MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Systém pro správu rezervací a zdrojů kolaborativních systémů

BAKALÁŘSKÁ PRÁCE

Markéta Trachtová

Brno, podzim 2011
login: xtracht
UČO: 255498

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracovala samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používala nebo z nich čerpala, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Markéta Trachtová

Vedoucí práce: RNDr. Michal Procházka

Poděkování

Ráda bych poděkovala panu RNDr. Michalu Procházkovi za trpělivé vedení bakalářské práce a jeho cenné rady poskytované v průběhu práce.

Shrnutí

Cílem této práce je najít nebo vytvořit softwarové řešení pro plánování schůzek v kolaborativních systémech. Plánování by mělo zahrnovat i možnost rezervace zdrojů spojených s konkrétním kolaborativním sezením. Přístup k rezervačnímu systému by měl být realizován přes federaci identit. Práce poukazuje na nedostatky existujících plánovacích systémů a klade si za cíl navrhnout a implementovat základ systému nového. Vývoj je dokumentován pomocí nástrojů jazyka UML, samotné implementační práce jsou provedeny v jazyce PHP.

Klíčová slova

Kolaborativní systémy, vzdálená spolupráce, webová aplikace, videokonference, eduID.cz, Shibboleth, OAuth.

Obsah

1 **Úvod** . . . 2
2 **Stávající podpora vzdálené spolupráce** . . . 4
2.1 *Federace identit* . . . 4
2.1.1 Obecně o federacích . . . 4
2.1.2 Shibboleth . . . 5
2.1.3 EduID a OpenID . . . 7
2.2 *Síťová komunikace a současné kolaborativní systémy* . . . 7
2.2.1 Současné kolaborativní systémy . . . 8
2.2.1.1 IM klienti . . . 8
2.2.1.2 Klienti pro elektronickou poštu . . . 9
2.2.1.3 EVO . . . 10
2.2.1.4 Adobe Connect . . . 10
2.2.2 Souhrn . . . 11
3 **Návrh nového systému** . . . 13
3.1 *Požadavky na systém* . . . 13
3.2 *Charakteristika nového systému* . . . 13
3.3 *Funkční model systému* . . . 15
3.3.1 Správa uživatelů . . . 15
3.3.1.1 Správa uživatelských dat . . . 16
3.3.1.2 Rozdělení uživatelů podle privilegií . . . 16
3.3.2 Správa aktualit . . . 17
3.3.3 Správa externích zdrojů . . . 17
3.3.4 Správa konferencí . . . 18
3.4 *Model grafického rozhraní* . . . 21
4 **Implementace** . . . 23
4.1 *Použité technologie* . . . 23
4.2 *Struktura implementace* . . . 23
4.2.1 Prezenční vrstva aplikace . . . 24
4.2.2 Datová vrstva aplikace . . . 26
4.2.3 Řídící vrstva aplikace . . . 28
4.2.4 Podpůrné skripty . . . 28
4.2.5 Zajímavé implementační prvky . . . 28
4.2.5.1 Přístup uživatelů . . . 28
4.2.5.2 Google Kalendář . . . 29
4.3 *Závěr* . . . 32
Literatura . . . 34
A **Obsah přiloženého CD** . . . 38
B **Poznámky k vývoji a instalaci systému** . . . 41

1 Úvod

Spolupráce lidí je stále nezanedbatelnou součástí každého vznikajícího projektu. Klíčovým prvkem samotné spolupráce je potom jistě komunikace. Aktéři musí mít příležitost ke konzultacím s ohledem na svá rozdílná zaměření, zkušenosti nebo geografické vzdálenosti. Fyzické setkávání často nebývá možné. Brání mu jednak množství času potřebné na překonání mnohdy i velkých vzdáleností, a jednak vysoké finanční náklady s takovýmto cestováním spojené. V úvahu v tomto případě, kromě telefonních hovorů, připadají prostředky poskytované počítačovými sítěmi.

Aplikace zaměřené na podporu vzdálené spolupráce lidí s využitím Internetových služeb jsou známé jako kolaborativní systémy. Setkání lidí realizované s jejich využitím je označováno jako kolaborativní setkání, případně kolaborativní schůzka. Mezi kolaborativní systémy se řadí například aplikace sloužící k zasílání textových zpráv v reálném čase (takzvaný „instant messaging“ klienti) nebo stále více oblíbené sociální sítě typu Facebook [15]. Rozhovor samotný může probíhat jako přenos audia, videa či textových zpráv v reálném čase. Častá je kombinace všech zmíněných možností.

Ačkoli jsou kolaborativní setkání velmi často pořádána neplánovaně (tak zvaně „ad-hoc“) je vhodné mít dispozici i možnost k jejich naplánování. Základním aspektem plánování je vždy dohoda týkající se typu komunikace. Od ní se pak odvíjí další prvky jako je určení termínu konání schůzky nebo složení a počet účastníků. Jedním z klíčových prvků, jehož dostupnost je potřeba ujednat s předstihem, jsou externí zařízení nezbytná pro některé typy kolaborativních schůzek.

Cílem této práce bylo najít nebo vytvořit softwarové řešení právě pro plánování schůzek v kolaborativních systémech. Shrnutí konkrétních problémů, k jejichž řešení má vyvinutá aplikace přispět, je k dispozici v Sekci 3.1. Pozornost je věnována také výše zmíněnému speciálnímu vybavení, které bývá pro realizaci vzdálené spolupráce potřebné.

Výsledná aplikace má za úkol řešit hlavní z problémů pořádání konferencí v kolaborativních systémech a současně umožnit využití externích zařízení potřebných pro realizaci komunikace. Důraz byl kladen na využití federace identit jako prvku pro sjednocení autentizace uživatelů přistupujících k systému.

Více o federacích identit, stejně jako souhrn hlavních komplikací souvisejících s plánováním vzdálené spolupráce, najdete v Sekci 2 této práce. Zde se nachází i popis několika kolaborativních systémů, které obsahují řešení některých z představených problémů. Sekce 3 je věnována návrhu a představení nového systému, jenž by měl mimo jiné poskytovat prostředky pro řešení nalezených problémů. Sekce 4 zahrnuje popis implementace systému a k ní využitých technologií. V závěru práce jsou shrnuty poznatky získané při návrhu a tvorbě systému stejně jako možnosti pro jeho rozšíření do budoucna.

2 Stávající podpora vzdálené spolupráce

Tato kapitola obsahuje výčet problémů objevujících se při přípravě na kolaborativní schůzky. Nachází se zde taktéž popis možností, které pro řešení těchto komplikací poskytují vybrané současné systémy pro podporu vzdálené spolupráce. Závěr je věnován souhrnu nejdůležitějších informací zmíněných v této kapitole.

Při přípravě na vzdálenou spolupráci je nutné nejprve zvolit způsob realizace komunikace a to pro plánování i průběh samotného jednání. Aplikace, které se organizátor rozhodne využít, musí být přístupné všem potenciálním účastníkům. Stejně tak i případné speciální vybavení potřebné k uskutečnění zvoleného typu komunikace. Zdánlivě jednoduchou, leč často nejobtížnější věcí, je zajistit, aby všichni potenciální účastníci věděli o konání kolaborativní schůzky. Mezi hlavní komplikace při plánování kolaborativního setkávání patří:

- **Nejednotná identifikace uživatelů v různých webových aplikacích** – rozdílné aplikace vyžadují rozdílné identifikační údaje. Některé aplikace nejsou dostupné širší veřejnosti.
- **Omezená dostupnost specializovaného vybavení nutného pro realizaci komunikace** – u většího počtu účastníků je ošetření tohoto problému složité. Využití a typ vybavení použitého ke komunikaci bývá nezřídka klíčové pro realizaci vzdálené spolupráce.

Částečné řešení zmíněných požadavků pro plánování kolaborativních setkání poskytují současné kolaborativní systémy. V jejich případě se většinou jedná o separátní aplikace, mezi kterými přetrvávají rozdíly v identifikaci uživatelů. Možné řešení identifikační nejednotnosti představují federace identit. Ty pro své uživatele definují jednotný tvar identifikačních údajů. Některá uživatelská data mohou být dále využita pro autorizaci a autentizaci vůči službám, jež jsou také zapojeny do federace identit. Ty svým uživatelům dovolují používat jediné identifikační údaje pro autentizaci k různým službám. Jde tedy o jednu z možností podpory vzdálené spolupráce. Více o federacích identit se dočtete dále v textu.

2.1 Federace identit

2.1.1 Obecně o federacích

Aplikace umožňující vzdálenou spolupráci logicky vyžadují vzájemnou identifikaci svých uživatelů. Většina systémů toto řeší požadavkem na registraci při prvním přístupu každého, kdo chce s programem pracovat. Nová identita uživatele vytvořená při registraci slouží pouze pro potřebu dané aplikace. Uživatel

s registrací získá přístupové údaje, které by měly být unikátní a kvůli bezpečnosti patřičně složité. Samozřejmě si je jejich majitel musí pamatovat a dostatečně často měnit. Díky tomuto principu se každému, kdo používá kolaborativní systémy nebo jiné z kvanta aplikací s nutností autentizace uživatele, hromadí přístupové údaje k zapamatování.

Alternativu k tomuto principu představují federace identit. Jejich cílem je poskytnout uživatelům možnost využít jedny přihlašovací údaje, tedy i jednu identitu, pro autentizaci k více službám. Zjednodušeně řečeno se uživatel zaregistruje (získá unikátní jméno a heslo) pouze jednou a na základě přihlašovacích údajů získaných při registraci mu bude zpřístupněno více služeb.

Princip federací stojí na faktu, že každý uživatel je registrován v rámci jedné jediné organizace. Tato je pro něj „domovskou“ a musí být členem federace identit. Všechny domovské instituce spravují data o svých uživatelích a poskytují je aplikacím zapojeným do federační infrastruktury. Uživatelé se díky tomu mohou dostat ke službám cizích systémů, které nejsou s jejich domovskou institucí přímo propojeny. Do postupu předávání identifikačních údajů přitom uživatel zasahuje pouze v okamžiku zadávání autentizačních údajů, kterými se prokazuje vůči své domovské organizaci.

Všechny domovské instituce mají ustanoveno jednotné rozhraní pro předávání klientských informací tak zvaného *poskytovatele identit* (Identity provider, IdP). Koncové služby využívající nabídnutá data bývají označovány jako *poskytovatelé služeb* (Service Providers, SP) a získané informace používají k autorizaci uživatelů.

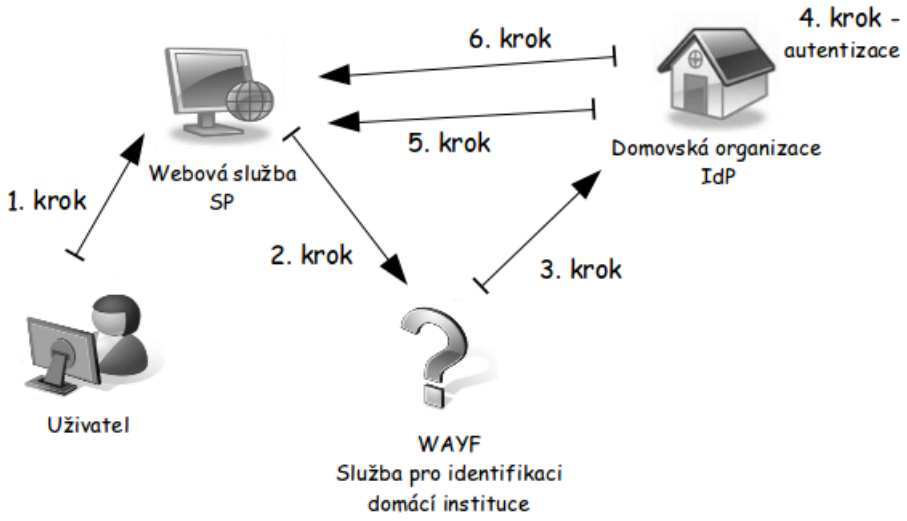
Důležitým aspektem je také formát předávaných uživatelských dat v podobě sady atributů. Rovněž je potřeba určit, které aspekty popisující uživatele jsou v rámci dané federace podstatné, tedy definovat tak zvané *schéma atributů*. Atributy mohou být například jméno, e-mail, domovská organizace nebo vztah uživatele vůči organizaci. Příkladem schématu atributů je objektová třída `eduPerson` [12] sloužící pro potřeby identifikace členů vzdělávacích institucí.

2.1.2 Shibboleth

Jedním z celosvětově známých systémů implementujících federační architekturu je Shibboleth [42]. Jedná se o volně dostupný middleware řešící federativní autentizaci a autorizaci (Authentication and Authorization Infrastructure, AAI) [1] vyvíjený americkým konsorciem Internet2 [25]. Technologicky je založen na Security Assertion Markup Language (SAML) [46], standardu pro tvorbu a výměnu bezpečnostních údajů mezi komunikujícími stranami v prostředí počítačových sítí.

Postup výměny informací je schematicky znázorněn na Obrázku 2.1 (převzato z [29]). S prvním přístupem uživatele k poskytovateli služeb

chráněnému systémem Shibboleth (Obrázek 2.1, 1. krok) není uživatel autentizován a poskytovatel služby (SP) netuší, na koho se obrátit se žádostí o jeho identifikaci. Pokud SP nemá implementované vlastní mechanismy na určení patřičného poskytovatele identit (IdP), zavolá službu zvanou WAYF („Where Are You From“) [45] (Obrázek 2.1, 2. krok). Ta má za úkol jedinou věc – zjistit, který z federačních členů je domovskou organizací přistupujícího uživatele a přesměrovat ho na něj (Obrázek 2.1, 3. krok). Zde se uživatel autentizuje pomocí svých běžných přihlašovacích údajů v rámci daného IdP (Obrázek 2.1, 4. krok). Následně je přesměrován zpět na původní chráněnou službu (Obráze 2.1, 5. krok) s informací o úspěšné autentizaci. Webová aplikace nyní může požádat uživatelovu instituci o poskytnutí všech potřebných atributů. Poskytovatel identity přihlášeného zašle vyžádaná data ve formě XML (Extensible Markupt Language) [47] dokumentu odpovídajícího standardu SAML (Obrázek 2.1, 6. krok). Na základě získaných informací se poskytovatel služby rozhodne, zda uživateli přístup povolí, nebo zamítne.



Obrázek 2.1: Postup výměny informací

Kompletní proceduru výměny informací, jak je popsána výše, uživatel absolvuje pouze při prvním přístupu k chráněnému zdroji v rámci jedné seance[1]. Při dalších přístupech ke službám je uživatel již implicitně autentizován, tudíž nemusí opakovaně zadávat své přihlašovací údaje.

1. V anglickém originále „session,“ označuje trvající síťové spojení mezi klientem a serverem.

2.1.3 EduID a OpenID

Zadání této práce formuluje požadavek na využití přístupu uživatelů přes federaci identit. Jelikož je tvořená aplikace určena pro prostředí vzdělávacích institucí, počítá s možným zapojením do České akademické federace identit eduID.cz [13]. Tato federace je technologicky založena na systému Shibboleth popsaném v Sekci 2.1.2 a sdružuje různé vzdělávací organizace a výzkumné instituty. V současnosti má několik desítek členských organizací, mezi nimiž nechybí Masarykova Univerzita, Univerzita Karlova v Praze nebo Národní knihovna České republiky.

Druhou často využívanou federací identit je OpenID [37]. Ta je oproti eduID.cz zaměřena na samotné uživatele spíše než na organizace. V rámci federace tedy neručí za korektnost informací o uživateli jeho domovská organizace, nýbrž on sám. Namísto poskytovatelů identit zde fungují poskytovatelé OpenID. Identifikátorem každého člena je jeho unikátní URL [44] a heslo. URL uživatel získá od svého poskytovatele OpenID a heslo je identické s tím, které běžně používá pro přístup k systému, od kterého získal OpenID identifikátor. Nejznámějšími službami podporujícím OpenID jsou například Google [17], Yahoo! [48] nebo sociální síť Facebook [15].

2.2 Síťová komunikace a současné kolaborativní systémy

Kolaborativní systémy podporují spolupráci svých uživatelů za pomoci Internetových služeb. Pokud chtějí lidé spolupracovat, musí mít možnost spolu komunikovat. Komunikaci realizovanou za pomoci Internetu můžeme rozdělit do tří základních typů:

- **Psaný text.** Může probíhat jako výměna textových zpráv v reálném čase (tak zvaný „chat“) nebo se zpožděním pomocí elektronické pošty. Chat je také nejvíce využívaný v kombinaci s ostatními typy komunikačních metod. Služba zprostředkovávající textovou komunikaci v reálném čase je běžně označována jako *instant messaging* (IM) [23]. Kdokoli ji může využívat pomocí volně dostupných specializovaných programů, tak zvaných IM klientů. V případě komunikace psaným textem jsou právě IM klienti kolaborativním nástrojem využívaným řadou lidí po celém světě.

- **Audio přenos.** Čistý audio přenos je variantou telefonních hovorů uskutečněných přes Internet. Většinou se jedná o Voice over Internet Protocol (VoIP) [30]. Tento protokol poskytuje hlasové služby podobné klasickému telefonování přes pevnou linku. Přenášený zvuk je však

digitalizován a „rozsekán“ na části, jejichž další postup sítí je ošetřen síťovými protokoly. Jako koncová zařízení slouží buď to počítače vybavené speciálním softwarem, nebo telefonní přístroje taktéž úzce zaměřené na tuto službu.

- **Audio-Video přenosy.** Zkráceně jsou označované jako videokonference. Kombinace přenosu zvukového a obrazového záznamu v reálném čase je nejvíce srovnatelná s osobním setkáním, a tudíž i nejpřitažlivější pro vzdálenou spolupráci. Tato varianta je poměrně nákladná na síťové připojení a samozřejmě není realizovatelná bez speciálního vybavení (webové kamery, mikrofony a podobně).

2.2.1 Současné kolaborativní systémy

Existující kolaborativní systémy jsou většinou zaměřeny primárně na jeden z výše uvedených typů komunikace. Najdou se však i takové systémy, které podporují jejich kombinaci. Pro každou ze zmíněných možností komunikace je dál uveden příklad současné aplikace, která je primárně zaměřena na její zprostředkování. Tato práce si neklade za úkol podat vyčerpávající seznam a komplexní popis funkcionality těchto aplikací. Naším cílem zde je ilustrovat zmíněné typy komunikace na konkrétních příkladech s důrazem především na jejich podporu plánování kolaborativních setkání. U vybraných systémů a služeb najdete jejich stručnou charakteristiku, hlavní zaměření, případně zajímavou funkcionalitu. Následuje popis jejich využitelnosti pro plánování kolaborativních schůzek se zaměřením na identifikaci uživatel a případné potřebné vybavení.

2.2.1.1 IM klienti

Služby známé jako instant messaging (IM) umožňují především přenos textových zpráv v reálném čase. Mezi jejich další funkce patří například možnost udržovat seznam přátel dostupných pomocí dané služby (tak zvaný „contact list“) nebo sledovat, který z uživatelů je k dispozici pro konverzaci. Příkladem těchto služeb jsou protokoly ICQ [24], Jabber [26] nebo AIM [3]. V zásadě není možné provádět komunikaci mezi uživateli různých protokolů – pokud používám službu AIM, nemohu pomocí ní kontaktovat osobu využívající Jabber.

Pro využití jednotlivých IM protokolů slouží aplikace nazývané IM klienti. Tyto programy většinou podporují služby více poskytovatelů a je jen na uživateli, kterou se rozhodne využít. Mezi IM klienty patří například Miranda [32], Pidgin [38] či Meebo [31]. Funkcionalitou se IM klienti příliš neliší. Všechny tyto programy dovolují svým uživatelům vyměňovat textové zprávy s naprosto minimálním zpožděním. Z dalších funkcí zmiňme například podporu různých upozornění na chování uživatel nebo možnost změny grafického stylu aplikace.

Z důvodu nemožnosti komunikace napříč různými IM službami je po uživateli vyžadována registrace v rámci každé služby zvlášť. Každá služba udržuje vlastní databázi uživatelů. Nenajdeme zde žádnou možnost pro autentizaci uživatele jednotně v rámci více IM služeb.

U běžně dostupných IM klientů bychom možnosti podpory plánování kolaborativních setkání hledali marně. I přesto je tento způsob komunikace při přípravě i realizaci vzdálené spolupráce hojně využívaný. Mnohé kolaborativní systémy mají implementovanou vlastní IM službu, čímž rozšiřují možnosti komunikace svých uživatelů.

Výhodou IM klientů je jejich velmi jednoduchá dostupnost a nízká cena – většinou jsou k dispozici zdarma. Kromě počítače či mobilního telefonu není potřeba žádného speciálního vybavení.

2.2.1.2 Klienti pro elektronickou poštu

Elektronická pošta (tak zvaný *e-mail*) je jeden ze způsobů komunikace za pomoci Internetu. Zprávy jsou nejčastěji textové, ale ve formě příloh je možné předávat i fotografie, hudbu a podobně. Oproti IM jsou zprávy předávány s časovou prodlevou. Nejedná se o komunikaci v reálném čase.

Programy zajišťující přenos zpráv od odesilatele k adresátovi jsou tzv. „mail transfer agents“ (MTAs). Uživatel nekomunikuje s těmito programy přímo, nýbrž přes klienta označovaného jako „mail user agent“ (MUA). Ten se stará například o poskytování uživatelsky přívětivého rozhraní pro zobrazení příchozí pošty nebo vytváření nových zpráv. Řadí se mezi ně například Microsoft Outlook [33] a Mozilla Thunderbird [35]. Přímo na Internetu je potom možné využít služeb webových aplikací jako Gmail [16] nebo AOL Mail [4] [20].

Elektronická pošta je velmi často využívána pro plánování kolaborativních schůzek. Umožňuje totiž hromadnou korespondenci[2], což je užitečné při předávání informací budoucím účastníkům schůzky. Klienti (MUA) rovněž udržují historii zpráv a umožňují je dál přeposílat, mazat a podobně. Účastníci kolaborativního setkání tedy mohou vzájemně rozesílat podklady pro setkání, nahraný průběh setkání nebo jiné související informace.

Uživatel je v rámci elektronické pošty identifikován svou elektronickou poštovní schránkou jejíž název je označován jako e-mailová adresa. K jejímu získání musí mít přístup k serveru poskytujícímu služby elektronické pošty. Některé servery společně se získáním elektronické poštovní schránky umožňují i zapojení do federace identit. Jedná například o Gmail, který s registrací dovoluje uživateli zažádat o zapojení do federace identit OpenID.

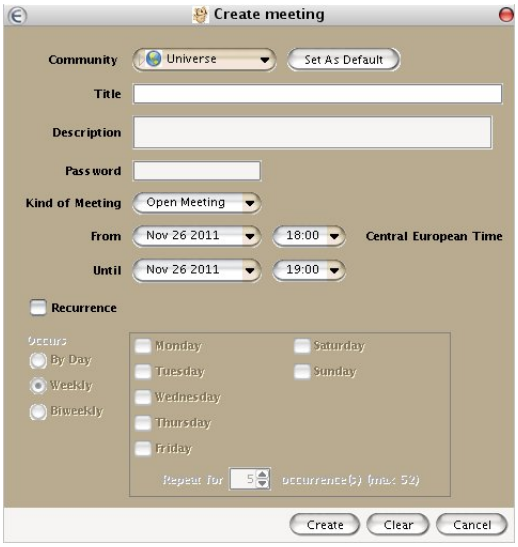
2. Rozesílání jedné zprávy na více adres současně.

2.2.1.3 EVO

Plný název systému zní Enabling Virtual Organizations [14] a náleží videokonferenčnímu kolaborativnímu systému využívanému převážně na akademické půdě. Uživatelé jsou v rámci systému seskupeni do tak zvaných *komunit*, v nichž je dál možné plánovat schůzky v podobě videokonferenčních hovorů. Tento systém disponuje množstvím funkcí souvisejících přímo s videokonferencemi (monitorování sítě, šifrování přenosu, nahrávání videohovorů a podobně). Jeho součástí je i služba pro instant messaging [14].

Jako jeden z mála kolaborativních systémů má EVO implementovanou podporu plánování kolaborativních schůzek. Umožňuje předem určit termín schůzky a dovoluje ji zabezpečit heslem. Kromě jednorázově konaných schůzek umožňuje plánovat i jejich periodické opakování. Možnosti pro plánovanou schůzku můžete vidět na Obrázku 2.2.

Ačkoli je pro konání videokonferencí nutně potřeba speciálního vybavení (mikrofonu, webové kamery, reproduktorů), tento systém se nezabývá se možnostmi plánování jeho využití.



Obrázek 2.2: Obrazovka pro plánování kolaborativní schůzky v systému EVO

2.2.1.4 Adobe Connect

Jediným komerčním kolaborativním systémem vybraným do tohoto přehledu je videokonferenční nástroj Adobe Connect [2]. Svým zaměřením je určen pro

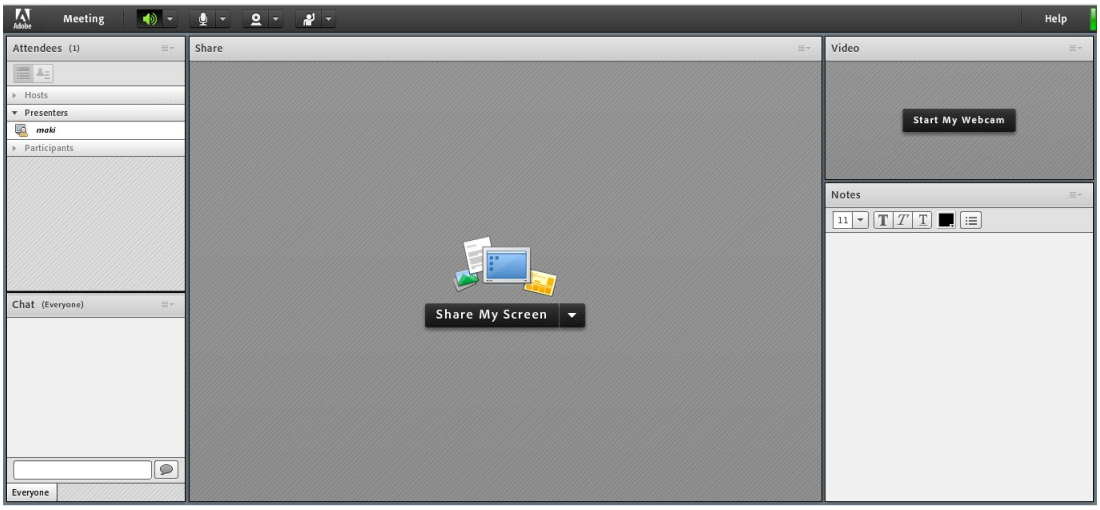
firmy k realizaci projektových schůzek, výuce přes Internet nebo webových seminářů. Systém obsahuje množství funkcí napomáhajících zmíněným aktivitám například možnost sdílet interaktivní tabuli, vytvářet poznámky nebo nahrávat konverzaci. [2]

Podpora plánování kolaborativních setkání je zde velmi propracovaná. Během přípravy organizátor do systému zadá název, pod kterým bude schůzka prezentována. Ke jménu aplikace automaticky přiřadí unikátní URL, která je pomyslnou bránou k samotné schůzce. Po vložení této adresy do webového prohlížeče proběhne pokus o spuštění komunikačního rozhraní. V rámci plánování je možné přiřadit datum a čas konání schůzky, dobu trvání, přístupové heslo a další (viz [2]).

Pro naplánování videokonference musí být uživatel registrovaný v systému. Připojit se k již probíhajícímu setkání však může i neregistrovaný uživatel, pokud je tato možnost povolena organizátorem nebo moderátorem setkání.

Ačkoli Adobe connect nemá implementovanou podporu federací identit, existují metody na jeho kombinaci se Shibboleth autentizací [9].

Systém neimplementuje podporu pro plánování využití speciálního vybavení nutného pro realizaci videokonferencí.



Obrázek 2.3: Komunikační okno systému Adobe Connect

2.2.2 Souhrn

Vzdálenou spolupráci je vhodné předem naplánovat. V rámci tohoto procesu je nezbytné určit způsob komunikace uživatelů během kolaborativní schůzky.

Může se jednat o přenos textových zpráv, audia, videa nebo jejich kombinace. Podle zvoleného typu komunikace musí být určen kolaborativní systém, který schůzku zprostředkuje a případné speciální vybavení potřebné pro její uskutečnění.

Pro každý typ komunikace jsou v Sekci 2.2.1 uvedeny příklady služeb či aplikací, které byly otestovány se zaměřením na jejich možnosti podpory plánování kolaborativních schůzek.

IM klienti a aplikace pro elektronickou poštu neposkytují žádnou podporu pro plánování kolaborativních setkání. IM klienti a elektronická pošta jsou však velmi často využívány jako prostředky pro samotné plánování.

Systém EVO poskytuje svým uživatelům možnost plánování schůzek. Neimplementuje však mechanismy pro vyrozumění potenciálních účastníků.

Aplikace Adobe Connect je nejvíce komplexním řešením ze všech testovaných programů. Umožňuje plánování i předávání informací účastníkům. Jedná se ovšem o komerční systém zaměřený spíše na firmy než běžné uživatele.

Systémy EVO a Adobe Connect vyžadují pro plné využití své funkcionality speciální vybavení. Žádný z nich však nemá implementovanou podporu pro řešení problémů s jejich dostupností.

Pro realizaci a plánování kolaborativních schůzek tedy není možné využít jednoho jediného nekomerčního kolaborativního systému. Většinou je organizátor kolaborativního setkání nucen použít více programů. Přitom, stejně jako ostatní účastníci, si pro každý z nich musí vytvořit identifikační údaje při registraci. Eliminaci násobného registrování a nutnosti pamatovat si velké množství identifikačních údajů umožňují federace identit. V rámci nich je uživateli dostupné více síťových služeb pod jedinými identifikačními údaji.

Žádný z existujících nekomerčních kolaborativních systémů tedy není možné využít jako samostatný nástroj pro plánování kolaborativních akcí s podporou rezervací speciálního vybavení. Zmíněné nekomerční programy rovněž nepodporují federace identit a tudíž nevyhovují požadavkům formulovaným v zadání této práce. Z těchto důvodů jsme se rozhodli navrhnout a implementovat vlastní systém.

3 Návrh nového systému

V této kapitole se nachází popis a ukázky z návrhu nového systému. Během něj byly nejprve určeny požadavky na systém v Sekci 3.1. Z nich byly odvozeny jeho hlavní vlastnosti popsané v Sekci 3.2. Následovalo vytvoření funkčního modelu, při kterém byla zvolena struktura aplikace a funkce, které by měla aplikace poskytovat (Sekce 3.3). V závislosti na funkčním modelu a žádaných funkcích vznikal návrh grafického rozhraní systému (Sekce 3.4).

3.1 Požadavky na systém

Návrh nového systému vychází ze zjištěných nároků na plánování kolaborativních schůzek a testování systémů představených v Sekci 2.2.2. Ta poukazuje na chybějící software pro jednotný způsob plánování kolaborativních setkání. Z prostudování problematiky přípravy na vzdálenou komunikaci v kombinaci se zadáním této práce vzešly základní požadavky na nový systém. Vytvářená aplikace musí:

- být přístupná pomocí federace identit
- umožňovat plánování kolaborativních setkání
- umožňovat rezervace zdrojů pro konkrétní kolaborativní setkání
- umožňovat jednoduchý způsob informování potenciálních účastníků o konání kolaborativního setkání

3.2 Charakteristika nového systému

Z požadavků na systém (Sekce 3.1) byly blíže určeny hlavní vlastnosti aplikace. Během tohoto procesu vznikly i specifikace funkcí a začala tvorba funkčního modelu popsaného v Sekci 3.3. V této sekci se nachází stručný popis hlavních vlastností aplikace vyplývajících ze základních požadavků.

- **Jednotná identifikace uživatelů v systému.**

  Požadavek na využití federace identit znamená absenci vlastní implementace autentizace uživatelů. Ta je prováděna v rámci dané federace a tudíž nebylo nutné ji samostatně vytvářet. Bylo však potřeba určit schéma atributů, které budou identifikovat příchozího uživatele. Vzhledem k akademickému zaměření práce bylo vybráno schéma definované

objektovou třídou `eduPerson`. Tomu musela být přizpůsobena i manipulace s uživatelskými daty což je popsáno v Sekci 4.

Podrobný soupis funkcí týkajících se uživatelů systému najdete v Sekci 3.3.

- **Plánování kolaborativních setkání**

  Systém by měl sloužit pro podporu plánování kolaborativních setkání. Nejprve bylo potřeba určit, co konkrétně bude do plánování zahrnuto. Aplikace se v tomto ohledu týká:

  – stanovení termínu setkání
  – stanovení probírané tématiky
  – zajištění dostupnosti speciálního vybavení potřebného pro realizaci setkání
  – informování potenciálních účastníků o konání setkání

  Ke každému z výše zmíněných bodů se váží funkce blíže popsané v Sekci 3.3. Plány by přirozeně mělo být možné upravovat nebo úplně rušit.

- **Správa systému**

  S plánováním kolaborativních setkání je spojená řada aspektů, které nespadají pod samotné plánování. Jedná se například o podpůrné funkce pracující s uživatelskými daty nebo s informacemi o speciálním vybavení. Tyto funkce a systém celkově by mělo být možné jednoduše spravovat.

- **Přehledné zobrazení plánovaných konferencí**

  Plánované konference bude potřeba uživatelům přehledně zobrazit. K tomuto účelu byla zvolena webová aplikace Google Kalendář [18]. Ta umožňuje tvorbu a editaci elektronických kalendářů. Hlavními důvody pro využití této aplikace je možnost vytvořené elektronické kalendáře integrovat v přehledné grafické formě na libovolné webové stránky. Aplikace je využitelná jako součást jiných webových systémů. Díky aplikačnímu rozhraní, jež je pro tuto aplikaci dostupné, je možné kalendáře a události v nich uložené vzdáleně vytvářet a upravovat. Podmínkou pro použití účtu u služby Google. V ideálním případě by však běžný uživatel při práci s vyvíjeným systémem neměl přijít s autentizací vůči této službě přímo do styku.

- **Praktické využití systému**

  Systém by měl být v praxi využitelný pro plánování schůzek v různých kolaborativních systémech. Záznamy vytvořené uživateli a udržované v systémové databázi by mělo být možné exportovat a distribuovat konkrétnímu kolaborativnímu systému. Ten by na jejich základě mohl například řídit přístup účastníků ke konkrétní schůzce. Způsob a typ exportovaní dat se může značně lišit podle nároků jednotlivých kolaborativním systémům. Tato práce však není zaměřená na propojení rezervačního systému s dalšími kolaborativními systémy. Problematika exportování a distribuce dat zde tedy není více rozebrána.

3.3 Funkční model systému

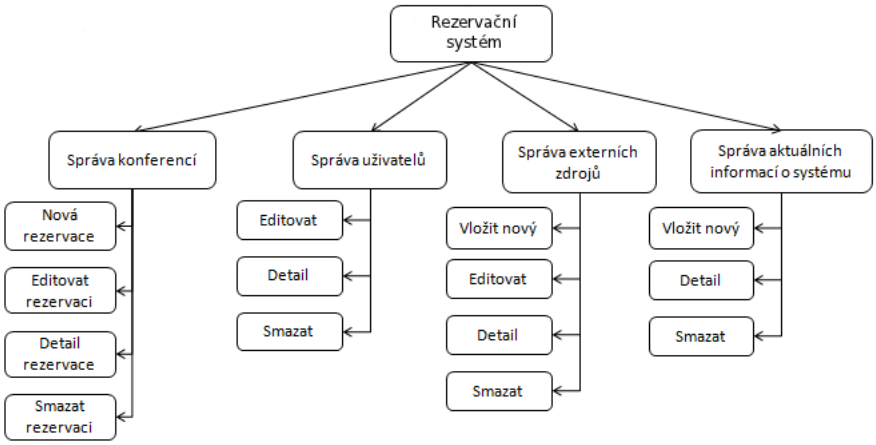
Tvorba funkčního modelu zahrnovala určení struktury aplikace a zvolení funkcionality jejích jednotlivých částí. Struktura systému bude členěna na čtyři správní části, které jsou vzájemně propojeny. Jedná se o

- **správu uživatelů** poskytující funkce pro manipulaci s uživatelskými daty v rámci systému
- **správu aktuálních informací o systému** poskytující funkce pro manipulaci se zprávami týkajícími se systému
- **správu externích zdrojů** poskytující funkce pro manipulaci s informacemi o speciálním vybavením
- **správu konferencí** poskytující funkce pro plánování kolaborativních setkání

V následujícím textu jsou zmíněné správní části blíže specifikovány. Strukturovaný popis systému najdete v elektronické Příloze A.

3.3.1 Správa uživatelů

Ačkoli uživatelé přistupující k systému jsou již autentizovaní vůči federaci identit, rozhodli jsme se implementovat v systému vlastní správu uživatelů. Jedná se především o udržování některých identifikačních údajů od všech, kteří přistupují k systému. V rámci systému jsou dále uživatelé děleni na dvě skupiny s různými privilegii.



Obrázek 3.1: Zjednodušená struktura systému

3.3.1.1 Správa uživatelských dat

Autentizovaní uživatelé budou přistupovat k systému se sadou atributů obsahujících jejich osobní údaje. Systém bude některá data uchovávat v databázi a využívat je k usnadnění plánování konferencí a správě systému jako celku. Uživatele nebude možné do systému přidávat manuálně, ale bude umožněno jejich smazání ze systémové databáze.

Dále by nebylo vhodné, aby všichni uživatelé mohli manipulovat s některými funkcemi systému, jako je například editace informací o specializovaném vybavení. Rovněž by se stylem „všichni můžou všechno“ mohla přibýt řada problémů se zásahem do cizích plánovaných akcí. Z těchto důvodů bylo nezbytné definovat privilegia uživatelů.

3.3.1.2 Rozdělení uživatelů podle privilegií

Předpokládáme, že většina uživatelů přistupujících k systému ho bude chtít využít pro plánování kolaborativního setkání. Tento typ lidí je v rámci systému označován jako *běžní uživatelé*. Možnosti práce se systémem jsou pro ně limitovány na:

- zobrazení všech plánovaných konferencí (jen jejich název a dobu konání)
- čtení aktuálních informací o systému
- plánování konferencí
- manipulaci s konferencemi, které sami plánují (upravovat, mazat)

O celkovou správu systému se potom budou starat jeho *administrátoři*. Tato skupina uživatelů systému bude mít stejné možnosti jako běžení uživatelé, obohacené navíc o možnost

- zobrazení kompletních informací o všech plánovaných konferencích
- manipulaci s aktuálními informacemi o systému (vytváření, mazání)
- manipulaci s externími zdroji (jejich vkládání do systému, úprava, smazání)
- manipulaci se všemi plánovanými konferencemi (jejich úprava, smazání)
- manipulaci se uživatelskými daty (změna privilegií, smazání uživatele)

Běžný uživatel tedy může povýšit svá práva na administrátora pouze za pomoci jiného uživatele systému, který již administrátorem je. Ten bude mít možnost přidělit uživateli příslušná privilegia.

Pro přehlednější ilustraci možností jednotlivých typů uživatelů najdete v elektronické příloze této práce UML [43] diagramy případů užití pro běžného uživatele a administrátora.

3.3.2 Správa aktualit

V rámci každého systému je občas potřeba informovat uživatele o změnách, novinkách či výpadcích služeb. Pod pojmem *aktualita* se rozumí právě takováto informace, související s funkcí systému a prvků s ním spojených (externích zdrojů, uživatelů, plánovaných konferencí). Aktuality bude možné

- vytvářet a editovat (včetně mazání) – dostupné pouze administrátorům
- přečíst aktuality – dostupné všem uživatelům systému

3.3.3 Správa externích zdrojů

Speciální vybavení je v rámci systému označováno jako *externí zdroje*. Není dáno přesné omezení na to, co jím je a co není. Rozhodnutí náleží uživatelům. Typicky by se však mělo jednat o monitory, kamery nebo multimediální místnosti.

Aby bylo možné určit kdy budou dané zdroje dostupné a případně je pro plánované kolaborativní setkání rezervovat, bude systém udržovat jejich záznamy. Každý záznam by měl obsahovat identifikační údaje zdroje. Na tyto údaje se budou vázát informace o možnostech využití konkrétního zdroje. Záznamy o externích zdrojích bude možné

- vytvářet a editovat (včetně mazání) – dostupné pouze administrátorům
- zobrazit – dostupné všem uživatelům systému
- rezervovat při plánování konference – dostupné všem uživatelům systému

3.3.4 Správa konferencí

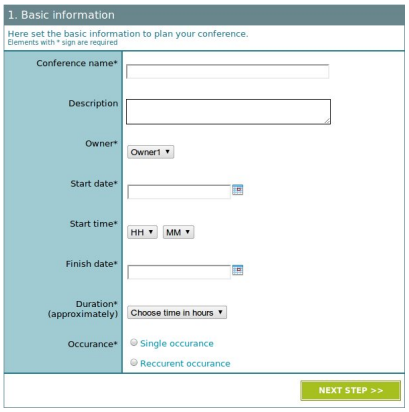
Pod pojmem *konference* se v rámci systému rozumí kolaborativní setkání. Metodika jejich plánování byla zvolena po vzoru systémů EVO [14] a Adobe Connect [2]. Zde se však místo virtuálních místností bude jednat o rezervační záznamy reprezentující jednotlivé konference.

Na tyto záznamy by měly být navázány veškeré další informace související s plánováním, jako například termín konání, probíraná témata nebo rezervace externích zdrojů. Rezervační záznamy by mělo být možné:

- Vytvářet – dostupné pro kteréhokoliv z uživatelů. Pro daný záznam dále bude figurovat jako jeho „vlastník“.
- Editovat (včetně smazání) – pro běžné uživatele by mělo platit, že plány smí editovat pouze jejich vlastník, správci systému by ovšem měli mít možnost manipulovat se rezervačními záznamy náležícími libovolným uživatelům.

Samotná tvorba rezervačního záznamu by měla být rozdělena do čtyř až pěti kroků v závislosti na počtu opakování plánované konference. Ty se totiž mohou konat jednorázově nebo opakovaně. V případě opakovaného konání konferencí by měl být při vytváření rezervačního záznamu obsažen krok pro definování periodicity (navíc oproti plánování jednorázové akce).

Krok 1: Základní informace Nejprve by měl být uživatel požádán o zadání názvu a stručného popisu konference. Tyto dva údaje by měly charakterizovat téma, kterému bude konference věnovaná. Následuje zadání termínu konání konference a volba, zda se bude jednat o jednorázovou nebo periodickou akci. V případě, že by plánovaná konference měla proběhnout pouze jednou, bude uživatel přesměrován přímo na Krok 2. V opačném případě je potřeba definovat periodicitu v Kroku 1.5.

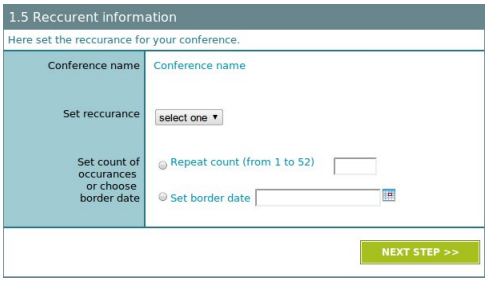


Obrázek 3.2: První krok tvorby rezervačního záznamu

Krok 1.5: Periodicita Tento krok slouží pouze k získání informací o případném opakování konference. Možnosti opakování by měly být:

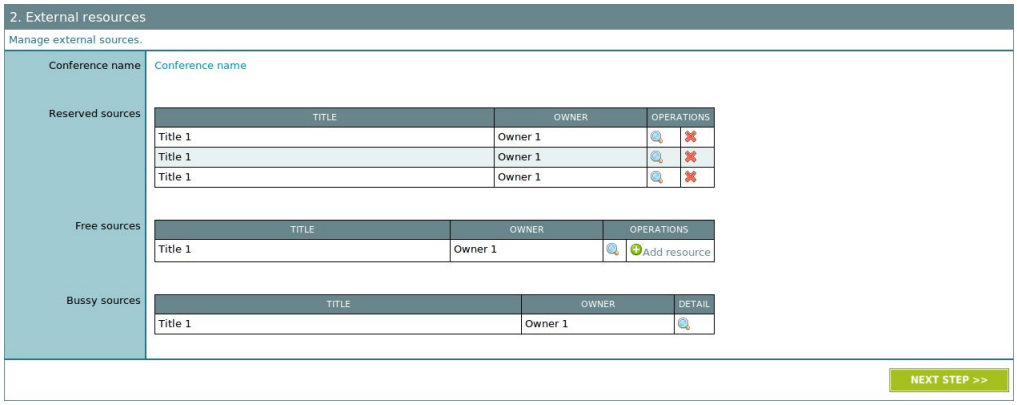
- týdně
- po dvou týdnech
- měsíčně

Dále bude možné vybrat den opakování. Hranice po kterou bude schůzka opakována bude možné určit krajním datem nebo počtem opakování konferencí.



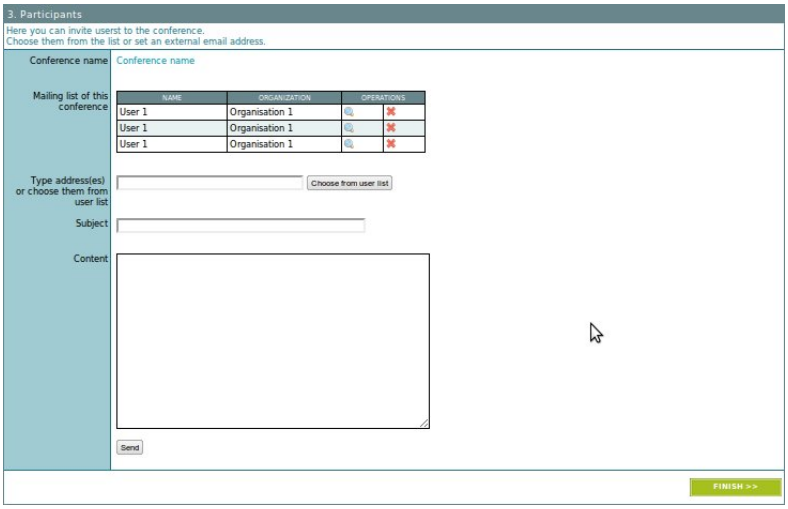
Obrázek 3.3: Zadání údajů o periodicitě

Krok 2: Rezervace zdrojů Zde bude možné vybrat ze seznamu zdrojů ten pravý pro potřeby konkrétní konference. Rezervovat půjde pouze takový zdroj, který bude pro čas konání konference k dispozici.



Obrázek 3.4: Rezervace externích zdrojů

Krok 3: Správa účastníků K plánované konferenci bude možné jednoduše přizvat potenciální účastníky zasláním e-mailu. Organizátor vybere uživatele buď to přímo z databáze systému nebo zadá libovolný e-mail. Vytvoří tak seznam kontaktů, který bude udržován pro daný konferenční záznam v databázi. Seznam bude možné editovat (přidávat, odebírat položky) a kdykoli na něj zaslat další hromadnou zprávu.



Obrázek 3.5: Email účastníkům

Krok 4: Výsledný detail konference Poslední krok poskytuje uživateli celkový přehled o konferenci. Informace zadané postupně ve všech krocích tvorby rezervačního záznamu budou zobrazeny pouze pro čtení.

3.4 Model grafického rozhraní

Návrh grafického rozhraní závisel především na rozdělení funkcionality na jednotlivé správní sekce s ohledem na delegaci privilegií uživatelů. Pro tvorbu byla využita technologie prototypování obrazovek.

Jedná se o tvorbu návrhů stránek zobrazovaných uživateli. Pravidla vytváření nejsou nikde specifikována je však možné prototypy různými způsoby dělit. Jeden z typů členění je na horizontální a vertikální.

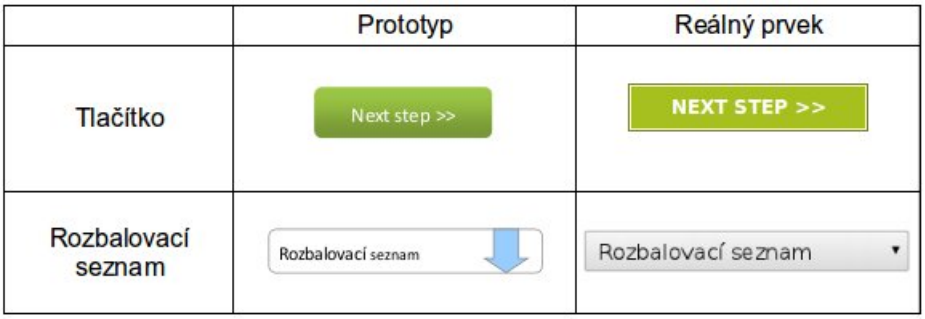
- vertikální prototypy – zaměřené na konkrétní části vyvíjené aplikace zahrnují hlubokou funkcionalitu
- horizontální prototypy – jsou soustředěny na celé vnější rozhraní aplikace, nezabývají se vnitřními funkcemi [40]

Při návrhu nového systému byly prototypy využity především k návrhu grafického rozhraní. Typem se tedy řadí mezi horizontální prototypy.

Ke tvorbě prototypů bylo přistupováno souběžně se specifikací funkcí a funkčního modelu systému. S každou funkcí interagující s uživatelem vznikl nový prvek v prototypu (nová obrazovka, formulář, tlačítko, odkaz a podobně). Úpravy funkčního modelu se tedy promítly i do příslušných prototypů. Postupně tak vzniklo několik sad prototypů obrazovek. Jejich vývoj byl ovlivněn i poznámkami získanými od laické veřejnosti. Tímto způsobem jsme se snažili docílit jisté intuitivnosti v ovládání systému.

Při tvorbě prototypů jsme se nesnažili o získání kvality grafického zpracování shodného s reálným systémem. Rozdíly v prvcích prototypů a reálného systému můžete vidět na Obrázku 3.6.

V elektronické příloze Ukázka_prototypů.pdf najdete jednu sadu návrhů obrazovek dostupných běžnému uživateli. Stejným způsobem byly vytvořeny i modely obrazovek administrátora.



Obrázek 3.6: Ukázka prvků prototypů v porovnání s reálnými prvky

4 Implementace

Tato kapitola je věnovaná způsobu implementace systému. Nejprve jsou zde představeny použité technologie. Následuje popis vnitřní struktury programu. Dále jsou podrobněji rozebrány některé zajímavé implementační prvky.

Texty a komentáře v implementovaném systému jsou v anglickém jazyce.

4.1 Použité technologie

Pro implementaci systému byla zvolena kombinace jazyka PHP [39], databázového systému MySQL [34] a serveru Apache [6]. Tyto tři prvky obsahují funkce pro vzájemnou podporu a v praxi jsou často využívány pro tvorbu webových aplikací. Grafická úpravu stránek je tvořena jazykem HTML [21] s podporou CSS [10].

Při implementaci bylo využito převážně objektově orientovaného přístupu, který jazyk PHP podporuje. Jako součást jazyka je využívaná knihovna cURL (Client URL Library) [11]. Pro některé implementační prvky bylo využito knihovny jQuery [27].

4.2 Struktura implementace

Systém je složen ze tří skupin skriptů, které se vzájemně doplňují. První skupina jsou třídy zajišťující zobrazení dat uživateli, tedy grafickou stránku systému. Další jsou třídy komunikující s databází a pracující s daty získanými od uživatele. Poslední jsou kontrolní skripty, řídící předávání a zpracování informací. Speciální skupina jsou skripty sloužící k podpoře a konfiguraci ostatních skriptů.

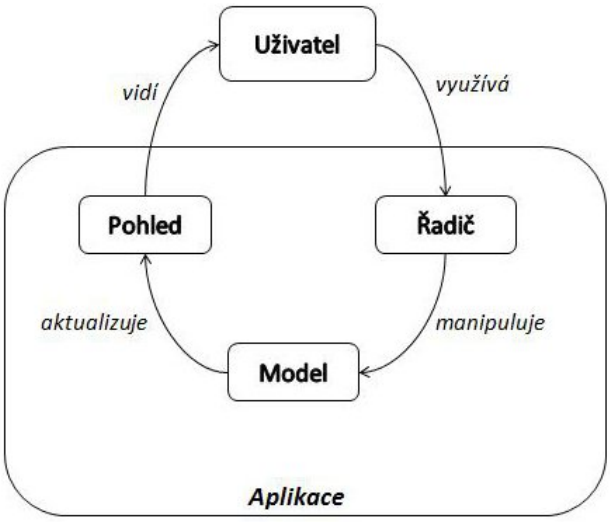
Tento způsob implementace vychází z návrhového vzoru označovaného jako Model-View-Controler (MVC) [8]. Přeloženo do češtiny se jedná o Model-Pohled-Řadič. Ačkoli jsou všechny části propojeny, řeší rozdílné problémy a mohou být implementovány různým způsobem.

- Model(model) – je zodpovědný za datovou vrstvu aplikace. Rozhodně neplatí, že by model byl srovnatelný s databází. Tato část MVC zahrnuje i funkce pro práci s daty. Zpracovává jakékoli dotazy týkající se dat, jako jsou aktualizace nebo čtení a zapisování údajů.

- Pohled(view) – spravuje presenční vrstvu aplikace. Dostává údaje, které jsou určeny na vypsání a může mít vlastní presenční logiku, která výpis zajistí. Pohled si můžeme představit jako jednotlivé stránky zobrazované uživateli. Ve striktním pojetí MVC architektury by zobrazované údaje

měly pocházet pouze od řadiče, v některých případech je ale možné se setkat i s daty předanými modelem. Ve skutečnosti by mělo být pohledu naprosto jedno odkud data pocházejí, důležité je poznat jejich typ a formát.

- Řadič(controler) – jak už název napovídá řadič je zodpovědný za řídící vrstvu. Získává informace o přáních uživatele z pohledů a předává dotazy modelu. Následně od modelu získá potřebné informace a předá je pohledu k zobrazení. Řadič se často skládá z více komponent, které sami o sobě mohou implementovat úplně jiné návrhové vzory [1].



Obrázek 4.1: Schéma návrhového vzoru MVC

4.2.1 Prezenční vrstva aplikace

Prezenční vrstva zahrnuje především grafickou stránku aplikace. Jednotlivé obrazovky vzniklé podle navržených prototypů slouží k získávání/výpisu dat od/pro uživatele.

Implementace grafického rozhraní vychází z faktu že některé elementy jsou pro všechny stránky identické, například hlavička a patička stránek. Mění se jen určitá část prvků a i ty si jsou do určité míry podobné, například tabulky, nadpisy a podobně. Vlastnosti společné po všechny stránky jsou zobrazovány

1. Více o problematice návrhového vzoru MVC zde [8].

metodami třídy `cPage`. Ta je děděna všemi ostatními třídami pro tvorbu grafického rozhraní. Obsahuje metody pro:

- interpretaci veškerého HTML kódu
- zobrazení hlavičky stránek (včetně skrytých meta informací a tak dál)
- zobrazení patičky stránek

Stránky zobrazované uživatelům se dají pomyslně rozdělit na dva typy. První zobrazuje detailní informace, ať už se jedná o uživatelská data, podrobnosti o zdrojích či konferencích. Druhý typ se soustředí na prezentaci seznamů (seznamy uživatelů, externích zdrojů, a tak dál). V rámci obou typů jsou opět některé prvky společné pro všechny stránky. O části společné pro zobrazení detailních informací se starají metody třídy `cPageDetail`. Zahrnují výpis:

- nadpisu k detailnímu formuláři
- hlavičky a patičky detailního formuláře

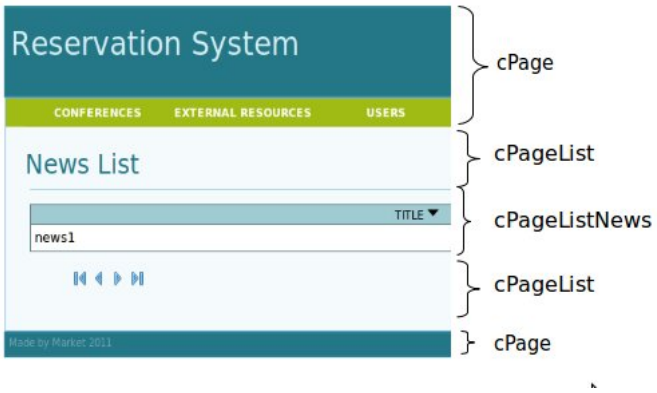
Ve druhém případě se jedná o třídu `cPageList`. Její metody realizují:

- nadpis k seznamu
- hlavičku a patičku seznamu
- navigaci v seznamu

Prvky, které jsou pro každou stránku jedinečné, jsou zobrazovány metodami tříd pojmenovaných kombinací určující typ stránky (`cPageList`, `cPageDet`) a zobrazovaných dat. Zobrazovaná data se mohou týkat aktualit, uživatelů, zdrojů nebo konferencí.

Formát pojmenování tříd je tedy `cPageListNázev_dat` nebo `cPageDetNázev_dat`. Kdy část „Název_dat“ je:

- News pro práci s aktualitami.
- User pro práci s uživatelskými daty.
- PSrc pro práci s externími zdroji.
- Conf pro práci s konferencemi.



Obrázek 4.2: Rozdělení stránky

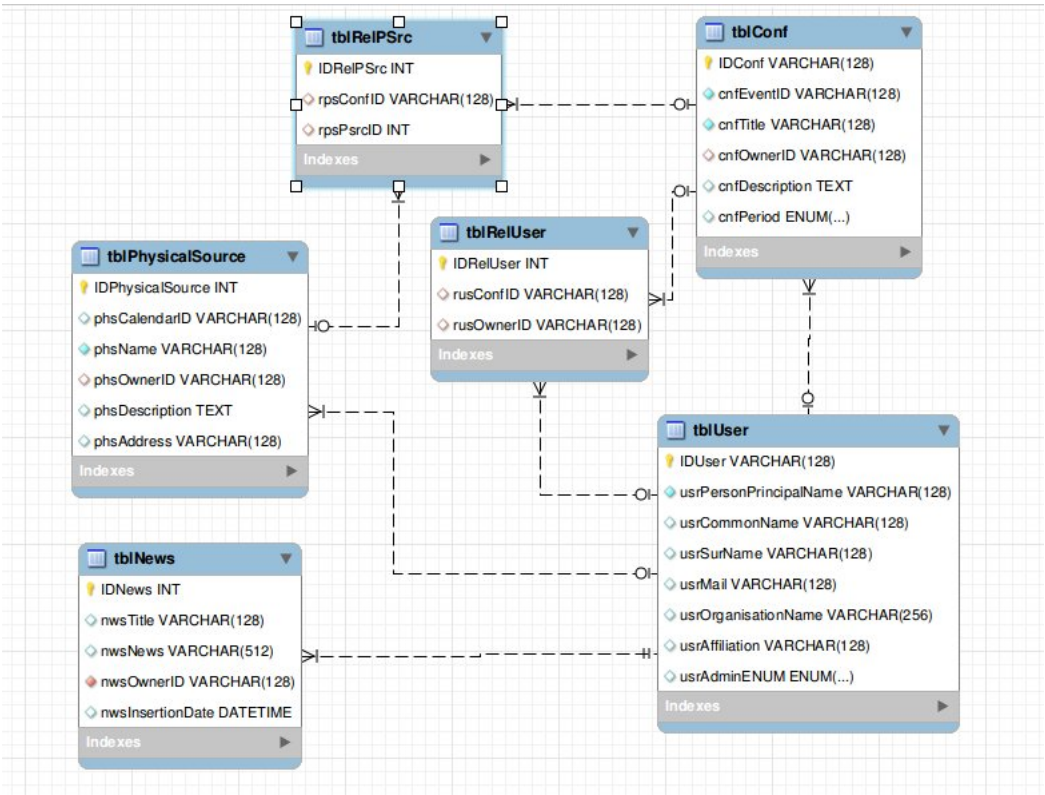
Většina metod všech tříd starajících se o zobrazení grafického rozhraní obsahuje určitou část kódu HTML. Stránka výsledně zobrazená uživateli je složená z kousků kódu vypsaných jednotlivými metodami. V rámci metod je každá řádka HTML kód uložena jako řetězec do pole jazyka PHP. O výpis tohoto pole pro webový prohlížeč se stará metoda `showHTML($f_array)` ze třídy cPage. Ta vypisuje postupně všechny prvky pole `$f_array` na klasický výstup, kde webový prohlížeč kód zpracuje a zobrazí.

4.2.2 Datová vrstva aplikace

Data v systému mohou pocházet od uživatele, potom je potřeba uložit je do databáze, nebo naopak pocházejí z databáze a je potřeba předat je grafickému rozhraní. V obou případech je nutné pracovat s databází. Její model můžete vidět na Obrázku 4.3.

Každý ze správních celků systému – aktuality, uživatelé, externí zdroje a konference – má vlastní databázovou tabulku. Pojmenování těchto tabulek je závislé na názvu správní části, jejíž data udržuje. Formát pojmenování je `tblNázev_části`. Jedná se o tabulky `tblNews`, `tblUser`, `tblPSrc` a `tblConf`. Dále databáze obsahuje tabulky udržující informace o vztazích mezi správními celky.

Datová vrstva nezahrnuje pouze databázi samotnou, ale i třídy zpracovávající data do ní vkládaná nebo z ní vybíraná. Komunikaci s databází, vkládání a získávání obsahů jejích tabulek zajišťuje skupina tříd v čele s `cDB`. Třída `cDB` je základní pro databázové operace, protože obsahuje metody pro připojení/odpojení k/od databázi a vlastní realizaci dotazů k databázi. `cDB` je dědičná pro všechny ostatní třídy manipulující s databází. Každý správní celek má opět



Obrázek 4.3: Model databáze systému

vlastní třídu zajišťující databázové operace pro data s ním spojená. Takováto skladba tříd je využívána kvůli velkému množství rozdílných parametrů spojených s jednotlivými správními celky.

V rámci datové vrstvy je implementovaná i třída `cSession` sloužící pro správu seancí v rámci systému. Ty jsou využívány k udržování některých dat při přechodech mezi skripty.

Dalží částí datové vrstvy systému je třída `cAuth`, které by měla sloužit ke komunikaci s Google Kalendářem. Obsahuje funkce zajišťující autentizaci do služby Google a řadu dalších, určených pro manipulaci kalendáři a událostmi v nich uložených. Jako hlavní zdroj při tvorbě této třídy sloužily webové stránky aplikačního rozhraní Google Kalendáře [19]. Více informací o způsobu začlenění Google Kalendáře do systému najdete v Sekci 4.2.5.

4.2.3 Řídící vrstva aplikace

Zobrazovaná grafického rozhraní a manipulaci s daty řídí několik skriptů, které jsou psány procedurálním stylem a pouze volají potřebné třídy a metody. Jejich hlavním obsahem je vždy několik příkazů `switch`, pomocí kterých jsou řízeny výpisy stránek a volané metody. Dále obsahují načítání pomocných souborů, nastavení časové zóny nebo startování seance.

Řídících skriptů je opět několik. Každý správní celek aplikace je řízen vlastním skriptem.

4.2.4 Podpůrné skripty

Aplikace využívá čtyři podpůrné skripty. Hlavním z nich je `support.inc`. Ten je volán všemi řídícími skripty a obsahuje řadu příkazů `include` pomocí nichž jsou vkládány obsahy dalších podpůrných skriptů. Mimoto obsahuje funkce pro automatické načítání tříd a ladící výpisy. Dalšími podpůrnými skripty jsou `dbSupport.inc` s jedinou funkcí pro získání parametrů pro připojení k databázi a `rsConstants.inc` obsahující definice některých globálních konstant využívaných v systému. Posledním skriptem je `rsError.inc` s funkcí pro případný výpis chyb generovaných systémem.

4.2.5 Zajímavé implementační prvky

4.2.5.1 Přístup uživatelů

Jak je zmíněno v 3.1 jedním z požadavků na systém byla jeho využitelnost ve federaci identit. Konkrétně jsme se soustředili na federaci eduID.cz (zdůvodněno v Sekci 2.1.3). Ta využívá k autentizaci uživatelů služby middlewaru Shibboleth, blíže popsaného v Sekci 2.1.2. Uživatel přistupující k systému by tedy již měl být autentizován v rámci federace identit. Poskytovatel identity uživatele by měl rovněž dát systému k dispozici sadu atributů souvisejících s daným uživatelem.

Způsob implementace přístupu uživatelů k systému vychází z informací získaných na stránkách České akademické federace identit eduID.cz [13]. Z tohoto zdroje bylo zjištěno, že pokud je uživatel autentizovaný v rámci federace identit je jeho hlavní identifikační atribut (atribut, který ho ve federaci jednoznačně identifikuje) dosažitelný v proměnné prostředí `REMOTE_USER`. S autentizací uživatele také vzniká seance na straně poskytovatele služeb. Po dobu platnosti této seance je uživatel implicitně autentizován vůči všem autentizačním požadavkům přicházejícím ze strany poskytovatele služeb, který seanci udržuje.

Seance na straně poskytovatele služeb mají omezenou dobu platnosti po jejímž vypršení se musí uživatel opět autentizovat. Tento princip má jisté nevýhody, například možnost ztráty dat při vypršení seance v nevhodnou dobu.

Řešení tohoto problému, jak je nastíněno na stránkách eduID.cz, je využití vlastní seance na úrovni systému v kombinaci se službami Shibbolethu pro autentizaci k poskytovateli služeb a výměně atributů.

Tuto kombinaci je možné realizovat tak, že jediná část systému, která bude vyžadovat platnou seanci od poskytovatele služeb bude stránka pro přihlážení uživatele k systému. Při pokusu uživatele o vstup do systému prostřednictvím přihlažovací stránky bude spuštěn proces pro vytvoření seance na straně poskytovatele služeb. Následně systém vytvoří vlastní seanci v rámci níž bude dál uchovávat získané uživatelské atributy. Uživatel je potom přesměrován na hlavní stránku systému, která už nevyžaduje a nekontroluje existenci platné seance od poskytovatele služeb.

Vyvíjený systém je vystavěn na popsaném způsobu kombinovaného využití vlastní seance a služeb Shibbolethu. Jeho součástí je skript `login.php`, na nějž by v případě zapojení systému do eduID.cz odkazovala upravená sekce `Location` v konfiguraci serveru Apache (ukázka konfigurace zde [5]).

Skript je volán stránkou pro zadání přihlašovacích údajů a obsahuje kód pro inicializaci procesu vytvoření seance na straně poskytovatele služeb. V případě, že by dosáhl vytvoření platné seance na straně poskytovatele služeb, vytvořil by také vlastní seanci pro systém a uložil si získané uživatelské atributy. Následně by uživatele přesměroval na kontrolní stránku systému. Pokud by však nebyla ověřena platnost seance na straně poskytovatele služeb, byl by uživatel přímo přesměrován zpět na stránku pro zadání přihlašovacích údajů.

4.2.5.2 Google Kalendář

Tato kapitola je věnována popisu začlenění webové aplikace Google Kalendář do vytvářeného systému. Najdete zde stručnou charakteristiku této aplikace, popis způsobu komunikace s jejím rozhraním, vybraný způsob autentizace webových aplikací k danému rozhraní a způsob jejího začlenění do vytvářeného systému.

Google kalendář je nástroj pro jednoduchou tvorbu a správu elektronických kalendářů. Do každého kalendáře lze vkládat tak zvané *události*. Ty je dál možné upravovat, mazat, barevně označovat, upozornit na ně další uživatele a podobně. Kalendáře je rovněž možné zakomponovat pomocí technologie iFrame [22] do webových stránek.

Uživatelé mohou využívat Google kalendář až po zaregistrování ke službě Google, která aplikaci provozuje. Každý registrovaný uživatel může v rámci svého účtu spravovat (vytvářet, upravovat, mazat) vlastní kalendáře a rozhodovat, zda bude jejich obsah veřejně dostupný nebo ne.

Aplikaci Google kalendář je možné spravovat i prostřednictvím jiných webových aplikací. V tom případě musí být operace schváleny vlastníkem kalendáře a webová aplikace žádající o jejich provedení musí být autorizovaná a

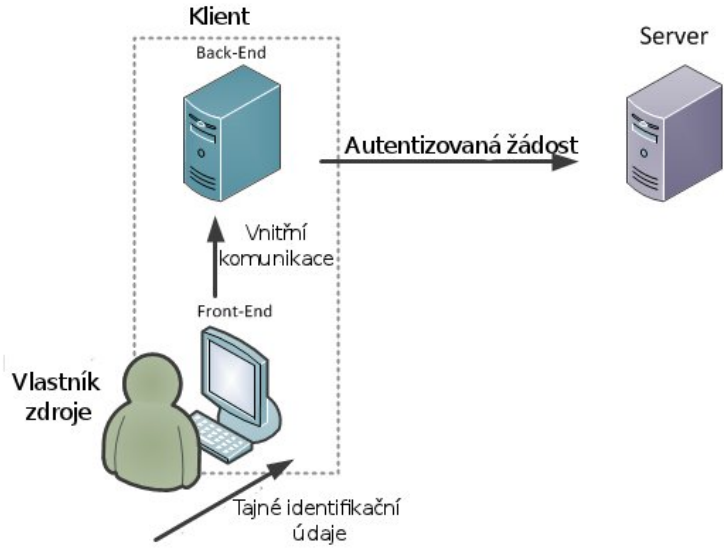
autentizovaná vůči aplikačnímu rozhraní služby Google.

Komunikace s aplikačním rozhraním Google Kalendáře je založena na principu zasílání požadavků definujících jaká akce má být vykonána s konkrétními daty. Technologie přenosu dat je inspirována softwarovou architekturou REST (Representational State Transfer) [41] a přenášená data jsou ve formátu JSON (JavaScript Object Notation) [28]. Více o této tématice najdete na stránkách aplikačního rozhraní aplikace Google Kalendář [19].

Autentizovat požadavky směřující na data chráněná aplikací Google Kalendář je možné různými způsoby. Pro nejnovější verzi aplikačního rozhraní je však doporučeno využití protokolu OAuth 2.0 [36]. Jedná se o otevřený autentizační protokol který byl vytvořen speciálně pro umožnění bezpečné autentizace vůči různým aplikačním rozhraním.

Ve specifikaci tohoto protokolu figurují tři hlavní postavy – klient, server a vlastník zdroje. Vlastníkem zdroje bývá většinou uživatel. Server udržuje data vlastněná uživatelem a klient může být například webová aplikace žádající o udržovaná data.

V případě využití webové aplikace v roli klienta, vlastník zdroje komunikuje s jednou částí jejího rozhraní, zatímco server s jinou. Klient však stále vystupuje pro autentizační mechanismus jako jednotný celek. [36]



Obrázek 4.4: Model autentizace přes webovou aplikaci (překresleno z [36])

OAuth autentizace může být podle počtu komunikujících stran rozdělena na dva typy. Takzvaný „3-Legged“ OAuth reprezentuje klasický model vlastník dat, klient a server. Varianta „2-Legged“ OAuth vynechává roli vlastníka dat, za

nějž je považován klient.

V rámci systému by měl být Google Kalendář využit při správě konferencí a externích zdrojů následovně:

Správa konferencí. Pro plánování konferencí by měl sloužit primární kalendář uživatelského účtu vytvořeného pro potřeby rezervačního systému. Do tohoto kalendáře by jako události byly vkládány základní informace oplánované konferenci – termín konání, název a popis. Tyto informace by byly k dispozici k nahlédnutí všem uživatelům systému. Při vytvoření události v kalendáři vrací aplikační rozhraní Google Kalendáře unikátní identifikátor dané události. Tento by byl uložen do databáze systému a využit pro potřeby editace základních informací o konferenci.

Správa zdrojů. Při vkládání zdroje do systému by měl být pro něj vytvořen vlastní elektronický kalendář v aplikaci Google Kalendář. Unikátní identifikátor získaný při jeho tvorbě by měl být uložen v databázi systému a použit pro další manipulaci se zdrojem. Při rezervaci zdroje v rámci některé konference by měla být do kalendáře příslušného zdroje vložena událost značící využití zdroje v daný termín.

Aby bylo možné aplikaci Google Kalendář ve vyvýjeném systému využít, byl vytvořen uživatelský účet u služby Google. V rámci něj byla registrována i samotná aplikace. Registrace jakékoli aplikace vůči službě Google je vždy jedinečná pro její konkrétní umístění. Stejně tomu je i u vytvářeného systému.

Následně bylo potřeba implementovat autentizační mechanismus pro přístup k aplikaci Google Kalendář. Nejprve jsme se snažili o implementaci varianty 2-Legged autentizace. Chtěli jsme se tak vyvarovat potřeby interakce uživatele přímo s rozhraním služby Google. Po zdlouhavém a značně komplikovaném testování různých, volně dostupných knihoven zabývajících se touto problematikou, jsme byli nuceni danou variantu autentizace opustit. S odstupem času jsem také zjistili, že ačkoli Google možnost 2-Legged OAuth autentizace podporuje, není k dispozici pro běžné uživatelské účty. Museli jsme tedy přejít na klasickou variantu 3-Legged OAuthu. Tato varianta má však značnou nevýhodu v nutnosti manuálního schválení požadavků na manipulaci s daty jejich vlastníkem.

Pro implementaci autentizačního mechanismu a komunikaci vytvářeného systému s aplikačním rozhraním Google Kalendáře bude používaná volně dostupná specializovaná PHP knihovna. Její obsah je v rámci vytvářeného systému volán třídou `cAuth`. Ta obsahuje metody zajišťující autentizaci vůči Google Kalendáři. Dále se v ní nachází také metody pro manipulaci s kalendáři a daty v nich obsaženými.

V elektronické příloze `Autentizace.pdf` dále najdete přístupové údaje k účtu služby Google vytvořenému pro potřeby vyvíjeného systému a stručný návod na zjištění a případnou úpravu identifikačních dat náležících vytvářenému systému.

4.3 Závěr

Cílem této práce bylo najít nebo vytvořit softwarové řežení pro plánování schůzek v kolaborativních systémech. Plánování by mělo zahrnovat i možnost rezervace speciálního vybavení spojeného s konkrétním kolaborativním setkáním. Přístup k rezervačnímu systému by měl být realizován přes federaci identit také kvůli možnostem autorizace uživatelů.

Při tvorbě práce bylo otestováno několik současných kolaborativních systémů. Testování bylo zaměřeno zejména na jejich podporu plánování kolaborativních setkání. V rámci plánování nás především zajímala přítomnost mechanismů pro zajištění dostupnosti speciálního vybavení potřebného pro realizaci kolaborativních setkání. Dalším důležitým faktorem ověřovaným u vybraných systémů byla jejich možnost zařazení do federace identit, kvůli možnostem jednotné autorizace uživatelů.

Žádný z testovaných systémů nevyhovoval požadavkům na podporu plánování kolaborativních setkání v kombinaci se schopností rezervování speciálního vybavení a přístupem přes federaci identit. Z toho důvodu začal vznikat nový systém. V průběhu práce byla kompletně navržena funkcionalita a struktura systému, stejně jako jeho grafické rozhraní. Při tvorbě návrhu byly využity poznatky získané při testování současných kolaborativních systémů. Následně byla implementována funkcionalita:

- umožňující správu uživatelů pracujících se systémem – ošetření jejich přístupu, možnosti přidělení privilegií, smazání uživatele z databáze systému; funkční jsou i mechanismy pro řízení přístupu k funkcí v závislosti na privilegiích konkrétního uživatele
- umožňující správu zdrojů – do systému je možné vložit informace o specializovaném vybavení a tyto informace dále upravovat (včetně smazání)
- umožňující správu aktuálních informací týkajících se systému – možnost vkládat, zobrazovat a editovat krátké zprávy informativního charakteru.

Při pokusu o implementaci aplikace Google Kalendář do námi vyvíjeného systému se vyskytly potíže s autentizačním procesem, které se nám nepodařilo odstranit ani s pomocí vedoucího práce. Během řešení těchto problém vyšlo

najevo, že Google vyžaduje pro využívání daného autentizačního mechanismu speciální druh účtu [7], přičemž tato informace není dostatečně zdůrazněna a je obtížné ji dohledat. Kvůli těmto komplikacím se nám nepodařilo dokončit integraci Google Kalendáře včas.

Hlavním námětem pro práci do budoucna je rozšíření uživatelské dokumentace, kterou v současné podobě nepovažujeme za dostačující pro praktické využití vyvinutého systému. Tato by však měla být dostatečná pro demonstraci jeho základní funkcionality, což bylo naším cílem.

Celkově bylo vyvinuto jádro nového systému pro plánování kolaborativních schůzek, které by nyní bylo vhodné obohatit o další funkcionalitu. Jako příklad můžeme uvést sdílení dokumentů nebo podporu skupin uživatelů.

Literatura

[1] *AAI*. Webová stránka popisující AAI [online] [cit. 2011-12-20] Dostupná z WWW: `http://www.switch.ch/aai/about/`.

[2] *Adobe*. Webová stránka projektu Adobe Connect[online] [cit. 2011-12-20] Dostupná z WWW: `http://www.adobe.com/products/adobeconnect.html`

[3] *AOL-IM*. Webové stránky služby AOL instant messaging [online] [cit. 2011-12-20] Dostupná z WWW: `http://www.aim.com/`.

[4] *AOL-mail*. Webové stránky služby AOL mail[online] [cit. 2011-12-20] Dostupná z WWW: `https://my.screenname.aol.com/`.

[5] *Apache konfigurace*. Webové stránky eduID.cz věnované integraci aplikací s Shibboleth [online] [cit. 2011-12-20] Dostupná z WWW: `http://www.eduid.cz/wiki/cztestfed/howto/sp/integration/index?s[]=shibboleth&s[]=session`.

[6] *Apache*. Webové stránky serveru Apache [online] [cit. 2011-12-20] Dostupná z WWW: `http://www.apache.org/`.

[7] *Authentication and Authorization for Google APIs*. Webové stránky o autentizaci OAuth vůči službě Google [online] [cit. 2011-12-20] Dostupná z WWW: `http://code.google.com/intl/cs-CZ/apis/accounts/docs/OAuth2Login.html`.

[8] BOREK B.. *Úvod do architektury MVC*. [online] 2011 [cit. 2011-12-20]. Dostupná z WWW: `http://zdrojak.root.cz/clanky/uvod-do-architektury-mvc/`.

[9] *Cesnet*. Webové stránky videokonferencí sdružení CESNET [online] [cit. 2011-12-20] Dostupná z WWW: `https://vidcon.cesnet.cz/videokonference/doc/howto/adobe_connect/index`

[10] *CSS*. Webové stránky serveru Wikipedie věnované problematice CSS [online] [cit. 2011-12-20] Dostupná z WWW: `http://ftp.ics.uci.edu/pub/ietf/html/rfc1866.txt`.

[11] *cURL*. Webové stránky knihovny cURL [online] [cit. 2011-12-20] Dostupná z WWW: `http://php.net/manual/en/book.curl.php`.

[12] *eduPerson*. Webová stránka obsahující specifikaci třídy eduPerson [online] [cit. 2011-12-20] Dostupná z WWW: http://middleware.internet2.edu/eduperson/docs/internet2-mace-dir-eduperson-200806.html.

[13] *eduID.cz*. Webové stránky federace identit eduID.cz [online] [cit. 2011-12-20] Dostupná z WWW: http://www.eduid.cz.

[14] *Evo*. Webové stránky projektu EVO: [online] [cit. 2011-12-20] Dostupná z WWW: http://evo.caltech.edu//

[15] *Facebook*. Webová sociální sítě Facebook [online] [cit. 2011-12-20] Dostupná z WWW: http://www.facebook.com/.

[16] *Gmail*. Webové stránky služby Gmail [online] [cit. 2011-12-20] Dostupná z WWW: https://mail.google.com/mail/help/intl/cs/about.html.

[17] *Google*. Webová stránka služeb Google [online] [cit. 2011-12-20] Dostupná z WWW: http://www.google.cz/intl/cs/about/.

[18] *Google Kalendář*. Webové stránky věnované obecnému sezanámení s aplikací Google Kalendář.[online] [cit. 2011-12-20] Dostupná z WWW: http://code.google.com/intl/cs-CZ/apis/calendar/

[19] *Google Kalendář API*. Webové stránky aplikačního rozhraní Google Kalendář [online] [cit. 2011-12-20] Dostupná z WWW: http://code.google.com/intl/cs-CZ/apis/calendar/.

[20] HAZEL P.. *Exim: the mail transfer agent.*. [online] 2001, O'Reilly & Associates, Inc. [cit. 2011-12-20], ISBN 0-596-00098-7. Dostupná z WWW: http://books.google.cz/books?id=2AVQHCwCBboC\&printsec=frontcover\&hl=cs\#v=onepage\&q\&f=false.

[21] *HTML*. Webové stránky standardu HTML [online] [cit. 2011-12-20] Dostupná z WWW: http://ftp.ics.uci.edu/pub/ietf/html/rfc1866.txt.

[22] *iFrame*. Webová stránka pojednávající o technologii iFrame [online] [cit. 2011-12-20] Dostupná z WWW: http://www.w3.org/TR/html4/present/frames.html#h-16.5.

[23] *IM*. Webová stránka popisující instant messaging [online] [cit. 2011-12-20] Dostupná z WWW: http://wiki.networkdictionary.com/index.php/Instant_Messenging.

[24] *ICQ*. Webové stránky služby ICQ [online] [cit. 2011-12-20] Dostupná z WWW: http://www.icq.com/cz.

[25] *Internet2*. Webová stránka konsorcia Internet2 [online] [cit. 2011-12-20] Dostupná z WWW: http://www.internet2.edu/.

[26] *Jabber*. Webové stránky služby Jabber [online] [cit. 2011-12-20] Dostupná z WWW: http://www.jabber.org/.

[27] *JQuerry*. Webové stránky knihovny jQuery [online] [cit. 2011-12-20] Dostupná z WWW: http://jquery.com/.

[28] *JSON*. Webové stránky věnované standardu JSON [online] [cit. 2011-12-20] Dostupná z WWW: http://www.json.org/

[29] KOUŘIL P., KUBA M., OSOVSKÝ M., PEŠA R., PROCHÁZKA M.. *Federace identit aneb spolčení totožností. Zpravodaj ÚVT MU*. [online] 2007, [cit. 2011-12-20] ISSN 1212-0901, 2007, roč. XVIII, č. 2, s. 1-7 Dostupná z WWW: http://www.ics.muni.cz/bulletin/articles/566.html.

[30] MATUŠKA P.. *VoIP - úvod do problematiky volání po internetu*. [online] 2011, [cit. 2011-12-20] Dostupná z WWW: http://pcworld.cz/internet/voip-uvod-do-problematiky-volani-po-internetu-8120.

[31] *Meebo*. Webové stránky IM klienta Meebo [online] [cit. 2011-12-20] Dostupná z WWW: https://www.meebo.com/.

[32] *Miranda*. Webové stránky IM klienta Miranda [online] [cit. 2011-12-20] Dostupná z WWW: http://www.miranda-im.org/.

[33] *MS Outlook*. Webové stránky programu Microsoft Outlook [online] [cit. 2011-12-20] Dostupná z WWW: http://office.microsoft.com/cs-cz/outlook/.

[34] *MySQL*. Webové stránky databázového systému MySQL: [online] [cit. 2011-12-20] Dostupná z WWW: http://www.mysql.com/.

[35] *MZ Thunderbird*. Webové stránky programu Mozilla Thunderbird [online] [cit. 2011-12-20] Dostupná z WWW: http://www.mozilla.org/cs/thunderbird/.

[36] *OAuth2.0*. Webové stránky protokolu OAuth [online] [cit. 2011-12-20] Dostupná z WWW: http://oauth.net/.

[37] *OpenID*. Webové stránky federace identit OpenID [online] [cit. 2011-12-20] Dostupná z WWW: http://openid.net/.

[38] *Pidgin*. Webové stránky IM klienta Pidgin [online] [cit. 2011-12-20] Dostupná z WWW: http://www.pidgin.im/.

[39] *PHP*. Webové stránky jazyka PHP [online] [cit. 2011-12-20] Dostupná z WWW: http://www.php.net/.

[40] *PV182*. Skripta předmětu PV182 Komunikace člověka s počítačem: Návrh zaměřený na uživatele a prototypování [online] [cit. 2011-12-20] Dostupná z WWW: https://is.muni.cz/auth/el/1433/podzim2011/PV182/um/04prototypingCZ.pdf

[41] *REST*. Více o architektuře REST [online] [cit. 2011-12-20] Dostupná z WWW: http://en.wikipedia.org/wiki/Representational_state_transfer.

[42] *Shibboleth*. Webová stránka middlewaru Shibboleth [online] [cit. 2011-12-20] Dostupná z WWW: http://shibboleth.internet2.edu/.

[43] *UML*. Webová stránka modelovacího jazyka UML [online] [cit. 2011-12-20] Dostupná z WWW: http://www.uml.org/

[44] *URL*. Webová stránka standardu URL [online] [cit. 2011-12-20] Dostupná z WWW: http://tools.ietf.org/html/rfc1738.

[45] *WAYF*. Webová stránka popisující službu WAYF [online] [cit. 2011-12-20] Dostupná z WWW: http://www.switch.ch/aai/support/tools/wayf.html.

[46] *Wikipedia*. Reference na související literaturu na servery Wikipedia [online] [cit. 2011-12-20] Dostupná z WWW: http://en.wikipedia.org/wiki/SAML_2.0.

[47] *XML*. Specifikace pracovní skupiny W3C [online] [cit. 2011-12-20] Dostupná z WWW: http://www.w3.org/XML/.

[48] *Yahoo!*. Webová stránka služeb Yahoo! [online] [cit. 2011-12-20] Dostupná z WWW: http://www.yahoo.com/.

A Obsah přiloženého CD

Níže je uveden seznam souborů obsažených v elektronické příloze této bakalářské práce. Složka `php` a její podsložky obsahují námi implementovanou funkcionalitu nově vyvinutého systému. Soubor `css.css` ve složce `css` definuje grafické elementy a vzhled uživatelského rozhraní. Obsahem složky `sql` jsou skripty sloužící k vytvoření databáze při instalaci systému. Složka `inc` obsahuje námi vytvořené konfigurační skripty.

Výpis obsahu přiloženého CD

- bp.pdf – elektronická verze tohoto textu

Přílohy k bakalářské práci

- Diagram_běžný _uživatel.jpg – jedná se o UML diagram případu užití systému běžným uživatelem
- Diagram_administrátor.jpg – jedná se o UML diagram případu užití systému administrátorem
- Ukázaka_protorypů.pdf – jedná se o jednu sadu prototypů obrazovek
- Specifikace_systemu.pdf – jedná se o strukturovaný popis základní funkcionality systému
- /html – složka obsahjící vytvořené html prototypy několika obrazovek systému

Zdrojový kód systému

- rs/
  - css/
    - /img – obrázky použité při implementaci systému
    - css.css
  - inc/
    - dbSupport.inc
    - rsConstants.inc

    - rsError.inc
    - support.inc

- lib/
    - /google-api-php-client2 – knihovna využívaná ke spolupráci s Google Kalendářem
    - /js – knihovna využívaná k implementaci výběru data z kalendáře

- php/
    - class/
        - cDB.php
        - cDBUser.php
        - cDBNews.php
        - cDBPSrc.php
        - cDBConf.php
        - cPage.php
        - cPageList.php
        - cPageListUser.php
        - cPageListNews.php
        - cPageListPSrc.php
        - cPageListConf.php
        - cPageDetail.php
        - cPageDetailUser.php
        - cPageDetailNews.php
        - cPageDetailPSrc.php
        - cPageDetailConf.php
        - cSession.php
        - cAuth.php
    - script/
        - ctrlConf.php
        - ctrlNews.php
        - ctrlPSrc.php
        - ctrlUser.php
        - login.php

- public/
  - index.html
- sql/
  - rsCreateDatabase.sql
  - rsCreateUser.sql

B Poznámky k vývoji a instalaci systému

Specifikace vývojového prostředí systému

Systém byl vyvíjen v prostředí operačního systému Linux Mint 11 Katya. Použitý programovací jazyk PHP5 využíval knihovny PEAR, cUrl a JSON. Dále bylo využito databázového systému MySQL verze 5.1. Jako vývojový server byl použit Apache verze 2.2.

Grafické rozhraní bylo testováno v prohlížečích Google Chrome verze 15 a Mozilla Firefox for Linux Mint – 1.0.

Inicializační soubor pro vstup do systému je `/public/index.php`. Po spuštění tohoto skriptu je možné z testovacích důvodů zadat libovolné údaje, kterými bude uživatel dále identifikován v systému. Při reálném nasazení systému by musela být tato vstupní obrazovka nahrazena adekvátní přihlašovací obrazovkou.

Veškeré přejaté dodatečné knihovny pro Google Kalendář a jQuery jsou obsaženy v elektronické příloze a není potřeba je speciálně implementovat. Úpravy, které je nezbytné udělat v případě změny v účtu vytvořeném u služby Google pro potřeby rezervačního systému, jsou popsány v elektronické příloze `Autentizace.pdf`.

Aplikace je k otestování dostupná na `https://alcor.ics.muni.cz/rs/public/index.html`

Poznámky k instalaci systému

Při instalaci systému je nezbytné nejprve vytvořit databázi, kterou systém využívá. K tomuto účelu slouží skripty `rsCreateDatabase.sql` a `rsCreateUser.sql`. Dále je při přenosu systému nezbytné upravit některé cesty k existujícím souborům a to i při dodržení adresářové struktury zdrojového kódu.