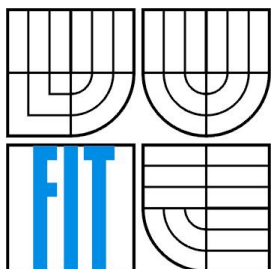


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM PRO SLEDOVÁNÍ PREZENCE OSOB NA ZÁKLADĚ VÍCE ZDROJŮ

INFORMATION SYSTEM FOR MONITORING OF A PRESENCE STATE FROM MULTIPLE POINTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. LIBOR RYŠAVÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. MAREK RYCHLÝ

BRNO 2008

Abstrakt

Diplomová práce se věnuje problému sledování prezence osob z více zdrojů. První část je věnována analýze způsobů propagace prezence osob v různých systémech a problematice sledování prezence z více zdrojů obecně. Následuje analýza požadavků a návrh informačního systému vyhodnocujícího prezenci osob z různých zdrojů s různými vlastnostmi a zobrazujícího prezenci osob graficky na mapách nebo plánech. Na závěr je popsána implementace a testování vytvořeného informačního systému.

Klíčová slova

prezence osob, sledování prezence z více zdrojů, grafické zobrazení prezence osob na plánech a mapách, Jabber, XMPP, BuddySpace, ICQ, MSN, SIP, webová aplikace, Java EE, Spring, JavaServer Faces, Hibernate

Abstract

This master's thesis deals with the problem of monitoring of a presence state from multiple points. It begins with analysis of presence information propagation in various systems and also with multiple points of presence in general terms. After that follows requirements analysis and design of information system which will evaluate presence information from multiple sources with various features and then display it graphically on plans or maps. At the conclusion there is described implementation and testing of developed information system.

Keywords

presence information, multiple points of presence, graphic representation of presence information on plans and maps, Jabber, XMPP, BuddySpace, ICQ, MSN, SIP, web application, Java EE, Spring, JavaServer Faces, Hibernate

Citace

Ryšavý Libor: Informační systém pro sledování prezence osob na základě více zdrojů. Brno, 2008, diplomová práce, FIT VUT v Brně.

Informační systém pro sledování prezence osob na základě více zdrojů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Mgr. Marka Rychlého.

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

.....
Libor Ryšavý
15.5.2008

Poděkování

Rád bych poděkoval Mgr. Marku Rychlému za odborné vedení a konzultace k tomuto projektu. Rád bych také poděkoval své rodině a přátelům za podporu během celého studia.

© Libor Ryšavý, 2008.

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
1 Úvod.....	3
2 Prezenze osob.....	4
2.1 Prezenze osob obecně.....	4
2.1.1 Praktické využití	5
2.1.2 Soukromí.....	5
2.1.3 Multiple Points of Presence	6
2.2 Propagace prezenze osob v různých systémech	6
2.2.1 Snahy o standardizaci prezenze osob.....	6
2.2.2 ICQ.....	7
2.2.3 Windows Live Messenger.....	8
2.2.4 Microsoft Unified Communications	9
2.2.5 Jabber / XMPP	9
2.2.6 BuddySpace	10
2.2.7 SIP.....	10
3 Návrh informačního systému	13
3.1 Analýza požadavků	13
3.1.1 Specifikace pojmů.....	13
3.1.2 Diagram případů použití	15
3.1.3 Specifikace aktérů.....	15
3.1.4 Specifikace případů použití.....	17
3.2 Konceptuální návrh tříd.....	29
3.3 Volba technologií pro implementaci	31
3.4 Návrh architektury.....	32
3.5 Databázové schéma	34
3.6 Návrh aplikace uživatelských práv.....	35
3.7 Návrh rozhraní modulů	37
3.7.1 Rozhraní modulů zdrojů informací.....	38
3.7.2 Rozhraní modulů zobrazení	38
3.8 Návrh uživatelského rozhraní.....	39
3.9 Lokalizace	40
4 Implementace	41
4.1 Programová dokumentace	41
4.2 Moduly	41

4.2.1	Implementované ukázkové moduly	41
4.3	Tvorba lokalizace	43
4.4	Instalace a konfigurace	43
4.5	Testování	44
4.5.1	Testování GUI	44
5	Závěr	45
	Literatura	46
	Seznam obrázků	47
	Seznam příloh	48

1 Úvod

S rozvojem Internetu a komunikačních technologií mají uživatelé stále větší potřebu detekovat prezenci jiných uživatelů připojených k síti. Typicky si takto ověřují, jestli je jejich partner pro komunikaci schopen a ochoten komunikovat. Například ve službách pro „instant messaging“ je prezence osob jednou ze základních poskytovaných funkcí. V tomto případě však bývá většinou informace o prezenci omezena pouze na zobrazení stavu (zda je uživatel připojen, odpojen, zaneprázdněn apod.), případně rozšířena o uživatelský text. V této práci se však budu zabývat prezencí osob v širším slova smyslu. Ve výsledném systému se budou definovat různá umístění a bude se zkoumat přítomnost osob v těchto místech. Informace o prezenci se budou navíc shromažďovat z více různých zdrojů (např. elektronický kalendář, IP telefon, „instant messaging“, docházkový nebo přístupový systém), které budou moci mít přiřazeny různé vlastnosti. Získanou informaci o prezenci osoby poté budu zobrazovat na mapách nebo plánech. Systém bude implementován jako webová aplikace s několika ukázkovými moduly pro vybrané zdroje a zobrazení.

První kapitola této práce se zabývá analýzou způsobů propagace prezence osob v různých existujících systémech, jako jsou různé protokoly pro „instant messaging“, VoIP (Voice over Internet Protocol) apod. V následující kapitole je proveden návrh informačního systému vyhodnocujícího prezenci osob z různých zdrojů s různými vlastnostmi. Následuje popis implementace a testování vytvořeného informačního systému. V závěru práce je uvedeno zhodnocení výsledků a návrh dalších možných rozšíření systému.

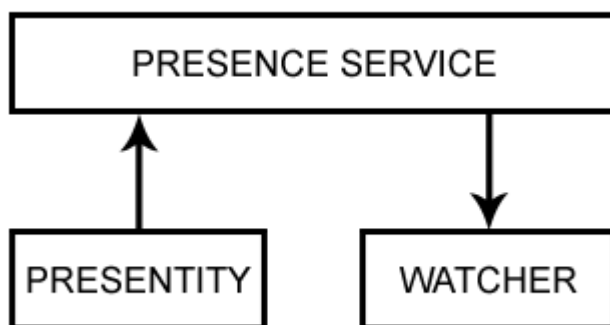
Tato práce navazuje na stejnojmenný semestrální projekt, ve kterém byla provedena analýza způsobů propagace prezence osob v různých existujících systémech a část návrhu informačního systému.

2 Prezenze osob

V této kapitole se nejprve dozvíte obecné informace o prezenci osob. Následuje analýza způsobů propagace prezenze osob v různých existujících systémech, jako jsou např. IM (instant messaging) nebo VoIP (Voice over Internet Protocol).

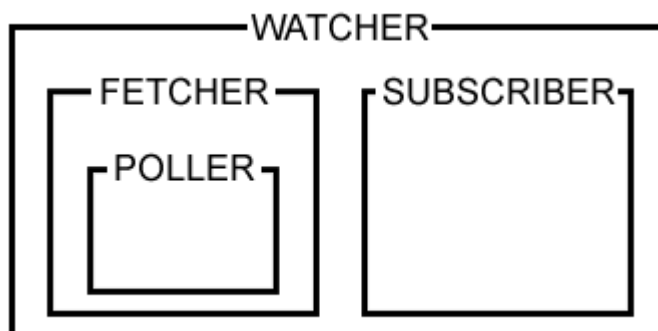
2.1 Prezenze osob obecně

Prezenze se dá definovat jako přítomnost ve smyslu účast nebo přítomnost na určitém místě v určitou dobu. Dále si popíšeme model předávání informace o prezenci (presence information) v počítačových a telekomunikačních sítích. Tento model se uplatňuje například u „instant messaging“, kde je prezenze chápána jako indikátor stavu, který zprostředkovává schopnost a ochotu potenciálního partnera pro komunikaci. Uživatelův klient zde poskytuje informaci o prezenci přes síťové připojení službě (presence service), která ji uloží jako záznam o dostupnosti osoby (personal availability record). Tato osoba se nazývá „presentity“. Záznam o dostupnosti osoby může být dále zpřístupněn pro distribuci jiným uživatelům nazývaným „watcher“, pro zjištění dostupnosti pro komunikaci (viz Obrázek 2.1).



Obrázek 2.1: Předávání informace o prezenci

Jsou dva druhy uživatelů typu „watcher“ zvané „fetcher“ a „subscriber“. „Fetcher“ se jednoduše dotazuje služby na aktuální hodnotu informace o prezenci nějaké „presentity“. Naproti tomu „subscriber“ si vyžádá oznamování (notification) od služby při každé změně informace o prezenci nějaké „presentity“. Speciálním případem „fetchera“ je „poller“, který načítá informace pravidelně (viz Obrázek 2.2).



Obrázek 2.2: Varianty uživatelů typu „watcher“

Služba si také zaznamenává informace o uživatelích typu „watcher“ (watcher information) a může je distribuovat dál stejným mechanismem, jako při distribuci informace o prezenci nějaké „presentity“ [1].

2.1.1 Praktické využití

Informace o prezenci má široké využití v mnoha komunikačních službách a je jednou z inovací, která může za zvyšující se popularitu „instant messaging“ (IM) a „voice over IP“ (VoIP). Nejběžnější použití prezence osob je dnes známo právě z klientů pro „instant messaging“, což je internetová služba, umožňující svým uživatelům sledovat, kteří jejich přátelé jsou právě připojeni, a dle potřeby jim posílat zprávy, soubory, chatovat apod. V těchto klientech se většinou zobrazuje grafický indikátor stavu, případně doplněný o textový popis. Běžnými stavy jsou „free for chat“ (můžu si povídat), „away“ (pryč) a „do not disturb“ (nerušit), které existují v různých obměnách ve všech moderních klientech pro „instant messaging“. Současné standardy podporují i bohatou nabídku dodatečných atributů k prezenci osoby, jako například uživatelovu náladu, polohu apod.

Informace o prezenci osob se stává stále významnější pro efektivní komunikaci při obchodním styku. Díky ní můžete například okamžitě vidět, kdo je dostupný ve vaší podnikové síti, což vám umožní větší flexibilitu při svolávání schůzek, či konferenčních hovorů. Výsledkem je přesná komunikace, která eliminuje neefektivnost emailových zpráv apod. Například pracovníci s přesnými informacemi o prezenci svých kolegů, se jich mohou snadněji dotazovat a dostat rychleji odpovědi, což může výrazně urychlit jejich práci.

Se všemi potenciálními výhodami, které prezence osob nabízí, bude jistě trvat jen krátkou dobu, než všechny významnější společnosti zavedou systémy pro sledování prezence osob.

2.1.2 Soukromí

Prezence je vysoce citlivá osobní informace a v netriviálních systémech by měly být definovány omezující podmínky, za kterých může být informace o prezenci poskytnuta. Například nějaký

pracovník chce, aby jeho kolegové viděli detailní informace o jeho prezenci pouze v pracovní době. Tento mechanismus je v jednoduché podobě implementován i v klientech pro „instant messaging“ pomocí blokování, čímž se uživatel může stát nedostupným pro konkrétní uživatele.

2.1.3 Multiple Points of Presence

Prezence začíná být pro komunikační systémy důležitá, pokud spojuje různé komunikační kanály. Kombinování stavů z různých komunikačních zařízení, sloužící k poskytování agregovaného pohledu na prezenci osoby, se nazývá „Multiple Points of Presence“ (MPOP). Tyto informace mají větší hodnotu, pokud jsou automaticky odvozené z pasivního sledování uživatelských akcí. Tato technika je například známá uživatelům „instant messaging“, kteří mají nastavený automatický přechod do stavu „away“ (pryč), pokud jsou určitou dobu neaktivní. Další zdroj informace o prezenci může být mobilní telefon, přihlášení k počítači nebo elektronický kalendář, kde se dá například zjistit, že je uživatel na jednání, služební cestě nebo třeba na dovolené.

MPOP stav může být následně použit pro automatické směrování příchozích zpráv napříč všemi přispívajícími zařízeními. Např. stav „out of office“ (mimo kancelář) může znamenat to, že systém směruje všechny příchozí zprávy a hovory na uživatelův mobilní telefon. Stav „do not disturb“ (nerušit) může ukládat všechny zprávy a doručit je později a telefonní hovory přesměrovat třeba do hlasové schránky.

MPOP vyžaduje spolupráci mezi více elektronickými zařízeními (např. mobilní telefon, IP telefon, IM klient, elektronický kalendář, ...) a službou (presence service) se kterou jsou všechna zařízení propojena. Doposud se nejvíce využívají uzavřené systémy se „Single Point of Presence“ (SPOP), kde stav publikuje pouze jedno zařízení. Pokud se uživatel pokusí připojit z dalšího zařízení, tak bývá z původního automaticky odhlášen a nebo mu připojení není vůbec umožněno. Pro MPOP je potřeba nejen schopnost komunikace více klientů mezi sebou, ale s informací o stavu musí být také vhodně zacházeno i všemi ostatními propojenými službami a MPOP schématem pro jejich klienty [2].

2.2 Propagace prezence osob v různých systémech

2.2.1 Snahy o standardizaci prezence osob

Několik pracovních skupin pracovalo a stále pracuje na standardizaci protokolů souvisejících s prezencí.

V roce 1999 byla organizací „Internet Engineering Task Force organization“ (IETF) založena skupina „Instant Message and Presence Protocol Working Group“ (IMPP WG), která měla za úkol vyvinout protokoly a datové formáty pro jednoduché služby prezence osob a „instant messaging“.

Naneštěstí se IMPP WG nebyla schopna shodnout na jednom protokolu pro prezenci. Místo toho vydala „Common Profile for Presence and Instant Messaging“ (CPP), který definoval sémantiku pro běžné služby prezence osob, aby usnadnily vytvoření brán mezi různými službami. Díky tomu jsou každé dva CPP-kompatibilní protokoly automaticky schopny komunikovat.

Roku 2001 IETF vytvořila pracovní skupinu SIMPLE (Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions) s cílem vytvořit CPP-kompatibilní standard pro prezenci a „instant messaging“ přes „Session Initiation Protocol“ (SIP). Činnost skupiny SIMPLE specifikovala rozšíření protokolu SIP, která umožnila sledování prezence osob a „instant messaging“. Tyto rozšíření obsahují bohaté formáty dokumentů o prezenci, kontrolu soukromí, specifické uveřejnění a oznámení, prezenci minulosti i budoucnosti, informace o pozorovateli a mnoho dalších zajímavých věcí. Zajímavé je, že navzdory svému jménu („simple“ znamená v angličtině jednoduchý) má tento standard k jednoduchosti daleko. Je popsán přibližně v 30 dokumentech (mnoho z nich jsou stále koncepty) na více než 1000 stranách.

Koncem roku 2001 vytvořily firmy Nokia, Motorola, a Ericsson iniciativu „Wireless Village“ (WV, „bezdrátová vesnice“), aby nadefinovala množinu univerzálních specifikací pro mobilní „Instant Messaging and Presence Services“ (IMPS) a služby pro prezenci v bezdrátových sítích. V říjnu 2002 byla „Wireless Village“ konsolidována do „Open Mobile Alliance“ (OMA) a o měsíc později vydala první verzi „OMA Instant Message and Presence Service“ založenou na XML. IMPS definuje architekturu systému, syntaxi a sémantiku pro reprezentaci prezence a množinu protokolů pro čtyři primární činnosti: prezence, IM, skupiny a sdílení obsahu. Prezence je klíčem umožňujícím technologii IMPS.

Na XML založený „Extensible Messaging and Presence Protocol“ (XMPP) byl navržen a je současně udržovaný nadací „XMPP Standards Foundation“ (dříve „Jabber Software Foundation“). Jde o robustní a široce rozšířený protokol. Je základem „Jabber Instant Messaging and Presence technology“. Je to také protokol používaný v komerční implementaci „Google Talk“. V říjnu 2004 publikovala pracovní skupina XMPP (pod IETF) dokumenty RFC 3920, RFC 3921, RFC 3922 a RFC 3923, aby standardizovala jádro XMPP protokolu.

2.2.2 ICQ

ICQ je nejrozšířenější služba pro „instant messaging“ v České republice. Používá protokol OSCAR (Open System for Communication in Realtime). Jde o proprietární protokol firmy AOL (America Online), jehož velká část je dnes pochopena díky reverznímu inženýrství. Níže uvedené informace pocházejí z neoficiální specifikace [3].

U ICQ jsou pro prezenci osob využívány uživatelské stavy, které se skládají ze dvou částí. První je různorodý příznak (např. narozeniny, zobrazení IP, ...) a druhá je příznak uživatelského stavu (např. připojen, zaneprázdněn, pryč, ...). Seznam možných hodnot první a druhé části:

STATUS_WEBWARE	Status webaware flag
STATUS_SHOWIP	Status show ip flag
STATUS_BIRTHDAY	User birthday flag
STATUS_WEBFRONT	User active webfront flag
STATUS_DCDISABLED	Direct connection not supported
STATUS_DCAUTH	Direct connection upon authorization
STATUS_DCCONT	DC only with contact users

Tabulka 2.1: První část stavu ICQ

STATUS_ONLINE	Status is online
STATUS_AWAY	Status is away
STATUS_DND	Status is no not disturb (DND)
STATUS_NA	Status is not available (N/A)
STATUS_OCCUPIED	Status is occupied (BUSY)
STATUS_FREE4CHAT	Status is free for chat
STATUS_INVISIBLE	Status is invisible

Tabulka 2.2: Druhá část stavu ICQ

2.2.3 Windows Live Messenger

Dříve pojmenován jako „MSN Messenger“ a „Windows Messenger“. Používá protokol „Microsoft Notification Protocol“ (MSNP). U tohoto protokolu jsou využívány 2 typy serverů:

- Notification Server (NS) – poskytuje prezenční služby. Připojení k tomuto serveru je nutné pro vytvoření sezení.
- Switchboard Server (SB) – poskytuje služby pro posílání zpráv. Připojení k tomuto serveru musí předcházet připojení k NS.

Informace o prezenci v tomto protokolu obsahuje stav (např. zaneprázdněn, pryč, ...), zobrazované jméno a číslo reprezentující různé doplňující informace o klientovi (např. přítomnost webové kamery). Jsou 2 základní stavy: připojen (open, online) a odpojen (closed, offline). Pokud se odpojíte od NS, tak je váš stav automaticky změněn na odpojen. Pokud se připojíte k NS, tak se stav automaticky nemění a uživatel si ho musí změnit sám. Uživatel ve stavu připojen si musí nastavit jeden z podstavů:

- NLN – Available (dostupný)
- BSY – Busy (zaneprázdněný)
- IDL – Idle (nečinný)
- BRB – Be Right Back (brzy budu zpět)
- AWY – Away (pryč)
- PHN – On the Phone (na telefonu)
- LUN – Out to Lunch (na obědě)
- HDN – Hidden (skrytý)

Tento podstav nijak neovlivňuje chování serveru k uživateli. Jde pouze o informaci pro ostatní uživatele [4].

Ve verzi „Windows Live Messenger 9“ se objevila podpora „Multiple Points Of Presence“, která umožňuje současné připojení z více klientů [5].

2.2.4 Microsoft Unified Communications

Toto řešení je postaveno na propojení komunikačních technologií (hardwarových i softwarových) do jednoho sourodého celku. Integruje e-mail, „instant messaging“, prezenci, hlas a video. Základem je Office Communications Server (OCS). Toto řešení je integrováno s aplikacemi Office (Word, Excel, Outlook atd.) a SharePoint. Je postaveno na standardních protokolech a podporuje SIP (Session Initiation Protocol) a SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions).

Je zde implementována i funkce „Multiple Points of Presence“ (MPOP), která umožňuje zjišťovat připojení uživatelů přes různá zařízení. Díky MPOP je možné zaslat uživateli zprávu na to zařízení, ke kterému je právě připojen, což zvyšuje efektivitu a rychlost komunikace [6].

2.2.5 Jabber / XMPP

Jabber byl projekt otevřeného komunikačního systému založený v roce 1998 Jeremiem Millerem. V roce 2004 byl na jeho základě vytvořen standard XMPP. V současné době označuje slovo Jabber ledasco, od protokolu až po komunikační síť. „Extensible Messaging and Presence Protocol“ (XMPP) je otevřená XML technologie pro komunikaci v reálném čase, která podporuje širokou škálu aplikací. Jeho základní standardizované normy jsou:

- RFC 3920 - Extensible Messaging and Presence Protocol (XMPP): Core [7]
- RFC 3921 - Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence [8]

Třebaže je jádro této technologie stabilní, komunita kolem XMPP pokračuje v definování různých rozšíření XMPP nazývaných XEP (XMPP Extension Protocol) , kterých je momentálně kolem dvou set (v různém stavu, od prvních návrhů až po standardy). Síť využívající XMPP protokol není centralizovaná do jednoho místa, jako je zvykem u většiny ostatních IM, ale je distribuovaná na servery po celém světě [9].

XMPP poskytuje i MPOP a to takovým způsobem, že přiřazuje klientům „resource“ (identifikátor) a pro každý „resource“ prioritu. Uživatel tedy může být připojen současně z více klientů s různým „resource“ a prioritou. Ve zprávě je poté buď uveden konkrétní cílový „resource“ a nebo je příchozí zpráva nasměrována na klienta s nejvyšší prioritou [8].

2.2.6 BuddySpace

BuddySpace přináší „instant messaging“ se čtyřmi důležitými vlastnostmi:

- Umožňuje použití map pro geografické a „office-plan“ (plány kanceláří) vizualizace jako rozšíření standardního seznamu kontaktů.
- Je postaven na „open source“ Jabberu, který umožňuje spolupracovat s ICQ, MSN, Yahoo a dalšími.
- Je implementován v Javě a je tedy multiplatformní.
- Byl vyvinut britskou výzkumnou laboratoří a je tedy dostupný zdarma se všemi zdrojovými kódy.

BuddySpace má za cíl poskytovat uživatelům rozšířené možnosti pro správu a vizualizaci prezenze kolegů a přátel, kteří spolupracují, komunikují nebo spolu třeba hrají hry. Důležitou roli zde hraje grafická reprezentace prezenze osob, která zahrnuje mapy, logické plány (např. plán budovy, časový plán projektu) a abstraktní umělecké nákresy (např. stěna pomalovaná sprejem). Skupina kolem tohoto projektu se také zabývá výzkumem sémantiky prezenze, abychom se posunuli od jednoduchých stavů jako jsou „připojen“ nebo „zanepřázdňen“ k bohatě kontextové a časoprostorové informaci, která bude více odpovídat uživatelově aktuální činnosti [10].

2.2.7 SIP

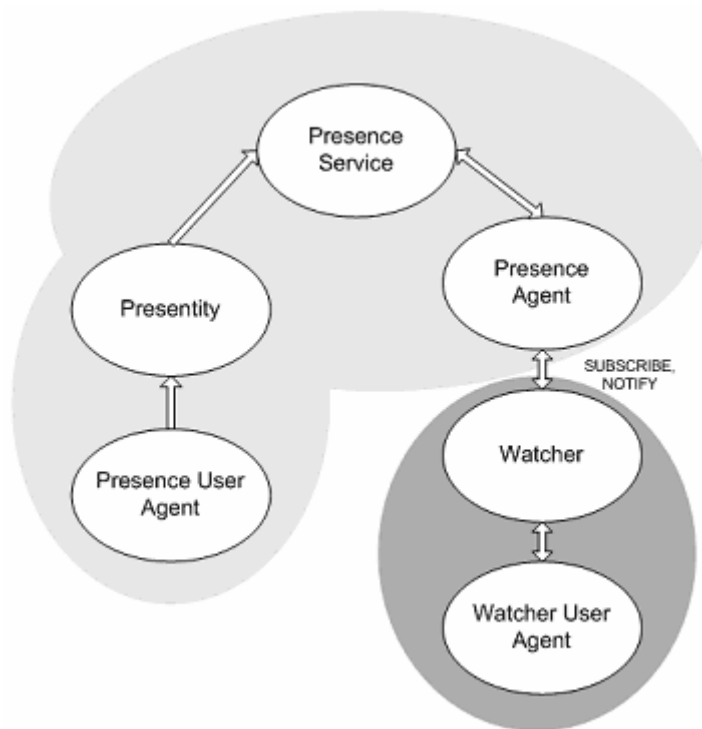
Session Initiation Protocol (SIP) je jednou z alternativ pro realizaci (nejen) hlasového přenosu v rámci IP sítě. V příslušném RFC je charakterizován jako signalizační protokol sloužící k sestavení, modifikaci a ukončení spojení mezi dvěma a více účastníky. Spojení může představovat obecně jakýkoliv multimediální přenos, v praxi je ale SIP nejčastěji využíván pro telefonování po IP síti.

Prezence osob je zde zastoupena zobrazením přítomnosti, které umožňuje, aby uživatel mohl na svém zařízení (telefon, aplikace) sledovat, zda konkrétní zvolení účastníci jsou dostupní (on-line), zda právě hovoří, či jsou nedostupní.

Pro účely transportu informací o přítomnosti účastníka jsou v protokolu SIP definovány dvě metody: „subscribe“ a „notify“. Zprávu „subscribe“ generuje účastník, který se chce dozvědět v jakém stavu (připojen, odpojen,...) je jiný účastník. Tato zpráva je stejně jako jiné SIP zprávy poslána prostřednictvím „proxy“ serverů na „presence agent“ (PA) zjišťovaného účastníka. PA přitom může být implementován přímo na „proxy“ serveru. Uvnitř každé zprávy „subscription“ je nastaven čas platnosti a před jeho vypršením je třeba generovat novou zprávu. PA na základě přijetí „subscription“ generuje zprávu „notify“, kterou oznamuje aktuální stav dotazovaného účastníka. Tato zpráva je pak samozřejmě posílána při každé změně stavu všem účastníkům, kteří zaslali „subscription“. V případě, že mě přestane zajímat v jakém stavu se konkrétní účastník nachází, je na jeho PA poslána „subscription“ s expirační dobou nula.

V předchozím odstavci byl použit pojem „presence agent“. Jedná se o jednu komponentu celé architektury (viz Obrázek 2.3) jejíž významné součásti jsou:

- Presence User Agent – Nástroj uživatele (aplikace, telefon), kterým ovládá stav své přítomnosti
- Presentity – Komponenta, která zprostředkovává standardní formou informaci o stavu přítomnosti na „presence service“
- Presence Service – Udržuje a distribuuje informace o aktuálním stavu svých uživatelů
- Presence Agent – SIP UA, reaguje na zprávy „subscribe“, komunikuje s „presence service“, odesílá zprávy „notify“.



Obrázek 2.3: Architektura systému pro zobrazení přítomnosti v protokolu SIP

Tato kapitola čerpá informace z [11].

3 Návrh informačního systému

3.1 Analýza požadavků

Základní požadavky na funkcionalitu a omezení systému budou definovány pomocí množiny případů použití zobrazených v diagramu podle notace UML a následně specifikovaných v textové podobě. Nejprve je ale potřeba specifikovat jednotlivé pojmy, které budou v analýze i dále v textu používány, aby se předešlo případným nedorozuměním.

3.1.1 Specifikace pojmů

3.1.1.1 Místo

Místem je rozuměna systémová reprezentace nějakého místa v reálném světě, které má nadefinováno unikátní jméno a adresu.

3.1.1.2 Zdroj

Zdroj obecně reprezentuje cokoli, co dodává systému informace o prezenci osob. Zdroje mohou být různých typů, které mají různorodé vlastnosti. Zdroje musí mít jednotné rozhraní, přes které se jich systém dotazuje na prezenci osob. Nové typy zdrojů mohou být do systému dodávány jako moduly implementující dané rozhraní.

3.1.1.3 Zobrazení

Zobrazení reprezentuje mechanismus, který je schopen graficky zobrazit prezenci osob na plánu nebo mapě. Zobrazení mohou být různých typů, které ale mají jednotné rozhraní. Pomocí tohoto rozhraní s nimi systém komunikuje a nechává zobrazit prezenci osob. Nové typy zdrojů mohou být do systému dodávány jako moduly implementující dané rozhraní.

3.1.1.4 Zobrazení místa

Zobrazení místa reprezentuje vyjádření určitého místa v určitém zobrazení. Jde tedy o vztah konkrétního místa a zobrazení, který obsahuje souřadnice daného místa používané v daném zobrazení.

3.1.1.5 Uživatel

Uživatelem je myšlena systémová reprezentace fyzické osoby, která pracuje se systémem. Uživatel může být buď autentizovaný, nebo anonymní. Autentizovaní uživatelé mají unikátní přihlašovací jméno, podle kterého jsou identifikováni. Anonymní uživatele od sebe nelze odlišit a je s nimi tedy v systému zacházeno jednotně.

Uživatelé nadefinovaní v systému současně reprezentují osoby, u kterých lze zjišťovat informace o prezenci.

3.1.1.6 Skupina

Skupiny reprezentují množiny uživatelů, které odpovídají skupinám osob v reálném světě, které mají nějaké společné rysy (např. zaměstnanci určité firmy). Skupiny se mohou překrývat - jeden uživatel tedy může patřit současně do více skupin.

U skupin lze následně vyhodnotit prezenci všech osob, které do skupiny náleží a zjistit tak například na jakém nejbližším místě se nějaká osoba z dané skupiny nachází.

3.1.1.7 Role

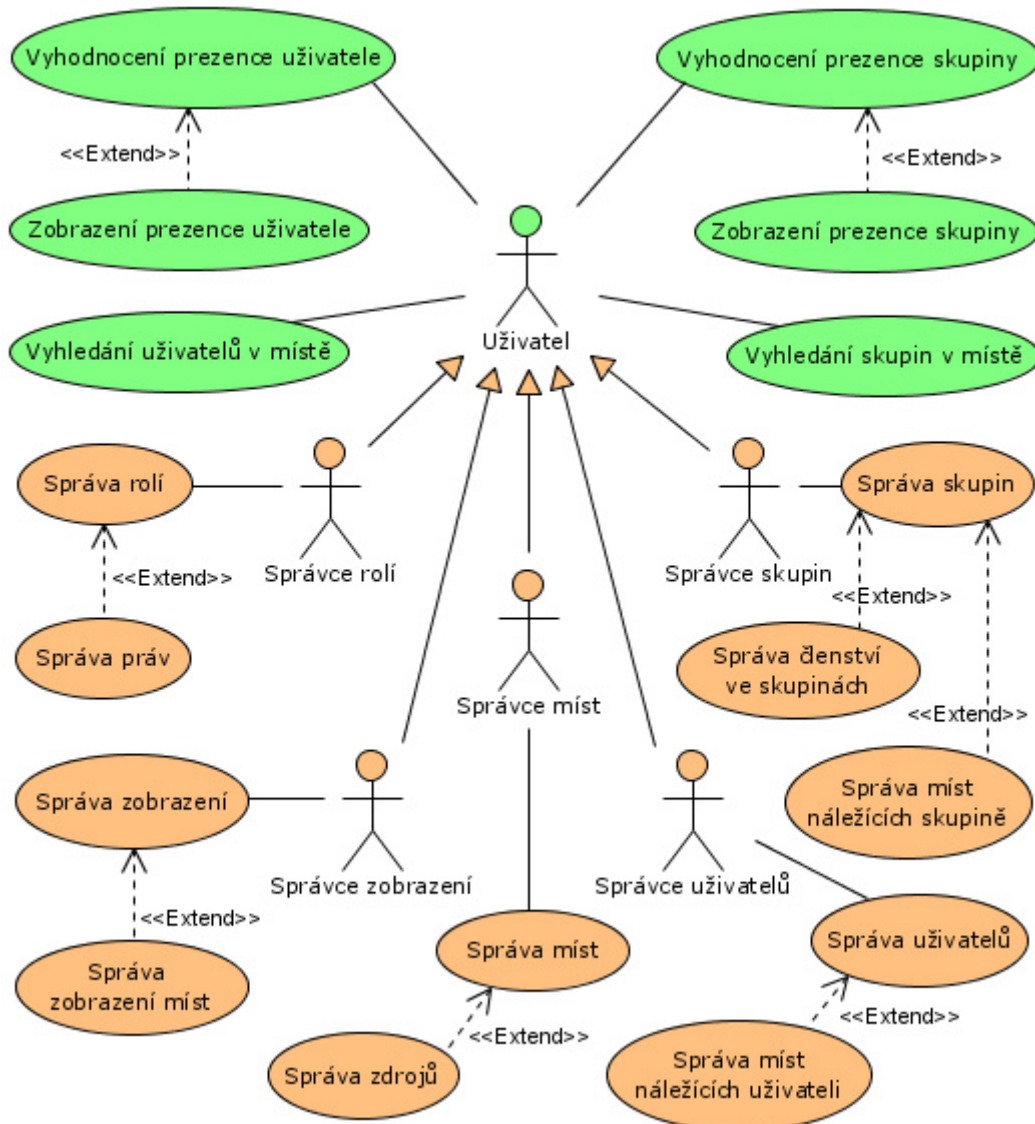
Role je míněna množina uživatelů, kteří mají v systému stejná práva (právo je specifikováno níže). Role jsou rozpoznávány podle unikátního jména. Každému uživateli může současně náležet více rolí.

3.1.1.8 Právo

Každé právo autorizuje své nositele k určitým činnostem v systému. Jednotlivá práva jsou přiřazována k rolím a jejich nositeli se tedy stávají uživatelé s danou rolí.

Práva jsou v systému dvojího typu. Prvním typem jsou práva „statická“, jejichž seznam je v systému nadefinován napevno. Jde například o práva správcovská (právo spravovat uživatele, právo spravovat místa apod.). Druhým typem jsou práva „na nahlížení na prezenci“ u jednotlivých uživatelů, skupin nebo míst. Tyto práva vymezují podmnožiny těchto entit, se kterými může uživatel disponující daným právem provádět odpovídající sledování prezence.

3.1.2 Diagram případů použití



Obrázek 3.1: Diagram případů použití

3.1.3 Specifikace aktérů

Nejprve specifikujeme jednotlivé aktéry v systému, kteří zastupují uživatele s určitými právy. Aktéři uvedení v diagramu případů použití odrážejí základní nastavení práv v systému. Tyto práva mohou být v systému dále různě seskupována a kombinována do uživatelských rolí. Lze například práva pro správu jednotlivých částí systému sjednotit do jedné role a vytvořit tak roli administrátora systému, který bude mít absolutní kontrolu nad systémem.

3.1.3.1 Aktér Uživatel

Aktér Uživatel reprezentuje obecného uživatele pracujícího se systémem. Může to být uživatel anonymní nebo uživatel autentizovaný. Anonymní uživatelé sdílí jednu speciální roli, protože je

od sebe nelze nijak odlišit. Autentizovaní uživatelé mohou být seskupováni do různých rolí a ty mohou být dále u jednotlivých uživatelů kombinovány.

Tento aktér může v systému používat základní prostředky pro sledování prezence osob. Tyto činnosti mohou být omezeny právy uživatelových rolí. U každé role je možnost nastavení viditelnosti jednotlivých míst, uživatelů a skupin. Lze tedy nastavit, že například anonymní uživatelé nebudou moci sledovat prezenci osob vůbec a nebo pouze na nějaké vybrané ukázkové množině.

3.1.3.2 Aktér Správce uživatelů

Právo na správu uživatelů je reprezentováno aktérem Správce uživatelů. Uživatel s tímto právem je autorizován k přidávání nových uživatelů do systému, editování jejich atributů a případnému odstraňování uživatelů ze systému.

3.1.3.3 Aktér Správce skupin

Aktér Správce skupin reprezentuje uživatele, jejichž role má přiřazeno právo na správu skupin. Tito uživatelé jsou oprávněni definovat do systému nové skupiny, stávající skupiny editovat a nebo mazat. Editace skupiny obsahuje kromě úpravy základních atributů také správu členů skupiny. Ta umožňuje přidávání členství ve skupině jednotlivým uživatelům systému a nebo toto členství rušit.

3.1.3.4 Aktér Správce rolí

Uživatelé s rolí, která má právo spravovat uživatelské role v systému je reprezentována aktérem Správce rolí. Uživatelé s tímto právem mohou přidávat do systému nové role, editovat role přidané dříve a případně je i mazat. Mohou také u každé role přidávat a odebírat uživatele, kteří danou rolí disponují. U rolí mají dále možnost nastavovat práva, kterými budou uživatelé s danou rolí disponovat.

3.1.3.5 Aktér Správce míst

Aktér Správce míst zastupuje uživatele, kteří mají roli disponující právem spravovat místa nedefinovaná v systému. Uživateli s tímto právem je umožněno přidávat do systému nová místa, jejich editaci a také mazání. Tento aktér je také oprávněn k nedefinovaným místům specifikovat zdroje informací o prezenci, upravovat jejich atributy a nebo je od místa odebírat.

3.1.3.6 Aktér Správce zobrazení

Právo na správu zobrazení je reprezentováno aktérem Správce zobrazení. Toto právo umožňuje uživateli definovat v systému nová zobrazení, editovat stávající a případně je mazat. Umožňuje také k jednotlivým obecným zobrazením přidávat zobrazení konkrétních míst, která již jsou v systému nedefinována. Zobrazení místa obsahuje například souřadnice daného místa potřebné k určení polohy ve zvoleném zobrazení. Již nedefinovaná zobrazení míst lze samozřejmě editovat a případně i mazat.

3.1.4 Specifikace případů použití

V této části je uvedena textová specifikace všech případů použití ve formě přehledných tabulek.

Název případu použití	Vyhodnocení prezence uživatele
Stručný popis	Systém vyhodnotí prezenci vybraného uživatele s ohledem na tazatelova práva.
Aktéři	Uživatel
Vstupní podmínky	Jsou načteny uživatelské role a jejich práva.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere z nabídky volbu „Prezence uživatele“. 2. Systém nabídne uživateli seznam osob, u kterých může být na základě jeho rolí zjištěna prezence. 3. Uživatel zvolí osobu, jejíž prezenci chce vyhodnotit. 4. Systém zjistí prezenci uživatele dotazy na zdroje všech míst, kde je hledaný uživatel viditelný hledajícímu uživateli. 5. Je zobrazen výsledek hledání. Pokud je uživatel nalezen, tak jsou vypsané údaje o nalezeném místě. Pokud je možné dané místo zobrazit, tak je uživateli nabídnuta tato volba (viz případ použití Zobrazení prezence uživatele).
Výstupní podmínky	Byl zobrazen výsledek hledání prezence uživatele.
Alternativní toky	Storno

Název případu použití	Zobrazení prezence uživatele
Stručný popis	Systém zobrazí graficky nalezenou prezenci uživatele na plánu nebo mapě.
Aktéři	Uživatel
Vstupní podmínky	Je vyhodnocena prezence uživatele a výsledné místo má nadefinováno zobrazení.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere volbu „Zobrazit místo“ (po případě použití Vyhodnocení prezence uživatele). 2. Systém nabídne uživateli seznam možných zobrazení daného místa. 3. Uživatel zvolí požadované zobrazení. 4. Systém zobrazí graficky prezenci uživatele na mapě nebo plánu.
Výstupní podmínky	Byla graficky zobrazena prezence osoby.
Alternativní toky	Storno

Název případu použití	Vyhodnocení prezence skupiny
Stručný popis	Systém vyhodnotí prezenci vybrané skupiny uživatelů s ohledem na tazatelova práva.
Akteři	Uživatel
Vstupní podmínky	Jsou načteny uživatelovy role a jejich práva.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere z nabídky volbu „Prezence skupiny“. 2. Systém nabídne uživateli seznam skupin, u kterých může být na základě jeho rolí zjištěna prezence. 3. Uživatel zvolí skupiny, jejíž prezenci chce vyhodnotit. 4. Systém zjistí prezenci uživatelů skupiny dotazy na zdroje všech míst, kde jsou uživatelé z vyhodnocované skupiny viditelní hledajícímu uživateli. 5. Je zobrazen výsledek hledání. Jde o seznam míst, ve kterých jsou přítomni někteří uživatelé z vyhodnocované skupiny. Pokud je to možné, tak se u jednotlivých míst zobrazí volba „Zobrazit místo“ a pokud je míst více, tak také volba „Zobrazit všechna místa“ (viz případ použití Zobrazení prezence skupiny).
Výstupní podmínky	Byl zobrazen výsledek hledání prezence skupiny.
Alternativní toky	Storno

Název případu použití	Zobrazení prezence skupiny
Stručný popis	Systém zobrazí graficky nalezenou prezenci skupiny uživatelů na plánu nebo mapě.
Akteři	Uživatel
Vstupní podmínky	Je vyhodnocena prezence skupiny uživatelů a některá z výsledných míst mají nadefinováno zobrazení.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere volbu „Zobrazit místo“ nebo „Zobrazit všechna místa“ (po případu použití Vyhodnocení prezence skupiny). 2. Systém nabídne uživateli seznam možných zobrazení vybraného místa nebo míst. 3. Uživatel zvolí požadované zobrazení. 4. Systém zobrazí graficky prezenci skupiny uživatelů na mapě nebo plánu.
Výstupní podmínky	Byla graficky zobrazena prezence skupiny uživatelů.
Alternativní toky	Storno

Název případu použití	Vyhledání uživatelů v místě
Stručný popis	Systém vyhledá uživatele přítomné ve vybraném místě s ohledem na tazatelova práva.
Akteři	Uživatel
Vstupní podmínky	Jsou načteny uživatelské role a jejich práva.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere z nabídky volbu „Uživatelé v místě“. 2. Systém nabídne uživateli seznam míst, u kterých může být na základě jeho rolí zjištěna prezence nějakých osob. 3. Uživatel zvolí místo, které chce prohledat. 4. Systém zjistí prezenci uživatelů v daném místě dotazy na zdroj informací na každého uživatele viditelného v daném místě. 5. Je zobrazen seznam osob přítomných v daný okamžik na vybraném místě.
Výstupní podmínky	Byl zobrazen výsledek hledání uživatelů v místě.
Alternativní toky	Storno

Název případu použití	Vyhledání skupin v místě
Stručný popis	Systém vyhledá skupiny u kterých je nějaký člen přítomen ve vybraném místě s ohledem na tazatelova práva.
Akteři	Uživatel
Vstupní podmínky	Jsou načteny uživatelské role a jejich práva.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere z nabídky volbu „Skupiny v místě“. 2. Systém nabídne uživateli seznam míst, u kterých může být na základě jeho rolí zjištěna prezence nějakých osob, které náleží do nějaké jemu viditelné skupiny. 3. Uživatel zvolí místo, které chce prohledat. 4. Systém zjistí prezenci členů skupin v daném místě dotazy na zdroj informací na každého člena skupiny viditelného v daném místě. 5. Je zobrazen seznam skupin, které mají alespoň jednoho člena přítomného v daný okamžik na vybraném místě.
Výstupní podmínky	Byl zobrazen výsledek hledání skupin v místě.
Alternativní toky	Storno

Název případu použití	Správa uživatelů
Stručný popis	Přidávání, editace a mazání uživatelů. V editaci také správa míst náležících uživateli.
Aktéři	Správce uživatelů
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa uživatelů.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere volbu „Správa uživatelů“. 2. Systém nabídne uživateli volbu „Přidat uživatele“ a také seznam všech uživatelů v systému a u každého volby „Editovat“ a „Smazat“. 3. Pokud uživatel zvolí „Přidat uživatele“ <ol style="list-style-type: none"> a. Systém zobrazí formulář pro vyplnění atributů nového uživatele. b. Dokud jsou zadané hodnoty neplatné Systém požaduje vyplnění atributů zákazníka. Systém ověří zadané hodnoty. c. Systém uloží nového uživatele. 4. Pokud uživatel zvolí „Editovat“ <ol style="list-style-type: none"> a. Systém zobrazí formulář s aktuálními hodnotami atributů vybraného uživatele, které lze editovat. Dále také možnosti pro správu míst náležících uživateli (viz případ použití Správa míst náležících uživateli). b. Dokud jsou vyplněné hodnoty neplatné Systém požaduje vyplnění atributů zákazníka. Systém ověří zadané hodnoty. c. Systém uloží upraveného uživatele. 5. Pokud uživatel zvolí „Smazat“ <ol style="list-style-type: none"> a. Uživatel je odstraněn ze systému.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa skupin
Stručný popis	Přidávání, editace a mazání skupin. V editaci také správa členů skupin a míst náležících skupině.
Akteři	Správce skupin
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa skupin.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere volbu „Správa skupin“. 2. Systém nabídne uživateli volbu „Přidat skupinu“ a také seznam všech skupin v systému a u každé volby „Editovat“ a „Smazat“. 3. Pokud uživatel zvolí „Přidat skupinu“ <ol style="list-style-type: none"> a. Systém zobrazí formulář pro vyplnění atributů nové skupiny. b. Dokud jsou zadané hodnoty neplatné Systém požaduje vyplnění atributů skupiny. Systém ověří zadané hodnoty. c. Systém uloží novou skupinu. 4. Pokud uživatel zvolí „Editovat“ <ol style="list-style-type: none"> a. Systém zobrazí formulář s aktuálními hodnotami atributů vybrané skupiny, které lze editovat. Dále také možnosti pro správu členů ve skupině (viz případ použití Správa členství ve skupinách) a míst náležících skupině (viz případ použití Správa míst náležících skupině). b. Dokud jsou vyplněné hodnoty neplatné Systém požaduje vyplnění atributů skupiny. Systém ověří zadané hodnoty. c. Systém uloží upravenou skupinu. 5. Pokud uživatel zvolí „Smazat“ <ol style="list-style-type: none"> a. Skupina je odstraněna ze systému.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa míst
Stručný popis	Přidávání, editace a mazání míst. V editaci místa také správa zdrojů informací o prezenci.
Akteři	Správce míst
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa míst.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere volbu „Správa míst“. 2. Systém nabídne uživateli volbu „Přidat místo“ a také seznam všech míst v systému a u každého volby „Editovat“ a „Smazat“. 3. Pokud uživatel zvolí „Přidat místo“ <ol style="list-style-type: none"> a. Systém zobrazí formulář pro vyplnění atributů nového místa. b. Dokud jsou zadané hodnoty neplatné Systém požaduje vyplnění atributů místa. Systém ověří zadané hodnoty. c. Systém uloží nové místo. 4. Pokud uživatel zvolí „Editovat“ <ol style="list-style-type: none"> a. Systém zobrazí formulář s aktuálními hodnotami atributů vybraného místa, které lze editovat. Dále také možnosti pro správu zdrojů informací o prezenci u daného místa (viz případ použití Správa zdrojů). b. Dokud jsou vyplněné hodnoty neplatné Systém požaduje vyplnění atributů místa. Systém ověří zadané hodnoty. c. Systém uloží upravené místo. 5. Pokud uživatel zvolí „Smazat“ <ol style="list-style-type: none"> a. Místo je odstraněno ze systému.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa zobrazení
Stručný popis	Přidávání, editace a mazání zobrazení. V editaci zobrazení také správa zobrazení konkrétních míst.
Aktéři	Správce zobrazení
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa zobrazení.
Hlavní tok	<p>6. Příklad použití se spustí, když uživatel vybere volbu „Správa zobrazení“.</p> <p>7. Systém nabídne uživateli volbu „Přidat zobrazení“ a také seznam všech zobrazení v systému. U každého volby „Editovat“ a „Smazat“.</p> <p>8. Pokud uživatel zvolí „Přidat zobrazení“</p> <ul style="list-style-type: none"> d. Systém zobrazí formulář pro vyplnění atributů nového zobrazení. e. Dokud jsou zadané hodnoty neplatné Systém požaduje vyplnění atributů zobrazení. Systém ověří zadané hodnoty. f. Systém uloží nové zobrazení. <p>9. Pokud uživatel zvolí „Editovat“</p> <ul style="list-style-type: none"> d. Systém zobrazí formulář s aktuálními hodnotami atributů vybraného zobrazení, které lze editovat. Dále také možnosti pro správu zobrazení konkrétních míst (viz případ použití Správa zobrazení míst). e. Dokud jsou vyplněné hodnoty neplatné Systém požaduje vyplnění atributů zobrazení. Systém ověří zadané hodnoty. f. Systém uloží upravené zobrazení. <p>10. Pokud uživatel zvolí „Smazat“</p> <ul style="list-style-type: none"> b. Zobrazení je odstraněno ze systému.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa rolí
Stručný popis	Přidávání, editace a mazání rolí. V editaci role také správa jejích práv.
Aktéři	Správce rolí
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa rolí.
Hlavní tok	<ol style="list-style-type: none"> 1. Příklad použití se spustí, když uživatel vybere volbu „Správa rolí“. 2. Systém nabídne uživateli volbu „Přidat roli“ a také seznam všech rolí v systému. U každé volby „Editovat“ a „Smazat“. 3. Pokud uživatel zvolí „Přidat roli“ <ol style="list-style-type: none"> a. Systém zobrazí formulář pro vyplnění atributů nové role. b. Dokud jsou zadané hodnoty neplatné Systém požaduje vyplnění atributů role. Systém ověří zadané hodnoty. c. Systém uloží novou roli. 4. Pokud uživatel zvolí „Editovat“ <ol style="list-style-type: none"> a. Systém zobrazí formulář s aktuálními hodnotami atributů vybrané role, které lze editovat. Dále také možnosti pro správu práv dané role (viz případ použití Správa práv). b. Dokud jsou vyplněné hodnoty neplatné Systém požaduje vyplnění atributů role. Systém ověří zadané hodnoty. c. Systém uloží upravenou roli. 5. Pokud uživatel zvolí „Smazat“ <ol style="list-style-type: none"> a. Role je odstraněna ze systému.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa členství ve skupinách
Stručný popis	Přidávání nebo odebrání členů skupin.
Aktéři	Správce skupin
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa skupin a edituje nějakou skupinu.
Hlavní tok	<ol style="list-style-type: none"> 1. Systém při editaci skupiny nabízí volbu „Přidat uživatele do skupiny“ a seznam všech uživatelů ve skupině. U každého člena také volbu „Odebrat ze skupiny“. 2. Pokud uživatel zvolí „Přidat uživatele do skupiny“ <ol style="list-style-type: none"> a. Systém zobrazí výběr uživatelů, které je možné přidat. b. Uživatel zvolí, koho chce do skupiny přidat. c. Vybranému uživateli je přidáno členství ve skupině. 3. Pokud uživatel zvolí „Odebrat ze skupiny“ <ol style="list-style-type: none"> a. Systém odebere vybraného uživatele ze skupiny.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa míst náležících skupině
Stručný popis	Přidávání nebo odebrání míst náležících skupině.
Aktéři	Správce skupin
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa skupin a edituje nějakou skupinu.
Hlavní tok	<ol style="list-style-type: none"> 4. Systém při editaci skupiny nabízí volbu „Přidat místo ke skupině“ a seznam všech míst náležících skupině. U každého místa také volbu „Odebrat od skupiny“. 5. Pokud uživatel zvolí „Přidat místo ke skupině“ <ol style="list-style-type: none"> d. Systém zobrazí výběr míst, které je možné skupině přidat. e. Uživatel zvolí místo, které chce skupině přidat. f. Vybrané místo je přidáno ke skupině. 6. Pokud uživatel zvolí „Odebrat od skupiny“ <ol style="list-style-type: none"> b. Systém odebere vybrané místo od skupiny.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa míst náležících uživateli
Stručný popis	Přidávání nebo odebrání míst náležících uživateli.
Aktéři	Správce uživatelů
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa uživatelů a edituje nějakého uživatele.
Hlavní tok	<p>7. Systém při editaci uživatele nabízí volbu „Přidat místo k uživateli“ a seznam všech míst náležících uživateli. U každého místa také volbu „Odebrat od uživatele“.</p> <p>8. Pokud uživatel zvolí „Přidat místo k uživateli“</p> <ul style="list-style-type: none"> g. Systém zobrazí výběr míst, které je možné uživateli přidat. h. Uživatel zvolí místo, které chce přidat. i. Vybrané místo je přidáno k vybranému uživateli. <p>9. Pokud uživatel zvolí „Odebrat od uživatele“</p> <ul style="list-style-type: none"> c. Systém odebere vybrané místo od uživatele.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

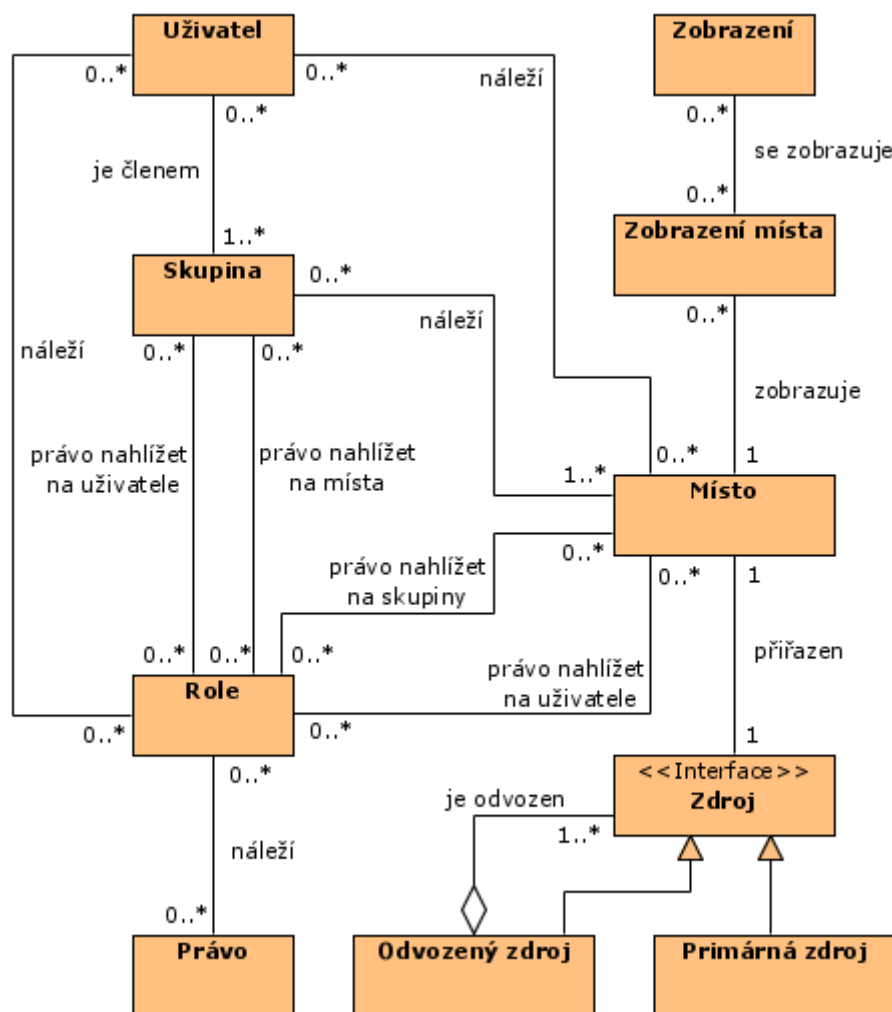
Název případu použití	Správa zdrojů
Stručný popis	Přidávání nebo odebrání zdrojů informací o prezenci u zvoleného místa.
Aktéři	Správce míst
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa míst a edituje nějaké místo.
Hlavní tok	<p>1. Systém při editaci skupiny nabízí volbu „Přidat zdroj k místu“ a hierarchii všech zdrojů přiřazených k místu. U každého zdroje také volbu „Odebrat“.</p> <p>2. Pokud uživatel zvolí „Přidat zdroj k místu“</p> <ul style="list-style-type: none"> a. Systém zobrazí výběr zdrojů, které lze k místu přidat. b. Uživatel zvolí zdroj, který chce přidat a také kam se má zařadit v hierarchii zdrojů u daného místa. c. Vybranému místu je přidán zvolený zdroj. <p>3. Pokud uživatel zvolí „Odebrat“</p> <ul style="list-style-type: none"> a. Systém odebere vybraný zdroj od místa.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa zobrazení míst
Stručný popis	Přidávání nebo odebrání zobrazení míst u zvoleného zobrazení.
Akteři	Správce zobrazení
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa zobrazení a edituje nějaké zobrazení.
Hlavní tok	<ol style="list-style-type: none"> 1. Systém při editaci zobrazení nabízí volbu „Přidat zobrazení místa“ a výpis všech zobrazení míst nadefinovaných u editovaného zobrazení. U každého zobrazení místa také volbu „Odebrat“. 2. Pokud uživatel zvolí „Přidat zobrazení místa“ <ol style="list-style-type: none"> a. Systém zobrazí výběr míst, které lze ve vybraném zobrazení zobrazit. b. Dokud jsou vyplněné hodnoty neplatné Systém požaduje výběr místa a vyplnění atributů vytvářeného zobrazení místa. Systém ověří zadané hodnoty. c. Vybranému zobrazení je přidáno právě nadefinované zobrazení místa. 3. Pokud uživatel zvolí „Odebrat“ <ol style="list-style-type: none"> a. Systém odebere od zobrazení vybrané zobrazení místa.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

Název případu použití	Správa práv
Stručný popis	Přidávání nebo odebrání práv u zvolené role.
Aktéři	Správce rolí
Vstupní podmínky	Uživatel je autentizován a má roli obsahující právo správa rolí a edituje nějakou roli.
Hlavní tok	<ol style="list-style-type: none"> 1. Systém při editaci role nabízí volbu „Přidat právo k roli“ a seznam všech práv přiřazených k roli. U každého práva také volbu „Odebrat“. 2. Pokud uživatel zvolí „Přidat právo k roli“ <ol style="list-style-type: none"> a. Systém zobrazí výběr práv, které lze k roli přiřadit. Jsou zde uvedena jak práva statická, tak i práva na nahlížení na různé entity v systému, u kterých je nutné ještě vybrat, k jaké konkrétní entitě se bude právo vázat. b. Uživatel zvolí právo, které chce přidat. Pokud se jedná o právo nahlížení na nějakou entitu, tak je mu nabídnut seznam možných entit, ze kterých jednu vybere. c. Vybrané roli je přidáno zvolené právo. 3. Pokud uživatel zvolí „Odebrat“ <ol style="list-style-type: none"> a. Systém odebere vybrané právo od role.
Výstupní podmínky	Úpravy nadefinované uživatelem jsou promítnuty do databáze a uživatel je o výsledku této akce informován.
Alternativní toky	Storno

3.2 Konceptuální návrh tříd

V předchozí kapitole byly pomocí případů použití specifikovány požadavky na vytvářený systém a nyní bude uveden konceptuální návrh tříd, který z nich vychází. Tento návrh zachycuje pouze koncept tříd zajišťujících hlavní „business“ logiku aplikace.



Obrázek 3.2: Konceptuální diagram tříd

System musí být schopný vyhodnocovat prezenci osob z více různých zdrojů s různými vlastnostmi (váha zdrojů, vzájemné vazby atd.). To bude realizováno možností vytvořit u každého místa stromovou hierarchii zdrojů, jejíž listy jsou primární zdroje. Ostatní uzly jsou zdroje odvozené, které mohou různým způsobem vyhodnocovat informace od zdrojů, které jim tvoří ve stromu zdrojů synovské uzly. Kořen stromu tvoří zdroj, který kompletně vyhodnotí prezenci pro dané místo. Tento model umožňuje snadnou tvorbu odvozených zdrojů, které budou pouze sjednocovat informace z několika zdrojů. Navíc je zde možnost zavést i mechanismus vyrovnávací paměti (cache) informací o prezenci. Jednoduše se vytvoří odvozený zdroj, který se při prvním dotazu dotáže zdroje podřízeného. Vracenou informací si však zapamatuje a po určité době bude na další dotazy automaticky vracet tuto uloženou informaci.

Pro vytvoření stromové hierarchické struktury zdrojů je zde využit návrhový vzor „composite“. Existuje zde jednotné rozhraní se jménem zdroj. To implementují jak primární zdroje, tak odvozené zdroje, které jsou tzv. „kompozitními objekty“. Klient se tedy nemusí starat o jaký zdroj se jedná, ale pracuje se všemi stejně.

Důležitou záležitostí v tomto modelu jsou také vztahy vyjadřující práva role na nahlížení na prezenci u konkrétních míst nebo skupin. Právo nahlížet na uživatele (vztah mezi rolí a skupinou) opravňuje jeho držitele zjistit, kteří uživatelé jsou členy dané skupiny. Právo nahlížet na místa u skupiny umožňuje svému nositeli zjistit místa které náleží vybrané skupině. Právo nahlížet na uživatele (vztah mezi rolí a místem) umožňuje uživateli, který tímto právem disponuje, zobrazit u daného místa uživatele, ke kterým toto místo náleží. Posledním z této skupiny práv je právo nahlížet na skupiny u nějakého místa, které opravňuje k zjištění skupin, kterým náleží vybrané místo.

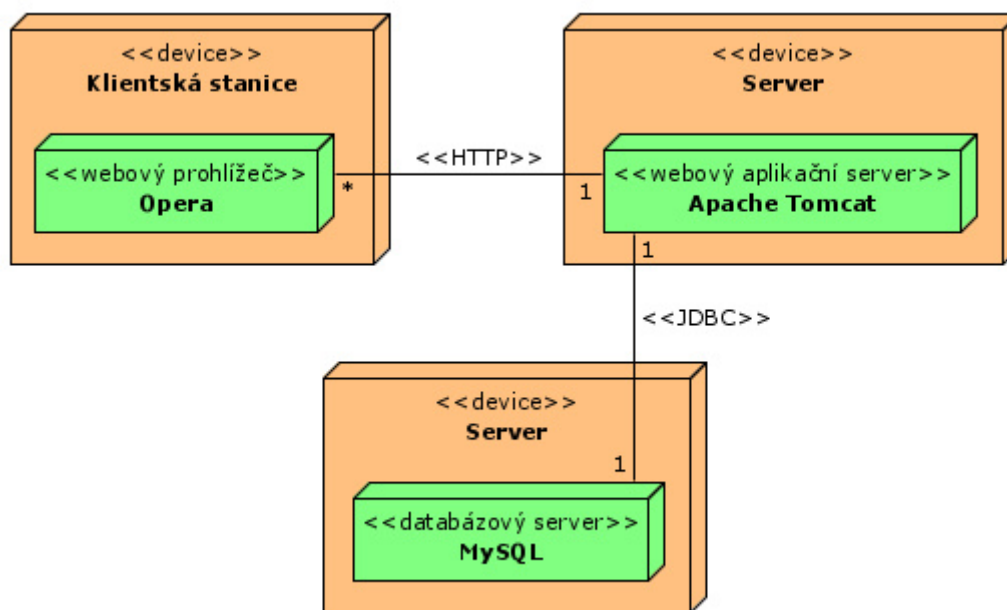
3.3 Volba technologií pro implementaci

Pro implementaci webového informačního systému máme na výběr velké množství technologií. Já vybíral platformu podle několika kritérií. Hlavním byla perspektivnost dané platformy a možnost případného uplatnění nabytých znalostí v budoucím zaměstnání. Dále také dostupnost informací, komunita, stabilita (zajištěna např. velkou firmou v pozadí), standardy apod. Má volba padla na platformu Java EE (celým názvem „Java Platform, Enterprise Edition“). Tu považuji za vhodnou volbu pro vývoj a provoz podnikových (enterprise) aplikací a informačních systémů a proto si velmi rád, prostřednictvím této práce, rozšířím znalosti z této oblasti. Díky této volbě se mi také otevřel velký výběr nejrůznějších aplikačních rámců (framework) a knihoven, které usnadňují práci, zlepšují vlastnosti, zpřístupňují nové technologie nebo umožňují přistupovat k vývoji jiným způsobem.

V této práci bude využit aplikační rámec Spring (někdy je mu dáván přívlastek „lightweight kontejner“). Ten poskytuje infrastrukturu pro řešení běžně se vyskytující aspekty implementace aplikací. Je to např. správa transakcí nebo slabá strukturovanost kódu. Spring implementuje návrhový vzor Inversion of Control (možno přeložit jako „přesun kontroly“). Ten umožňuje přesun kontroly nad vytvářením a provázáním objektů z aplikace do aplikačního rámce. Tomu je pouze dodána konfigurace např. v podobě konfiguračního XML souboru, na jehož základě již rámec automaticky objekty tvoří a provazuje. Spring také podporuje používání dobrých programátorských praktik při vývoji, jako je například „programming to interfaces“ (využívání rozhraní a následná volba implementace daného rozhraní v konfiguračním souboru). Dále Spring přináší podporu pro implementaci komponent pro přístup k datům. V tomto projektu bude využito objektově relační mapování (ORM, object-relation mapping) technologií Hibernate. Ta umožňuje mapování mezi databázovými tabulkami a Java třídami pomocí konfiguračních XML souborů. Kromě toho umožňuje získávání dat z databáze pomocí vlastního, plně objektově orientovaného dotazovacího jazyka nazvaného HQL (Hibernate Query Language). Ten má navíc velmi podobnou syntaxi se standardním SQL. Při vývoji tohoto informačního systému bude použita databáze MySQL, ale díky Hibernate může být kdykoliv v budoucnu snadno nahrazena jinou databází (např. Oracle, Microsoft SQL Server atd.).

Jako vývojový webový aplikační server jsem zvolil Apache Tomcat verze 6. Ten poskytuje implementaci technologií Java Servlet a JavaServer Pages (JSP). My zde tyto technologie nebudeme používat přímo, ale umístíme nad ně ještě rámec JavaServer Faces (JSF), který zjednodušuje vývoj uživatelského rozhraní Java EE webových aplikací. V JSF je využíván komponentově orientovaný přístup k vývoji.

Celková fyzická architektura systému je znázorněna na diagramu nasazení (viz Obrázek 3.3).



Obrázek 3.3: Diagram nasazení

K vývoji bude použito vývojové prostředí (IDE, Integrated Development Environment) Eclipse. To umožňuje pohodlný vývoj Java aplikací a také možnost dodání různých rozšíření (plugin) pro podporu vývoje s ostatními zvolenými technologiemi. Pro sestavení aplikace mimo Eclipse (např. na cílovém serveru) bude použit Apache Ant, který pracuje na základě konfiguračního XML souboru.

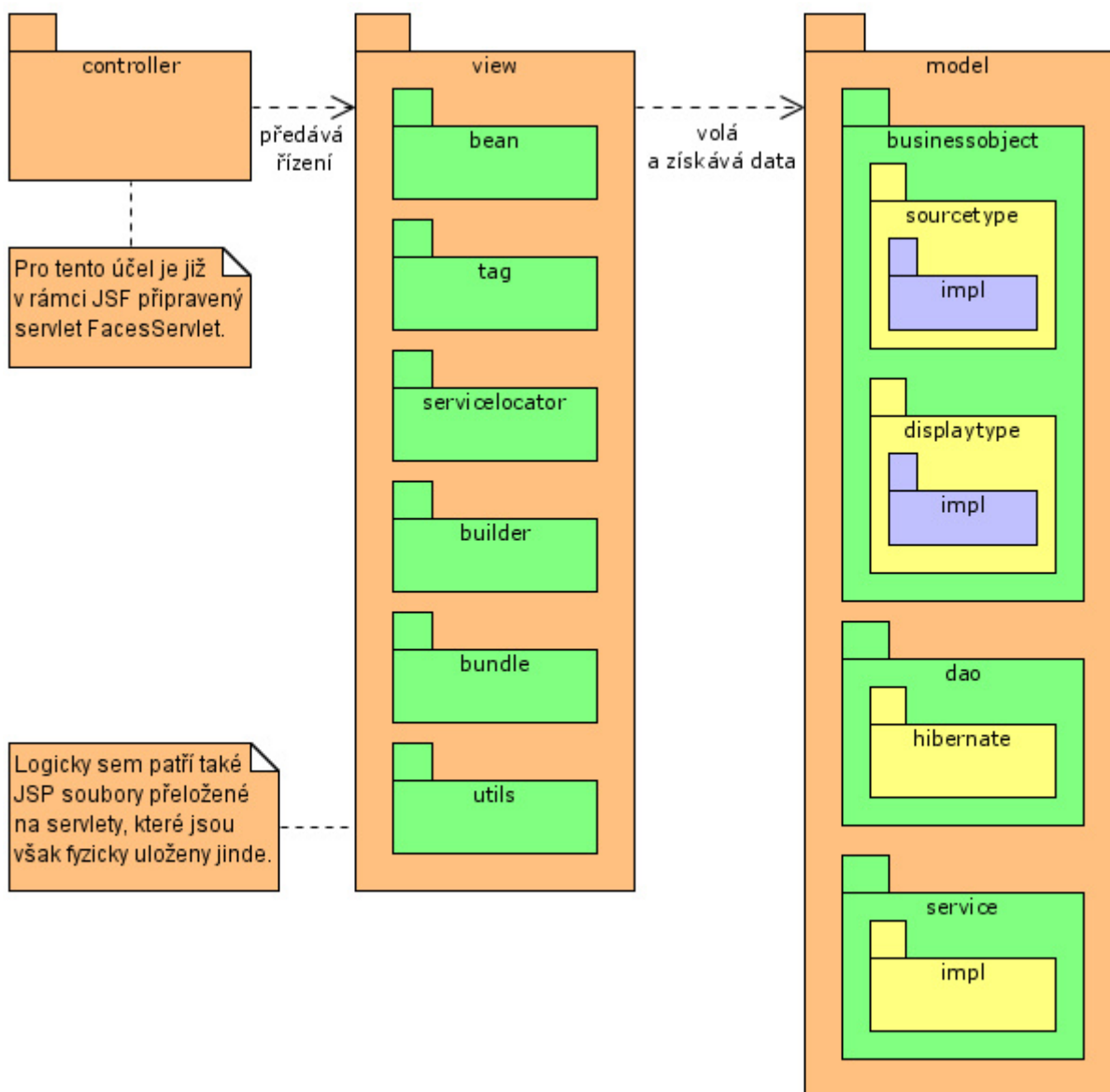
Všechny zvolené technologie jsou „Open Source“ projekty, které kolem sebe navíc mají velkou komunitu vývojářů a to znamená, že u případných problémů je velká pravděpodobnost, že už tento problém řešil někdo jiný. Na internetu je tedy bohatá zásoba informací k těmto technologiím.

3.4 Návrh architektury

Pro implementaci tohoto informačního systému jsem zvolil architekturu MVC (Model View Controller), která důsledně odděluje prezentační a aplikační logiku. V podstatě rozděluje aplikaci na tři komponenty. První je model, který obsahuje data (stav aplikace) a zprostředkovává je ostatním komponentám. Obsahuje „business“ logiku systému. Druhou je pohled (view), která slouží pro vytvoření prezentační vrstvy obvykle ve formě grafického uživatelského rozhraní (GUI).

Zprostředkovává interakci uživatele s aplikací a zobrazuje stav aplikace (data), který je dodáván komponentou model. Řadič (controller) tvoří mezivrstvu mezi prezentační (pohled) a aplikační (model) vrstvou. Jejím úkolem je zpracovat události vzniklé v prezentační vrstvě a vyvolat odpovídající akci v aplikační logice. Toto rozdělení zajišťuje nízké provázání (low coupling).

Následuje návrh struktury aplikace na úrovni balíčků s rozdělením do výše specifikovaných vrstev. Základy struktury jsou inspirovány článkem [12]. Pro aplikaci byl zvolen pracovní název „PresMon“ (začátky slov spojení „Presence Monitoring“) a takto je tedy pojmenován kořenový balíček. O úroveň níže následuje rozdělení do vrstev (model, view) a v nich už dochází na specifické dělení pro jednotlivé implementace prvků systému (viz Obrázek 3.4). Třída implementující např. jeden typ zdroje informací o prezenci osob se bude nacházet v balíčku „presmon.model.businessobject.sourcetype.impl“.



Obrázek 3.4: Diagram balíčků

Balíček „controller“ není třeba v systému vytvářet, protože funkčnost řadiče zajistí servlet FacesServlet dodaný rámcem JSF.

Balíček „view“ obsahuje třídy implementující pohled. Uvnitř je balíček „bean“ obsahující „backing beans“, což jsou objekty typu „JavaBean“ obsluhované JSF rámcem, který je při prvním požadavku zavádí v určeném rozsahu viditelnosti (scope). Tyto objekty obsahují hodnoty svázané s komponentami uživatelského rozhraní pro vstup a výstup dat a také metody obsluhující uživatelské akce. Balíček „tag“ obsahuje implementaci vlastních komponent (např. pro zobrazení presence osob), „servicelocator“ rozhraní pro volání aplikační logiky. V pohledu je také přítomen balíček „builder“, který zahrnuje třídy pro převody „backing beans“ na „business“ objekty a zpět. To je využíváno pro oddělení logiky aplikační a prezentační vrstvy. Balíček „bundle“ obsahuje zdrojové soubory zahrnující například přizpůsobené chybové hlášky nebo zdroje pro internacionalizaci aplikace. Poslední v tomto balíčku je „utils“, který obsahuje pomůcky (utility) využívané v prezentační vrstvě, například pro práci s JSF kontextem.

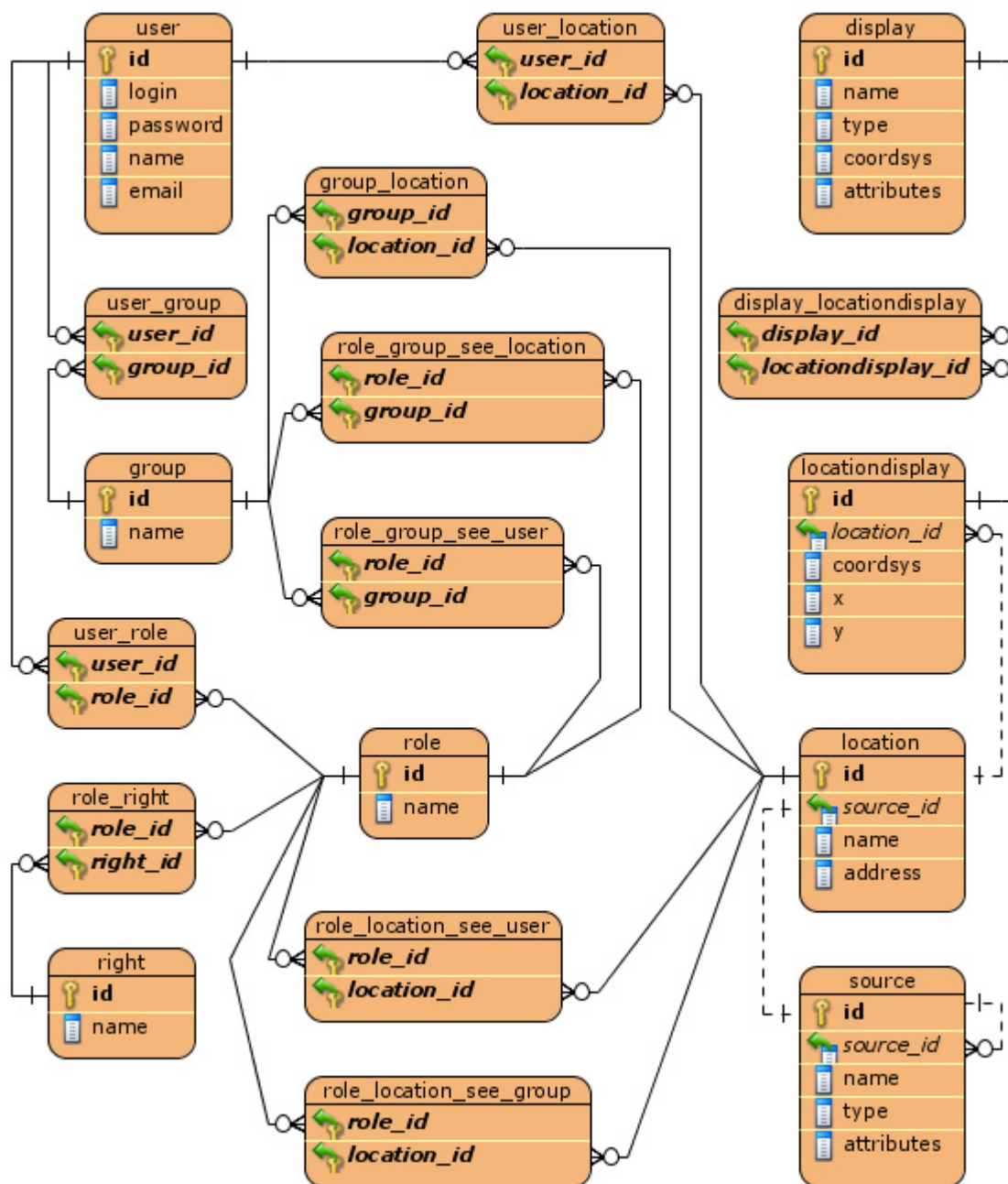
Balíček „model“ obsahuje třídy implementující jádro aplikační logiky. Balíček „businessobject“ obsahuje datové objekty namapované pomocí Hibernate na databázové tabulky a také XML soubory se samotným mapováním. Dále ještě obsahuje vnořené balíčky s rozhraním a implementací modulů pro zdroje presence osob a modulů pro zobrazení presence. V modelu je dále přítomen balíček „dao“, který obsahuje DAO (Data Access Object) objekty. Ty pracují pomocí jednotného rozhraní s datovými zdroji. Jsou zde tedy přítomna tato rozhraní a také jejich implementace např. pro práci s Hibernate. Balíček „service“ obsahuje třídy služeb, zpřístupňujících aplikační logiku objektům z prezentační vrstvy. Opět je zde přítomno jednotné rozhraní a poté implementace, kterou lze snadno zaměnit za jinou.

3.5 Databázové schéma

Následuje návrh databázových tabulek. V diagramu schématu databáze (Obrázek 3.5) jsou znázorněny jednotlivé tabulky (entity) a jejich vzájemné vztahy (relace). Ikony klíče u jednotlivých atributů označují primární klíče. Zelené šipky označují cizí klíče.

Základem jsou entity reprezentující třídy navržené v konceptuálním návrhu tříd (viz kapitola 3.2): „user“ (uživatel), „group“ (skupina), „role“ (role), „right“ (právo), „location“ (místo), „source“ (zdroj), „display“ (zobrazení) a „locationdisplay“ (zobrazení místa). Dále jsou tu vazební tabulky, jejichž jméno se skládá ze jmen propojovaných tabulek oddělených podtržítkem: „user_group“, „user_location“, „user_role“, „group_location“, „display_locationdisplay“ a „role_right“. Posledním typem jsou vazební tabulky, které navíc vyjadřují právo nahlížet na uživatele, skupiny nebo místa: „role_group_see_user“, „role_group_see_location“,

„role_location_see_user“ a „role_location_see_group“ (popis významu hodnot v těchto tabulkách naleznete v kapitole 3.6).



Obrázek 3.5: Schéma databáze

3.6 Návrh aplikace uživatelských práv

Prvním typem uživatelských práv v systému jsou „statická“ práva. Jejich kompletní seznam je uložen v databázové tabulce „right“. Tyto práva mohou být přiřazována k jednotlivým rolím a uživatelé s danou rolí poté takto přiřazenými právy disponují. Tento typ práv obsahuje následující položky (v závorkách jsou uvedeny skutečné názvy práv v databázi):

- Správa uživatelů („UsersAdministration“) – toto právo umožňuje přidávat, upravovat a mazat uživatele.
- Správa skupin („GroupsAdministration“) – uživatelé s rolí obsahující toto právo mají povoleno přidávání, úpravu a mazání skupin.
- Správa míst („LocationsAdministration“) – toto právo umožňuje svým nositelům přidávat, upravovat a mazat místa.
- Správa zobrazení („DisplaysAdministration“) – uživatelům s rolí obsahující toto právo je umožněno přidávat, upravovat a mazat zobrazení.
- Správa rolí („RolesAdministration“) – toto právo umožňuje přidávat, upravovat a mazat role.

Druhým typem práv v systému jsou práva na „nahlížení na prezenci“ u jednotlivých uživatelů, skupin nebo míst. Jsou používána pro vymezení podmnožin těchto entit, které jsou při hledání informací o prezenci danému uživateli viditelná. Tyto práva jsou přiřazována vždy mezi roli a konkrétní místo nebo skupinu. Jsou tedy v systému reprezentovány záznamy ve vazebních tabulkách (jejich názvy jsou uvedeny v závorkách) a mají následující význam:

- Právo nahlížet na uživatele („role_location_see_user“) – vazební tabulka mezi rolí a místem. Záznam v této tabulce umožňuje uživateli s touto rolí v daném místě nahlížet na prezenci uživatelů.
- Právo nahlížet na skupiny („role_location_see_group“) – vazební tabulka mezi rolí a místem. Záznam v této tabulce umožňuje uživateli s touto rolí v daném místě nahlížet na prezenci skupin.
- Právo nahlížet na uživatele („role_group_see_user“) – vazební tabulka mezi rolí a skupinou. Záznam v této tabulce umožňuje uživateli s touto rolí nahlížet na členy dané skupiny.
- Právo nahlížet na místa („role_group_see_location“) – vazební tabulka mezi rolí a skupinou. Záznam v této tabulce umožňuje uživateli s touto rolí nahlížet na místa náležící dané skupině.

Výše uvedená práva jsou v systému aplikována v procesech vyhodnocování informací o prezenci osob a to následujícím způsobem:

- Vyhodnocení prezence uživatele – při tomto procesu je nejprve nabízen uživateli seznam osob, u kterých má právo vidět alespoň jedno místo. Tento seznam je složen z osob, kterým náleží nějaké místo, na které má daný uživatel právo nahlížet na uživatele („role_location_see_user“). A také z osob, které jsou členy nějaké skupiny, na kterou má právo nahlížet na uživatele („role_group_see_user“) a současně právo nahlížet na

místa („role_group_see_location“). Po výběru požadované osoby jsou na základě stejných práv vybrány místa, kde se může tato osoba nacházet a uživatel tam má právo ji vidět. Nad touto množinou míst je poté iterován dotaz na prezenci uživatele.

- Vyhodnocení prezence skupiny – při tomto procesu musí být nejprve uživateli nabídnut seznam skupin, u kterých má právo vidět alespoň jedno místo. Tento seznam je složen ze skupin, kterým náleží nějaké místo, na které má daný uživatel právo nahlížet na skupiny („role_location_see_group“). A dále ze skupin na které má právo nahlížet na uživatele („role_group_see_user“) a dané skupiny mají alespoň jednoho člena jemuž náleží místo na které má právo nahlížet na uživatele („role_location_see_user“). Po uživatelově výběru skupiny jsou na základě stejných práv vybrány místa, kde se může někdo ze skupiny nacházet a uživatel má právo ho tam vidět. Na tyto místa jsou poté vedeny dotazy na prezenci členů vybrané skupiny.
- Vyhledání uživatelů v místě – při tomto procesu je uživateli nejprve nabídnut seznam míst, ve kterých má právo vidět alespoň jednu osobu. Tento seznam je složen z míst, u kterých má právo nahlížet na uživatele („role_location_see_user“) a zároveň tato místa náleží alespoň jednomu uživateli. Dále také z míst, která náleží nějaké skupině, na kterou má právo nahlížet na uživatele („role_group_see_user“) a současně právo nahlížet na místa („role_group_see_location“). Po výběru požadovaného místa je dotázán zdroj informací na přítomné osoby a výsledek je omezen podle stejných práv na ty, které má v daném místě uživatel právo vidět.
- Vyhledání skupin v místě – při tomto procesu musí být nejprve uživateli nabídnut seznam míst, u kterých má právo vidět nějakou skupinu. Tento seznam je složen z míst, u kterých má právo nahlížet na skupiny („role_location_see_group“). A dále také z míst u kterých má právo nahlížet na uživatele („role_location_see_user“) a tato místa náleží nějakému uživateli, který je členem skupiny, na kterou má právo nahlížet na uživatele („role_group_see_user“). Po výběru požadovaného místa je dotázán zdroj informací na přítomné skupiny a výsledek je omezen podle stejných práv na ty, které má v daném místě uživatel právo vidět.

3.7 Návrh rozhraní modulů

Požadavkem bylo dekomponovat systém na moduly pro zjišťování a zobrazení prezence se specifikovaným rozhraním. V této kapitole tedy bude uvedena specifikace rozhraní nejprve pro zdroje informací o prezenci osob a následně pro moduly zobrazení prezence.

3.7.1 Rozhraní modulů zdrojů informací

Toto rozhraní bude umístěno v balíčku „presmon.model.businessobject.sourcetype“ ve zdrojovém souboru „SourceType.java“ a bude deklarovat následující metody:

```
public interface SourceType
{
    public boolean isUserInLocation(Source source, String userId);

    public boolean isSomebodyFromSetInLocation(Source source,
        Set<String> userIds);

    public Set<String> getUsersInLocation(Source source,
        Set<String> usersCanBeSeenIds);
}
```

Metoda „isUserInLocation“ představuje dotaz na zdroj informací, jestli je konkrétní osoba přítomna v místě, které daný zdroj zastupuje. Prvním parametrem je objekt zdroje (typu „presmon.model.businessobject.Source“), který obsahuje atributy pro konfiguraci konkrétního zdroje (uloženy v textovém řetězci) a také případnou stromovou hierarchii zdrojů, které jsou mu podřízeny (pokud zdroj vyhodnocuje informace z více různých zdrojů). Druhým atributem je systémový identifikátor hledané osoby. Metoda vrací hodnotu „true“ (typu „boolean“) pokud je uživatel v daném místě přítomen a „false“ pokud není. Další metoda se nazývá „isSomebodyFromSetInLocation“ a slouží ke zjištění, jestli je některý uživatel z předané skupiny uživatelů v daném místě. Jako parametry má opět objekt zdroje a dále množinu identifikátorů uživatelů, kteří se mají prozkoumat. Návrátová hodnota typu „boolean“ nabývá hodnoty „true“, pokud je někdo v množině uživatelů v místě přítomen. V opačném případě nabývá hodnoty „false“. Poslední metodou je „getUsersInLocation“, která slouží ke zjištění všech uživatelů přítomných v daném místě. Jako parametry má objekt zdroje a dále množinu identifikátorů uživatelů, u kterých je prozkoumávána přítomnost v daném místě. Metoda vrací množinu řetězců, které představují systémové identifikátory nalezených uživatelů v místě.

Stěžejní je pro každý modul zdroje informací první metoda. Další dvě se dají implementovat pouze pomocí volání první metody v cyklu. Tyto dvě metody jsou v rozhraní přítomny proto, že některé zdroje mohou v jejich těle implementovat značné optimalizace pro tyto dotazy (např. eliminace opakovaného navazování spojení se vzdáleným zařízením).

Pokud zdroj používá k identifikaci osob vlastní identifikátory, tak si musí v rámci zachování jednotnosti rozhraní spravovat informace o mapování ve vlastní režii.

3.7.2 Rozhraní modulů zobrazení

Toto rozhraní bude umístěno v balíčku „presmon.model.businessobject.displaytype“ ve zdrojovém souboru „DisplayType.java“ a bude deklarovat následující metodu:

```
public interface DisplayType
{
    public void display(ResponseWriter writer, String displayAttributes,
        List<String[]> locationsDisplays) throws Exception;
}
```

Metoda „display“ slouží k vytvoření grafického výstupu se zobrazenými místy ve zvoleném zobrazení. Prvním parametrem je objekt typu „javax.faces.context.ResponseWriter“, do kterého je vytvořený výstup zapsán. Druhým parametrem je řetězec obsahující atributy zvoleného zobrazení. Posledním atributem je seznam informací o místech, které se mají graficky zobrazit. Tento seznam obsahuje souřadnici x, souřadnici y, název místa a jeho adresu.

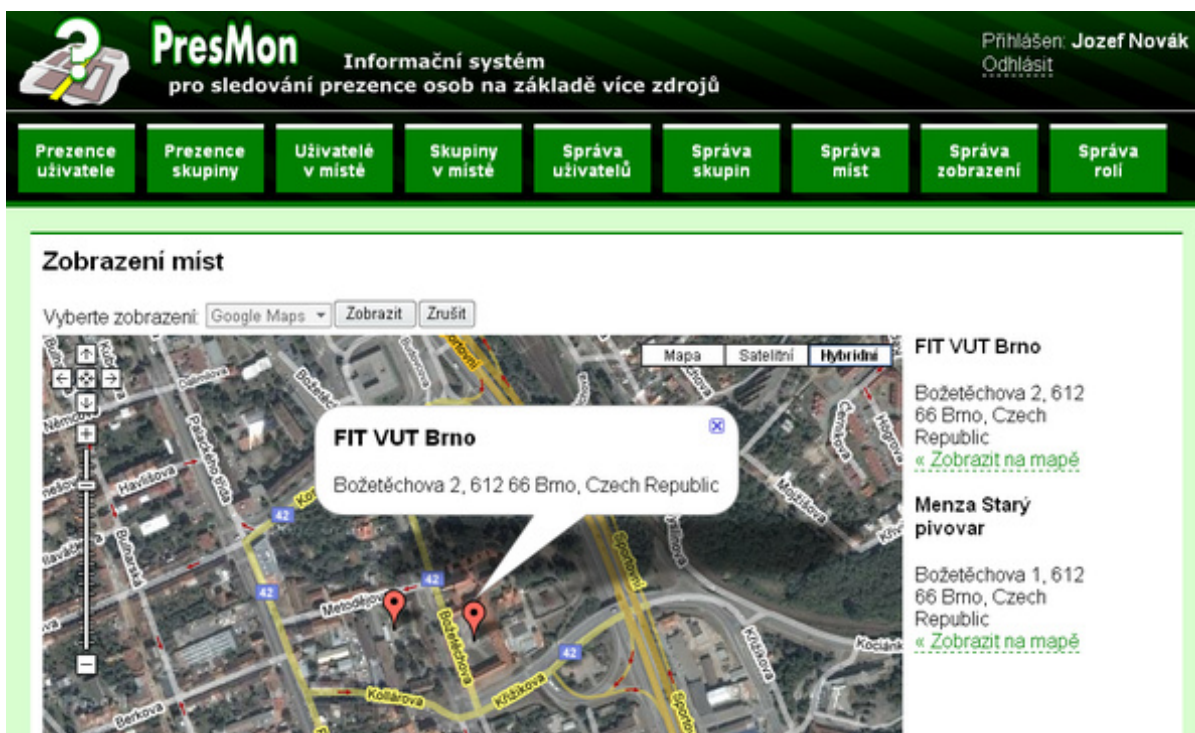
3.8 Návrh uživatelského rozhraní

Na každé stránce v aplikaci by měla být přítomna přehledná nabídka (menu). Ta bude obsahovat odkazy na funkce systému, které má daný uživatel právo používat. Kompletní nabídka bude obsahovat následující:

- **Prezence uživatele** – umožňuje vyhledání prezence konkrétní osoby. Nejprve uživatel podle jména zvolí, koho chce hledat a systém mu poté zobrazí (textově) nalezenou prezenci. Pokud je to u daného místa možné, tak systém nabídne uživateli možnost zobrazení prezence graficky.
- **Prezence skupiny** – slouží k vyhledání prezence osob, náležících do nějaké skupiny. Nejprve uživatel podle názvu zvolí skupinu, jejíž prezenci chce vyhodnotit. Systém danou skupinu osob prozkoumá a zobrazí seznam míst, kde všude jsou členové skupiny přítomni. Pokud jsou u míst nadefinována zobrazení, tak má uživatel možnost prezenci zobrazit graficky.
- **Uživatelé v místě** – umožňuje zjištění seznamu osob, přítomných v daný okamžik na vybraném místě. Uživatel nejprve vybere místo, které chce prozkoumat a systém mu zobrazí nalezenou množinu osob.
- **Skupiny v místě** – slouží k vyhledání skupin, jejichž členové jsou přítomni na zvoleném místě. Uživatel nejprve zvolí prohledávané místo a systém mu pak zobrazí seznam nalezených skupin.
- **Správa uživatelů** – umožňuje vkládání, úpravu a mazání uživatelů. U jednotlivých uživatelů je také možnost spravovat místa, která danému uživateli náleží.
- **Správa skupin** – slouží ke vkládání, úpravě a mazání skupin. U jednotlivých skupin je také možnost spravovat členství uživatelů a místa, které dané skupině náleží.

- Správa míst – umožňuje vkládání, úpravu a mazání míst. U každého je také možnost spravovat hierarchii zdrojů informací o prezenci.
- Správa zobrazení – slouží ke vkládání, úpravě a mazání zobrazení. U každého je také možnost zprávy zobrazení konkrétních míst.
- Správa rolí – umožňuje vkládání, úpravu a mazání rolí. U jednotlivých rolí je také možnost spravovat jejich práva.

Na každé stránce musí být také u nepřihlášených uživatelů přítomen odkaz vedoucí na přihlašovací formulář. U přihlášených uživatelů zde pak bude místo tohoto odkazu vypsáno jejich jméno a odkaz na odhlášení ze systému. V celém systému by také mělo být zachováno konzistentní záhlaví a zápatí stránek. Také vzhled všech formulářů, výpisů a hlášek by se měl držet jednotného stylu.



Obrázek 3.6: Příklad uživatelského rozhraní

3.9 Lokalizace

U výsledného systému by měla být možnost snadné lokalizace do jiných jazyků. Ideální je podpora textového souboru, ze kterého jsou načítány všechny texty a hlášky. Není potřeba interaktivního přepínání lokalizace za běhu systému, ale stačí možnost nastavení během instalace.

4 Implementace

Pro implementaci bylo použito prostředí a technologie navrhnuté v kapitole 3.3. Veškeré zdrojové kódy včetně komentářů byly psány v anglickém jazyce.

4.1 Programová dokumentace

Zdrojové kódy byly pečlivě komentovány pro možnost automatického vygenerování programové dokumentace na základě těchto komentářů. Všechny významné veřejné metody jsou opatřeny dokumentačním komentářem s popisem činnosti dané metody, parametrů a návratové hodnoty.

Programová dokumentace byla vygenerována nástrojem „javadoc“ do adresáře „doc“. Dokumentace je ve formátu HTML a vstupním souborem je „index.html“.

4.2 Moduly

V této kapitole jsou popsány pokyny pro implementaci nových modulů zobrazení a zdrojů informací o prezenci.

Modul nového zdroje informací o prezenci musí být umístěn v balíčku „presmon.model.businessobject.sourcetype.impl“. Jeho název musí končit řetězcem „ST“ (např. „TestST.java“). Třída modulu musí implementovat rozhraní „presmon.model.businessobject.sourcetype.SourceType“ specifikované v kapitole 3.7.1. Stěžejní je implementace metody „isUserInLocation“. Zbylé dvě metody mohou být implementovány s její pomocí (lze převzít např. z modulu „MergeAndST“).

Modul nového zobrazení musí být umístěn v balíčku „presmon.model.businessobject.displaytype.impl“. Jeho název musí končit řetězcem „DT“ (např. „TestDT.java“). Třída modulu musí implementovat rozhraní „presmon.model.businessobject.displaytype.DisplayType“ specifikované v kapitole 3.7.2.

4.2.1 Implementované ukázkové moduly

Jako ukázkové moduly zdrojů informací o prezenci osob byly implementovány:

- MergeOrST – typ odvozeného zdroje. Slučuje informace získané od více zdrojů (primárních i odvozených) pomocí logického součtu. Při nalezení prvního zdroje, který mu vrátí kladnou odpověď ukončí zpracování a odpoví kladně. Pokud mu všechny zdroje odpoví záporně, pak odpoví také záporně. V atributu zdroje lze nastavit váhy podřizovaných zdrojů (číselné hodnoty oddělené středníky, které jsou postupně aplikovány na zdroje seřazené vzestupně podle

„id“). Pořadí zpracování je poté dáno vahou zdroje (začíná se od zdrojů s nejvyšší vahou postupně až po zdroje s nejnižší vahou).

- MergeAndST – typ odvozeného zdroje. Slučuje informace získané od více zdrojů (primárních i odvozených) pomocí logického součinu. Při nalezení prvního zdroje, který mu vrátí zápornou odpověď ukončí zpracování a odpoví záporně. Pokud mu všechny zdroje odpoví kladně, pak odpoví také kladně.
- XMPPJabberST – typ primárního zdroje, který zjišťuje prezenci pomocí protokolu XMPP/Jabber. V atributu zdroje musejí být následující údaje (oddělené středníky): adresa serveru, přihlašovací jméno a heslo. Pomocí těchto údajů se modul připojí k serveru (uvedený uživatelský účet již musí být vytvořen). Mapování mezi id uživatelů v systému a XMPP adresami je prozatím zachyceno pevně v kódu modulu (v konstruktoru). V další verzi by se mohlo dodělat načítání mapování např. z externího souboru. Pro implementaci tohoto modulu byla využita „open source“ knihovna Smack API (<http://www.igniterealtime.org/projects/smack/index.jsp>).
- TestPresentST – typ primárního zdroje používaný při testování. Na všechny dotazy odpovídá kladně, tedy že je osoba v místě přítomna.
- TestNotPresentST – typ primárního zdroje používaný při testování. Na všechny dotazy odpovídá záporně, tedy že osoba není v místě přítomna.
- TestRandomST – typ primárního zdroje používaný při testování. Na všechny dotazy odpovídá náhodně.

Jako ukázkové moduly pro grafické zobrazení prezence byly implementovány:

- ImageDT – zobrazení na mapách nebo plánech dodaných jako rastrové obrázky. Nastavení se provádí pomocí atributu zobrazení, které musí obsahovat následující položky (oddělené středníky): URL (Uniform Resource Locator) obrázku s mapou nebo plánem, šířka tohoto obrázku (všechny rozměry jsou uváděny v obrazových bodech), výška tohoto obrázku, URL obrázku k označení nalezených míst, šířka označovacího obrázku, výška označovacího obrázku. Souřadným systémem tohoto zobrazení jsou obrazové body (pixels) určující vzdálenost od levého horního rohu obrázku mapy nebo plánu.
- GoogleMapsDT – zobrazení na mapách Google Maps. Tento modul využívá Google Maps API, které umožňuje používání těchto map ve vlastních webových stránkách [13]. Jejich použití je však vázáno na URL stránky. Proto jsem nejprve na internet umístil jednoduchou stránku (vytvořena v PHP, zdrojový kód je přiložen v adresáři „dist“), která na základě předaných parametrů zobrazí odpovídající mapu. Náš modul importuje tuto stránku s odpovídajícími parametry, které nesou informace o požadovaných místech k zobrazení.

Adresa této zprostředkovávací stránky je uložena v atributu zobrazení. Souřadným systémem využívaným těmito mapami je WGS84 (World Geodetic System, revision 1984).

4.3 Tvorba lokalizace

Zdrojové soubory lokalizačních souborů jsou umístěny v balíčku „src.presmon.view.bundle“. Jako předloha pro novou lokalizaci poslouží soubor se základní lokalizací („Messages_cz.src“). Vytvořte kopii a řetězec „cz“ nahraďte zkratkou nové lokalizace. Poté otevřete soubor v libovolném textovém editoru a přepište hlášky (všechny řetězce vpravo od znaménka „=“). Pokud nesestavujete aplikaci pomocí Apache Ant (např. přímo v Eclipse), tak je nutné ještě po kompilaci provést přeložení lokalizací příkazem „ant msgs“.

4.4 Instalace a konfigurace

Programové vybavení, které je předpokladem instalace systému (jsou uvedeny programy a verze použité při vývoji, ale systém může být schopen bezproblémového provozu i na jiných verzích, databázi nebo aplikačním serveru):

- Java 2 Standard Edition Runtime Environment (JRE) verze 1.6.0_04
- Apache Tomcat verze 6.0.16
- MySQL verze 5.0.51
- Apache Ant verze 1.7.0

Pro úplnost dodávám, že vše bylo provozováno na operačním systému MS Windows.

Nejprve je nutné vytvořit databázi – skript je přiložen v souboru „dist/presmon.sql“. Poté je nutné aplikaci nakonfigurovat pro vaše běhové prostředí:

- Konfigurace databáze – v souboru „web\WEB-INF\applicationContext.xml“ naleznete definici beanu „dataSource“. V ní je konfigurace připojení k databázi (driver, url, přihlašovací jméno a heslo), které si nastavte pro připojení k vaší databázi.
- Konfigurace lokalizace – v souboru „web\WEB-INF\faces-config.xml“ naleznete tag „application“, v něm „locale-config“ a do těla tagu „default-locale“ vepište požadovanou hodnotu (tvorba lokalizace je popsána v kapitole 4.3).
- Konfigurace logování – v souboru „web\WEB-INF\classes\log4j.properties“ si nastavte parametry dle libosti.

Následuje sestavení (build) aplikace příkazem „ant“ (z adresáře, kde je umístěn soubor „build.xml“). Vznikne soubor „dist\presmon.war“, který lze přímo překopírovat do adresáře webových aplikací serveru „Apache Tomcat“ (standardně adresář se jménem „webapps“). Při spuštění serveru je pak aplikace zavedena automaticky. Další možností je zavedení aplikace pomocí aplikace „manager“, která je standardně přítomna v instalaci tohoto serveru.

4.5 Testování

Testování probíhalo v různých etapách vývoje. Primárně byla používána metoda „white box“, tedy testování na základě znalosti implementace. Většinou byla nejprve vytvořena nějaká testovací data a sledovalo se, jestli s nimi sledovaná akce proběhne správně (jsou zobrazeny správné informace), případně že se správně promítne do databáze.

4.5.1 Testování GUI

Grafické uživatelské rozhraní systému je tvořeno XHTML kódem generovaným automaticky JSF rámcem. Tento kód je formálně správný a bylo k němu pouze potřeba dopsat kaskádové styly (CSS) pro definici vzhledu. Různé webové prohlížeče však některé styly interpretují rozdílně a proto bylo nutné testování ve více prohlížečích. Testováno bylo chování v následujících prohlížečích:

- Opera 9.27 (MS Windows XP)
- Firefox 2.0.0.14 (MS Windows XP)
- MS Internet Explorer 6.0 (MS Windows XP)
- MS Internet Explorer 7.0 (MS Windows XP)

Kaskádové styly byly upraveny tak, aby se stránky ve výše uvedených prohlížečích zobrazovali správně (v zobrazení existují drobné rozdíly, které však nenarušují chování aplikace ani navržený vzhled).

5 Závěr

Vývoj projektu hodnotím jako velmi dobrý. Všechny úkoly vyplývající ze zadání byly splněny. Nejprve jsem analyzoval způsoby propagace prezence osob v různých systémech a prostudoval problematiku sledování prezence z více zdrojů. Následně jsem zpracoval analýzu a návrh informačního systému, který se nedívá na prezenci osoby pouze jako na zobrazení stavu partnera pro komunikaci. V tomto systému jsou definována různá místa a následně se zkoumá přítomnost osob v těchto místech. Zdrojů informací o prezenci může být u každého místa více a mohou mít různé vlastnosti. Je zde i možnost tvorby fiktivních zdrojů, které mohou určitým způsobem zpracovávat informace z jiného zdroje nebo zdrojů. U vyhodnocené prezence je navíc možnost grafického zobrazení na mapě nebo plánu. Do systému je možno podle specifikovaného rozhraní doimplementovat další moduly různých zdrojů a také zobrazení informací o prezenci. Vyvinutý systém je plně funkční a lze si představit jeho nasazení např. v nějakých organizacích.

Z hlediska dalšího vývoje této práce vidím několik možných směrů rozšíření. Jednou možností by mohlo být zobrazování rozšiřujících kontextových informací získaných ze zdrojů informací o prezenci (např. stav u IM klienta nebo název položky z elektronického kalendáře). Zajímavé by také mohlo být udržování historie prezence uživatelů. Užitečným rozšířením systému by mohlo být také zavedení vyhledávání uživatelů, míst nebo skupin podle různých parametrů.

Literatura

- [1] Day, M., Rosenberg, J., Sutano, H.: *A Model for Presence and Instant Messaging*. RFC 2778, [online], únor 2000, [cit. 11.12.2007].
URL <<http://rfc.net/rfc2778.html>>
- [2] Rodriguez, L.: *MPOP and presence by observation*. [online], 22.10.2006, [cit. 11.12.2007].
URL <<http://presence-on-the-move.blogspot.com/2006/10/mpop-and-presence-by-observation.html>>
- [3] Shutko, A. aj.: *OSCAR (ICQ v7/v8/v9) protocol documentation*. [online], únor 2003, rev. 07.02.2005, [cit. 10.12.2007].
URL <<http://iserverd.khstu.ru/oscar/>>
- [4] Mintz, M.: *MSN Messenger Protocol*. [online], 10.03.2003, rev. 19.12.2003, [cit. 12.12.2007].
URL <<http://www.hypothetic.org/docs/msn/general/overview.php>>
- [5] Kirk, J.: *Microsoft to clamp down on spam over IM*. [online], 23.11.2007, [cit. 12.12.2007].
URL <<http://www.computerworlduk.com/technology/security-products/prevention/news/index.cfm?RSS&NewsId=6359>>
- [6] Henč, M.: *Microsoft Unified Communications*. [online], 14.06.2007, [cit. 12.12.2007].
URL <http://www.microsoft.com/cze/technet/clanky/ms_uc.aspx>
- [7] Saint-Andre, P. editor: *Extensible Messaging and Presence Protocol (XMPP): Core*. RFC 3920, Jabber Software Foundation, [online], říjen 2004, [cit. 11.12.2007].
URL <<http://rfc.net/rfc3920.html>>
- [8] Saint-Andre, P. editor: *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence*. RFC 3921, Jabber Software Foundation, [online], říjen 2004, [cit. 11.12.2007].
URL <<http://rfc.net/rfc3921.html>>
- [9] XMPP Standards Foundation: *XMPP*. [online], 1999, [cit. 29.10.2007].
URL <<http://www.xmpp.org>>
- [10] Eisenstadt, M. aj.: *BuddySpace*. [online], říjen 2002, rev. 01.05.2007, [cit. 11.12.2007].
URL <<http://buddyspace.sourceforge.net>>
- [11] Hrubý, P.: *SIP (V) – Instant messaging*. [online], 10.07.2003, [cit. 11.12.2007].
URL <<http://www.isdn.cz/clanek.php?cid=5037>>
- [12] Shen, D. Y.: *Put JSF to work*. [online], 19.07.2004, [cit. 02.05.2008].
URL <<http://www.javaworld.com/javaworld/jw-07-2004/jw-0719-jsf.html>>
- [13] Václavek, P.: *Návod na zabudování Google maps do vašich stránek*. [online], 28.08.2007, [cit. 05.05.2008].
URL <<http://petr.vaclavek.com/article/471/Navod-zabudovani-Google-maps-do-stranek/>>

Seznam obrázků

Obrázek 2.1: Předávání informace o prezenci	4
Obrázek 2.2: Varianty uživatelů typu „watcher“	5
Obrázek 2.3: Architektura systému pro zobrazení přítomnosti v protokolu SIP	12
Obrázek 3.1: Diagram případů použití	15
Obrázek 3.2: Konceptuální diagram tříd.....	30
Obrázek 3.3: Diagram nasazení.....	32
Obrázek 3.4: Diagram balíčků	33
Obrázek 3.5: Schéma databáze	35
Obrázek 3.6: Příklad uživatelského rozhraní	40

Seznam příloh

Příloha 1. Nosič CD-ROM, který obsahuje následující adresářovou strukturu:

- *zprava* – zdrojový tvar této technické zprávy pro tisk (formát PDF)
- *presMon* – projekt (z prostředí Eclipse) s kompletními zdrojovými kódy informačního systému (obsahuje také soubor „readme.txt“ s návodem na instalaci a konfiguraci)
- *doc* – programová dokumentace (JavaDoc).