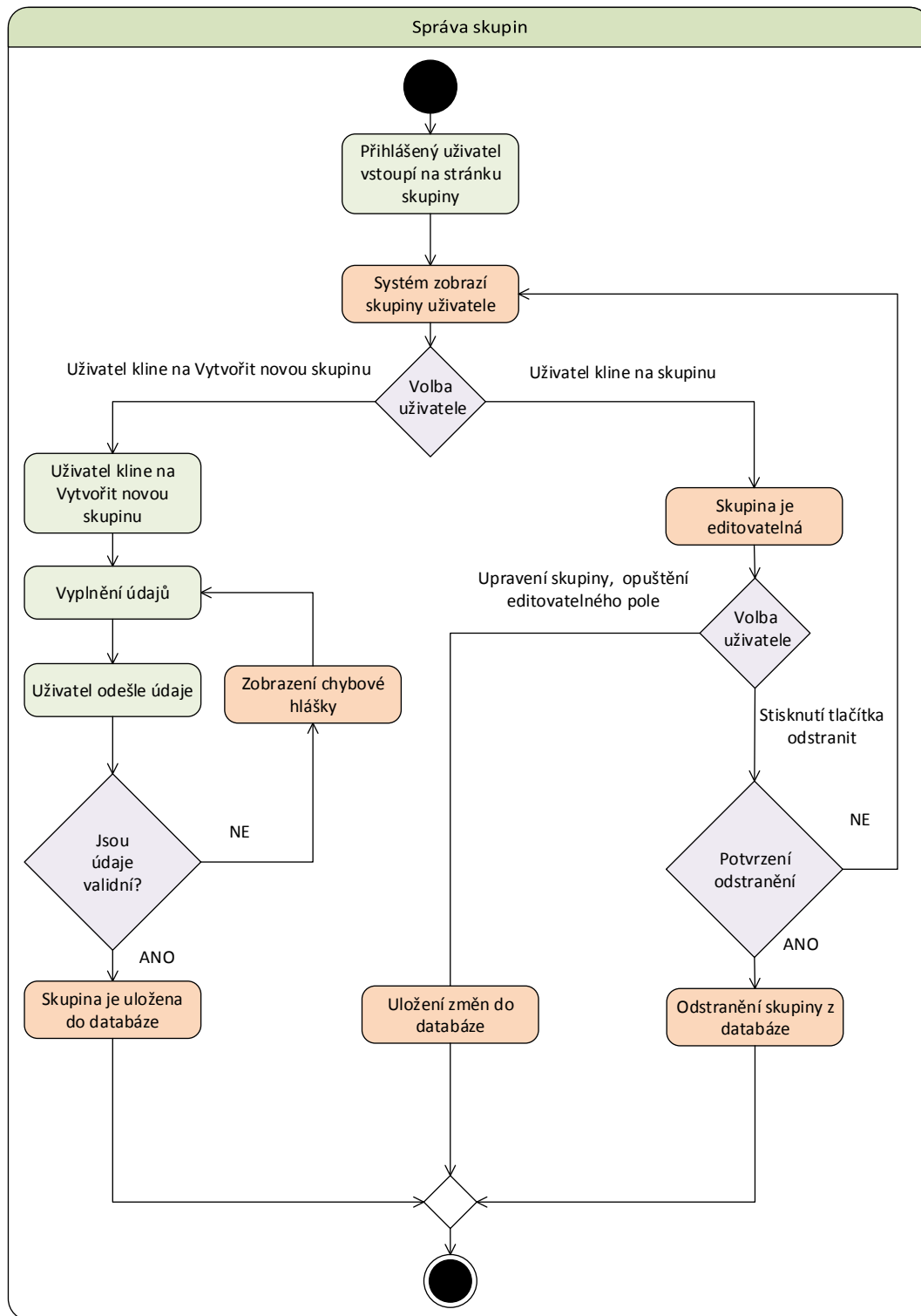


Obrázek 20 – Activity diagram správa kontaktů

Zdroj: Vlastní zpracování

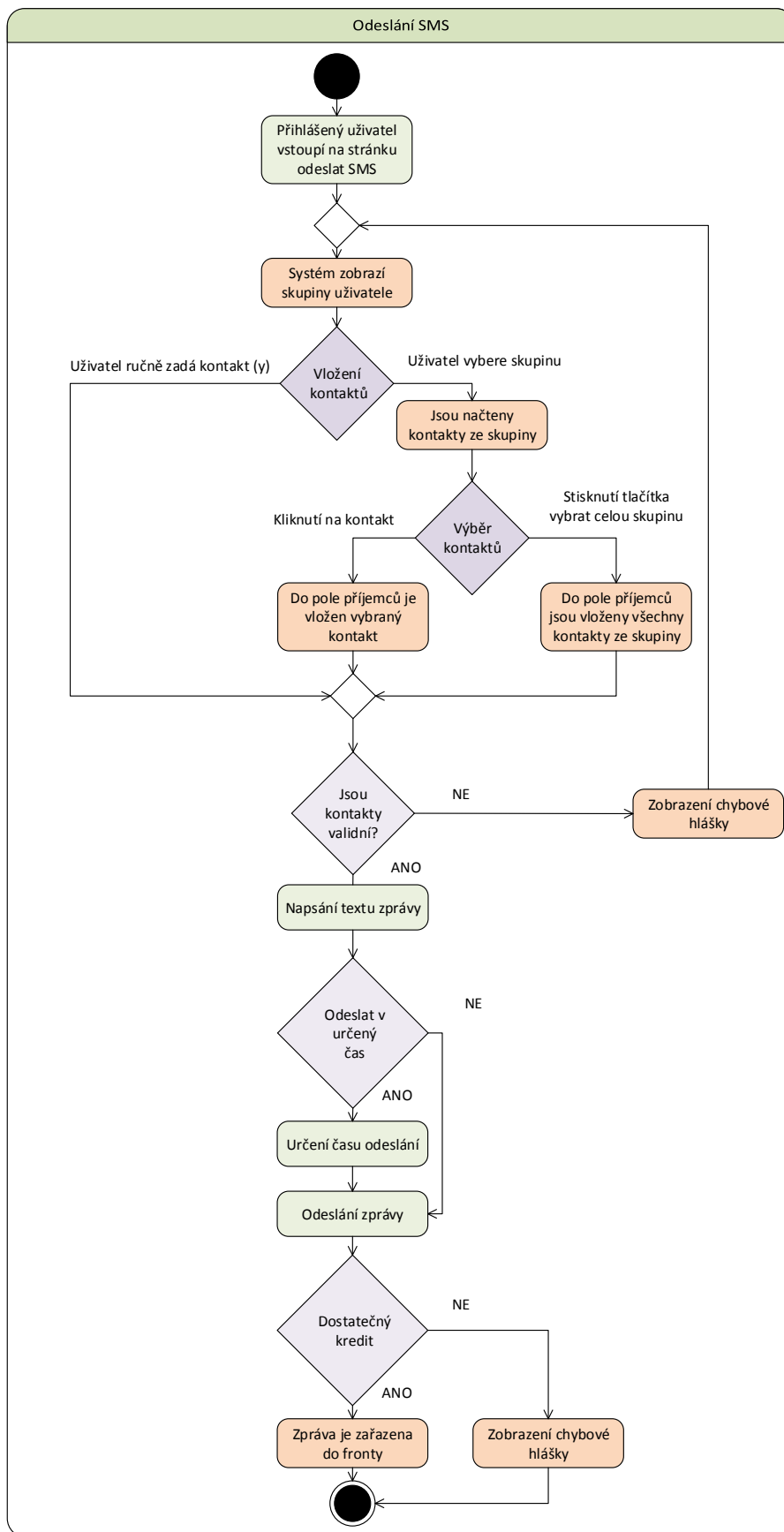
Správa skupin

Se správou kontaktů úzce souvisí také správa skupin (Obrázek 21), do kterých jsou kontakty přiřazeny. Stejně jako je tomu u kontaktů lze skupiny editovat, přidávat a odebírat. Při vytváření skupiny je nutné zadat název, volitelný je popis skupiny a podpis uživatele pro skupinu.



Obrázek 21 – Activity diagram správa skupin
Zdroj: Vlastní zpracování

Odeslání SMS



Obrázek 22 – Activity diagram odeslání SMS

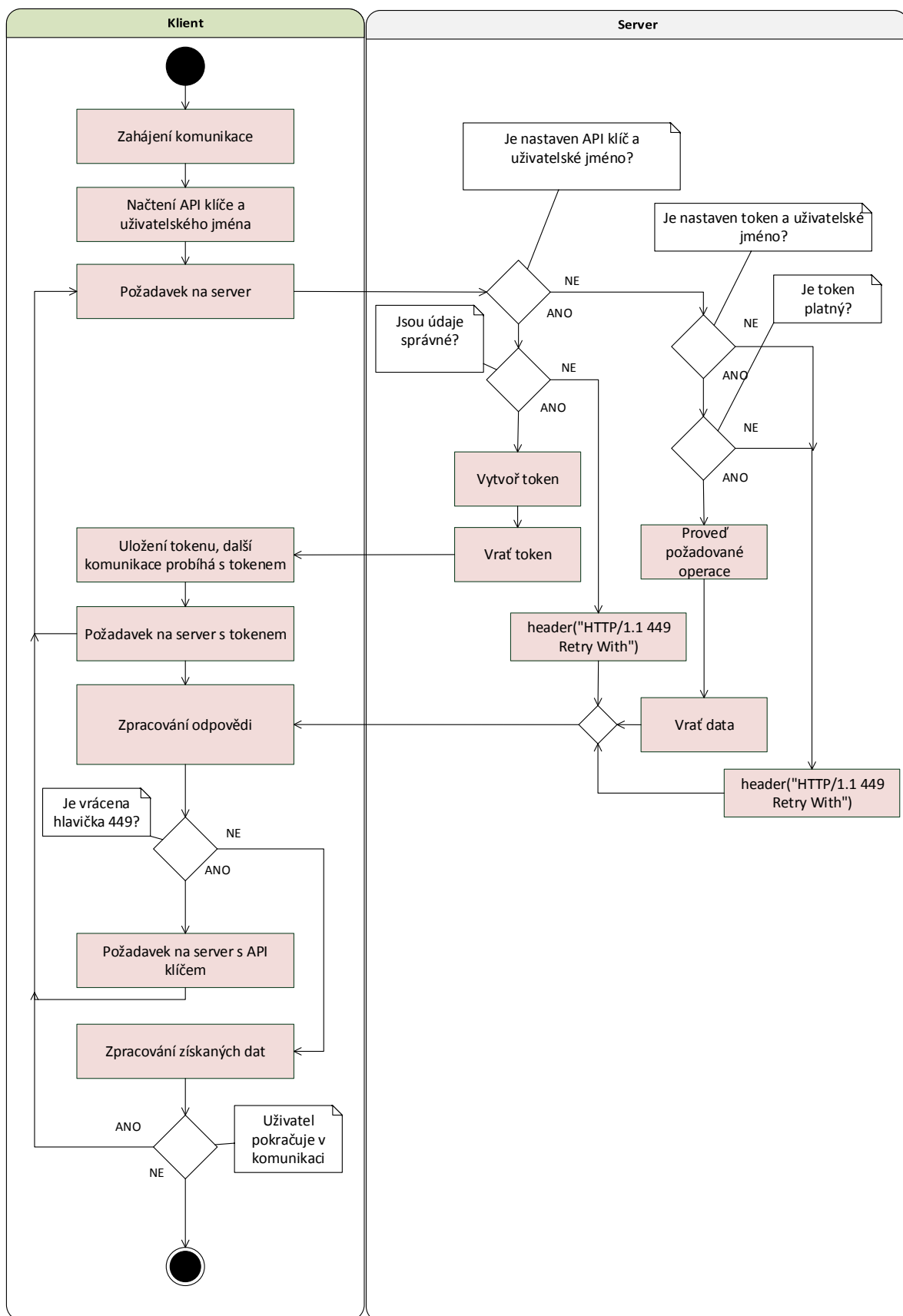
Zdroj: Vlastní zpracování

Diagram aktivit z předchozí strany (Obrázek 22) popisuje hlavní funkcionalitu webového rozhraní, samozřejmě se jedná o odeslání textové zprávy. Telefonní číslo lze zadat klikem na kontakt z adresáře, přidáním celé skupiny nebo ručním zadáním čísla. Telefonní číslo je validováno, aby se předešlo odeslání SMS zprávy na neplatné číslo. Délka SMS zprávy je průběžně kontrolována a není možné odeslat zprávu delší než 1000 znaků.

Volitelnou vlastností je naplánování odeslání textové zprávy v určený čas. Pokud tak uživatel učiní, je formát zadaného času a data validován a v případě nevyhovujícího formátu je o tom uživatel upozorněn. Vzhledem k rozsáhlosti diagramu tato validace není graficky znázorněna. Po stisku tlačítka pro odeslání zprávy proběhne nejprve kontrola, zda má přihlášený uživatel dostatečný kredit pro odeslání zprávy. Pokud je zbývající kredit dostatečný, zpráva je odeslána do fronty zpráv čekajících na zpracování.

API rozhraní

Důležitou součástí serverové části práce je API rozhraní poskytující základní funkcionalitu SMS brány. Rozhraní umožňuje odesílání SMS zpráv na server, ale také přijímání archivu odeslaných zpráv a SMS čekajících na odeslání. Protože k rozhraní může přistupovat kdokoliv, důležitou složkou je zabezpečení komunikace. To probíhá ve dvou fázích. Při zahájení komunikace se uživatel prokáže soukromým API klíčem. Obratem obdrží token s platností 5 minut, kterým se prokazuje při dalších požadavcích. Po uplynutí doby platnosti tokenu je serverem vyzván k opětovnému zaslání soukromého API klíče. Aktivitní diagram popisující autentizaci uživatele je uveden na následující straně (Obrázek 23).



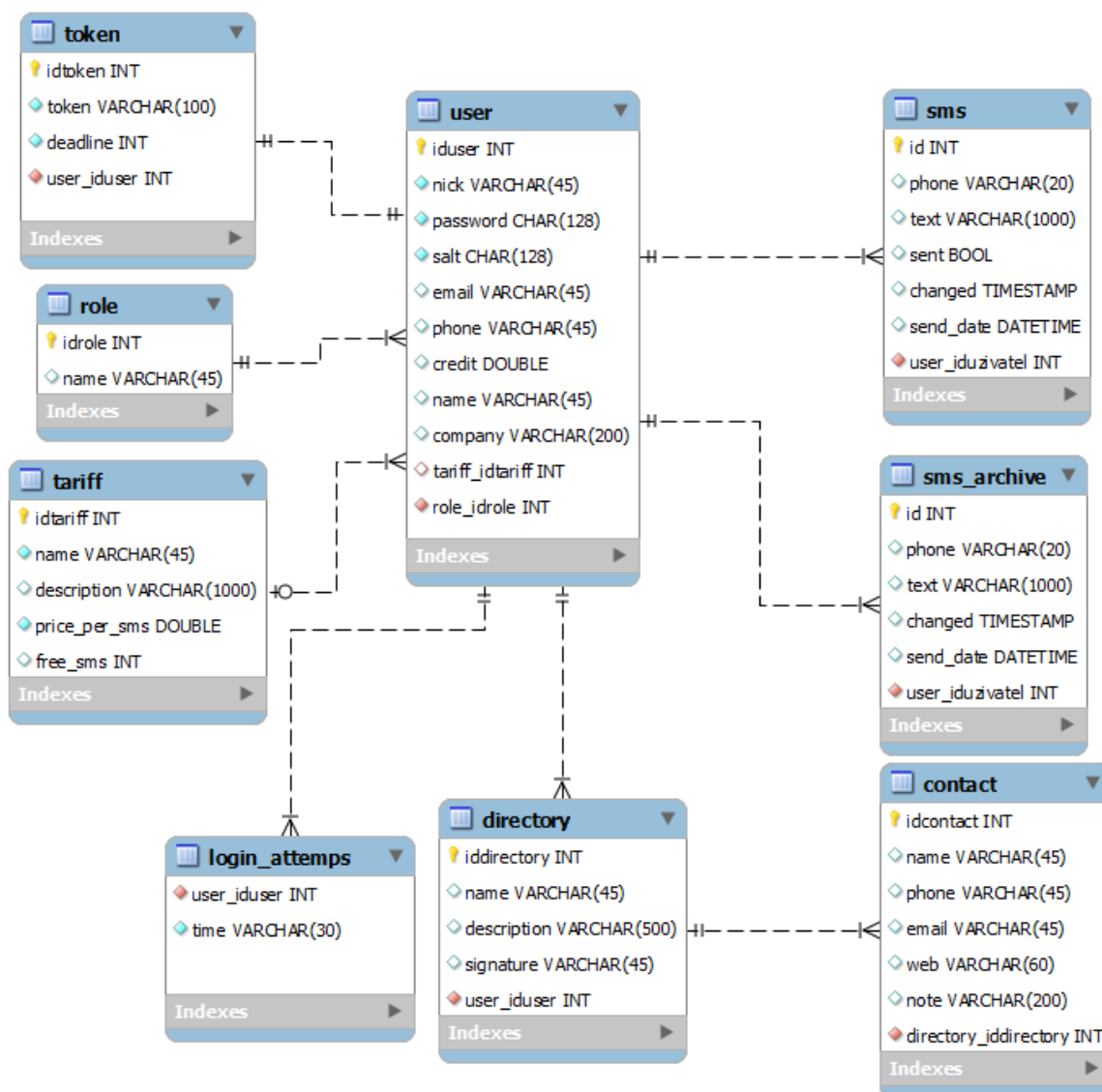
Obrázek 23 – Activity diagram API

Zdroj: Vlastní zpracování

4.11 Návrh databázového modelu

Diagram tříd nastíní, jaká data je potřeba uchovávat. Tyto informace slouží jako základ pro databázový model. V praxi jsem v průběhu implementace řešení databázový model mírně upravoval, protože jsem zjišťoval nové implementační detaily, se kterým nebylo při návrhu počítáno. Databázový model byl vytvořen v programu MySQL Workbench od společnosti Oracle, který umožňuje vygenerovat SQL Script pro vytvoření tabulek a vazeb dle modelu.

4.11.1 EER model serverové části



Obrázek 24 – EER model
Zdroj: Vlastní zpracování

Obrázek 24 představuje model databáze serverové části. Obsahuje 9 tabulek s vazbami mezi tabulkami. Význam každé tabulky je vysvětlen níže.

4.12 Význam tabulek

Tabulka user – Tabulka user obsahuje údaje o registrovaném uživateli webové aplikace. Ve sloupci salt je uložen náhodný 128 znaků dlouhý řetězec, který byl vygenerován při registraci uživatele. Při ověřování správnosti hesla je řetězec salt smíchán s heslem a tak je zaručený jedinečný hash hesla uživatelů i v případě zadání stejných hesel. Dalším zajímavým sloupcem je credit, který uchovává hodnotu kreditu uživatele pro odesílání zpráv. Odesílané zprávy jsou účtovány dle tarifu, jehož cizí klíč tariff_idtariff je obsazen v tabulce. Dalším cizím klíčem je role_idrole, který přiřazuje uživateli roli a slouží tak k autorizaci.

Tabulka sms – Tato tabulka slouží jako fronta všech zpráv čekajících na odeslání. Sloupec changed uchovává časové razítko vložení SMS zprávy do tabulky, send_date určuje, v jaký čas má být zpráva odeslána. Pokud není čas odeslání vyplněn, je zpráva odeslána ihned. Po potvrzení odeslání je zpráva přesunuta do archivu.

Tabulka sms_archive – Zde jsou archivovány zprávy, které byly odeslány přes mobilní telefon. Na rozdíl od tabulky sms zde má sloupec changed význam časového razítka odeslání zprávy.

Tabulka directory – Uživatel si může vytvářet svoje adresáře, které může mít pro lepší přehlednost popsané (sloupec description). Pro každou skupinu si uživatel může určit automatický podpis, který je možné automaticky připojit na konec zprávy (sloupec signature)

Tabulka contact – Obsahuje kontakty, které jsou zařazeny do jednotlivých adresářů.

Tabulka role – Uchovává název role a identifikátor, sloužící pro autorizaci uživatele.

Tabulka tariff – Aplikace počítá s možností zpoplatnění SMS zpráv. Dle zvoleného tarifu je naúčtována cena za odeslanou SMS (price_per_sms). Některé tarify poskytují také určitý počet volných SMS (free_sms).

Tabulka login_attempts – Slouží pro registraci chybných pokusů o zalogování do aplikace. Slouží jako ochrana proti útokům na heslo hrubou silou.

Tabulka token – Při prvotní autentizaci přes API je uživateli přidělen token, což je alfanumerický náhodný řetězec. Token má platnost pět minut. Doba platnosti je uložena ve sloupci deadline, který uchovává počet vteřin od 1.1.1970 + 300. Po dobu platnosti je token používán pro autentizaci při ověření identity přes API.

4.13 SQLite databáze

V této databázi mobilního telefonu je vytvořena pouze jedna tabulka s názvem archive, která slouží pro ukládání určitých parametrů odesílaných zpráv. Data této tabulky slouží jako potvrzení o odeslání SMS zpráv a také pro vytváření grafu vytíženosti mobilní aplikace.

Tabulka archive obsahuje tyto sloupce:

- `_id` – Id zprávy v SQLite databázi.
- `Mid` – Id zprávy v MySQL databázi – pro spárování s detaily zprávy v serverové části.
- `Sender` – Odesílatel zprávy.
- `Recipient` – Příjemce zprávy.
- `Date` – Datum a čas odeslání zprávy.

5 Implementace řešení

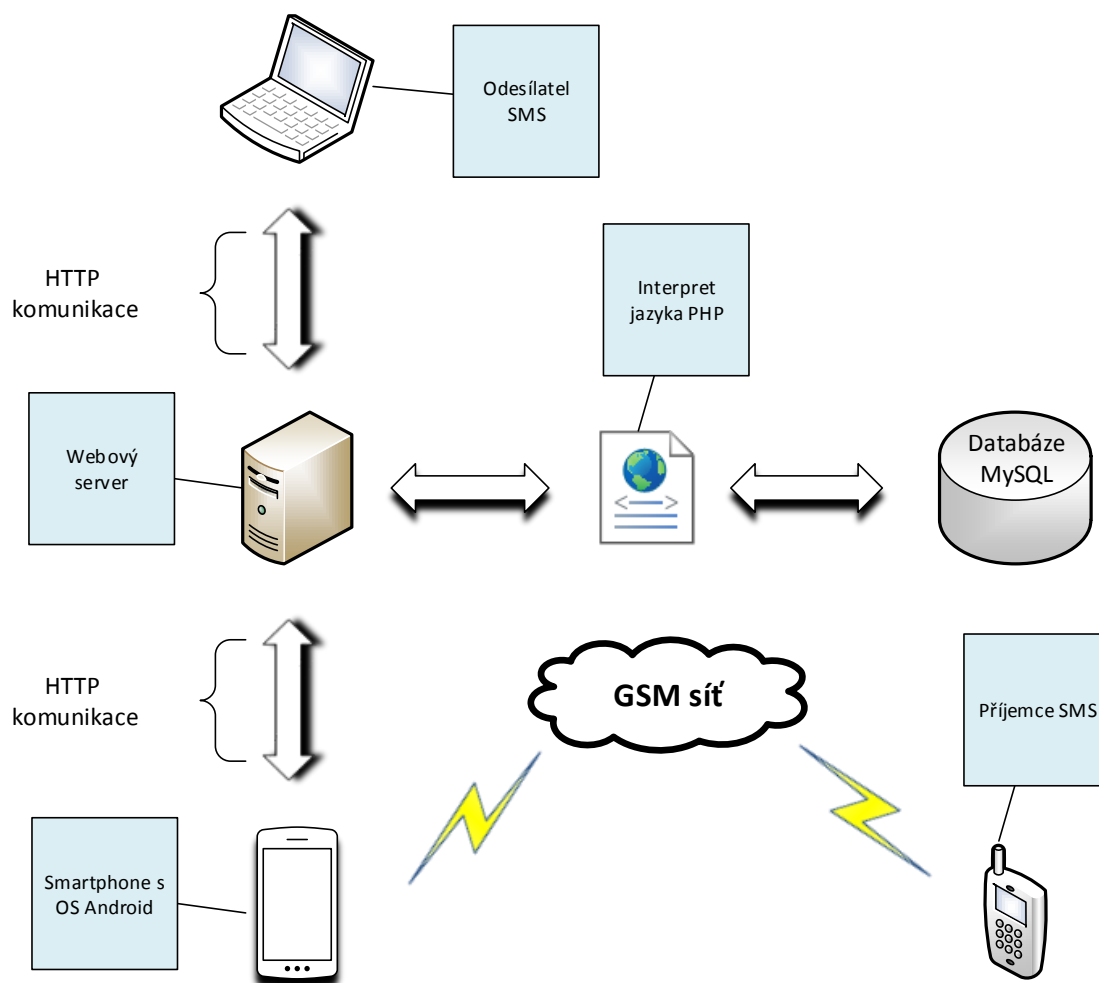
V této kapitole jsou popsány použité technologie a jsou zde rozepsány konkrétní implementační aspekty. Kapitola také poskytuje náhled do uživatelského rozhraní systému. Na závěr vyhodnotím výsledek své práce.

V rámci této práce byly vytvořeny dvě spolupracující aplikace. Mobilní aplikace je určena pro platformu Android. Tato aplikace byla vyvíjena jako první, protože poskytuje základní funkcionalitu celé práce.

Druhá aplikace je pro webový server a má za úkol poskytnout veřejně dostupné API a možnost odeslat SMS zprávu příjemci z adresáře přes webové rozhraní. Ostatní popisované funkcionality jsou nad rámec zadání.

5.1 Grafické schéma systému

Na obrázku níže (Obrázek 25) je znázorněno schéma systému pro odesílání textových zpráv z SMS brány přes mobilní telefon. Schéma je poměrně přehledné a tak k němu dodám pouze informaci, že SMS zprávy nemusí být odesílány pouze prostřednictvím webového prohlížeče, ale také pomocí implementace API do vlastního systému.



Obrázek 25 – Grafické schéma systému

Zdroj: Vlastní zpracování

5.2 Použité technologie

Mobilní aplikace je naprogramována pro platformu Android ve vývojovém prostředí Eclipse (verze 4.2.2) s nainstalovaným Android Software Development Kit (SDK). Mobilní aplikace disponuje databází SQLite, která uchovává data pro vytvoření statistik vytíženosti SMS brány. Aplikace je podporována od API verze 9 (Android 2.3.2).

Serverová část je napsána ve skriptovacím jazyku PHP (verze 5.4.17). Pro vývoj bylo využito IDE Netbeans 6.9.1. Pro uchování dat serverové části je využit databázový systém MySQL. Komunikace mezi mobilní aplikací a serverovou částí probíhá odesíláním a přijímáním dat ve formátu JSON.

Android

Android je **open source**¹² operační systém pro mobilní zařízení vytvořený společností **Google**. Společnost Android Inc. byla založena roku 2003 ve městě Palo Alto v Kalifornii. V roce 2005 došlo k odkupu společností Google Inc. Beta verze systému Android byla představena na konci roku 2007. Operační systém je založen na **Linuxovém jádře 2.6**, které zajišťuje zabezpečení, správu paměti a procesů, přístup k síti a ovladačům a další nezbytné základní funkcionality. K funkcím jádra jednotlivé aplikace nepřistupují přímo, ale prostřednictvím **Android API**. Verze API je pro vývojáře velice důležitá, poskytuje velký počet služeb, které mohou být snadno použity v aplikacích. Při vývoji aplikace pro Android je třeba určit minimální úroveň API, kterou aplikace podporuje. Nejnovější verze rozhraní je nyní (srpen 2013) s pořadovým číslem 18 pro operační systém Android 4.3 Jelly Bean Milestone Release 2. [27]

5.2.1 SQLite

SQLite je populární databáze zabudovaná v systému Android. Jak název napovídá, obsahuje čisté SQL rozhraní. Databáze je paměťově nenáročná a přitom dostatečně rychlá. SQLite databázi může vytvářet jednoduše každá aplikace pro platformu Android.

Velký rozdíl mezi ostatními databázemi založenými na SQL spočívá v typování dat. Specifikace datových typů sloupců tabulky bere SQLite pouze jako doporučení. Do jakéhokoli sloupce lze tedy uložit data libovolného typu. Také tato databáze nepodporuje několik základních vlastností SQL, jako je spojení typu RIGHT OUTER JOIN, FULL OUTER JOIN, omezení typu FOREIGN_KEY a některé verze příkazu jako ALTER TABLE. Až na tyto výjimky je však SQLite propracovaná databáze s plnou funkcí včetně transakcí, triggerů apod. [28]

5.2.2 Eclipse

Eclipse je vývojové prostředí určené především pro programování v jazyce Java. Velikou výhodou tohoto prostředí je snadná rozšiřitelnost o podporu dalších programovacích jazyků, včetně nástrojů pro tvorbu grafického uživatelského rozhraní. Pro získání vývojového prostředí je třeba provést instalaci pluginu (doplňku) Android SDK. [29]

Po instalaci pluginu máme k dispozici emulátor, který umožňuje otestovat a ladit vyvíjenou aplikaci bez nutnosti připojení fyzického zařízení s operačním systémem Android. Velkou výhodou emulátoru je možnost vytvářet různá virtuální zařízení s rozdílnou velikostí paměti, displeje a především s různou verzí operačního systému. Programátor si tak snadno může vyzkoušet, jak se bude aplikace chovat na mnoha zařízeních, aniž by je musel vlastnit.

Pro rychlejší práci při ladění a testování aplikace však doporučuji připojení fyzického zařízení přes datový kabel. Aplikace tak není zpomalena emulací a navíc lze reálně vyzkoušet interakci s hardwarem součástí přístroje. Například simulace GPS zařízení nelze reálně simulovat.

¹² Počítačový software s otevřeným zdrojovým kódem.

5.2.3 Android SDK

Android SDK poskytuje knihovny Javy a vývojářské nástroje nezbytné pro vývoj, testování a ladění aplikací pro platformu Android. Obsahuje ukázky zdrojových kódů a základní konstrukce pro rychlé vytváření aplikací. Součástí sady je také emulátor, který umožňuje vytvářet virtuální zařízení se zvolenými parametry a verzí operačního systému. [30]

5.2.4 PHP

PHP neboli hypertextový preprocesor je open source skriptovací programovací jazyk určený především pro programování dynamických internetových stránek a webových aplikací. Skripty jsou prováděny na straně serveru a k uživateli je přenášén pouze výsledek. PHP je nezávislý na platformě a tak lze skripty přenášet bez úprav mezi různými operačními systémy. Jednoduchost použití a bohatá zásoba funkcí zajistila, že PHP je nejrozšířenější skriptovací jazyk pro tvorbu webu. Vyvíjené aplikace je možno psát procedurálně, ale také pomocí objektově orientovaného programování. [31]

5.2.5 JavaScript

JavaScript, často označován zkratkou JS, je multiplatformní objektový skriptovací jazyk od Brendana Eichera z tehdejší společnosti Netscape. Zpravidla se používá jako interpretovaný programovací jazyk pro webové stránky. Jedná se o klientský skript, to znamená, že skript je vykonáván prostřednictvím prohlížeče na straně klienta. Velkou výhodou je možnost zápisu přímo do HTML kódu. Ve webovém rozhraní je JavaScript využit například k implementaci technologie AJAX¹³. AJAX umožňuje měnit obsah stránek bez nutnosti jejich znovunačítání. To je využito například pro vypsání obsahu zvoleného adresáře bez nutnosti obnovit okno prohlížeče. Pro snadnou implementaci složitějších JavaScript konstrukcí byla využita knihovna jQuery. [32]

5.2.6 HTML

HyperText Markup Language je hypertextový značkovací jazyk pro definici struktury, obsahu a vzhledu stránek. Jazyk pro tvorbu struktury používá takzvané HTML značky neboli tagy. Tagy mohou být párové: `<tag></tag>` nebo nepárové `<tag/>`. Značky se zapisují do špičatých závorek.

5.2.7 CSS

Cascading Style Sheets, česky kaskádové styly se starají o grafickou úpravu webových stránek. Kaskádové se nazývají proto, že je možné na sebe vrstvit definice stylů. Kaskádové styly umožňují oddělit strukturu a obsah stránek od jejich vzhledu. CSS napomáhá udržovat jednotný styl stránek a provádět grafické změny pro celý HTML dokument.

¹³Asynchronous JavaScript and XML

5.2.8 NetBeans

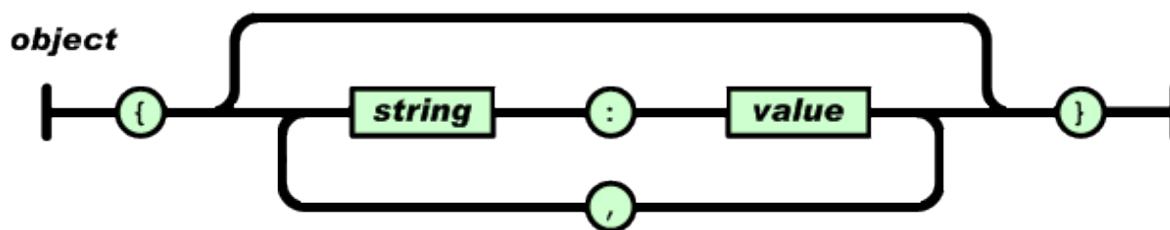
NetBeans je vývojové prostředí s podporou nejen jazyka Java, ale také mnoha dalších programovacích jazyků a frameworků. Vývojové prostředí bylo vytvořeno jako open source v roce 2000 společností Sun Microsystems. Od roku 2010 je Sun Microsystems dceřinou společností firmy Oracle. Vývojové prostředí NetBeans jsem použil při vývoji serverové části aplikace pro jazyky PHP a HTML. Kromě přehledného formátování a kontroly syntaxe toto IDE, umožňuje také automatické odesílání souborů přes FTP/SFTP¹⁴ na vzdálený webový server, čehož jsem využil při testování a ladění PHP skriptů. [33]

5.2.9 MySQL

MySQL je relační databázový systém typu DBMS¹⁵, který vlastní společnost Oracle. MySQL je open source řešení a vychází z deklarativního programovacího jazyka SQL. Databáze obsahuje tabulky, každá tabulka má definované sloupce s určeným datovým typem, a také řádky, které tvoří data. Práce s touto databází je možná v jazycích PHP, Python, Perl, Java, .NET, C++, C, Visual Basic atd. Pro vzdálenou správu MySQL přes webové rozhraní je vhodný phpMyAdmin nebo Adminer.[34]

5.2.10 JSON

JSON¹⁶ je datový formát nezávislý na počítačové platformě. JSON je určený pro přenos dat v polích nebo objektech. Tento formát může nést data typu string, number, object (Obrázek 26), array, true, false nebo null. Výhodou je snadný zápis a čtení formátu pro člověka i libovolný programovací nebo skriptovací jazyk. JSON je považován za odlehčenou a jednodušší alternativu k XML, které obsahuje průměrně 40% značek a atributů. Nevýhodou tohoto formátu je to, že neumožňuje definovat vlastní znakovou sadu. Data jsou přenášena ve znakové sadě UTF-8. [35]



Obrázek 26 – JSON objekt

Zdroj:[35]

Na ukázce dat ve tvaru JSON je znázorněno pole s názvem data, nesoucí hodnoty s názvem parametrů: id, text, phone, sent, send_date a id_user. Toto pole nese SMS zprávy ze serveru do mobilního telefonu.

¹⁴(Secure) File Transfer Protocol

¹⁵Database Management System

¹⁶JavaScript Object Notation

```
{"data":[{"id":"190","text":"Casovana SMS", "phone":"777111222",  
"sent":"0", "send_date":"2013-08-07 19:55:00", "id_user":"1"}  
, {"id":"189","text":"Další zprava", "phone":"777123456", "sent":"0",  
"send_date":"2013-08-07 19:55:00", "id_user":"1"}]}
```

5.3 Mobilní aplikace

Implementace mobilní aplikace probíhala pro platformu Android. V úvodu nejprve stručně popíši základní součásti aplikace pro tuto platformu, později odhalím implementační detaily.

5.3.1 Architektura aplikace pro Android

V této části v rychlosti představím základní stavební kameny každé aplikace pro platformu Android.

Activity

Activity je hlavní součástí programu, bez které by nebylo možné aplikaci spustit. Activity zajišťuje, aby se všechna data od nižších vrstev korektně zobrazila uživateli. Je to tedy prezenční vrstva aplikace. Všechny activity jsou potomkem třídy Activity.

View

Každá activity si vytvoří svoje view, případně načte view XML soboru, kde jsou všechna view zapsána. View představuje základní stavební blok pro komponenty uživatelského rozhraní. Pokud by se jednalo o vývoj webové aplikace, view by bylo něco jako HTML kód. Ve většině případů se view specifikuje staticky v XML souborech, které jsou uloženy ve složce `/res/layout`. View jsou pomocí Activity plněny daty. [36]

Manifest

Každá aplikace musí obsahovat soubor `AndroidManifest.xml` umístěný v kořenovém adresáři. Manifest poskytuje základní informace o aplikaci systému Android. Manifest je načten systémem ještě před samotným spuštěním aplikace. V manifestu musí být zapsány všechny activity a ostatní komponenty aplikace jako jsou služby, broadcast recivery, content providery, název Java balíčku aplikace atd. Manifest také deklaruje, jaké oprávnění musí být povoleny, aby aplikace fungovala. Také určuje minimální úroveň API, na které bude aplikace funkční. [37]

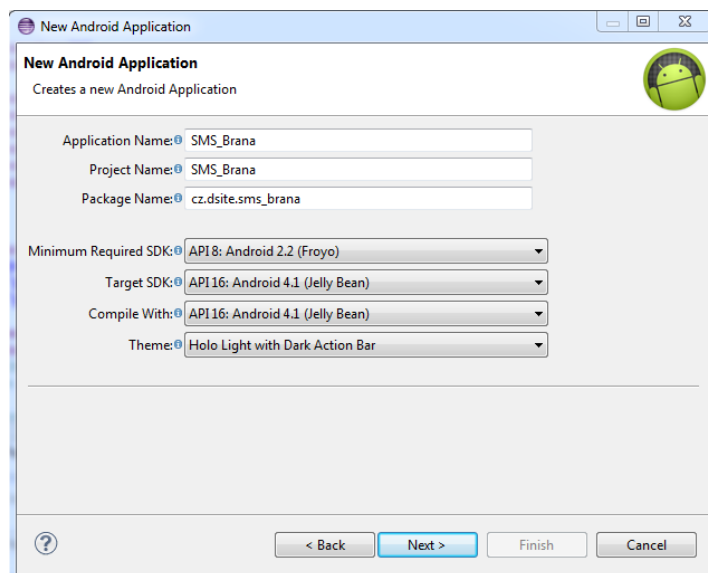
5.3.2 Založení projektu

Aplikace je napsána ve vývojovém prostředí Eclipse s ADT pluginem Android. Prvním krokem při tvorbě aplikace je založení aplikace. V IDE Eclipse založení projektu probíhá takto. Vyberte z menu `File->New->Project...` rozklikněte složku `Android` a vyberte `Android Application Project`. Po provedení těchto kroků, se zobrazí okno, kde je třeba vyplnit základní údaje vytvářené aplikaci (Obrázek 27). Jedná se o název aplikace – dle

něho se jmenuje launcher. Pod tímto názvem se dohledá aplikace v prostředí Google Play v případě publikace. Dalším parametrem je název projektu – ten se zobrazuje pouze vývojáři v prostředí Eclipse. Poslední název, který vývojář musí určit je název balíčku. Balíček musí mít unikátní jméno, většinou se tvoří z názvu domény, kterou vývojář vlastní v opačném pořadí např. cz.domena.

Dále je potřeba nastavit minimální, cílová a sestavovací verze SDK. Minimální verze SDK určuje, na které nejnižší verzi Androidu bude možné aplikaci spustit. Já jsem zvolil verzi API 9, která odpovídá Androidu 2.3, nesoucí označení Gingerbread. Cílové SDK určuje nejvyšší verzi API, ze které jsou využívány funkce. Cílové SDK oznamuje, že jsou využity funkce z vyššího SDK, než je minimální verze, ale je zajištěna zpětná kompatibilita. Pokud jsou například využity efekty, které se zobrazí pouze v nejnovějším SDK, můžeme je do aplikace zahrnout. Ve starších verzích SDK tyto efekty nebudou dostupné. Poslední verzí SDK, kterou je třeba určit, je číslo sestavovací verze. Číslo sestavovací verze určuje, s jakým SDK se bude aplikace kompilovat. Protože se Android velice rychle vyvíjí, je doporučeno pro kompilaci využívat nejvyšší dostupné SDK, aby aplikace mohla využít i ty nejnovější dostupné funkce. Jako cílovou a testovací verzi jsem použil API 16. Tato verze API je implementována v Android 4.1 Jelly Bean, v době vývoje to byly nejnovější verze SDK. Poslední volbou je určení barevného tématu aplikace.

Veškeré údaje, které jsou nastavovány, je možné měnit v průběhu vývoje pomocí úpravy daných hodnot v manifestu.

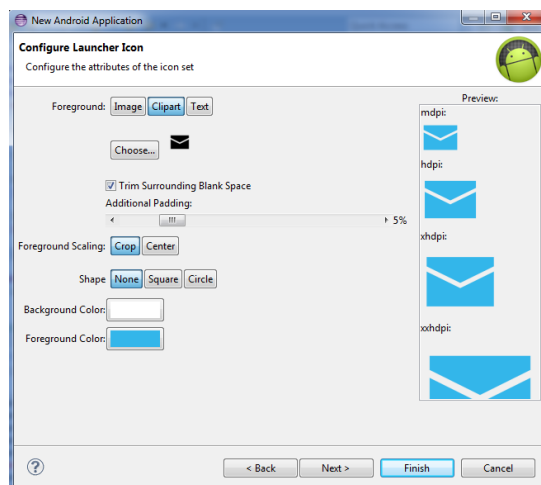


Obrázek 27 – Založení nové Android Aplikace

Zdroj: Vlastní zpracování

Dalším krokem je volba, zda vyžadujeme vytvoření vlastní ikony a hlavní aktivity. Můžeme si také nastavit cestu k projektu, nebo označit projekt jako knihovnu. Pokud chceme vytvořit vlastní ikonu, v dalším kroku si jí můžeme vybrat (Obrázek 28). Ikonu může tvořit obrázek, clipart nebo text, přičemž ikona musí být ve čtyřech velikostech kvůli rozdílnému DPI zařízení. Pokud chceme nastavit jako ikonu clipart, vývojové prostředí

nám nabízí několik desítek připravených klipartů k volnému použití. U těchto obrázků je možné zvolit jejich velikost, zbarvení a také velikost a tvar okrajů.

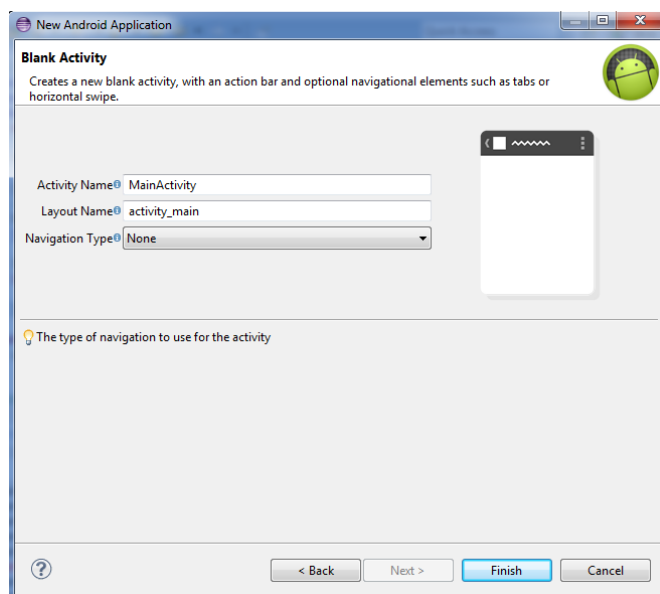


Obrázek 28 – Volba ikony aplikace

Zdroj: Vlastní zpracování

V předposledním kroku si můžeme vybrat typ hlavní aktivity. Na výběr je Blank Activity, FullScreenActivity a Master/Detail Flow. Většinou je nejvhodnější zvolit Blank Activity, což je prázdná aktivita. Té později upravíme layout dle vlastních požadavků.

Poslední obrazovka (Obrázek 29) nabízí volbu názvu aktivity, který jsem ponechal na základní MainActivity.java. Stejně je možné změnit název layoutu přidruženého k aktivitě. Na závěrečné obrazovce můžeme také zvolit typ navigačního menu. Doporučuji však vytvořit vlastní menu, protože nabízené možnosti jsou podporovány až od SDK 11. Zbývá už jen dokončení založení projektu tlačítkem finish.



Obrázek 29 – Volba navigace aktivity

Zdroj: Vlastní zpracování

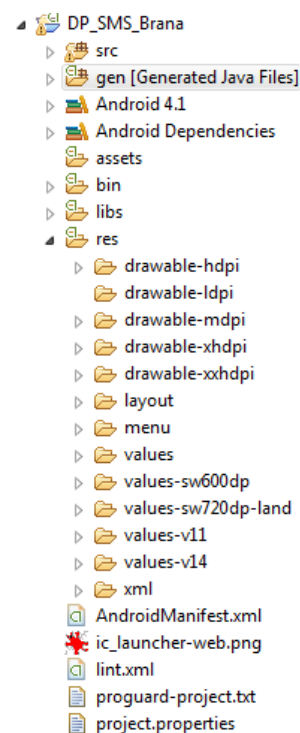
5.3.3 Struktura adresářů

Obrázek 30 ukazuje strukturu adresářů mobilní aplikace. Ve složce `src` se nachází balíčky se zdrojovými kódy. Složka `gen` je generována kompilátorem a slouží pro build aplikace. `Android 4.1` a `Android Dependencies` obsahují knihovny, se kterými aplikace pracuje. Do ostatních složek uživatel běžně nezasahuje a důležitá je pro vývojáře až složka `res`. Význam podsložek adresáře `res` je:

Složky `drawable` uchovávají grafické soubory pro displeje s různým DPI¹⁷ (`hdpi`, `ldpi`, `mdpi`, `xhdpi`, `xxhdpi`). Pokud jsou tedy vytvořeny obrázky, které aplikace bude zobrazovat, měly by být uloženy ve všech pěti kvalitách.

Složka `layout` uchovává xml soubory pro definici struktury aktivit. Tyto soubory jsou načteny jako view jednotlivých aktivit v metodě `onCreate`, která se volá při vytváření aktivity. View aktivity `main` načítá aktivita `ActivityMain` tímto způsobem:

```
setContentView(R.layout.activity_main);
```



Obrázek 30 – Struktura adresářů
Zdroj: Vlastní zpracování

Další složkou, která je využívána v projektu, nese název `menu`. Jak název napovídá, uchovává v sobě menu aplikace. Soubor `main.xml` obsahuje hlavní menu, které je zobrazeno po stisku hardwarového tlačítka menu. Obsah souboru `main.xml` vypadá takto:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item android:id="@+id/queue"
      android:icon="@android:drawable/arrow_down_float"
      android:title="Zobrazit Frontu"/>
.
.
.
</menu>
```

Složka `values` uchovává xml soubory, které slouží pro nastavování datových typů string, polí, vzhledu a dalších surovin, se kterými dále pracují layouty aktivity a třídy aplikace. Pokud bychom tedy chtěli nastavit nějaký text do layoutu, nezapisujeme ho přímo, ale pomocí odkazu do patřičného souboru ve složce `values`. Soubor `strings.xml` by měl tedy obsahovat všechny řetězce, které jsou staticky načítány v layoutech.

¹⁷ Dots per inch

Pokud chci nastavit nějaký text v layoutu, nepostupuji tedy takto (ačkoliv postup je také funkční):

```
<EditTextPreference
android:key="interval"
android:summary="Zadejte celé číslo"
android:title="Interval stahování zpráv [s]"
android:defaultValue="5"/>
```

Ale nejprve bych měl hodnoty definovat v patřičném souboru values, v tomto případě se jedná o soubor strings.xml, kde přidám mezi tagy <resources></resources> hodnoty:

```
<stringname="interval">Interval stahování zpráv [s]</string>
<stringname="intervalDesc">Zadejte celé číslo</string>
<stringname="intervalValue">5</string>
```

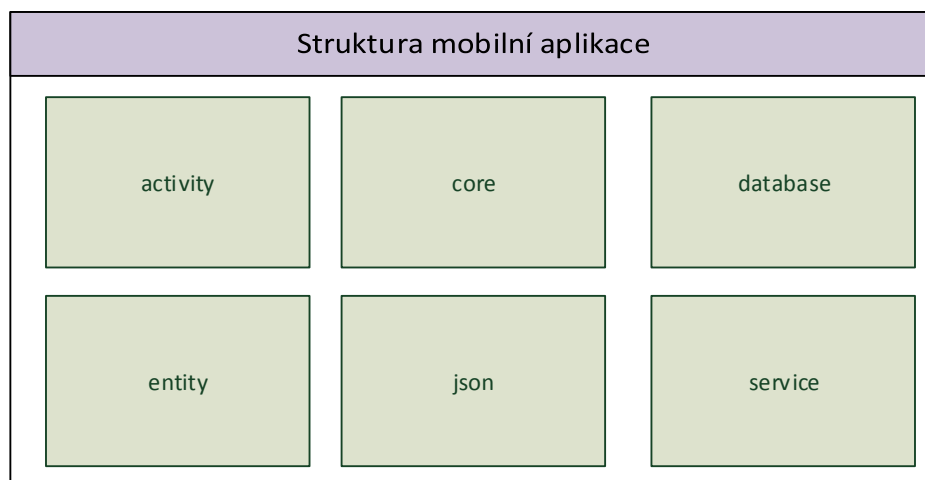
V souboru definující layout poté hodnoty načtu tímto způsobem:

```
<EditTextPreference
android:key="interval"
android:summary="@string/intervalDesc"
android:title="@string/interval"
android:defaultValue="@string/intervalValue"/>
```

Výhoda tohoto řešení spočívá v tom, že uložené hodnoty mohou opakovaně být použity v různých layoutech, aktivitách a souborech. Hlavní výhodou je však to, že si mohou definovat příslušné hodnoty surovin např. pro různé jazykové mutace, rozlišení a verze API (složky values-v11, values-sw600...).

5.3.4 Balíčky aplikace

Pro lepší přehlednost je aplikace rozdělena do balíčků, které ilustruje Obrázek 31.



Obrázek 31 – Balíčky mobilní aplikace

Zdroj: Vlastní zpracování

Balíček activity sdružuje všechny třídy dědicí od Activity. Z toho vyplývá, že se jedná o třídy disponující uživatelským rozhraním.

Core obsahuje hlavní třídu SmsSender, která obstarává klíčovou funkci rozesílání zpráv. Tato třída obsahuje rozhraní ISmsSender. Balíček zahrnuje také třídu Token, která udržuje, nastavuje a vrací hodnotu tokenu při komunikaci přes API.

Balíček database obsahuje adapter, handler a další třídy, které slouží pro veškerou práci s interní databází SQLite.

Entity obsahuje datový typ Message, navržený pro účely udržování hodnoty textové zprávy, se kterou právě systém pracuje.

Balíček json slouží pro získávání dat ve formátu JSON přes HTTP requesty. Obsahem balíčku je také třída, která zpracovává získaná data v podobě JSON do datového typu Message.

Service sdružuje třídy, zajišťující doplňkové služby, které využívají ostatní třídy a activity. Jedná se o třídu CalendarFunctions, jenž poskytuje potřebné funkce při práci s datem a také o třídu ConnectonInfo poskytující informace o připojení k síti.

5.3.5 Obsah manifestu

Význam slova manifest je popsán v kapitole Architektura aplikace pro Android. Zde již pouze ukáží konkrétní implementaci souboru `AndroidManifest.xml`. Obsah manifestu je uzavřen do párového tagu `<manifest>`, tento tag obsahuje tyto důležité parametry:

```
package="cz.dsite.dp_sms_brana"  
android:versionCode="1"  
android:versionName="1.0"
```

Package obsahuje název hlavního balíčku aplikace. Pokud budu dále chtít zapsat nějakou třídu do manifestu, stačí zadat `.NázevTřídy` místo `cz.dsite.dp_sms_brana.NázevTřídy`. Parametry `versionCode` a `versionName` slouží k verzování systému při aktualizacích.

Následuje tag `uses-sdk`, který určuje minimální a cílovou verzi SDK:

```
<uses-sdk  
android:minSdkVersion="9"  
android:targetSdkVersion="16"/>
```

Velice důležitou součástí jsou tagy `uses-permission`. Ty definují práva, která aplikace vyžaduje pro svůj běh. Tyto požadavky jsou uživateli zobrazeny před instalací aplikace a musí je pro instalaci odsouhlasit. V případě mé aplikace je potřeba odsouhlasit přístup aplikace k internetu, odesílání SMS zpráv, běhu služby na pozadí, a zjišťování stavu sítě a telefonu.

```
<uses-permission android:name="android.permission.INTERNET" />
```



```

<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permissionandroid:name="android.permission.SEND_SMS" />
<uses-permissionandroid:name="android.permission.WRITE_SMS" />
<uses-
permissionandroid:name="android.permission.ACCESS_BACKGROUND_SERVICE"/>
<uses-permissionandroid:name="android.permission.READ_PHONE_STATE" />

```

Následuje značka <application>, která definuje obsah aplikace. Obsahuje atributy icon pro definici ikony, label pro nastavení názvu aplikace a theme pro volbu vzhledu. V tomto tagu musí být zapsány všechny používané activity, jinak je nelze spustit. Za povšimnutí stojí rozdílný zápis hlavní aktivity MainActivity, který zajistí, že tato aktivita se po instalaci programu zobrazí v seznamu instalovaných aplikací.

```

<application
android:allowBackup="true"
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme">
<activity
android:name=".MainActivity"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>

<activity android:name=".SMSListActivity"/>
<activity android:name=".SendSmsToDatabase"/>
<activity android:name=".StatisticsActivity"/>
<activity android:name=".SettingsActivity"
android:label="@string/title_activity_settings">
</activity>

</application>

```

5.3.6 Implementace odesílání zpráv

Odesílání textových zpráv musí být prováděno ve vlákne, protože než jsou textové zprávy odeslány, musí být kontrolována fronta zpráv dostupná přes API na síti Internet. Jedná se o asynchronní komunikaci, kterou Android nepovolí vykonávat v běžném UI vlákne. Proto třída SmsSender dědí od třídy AsyncTask, která automaticky vytváří a spravuje vlákna.

Metoda pro odesílání zpráv je vyvolávána v na pozadí, každých x vteřin v intervalu, který je stanoven v nastavení aplikace. Pokud jsou však právě rozesílány zprávy, aplikace před zpracováním nových musí počkat, než jsou všechny odeslány.

```

@Override
protected String doInBackground(String... arguments) {
    while (running==true) {
        try {
// pokudjižneodesílámzprávy, zkontroluji, zdanejsouvefrontěnové
            if (sending == false)
            {

```

```
// odeslání nových sms stažených přes JSON, kontrola nových sms každých x
// vteřin dle nastavení uživatele .
        sendNewSms ();
    }
    // uspání vlákna x vteřin
    int timeSinceLastUpdate = 1000 *
SettingsActivity.getInterval(this.context);
Thread.sleep(timeSinceLastUpdate);
```

Metoda sendNewSms nejprve kontroluje, zda je dostupné připojení k síti Internet:

```
if (ConnectionInfo.isConnected(context)) {
    Pokud je připojení v pořádku, jsou stažena data ve tvaru JSON:
```

```
JSONObject json =
JSONFunctions.getJSONfromURL(SettingsActivity.getApiData(this.context));
```

Pokud není vrácený objekt hodnoty null, načtu z dat JSON pole:

```
JSONArray messages = json.getJSONArray("data");
```

V případě, že pole zpráv je nenulové, jsou zprávy načteny do HashMap a dále jsou po jedné zpracovávány, nejprve je odstraněna diakritika:

```
String normalizedText = removeDiacritics(e.getString("text"));
```

Dále probíhá kontrola, zda se jedná o časovanou zprávu. Pokud ano, musí metoda readyToSend zjistit, zda je již možné zprávu odeslat:

```
String sendDate = e.getString("send_date");
if (sendDate != "null" && sendDate != null && sendDate != "") {
    if (CalendarFunctions.readyToSend(sendDate)) {
```

Pokud je zpráva připravena k odeslání, je zavolána metoda sendSms, která zprávu odešle do GSM sítě a dále metoda updateSmsStatus, která oznámí serverové části, že zpráva byla odeslána. Nakonec je odeslaná zpráva (bez obsahu textu) uložena do SQLite databáze.

```
result=sendSms(e.getString("phone"), normalizedText);
if (result==1) {
updateSmsStatus(e.getInt("id"),e.getInt("id_user"));
message = new Message((long) e.getInt("id"),(long) e.getInt("id"),
"sender", e.getString("phone"), CalendarFunctions.now(),null,
this.context);
databaseHandler.saveMessage(message); // vložení do SQLite
```

Samotné rozeslání zpráv přes GSM je už velice jednoduché, pouze je potřeba zkontrolovat, zda zpráva je delší než maximální délka zprávy 160 znaků. Pokud je text delší, zpráva musí být rozdělena do více částí. Odesílání zpráv má na starosti

knihovna: android.telephony.SmsManager.

```
SmsManager manager = SmsManager.getDefault();
int length = message.length();
if (phone.length() > 0 && length > MAX_SMS_MESSAGE_LENGTH) {
ArrayList<String> messagelist = manager.divideMessage(message);
manager.sendMultipartTextMessage(phone, null, messagelist, null, null);
```

```

        return 1;
    } else {
        if (phone.length() > 0) {
            manager.sendTextMessage(phone, null, message, null, null);
        }
        return 1;
    }
    return 0;
}

```

5.3.7 Odstranění Android limitu

Při programování aplikace pro hromadné rozesílání zpráv jsem narazil na nepříjemnou věc, o které jsem se před implementací návrhu nikde nedozvěděl. Jedná se o tak závažnou skutečnost, že jsem jí vyhradil samostatnou podkapitolu. Systém Android má z nějakého důvodu nastavený limit 100 odeslaných zpráv za hodinu. Poté je nutné každou další odeslanou SMS zprávu ručně potvrdit. Naštěstí je Android otevřený systém, a tak jsem našel radu, jak se tohoto omezení zbavit. Stačí mít systém Android s root oprávněním. Toto oprávnění je nejvyšší možné a činí z běžného uživatele systémového superuživatele. Návody na to, jak provést root telefonu se různí od typu, a tak nebudu uvádět konkrétní způsob. Já na svém testovacím přístroji HTC Dezire Z toto oprávnění vlastním, takže nebyl problém provést následující kroky.

1. Telefon je připojený k počítači s povoleným laděním přes USB.
2. Do příkazové řádky je zadán tento příkaz:

```
adb shell
sqlite3 /data/data/com.android.providers.settings/databases/settings.db
```

3. Pokud tato operace proběhla v pořádku, je možné provádět SQL dotazy nad databází `settings` uchovávající systémové nastavení telefonu. Na začátku řádky by po tomto kroku mělo být toto:

```
sqlite>
```

4. Následujícím příkazem se zvýší limit odchozích SMS zpráv na libovolnou hodnotu, v tomto případě 999 999.

```
INSERT INTO secure (name, value) VALUES ('sms_outgoing_check_max_count',
999999);
```

5. Posledním příkazem se zruší interval kontroly limitu odchozích zpráv.

```
INSERT INTO secure (name, value) VALUES ('sms_outgoing_check_interval_ms',
0);
```

V bodech 4 a 5 se může lišit název tabulky na základě verze systému Android. Pokud databáze `settings` neobsahuje tabulku `secure`, bude se jednat o starší název tabulky `gservices`. Uvedené úpravy hodnot tabulky lze provést i pohodlnějším způsobem přes grafické rozhraní správce souborů podporujících root oprávnění (např. Root Browser). [38]

5.3.8 Implementace fronty zpráv

Mobilní aplikace poskytuje zobrazení fronty zpráv čekajících na odeslání. O tuto funkci se stará třída `SMSListActivity`. Zobrazení zpráv v seznamu `listView` zajišťuje třída `SimpleAdapter`. [39]

Nejprve je fronta zpráv stažena z adresy poskytující data ve formátu JSON.

```
jObject = JSONfunctions.getJSONfromURL(adress);
```

Poté jsou zprávy parsovány do seznamu.

```
message = dataJsonParser.parse(jObject);  
/** Použité klíče pro Hashmap */  
String[] from = { "text", "phone" };  
  
int[] to = { R.id.sms_text, R.id.sms_phone };  
  
//adaptér pro zprávy čekající na odeslání  
  
adapter = new SimpleAdapter(getBaseContext(), message,  
R.layout.lv_messages, from, to);  
setListView((ListView) findViewById(R.id.lv_messages));
```

Nakonec je adapter načten do listView.

```
listView.setAdapter(adapter);
```

5.3.9 Implementace tříd spravujících databázi

Aplikace ukládá odeslané SMS zprávy do interní databáze SQLite. Aplikace přistupuje k tabulkám přes speciální třídy. Nejnižší se nachází třída Database, která dědí od SQLiteOpenHelper. Tato třída vytváří novou databázi, pokud databáze ještě nebyla založena, a také ji aktualizuje, pokud se vyskytne nová verze.

Nad třídou Database se nachází DBAdapter, který vytváří instanci třídy Database. Tento adaptér poskytuje metody open a close pro otevření a uzavření databáze.

Třidu DBAdapter dědí potom adaptéry pro jednotlivé tabulky. V případě mé aplikace se jedná o adapter DBArchiveAdapter, který definuje a vytváří tabulku archive. Také poskytuje metody, které vykonávají SQLite dotazy nad tabulkou. Vložení záznamu do tabulky vypadá v adapteru tabulky takto:

```
public long insertMessage(long mid, String sender, String recipient,  
String date)  
{  
    ContentValues values = buildValues(mid, sender, recipient, date);  
    return db.insert(TABLE_ARCHIVE, null, values);  
}
```

Z jednotlivých tříd tabulek jsou vytvořeny instance ve třídě DatabaseHandler, která si otevře databázi, provede SQL dotaz a databázi opět zavře. Finální metoda pro vložení zprávy do tabulky pak vypadá takto:

```
public void insertMessage(Message message){  
    dbArchiveAdapter.open();  
    dbArchiveAdapter.insertMessage(message.getId(), message.getSender(),  
message.getRecipient(), message.getDate());  
    dbArchiveAdapter.close();  
}
```

Pokud jsou pomocí dotazu vrácena data, jsou načtena do kurzoru, který postupně procházím pomocí operací `cursor.moveToFirst()`, `cursor.isAfterLast()` a `cursor.moveToNext()`. Data jsou z kurzoru načtena např. pomocí zavolání metody `GetDatovýTyp (getString, getLong..)` s indexem sloupce, který je načítán.

```
cursor.getString(cursor.getColumnIndex(DBArchiveAdapter.COLUMN_SENDER))
```

5.3.10 Komunikace se serverovou částí

Komunikace se serverovou částí probíhá pomocí veřejně dostupného API. Data jsou načítána ze serveru metodou `getJSONFromUrl`. Před zahájením komunikace je nejdříve ověřeno, zda již existuje token pro komunikaci. Pokud token není dostupný, zažádá aplikace server o přidělení tokenu na základě autentizace pomocí API klíče. Další komunikace probíhá s tokenem, dokud nevyprší jeho platnost, což je oznámeno návratovým kódem HTTP hlavičky 449.

Při prvním spuštění metody nejprve zažádám server o přidělení tokenu:

```
if (Token.getToken()==null) {
    Token.setToken(getTokenFromJson(url));
}
```

Po přidělení tokenu uložím jeho hodnotu spolu s uživatelským jménem do `ArrayListu`, který je později předán jako parametry metody `POST`.

```
nameValuePairs.add(new BasicNameValuePair("nick", SettingsActivity.getUserName(this.context)));
nameValuePairs.add(new BasicNameValuePair("token", Token.getToken()));
```

Komunikace se serverem probíhá pomocí třídy `HttpPost`, která zajišťuje zaslání požadavků `POST` prostřednictvím `http` protokolu.

```
// http post
try {
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(url);
    // heslo pro ověření identity

    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
    HttpResponse response = httpClient.execute(httpPost);
    // získání návratového kódu - 449 = nutná reautORIZACE
    responseCode = response.getStatusLine().getStatusCode();
    if (responseCode == 449) {
        Token.setToken(null);
        // spustíme metodu znovu, načteme při tom token
        getJSONFromUrl(url);
    }
    HttpEntity entity = response.getEntity();
    is = entity.getContent();
}
```

Odpověď od serveru je uložena do pole dat typu JSON:

```
// convert response to string
try {
    BufferedReader reader = new BufferedReader(new InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    result = sb.toString();
} catch (Exception e) {
    Log.e("log_tag", "Error converting result " + e.toString());
}

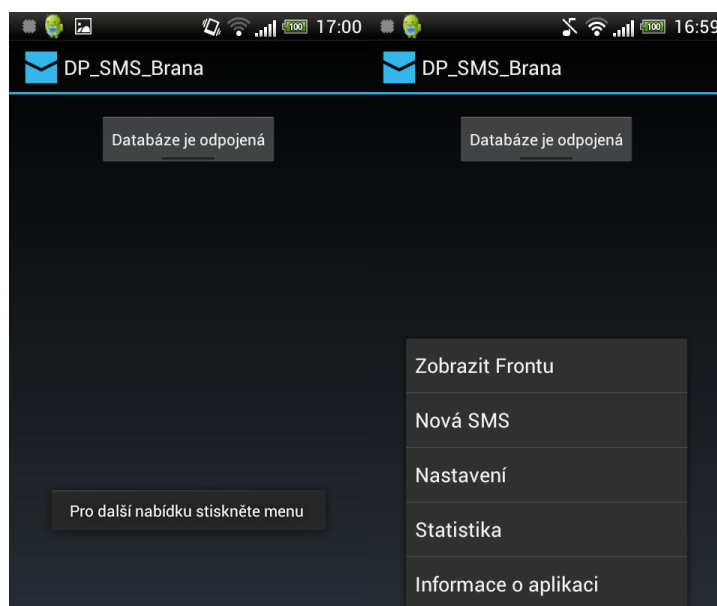
try {
    JSONArray jsonArray = new JSONArray(result);
} catch (JSONException e) {
    Log.e("log_tag", "Error parsing data " + e.toString());
}

return jsonArray;
```

5.3.11 Uživatelské rozhraní

Mobilní aplikace disponuje uživatelským rozhraním, které umožňuje snadnou správu programu, a poskytuje také další doplňkové funkce. V této části jsou vyobrazeny jednotlivé části uživatelského rozhraní.

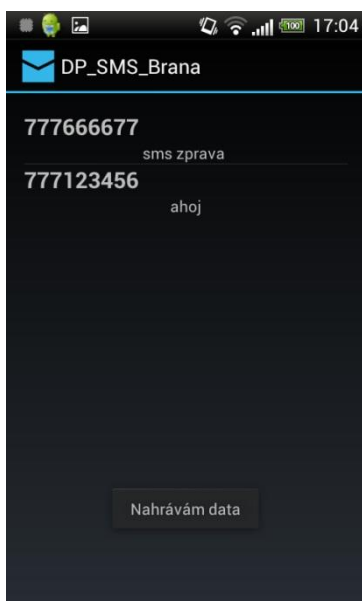
První dvě obrazovky (Obrázek 32) ukazují vzhled aplikace hlavní aktivity. Hlavní obrazovka obsahuje tlačítko pro zahájení rozesílání zpráv a také menu, které je zobrazeno po stisku hardwarového tlačítka menu.



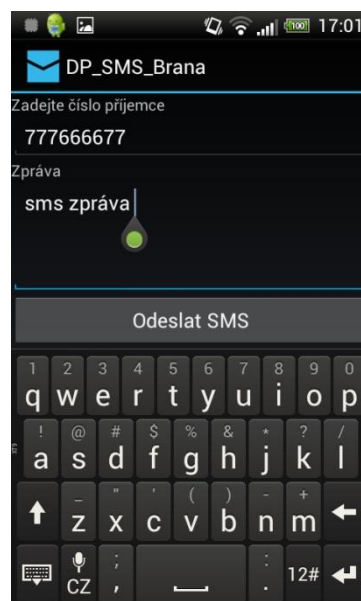
Obrázek 32 – Úvodní obrazovka aplikace

Zdroj: Vlastní zpracování

Na obrázku vlevo dole (Obrázek 33) je znázorněn výpis zpráv čekajících na odeslání, vpravo spuštěna aktivita pro odesílání zprávy do databáze.

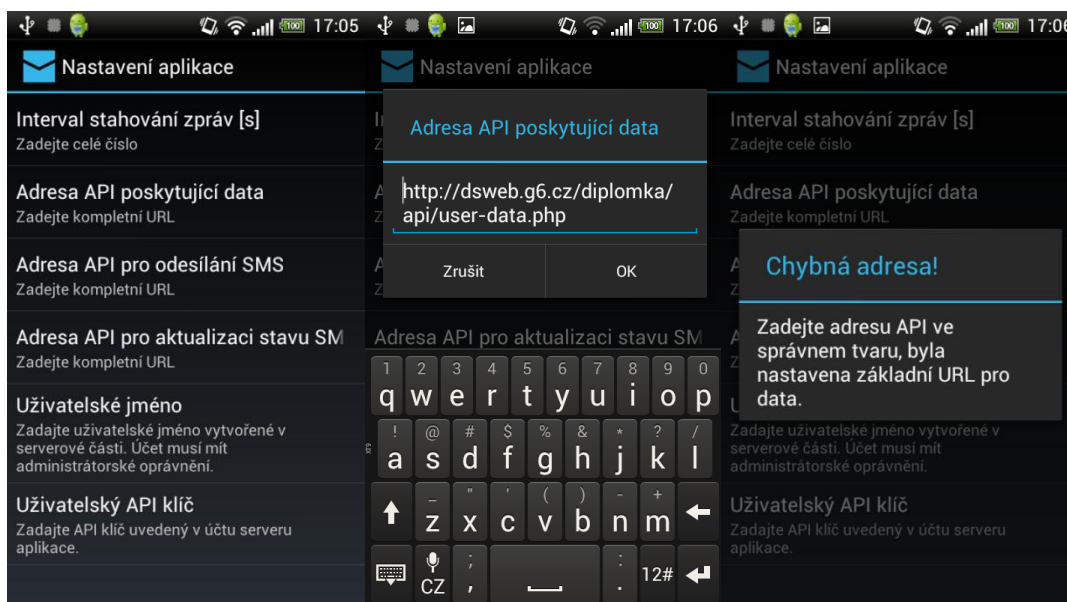


Obrázek 33 – Fronta zpráv
Zdroj: Vlastní zpracování



Obrázek 34 – Psaní SMS
Zdroj: Vlastní zpracování

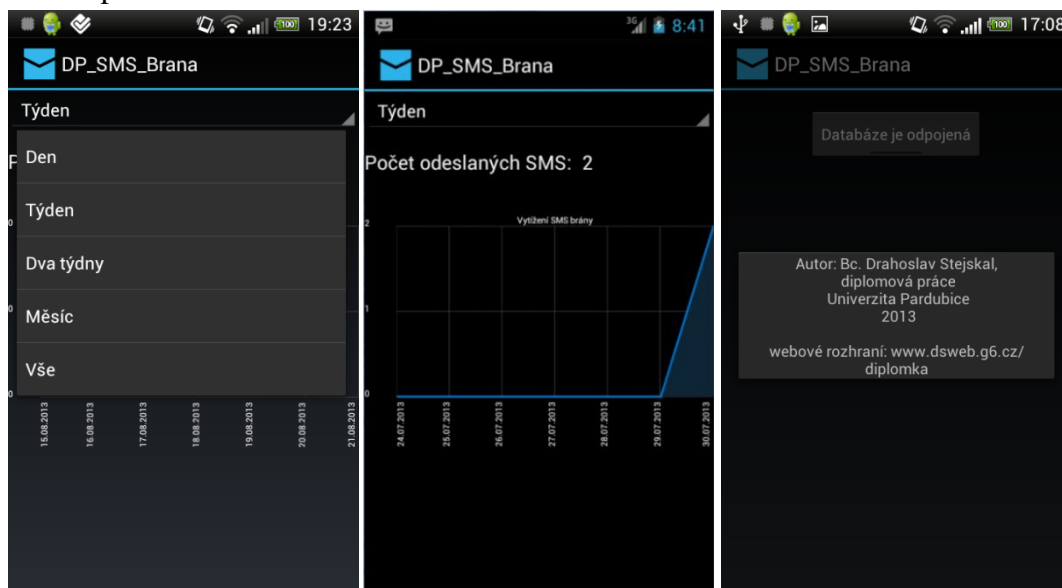
Na následujících obrazovkách (Obrázek 35) je uživatelské rozhraní pro nastavení aplikace. Obrazovka *a* ilustruje položky nastavení aplikace. Prostřední obrazovka *b* znázorňuje, jak vypadá editace záznamu nastavení, obrazovka *c* vpravo nastane při zadání nastavení v nesprávném formátu.



(a) Položky nastavení (b) Editace hodnoty nastavení (c) Validace hodnot

Obrázek 35 – Nastavení aplikace
Zdroj: Vlastní zpracování

Poslední obrazovky (Obrázek 36) znázorňují zleva volbu intervalu pro vykreslení grafu, dále graf vykreslující počet odeslaných zpráv za uplynulý týden a poslední obrazovka nese informace o aplikaci.



(a) Volba intervalu

(b) Vykreslení grafu

(c) Informace o aplikaci

Obrázek 36 – Statistika a informace o aplikaci

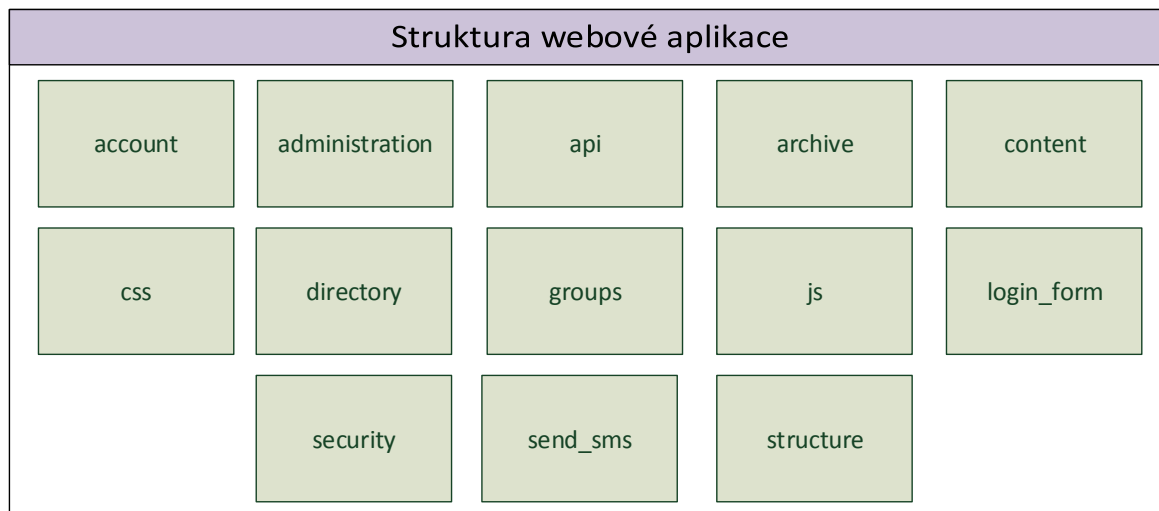
Zdroj: Vlastní zpracování

5.4 Webová aplikace

Webová aplikace slouží jako serverová část pro vkládání SMS zpráv do systému, ale také poskytuje API s dokumentací. Pracovní název projektu je AndrGate a vznikl spojením slov Android a Gate. Webová aplikace je umístěna na bezplatném hostingu Endora na adrese www.dsweb.g6.cz/diplomka/. Webová aplikace je napsána ve skriptovacím jazyce PHP a systém běží na webovém serveru Apache. Aplikace je rozdělena do jednotlivých složek – modulů, které zpřehledňují celou aplikaci.

5.4.1 Struktura a funkce aplikace

Webová aplikace je pro lepší přehlednost členěná do balíčků (Obrázek 37), které uchovávají související PHP skripty, kaskádové styly, obrázky a JavaScript soubory.



Obrázek 37 – Struktura webové aplikace

Zdroj: Vlastní zpracování

Jednotlivé balíčky poskytují aplikaci tyto funkcionality:

- **Account** – Poskytuje informace o uživatelském účtu, umožňuje vytvořit a editovat uživatele.
- **Administration** – Balíček poskytující administrátorské nástroje aplikace.
- **API** – Veřejné rozhraní poskytující základní služby SMS brány.
- **Archive** – Poskytuje archiv zpráv uživatele.
- **Content** – Veřejně dostupný obsah stránek.
- **CSS** – Kaskádové styly a obrázky aplikace.
- **Directory** – Poskytuje soubory pro vytváření, správu a mazání kontaktů.
- **Groups** – Zajišťuje vytváření, správu a mazání skupin.
- **Js** – JavaScript soubory využívané v aplikaci především pro techniku AJAX.
- **Login_form** – Zajišťuje přihlášení a registraci uživatele.
- **Security** – Skripty pro zabezpečení webové aplikace a připojení k databázi.
- **Send_sms** – Umožňuje odesílání textových zpráv do mobilní aplikace.
- **Structure** – Základní stavební prvky webu, které se opakují na více stránkách.

5.4.2 Zabezpečení

Z webového rozhraní mohou uživatelé odesílat SMS zprávy pouze po provedení registrace a přihlášení do systému. V uživatelském účtu se nachází také archiv odeslaných zpráv, proto je třeba zajistit odpovídající zabezpečení. Veškeré vykonávané dotazy jsou zabezpečeny proti SQL injection. V aplikaci je implementován způsob zabezpečení dle dostupných materiálů na internetu. [40]

Prvním krokem je vytvoření databázové tabulky, která bude uchovávat neúspěšné pokusy o přihlášení uživatele. To ochrání účet před útoky hrubou silou.

```
CREATE TABLE `login_attempts` (  
  `user_id` int(11) NOT NULL,  
  `time` VARCHAR(30) NOT NULL  
)
```

Při přihlášení uživatele probíhá kontrola, zda nezadal pětkrát špatné heslo v průběhu dvou hodin. V takovém případě je účet dočasně uzamčený.

```
function checkbrute($user_id, $mysqli) {  
  
    $now = time();  
    // Počítají se loginy z posledních dvou hodin  
    $valid_attempts = $now - (2 * 60 * 60);  
  
    if ($stmt = $mysqli->prepare("SELECT time FROM login_attempts WHERE  
user_id = ? AND time > '$valid_attempts'")) {  
        $stmt->bind_param('i', $user_id);  
        $stmt->execute();  
        $stmt->store_result();  
        if($stmt->num_rows > 5) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```

Pokud se uživatel přihlásí do systému, jsou nastaveny patřičné session pro sezení. Na všech stránkách, kde je nutné pracovat se session je potřeba uvést na vrchu stránky `session_start()`. Pro vyšší stupeň zabezpečení je však užito modifikované obměny.

```
function sec_session_start() {  
    $session_name = 'sec_session_id'; // Název session  
    $secure = false; // Nastaveno na true při použití https  
    $httponly = true; // Toto zaamezí přístupu javascriptu k session  
    // Donucení používání pouze cookies  
    ini_set('session.use_only_cookies', 1);  
    // Získání parametrů cookie  
    $cookieParams = session_get_cookie_params();  
    session_set_cookie_params($cookieParams["lifetime"],  
$cookieParams["path"], $cookieParams["domain"], $secure, $httponly);  
    session_name($session_name); // Nastavení názvu session  
    session_start(); // Spuštění php session  
    session_regenerate_id(true); // obnovení session, smazání té staré.  
}
```

Tato funkce zamezí útočníkům přístup k session id cookie pomocí JavaScriptu. Další ochranou je regenerace session při každém obnovení stránky.

Uživatelské heslo je v databázi uloženo pomocí techniky salted hash a algoritmu SHA-512 což odpovídá 128 hexa číslicím. K heslu je tak navíc připojena náhodná posloupnost znaků, takzvaný salt o stejné délce. Teprve poté je heslo uloženo pomocí hash algoritmu SHA-512. Žádní uživatelé tak nebudou mít stejnou podobu hashe hesla, i když je jejich nehashovaná podoba stejná.

Dalším důležitým prvkem bezpečnosti je provádění hashe hesla již při jeho odeslání z formuláře. To probíhá pomocí JavaScriptu. Následující JavaScript provede hash hesla:

```
function formhash(form, passw) {  
    //  
    var p = document.createElement("input");  
    // přidá nový element do formuláře  
    form.appendChild(password);  
    password.name = "password";  
    password.type = "hidden"  
    password.value = hex_sha512(passw.value);  
    //Nehashované heslo je smazáno  
    passw.value = "";  
    // odeslání formuláře  
    form.submit();  
}
```

Tato technika je v práci využita například ve formuláři pro přihlášení. Na tlačítku pro odeslání funkce zavolána:

```
<input type="submit" name="submit" value="Login" class="button"  
onclick="formhash(this.form, this.form.password);" />
```

Výše uvedená zabezpečení jsou implementována v serverové části. Při nasazení v reálném provozu, by však toto stále nestačilo. Bylo by třeba šifrovat archiv zpráv a kontakty uživatelů pro případ napadení databáze. Dalším nutným krokem by byla realizace API přes https protokol.

5.4.3 Odeslání zprávy

Jedna z hlavních funkcí, jež měla serverová část poskytovat je odesílání zpráv přes webové rozhraní. Kromě odeslání zprávy na zadané číslo aplikace poskytuje také adresář kontaktů. Ten je zobrazen vedle pole pro text zprávy a pomocí techniky AJAX dynamicky načítá kontakty ze skupin a vkládá je do pole příjemců. U odesílané zprávy je možné určit čas odeslání, který je dynamicky zobrazen po stisku odkazu „Určit čas odeslání“. Ukázka uživatelského rozhraní pro odeslání zprávy je znázorněna v kapitole Uživatelské rozhraní.

SMS brána umožňuje odeslání textové zprávy více příjemcům. Jednotlivé telefonní kontakty musí být odděleny středníkem. Po odeslání zprávy jsou jednotliví příjemci zprávy z formuláře načteny tímto způsobem:

```
$numbers = preg_replace('/\+/','00', $_POST['number']); //odstranění +
$numbers = explode(";", $numbers);
$length = count($numbers);
```

Dále následuje ověření, zda má uživatel dostatečný kredit na odeslání všech zpráv.

```
$q = "SELECT credit from user where nick=?";

$stmt = $mysqli->prepare($q);
$stmt->bind_param("s", $_SESSION['username']);
$stmt->execute();

if ($stmt->bind_result($credit_r)) {
    while ($stmt->fetch()) {
        $credit = $credit_r;
    }
}
```

Zprávy jsou ukládány do databáze do tabulky sms, která dočasně uchovává zprávy k odeslání do té doby, než jsou odeslány přes mobilní telefon.

```
for ($i = 0; $i < $length; $i++) {
    $number = trim($numbers[$i]);

    if (is_numeric($number)) {
        if ($credit > $length - 1) {

            if ($timing == 1) {

                $q = "INSERT INTO sms (text, phone, sent, send_date, user_iduzivatel)
VALUES(?,?,0,?,?)";
                $stmt = $mysqli->prepare($q);
                $stmt->bind_param("sisi", $_POST['text'],$number, $send_time,
$_SESSION['user_id'] );
            }
        }
    }
}
```

Pokud není nastaven čas odeslání zprávy, provede se alternativní SQL dotaz, který nevkládá do tabulky sms parametr se send_date. Připravené SQL dotazy jsou provedeny příkazem:

```
$sent = $stmt->execute();
```

Pokud nabývá proměnná \$sent hodnoty 1, SMS byla úspěšně odeslána. Počet úspěšných vložení do databáze musí být roven počtu odeslaných zpráv, jinak je uživatel informován o tom, že některé zprávy nebyly úspěšně odeslány. Uživatel je rovněž upozorněn, pokud nemá dostatečný kredit na odeslání textové zprávy.

Kredit za odeslání zprávy je uživateli odečten až po odeslání zprávy přes mobilní aplikaci. Po odeslání je také zpráva přesunuta z fronty zpráv SMS do tabulky archive. Protože je u těchto operací nutné zachovat konzistenci, musí být tyto operace provedeny pomocí transakce. Toho se docílí zakázáním automatického commitu kritických SQL dotazů:

```
$mysqli->autocommit (FALSE);
```

Na konci kritické sekce je `autocommit` opět povolen a v případě jakékoliv chyby jsou všechny dotazy vráceny do původního stavu metodou `rollback`.

```
$mysqli->autocommit(TRUE);
} catch (Exception $e) {
    $mysqli->rollback();
}
```

5.4.4 API

Serverová část dle zadání disponuje veřejným rozhraním, které poskytuje základní operace SMS brány. Umožňuje uživateli odesílat textové zprávy do centrální databáze z jeho systému, případně načíst archiv odeslaných zpráv a frontu jeho zpráv čekajících na odeslání. Funkčnost API je důkladně otestována, protože přes něj úspěšně komunikuje aplikace na mobilním telefonu. Data mohou být zasílána pomocí http metod GET a POST, návratové hodnoty jsou ve formátu JSON. Návratové kódy a podrobná dokumentace jsou uvedeny v příloze B této práce. Tato dokumentace, je dostupná také online ve webové části.

Níže uvedená funkce je volána na všech stránkách poskytující veřejné rozhraní. Tento kód zajistí, že přístup k datům a komunikaci se serverovou částí mají pouze oprávnění uživatelé. Funkce je volána při zaslání soukromého API klíče, což probíhá při zahájení komunikace nebo při vypršení platnosti přiděleného tokenu.

```
function apiAuthentication ()
{
if(isset($_REQUEST['password'])&& isset($_REQUEST['nick'])) {
createToken();
break;
} else if (isset($_REQUEST['token'])&& isset($_REQUEST['nick'])) {
//kontrola platnosti tokenu 1 - platný, 0 neplatný
$validToken = validToken($_REQUEST['token'],$_REQUEST['nick']);
if ($validToken==0) {
//tímto vyzvu uživatele k zaslání hesla, pokud vyprší platnost
tokenu
header("HTTP/1.1 449 Retry With");
}
} else {
echo "Není nastaven token ani přihlašovací údaje!";
header("HTTP/1.1 401 Unauthorized");
}
```

Funkce `validToken` kontroluje platnost tokenu. Platnost tokenu je v databázi uložena jako celé číslo vyjadřující počet vteřin od 1.1.1970. Pokud platnost tokenu vypršela, nebo je zaslán chybný token, je uživatel vyzván k autorizaci zasláním HTTP hlavičky `header("HTTP/1.1 449 Retry With");`

5.4.5 AJAX

V implementaci webové aplikace je poměrně hojně zastoupen AJAX, který zajišťuje dynamické zobrazování údajů bez nutnosti obnovení stránky. Proto zde uvedu příklad, který popisuje, jakým způsobem je AJAX použitý. Ukázkový příklad zobrazuje dynamické načítání kontaktů dle zvolené složky kontaktů a následné zkopírování kontaktu do pole příjemců po kliknutí na kontakt. Skript má také na starosti zobrazování zbývajících počtu

znaků SMS zprávy, odesílané zprávy a zobrazení / skrytí časovaného odeslání zprávy. Uváděný skript se nachází ve složce `js` pod názvem `smsSender.js`. Používané operace jsou dostupné díky JavaScript knihovně jQuery.

Protože jsou pomocí techniky AJAX upravovány hodnoty v HTML elementech, jsou skripty prováděny až po úplném načtení celé stránky, proto jsou takto obaleny:

```
$(document).ready(function() {  
  
});
```

Počet zbývajících znaků je aktualizován po každém stisku a uvolnění klávesy v elementu s identifikátorem `text`. K tomu slouží funkce `keydown` a `keyup`. Výsledný počet znaků je vykreslen do elementu s názvem `counter`. Proměnná JavaScriptu je vytvořena pomocí klíčového slova `var`. Pokud je zbývajících počet znaků menší nebo roven deseti, je počítadlo zabarveno na červenou barvu.

```
var characters = 1000;  
$("#counter").append("Zbývá <strong>"+ characters+"</strong> znaků");  
$("#text").keydown(function() {  
if($(this).val().length > characters){  
$(this).val($(this).val().substr(0, characters));  
}  
var remaining = characters - $(this).val().length;  
$("#counter").html("Zbývá<strong>"+ remaining+"</strong>  
znaků");  
if(remaining <= 10)  
{  
$("#counter").css("color", "red");  
}  
else  
{  
$("#counter").css("color", "black");  
}  
});
```

Další funkcionalitou, kterou zajišťuje AJAX je kontrola formátu zadávaného čísla. Ta probíhá při každém stisku tlačítka v poli příjemců. Pokud je zadán nepovolený znak, je zobrazen skrytý element s třídou `error_phone`.

```
if(!ValidatePhone(phones)) {  
$(".error-phone").show();  
}else $(".error-phone").hide();
```

Pro zobrazení kontaktů ze skupiny, je třeba nejprve zvolit klikem myši požadovanou skupinu kontaktů. Po kliku jsou kontakty ze skupiny vypsány do elementu nesoucí třídu `contact`. To je umožněno tak, že parametr `value` HTML elementu `option` obsahuje identifikátor shodný s `id` adresáře v databázi. Kontakty jsou ve skutečnosti vypsány do pomocného PHP skriptu `contacts.php`, jehož obsah je načten do prázdného HTML elementu `select` nesoucí třídu `contact`.

```
$(".directory").click(function()
```

```

    {
        var id=$(this).val();
        var dataString = 'id='+ id;

        $.ajax
        ({
            type: "POST",
            url: "contacts.php",
            data: dataString,
            cache: false,
            success: function(html)
            {
                $(".contact").html(html);
            }
        });
    });
});

```

Při výběru kontaktu je díky parametru `onchange` HTML elementu `select` volána metoda JavaScriptová metoda `getSelectedContact`.

```
onchange="getSelectedContact () "
```

Kontakt je po výběru pomocí vlastnosti `selectedIndex` přidán do pole příjemců s identifikátorem `recipients`. Z hodnoty kontaktu je získáno telefonní číslo pomocí metody `getNumber`.

```

function getSelectedContact() {
var contact = document.getElementById("contact");
document.getElementById("recipients").value
+=getNumber(contact.options[contact.selectedIndex].text) + "; ";
}

```

Pokud je stisknuto tlačítko pro výběr všech kontaktů ze skupiny, je zavolána metoda `chooseAll`, která postupně přidá do pole příjemců všechny kontakty.

```

function chooseAll() {
var contact = document.getElementById("contact");
var count= $('select.contact option').length;
for (var i=0;i<=count;i++)
{
    document.getElementById("recipients").value
+=getNumber(contact.options[i].text) + "; ";
}
}

```

Funkce pro zobrazení a skrytí časovaného odeslání zprávy používá funkce `show` a `hide`. Při volání těchto funkcí je nejen zobrazena a skryta část pro určení času odeslání, ale rovněž je nastavena hodnota elementu s identifikátorem `timing`. Na základě této hodnoty je do databáze vložena zpráva s uvedeným časem odeslání, pokud je `timing = 1`.

```

$(".send-time").hide();
$(".display-send-time").show();

$(".display-send-time").click(function()
{
$(".send-time").slideToggle();
$(".display-send-time").hide();
$(".hide-send-time").show();

```

```

        document.getElementById("timing").value=1;
    });

    $(".hide-send-time").click(function()
    {
    $(".send-time").slideToggle();
    $(".hide-send-time").hide();
        $(".display-send-time").show();
        document.getElementById("timing").value=0;
    });

```

Vkládané časové hodnoty jsou validovány při každé změně pomocí validačních funkcí ze skriptu `validation.js`. Pokud zadané hodnoty času a dne odeslání nejsou platné, je zobrazena chybová hláška a není možné odeslat zprávu dříve, než jsou tyto hodnoty opraveny.

5.4.6 Uživatelské rozhraní

Webové rozhraní disponuje uživatelským rozhraním, které využívá modifikovanou CSS šablonu dostupnou na adrese <http://css-free-templates.com/corporate/business-css-template-with-jquery/>. Šablona je volně ke stažení pro osobní i komerční účely. V šabloně byly počestěny popisky a dále bylo vytvořeno logo aplikace s pracovním názvem AndrGate.



Obrázek 38 – Úvodní obrazovka
Zdroj: Vlastní zpracování

Úvodní obrazovka webové aplikace (Obrázek 38) zobrazuje stav, po přihlášení uživatele. Pokud by na stránce byl nepřihlášený uživatel, z horního menu by měl k dispozici pouze odkazy Domů, API a Tarify. Pod hlavním menu se jQuery slider, který je zabudovaný v šabloně.

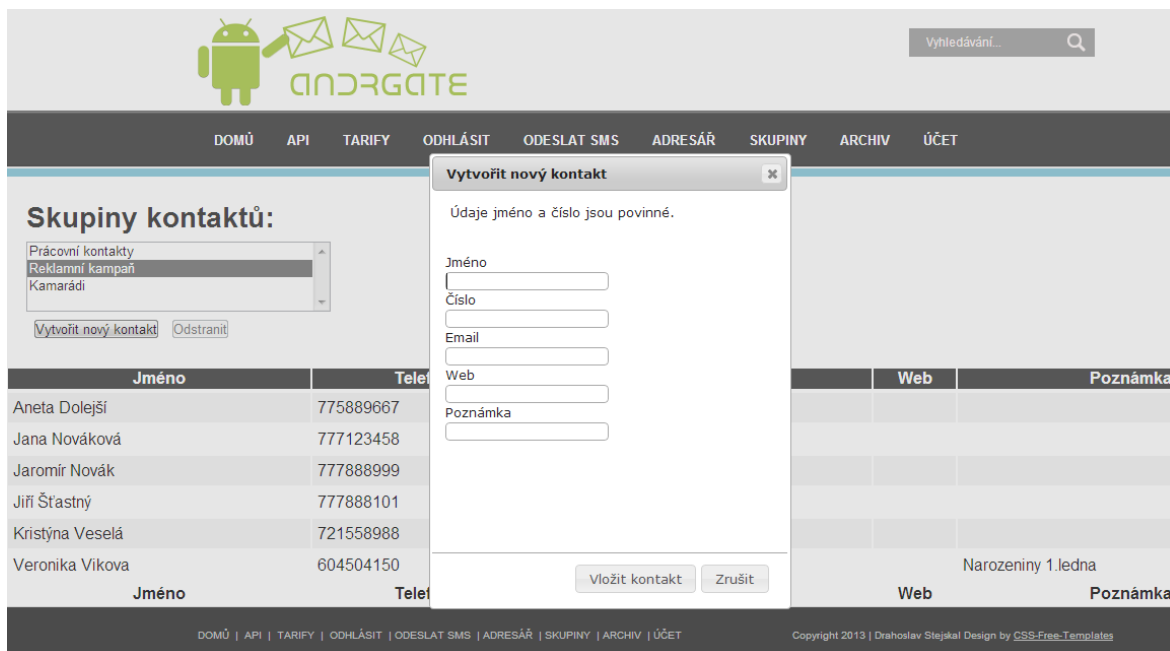
Obrázek 39 ukazuje obrazovku pro odesílání SMS zprávy. Po zvolení možnosti Určit čas odeslání jsou zobrazeny pole pro zadání data a času odeslání zprávy.

Obrázek 39 – Uživatelské rozhraní pro odeslání SMS zprávy
Zdroj: Vlastní zpracování

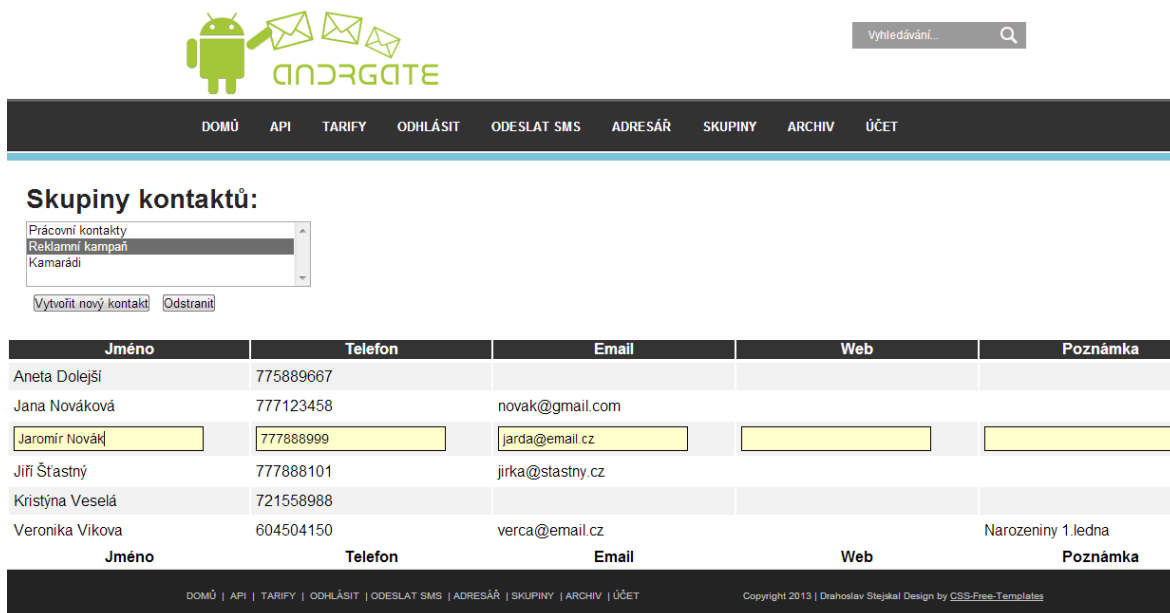
Další obrazovka (Obrázek 40) zobrazuje stránku pro přihlášení uživatele a registraci nového uživatele. Vpravo je vyobrazena situace při zadání nepovolených údajů do polí.

Obrázek 40 – Přihlášení a registrace uživatele
Zdroj: Vlastní zpracování

Obrázek 41 zobrazuje adresář uživatele po stisku tlačítka pro vytvoření nového kontaktu. Poslední obrazovka (Obrázek 42) ukazuje editaci kontaktu po stisku levého tlačítka myši nad kontaktem.



Obrázek 41 – Adresář, vytvoření kontaktu
Zdroj: Vlastní zpracování



Obrázek 42 – Adresář, editace kontaktu
Zdroj: Vlastní zpracování

5.5 Shrnutí

Serverová část aplikace obsahuje všechny funkčnosti, které byly požadovány ve fázi návrhu. Uživatelské prostředí je díky JavaScriptu velice dynamické a umožní uživateli pohodlné užívání všech potřebných nástrojů. Pokud bych v implementaci pokračoval, následovalo by vytvoření administrátorského prostředí pro pohodlnou správu uživatelů, tarifů a také obsahu celého webu. Pro nasazení v provozu by bylo dále třeba šifrovat některé údaje v databázi a poskytnout API přes HTTPS protokol.

Závěr

Diplomová práce ukázala cestu, již je možno vytvořit vlastní SMS bránu prostřednictvím mobilního telefonu s operačním systémem Android. I když je mobilní aplikace poměrně kompaktní, byla pro mě nejobtížnější částí, protože jsem neměl žádné zkušenosti s programováním pro Android. Vývoj pro tuto platformu je v mnoha ohledech rozdílný od programování čisté Javy a tak jsem si nejdříve musel nastudovat literaturu zabývající se vývojem pro tento operační systém. Vývoj pro tuto platformu byl pro mne velice přínosný a dovolil mi alespoň trochu nahlédnout do světa programátorů mobilních aplikací.

Serverová část obsahuje všechny funkčnosti, které by měla poskytovat SMS brána. Práce s aplikací je intuitivní a díky použité technologii AJAX i dynamická. Oblast načítání dat touto technologií pro mě byla nová, a tak chvíli trvalo, než jsem do problematiky pronikl. Nakonec se přes všechny obtíže podařilo vytvořit funkční a použitelné řešení. V serverové části je naznačeno, jakým směrem by se práce mohla dále ubírat, pokud by se jednalo o komerční řešení. V současnosti je však aplikace určena pro osobní použití a kdokoli může dané řešení bezplatně na svém serveru a mobilním telefonu používat.

Výslednou aplikaci jsem testoval v reálném provozu a odeslal jsem tisíce textových zpráv prostřednictvím mobilního telefonu. Zpočátku jsem měl obavy, zda operátor nezablokuje SIM kartu, která odesílá takové množství zpráv. Později se ukázalo, že počet odeslaných SMS zpráv skutečně u operátora Vodafone není omezený. Tuto informaci jsem si ověřil i u samotného operátora, takže nic nebrání hromadnému rozesílání zpráv touto cestou. Věřím, že výsledek práce by mohl být užitečný pro uživatele a menší firmy a budu rád, když někdo bude aplikaci používat v reálném provozu.

Literatura

- [1] **SIM hosting s podporou SMS.** *Daktela* [online]. 2012 [cit. 2013-03-13].
Dostupné z: <http://www.daktela.com/sluzby/simhosting>
- [2] **KUPKA, Pavel a David ŠMÍDA.** *DreamCom* [online]. 2011 [cit. 2013-03-13].
Dostupné z: <http://www.dreamcom.cz/>
- [3] **SIM hosting s podporou SMS.** *Daktela* [online]. 2012 [cit. 2013-03-13].
Dostupné z: <http://www.daktela.com/sluzby/simhosting>
- [4] **SMS brána - o nás.** *Smsbrana* [online]. 2013 [cit. 2013-03-13].
Dostupné z: <http://www.smsbrana.cz/>
- [5] **Direct SMS - vysokorychlostní SMS brána pro firmy.** *Axima* [online]. [cit. 2013-03-17]. Dostupné z: <http://www.axima-brno.cz/direct-sms-ceny.html>
- [6] **Levné volání a SMS přímo z Vašeho telefonu!.** *Klikniavolej* [online]. 2013 [cit. 2013-03-17]. Dostupné z: <http://www.klikniavolej.cz/>
- [7] *Klikniavolej* [online]. [cit. 2013-03-19]. Dostupný na WWW:
http://www.avantura.cz/akce/Android_8.png - obrázek
- [8] **Droid Messenger.** *Droidmessenger* [online]. [cit. 2013-03-19].
Dostupné z: <https://www.droidmessenger.com/>
- [9] **POLÍVKA, Pavel. Remote sms sender.** *Google play* [online]. 2011 [cit. 2013-03-20].
Dostupné z: <https://play.google.com/store/apps/details?id=geekcode.pavel>
- [10] **O2 SMS Connector.** *O2* [online]. [cit. 2013-03-20].
Dostupné z: https://www.o2.cz/pa/191744-sms_mms_e_mail/87271-sms_connector.html
- [11] **SMSC Direct.** *T-mobile* [online]. 2012 [cit. 2013-03-20]. Dostupné z: <http://www.t-mobile.cz/web/cz/firmy/firemni-reseni/smsc-direct#tab9510571748560436>
- [12] **SMS manager.** *Android Developers* [online]. 2013 [cit. 2013-03-22].
Dostupné z: <http://developer.android.com/reference/android/telephony/gsm/SmsManager.html>
- [13] **How to use the SMS compose task for Windows Phone.** *Msdn.microsoft* [online]. 2013 [cit. 2013-08-05]. Dostupné z: [http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh394005\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh394005(v=vs.105).aspx)
- [14] **Interface MessageConnection.** *Blackberry* [online]. 2010 [cit. 2013-04-05].
Dostupné z:
<http://www.blackberry.com/developers/docs/6.0.0api/javax/wireless/messaging/MessageConnection.html>

- [15] **Archived: How to send and receive SMS in Qt for Symbian.** *Nokia Developer* [online]. 2013 [cit. 2013-04-02]. Dostupné z: http://developer.nokia.com/Community/Wiki/Archived:How_to_send_and_receive_SMS_in_Qt_for_Symbian
- [16] **MFMessageComposeViewController Class Reference.** *Apple Developer* [online]. 2011 [cit. 2013-04-02]. Dostupné z: https://developer.apple.com/library/ios/documentation/MessageUI/Reference/MFMessageComposeViewController_class/Reference/Reference.html
- [17] **VĚTVIČKA, Václav.** Spam na mobilu. In: *Dsl* [online]. 2008 [cit. 2013-04-15]. Dostupné z: <http://www.dsl.cz/clanek/1080-spam-na-mobil>
- [18] **NOVÁK, Filip.** Podvodná sms pro zaslání kódu. In: *Silawebu* [online]. 2013 [cit. 2013-04-15]. Dostupné z: <http://blog.silawebu.cz/2013/06/podvodna-sms-pro-zaslani-kodu.html>
- [19] **Elektronické prostředky a elektronické nástroje.** In: *Eagri* [online]. 2013 [cit. 2013-04-15]. Dostupné z: <http://eagri.cz/public/web/mze/legislativa/ostatni/100075017.html>
- [20] **JANÁK, Martin.** Komunitární právní úprava antispamové legislativy a její implementace do právního řádu České republiky. In: *Itpravo* [online]. 2009 [cit. 2013-04-17]. Dostupné z: <http://www.itpravo.cz/index.shtml?x=2202972>
- [21] **JULDA, Jan.** Stalking – co je to stalking, trestní zákon o stalkingu. In: *Julda* [online]. 2010 [cit. 2013-04-18]. Dostupné z: <http://www.julda.cz/2010/02/stalking-co-je-to-stalking-trestni-zakon-o-stalkingu/>
- [22] **40. Zákon trestní zákoník.** In: *Mvcr* [online]. 2009 [cit. 2013-04-19]. Dostupné z: <http://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=c&id=5405>
- [23] **MACICH, Jiří.** Čeští spammeři dostali loni pokuty za 147 000 Kč. Proč je systém bezzubý?. In: *Lupa* [online]. 2013 [cit. 2013-04-19]. Dostupné z: <http://www.lupa.cz/clanky/cesti-spammeri-dostali-loni-pokuty-za-147-000-kc-proc-je-system-bezzuby/>
- [24] **Processes and Threads.** In: *Android developers* [online]. 2013 [cit. 2013-05-03]. Dostupné z: <http://developer.android.com/guide/components/processes-and-threads.html#AsyncTask>
- [25] **KYPTA, Tomáš.** Vývoj pro Android – vylepšujeme aktivity. In: *Abclinuxu* [online]. 2011 [cit. 2013-08-22]. Dostupné z: <http://www.abclinuxu.cz/clanky/vyvoj-pro-android-vylepsujeme-aktivity#asynctask>

- [26] **PreferenceActivity**. In: *Android developers* [online]. 2013 [cit. 2013-05-04].
Dostupné z: <http://developer.android.com/reference/android/preference/PreferenceActivity.html>
- [27] **UJBÁNYAI, Miroslav**. *Programujeme pro android*. Grada, 2012, 13 – 14 str. ISBN 978-80-247-3995-3.
- [28] **MURPHY, Mark L**. *Android 2*. Computer Press, 2011, 230 - 231 str. ISBN 978-80-251-3194-7.
- [29] **UJBÁNYAI, Miroslav**. *Programujeme pro android*. Grada, 2012, 30 str. ISBN 978-80-247-3995-3.
- [30] **Get the Android SDK**. In: *Android developers* [online]. 2013 [cit. 2013-07-02].
Dostupné z: <http://developer.android.com/sdk/index.html>
- [31] **PHP 5 Introduction**. In: *W3schools* [online]. 2013 [cit. 2013-07-02]. Dostupné z: http://www.w3schools.com/php/php_intro.asp
- [32] **JANOVSKÝ, Dušan**. Úvod do JavaScriptu. In: *Jakpsatweb* [online]. 2012 [cit. 2013-07-08]. Dostupné z: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- [33] **Welcome to the NetBeans Community**. In: *Netbeans* [online]. 2013 [cit. 2013-07-12]. Dostupné z: <https://netbeans.org/about/index.html>
- [34] **MySQL databáze - český manuál**. In: *Junext* [online]. 2013 [cit. 2013-07-14].
Dostupné z: <http://www.junext.net/mysql/>
- [35] **Introducing JSON**. In: *Json* [online]. [cit. 2013-07-16]. Dostupné z: <http://www.json.org/>
- [36] **KONEČNÝ, Matěj**. Vyvíjíme pro Android: První krůčky. In: *Zdrojak* [online]. 2012 [cit. 2013-07-16]. Dostupné z: <http://www.zdrojak.cz/clanky/vyvijime-pro-android-prvni-krucky/>
- [37] **The AndroidManifest.xml File**. In: *Android developers* [online]. 2013 [cit. 2013-07-18]. Dostupné z: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [38] **Need help to increase the SMS limit!**. In: *Xda-developers* [online]. 2010 [cit. 2013-07-19]. Dostupné z: <http://forum.xda-developers.com/showthread.php?p=5519546#post5519546>
- [39] **SimpleAdapter**. In: *Android developers* [online]. 2013 [cit. 2013-07-20]. Dostupné z: <http://developer.android.com/reference/android/widget/SimpleAdapter.html>
- [40] **How to Create a Secure Login Script in PHP and MySQL**. In: *Wikihow* [online]. [cit. 2013-07-21]. Dostupné z: <http://www.wikihow.com/Create-a-Secure-Login-Script-in-PHP-and-MySQL>

Příloha A – Příložené CD

Přílohu tvoří CD s elektronickou podobou diplomové práce ve formátu pdf. Dále jsou na nosiči zdrojové kódy k mobilní a webové aplikaci, zkompilovaná mobilní aplikace, soubory obsahující příkazy pro vytvoření databáze a její naplnění daty a pokyny pro instalaci.

Příloha B – Dokumentace API

Dokumentace API

SMS brána disponuje jednoduchým API, které umožňuje implementaci základních funkcí do vlastních aplikací. Přenos dat probíhá pomocí HTTP protokolu. Autorizace je uskutečněna na základě Vašeho nicku a soukromého API klíče, který naleznete v nastavení Vašeho účtu.

Veškerý přenos dat je zabezpečen pomocí tokenů, které jsou přidělovány na dobu 5 minut. Po vypršení platnosti tokenu, nebo v případě nového spojení je nutná autentifikace pomocí API klíče a Vaší přezdívky.

Pokud chcete implementovat odesílání SMS zpráv z Vašeho systému, musíte implementovat alespoň API pro odesílání SMS zpráv (send-sms). Pro přehled o SMS, čekajících na odeslání (např. z důvodu fronty, nebo časované SMS), využijete API pro příjem uživatelských dat (user-data). Ihned po odeslání zprávy přes GSM síť je změněn stav zprávy na odeslaný a zpráva je z fronty přesunuta do archivu. K archivu Vašich zpráv máte přístup přes API archive. V archivu zpráv si můžete zkontrolovat, v kolik hodiny byla Vaše zpráva odeslána.

API umožňuje tyto funkce:

- Odesílat SMS do centrální databáze.
- Přijímat Vaše SMS určené k odeslání.
- Prohlížet a zálohovat archiv zpráv.

Možnosti využití rozhraní pro odesílání SMS

- Pohodlné odesílání textových zpráv z počítače.
- Hromadné rozesílání SMS v určený čas.
- Automatické odeslání zprávy - změna stavu objednávky, komunikační kanál se zákazníky, napojení na alarm.
- Archivace odeslaných zpráv.

HTTP návratové kódy

SMS brána používá návratové kódy protokolu HTTP, který nese informace o výsledku akce. Aplikace poskytuje následující stavové kódy:

Návratový kód	Význam návratového kódu
200 OK	Požadavek byl úspěšný.
201 Created	Objekt byl vytvořen.
401 Unauthorized	Není nastaven token ani přihlašovací údaje, proveďte autentizaci.
402 Payment Required	Pro odeslání SMS nemáte dostatečný kredit.
404 Not Found	Objekt nebyl nalezen.
422 Unprocessable entity	Objekt nebyl vytvořen kvůli chybnému vstupu.
449 Retry With	Platnost tokenu vypršela, je třeba provést autentifikaci.
500 Internal Server Error	Vnitřní chyba při zpracování požadavku.

API požadavky

Všechny požadavky musí být prováděny s URL <http://dsweb.g6.cz/diplomka/> např.: <http://dsweb.g6.cz/diplomka/api/user-data.php>

URL	HTTP metoda	Parametry	Odpověď
/api/user-data.php	GET, POST	nick - (požadováno) přezdívka uživatele token - (požadováno po autentizaci) token s platností 5 minut password - (požadováno při autentizaci) Soukromý API klíč	Výsledkem je pole dat ve tvaru JSON. Data obsahují SMS uživatele, čekající na odeslání. JSON pole dat: { "data": [{ "id", "text", "phone", "sent" , "send_date", "id_user" }] } Při návratovém kódu 449 nebo 401 je nutné provést autentizaci pro získání tokenu, který používáme spolu s parametrem nick, dokud nevyprší platnost tokenu. Po autentizaci je vrácen JSON objekt { "token": token }.

URL	HTTP metoda	Parametry	Odpověď
/api/send-sms.php	GET, POST	<p>text - (požadováno) text zprávy</p> <p>phone - (požadováno) telefon příjemce</p> <p>send_date - (volitelné) čas odeslání ve tvaru: yyyy-MM-dd hh:mm:ss</p> <p>nick - (požadováno) přezdívka uživatele</p> <p>token - (požadováno po autentizaci) token s platností 5 minut</p> <p>password - (požadováno při autentizaci) Soukromý API klíč</p>	<p>Rozhraní zasílá data - SMS zprávy do hlavní databáze. Odtud jsou staženy klientskou stanicí - mobilní telefon se systémem Android a dále rozeslány přes GSM síť. Návrátové kódy dle tabulky návratových kódů.</p> <p>Po autentizaci je vrácen JSON objekt {"token":token}.</p> <p>Při návratovém kódu 449 nebo 401 je nutné provést autentizaci pro získání tokenu, který používáme spolu s parametrem nick, dokud nevyprší platnost tokenu.</p>
/api/archive.php	GET, POST	<p>nick - (požadováno) přezdívka uživatele</p> <p>token - (požadováno po autentizaci) token s platností 5 minut</p> <p>password - (požadováno při autentizaci) Soukromý API klíč</p>	<p>Výsledkem je pole dat ve tvaru JSON. Data obsahují archivované (odeslané) SMS uživatele. JSON pole dat</p> <pre>{ "archive": [{ "id", "text", "phone", "changed", "send_date" }, ...] }</pre> <p>Při návratovém kódu 449 nebo 401 je nutné provést autentizaci pro získání tokenu, který používáme spolu s parametrem nick, dokud nevyprší platnost tokenu.</p>