

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

NÁVRH A REALIZACE SYSTÉMU JABBER2GO

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

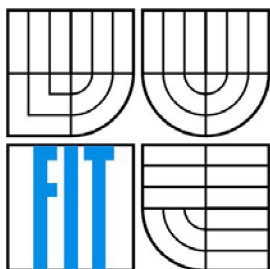
AUTOR PRÁCE
AUTHOR

RADOVAN DRLÍK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

NÁVRH A REALIZACE SYSTÉMU JABBER2GO

DESIGN AND IMPLEMENTATION OF JABBER2GO SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADOVAN DRLÍK

VEDOUČÍ PRÁCE
SUPERVISOR

ING. JAROSLAV ŠKARVADA

BRNO 2007

Zadání bakalářské práce

Řešitel: **Drlík Radovan**
Obor: Informační technologie
Téma: **Návrh a realizace systému Jabber2Go**
Kategorie: Web

Pokyny:

1. Seznamte se s jazyky PHP, XML, XHTML.
2. Nastudujte IM protokol Jabber.
3. Navrhněte WWW Jabber klienta, který umožní pohodlně realizovat veškeré IM funkce přes WWW nejlépe bez použití Java appletů a ActiveX.
4. Navržený systém implementujte.

Literatura:

- Dostupná na URL: <http://www.w3.org>
- Dostupná na URL: <http://www.php.net>
- Dostupná na URL: <http://www.jabber.org>
- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

1. Seznamte se s jazyky PHP, XML, XHTML.
2. Nastudujte IM protokol Jabber.
3. Navrhněte WWW Jabber klienta, který umožní pohodlně realizovat veškeré IM funkce přes WWW nejlépe bez použití Java appletů a ActiveX.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Škarvada Jaroslav, Ing.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 00 Brno, Božetěchova 2



doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Radovan Drlík**
Id studenta: 84422
Bytem: Piskov 76, 783 83 Troubelice
Narozen: 18. 03. 1985, Šternberk
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Návrh a realizace systému Jabber2Go
Vedoucí/školitel VŠKP: Škarvada Jaroslav, Ing.
Ústav: Ústav počítačových systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracování díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

Radovan Dvůřák

.....
Autor

Abstrakt

Pro současnou společnost je stále důležitější být v neustálém kontaktu, proto je také instant messaging stále používanější. Cílem práce je navrhnout a realizovat instant messaging klienta pro komunikaci prostřednictvím sítě Jabber, který by kladl minimální nároky na vybavení uživatele. Práce pojednává o obecných vlastnostech instant messagingu a zaměřuje se na Jabber a Jabber protokol. Hlavní část práce představuje analýza existujících Jabber klientů a návrh a realizace nového klienta, který bude komunikovat s uživatelem prostřednictvím webového prohlížeče.

Klíčová slova

Jabber, Jabber protokol, programování Jabberu, elektronická komunikace, instant messaging, systémy pro komunikaci v reálném čase, XML, XHTML, PHP, CSS

Abstract

For contemporary society it is ever more important to be in permanent contact therefore instant messaging is constantly more used. Object of this thesis is to design and to realize instant messaging client for communication over Jabber system. Client demands of minimal equipment requirements for user of the system. Thesis denotes about general characteristics of instant messaging and more detailed definition of Jabber and Jabber protocol. Thesis gives also attention to analysis of existing Jabber clients, and design and realisation of newly-made client which communicates with user via web browser.

Keywords

Jabber, Jabber protocol, Jabber programming, electronic communication, instant messaging, real-time communication systems, XML, XHTML, PHP, CSS

Citace

Radovan Drlík: Návrh a realizace systému Jabber2Go, bakalářská práce, Brno, FIT VUT v Brně, 2007

Návrh a realizace systému Jabber2Go

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Škarvady.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Radovan Drlík
1. 5. 2007

Poděkování

Děkuji vedoucímu práce panu Ing. Jaroslavovi Škarvadovi za odborné vedení, poskytnuté konzultace a podnětné náměty.

© Radovan Drlík, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah	1
Úvod	3
1 Instant messaging (IM)	4
1.1 Cíle IM	4
1.2 Základní funkce	4
1.3 Existující služby	5
2 Jabber a Jabber protokol	7
2.1 Popis základních funkcí	7
2.1.1 Zprávy	7
2.1.2 Seznam kontaktů – roster	8
2.1.3 Dostupnost kontaktu	8
2.1.4 Správa oprávnění – autorizace	9
2.2 Jabber protokol	9
2.2.1 Způsob komunikace mezi klienty	9
2.2.2 Rozlišování uživatelů Jabberu	9
2.2.3 Přihlášení k Jabber serveru	10
2.2.4 Odesílání a přijímání zpráv	10
2.2.5 Distribuce stavu uživatelů	10
2.2.6 Autorizace kontaktů	11
3 Použité technologie	12
3.1 XML	12
3.2 XHTML	12
3.3 CSS	13
3.4 PHP	13
4 Analýza a návrh systému	15
4.1 Existující klienti systému Jabber	15
4.2 Klient Jabber2Go	16
4.3 Daemon	17
4.3.1 Spuštění a běh daemona	18
4.3.2 Obsluha uživatelů systému	18
4.4 Webový klient	19
4.4.1 Seznam kontaktů	19
4.4.2 Správa kontaktů	19
4.4.3 Okno komunikace mezi uživatelem a kontaktem	19

4.4.4	Nastavení stavu uživatele systému.....	20
4.4.5	Nastavení vlastností webového klienta.....	20
4.5	Komunikace mezi daemonem a webovým klientem.....	21
4.5.1	Zabezpečení.....	22
5	Realizace systému Jabber2Go.....	23
5.1	Komunikace mezi daemonem a webovým klientem.....	23
5.2	Daemon.....	24
5.2.1	Spuštění daemona.....	25
5.2.2	Běh daemona.....	25
5.2.3	Obsluha uživatele.....	25
5.2.4	Signály k probuzení daemona.....	26
5.2.5	Spojení s Jabber serverem.....	26
5.3	Webový klient.....	27
5.3.1	Obnova stránky.....	27
5.3.2	Realizace webového klienta.....	27
5.3.3	Části webového klienta.....	28
5.4	Způsob realizace základních funkcí.....	30
5.4.1	Přihlášení uživatele.....	30
5.4.2	Nastavení stavu uživatele.....	30
5.4.3	Získání seznamu kontaktů.....	30
5.4.4	Stav kontaktů.....	31
5.4.5	Přidávání, odebrání a editace kontaktů.....	31
5.4.6	Autorizace kontaktů.....	31
5.4.7	Přijímání a odesílání zpráv.....	32
5.4.8	Odhlášení uživatele.....	32
5.5	Požadavky systému na software.....	33
5.6	Monitorování a správa systému.....	33
6	Závěr.....	34
	Literatura.....	35
	Seznam příloh.....	36
	Příloha 1: Příklady XML elementů Jabber protokolu.....	37
	Příloha 2: Konfigurace systému Jabber2Go.....	40
	Příloha 3: Přehled zdrojových souborů.....	41
	Příloha 4: Uživatelská příručka.....	42

Úvod

Ačkoliv je historie instant messagingu (IM) poměrně dlouhá, IM nabývá své důležitosti teprve v posledních letech, kdy se internet stává dostupným téměř pro všechny a lidé se díky výtoky moderní doby snaží být v neustálém kontaktu.

V současné době existuje obrovské množství služeb a klientů pro IM. V této práci se však zaměřím zejména na systém Jabber, který je celosvětově hojně používán a jehož specifikace je volně dostupná.

Systém Jabber2Go bude představovat Jabber klienta, který bude umožňovat komunikaci s ostatními uživateli Jabberu. Narozdíl od ostatních klientských programů však bude klást co nejmenší nároky na zařízení a programové vybavení uživatele. Ke správné funkci by mu měl postačit prohlížeč webových stránek a samozřejmě také připojení k internetu.

Cílem práce je tedy navrhnout a vytvořit klienta pro síť Jabber, který bude fungovat i v jednoduchých prohlížečích webových stránek.

V 1. kapitole bude zmíněna základní charakteristika IM, druhy nabízených služeb a stručné informace o nejvýznamnějších IM systémech.

Ve 2. kapitole se tato obecná část rozšíří popisem systému Jabber a Jabber protokolu. Budou zde naznačeny základní vlastnosti Jabberu a princip jeho fungování. Podrobnosti Jabber protokolu budou uvedeny v příloze.

Pro úspěšný návrh a realizaci systému Jabber2Go budou informace poskytnuté v obou dvou kapitolách nezbytné.

V kapitole 3 popíši základní vlastnosti technologií, které budou převážně využity při implementaci systému. Zaměřím se na popis jazyka XML, XHTML, PHP a CSS.

Ve 4. kapitole se budu zabývat návrhem systému Jabber2Go. Nejdříve budu věnovat pozornost existujícím Jabber klientům a poté se zaměřím na analýzu a návrh vlastního systému. V této části bude systém popsán obecnou formou, bez specifických implementačních detailů.

5. kapitola bude navazovat na kapitolu 4 a zaměřím se v ní na vlastní realizaci systému. Bude zde naznačen způsob implementace systému a řešení problémů, které v průběhu práce vznikly.

1 Instant messaging (IM)

V této kapitole Vás seznámím s významem pojmu instant messaging (IM), s možnostmi využití, s druhy IM služeb a četností použití různých druhů těchto služeb. Cílem kapitoly je podat stručný přehled vlastností, odstíněný od konkrétních IM služeb. Více o instant messagingu lze nalézt na [1].

Pojem instant messaging

Instant messaging je způsob komunikace mezi alespoň dvěma lidmi, která probíhá v reálném čase (nebo takřka v reálném čase). Zpravidla je založena na zasílání textových zpráv mezi uživateli využívajícími danou službu.

Princip fungování IM

Uživatel, který chce využívat funkcí IM, musí používat klientský program, jež je schopen komunikovat s ostatními klientskými programy nebo se připojit na server dané služby, který komunikaci zprostředkovává.

Komunikace mezi klienty a servery probíhá podle předem určených pravidel, která jsou definována v protokolu dané služby. Některé z těchto protokolů nejsou veřejné a jsou „tajemstvím“ vlastníci firmy, některé z nich jsou volně dostupné.

1.1 Cíle IM

Hlavním cílem IM je výměna zpráv mezi uživateli v reálném čase. Aby komunikace měla smysl, měla by probíhat prostřednictvím počítačové sítě, typicky prostřednictvím sítě Internet.

Výměnu zpráv umožňuje také elektronická pošta, nabízí tedy IM něco nového? Hlavním rozdílem oproti elektronické poště je možnost průběhu komunikace v reálném čase. Další výhodou může být také možnost zjištění dostupnosti uživatelů. Lze se dozvědět, jestli je daný uživatel online a dostupný ke komunikaci, či momentálně nedostupný.

Kromě těchto základních funkcí nabízí různé instant messaging služby ještě další funkce, kterými mohou zpříjemnit práci svým uživatelům. K nejčastějším rozšířením patří například udržování seznamu kontaktů, povolení komunikace pouze s některými uživateli nebo také možnost odesílání souborů.

1.2 Základní funkce

Dostupné funkce se liší podle použité služby a klientského programu. Většinou je však umožněno přijímání a odesílání textových zpráv, udržování seznamu kontaktů a správu dostupnosti kontaktů.

Přijímání a odesílání zpráv

Hlavním cílem IM je umožnit zaslání zprávy osobě, která používá danou službu, a umožnit přijetí zprávy od této osoby.

Uživateli, který je aktuálně dostupný a je přihlášen ke službě, by zpráva měla dojít takřka okamžitě. Pro uživatele, který není momentálně dostupný, by měla být zpráva uchována pro pozdější vyzvednutí a měl by být také zaznamenán čas jejího přijetí.

Seznam kontaktů

Pro IM klienty je typické uchovávání seznamu kontaktů, se kterým může uživatel manipulovat (přidávat nové kontakty, upravovat a odebírat kontakty již existující). Tento seznam slouží jako jakýsi adresář „známých“ kontaktů, který je uživateli dostupný po přihlášení ke službě. Uživatel si tak nemusí pamatovat velkou spoustu adres či jiných identifikátorů svých kontaktů.

Seznam je také často udržován kvůli autorizaci daných kontaktů (tzn. určení oprávnění, zda kontakt může sledovat informace o uživateli služby, popřípadě mu zasílat zprávy).

Informace o dostupnosti kontaktu

Většina služeb umožňuje také zobrazení informace o dostupnosti kontaktu. Uživatel tak může zjistit, zda je daný kontakt připojen a je k dispozici pro komunikaci.

Většina moderních klientských programů a služeb umožňuje také nastavení rozšířených stavů, včetně informačního popisu stavu, který si uživatel může sám zvolit.

Další rozšíření

Různé služby a klienti nabízí další funkce, které mohou uživateli zjednodušit a zpříjemnit práci. Za nejčastější se dá považovat ukládání historie komunikace, odesílání souborů, zobrazení informace o tom, že kontakt právě píše zprávu a podobně.

1.3 Existující služby

V současné době existuje obrovské množství služeb, poskytujících IM funkce, proto zde zmíním jen ty nejznámější a nejpoužívanější.

Pro všechny služby však platí, že využívají předem definovaný způsob komunikace mezi klienty a servery. Tato pravidla jsou definována v protokolu dané služby (který však nemusí být vždy veřejně dostupný).

V historii probíhalo několik pokusů o sjednocení těchto protokolů a vytvoření protokolu, který by byl obecně platný pro všechny služby. Tyto pokusy selhaly, avšak bylo vytvořeno několik obecných protokolů, ze kterých stojí za zmínku XMPP (Extensible Messaging and Presence Protokol), který je ve formátu XML a je základem Jabber protokolu.

V současné době lze využít klientské aplikace, které dovedou používat více služeb současně. Možné je také použít IM server, který umožňuje komunikaci s jinými IM sítěmi. Těto službě se často říká transport.

V tabulce Tabulka 1-1 je znázorněn počet aktivních uživatelů vybraných služeb. Zdrojem těchto čísel je literatura [1]. Protože je problematika určení počtu aktivních uživatelů (tzn. uživatelů, kteří službu opravdu využívají) složitá, je třeba brát uvedená čísla s rezervou.

Služba	Počet aktivních uživatelů [milióny]
AIM	53
Jabber	40-50
Windows Live Messenger	27
Yahoo! Messenger	22
ICQ	4

Tabulka 1-1: Počet aktivních uživatelů

Jabber

Systém Jabber je velice používaný a oblíbený po celém světě. Jeho hlavní výhodou je otevřenost – veřejná dostupnost protokolu. Jabber protokol je založen na standardizovaném protokolu XMPP (Extensible Messaging and Presence Protokol).

Jelikož je pro tuto práci Jabber protokol nezbytný, budu se jím zabývat ještě v dalších kapitolách, zejména v kapitole 2 – Jabber a Jabber protokol.

AIM

AOL Instant Messenger (zkráceně AIM) je aplikace, která umožňuje registrovaným uživatelům zasílat zprávy a také přenášet hlas či video v reálném čase. Pro svou činnost využívá OSCAR protokolu (Open System for CommunicAtion in Realtime), který není bohužel veřejně dostupný.

Hlavní nevýhodou služby je její uzavřenost a svázanost s firmou, která vyvíjí klientskou aplikaci a určuje tak vývoj tohoto systému. Nevýhodou může být také centralizace služby.

ICQ

ICQ (z anglického „I seek you“) není zdaleka celosvětově nejrozšířenější IM službou, přesto je v České republice nejpoužívanější. Nynějším vlastníkem je firma AOL.

ICQ je velice podobné AIM, dokonce ke své činnosti využívá OSCAR protokolu. Z toho plynou stejné nevýhody jako v případě AIM.

2 Jabber a Jabber protokol

V této kapitole Vás blíže seznámím s Jabberem a Jabber protokolem. Pro celou práci, a především pro implementaci systému Jabber2Go, bude nezbytné pochopit fungování a způsob komunikace Jabberu, proto se zde této problematice budu věnovat podrobněji. Přesto bude následující popis obecného charakteru a jednotlivé XML elementy používané Jabber protokolem bude možné nalézt v příloze nebo na stránkách specifikace protokolu [2].

Co je to Jabber?

„Jabber není žádný konkrétní program ani žádný konkrétní server. Je to jen jazyk, který má sjednotit komunikátory a zbořit hranice mezi lidmi. Něco jako esperanto.“ [3]

Jabber je otevřený komunikační protokol a existuje tak nepřeberné množství klientských programů, kteří tento protokol podporují. Díky otevřenému standardu si každý může napsat vlastního klienta, či dokonce server.

Používání Jabberu

Abyste mohli komunikovat s dalšími uživateli Jabberu, je nezbytné se registrovat na některém ze serverů a získat tak svůj unikátní identifikátor, jakousi adresu, v síti Jabber. Existuje velké množství serverů, které Vám umožní registraci a používání účtu zdarma.

Některé Jabber servery umožňují také komunikovat s uživateli jiných IM systémů. To se děje prostřednictvím tzv. transportů. Výhodou je, že se pro klientskou aplikaci způsob komunikace neliší a až na straně serveru se zajistí správné odeslání do cílové IM sítě.

2.1 Popis základních funkcí

Mezi základní funkce Jabberu, jakožto systému pro IM, patří především odesílání zpráv ostatním uživatelům. Důležité je také udržování seznamu kontaktů (contact listu), který se označuje jako roster. Pro každý kontakt v rosteru je určeno oprávnění získávat informace o uživatelově dostupnosti (tzv. subscription). Oprávněné kontakty mají svolení odebírat tyto informace. Dostupnost uživatele se nazývá presence.

2.1.1 Zprávy

Asi nejdůležitější a nejpoužívanější částí protokolu je výměna zpráv mezi uživateli. Díky používání Unicode je možné používat diakritiku i znaky z jiných, pro nás exotických, abeced. Dokonce je podle protokolu možné jednu zprávu napsat ve více jazykových mutacích. Klient si následně vybere jazyk, který je mu nejbližší.

U každé zprávy je znám adresát a odesílatel, určený Jabber identifikátorem (JID), o kterém se ještě zmíním dále. Formát XML odesílaných a přijatých zpráv je naznačen v příloze.

Pokud je zpráva zaslána uživateli, který není momentálně dostupný, je na serveru uložena pro pozdější doručení. Do zprávy je pak přidán čas jejího doručení. Tento čas je ve formátu UTC (Universal Time Coordinated) [YYYY-MM-DDThh:mm:ss]. Pro střední Evropu je čas dán jako UTC +1:00 hodina v zimním čase a UTC +2:00 hodiny v čase letním.

2.1.2 Seznam kontaktů – roster

Pro zpříjemnění práce a ulehčení naší paměti Jabber také umožňuje spravovat seznam kontaktů, tedy ostatních uživatelů Jabberu, kteří jsou pro nás významní. Jedná se vlastně o jakýsi adresář, ve kterém je možné si ke každé adrese kontaktu přiřadit jméno (přezdívku) a skupinu, do které tento kontakt bude patřit. Seznam je ukládán na server a je tak dostupný ze všech uživatelských zdrojů.

Každá položka rosteru je identifikována podle unikátního JID kontaktu, které je udáno bez názvu zdroje.

Samozřejmě je umožněna kompletní údržba rosteru, jako je přidávání a odebrání kontaktů a úprava údajů o již existujících kontaktech. Při jakékoliv změně rosteru server zajistí odeslání změn na všechny dostupné zdroje, tzv. roster push. Roster je tak vždy na všech zdrojích v aktuální podobě.

Hlavním úkolem rosteru není pouze udržování jakéhosi adresáře, jak by se z dosavadního popisu mohlo zdát. Tento seznam kontaktů se používá také kvůli udržování oprávnění kontaktu k odebrání informací o dostupnosti.

Formát XML pro získání, přidávání a odebrání kontaktu rosteru je uveden v příloze.

2.1.3 Dostupnost kontaktu

Většina IM služeb nabízí distribuci informací o dostupnosti uživatele. V případě Jabberu se jedná o síťovou dostupnost uživatele, která může být v případě dostupného uživatele upřesněna dalšími podstavami.

Volitelně může být také zadán popis stavu a vždy je zadána jeho priorita. Priorita je určena číslem v rozmezí od -128 do 127. Priorita s vyšším číslem je považována za „důležitější“.

Stavové informace (stav, priorita, popis) se vždy vážou ke zdroji (jednotlivá přihlášení uživatele – 2.2.2). Každý zdroj má svůj vlastní stav (to znamená, že může být pro každý zdroj jiný). Pokud nspecifikujeme zdroj kontaktu, je jeho stav určen zdrojem s nejvyšší prioritou. Existuje-li více zdrojů se stejnou prioritou, bere se stav zdroje, od kterého byla přijata stavová zpráva jako poslední.

Po vytvoření spojení by měl klient nastavit úvodní stav a při odhlášení stav „unavailable“, tedy informaci o nedostupnosti.

Formát XML pro nastavení stavové informace je uveden v příloze.

2.1.4 Správa oprávnění – autorizace

Aby bylo ochráněno soukromí uživatelů, mohou sledovat informace o dostupnosti pouze oprávnění uživatelé. Toto oprávnění se označuje termínem *subscription*. Lze jej získat pouze zasláním požadavku kontaktu, který ho následně musí povolit. Přijatému požadavku musí klient buď vyhovět, a nebo jej odmítnout.

Dříve autorizovanému kontaktu lze kdykoliv odebrat jeho oprávnění nebo je také možné se odhlásit od odebrání stavových informací o kontaktu.

Formát XML pro požadavek o autorizaci a její přijetí či odmítnutí je uveden v příloze.

2.2 Jabber protokol

Komunikační protokol Jabberu je založen na obecném značkovacím jazyce XML (eXtensible Markup Language). Jabber protokol vychází ze standardu XMPP (eXtensible Messaging and Presence Protocol), který definuje způsob rozesílání XML elementů za účelem výměny zpráv a informací o dostupnosti uživatelů či služeb. Standard XMPP můžete nalézt v RFC 3920 a RFC 3921 [2].

2.2.1 Způsob komunikace mezi klienty

Klienti mezi sebou nekomunikují přímo, ale využívají služeb Jabber serveru, který zajišťuje komunikaci mezi jednotlivými klienty i servery.

Jabber je narozdíl od většiny konkurenčních instant messaging služeb decentralizovaný. Přitom uživatelé spolu mohou bez problému komunikovat, přestože je každý připojen k jinému Jabber serveru.

Dokonce si každý může zřídit vlastní server, a to buď pouze pro komunikaci na lokální síti, nebo i pro komunikaci s ostatními uživateli Jabberu prostřednictvím internetu.

2.2.2 Rozlišování uživatelů Jabberu

Každý uživatel má přiřazeno jedinečné uživatelské jméno, kterému se říká Jabber ID (JID). Toto jméno je velice podobné e-mailové adrese.

JID uživatele

Skládá ze jména či přezdívky uživatele, doménového jména a jména zdroje. Formát JID je tedy: `jmeno@domena/zdroj`.

V jeden okamžik můžete být přihlášení ke svému Jabber účtu vícekrát. Aby se dalo například rozlišovat, odkud je zpráva odesílána nebo kam je směřována, přidává se za standardní adresu ještě název zdroje (anglicky *resource*), který slouží vlastně jako rozšířený popis adresy. Pro každý zdroj je udržována jeho priorita a jeho dostupnost.

Důležité je, že názvy nejsou předem určeny a je možné použít takřka libovolný název. Dokonce je možné díky používání Unicode používat diakritiku. Nejčastěji se používá název zdroje reprezentující umístění: „práce“, „doma“ nebo používaného klienta: „Psi“, „Miranda“ a podobně.

2.2.3 Přihlášení k Jabber serveru

Přihlášení probíhá tak, že klientská aplikace vytvoří spojení na Jabber server a následně sérií XML elementů vyjedná přihlášení.

V první fázi je nezbytné, aby klient určil použitou verzi XML a používanou znakovou sadu. Server zašle klientovi odpověď, ve které jsou vyjmenovány dostupné metody pro přihlášení a identifikátor, který může být použit pro zašifrování hesla, aby po síti neprocházelo v otevřené podobě.

Klientský program zašle na server jméno uživatele, název zdroje a heslo, které nemusí být v čitelné podobě, ale musí být správně zakódováno.

Pokud proběhne vše v pořádku – uživatelské jméno na daném serveru existuje a je zadáno také správné heslo, server zašle kladnou odpověď a komunikace mezi klientem a serverem může pokračovat. Pokud při přihlašování nastane problém, odpoví server chybovou zprávou a komunikace je ukončena.

V jeden okamžik se není možné vícekrát přihlásit s totožným názvem zdroje, přičemž se rozlišuje velikost písmen.

2.2.4 Odesílání a přijímání zpráv

Zpráva se neodesílá přímo kontaktu, ale prostřednictvím uživatele na Jabber serveru. Zpráva je odeslána prostřednictvím elementu `message`, jehož atributem je JID adresáta a obsahem je text zprávy.

Pro odesílání zpráv není nutné zadávat celé JID adresáta (tzn. včetně zdroje). Pokud název zdroje není zadán, zpráva se doručí na zdroj, který má nastaven nejvyšší prioritu. V případě, že jsou všechny zdroje adresáta nedostupné, uloží se zpráva na serveru a je mu při příštím připojení doručena.

Příjem zpráv probíhá obdobným způsobem. Zpráva je doručena z uživatele na Jabber serveru prostřednictvím elementu `message`, ve kterém je jedním z atributů odesílatel zprávy.

2.2.5 Distribuce stavu uživatelů

Nastavení stavu uživatele se děje prostřednictvím elementu `presence`, jehož obsahem je název stavu. Klientská aplikace zasláním elementu `presence` na Jabber server nastaví stav uživatele. Vlastní distribuci stavu zajišťují Jabber servery, které oprávněným a dostupným uživatelům rozesílají stavové zprávy.

Různé typy dostupnosti uživatele a jejich význam je naznačen v tabulce Tabulka 2-1.

Dostupnost	Popis	Význam
unavailable		Nedostupný
online		Dostupný, bez bližší specifikace
chat		Volný ke komunikaci
away		Dočasně vzdálený
xa	eXtended Away	Vzdálený po delší dobu
dnd	Do Not Disturb	Nevyrušovat

Tabulka 2-1: Typy dostupnosti

2.2.6 Autorizace kontaktů

Sledovat stav uživatele mají pouze oprávněné kontakty. Oprávnění odebírat stavové informace o kontaktu se nazývá také *subscription*. Oprávnění je možné získat zasláním elementu `presence`, jehož atribut `type` je nastaven na hodnotu `subscribe`. Kontakt, který je požádán o autorizaci, by ji měl následně vyhovět či ji zamítnout.

Typy požadavků elementu `presence` jsou vyjmenovány v tabulce Tabulka 2-2.

Typ požadavku	Význam
subscribe	Požadavek o autorizaci
subscribed	Vyhovění autorizaci – povolení kontaktu odebírat stavové informace
unsubscribe	Kontakt zrušil odebíráni stavových informací o uživateli
unsubscribed	Zamítnutí autorizace – zakázání kontaktu odebírat stavové informace

Tabulka 2-2: Typy požadavků o autorizaci

Přehled typů oprávnění a jejich význam je znázorněn v tabulce Tabulka 2-3.

Oprávnění – subscription	Význam
none	Ani uživatel, ani kontakt není oprávněn odebírat informace o stavu
to	Uživatel má povoleno odebírat informace o stavu kontaktu, ale kontakt o uživateli ne
from	Kontakt může přijímat informace o přítomnosti uživatele, ale uživatel o kontaktu ne
both	Oba navzájem mohou přijímat informace o stavu

Tabulka 2-3: Typy oprávnění

3 Použité technologie

V této kapitole budou popsány stěžejní technologie, použité při implementaci systému Jabber2Go. Cílem není podrobný popis těchto technologií, ale základní informace o nich.

3.1 XML

XML (eXtensible Markup Language), v překladu tedy rozšiřitelný značkovací jazyk, je obecný značkovací jazyk pro vytváření dokumentů reprezentujících strukturovaná data. Byl standardizován World Wide Web Consortiumem (W3C), které dohlíží na standardy pro World Wide Web (WWW). V současné době je jazyk XML hojně používán a stále nabývá na své důležitosti. Bohužel není možné se v této práci podrobněji věnovat problematice XML, více informací je možné nalézt v literatuře [4] nebo na oficiálních webových stránkách organizace W3C [5].

Hlavní výhodou jazyka XML je nutnost dodržování definovaných pravidel. Například celý dokument musí být uzavřen v jednom, kořenovém, elementu. Neprázdné elementy musejí být omezeny počáteční a koncovou značkou. Prázdné elementy musejí být speciálně ukončeny. Všechny hodnoty atributů musejí být uzavřeny v uvozovkách (pravidlo platí i pro číselné hodnoty). Dodržováním těchto pravidel dochází k zjednodušení analýzy dokumentu.

Další výhodou tohoto jazyka je to, že předem nedefinuje žádné značky. Může se tak používat nejen pro reprezentaci struktury dokumentu, ale je vhodný i pro výměnu dat mezi aplikacemi.

Systém Jabber využívá zmiňovaných výhod a pro komunikaci mezi svými klienty a servery používá právě XML. Jabber protokolem jsem se zabýval ve 2. kapitole a podrobné schéma bude naznačeno v příloze.

3.2 XHTML

XHTML (eXtensible HyperText Markup Language) je aplikací jazyka XML. Jedná se o značkovací jazyk určený pro tvorbu webových stránek. O jeho specifikaci a udržování se také stará World Wide Web Consortium. Podrobnosti o jazyce XHTML lze nalézt v literatuře [6] a nebo na stránkách organizace W3C [7].

Jazyk XHTML je velice podobný svému předchůdci HTML (HyperText Markup Language). Hlavním rozdílem je dodržování pravidel XML – názvy značek je nutné uvádět malými písmeny, hodnoty atributů musí být uzavřeny v uvozovkách a podobně.

Současnou snahou je odstranit značky, které mají spíše formátovací a ne popisný charakter (příkladem může být značka *b* nebo *i*). Cílem je dát dokumentu logickou strukturu a vlastní vzhled definovat odděleně, například použitím kaskádových stylů (CSS).

Formát XHTML je tedy přísnější a přesnější. Díky tomu by mohly být prohlížeče rychlejší a jednodušší. Tento formát, a obecně formát XML, je také vhodnější pro automatizované zpracování a měl by tak umožnit jednodušší práci vyhledávačů.

Celé uživatelské rozhraní systému Jabber2Go bude realizováno v jazyce XHTML. Výsledné webové stránky, které uživatel obdrží, budou generovány na straně serveru prostřednictvím jazyka PHP. Cílem je vytvořit všechny stránky validní, to znamená platné podle standardu W3C.

3.3 CSS

CSS (Cascading Style Sheets), tedy kaskádové styly, slouží ke specifikaci vzhledu stránek. Cílem je oddělit strukturu dokumentu od jeho vzhledu. Při práci byla použita literatura [8], která sloužila především jako referenční příručka vlastností kaskádových stylů.

Díky použití kaskádových stylů není třeba předělávat zdrojový kód XHTML (popř. HTML) při jakékoliv změně vzhledu. Další výhodou je také možnost jednotného stylu pro více stran nebo různého stylu pro zobrazení na monitoru, mobilních zařízeních nebo při tisku. Používáním kaskádových stylů dochází často ke zrychlení načítání webových stránek. Styl definovaný v externím souboru se může totiž uložit do vyrovnávací paměti prohlížeče a při opětovném načtení strany se tak již nemusí stahovat z internetu.

Nevýhodou může být horší dostupnost v prohlížečích, zejména u starých prohlížečů nebo také u některých prohlížečů na mobilních zařízeních. Na obsah dokumentu tato skutečnost však nemá žádný vliv a dokument je v prohlížeči zobrazen, přestože jeho vzhled může být dost odlišný.

Vzhledu dokumentů systému Jabber2Go bude definován pomocí kaskádových stylů. Dokumenty však musí být přehledné a srozumitelné i bez použití těchto stylů (zejména kvůli dostupnosti na prohlížečích mobilních telefonů). Struktura XHTML dokumentu musí být tedy ve správném logickém uspořádání a není vhodné používat plovoucí objekty nebo absolutní pozicování, které CSS nabízí.

3.4 PHP

PHP (Hypertext Preprocessor, původně Personal Home Page) je skriptovací jazyk, který se nejčastěji využívá pro vytváření dynamických webových stránek. Je to však obecný skriptovací programovací jazyk, který se může používat nejen pro tvorbu webových aplikací, ale i pro běžné programování – skriptování z příkazové řádky. Protože je jazyk PHP velmi rozsáhlý a nabízí obrovské množství funkcí, zmíním se pouze o jeho obecných vlastnostech, podrobnosti o tomto jazyce je možné nalézt v literatuře [9] a nebo na webových stránkách [10].

PHP je také nezávislý na platformě. Obsahuje obrovské množství knihoven funkcí, za nejdůležitější považují funkce pro práci s textem, regulárními výrazy a jednoduchý přístup k databázovým serverům.

PHP skripty jsou prováděny na straně serveru. Výhodou může být omezení přenášených dat, jelikož se k uživateli dostane pouze výsledná webová stránka. Pro uživatele také není dostupný zdrojový kód skriptu a je tak znesnadněno jeho odcizení.

Nevýhodou použití PHP ve webových stránkách je nemožnost interakce s uživatelem. Není možné zjistit, zda se uživatel pohybuje nad určitou částí stránky, posunuje se v textu a podobně. Při jakékoliv definované akci uživatele se musí zaslat požadavek na server a dojde k odeslání celé webové stránky (i když by se změnila pouze část výsledku). Tyto negativní vlastnosti se dají eliminovat použitím např. Javascriptu, který se provádí na straně klienta. Popsaný problém není způsoben přímo použitím jazyka PHP, ale principem fungování protokolu HTTP (HyperText Transfer Protocol).

Díky jednoduché integraci PHP skriptu do XHTML kódu a rovněž díky obrovskému množství nabízených funkcí (zejména přístup k mySQL databázi, analýza XML, spojení pomocí socketů) bude hlavní jádro systému Jabber2Go implementováno v jazyce PHP.

4 Analýza a návrh systému

Nejdříve se v této kapitole budu zabývat existujícími systémy Jabber a jejich vlastnostmi. Poté přejdu k návrhu vlastního systému Jabber2Go. Budu se snažit o obecný popis, odstíněný od implementačních detailů, kterým se budu věnovat v kapitole 5.

Protože má být výsledný systém dostupný i pro přenosná zařízení a mobilní telefony, které disponují webovým prohlížečem, je třeba rozdělit systém a dekomponovat problémy na části, které je možné řešit ve webovém prohlížeči uživatele, a na části, které je nezbytné řešit mimo něj.

Kvůli dostupnosti systému na jednoduchých prohlížečích není vhodné použít žádná rozšíření, jako jsou například Java applety nebo ActiveX prvky. Uživateli bude tedy předána webová stránka v čistém XHTML, kterou by měl být schopen zobrazit každý prohlížeč.

4.1 Existující klienti systému Jabber

Jak jsem se již dříve zmínil, díky otevřenému protokolu Jabberu si každý může naprogramovat svého vlastního klienta či dokonce server. Existuje tedy obrovské množství klientů, ze kterých je možné si vybrat. Velká část z nich je dostupná dokonce zdarma.

Klienti ve formě spustitelného programu

Proč tedy navrhnout něco, čeho je dostatek? Problémem je, že většina klientů, o kterých mluvím, je ve formě binárního souboru, tedy spustitelného programu. Velmi často se takovýto program musí nainstalovat a k tomu nemusíte mít oprávnění nebo se to zkrátka při použití na cizím počítači nevyplatí.

Výhodou těchto klientů je nejčastěji rychlost odezvy, vzhled a také množství doplňkových funkcí, které nabízí.

Pokud používáte svůj počítač a máte dostatečná oprávnění na instalaci nebo spouštění této formy klienta, bude pro Vás tato forma klientského programu asi nejlepší volbou.

Klienti využívající webový prohlížeč

Výhodní a použitelní se zdají klienti, kteří se dají spustit z prohlížeče webových stránek. Skutečně je tomu tak, většinou s nimi vystačíte i na cizím počítači, protože se často nemusí nic instalovat.

Tito klienti jsou ale obvykle založeni na použití nějaké rozšiřující technologie, často Java appletu. Jedná se vlastně o program, který si prohlížeč stáhne a poté se automaticky spustí. Nevýhodou je, že Java applety vyžadují použití Java Virtual Machine (JVM), která nemusí být vždy dostupná.

Méně častí jsou pak klienti, kteří pro svůj běh vyžadují pouze webový prohlížeč se zapnutou podporou JavaScriptu. Tito klienti již existují ve většině prohlížečů, avšak JavaScript může být vypnut nebo jej některé prohlížeče, zejména mobilních telefonů, nemusí podporovat.

Cílem systému Jabber2Go je fungování v prohlížečích, které podporují zobrazení pouze HTML (XHTML) stránek, bez nutnosti dalších rozšíření.

4.2 Klient Jabber2Go

Klienti, kteří nevyžadují žádné rozšiřující technologie, prakticky neexistují z toho důvodu, že se veškerá zátěž musí přesunout od uživatele někam jinam.

Důvodem je především nemožnost vytvoření spojení na cílový Jabber server a poměrně složitá analýza XML elementů, pomocí kterých probíhá komunikace. Musí tak existovat nějaký mezičlánek v komunikaci mezi webovým klientem a Jabber serverem. Tímto mezičlánkem bude v systému Jabber2Go daemon, který poběží na některém serveru a zajistí spojení a komunikaci s Jabber serverem. Způsob komunikace je naznačen na obrázku Obrázek 4-1.



Obrázek 4-1: Komunikace mezi Jabber serverem a klientem

Problémy v důsledku použití HTTP protokolu

HTTP protokol (HyperText Transfer Protocol) je založen na principu dotaz-odpověď. Komunikaci tedy vždy zahajuje klient, webový prohlížeč, který určí, kterou stránku po serveru požaduje.

Důsledkem tohoto chování je nemožnost informování uživatele ze strany serveru v případě přijetí nové zprávy, příchodu kontaktu a podobně. Situaci lze řešit periodickým obnovováním stránky, jedná se o takzvaný refresh rámce.

Dalším problémem je, že není možné běžným způsobem zjistit, zda si uživatel stránku stále prohlíží nebo ji již opustil. Protože dochází k přihlašování uživatele ke službě Jabber a udržuje se informace o jeho dostupnosti, musí dojít po určité době nečinnosti k automatickému odhlášení uživatele od systému Jabber2Go a také od příslušného Jabber serveru.

Problémy s bezstavovostí HTTP protokolu lze zmírnit použitím PHP sessions, do kterých se mohou uložit důležité informace o uživateli. Vhodné bude například ukládání doby obnovy strany, která by měla s dobou nečinnosti uživatele progresivně růst.

Omezení pro použití na PDA a mobilních telefonech

Na těchto zařízeních se používají prohlížeče, které obecně nemusí podporovat některá rozšíření, jako jsou Java applety, ActiveX prvky, Flash animace a podobně. Někdy není dostupný ani JavaScript a nebo dokonce některé části HTML či XHTML. Obvykle se jedná o rámy, které by na malém displeji neměly ani význam. Zabezpečené připojení pomocí SSL nemusí na těchto prohlížečích fungovat. V některých starších nebo jednodušších prohlížečích nemusí být plně podporovány kaskádové styly (CSS), přesto by však měla být zachována logická struktura a přehlednost systému.

S těmito problémy se musí systém vyrovnat a musí fungovat i bez popsaných prvků.

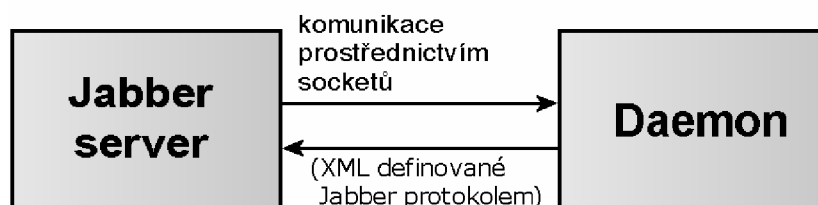
4.3 Daemon

Stěžejní částí systému bude daemon, o kterém by se dalo říci, že tvoří jádro celého klienta.

Daemon poběží na straně serveru a bude zprostředkovávat komunikaci s Jabber serverem. Bude implementován ve skriptovacím jazyce PHP, který poskytuje systému důležité funkce pro používání socketů, analýzu XML, práci s databázemi, textem a regulárními výrazy.

Komunikace s Jabber serverem bude probíhat prostřednictvím socketů, což je znázorněno na obrázku Obrázek 4-2. Pro vytvoření socketu a manipulaci s ním je možné použít PHP funkce. Celá komunikace s Jabber serverem probíhá podle Jabber protokolu, o kterém jsem se již zmínil ve 2. kapitole a příklad této komunikace bude naznačen v příloze.

Protože je podle Jabber protokolu komunikace řízena pomocí XML elementů, bude zapotřebí pro odeslání tyto elementy vytvářet a přijaté elementy analyzovat a dále zpracovávat.



Obrázek 4-2: Komunikace daemona s Jabber serverem

Hlavním úkolem daemona je transformovat požadavky od uživatele na příslušné XML elementy, odpovídající Jabber protokolu. Získané elementy musí zase transformovat do podoby, které rozumí webový klient. Daemon a webový klient tak vlastně představují dvě oddělené části, které spolu budou komunikovat prostřednictvím definovaných pravidel.

Daemon bude zprostředkovávat přihlášení uživatele k Jabber serveru, nastavení stavu a stavové hlášky uživatele, přijímání a odesílání zpráv, správu kontaktů a jejich autorizaci.

Existuje více možností přístupu k fungování daemona, přičemž každá z nich má své výhody a nevýhody. Snahou systému Jabber2Go bude umožnění zvolit administrátorovi systému jeden z přístupů pomocí konfiguračního souboru.

4.3.1 Spuštění a běh daemona

Daemon bude buď neustále běžet na pozadí a nebo dojde k jeho spuštění až v případě potřeby. U tohoto problému nelze jednoznačně určit, co je výhodnější. Rozhodnutí závisí především na počtu uživatelů, kteří využívají systém. Pokud by byl téměř neustále někdo přihlášen, je výhodnější nechat daemona běžet nepřetržitě. V opačném případě je vhodné daemona spustit až při požadavku na připojení a ukončit ho v okamžiku odhlášení posledního uživatele systému.

Pokud daemon neběží neustále, je ho nutné při přihlášení prvního uživatele systému spustit. Spuštění je možné realizovat voláním skriptu, který jej zajistí. Po odhlášení posledního uživatele systému dochází k ukončení daemona. V případě neustálého běhu nedochází ani ke spouštění, ani k ukončování daemona.

Administrátor by měl mít možnost zvolit jeden ze způsobů běhu, a proto by výsledný systém měl podporovat oba výše zmiňované způsoby. Výběr jedné možnosti se bude provádět příslušným nastavením konfiguračního souboru.

Problém spuštění daemona není triviální, jelikož požadavky o přihlášení mohou přicházet v náhodnou dobu a může se tak stát, že vzniknou dva požadavky o přihlášení zároveň. Skripty pro spuštění daemona by v tomto případě zjistily, že žádný daemon ještě neběží a spustily by jej. Vznikly by tak dva hlavní daemoni, kteří by se navzájem předbíhali v obsluze uživatelů.

Problém vícenásobného spuštění hlavního daemona je možné vyřešit použitím zámku souboru pro zajištění atomicity operace spuštění daemona.

4.3.2 Obsluha uživatelů systému

Daemon může buď obsluhovat všechny přihlášené uživatele, a nebo se může vytvořit proces pro každého uživatele zvlášť. Výhodnější je vytvořit proces pro každého uživatele. Při případných problémech při obsluze uživatele nemusí totiž dojít k ovlivnění ostatních procesů. Vytvořit nový proces ale nemusí být vždy možné – buď z důvodu omezení na serveru, a nebo nedostupností PHP funkce pro rozvětvení procesu (známé jako `fork`).

Vždy bude existovat hlavní daemon, který bude periodicky kontrolovat existenci požadavků na přihlášení uživatelů a při jejich existenci zajistí přihlášení uživatele do systému. Pokud není možné nebo je zakázáno vytváření nových procesů pro přihlášené uživatele, bude zajišťovat tento hlavní daemon také obsluhu všech přihlášených uživatelů. V opačném případě se stará pouze o vytváření nových procesů přihlašujících se uživatelů, které jej budou obsluhovat.

Pokud se vytváří nový proces pro každého uživatele, dojde při odhlášení uživatele také k ukončení tohoto procesu.

4.4 Webový klient

Funkcí webového klienta bude zprostředkování uživatelského rozhraní systému Jabber2Go. Rozhraní bude implementováno pomocí jazyka PHP. Výsledné webové stránky, které jsou odesílané uživateli, budou v jazyce XHTML.

4.4.1 Seznam kontaktů

Velice důležitou částí webového klienta bude zobrazení seznamu kontaktů (rosteru) a jejich stavu. Seznam by měl být podobný jako ve většině klientů pro IM, tedy měl by být zobrazen stav kontaktu, znázorněný obrázkovou ikonou, a jméno kontaktu. Pokud existují nově přijaté zprávy od kontaktu, musí být tato skutečnost zřejmá, opět je možné použít ikonu.

Jabber umožňuje přiřadit každému kontaktu také skupinu, do které je zařazen. Bylo by tedy vhodné rozdělovat kontakty do skupin a umožnit skrytí/zobrazení všech kontaktů ve skupině. Další vhodnou vlastností je možnost přepínání zobrazení všech nebo jen dostupných kontaktů.

Seznam kontaktů by po přihlášení měl tvořit úvodní stranu, na které by měl být zobrazen aktuální stav uživatele a také nabídka možných akcí.

4.4.2 Správa kontaktů

Systém by měl uživateli poskytnout možnost manipulovat se seznamem kontaktů (rosterem). Měl by umožnit přidávat a odebírat kontakty a upravovat informace o existujících kontaktech. Základními informacemi o kontaktu jsou: JID kontaktu, jméno (přezdívk) kontaktu a skupina, do které patří.

Nedílnou součástí správy kontaktů je autorizace kontaktu. Uživatel musí mít možnost požádat kontakt o autorizaci a přidávat a odebírat autorizace. V případě příchozího požadavku o autorizaci by měl mít možnost se rozhodnout, zda tento požadavek přijme či zamítne.

4.4.3 Okno komunikace mezi uživatelem a kontaktem

Nezbytnou částí webového klienta je také zobrazování přijatých zpráv a umožnění odesílání zpráv. Bylo by velice nepřehledné zobrazovat všechny přijaté a odesílané zprávy na jednu stránku, proto se po kliknutí na jméno kontaktu přepne do okna komunikace pouze s tímto kontaktem.

Okno komunikace by bylo vhodné vytvořit pomocí rámu. A to především z toho důvodu, že se musí neustále obnovovat část s přijatými a odeslanými zprávami. Pokud by se obnovovalo celé okno, včetně formuláře pro odeslání zprávy, docházelo by při obnově ke smazání již napsaného, avšak neodeslaného, textu.

Jiným řešením, než je použití rámu, by mohlo být použití JavaScriptu, kdy by se před obnovením stránky uložil obsah textového pole do cookie a při znovunačtení stránky by se vložil text

z cookie. Toto řešení má však několik nedostatků, z nichž největší je nutnost podpory a povolení JavaScriptu v prohlížeči uživatele.

Hlavní okno komunikace se bude tedy skládat z rámu pro menu zobrazujícího popis stavu zvoleného kontaktu, které bude umožňovat specifikaci akce přidružené ke kontaktu (výběr zdroje, zobrazení historie zpráv, smazání zpráv ...). Dále se bude skládat z okna, ve kterém bude zobrazen průběh komunikace mezi uživatelem a kontaktem. Ve spodním rámu bude formulář pro odeslání zprávy vybranému kontaktu. Vzhled hlavního okna komunikace je naznačen na obrázku Obrázek 4-3.



Obrázek 4-3: Rozložení rámu okna komunikace

Protože však nemusí být rámy všude dostupné, především ne na mobilních zařízeních, je třeba s tím počítat a v případě nutnosti umožnit náhradní řešení. Náhradním řešením může být zobrazení průběhu komunikace po výběru kontaktu a umístění zjednodušené nabídky v horní části strany. V této nabídce by se měla nacházet alespoň volba zpět na roster, odkaz na menu a odkaz na formulář pro zaslání zprávy.

4.4.4 Nastavení stavu uživatele systému

Uživatel by měl mít možnost nastavit všechny možné informace o své dostupnosti. Systém by měl umožnit nastavit i podstavy (možné stavy jsou popsány v kapitole 2.1.3, tabulce Tabulka 2-1). Nastavení priority stavu a jeho slovní popis by mělo být také umožněno.

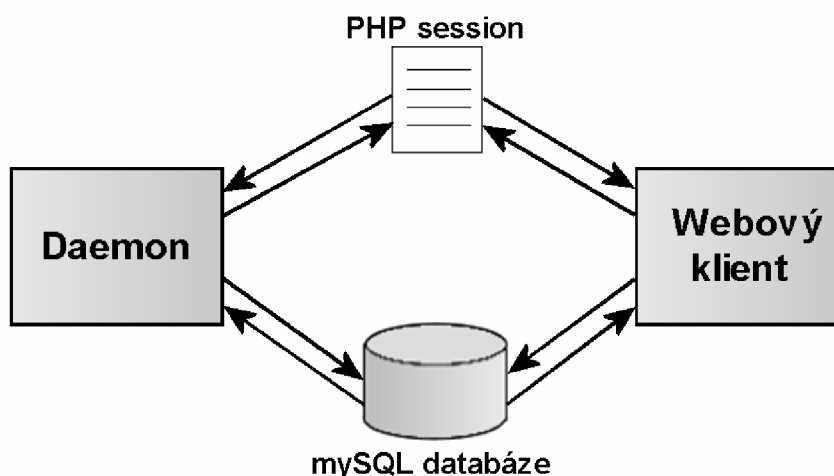
4.4.5 Nastavení vlastností webového klienta

Uživateli systému Jabber2Go by mělo být umožněno přizpůsobit systém svým představám a potřebám. Jako užitečné se zdá umožnit nastavení obnovovací doby stránek. Častou obnovou uživatel získá aktuálnější přehled o dostupnosti kontaktů a přijatých zpráv. Dochází však k častému přenosu dat a celkově tak ke většímu objemu přenesených dat. Mělo by být v rukou uživatele systému, zda si přeje aktuálnější informace nebo raději menší množství celkově přenesených dat.

Uživatel by měl mít rovněž možnost povolit či zakázat používání rámů a zabezpečeného připojení přes SSL. Oba dva prvky budou implicitně povoleny a jejich zakázání se doporučuje jen v opodstatněných důvodech, zejména při nemožnosti použití těchto prvků.

4.5 Komunikace mezi daemonem a webovým klientem

Protože daemon a webový klient fungují odděleně, je potřeba definovat způsob komunikace mezi nimi. Díky tomu, že budou obě části implementovány v jazyce PHP, lze s výhodou použít PHP sessions, tedy jakési dočasné soubory, obsluhované pomocí funkcí PHP. Sessions se budou používat spíše pro proměnlivá data (doba obnovy stránky, stav uživatele ...). Pro data trvalejšího rázu (zprávy, kontakty ...) se bude využívat databáze mySQL.



Obrázek 4-4: Způsob komunikace mezi daemonem a webovým klientem

Komunikace prostřednictvím PHP sessions

Proměnlivá data je možné pro každého uživatele ukládat do PHP session. Jedná se vlastně o soubor na straně serveru, ve kterém jsou obsaženy názvy uložených proměnných a jejich hodnoty. Výhodou oproti použití cookies je, že hodnoty jsou ukládány na serveru a nemohou být uživatelem zmanipulovány.

V systému Jabber2Go budou sessions použity pro komunikaci mezi daemonem a webovým klientem. To je umožněno díky tomu, že obě části budou implementovány pomocí jazyka PHP. Použití PHP funkcí pro práci se sessions zajistí oproti použití obyčejných souborů vyšší efektivitu.

Základní funkce pro práci se session jsou popsány v tabulce Tabulka 4-1. Bližší informace je možné nalézt v literatuře [9] nebo na internetu [10]. K hodnotě proměnné uložené v session se přistupuje pomocí superglobálního pole `$_SESSION`.

PHP funkce	Význam
<code>session_id()</code>	nastavení identifikátoru session
<code>session_start()</code>	inicializace session
<code>session_regenerate_id()</code>	vytvoření nového identifikátoru session
<code>session_write_close()</code>	okamžité zapsání změn do session
<code>session_unregister()</code>	zrušení proměnné v session
<code>session_destroy()</code>	zrušení všech dat uložených v session

Tabulka 4-1: PHP funkce pro práci se session

Komunikace prostřednictvím mySQL databáze

Data, která mají trvalejší charakter, budou uložena do mySQL databáze. Bude se jednat především o přijaté a odeslané zprávy, které se budou do databáze ukládat, čímž se umožní uživateli zobrazovat historii komunikace s daným kontaktem. Dále se do databáze budou ukládat uživatelovy kontakty, jejich stav a stavové informace, jako je priorita a textový popis.

Ke komunikaci s databází se bude používat standardních funkcí PHP, jejichž parametry často bývají SQL dotazy.

4.5.1 Zabezpečení

Aby se zabránilo případnému zjištění hesla uživatele nebo sledování jeho komunikace, bude možné používat bezpečnou komunikaci pomocí protokolu SSL (Secure Socket Layer). Tento protokol nabízí šifrováním zabezpečené spojení mezi komunikujícími stranami. Šifrována je celá komunikace, tedy i názvy požadovaných stránek, včetně dalších parametrů (například identifikátoru session).

Bohužel některé webové prohlížeče, zejména prohlížeče mobilních telefonů, nemusí tento způsob komunikace podporovat. Proto je možné komunikaci prostřednictvím SSL zakázat. Celá komunikace mezi daemonelem a webovým klientem je potom nešifrovaná a uživatel se tak vystavuje nebezpečí odposlechu hesla a únosu session neoprávněnou osobou.

5 Realizace systému Jabber2Go

Jak vyplývá z návrhu, systém se musí skládat ze dvou částí, a to daemona, který zprostředkovává komunikaci s Jabber serverem, a z webového klienta, který představuje uživatelské rozhraní systému. V této kapitole Vás seznámím se způsobem realizace těchto částí.

5.1 Komunikace mezi daemonem a webovým klientem

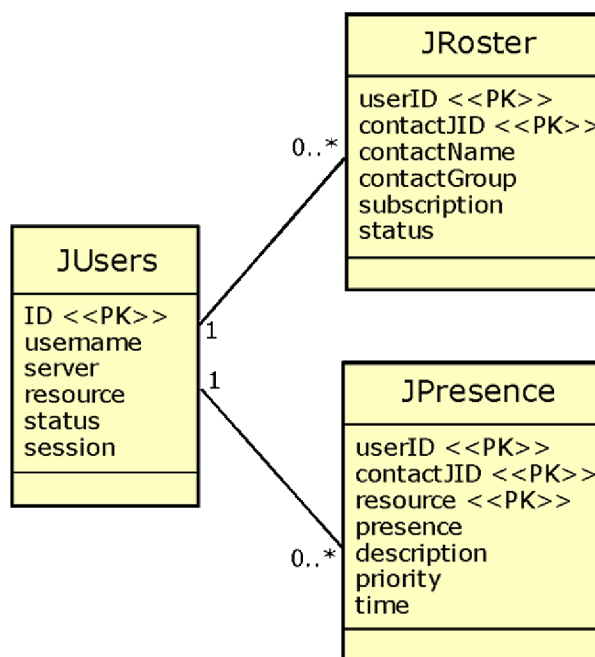
Pro další popis jednotlivých částí je nezbytné nejprve naznačit způsob komunikace mezi daemonem a webovým klientem.

Zvolený způsob komunikace závisí na tom, zda se jedná o data trvalejšího rázu nebo pouze o data dočasná. Data trvalejšího rázu jsou ukládána do databáze mySQL a data dočasná se uchovávají v PHP session přihlášeného uživatele.

Komunikace prostřednictvím databáze

Data, která jsou platná po dobu přihlášení uživatele nebo déle, jsou ukládána v mySQL databázi. Jedná se o seznam přihlášených uživatelů, roster s kontakty uživatele, stavy kontaktů a především přijaté a odeslané zprávy, které se uchovávají i po odhlášení uživatele od systému.

Pro manipulaci s databází se využívá standardních PHP funkcí, které jsou použity ve třídě mySQL (definované v souboru `mysql_class.php`).



Obrázek 5-1: Databázové tabulky (přihlášení uživatelé, seznam kontaktů, stav kontaktů)

JMessages
ID <<PK>>
JUser
remoteUser
time
body
subject
type

Obrázek 5-2: Databázová tabulka se zprávami

Komunikace pomocí PHP sessions

Jazyk PHP nabízí funkce pro ukládání proměnných do dočasných souborů na straně serveru. Tyto funkce vznikly kvůli bezstavovosti HTTP protokolu a umožňují udržovat vybrané informace, které uživatel během své práce se systémem zadal, nebo informace, které pro nás budou v pozdější době užitečné. Každá session je reprezentována jednoznačným identifikátorem, který se často označuje jako session ID, SID nebo PHPSESSID.

Aby bylo možné přihlášení více uživatelů z jednoho počítače najednou, bude se pro předávání session ID používat metoda GET HTTP protokolu (předávání v adrese stránky). Pokud by se k uložení session ID použily cookies, docházelo by k přepisování tohoto identifikátoru v rámci prohlížeče a mohl by tak být přihlášen pouze jeden uživatel (v rámci jednoho prohlížeče). Únosu identifikátoru session se není třeba obávat, protože systém Jabber2Go podporuje SSL šifrování, viz. kapitola 4.5.1 o zabezpečení systému.

5.2 Daemon

Jak již bylo několikrát řečeno, hlavním úkolem daemona je zprostředkovávat komunikaci s Jabber serverem přihlášeného uživatele. Daemon tak vlastně představuje vlastního klienta, který je ale řízen přes webové rozhraní.

Na podněty od webového klienta daemon reaguje příslušnou akcí a popřípadě také odesláním odpovídajícího XML elementu Jabber serveru. Webový klient dává podněty prostřednictvím PHP session a data trvalejšího rázu ukládá do mySQL databáze. Podrobnější popis realizace základních funkcí klienta bude uveden dále, v části 5.4.

Daemon je implementován v jazyce PHP a jeho hlavní části představují soubory `daemon.php` a `daemon_handle_user.php`.

5.2.1 Spuštění daemona

Podle konfiguračního souboru daemon může buď běžet neustále, nebo jen pokud je přihlášený nějaký uživatel. Konfigurace se děje pomocí parametru `DAEMON_RUN_STILL` v konfiguračním souboru `config.php`.

Podnět ke spuštění daemona dává webový klient při vyplnění všech nezbytných údajů k přihlášení a požadavku o přihlášení ze strany uživatele. Webový klient zajistí zapsání všech nezbytných údajů do session a do databázové tabulky `JUsers`, která představuje seznam uživatelů systému, vloží základní údaje o uživateli, zejména identifikátor session, který bude daemon dále používat pro komunikaci s webovým klientem.

V případě stálého běhu daemona odpadá problém s jeho spouštěním, protože se předpokládá, že již běží. Také po odhlášení všech uživatelů nedochází k jeho ukončení.

Pokud daemon neběží neustále, je třeba jej při přihlášení uživatele spustit. To se děje pomocí skriptu (`daemon_start.sh`), který zajistí spuštění daemona, pokud spuštěný ještě není. Skript je realizován pomocí Shell Scriptu, ve kterém se využívá funkcí a programů operačního systému. Pro zjištění, zda je daemon již spuštěn, se využívá programu `pgrep`.

Aby nedošlo k vícenásobnému spuštění daemona, je spouštění prováděno za pomoci uzamykání souborů. Spuštění pak probíhá jako atomická operace a nedochází k situaci, kdy by se více daemonů předbíhalo v obsluze uživatelů.

5.2.2 Běh daemona

V přístupu k běhu daemona je více možností, a to buď že jeden daemon obsluhuje všechny přihlášené uživatele, nebo že hlavní daemon zajišťuje vytváření podprocesů, které se starají o obsluhu pouze jednoho uživatele.

Soubor `daemon.php` představuje hlavního daemona, který v závislosti na konfiguraci zajistí buď obsluhu všech přihlášených uživatelů, nebo vždy vytvoří nový proces pro přihlašujícího se uživatele. Způsob konfigurace a konfiguračních parametrů systému Jabber2Go je uveden v příloze.

Aby daemon nebyl neustále aktivní a nepřetěžoval tak systém, je jeho proces vždy na určitou dobu uspán. Využívá se PHP funkce `sleep()`, která na určitou dobu odloží provedení. Konkrétní dobu je možné určit příslušnou hodnotou v konfiguračním souboru (parametr `DAEMON_SLEEP`).

5.2.3 Obsluha uživatele

Obsluha uživatele je implementována v souboru `daemon_handle_user.php`. Při obsluze uživatele se intenzivně využívá funkcí třídy `Jabber` (ze souboru `jabber_class.php`), v níž jsou dostupné funkce umožňující práci s Jabberem. Jedná se o funkce, které zajistí odeslání správného XML

elementu, vyhovujícímu Jabber protokolu. Dále se také využívá funkce třídy `XmlParser` (ze souboru `xml_class.php`), která zprostředkovává analýzu přijatých XML elementů z Jabber serveru.

Ideálním způsobem obsluhy uživatelů systému Jabber2Go je vytvoření procesu pro každého uživatele. V tomto případě existuje jeden hlavní daemon, který se stará o vytváření nových procesů pro obsluhu přihlášených uživatelů.

Nové procesy se vytváří pomocí PHP funkce `pcntl_fork()`. Process Control Functions (pcntl) jsou dostupné pouze na UNIXových platformách a PHP musí být zkompileováno s podporou těchto funkcí. Podpora se při kompilaci PHP aktivuje volbou `--enable-pcntl`.

Bohužel tyto funkce nemusí být vždy dostupné a systém se nesmí na jejich existenci spoléhat. Pokud tyto funkce nelze použít, bude jeden daemon obsluhovat všechny uživatele. Nastavení způsobu obsluhy se děje prostřednictvím konfiguračního souboru (parametrem `DAEMON_HANDLE_ALL`).

Pokud není možné vytvořit pro každého uživatele proces, bude jeden daemon obsluhovat všechny přihlášené uživatele.

5.2.4 Signály k probuzení daemona

Aby se nemuselo vždy čekat až do konce doby odložení zpracování daemona, je možné jej zasláním signálu probudit a vynutit tak okamžité provedení obsluhy uživatele. To se děje prostřednictvím zaslání signálu `SIGUSR1` procesu daemona.

Aby probuzení fungovalo, musí mít daemon nadefinovanou obsluhu tohoto signálu. Funkce pro práci se signály jsou implementovány v souboru `signal_functions.php` a obsluha signálů v souboru `signal_handler.php`. Pro zpracování signálů se v PHP používají Process Control Functions (pcntl), o kterých jsem se již zmínil v části 5.2.3.

Povolení nebo zakázání používat signály se děje prostřednictvím parametru nastaveného v konfiguračním souboru (parametr `SIGNALS_USE`).

5.2.5 Spojení s Jabber serverem

Spojení a komunikace s Jabber serverem probíhá prostřednictvím socketů. Využívá se PHP funkce pro práci se sockety. Důležitou funkcí je `fsockopen()`, která zajistí vytvoření socketu. Tato funkce vrací descriptor na otevřený socket, který je podobný descriptoru otevřeného souboru a může se s ním také tak pracovat. Je možné tedy používat funkci pro zápis `fwrite()` a pro čtení `fread()`.

Protože se spojení s Jabber serverem může nečekaně přerušit, musí se kontrolovat, zda je socket stále otevřen. Výhodné je kontrolovat, zda proběhl bez problémů zápis do socketu. V případě, že socket již není otevřen, zajistí daemon informování webového klienta, že uživatel již není připojen. Informování probíhá prostřednictvím `session`, ve které se nastaví hodnota stavu uživatele na stav „nedostupný“.

Práce se sockety je implementována ve třídě `Socket` (v souboru `socket_class.php`).

5.3 Webový klient

Webový klient představuje uživatelské rozhraní systému. Je implementován v jazyce PHP a vygenerované webové stránky jsou v XHTML verze 1.0. Pro dotvoření vzhledu se využívá kaskádových stylů (CSS). Podpora kaskádových stylů však není od prohlížeče vyžadována a systém bude mít zachovanou logickou strukturu i bez jejich použití.

5.3.1 Obnova stránky

Kvůli problémům vzniklým v důsledku použití HTTP protokolu, o kterých jsem se zmínil v části 4.2, je nutné stránku představující webového klienta neustále obnovovat (refresh rámce). XHTML tuto funkci umožňuje, jedná se o použití meta tagu `refresh`.

Aby nedocházelo ke zbytečnému zatěžování webového serveru a nepřiměřeným přenosům dat směrem k uživateli, umožňuje systém nastavit dobu obnovy stránky v závislosti na činnosti uživatele.

Pokud je uživatel pasivní (nedodesílá ani nepřijímá zprávy), roste postupně doba obnovy stránky, až tato hodnota dosáhne svého maxima. Pokud uživatel projeví činnost, nastaví se perioda obnovy na minimální hodnotu. Růst doby s každým načtením stránky je naznačen v tabulce Tabulka 5-1.

Aktuální hodnota obnovy [s]	Nová hodnota obnovy [s]
< 10	+1
<10; 20)	+2
<20; 40)	+5
>= 40	+10

Tabulka 5-1: Růst doby obnovy stránky

U některých prohlížečů dochází k nepřijemnému posouvání na začátek strany při každé její obnově. Tomuto neblahému jevu se dá předejít použitím JavaScriptu. Před obnovou stránky je do cookie uložena aktuální pozice a po následné obnově stránky je pozice znovu nastavena na uloženou hodnotu.

5.3.2 Realizace webového klienta

Jak jsem se již zmínil, celý webový klient je implementován v jazyce PHP, s využitím kaskádových stylů a doplnkově také s využitím JavaScriptu.

Výsledné stránky jsou generovány za častého využívání funkcí ze souboru `client_functions.php`. V tomto souboru jsou implementovány používané funkce, jako například zjištění seznamu kontaktů, informací o kontaktu, existence nepřečtených zpráv, nastavení nové obnovy stránky a podobně.

Pro komunikaci s daemonem a s Jabber serverem uživatele se využívá mySQL databáze a PHP session. Zprávy se získávají z databázové tabulky JMessages, seznam kontaktů z databázové tabulky JRoster a stavy kontaktů z tabulky JPresence.

Informace o přihlášeném uživateli jsou ukládány do session uživatele. Do session je ukládána také perioda obnovy stránky, skryté skupiny rosteru, povolení zobrazovat nedostupné kontakty a další nastavení, jako je například zákaz používání rámu nebo SSL spojení.

5.3.3 Části webového klienta

Seznam kontaktů – roster

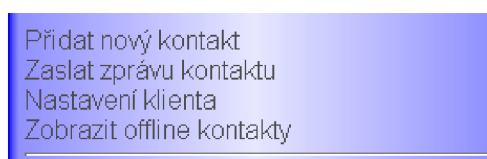
Hlavní část představuje titulní strana s rosterem a nabídkou možných akcí. Tato strana je implementována v souboru client.php.

Důležité je upozornit, že pokud uživatel není přihlášen, zobrazí se na této stránce formulář pro zadání přihlašovacích údajů uživatele.

Vzhled úvodní strany s hlavní nabídkou a seznamem kontaktů je znázorněn na obrázcích Obrázek 5-3, Obrázek 5-4, Obrázek 5-5.



Obrázek 5-3: Hlavní strana - logo a aktuální stav uživatele



Obrázek 5-4: Systémové menu



Obrázek 5-5: Seznam kontaktů

Komunikace s kontaktem

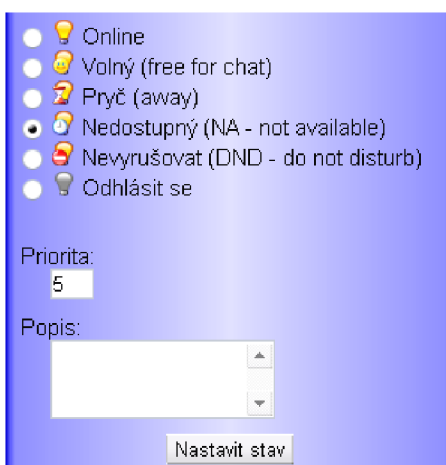
Okno komunikace s kontaktem se skládá ze tří ráků. V horní části je rám zobrazující možné akce přidružené ke kontaktu, který je implementován v souboru `chat_menu.php`. Uprostřed je rám zobrazující průběh komunikace s kontaktem, jež je implementován v souboru `chat_window.php`. Ve spodní části je formulář pro odeslání zprávy kontaktu. Tento formulář je implementován v souboru `chat_message.php`.

Pokud není povoleno používání ráků, je zobrazen pouze průběh komunikace s doplňkovou nabídkou akcí, zejména s odkazy na formulář pro odeslání zpráv a hlavní menu.

Nastavení stavu uživatele

Další funkcí webového klienta je umožnit nastavení aktuálního stavu uživatele. Formulář pro změnu stavu je implementován v souboru `status.php`.

Uživatel si vybere jeden z definovaných stavů a volitelně může nastavit také prioritu a popis stavu. Vzhled formuláře pro změnu stavu je naznačen na obrázku.



Obrázek 5-6: Část okna změny stavu uživatele

Správa kontaktů

Systém Jabber2Go umožňuje také kompletní správu kontaktů, jako je přidávání, editace, odstraňování a autorizace kontaktů. Formulář pro správu kontaktu je implementován v souboru `contact.php`.

Změna v rosteru probíhá uložením změny do příslušné databázové tabuky a nastavením příznaku, určujícího typ změny. Podrobnější popis realizace je popsán v části 5.4.

Nastavení klienta

Nastavení vlastností klienta Jabber2Go probíhá pomocí formuláře implementovaného v souboru `settings.php`. Nastavit je možné například pravidla a dobu obnovy stránky, používání ráků a SSL zabezpečení a dále také umožnění odstranění historie zpráv.

Veškeré změny nastavení se ukládají do session uživatele, pomocí které server generuje výslednou webovou stránku, která bude uživateli zobrazena.

5.4 Způsob realizace základních funkcí

V následujících podkapitolách bude naznačen způsob realizace základních funkcí, které systém Jabber2Go nabízí. Pro udržení přehlednosti nebudou vypsány jednotlivé XML elementy, pomocí kterých probíhá komunikace mezi Jabber serverem a daemonem. Přehled těchto elementů je možné nalézt v příloze.

5.4.1 Přihlášení uživatele

Přihlášení uživatele probíhá přes formulář, do kterého uživatel zadá údaje nezbytné k přihlášení. Po odeslání formuláře (přes zabezpečené spojení SSL) dojde k vytvoření nové session, do které jsou uloženy základní informace o uživateli. Dojde také k přidání nového záznamu do databáze, konkrétně tabulky `JUsers`. Nezbytnou položkou v tabulce je session ID, které se musí daemon dozvědět.

V tabulce `JUsers` je také udržováno jméno uživatele, název uživatelova Jabber serveru a uživatelův zdroj. Tyto informace musí být udržovány kvůli tomu, aby při pokusu o připojení ke stejnému účtu, se stejným názvem zdroje, došlo k vrácení správného session ID. Funkce je užitečná například při nechtěném uzavření okna, protože by pokus o nové přihlášení prošel až po vypršení doby na ohlášení uživatele. (Důvodem je, že není možné se dvakrát přihlásit pod stejným JID – vyplývá z Jabber protokolu.)

Jakmile daemon zjistí, že se uživatel chce připojit, vytvoří nový proces (pokud je to povoleno) a tento proces sérií XML elementů zajistí přihlášení k Jabber serveru. Podle úspěšnosti přihlášení daemon nastaví příznak v session, z kterého se webový klient dozví, jestli je uživatel přihlášen, či nikoliv.

5.4.2 Nastavení stavu uživatele

Nastavení stavu uživatele kompletně probíhá prostřednictvím session. Uživatel nejprve ve formuláři pro změnu stavu vybere druh stavu a volitelně může nastavit také prioritu a popis stavu. Tyto informace, včetně příznaku o změně stavu, se uloží do session uživatele. Pokud je to povoleno, dojde k zaslání signálu k probuzení daemona, který následně zajistí odeslání příslušného XML elementu na Jabber server. Pokud není povoleno probouzení daemona, dojde k této akci až po uplynutí doby odložení vykonání daemona.

5.4.3 Získání seznamu kontaktů

Požadavek na získání rosteru odesílá daemon již po úspěšném přihlášení uživatele. Získané položky daemon uloží do databáze, konkrétně do tabulky `JRoster`, znázorněné na obrázku Obrázek 5-1.

O kontaktech se ukládají standardní údaje, jako je jejich JID, jméno, skupina a autorizace (subscription).

Protože může dojít ke změně rosteru v průběhu připojení, Jabber server zajišťuje informování všech dostupných zdrojů uživatele o změně. Vykonává se tzv. roster push – zaslání nových informací o kontaktu. Daemon po přijetí aktualizace rosteru provede změny v databázové tabulce `JRoster` a seznam kontaktů se tak dostává do aktuální podoby.

5.4.4 Stav kontaktů

V průběhu přihlášení uživatele chodí informace o stavu kontaktů, které má uživatel povoleny odebírat. Tyto informace jsou ukládány do databázové tabulky `JPresence`, ve které se ukládá JID kontaktu (včetně zdroje), typ stavu, jeho prioritita, popis a také čas, kdy došlo ke změně stavu.

Stavem kontaktu se rozumí buď stav konkrétního zdroje kontaktu a v případě, že není zadán, určuje jej stav zdroje s nejvyšší prioritou. Pokud nastává situace, že více zdrojů má nastavenou stejnou a nejvyšší hodnotu priority, bere se stav zdroje, který zaslal informaci o svém stavu jako poslední.

V systému je implementována funkce „naposledy spatřen“, která umožňuje uživateli zjistit, kdy byl uživatel naposledy dostupný. Tento čas je časem, kdy byla přijata zpráva o nedostupnosti kontaktu.

5.4.5 Přidávání, odebrání a editace kontaktů

Uživateli je dále umožněno přidávání nových kontaktů do rosteru, odstraňování kontaktů a úprava informací (jméno, skupina) o nich.

Jak jsem se dříve zmínil, roster uživatele je udržován v tabulce `JRoster`. Standardní položky v rosteru mají nastaven příznak stavu položky (`status`) na hodnotu `ROSTER_NORMAL`, která značí, že se jedná o standardní položku.

Při přidání, úpravě či požadavku o smazání položky se u ní nastaví nový příznak – podle operace, která se má provést. Příznaky jsou: `ROSTER_ADD` nebo `ROSTER_REMOVE`.

Přidávání a úprava kontaktu probíhá podle Jabber protokolu stejně, proto se při editaci již existující položky rosteru nastaví příznak `ROSTER_ADD`.

5.4.6 Autorizace kontaktů

Důležitou částí systému je také autorizace kontaktů. Systém umožňuje zasílat požadavky na odebrání stavových informací uživatele, akceptování či zamítnutí přijatého požadavku a zrušení dříve akceptovaného požadavku.

V databázové tabulce `JRoster` dochází k nastavení příznaku položky kontaktu, která má být zpracována.

Protože může vzniknout v jeden okamžik více požadavků (typicky je to přidávání kontaktu a požadavek o autorizaci), jsou jednotlivé příznaky navrženy tak, aby se daly všechny rozpoznat. Příznak představuje číslo, v kterém je nastaven pouze jeden bit – mocniny 2 (2, 4, 8, ...). Pokud je

nastaveno více příznaků, bude mít výsledná hodnota nastaveno více jednotkových bitů a podle pozice těchto bitů se dá potom zjistit, o jaký požadavek se jedná.

Příznak	Význam
ROSTER_SUBSCRIBE_RECV	Přijat požadavek o autorizaci
ROSTER_SUBSCRIBED_RECV	Kontakt Vás autorizoval
ROSTER_UNSUBSCRIBE_RECV	Přijat požadavek o odstranění autorizace
ROSTER_UNSUBSCRIBED_RECV	Zrušena autorizace od kontaktu
ROSTER_SUBSCRIBE_SEND	Požadavek o autorizaci
ROSTER_SUBSCRIBED_SEND	Autorizace kontaktu
ROSTER_UNSUBSCRIBE_SEND	Požadavek na zrušení autorizace
ROSTER_UNSUBSCRIBED_SEND	Zrušení autorizace

Tabulka 5-2: Typy příznaků autorizace

5.4.7 Přijímání a odesílání zpráv

Přijaté a odeslané zprávy jsou udržovány v databázové tabulce `JMessages`. Protože se v tabulce udržují jak přijaté, tak odeslané zprávy, je třeba opět používat příznaky k určení typu zprávy. Základní typy zpráv jsou naznačeny v tabulce Tabulka 5-3.

Příznak (type)	Význam
MESSAGE_INCOMING	pro přijaté zprávy
MESSAGE_OUTGOING	pro odeslané zprávy
MESSAGE_TO_READ	pro nepřečtenou přijatou zprávu
MESSAGE_TO_SEND	pro nedeslanou zprávu

Tabulka 5-3: Příznaky určující typ zpráv

5.4.8 Odhlášení uživatele

Odhlášení probíhá buď na požadavek uživatele (nastavením příznaku v session), a nebo se děje automaticky po vypršení timeoutu pro ohlášení uživatele. Pokud vyprší tento timeout, značí to, že uživatel pravděpodobně uzavřel okno bez toho, aby se odhlásil.

Pokud se uživatel odhlašuje ručně, dojde zapsání požadavku na odhlášení do session. Jakmile daemon tento požadavek zjistí, zajistí odhlášení uživatele.

Daemon při každé obsluze uživatele zjišťuje, kdy se uživatel naposledy ohlásil. Čas posledního ohlášení se udržuje v session. Pokud tento čas překročí určitou mez (kterou je možno nastavit v konfiguračním souboru – parametr `CLIENT_TIMEOUT`), daemon zajistí odhlášení uživatele od systému a příslušného Jabber serveru.

Prvním krokem pro odhlášení je odeslání stavové informace s nastaveným stavem „unavailable“ – nedostupný. Následně dojde k uzavření socketu a ukončení obsluhujícího daemona (pokud se jedná o samostatný proces).

5.5 Požadavky systému na software

Požadavky na straně serveru

Protože je většina částí systému realizována ve skriptovacím jazyce PHP, je pro funkčnost systému vyžadována podpora PHP skriptů. Kvůli použití tříd je zapotřebí alespoň verze PHP 5.

Ačkoliv je PHP nezávislé na platformě, bude plná funkčnost systému zachována pouze na UNIXových platformách. A to především kvůli použití funkcí pro vytváření podprocesů a zaslání signálů. Na jiných platformách by systém mohl běžet s jistými omezeními (konfigurace systému viz. příloha).

Pro správnou funkčnost je třeba také databáze mySQL, postačující je verze 4.

Pro plnou funkčnost je také zapotřebí interpret Shell Scriptu. Pokud by nebyl dostupný, musí systém běžet bez podpory automatického spouštění daemona, nebo by se v konfiguračním souboru `config.php` musela nastavit cesta k jiným skriptům, které by spouštění zajistily.

Požadavky na klientské straně

Cílem bylo minimalizovat požadavky na aplikace a zařízení na klientské straně. Ke správné funkčnosti systému je zapotřebí pouze webový prohlížeč, který dovede zobrazit stránky XHTML verze 1.0.

Pro pohodlí uživatele se doporučuje prohlížeč, který podporuje kaskádové styly, rámy a JavaScript. Není to však podmínkou a systém funguje i bez nich.

Aby byla zachována bezpečnost uživatele, doporučuje se používat zabezpečené připojení přes SSL. Kvůli tomu, že jej všechny prohlížeče nemusí podporovat, dovoluje systém přihlášení i bez něho. Uživatel se ale vystavuje riziku, že bude odposlechnuto jeho heslo a nebo unesen jeho identifikátor session.

5.6 Monitorování a správa systému

Systém Jabber2Go umožňuje také jeho monitorování a správu. Je umožněno zobrazení statistik (např. celkový počet připojených uživatelů, počet aktuálně připojených uživatelů, přijatých a odeslaných zpráv a podobně.). Dále je umožněna správa systému (vytvoření potřebných tabulek v databázi, optimalizace tabulek databáze, odstranění starých zpráv atd.).

Správa systému je implementována v souboru `setup.php`. Funkce správy a monitorování jsou z prohlížeče webových stránek dostupné pouze po zadání administrátorského jména a hesla, které se mohou nastavit konstantami v souboru `setup.php`. Heslo je z bezpečnostních důvodů uloženo v zašifrované podobě (k šifrování se používá funkce `crypt()`).

Přihlášení ke správě systému a vlastní správa probíhá přes zabezpečené připojení pomocí SSL.

6 Závěr

Cílem práce bylo navrhnout a vytvořit instant messaging klienta pro komunikaci prostřednictvím sítě Jabber. Oproti jiným klientským aplikacím se systém liší tím, že klade minimální nároky na výbavu uživatele. K používání systému potřebuje uživatel pouze webový prohlížeč podporující jazyk XHTML.

Mohu konstatovat, že vytyčené cíle a požadavky byly splněny. Některé z cílů byly oproti původní představě dokonce překonány.

Výhodou systému je jeho nenáročnost a kladení minimálních nároků na výbavu uživatele. Díky tomu by neměl být problém systém Jabber2Go používat také na přenosných zařízeních PDA a mobilních telefonech, které disponují webovým prohlížečem. Aby byla uživatelům práce se systémem příjemná, jsou použity technologie, které nemusí některé starší či jednoduché prohlížeče podporovat. Jedná se především o JavaScript, rámy a SSL šifrování. Výhodou je, že popsané technologie nejsou pro fungování nezbytné, ale pouze poskytují uživatelům větší komfort při používání systému.

Součástí systému je také služba běžící na straně serveru, která přebírá veškerou zátěž klientského programu na sebe a zajišťuje komunikaci s uživatelským Jabber serverem. Tato část, označovaná jako daemon, je z uživatelského pohledu neviditelná a je pro něj důležité pouze to, že funguje bez ohledu na způsob realizace.

Ačkoliv si troufám tvrdit, že vzniklý systém je možné bez problému používat, bylo by pro plné uživatelské pohodlí vhodné realizovat několik dalších funkcí. Jedná se především o možnost přijímání a odesílání souborů a také o získávání „navštívenek“ (tzv. visit cards), které obsahují informace o uživateli, případně také jeho foto či jiný obrázek.

Přestože jsem při realizaci systému zažil spoustu krušných chvil a bezesných nocí, přinesla mi práce obrovské množství nových poznatků a zkušeností. Především jsem získal nový pohled na možnosti webových aplikací, kterým bych se chtěl v budoucnu věnovat.

Literatura

- [1] Farmbrough, R.: *Instant messaging* [online]
Dostupné z <http://en.wikipedia.org/wiki/Instant_messaging> (duben 2007)
- [2] Saint-Andre, P.: *XMPP RFCs*. [online]
Dostupné z <<http://www.xmpp.org/rfc/>> (duben 2007)
- [3] Český Jabber: *Jabber protokol* [online]
Dostupné z <<http://www.jabber.cz/wiki/>> (duben 2007)
- [4] Bradley, N.: *XML – kompletní průvodce*. Brno, Grada Publishing 2000. 540 s.
ISBN 80-7169-949-7
- [5] Quin, L.: *Extensible Markup Language (XML)* [online]
Dostupné z <<http://www.w3.org/XML/>> (duben 2007)
- [6] Boumphrey, F.; aj.: *XHTML Průvodce vývojáře*. Praha, Mobil Media a.s. 2002. 480 s.
ISBN 80-86593-14-2
- [7] Pemberton, S.; aj.: *XHTML 1.0 The Extensible HyperText Markup Language* [online]
Dostupné z <<http://www.w3.org/TR/xhtml1/>> (duben 2007)
- [8] Prokop, M.: *CSS pro webdesignery*. Praha, Mobil Media a.s. 2003. 288 s.
ISBN 80-86593-35-5
- [9] Castagnetto, J.; aj.: *Programujeme PHP profesionálně*. Praha, Computer Press 2001. 676 s.
ISBN 80-7226-310-2
- [10] Achour, M.: *PHP Manual* [online]
Dostupné z <<http://www.php.net/manual/>> (duben 2007)

Seznam příloh

Příloha 1. Příklady XML elementů Jabber protokolu

Příloha 2. Konfigurace systému Jabber2Go

Příloha 3. Přehled zdrojových souborů

Příloha 4. Uživatelská příručka

Příloha 5. CD

Příloha 1: Příklady XML elementů

Jabber protokolu

Zprávy

Odesílaná zpráva (z Jabber klienta):

```
<message to='A@s1.org' type='chat'>
  <subject>Subject</subject>
  <body>Message body</body>
</message>
```

Přijatá zpráva:

```
<message to='A@s1.org' from='B@s2.com/res' type='chat'>
  <subject>Subject</subject>
  <body>Message body</body>
  <x xmlns='jabber:x:delay' stamp='20070310T15:28:37' />
</message>
```

Nastavení informace o dostupnosti uživatele

Blíže nespecifikovaný stav – online

```
<presence>
  <status>Popis</status>
  <priority>1</priority>
</presence>
```

Stav je jeden z chat, away, xa, dnd

```
<presence>
  <show>xa</show>
  <status>Popis</status>
  <priority>1</priority>
</presence>
```

Odhlášení – nedostupný stav

```
<presence type="unavailable">
  <status>Odhlášen</status>
</presence>
```

Přijetí informací o stavu kontaktu

```
<presence from='abc@jabber.org/res' to='someone@somewhere.com/resource'>
  <priority>1</priority>
</presence>
```

Autorizace

Požadavek o autorizaci

```
<presence to='someone@somewhere.com' type='subscribe' />
```

Akceptování požadavku o autorizaci

```
<presence to='abc@jabber.org' type='subscribed' />
```

Zamítnutí požadavku o autorizaci

```
<presence to='abc@jabber.org' type='unsubscribed' />
```

Zrušení autorizace kontaktu

```
<presence to='abc@jabber.org' type='unsubscribed' />
```

Požadavek o zrušení autorizace

```
<presence to='someone@somewhere.com' type='unsubscribe' />
```

Roster

Požadavek na získání rosteru

```
<iq from='abc@jabber.org/resource' type='get' id='roster_1'>  
  <query xmlns='jabber:iq:roster' />  
</iq>
```

Odpověď s položkami rosteru

```
<iq to='abc@jabber.org/resource' type='result' id='roster_1'>  
  <query xmlns='jabber:iq:roster'>  
    <item jid='petr@jabber.org'  
      name='Petr'  
      subscription='both'>  
      <group>Friends</group>  
    </item>  
    <item jid='pavel@jabbim.cz'  
      name='Paja'  
      subscription='from'>  
      <group>Friends</group>  
    </item>  
    <item jid='emil@jwchat.org'  
      name='Emil'  
      subscription='to'>  
    </item>  
  </query>  
</iq>
```

Přidávání položky do rosteru nebo úprava již existující

```
<iq from='abc@jabber.org/resource' type='set' id='roster_2'>
  <query xmlns='jabber:iq:roster'>
    <item jid='someone@somewhere.com'
      name='Someone'>
      <group>Group name</group>
    </item>
  </query>
</iq>
```

Odstranění položky z rosteru

```
<iq from='abc@jabber.org/resource' type='set' id='roster_4'>
  <query xmlns='jabber:iq:roster'>
    <item jid='someone@somewhere.com' subscription='remove' />
  </query>
</iq>
```

Série XML elementů pro přihlášení uživatele

```
<?xml version='1.0' encoding='UTF-8'?>
<stream:stream xmlns:stream='http://etherx.jabber.org/streams'
  xmlns='jabber:client' to='jabbim.cz'>
<iq type='get' id='1000' to='jabbim.cz'>
  <query xmlns='jabber:iq:auth'><username>servac</username></query>
</iq>
<iq type='result' from='jabbim.cz' id='1000'>
  <query xmlns='jabber:iq:auth'><username>servac</username><password/>
  <digest/><resource/>
</query>
</iq>
<iq type='set' id='1001' to='jabbim.cz'>
  <query xmlns='jabber:iq:auth'><username>servac</username>
  <digest>9092d26c17fd3d1882e72c89ded9f428b7b79a25</digest>
  <resource>J2GO</resource>
</query>
</iq>
<iq type='result' from='jabbim.cz' id='1001' />
```

Odhlášení uživatele

```
<presence type='unavailable' />
</stream:stream>
```

Příloha 2: Konfigurace systému Jabber2Go

Aby byla umožněna funkčnost systému Jabber2Go na co nejvíce platformách a serverech, lze pomocí konfiguračního souboru nastavit chování systému.

Základními prvky konfigurace je nastavení databázového připojení, povolení používání signálů, povolení vytváření podprocesů, cesty ke skriptům a podobně. Seznam parametrů je zobrazen v tabulce.

Konfigurace probíhá prostřednictvím konfiguračního souboru `config.php`, ve kterém jsou obsaženy systémem používané konstanty, které však mohou být specifické pro konkrétní server.

Parametr	Význam
CLIENT_TIMEOUT	Doba timeoutu klienta – po vypršení této doby bez ohlášení uživatele dojde k automatickému odhlášení [doba v sekundách]
DAMEON_TIMEOUT	Doba timeoutu daemona – po vypršení této doby předpokládá webový klient, že daemon neočekávaně „zemřel“ a neobsluhuje již uživatele [doba v sekundách]
SIGNALS_USE	Příznak povolení používat signály [0/false – signály se nepoužívají; 1/true – signály povoleny]
DAEMON_RUN_STILL	Příznak nepřetržitého běhu daemona [0/false – daemon se spouští a ukončuje podle potřeby; 1/true – daemon běží neustále]
DAEMON_HANDLE_ALL	Příznak obsluhy všech uživatelů jedním daemonem [0/false – je povoleno vytváření podprocesů a pro každého uživatele se vytvoří nový proces; 1/true – jeden daemon obsluhuje všechny uživatele]
DAEMON_START_PATH	Cesta ke skriptu pro spuštění daemona
DAEMON_STOP_PATH	Cesta ke skriptu pro ukončení daemona
DAEMON_WAKEUP_PATH	Cesta ke skriptu pro probuzení daemona
DAEMON_MONITOR_PATH	Cesta ke skriptu pro monitorování daemona
MYSQL_SERVER	MySQL databázový server
MYSQL_DATABASE	Jméno používané mySQL databáze
MYSQL_USER	Jméno uživatele databáze
MYSQL_PASSWORD	Heslo uživatele databáze

Příloha 3: Přehled zdrojových souborů

Název souboru	Význam
client.php	Hlavní strana webového klienta, vložení částí podle stavu uživatele
client_connect.php	Přihlašovací formulář, zajištění přihlášení uživatele
client_connecting.php	Zobrazení průběhu přihlašování
client_defines.php	Definice používané webovým klientem
client_functions.php	Funkce používané webovým klientem
client_online.php	Strana přihlášeného uživatele, zobrazení stavu, menu a rosteru
common_functions.php	Funkce používané daemonem i webovým klientem
config.php	Konfigurační soubor systému
contact.php	Přidávání, úprava informací a mazání kontaktu
daemon.php	Hlavní část představující daemona
daemon_handle_user.php	Obsluha uživatele přihlášeného k systému
daemon_start.sh	Skript pro zajištění spuštění daemona
daemon_wakeup.sh	Skript pro probuzení daemona
defines.php	Systémem používané definice
error.php	Chybové hlášky
chat.php	Okno komunikace s kontaktem, popř. vybrání kontaktu ke komunikaci
chat_menu.php	Hlavní nabídka při komunikaci s kontaktem
chat_message.php	Formulář pro odeslání zprávy, odeslání zprávy
chat_window.php	Průběh komunikace s kontaktem
index.php	Úvodní strana – formulář pro přihlášení
install.php	Instalace systému – vytvoření potřebných tabulek
jabber_class.php	Třída realizující práci s Jabber protokolem
log_class.php	Třída zajišťující logování ladících informací do souboru
mysql_class.php	Třída pro práci s mySQL databází
settings.php	Uživatelské nastavení webového klienta
setup.php	Správa a monitorování systému pro administrátora
signal_functions.php	Funkce pro práci se signály
signal_handler.php	Obslužná rutina pro reakci na přijaté signály
socket_class.php	Třída pro manipulaci se sockety
status.php	Formulář změny stavu uživatele
template.php	Funkce pro vložení často používaných bloků kódu webového klienta
xml_class.php	Třída zajišťující analýzu přijatých XML elementů

Příloha 4: Uživatelská příručka

Přihlášení uživatele

V první fázi je zapotřebí zadat údaje, které jsou nezbytné k přihlášení do systému Jabber2Go a k Jabber serveru. Nezbytné je zadat uživatelské jméno, Jabber server a heslo k danému účtu. Povinný je také název zdroje. Jeho hodnota je přednastavena na „J2GO“. Pokud si uživatel přeje jiný název, může tuto hodnotu změnit. Přihlašovací formulář je znázorněn na obrázku Obrázek 1 a Obrázek 2.

Uživatel může také zvolit svůj počáteční stav. Pokud si uživatel nevybere některý z nabízených stavů, bude přihlášen se stavem online.

System umožňuje nastavit také některé další volby, jako je například číslo portu, na kterém běží uživatelův Jabber server, prioritu zdroje a také povolit či zakázat používání rámu nebo zabezpečeného připojení přes SSL. Tento rozšířený přihlašovací formulář se zobrazí po zvolení odkazu „Pokročilé volby“ v dolní části základního formuláře.

Po zadání nezbytných přihlašovacích údajů dojde k přihlášení uživatele k systému Jabber2Go a přihlášení k příslušnému Jabber serveru uživatele. Protože přihlášení k Jabber serveru nemusí proběhnout okamžitě, může být uživateli zobrazena informační hláška o tom, že probíhá připojování.

Po úspěšném přihlášení je zobrazena úvodní strana se seznamem kontaktů. V případě neúspěchu přihlášení je zobrazen úvodní formulář s hláškou o neúspěšnosti připojení.

Seznam kontaktů – roster

Po přihlášení uživatele je zobrazena strana se seznamem kontaktů, která představuje hlavní stranu systému Jabber2Go.

Jak je patrné z obrázku Obrázek 3, je zde zobrazen seznam kontaktů a obrázkovou ikonou žárovky je zobrazen jejich stav. Standardně jsou zobrazeny pouze kontakty, které jsou dostupné ke komunikaci. Dále zde uživatel může nalézt hlavní nabídku akcí, které je možné provést.

Odhlášení uživatele

Pokud uživatel již nechce pracovat se systémem a hodlá ukončit připojení k Jabberu, zvolí položku „Odhlásit“ na hlavní straně.

Pokud uzavře pouze okno a neodhlásí se, bude po vypršení určité doby (standardně 2 minuty) automaticky odhlášen.

Zobrazení / skrytí offline kontaktů

Pokud chce uživatel zobrazit všechny kontakty, které má v rosteru (včetně offline kontaktů), zvolí položku „Zobrazit offline kontakty“ v nabídce akcí na hlavní straně.

Opětovné skrytí offline kontaktů se provede volbou „Skrýt offline kontakty“.

Skrytí / zobrazení kontaktů ve skupině

Kliknutím na název skupiny dojde ke skrytí, popřípadě zobrazení kontaktů v příslušné skupině. Kontakty ve skupině zůstanou skryty i po obnově stránky nebo po přechodu na jinou stranu.

Změna stavu uživatele

Po výběru volby „Změnit stav“ na hlavní straně dojde k zobrazení formuláře pro změnu stavu uživatele (viz. Obrázek 4).

Je možné vybrat jeden z nabízených stavů a dále je také možné nastavit prioritu tohoto stavu, včetně informačního popisu stavu.

Kontakt a komunikace s ním

Po kliknutí na jméno kontaktu dojde k zobrazení okna komunikace s tímto kontaktem. Pokud je povoleno používání rámu (implicitní stav), skládá se toto okno komunikace z horního rámu, ve kterém je zobrazeno jméno a stav kontaktu a také nabídka akcí. Hlavní částí okna komunikace je zobrazení průběhu komunikace s kontaktem. V dolní části okna komunikace se nachází formulář pro odeslání zprávy kontaktu.

Zaslání zprávy kontaktu

Zprávu je možné odeslat prostřednictvím formuláře, který se nachází ve spodní části okna komunikace s kontaktem. Napsaná zpráva se odešle po stisku tlačítka „Odeslat“.

Odelaná zpráva se zobrazí buď okamžitě, nebo až po obnově stránky s průběhem komunikace. Tato skutečnost závisí na době obnovy stránky. Pokud je příliš velká, dojde k okamžité obnově, jinak se vyčká do konce naplánované obnovy.

Zaslání zprávy osobě, která není v rosteru

Aby bylo možné zaslat zprávu také kontaktům, které nejsou v seznamu, dojde po výběru „Zaslat zprávu kontaktu“ k zobrazení formuláře, kde může být zadán JID cílového kontaktu (Obrázek 6). Po výběru tohoto kontaktu dojde k zobrazení okna komunikace, jako je tomu po výběru kontaktu, který je v seznamu.

Přidání nového kontaktu

Přidat nový kontakt do seznamu kontaktů je možné po výběru volby „Přidat nový kontakt“ na hlavní straně. Ve formuláři pro přidání kontaktu je nezbytné zadat JID kontaktu a dále je možné také zadat zobrazované jméno a skupinu kontaktu. Zatržením „Požádat o autorizaci“ dojde také k zaslání požadavku o autorizaci kontaktu. Přidávání nového kontaktu probíhá prostřednictvím formuláře znázorněného na obrázku Obrázek 7.

Úprava informací o kontaktu

Upravit informace o již existujícím kontaktu je možné po výběru „Správa kontaktu“ v menu komunikace s kontaktem (Obrázek 5). Vlastní změna informací probíhá stejně jako přidávání kontaktu nového, s tím rozdílem, že není možné měnit JID kontaktu. Formulář editace kontaktu představuje Obrázek 8.

Smazání kontaktu

Odstranit kontakt ze seznamu kontaktů je možné po zvolení „Správa kontaktu“ v menu komunikace s příslušným kontaktem. Ve správě kontaktu je nutné zvolit „Odebrat tento kontakt“.

System se může následně dotázat, zda to myslíte s odebráním kontaktu vážně. Pro odebrání musíte svůj úmysl potvrdit.

Nastavení klienta

System Jabber2Go umožňuje přizpůsobit některé své vlastnosti podle představ uživatele. Jedná se především o povolení používání rámu a zabezpečeného spojení prostřednictvím SSL.

Dále je možné také nastavit dobu obnovy strany a nastavit způsob růstu periody obnovy v závislosti na nečinnosti uživatele.

Formulář pro změnu nastavení znázorňuje Obrázek 9.

Obrázky k uživatelské příručce



Uživatelské jméno:
Server:
Heslo:
Zdroj: J2GO
Stav: Online

Přihlásit


Pokročilé volby

Uživatelské jméno:
Server:
Port: 5222
Heslo:
Zdroj: J2GO
Priorita: 5
Stav: Online

Přihlásit

Pokročilé volby:
Zakázat rámy:
Zakázat SSL:
Aplikovat volby

Obrázek 1, Obrázek 2: Přihlašovací formulář



Online
 Volný (chat)
 Pryč (away)
 Nedostupný (na, xa)
 Nevyrušovat (dnd)
 Odhlásit se

Priorita:
8

Popis:
Nemám čas

Nastavit stav

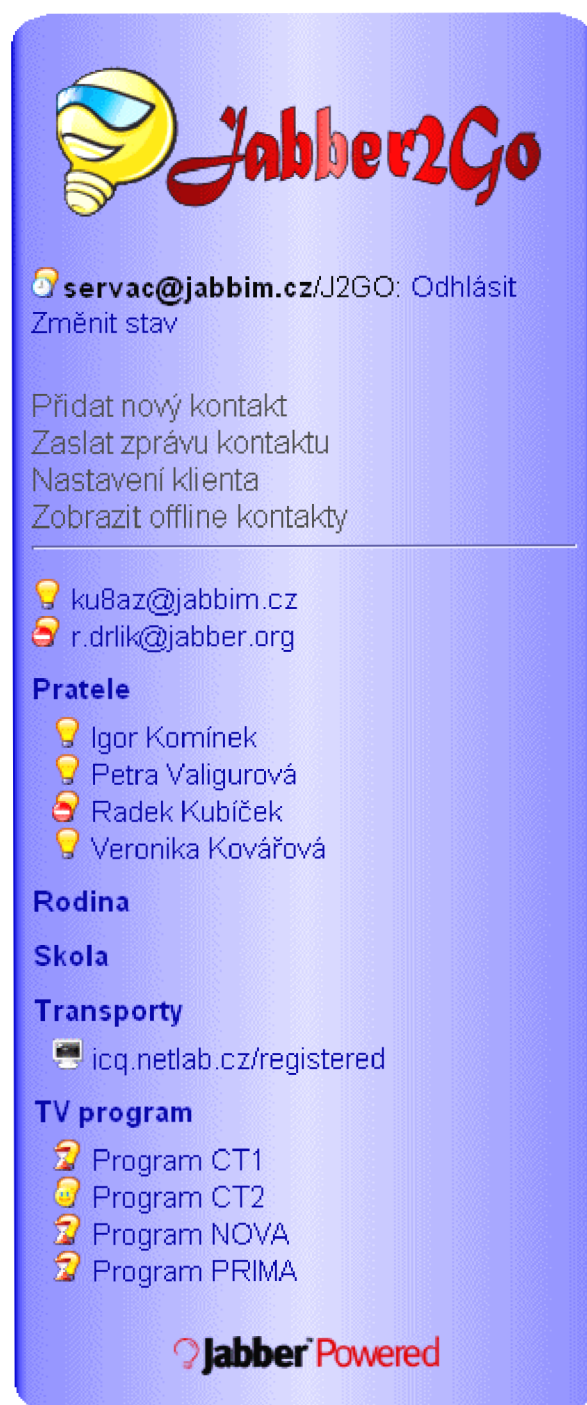
Obrázek 3: Nastavení stavu přihlášeného uživatele



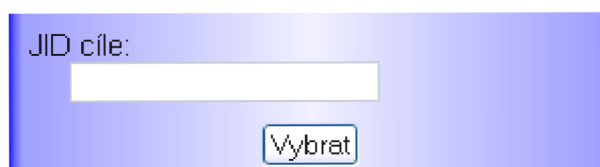
Zpět na roster Smazat zprávy Zobrazit historii Správa kontaktu Zastavit obnovu

r.drlik (subscription: both): Mám právě čas
r.drlik@jabber.org/ Defaultní zdroj Vybrat

Obrázek 4: Menu komunikace s kontaktem



Obrázek 5: Hlavní strana – seznam kontaktů



Obrázek 6: Zvolení cíle komunikace

JID kontaktu:

Jméno:

Skupina:

Požádat o autorizaci:

Obrázek 7: Formulář pro přidání nového kontaktu

[Odebrat tento kontakt](#)
[Odebrat autorizaci kontaktu](#)
[Autorizovat kontakt](#)

JID kontaktu:

Jméno:

Skupina:

Požádat o autorizaci:

Obrázek 8: Formulář editace kontaktu

Zvukové upozornění na novou zprávu:

Zobrazovat smajlíky jako obrázek:

Povoleno používání rámců:

Povoleno používání SSL:

Stálá doba obnovy:

Minimální doba obnovy: s

Maximální doba obnovy: s

Označit zprávu za historii po min

[Vymazat historii zpráv](#)
[Vymazat všechny zprávy](#)

Obrázek 9: Nastavení klienta