

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Portál pro DDM

Plzeň, 2012

David Adam

Oficiální zadání

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 17. května 2012, David Adam

Abstract

Portal for DDM

This diploma thesis analyses and implements an information system, which is usable in organizations that work with children and teenagers. The first goal of work is to define requirements for functions. Main criterion of design is effective usage of available information technology and possibility for automate selected processes. The second goal is to implement the prototype of designed portal, create environment for testing and deploy application to prove usability of final solution.

Obsah

1	ÚVOD	1
2	HISTORIE PROJEKTU	2
3	REALIZAČNÍ PROSTŘEDKY	3
3.1	OPERAČNÍ SYSTÉM.....	3
3.1.1	Microsoft Windows XP.....	3
3.1.2	Windows Server 2003	3
3.1.3	Windows server 2008 R2	3
3.1.4	CentOS 6	4
3.2	DATABÁZE.....	4
3.3	VÝVOJOVÉ PROSTŘEDÍ.....	4
3.4	SPRÁVA A PROVOZ PROJEKTU	5
3.4.1	Redmine	5
3.4.2	Apache Subversion.....	5
3.4.3	GlassFish.....	5
3.5	GRAFIKA.....	6
3.5.1	Základní grafika	6
3.5.2	Ikony	6
3.5.3	Bannery	6
3.6	PROGRAMOVACÍ JAZYKY	7
3.6.1	Java Platform, Enterprise Edition.....	7
3.6.2	XHTML, CSS	10
3.6.3	JavaScript	11
3.7	JAVASCRIPTOVÉ FRAMEWORKY	12
3.7.1	JAK	12
3.7.2	jQuery.....	12
3.7.3	Yahoo! User Interface Library	13
3.8	LICENCE	14
3.8.1	Berkeley Software Distribution (BSD)	14
3.8.2	Massachusetts Institute of Technology (MIT).....	14
3.8.3	Lesser General Public License (LGPL).....	14
3.8.4	Creative Commons (CC).....	15
3.8.5	GNU General Public License (GPL).....	16
3.8.6	Apache Licence	16
4	SPECIFIKACE POŽADAVKŮ A TECHNOLOGIE	17
4.1	PREZENTACE ORGANIZACE	17
4.1.1	Web a rozesílání emailových zpráv	17
4.2	AGENDA	19
4.2.1	Docházkové akce	19
4.2.2	Výkaz	20
4.3	SPRÁVA	21
4.3.1	Servis.....	21
4.3.2	Vedoucí – správa učeben.....	22
4.3.3	Administrace zaměstnanců.....	22
4.4	NÁPOVĚDNÉ A NAVIGAČNÍ PROSTŘEDKY	23
4.4.1	Dynamická menu	23
4.4.2	Formulářové a tabulkové prvky	23
4.4.3	Lokalizace	24
4.4.4	Oznámení o provedení akce	24
4.5	ACTIVE DIRECTORY.....	24
5	PROGRAMÁTORSKÁ DOKUMENTACE	27
5.1	ARCHITEKTURA APLIKACE.....	27
5.2	ADRESÁŘOVÁ STRUKTURA PROJEKTU.....	28
5.3	OBJEKTY A METODY	28
5.3.1	ADAdm.....	28
5.3.2	Hwutils.....	30

5.3.3	MailUtils	31
5.3.4	Gallery a GalleryAdmin	31
5.4	FORMULÁŘE	32
5.4.1	Multipart.....	32
5.4.2	Poslední zapsaný index	33
5.5	ŘÍZENÍ OPRAVNĚNÍ	33
5.5.1	Autentizace uživatele	33
5.5.2	Autorizace uživatele.....	34
5.5.3	Proces vyhodnocení oprávnění.....	35
5.6	JAVA NAMING AND DIRECTORY INTERFACE.....	36
5.7	DIAGRAM NASAZENÍ.....	37
5.8	ERA MODEL	38
5.9	MONITOROVÁNÍ SÍTĚ	41
5.9.1	Volání funkce ping přes příkazový řádek Windows jako runtime	41
5.9.2	Využití funkce isReachable()	41
5.9.3	Implementace ICMP	42
5.9.4	Porovnání výkonnosti.....	42
5.10	DEPLOY DESCRIPTOR	43
6	NASAZENÍ SYSTÉMU	44
6.1	INSTALACE POTŘEBNÝCH APLIKACÍ.....	44
6.2	INSTALACE CERTIFIKÁTU	44
6.3	VYTVOŘENÍ POTŘEBNÝCH ADRESÁŘŮ, ÚPRAVA DEPLOY DESCRIPTORU.....	45
6.4	OMEZENÍ FUNKCÍ NA OS LINUX	45
7	TESTOVÁNÍ	46
7.1	HARDWAROVÉ NÁROKY	46
7.1.1	Server	47
7.1.2	Klient.....	47
7.2	VERZE SOFTWARE.....	47
7.3	TESTOVÁNÍ KOMPATIBILITY	48
7.4	TESTOVÁNÍ POUŽITELNOSTI.....	48
7.5	METRIKY	49
7.6	ZHODNOCENÍ VÝSLEDKŮ	49
8	ZÁVĚR.....	50
	POUŽITÉ ZKRATKY A SLOVNÍK POJMŮ	51
	LITERATURA	52
	PŘÍLOHY.....	56

1 Úvod

S rozvojem internetu, rozšířením technologií do společnosti a celkovou vyšší dostupností služeb v oblasti vzdělání, se zásadním způsobem mění funkce základního školství a organizací pracujících s dětmi a mládeží, obecně nazývanými Domy dětí a mládeže. Tyto instituce, které dříve fungovaly na společenském modelu, kdy navštěvování tzv. kroužků (dnes se používá termín zájmové útvary) bylo samozřejmostí, musí dnes řešit problém konkurenčního boje, obecného nedostatku dětí a z toho plynoucího snižování počtu úvazků. Jedním z předních důvodů úpadku je kritický stav vzdělání pedagogů i managementu v oblasti počítačové gramotnosti, propagace a informačních technologií. Důsledkem je často nedostatečná organizace práce, vedení papírových matrik a neexistující agenda. Propagační funkce je zastávána statickým webem poskytujícím souhrnné informace o činnosti organizace.

Tato diplomová práce se zabývá analýzou a implementací modelu portálové aplikace, která by významným způsobem omezila výše uvedené nedostatky. V první části textu je rozebrán výběr vhodných prostředků a nástrojů k realizaci projektu, jsou zhodnoceny a popsány klíčové funkce k implementaci a technologie zvolené k integraci do portálu. Většina realizovaných služeb spadá do kategorií:

- základní monitoring počítačové sítě,
- administrace uživatelských účtů v Active Directory,
- správa uživatelů informačního systému a agendy,
- webová prezentace.

Následuje popis licencí, pod kterými je většina užitého software publikována, rozebrány jejich podmínky a specifika. Výběr nejvolněji definovaných licencí dává možnost s dílem rozumně nakládat při jeho případném šíření.

V realizační části práce jsou popsány hlavní metody aplikace, způsob jejich realizace a řešené problémy. Samostatné kapitoly jsou věnovány nejdůležitějším procesům v aplikaci, jakými jsou například metody autorizace a autentizace, či komunikace s Active Directory.

Na konci práce se nachází popis metodik testování aplikace a specifikace testovacího prostředí projektu. Pro získání představy o rozsahu projektu je uvedeno několik metrik.

V příloze je uveden text nezbytné uživatelské dokumentace a ukázky rozhraní.

2 Historie projektu

Nápad vytvořit aplikaci, která by pomocí webového rozhraní usnadňovala některé rutinní úkony ve školách a organizacích pracujících s dětmi, byl rozvíjen a realizován od mého nástupu na magisterské studium. Protože jde o zajímavou myšlenku, povedlo se na velkém množství oborových předmětů nechat vypsát individuální semestrální práce, což poskytlo čas a informační zdroje pro vývoj základů částí projektu, které do daného předmětu spadaly.

Přenos dat

Podstatou semestrální práce v předmětu *KIV/PD* bylo prostudování možnosti navázání komunikace s adresářovou službou Windows Serveru pomocí Javy. Následně pak vypracování jednoduchého nástroje, který uživateli umožní vytváření a mazání kont v Active Directory (viz kapitola 4.5), výpis všech záznamů v zadaném umístění, jejich řazení apod. Tato práce se stala základem autentifikačního modulu portálu a servletu pro správu kont návštěvníků technických zájmových útvarů.

Programování internetových aplikací

Individuální semestrální práci v předmětu *KIV/PIA* začala konstrukce prototypu webového rozhraní portálu. Vytvořen byl základ algoritmu autorizace uživatele a první verze rozhraní na správu oprávnění. Součástí práce byl také layout aplikace a zpracování grafických prvků webové stránky (záhlaví, zápatí, menu atd.). Výstupem byl specifický redakční systém, který diplomová práce ke svým potřebám dále rozvíjí.

Pokročilé softwarové inženýrství

Mou náplní předmětu *KIV/ASWI* bylo řízení týmu čtyř lidí, jehož úkolem byla analýza a implementace digitálního docházkového systému, jako náhrada docházkové knihy. Nápad se osvědčil, ale jelikož se následně zvolená konstrukce ukázala jako nevyhovující, byla z původní analýzy přepracována implementace jako součást této diplomové práce.

Databázové systémy 2

V rámci předmětu *KIV/DB2* bylo zpracováno základní schéma databáze, tabulky prezentační části webu a jejich management. Aplikace díky tomu disponuje databázovou vrstvou, která přispívá k přehlednosti a udržitelnosti kódu. Užity byly také SQL rutiny a triggerů.

3 Realizační prostředky

V této kapitole jsou popsány prostředky a nástroje, které jsou využity při správě projektu, vývoji a testování. U každého software lze získat představu o tom, proč byl do projektu zapojen a je-li to možné, jsou uvedeny i další alternativy.

3.1 Operační systém

Pro potřeby vývoje a testování aplikace jsou nainstalovány níže uvedené operační systémy (dále jen OS). Microsoft Windows Server 2003 je nasazen fyzicky na hardware v testovacím prostředí (více v kapitole 7), ostatní OS jsou provozovány virtualizovaně.

3.1.1 Microsoft Windows XP

Tento OS je zvolen pro jeho, v této době ještě stále vysokou, rozšířenost. Vzhledem k faktu, že mnoho cílových organizací nemá kapacity a finanční zdroje na nasazení edice Windows Server, je velmi pravděpodobný provoz aplikace na upraveném stolním PC¹ s Windows XP SP3. Nasazen v rámci projektu je tedy hlavně z důvodu testování kompatibility s neserverovou verzí OS.

3.1.2 Windows Server 2003

Pro účely testování zpětné kompatibility je do testovacího prostředí zahrnuta i starší verze Windows Server, která je v současné době nasazena přibližně ve 40 procentech případů. Na tomto operačním systému probíhá hlavně testování součástí pracujících s *Active Directory* [1] (dále jen AD), také je zde nasazen server *Glassfish* (viz kapitola 3.4.3) pro produkční testování.

3.1.3 Windows server 2008 R2

Ač nebyl nalezen případ, kde by byl nainstalován Windows server ve verzi 2008, dá se předpokládat, že postupně se právě tento OS stane primárním. Důvodem, proč dnes není nasazen, je ve většině případů situace, kdy již má organizace nějakou funkční infrastrukturu a bez zisku, který by aktualizace přinesla, si ji nemůže dovolit. Rozšířenější nasazení se dá předpokládat vlivem stárnoucího hardware. Tento OS je pro potřeby projektu provozován jako primární vývojový server, instalováno zde bylo centrální úložiště zdrojových kódů (viz kapitola 3.4.2) a projektový manažer *Redmine* (viz kapitola 3.4.1).

¹ Z praxe vyplývá, že 80% strojů pracujících v DDM v serverovém nasazení jsou běžné stolní počítače, opatřené UPS a disponující dostatečným výpočetním výkonem.

3.1.4 CentOS 6

Tato linuxová distribuce, vycházející z enterprise řešení *Red Hat Linux* [2], je v rámci projektu provozována pro účely testování možností nasazení aplikace pod systémem Linux. Ač se s tímto řešením v praxi příliš nepočítá, přinese analýza této možnosti důležité informace o službách portálu závislých na platformě Windows.

3.2 Databáze

Jako databáze je zvolena *MySQL 5*, důvodem je její vysoké rozšíření. Organizace, které nějakou formu databáze provozují, mají v naprosté většině nasazeny *MySQL*. Často není databáze nasazena jako doplněk webu, ale jako podpora pro zcela jiný systém, například monitorovací systém (v praxi se často využívá *Zabbix*), zálohovací systém (např. *Bacula*) apod. Další vlastnosti, díky kterým je pro tento projekt nejvhodnější, jsou:

- cena
- dostačující robustnost
- jednoduchost správy a nasazení

Databází využitelných v tomto projektu je samozřejmě více, za všechny je možné zmínit *Oracle Database 11g Express Edition* či *PostgreSQL* (také nazývána *Postgres*), které jsou též nabízeny pod různými bezplatnými licencemi (např. MIT pro *PostgreSQL*). Kromě výše zmíněných výhod *MySQL* nebyly tyto zvoleny z důvodu větších HW nároků, silných omezení v nasazení a měnících se podmínkách licence. Verze *Oracle 11g EE* je nastavena na využití maximálně 1GB RAM, 1CPU a 11GB uživatelských dat [3]. *PostgreSQL* je databáze vyvíjená pro platformu Linux, ač existují i balíčky pro win32.

3.3 Vývojové prostředí

Vývojové prostředí (dále jen IDE) bylo voleno mezi řešením *Netbeans 7* [4] a *Eclipse 3.6.2* [5]. Prostředí *Eclipse* bylo nakonec zavrženo z důvodu vyšších HW nároků při delším běhu a méně intuitivnímu rozhraní. *Netbeans* disponují velmi robustním a propracovaným prostředím, které vývojáři nabízí:

- Podporu vývoje v J2EE prostřednictvím propojení na *Glassfish*² server.
- Integraci verzovacího nástroje *Subversion* (viz kapitola 3.4.2).

²Tento aplikační server je zvolen pro menší HW nároky, volitelně lze provozovat i servery *Apache Tomcat* [28] či *Red Hat JBoss* [29].

- Přímou podporu několika nejrozšířenějších frameworků³ (Spring Web MVC, Struts, JavaServer Faces, Hibernate...).

3.4 Správa a provoz projektu

3.4.1 Redmine

Software pro řízení projektu a bug tracking systém⁴. Je napsán za použití *Ruby on Rails* (dále jen RoR) frameworku, pracuje na více platformách i databázích. Vydán jen pod licencí GPL. Redmine podporuje správu více projektů, definuje uživatelské role pro přístup k nim a na základě autorizace nabízí uživatelům moduly Ganttova diagramu⁵, kalendáře, správu úkolů a další nástroje pro správu projektu. Integrovat lze i softwarová úložiště, jako *SVN*, *CVS* či *GIT*. Funkčnost je možno rozšiřovat použitím pluginů [6].

V projektu je Redmine nasazen pro řízení vývoje. Užíván je jako centrální úložiště textů, pro vizualizaci stromu *SVN*, pro správu verzí a evidenci chyb a funkcí.

3.4.2 Apache Subversion

Dříve známý jako *Subversion* nebo *SVN* je nástroj na verzování zdrojových kódů. Svou funkčností a stylem vychází z *CVS*, jehož nedostatky se snaží opravit. *Subversion* je k dispozici pro mnoho platforem, včetně *Windows*.

Projekt využívá tento verzovací systém z důvodu jeho přímého propojení na dva další nástroje, projektový manažer *Redmine* a vývojové prostředí *Netbeans*.

3.4.3 GlassFish

Aplikační server vyvinutý společností *Sun Microsystems* v roce 2005. Důvodem pro vznik bylo vytvořit plně certifikovaný *Java* server (viz kapitola 3.6), který by mohl sloužit jako referenční implementace. *Glassfish* je možno provozovat na platformě *Linux* i *Windows*, aktuálně je ve verzi 3.1.2 a plánuje se verze 4, která bude, stejně jako verze aktuální, řešena jako open-source⁶.

Pro tento projekt byl server vybrán pro nízké HW nároky a snadnou obsluhu.

³ Pojmem framework je v kontextu vývoje software myšlen ucelený kód (softwarová knihovna), který tvoří architekturu určité části aplikace (viz kapitola 3.7)

⁴ Softwarová aplikace, která pomáhá zajišťovat kvalitu aplikace sledováním hlášených chyb a jejich řešení vývojáři.

⁵ Vizualizace posloupnosti činností v čase, používaných při plánování projektů.

⁶ Open-source znamená, že k počítačovému softwaru je dostupný jak zdrojový kód, tak licence, která při dodržení určitých podmínek uživateli umožňuje jeho užití a modifikace.

3.5 Grafika

V rámci implementace projektu bylo pro testovací provoz vytvořeno množství grafických prvků a použita grafická knihovna.

3.5.1 Základní grafika

Pro tvorbu layoutu testovacího nasazení pro SMT (popsáno v kapitole 7) byl použit software *CorelDRAW Graphics Suite X4* [7] a *Adobe Photoshop Elements 7.0* [8], na které má tato organizace licenci, stejně jako na použitá loga. Surové podklady byly fotograficky vytvořeny vlastní a k jejich zpracování byl využit tablet.

3.5.2 Ikony

Pro doplnění základní grafiky byl zvolen balík ikon *Cristal Clear* [9], jehož autorem je Everaldo Coelho a který je šířen pod licencí LGPL (viz kapitola 3.8.3). Tato licence je určena pro softwarové knihovny a její použití pro grafiku je silně nešťastné řešení, neboť se může snadno dostat do sporu. Proto bylo věnováno mnoho času jejímu studiu a způsobu nasazení do portálu, aby byla dodržena.

3.5.3 Bannery

V portálu se nachází i několik ikon sociálních sítí, různých zájmových skupin a organizací. Všechny tyto bannery jsou šířeny volně bez jakýchkoliv restrikcí.

3.6 Programovací jazyky

V následující kapitole jsou popsány použité programovací jazyky, značkové jazyky a technologie pro tvorbu a formátování webového obsahu. Závěr kapitoly je věnován technologiím, jejichž použití by bylo vhodné při dalším vývoji projektu.

3.6.1 Java Platform, Enterprise Edition

Jako hlavní vývojový jazyk byla zvolena Java ve verzi *Enterprise Edition* (dále označována jako J2EE), tedy platforma určená pro tvorbu informačních systémů a podnikových aplikací. J2EE je vyvíjena od roku 2000 firmou Sun Microsystems.

Java Servlety

Jsou ekvivalentem k využívání *Common Gateway interface* (dále jen CGI) a na webovém serveru zastávají funkci střední vrstvy, tedy leží mezi prezentačním prvkem a třídami obsluhujícími databázi. Hlavními úkoly servletu je zpřístupnit data zadaná uživatelem (nejčastěji do formuláře), doplnit další informace z HTTP požadavku a na základě těchto hodnot vytvořit výsledky, které jsou předány ke zformátování technologii prezentační vrstvy, kterou v základu JSP [10]. Cyklus zpracování požadavku je ilustrován na obrázku 3.1.

Hlavními výhodami použití servletů oproti CGI jsou:

- **Vyšší účinnost**

Rozdíl mezi užitím java servletů a CGI v oblasti výkonosti je v systému obsluhy více požadavků. V případě CGI je pro N současných požadavků na stejný program, vytvořen N -krát nový proces (každý s vlastním zavedením kódu do paměti) a po ukončení je program z paměti opět odstraněn. Tento přístup je výrazně nevýhodný v případě, kdy je samotné vykonávání programu krátké a administrativa spouštění procesu (které výrazně zatěžuje operační systém) může být delší než jeho samotné provádění. Servlety jsou zpracovávány ve virtuálním stroji Java pomocí lehkého java vlákna (v paměti je pouze jediná kopie třídy servletu) a zůstávají v paměti i po obsluze požadavku, což umožňuje sdílet data mezi jednotlivými dotazy.

- **Vhodnost**

Servlety disponují rozsáhlými kolekcemi knihoven a infrastrukturou navrženou pro práci s webovým obsahem (nastavování záhlaví HTML, obsluha session a

cookies, analýza formulářových dat...), navíc jsou psány v jazyce Java a poskytují tak všechny výhody OOP⁷.

- **Výkonnost**

Mezi schopnosti, kterými obecně CGI vůbec nedisponují (bez použití speciálního serverového API, které vývoj zásadně komplikuje) patří komunikace se serverem. Ta umožňuje například sdílení zdrojů mezi servlety, kešování výpočtů, předávání informací mezi požadavky a podobné optimalizace.

- **Přenositelnost**

Protože jsou servlety psány v programovacím jazyce Java (s použitím standardizovaného API), je možné je provozovat na různých webových serverech. Webovou aplikaci napsanou pro Apache Tomcat je možno s žádnými, či minimálními úpravami napsat např. na IBM WebSphere či GlassFish.

- **Bezpečnost a ošetření kódu**

Po výkonnosti je hlavní nevýhodou CGI technologie zranitelnost. S použitím univerzálních příkazových interpretů operačního systému je třeba řešit činnosti jako filtrování speciálních znaků (uvozovky, středníky...), práce s pamětí a přístupem do ní apod. (kontroly délky řetězců nebo polí nejsou v některých jazycích prováděny automaticky). Neprovádění těchto kontrol pak představuje zásadní slabinu v bezpečnosti. Servlety žádným z těchto problémů netrpí.

Java Server Pages

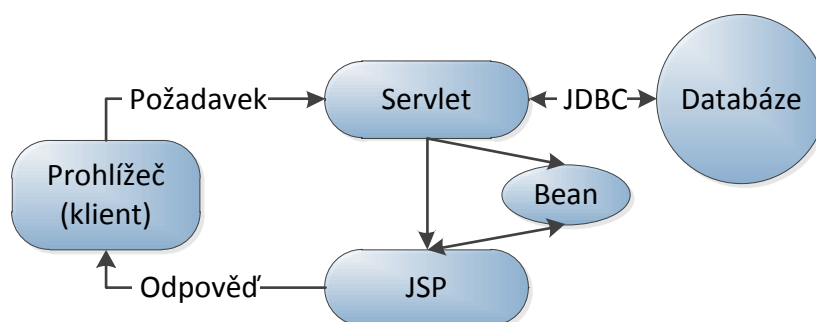
Dále označována JSP, je technologie pro dynamické generování webových stránek založených na HTML. Je součástí J2EE od počátku vývoje jako prezentační vrstva pro aplikace. V JSP dokumentu je možno kombinovat syntaxe Java, HTML, JavaScript a XML a používat skriptovací značky HTML `<element>` a JSP `<% element %>`. Tento přístup umožňuje vývojáři směšovat staticky a dynamicky generovaný obsah a vytvářet tak šablony pro formátování výstupu servletů.

⁷ Objektově orientované programování poskytuje opakovaně použitelný a spolehlivý kód, s využitím zapouzdření, dědičnosti a dalších možností. Třídy jsou transparentnější, poskytují lepší možnosti testování.

Oproti alternativám JSP nabízí:

- **Oproti PHP: Hypertext Preprocessor** – rozsáhlé API pro práci se zdroji a daty, psaní dynamické části v jazyce Java a snadné rozšiřování definovaných tagů pomocí knihoven (např. JSTL viz dále).
- **Oproti ASP** – kód přenositelný mezi různými platformami (ASP je vázáno na webový server a operační systém firmy Microsoft).
- **Oproti SSI** – nástroje na vytváření vnější části obsahu a práci s daty. SSI je technologie pro vkládání definovaných částí do statické stránky a pro velké projekty se nehodí.

K robustnímu vývoji prezentační vrstvy aplikace za použití JSP navíc silně přispívá JavaServer Pages Standard Tag Library (dále jen JSTL), nabízející sadu předem připravených knihoven tagů. Ty jsou členěny do kategorií Core, Formating, XML a SQL a umožňují eliminovat použití scriptletů v zápisu JSP.



Obrázek 3.1: Zpracování požadavku v J2EE.

Platforma disponuje i mnoha dalšími prvky a frameworky, usnadňujícími různé oblasti vývoje. Jejich integrace do projektu nebyla aktuálně realizována z důvodu nemožnosti získat zkušenosti s implementací během vývoje, avšak do budoucna se počítá s jejich začleněním. Mezi nejpoužívanější patří:

- **Spring**
Spring framework je aplikační rámec vydaný v roce 2002 Rodem Johansonem pod licencí Apache 2.0. Důvodem vzniku byla abstrakce dílčích součástí J2EE, více možností implementace *bussiness vrstvy*⁸ a implementace návrhového

⁸ Vrstva s logikou aplikace, na které se provádí výpočty, vyhodnocení dat, autentifikace atd. V případě vzoru MVC se zde nachází model.

vzoru *Inversion of Control*⁹. Spring je postaven jako modulární framework, díky čemuž lze v aplikaci využít jen potřebnou funkčnost. Jednotlivé moduly jsou organizovány do skupin Web, Core Container, Data Access/Integration, Instrumentation, AOP (Aspektově orientované programování) a Test.

- **JavaServer Faces**

Je technologie (dále označovaná JFS) vytvořená firmou Sun Microsystems, Inc, jejímž účelem je komplexnější tvorba uživatelského rozhraní a čistější oddělení definice uživatelského rozhraní od logiky. V případě JSP je stránka generována odshora dolů s cílem vytvořit výstup pro uživatele. Životní cyklus JFS je více rozfázován a dochází v něm k budování stromu komponent, validaci uživatelských vstupů, jejich konverzi a renderování komponent.

- **Hibernate**

Framework pro práci s databází pomocí modelu objektově-relačního mapování (ORM). Projekt je aktuálně vyvíjen firmou Red Hat, a implementuje mapování objektů na entity v relační databázi. Umožňuje zachování dat z objektů i po skončení aplikace – udržuje data persistentní. Pro komunikaci s objekty je použit jazyk HQL (jehož syntaxe je podobná SQL, navíc rozšířena o vlastnosti OOP), čímž je odstíněna použitá databáze využitím Hibernate API.

3.6.2 XHTML, CSS

Extensible HyperText Markup Language je značkovací jazyk vyvíjený *World Wide Web Consortium* (W3C). Jazyk vychází z HTML, jehož účelem je popsat obsah a strukturu dokumentu. Protože je zápis HTML obecně známý, bude se následující odstavec zabývat rozdíly mezi HTML a XHTML. V projektu je XHTML použito pro tvorbu výstupu aplikace. Generováno je prostřednictvím JSP (popsané výše).

Původní záměr nahradit standardní HTML (které mělo být verzí 4.01 ukončeno) byl v roce 2007 revidován a nyní jsou HTML 5 a XHTML vyvíjeny paralelně. Důvodem vzniku XHTML byly volně definované specifikace HTML, vedoucí k rozdílným implementacím v jednotlivých prohlížečích a tím i k různé interpretaci zapsaného kódu. XHTML je restriktivnější a přidáním přesných pravidel a specifikací se snaží docílit jednotného zobrazení. Hlavními rozdíly jsou:

- Všechny elementy musí být ukončené, včetně tzv. nepárových.

Např: zkrácené `` i celé ``

⁹ Návrhový vzor pracující na principu přesunu odpovědnosti za vytváření objektů z aplikace na framework. Vsazování objektů je řešeno prostřednictvím Dependency Injection.

- Dokument musí být uvozen XML deklarací (kromě kódování UTF8) a DTD specifikací.
- Elementy i atributy se zapisují malými písmeny (XHTML je case sensitive), lze však definovat vlastní elementy v DTD, které velká písmena obsahovat mohou.
- Hodnoty atributů se vždy zapisují do uvozovek, včetně jednoslovných.

Také množina použitelných elementů je redukována na takové, které svou funkcí a definicí nekolidují s jinými. Podle míry omezení je tak XHTML dále děleno na verze *Strict*, *Transitional* a *Frameset*.

CSS (kaskádové styly) je značkovací jazyk, jehož účelem je specifikování zobrazení HTML entit, čímž dochází k oddělení obsahu dokumentu od vzhledu. CSS je navržen W3C a aktuálně je vydána verze CSS3.

3.6.3 JavaScript

Multiplatformní objektově orientovaný programovací jazyk, vyvinutý firmou Netscape jako doplněk jazyka HTML. JavaScript je interpretován na straně klienta a jeho základní funkcí je tvorba dynamických prvků v HTML dokumentu. Syntaxí je podobný jazykům C či Java a je taktéž case sensitive. Zapisuje se přímo do HTML dokumentu (kód je odesílán serverem v odpovědi). Jeho použití lze v prohlížeči zakázat, či nemusí být v prohlížeči podporován, což je třeba zohlednit při návrhu webu. [11]

V dnešní době jsou masivně využívány Javascriptové frameworky (viz kapitola 3.7), mezi nejvýznamnější patří *JQuery JS Library*, *Dojo JS Toolkit*, *Prototype JS Library* a *Yahoo UI Library*.

3.7 Javascriptové frameworky

Při vývoji aplikace bylo využito množství knihoven a frameworků, které projektu umožnili rozumný vývoj a funkčnost, při zachování vysoké stability.

Pojmem framework je v kontextu softwarového vývoje myšlena struktura, sloužící jako podpora při programování. Typicky zastřešuje množinu funkcí z dané oblasti a poskytuje k nim přístup na vyšší úrovni. Hlavní výhodou frameworků je poskytnutí API, které zastřešuje přístup k nižším službám, což v konečném důsledku snižuje množství chyb v kódu, zrychluje vývoj a zpřehledňuje vlastní kód aplikace. Častá nevýhoda frameworků je v jejich velikosti, která je dána univerzálností kódu. Obecně pak lze též mluvit o neefektivitě výsledného řešení. S rostoucím rozsahem projektu se však tato negativa vytrácí.

Aktuálně použité JavaScriptové frameworky jsou z větší části implementovány pro zlepšení grafického rozhraní aplikace (kontextová menu, dynamické načítání obrázků, odpočty času...), popř. pro rozšíření možností daných webovou aplikací (výběr více souborů naráz při odesílání formulářem).

3.7.1 JAK

Objektově orientovaný, minimalistický, JavaScriptový framework vyvíjený firmou Seznam.cz, a.s. Cílem implementace je zjednodušení a standardizace často se opakujících konstrukcí. Funkcionalitu zapouzdřuje do vlastního jmenného prostoru, nemodifikuje základní běhové prostředí JavaScriptu. Je šířen pod licencí MIT [12].

Do projektu byl začleněn z důvodu budoucího využití funkcí pro práci se seznamy.

3.7.2 jQuery

jQuery je JavaScriptová knihovna, vydaná v roce 2006 Johnem Resigem. Šířena je pod duální licencí MIT a GPL. Principem fungování jQuery je oddělování HTML struktury od jejího chování. Celý kód knihovny se nachází v jednom souboru. Výhodou tohoto řešení je velice snadná integrace do projektu, nevýhodou fakt, že je třeba stáhnout při načtení první stránky i celou knihovnu, což v aktuální verzi 1.7 znamená 92kB dat. [13]

Nejčastější použití jQuery při výsledném vývoji vlastní aplikace je formou tzv. pluginů, neboli drobných jednoúčelových aplikací. V rámci projektu portálu pro DDM byly integrovány pluginy:

- **AD Gallery** [14]
Plugin pro vizualizaci obrázků formou slideshow.
David Miles, licence MIT
- **FancyBox** [15]
Plugin pro vizualizaci obrázků a multimédií formou plovoucích oken.
fancybox.net, licence MIT a GPL
- **LanguageSwitcher** [16]
Plugin pro tvorbu rozbalovací rolety, primárně určený k nastavení jazyka.
Kevin Potts, licence WTFPL¹⁰
- **MaxLength** [17]
Plugin pro vizualizaci omezení vstupních polí formuláře.
Emil Stjerneman, licence GPL
- **ToolTip** [18]
Plugin pro formátování bublinových zpráv.
Jörn Zaefferer, licence MIT/GPL

3.7.3 Yahoo! User Interface Library

Jinak také označována jako YUI je rozsáhlá JavaScriptová knihovna z dílny společnosti Yahoo! vyvíjená od roku 2005. YUI je šířena pod licencí BSD. Specializací knihovny je tvorba velice bohatých a komplikovaných grafických prvků, za použití technologií AJAX, DHTML a práce s DOM¹¹. Aktuální je stabilní vydaná verze 3.4. V projektu je začleněna verze YUI2, protože v době implementace tvořila nejvyšší stabilní vydání. Knihovna je členěna do mnoha funkčních bloků, součástí vydání jsou i prototypy konstrukcí, které je možno využít jako základ vlastní implementace. [19]

V projektu byly využity prototypy funkcí:

- **Uploader** – okno pro výběr a nahrávání více souborů paralelně.
- **DataTable** – vizuální prvek pro tvorbu dynamických tabulek.
- **TabSheet** – vizuální prvek pro tvorbu dynamických záložek.

¹⁰ Licence *WTFPL* (Do What The Fuck You Want To Public License) je řídko užívaná licence, obecně užitá pokud autor o licencování nemá zájem.

¹¹ *HTML DOM* představuje standardní způsob zpřístupnění objektů HTML dokumentu a práci s nimi. Každý validní HTML dokument je přístupný ve formě tzv. DOM stromu. AJAX je technologie pro změnu obsahu webové stránky bez nutnosti opětovného načtení celého dokumentu.

3.8 Licence

Vzhledem k rozsahu projektu bylo při vývoji použito několik softwarových knihoven, grafických balíčků a jiných děl, které jsou autory šířeny pod různými licencemi. Bylo proto provedeno jejich nastudování, zjištěna omezení a problematické pasáže a byly vyloučeny ty, jejichž restrikce by v budoucnu mohly být rizikové. Následující kapitola obsahuje seznam nejčastěji použitých licencí, v bodech shrnuje jejich specifika. Pro přehlednost jsou odkazy na původní texty uvedeny zde, nikoliv v bibliografii.

3.8.1 Berkeley Software Distribution (BSD)

- Zcela volná
- Nutno uvést autora a licenční podmínky
- Netřeba distribuovat kód

Originální text: <http://www.opensource.org/licenses/bsd-license.php>

Český překlad: http://cs.wikipedia.org/wiki/BSD_licence

Integrovaný software pod touto licencí:

- Yahoo! User Interface Library (viz kapitola 3.7.3)

3.8.2 Massachusetts Institute of Technology (MIT)

- Volná, podobná BSD licenci, méně precizní
- Vztahuje se i na dokumentaci
- Navíc uvedena bezúplatnost poskytované podlicence

Originální text: <http://www.opensource.org/licenses/mit-license.php>

Český překlad: http://cs.wikipedia.org/wiki/Licence_MIT

Integrovaný software pod touto licencí:

- JAK – javascriptová knihovna (viz kapitola 3.7.1)
- jQuery (viz kapitola 3.7.2)

3.8.3 Lesser General Public License (LGPL)

- Umožnit změnu knihovny jejím dynamickým linkováním
- Permisivnější verze GNU GPL

Originální text: <http://www.gnu.org/licenses/lgpl.html>
























Český překlad: http://cs.wikipedia.org/wiki/GNU_Lesser_General_Public_License

Integrovaný software pod touto licencí:







- Crystal Clear icon set [9]

3.8.4 Creative Commons (CC)

Licence CC je mezinárodně populární licence určená pro multimediální obsah, jejíž základ je postaven na tzv. licenčních prvcích [20], jejichž kombinací si autor definuje nastavení CC licence pro své dílo. Obecně lze vybírat jednu z šesti kombinací, které zobrazuje tabulka 3.1.

Označení licence	Práva	Povinnosti	Popis licence
BY	 		<ul style="list-style-type: none">• Uveďte autora
BY-SA	 	 	<ul style="list-style-type: none">• Uveďte autora• Zachovejte licenci
BY-ND		 	<ul style="list-style-type: none">• Uveďte autora• Nezasahujte do díla
BY-NC	 	 	<ul style="list-style-type: none">• Uveďte autora• Neužívejte dílo komerčně
BY-NC-SA	 	  	<ul style="list-style-type: none">• Uveďte autora• Neužívejte dílo komerčně• Zachovejte licenci
BY-NC-ND		  	<ul style="list-style-type: none">• Uveďte autora• Neužívejte dílo komerčně• Nezasahujte do díla

Tabulka 3.1: Typy CC licencí.

	Právo dílo šířit
	Právo dílo upravovat
	Uveďte autora
	Neužívejte dílo komerčně
	Zachovejte licenci
	Nezasahujte do díla

Jednotlivé kombinace jsou v tabulce seřazeny od nejotevřenějších po nejrestriktivnější. Všechny licence dovolují šíření díla a musí být vždy uveden autor.

Dále je třeba přidat referenci na CC licenci (URL adresu). Licence jsou neodvolatelné a zanikají v případě narušení podmínek nabyvatelem.

Originální text: <http://creativecommons.org/licenses/>

Český překlad: http://cs.wikipedia.org/wiki/Creative_Commons

Integrovaný software pod touto licencí:

- Zamýšleny různé grafické knihovny, nakonec nepoužito.

3.8.5 GNU General Public License (GPL)

- Základem licence je filosofie: vše, co využívá sw licencovaný pod GPL, musí být s touto licencí kompatibilní.

Pročtení dostupných zdrojů naznačuje, že nasazení sw pod touto licencí k podpoře vlastní aplikace, navíc bez zahrnutí do vlastní distribuce či instalace, není odvozením.

Originální text: <http://www.gnu.org/licenses/gpl.html>

Český překlad: http://cs.wikipedia.org/wiki/GNU_General_Public_License

Integrovaný software pod touto licencí:

- jQuery [13]
- Glassfish

3.8.6 Apache Licence

Originální text: <http://www.apache.org/licenses/>

Použitý software pod touto licencí:

- Apache Subversion

4 Specifikace požadavků a technologie

V následující kapitole je definována funkčnost jednotlivých modulů. Popsány jsou všechny funkční požadavky na moduly, mimofunkční požadavky na portál a technologie, které jsou s implementací spojeny.

4.1 Prezentace organizace

4.1.1 Web a rozesílání emailových zpráv

Jedním z neúčinnějších a nejlevnějších způsobů propagace organizací je prezentační web a rozesílání vyžádaných emailových zpráv. V závislosti na prioritách návštěvníka může být systém užít k distribuci novinek, informací o připravovaných akcích či táborech, o nově otevřených zájmových útvarech apod.

Pokud organizace realizují odesílání hromadných emailů, bývá pro tento účel nevhodně využít statický distribuční seznam (textový soubor obsahující seznam e-mailových adres). Kromě nevýhod, jakými jsou správa a sdílení aktuální verze dochází k mnohem závažnějšímu problému se změnou priorit uživatele. Nejčastějšími komplikacemi jsou:

- Organizace vede pouze jeden distribuční seznam, tedy uživatel dostává i mnoho emailů s informacemi, o které nemá zájem.
- Organizace vede více seznamů dle kategorií. Uživatel, který odebírá více kategorií, je v systému veden vícekrát a v případě změny emailové adresy dojde snadno k opomenutí staré.
- Uživatel si již nepřeje dostávat zprávy. Jediný mechanismus, kterým může být ze systému odstraněn, je informování organizace prostřednictvím emailu s žádostí o odstranění. To si vynucuje využití jednoho z pracovníků k administraci tohoto systému. Navíc dojde snadno k opakování žádosti při nedostatečném vyhledání adresy ve všech seznamech. Stejná situace nastává, pokud si uživatel pouze přeje změnit kategorie odebíraných zpráv.

Funkční požadavky

Administrační rozhraní portálu bude nabízet modul pro správu číselníků webové prezentace organizace. Při vstupu do modulu uživatel uvidí souhrnné statistiky o

návštěvnosti webu a o využívání odběru zpráv prostřednictvím emailu. Statistiky budou zpracovávány vlastní aplikací, nebude užito služeb třetích stran.

Kontextové menu nabídne dle přidělených oprávnění správu číselníků v kategoriích:

- Novinky
- Zájmové útvary
- Akce
- Soutěže
- Tábory
- Informace o organizaci, kontakty a odkazy

V každé kategorii budou dostupné tabulky číselníků, které do ní náleží a budou nabízet standardní sadu funkcí (vklad, výmaz, editace). Realizace tabulek a záložek bude jednotná pro celý portál. Evidované parametry budou použity dle definovaného ERA modelu (viz kapitola 5.8).

Modul bude dále nabízet funkci pro správu galerie. Zde bude uživateli umožněno nahrát na web fotografie, pracovat s alby a opatřovat všechny soubory poznámkami. Funkce uživateli umožní vybrat více souborů pro nahrání najednou. Postup přenosu souborů bude vizuálně indikován formou ukazatele stavu přenosu. Z jednotlivých přenesených souborů budou automaticky vytvořeny miniatury potřebné pro galerii. Uživatel bude zároveň informován o kapacitě úložiště a aktuálně použitým místě.

Další funkcí modulu bude rozhraní pro odesílání zpráv prostřednictvím emailu. Funkce umožní vklad textu zprávy, zadání jejího typu (novinka, akce, tábor, zájmový útvar, soutěž) a odeslání. K dispozici bude též archiv zpráv, který kromě textu a data odeslání bude evidovat množství schránek, do kterých byla zpráva doručena a jejího autora.

Pro registraci a úpravy zadaných schránek bude realizováno pro nepřihlášeného uživatele prostředí na hlavní stránce. Po zadání emailové adresy bude uživatel vyzván k výběru výše uvedených kategorií, dle svých preferencí. Systém po vyplnění na zadanou adresu zašle mail, informující uživatele o provedené registraci a odkaz na aktivaci služby. Po provedení aktivace uživatel začne odebírat správy. Úprava této služby se bude provádět prostřednictvím odkazu v zápatí každé doručené zprávy.

4.2 Agenda

4.2.1 Docházkové akce

V posledních letech přibývá akcí, u kterých je nutnost evidovat pohyb účastníků. Je to způsobeno potřebou vést statistiky, které doloží návratnost a význam financování daných projektů (ve většině případů jde o grantovou politiku EU či kraje). V praxi tak existují v každé organizaci v horším případě stohy papírů se jmény, kterým dávají jistý řád pouze desky, ve kterých jsou umístěny. V tom lepším případě je zřízen sešit tabulkového procesoru Excel, kam jsou příchody evidovány. I v tomto případě však časem dochází k silné nekonzistenci záznamů, protože každý zaměstnanec provede zápis jinak, v krajních případech pak dochází k chybám zaviněným nepozorností (špatná data, časy, částečně vyplněné údaje...).

Funkční požadavky

Hlavní obrazovka modulu bude obsahovat tabulku právě přítomných návštěvníků a rozbalovací seznam s registrovanými návštěvníky. Každý záznam v tabulce bude obsahovat identifikátor návštěvníka (nejčastěji jméno), čas vytvoření záznamu (příchod) a tlačítko pro nastavení odchodu návštěvníka.

Při příchodu návštěvníka bude vybrán jeho identifikátor z rozbalovacího seznamu a potvrzením bude vložen do tabulky právě přítomných uživatelů. Seznam bude filtrován na přítomné uživatele, jejich záznamy nebudou nabízeny. Pod seznamem bude uveden celkový počet položek. Vypisování budou pouze aktivní uživatelé (viz níže – Správa návštěvníků).

Odchod návštěvníka bude proveden stisknutím tlačítka odchodu u korespondujícího záznamu v tabulce. Záznam bude okamžitě z tabulky odebrán a bude vytvořen log pro tvorbu statistiky.

Speciální oprávnění bude možno nastavit pro funkce Statistiky a Správy návštěvníků. Při volbě funkce statistiky bude zobrazena tabulka se statistikou pro aktuální měsíc. Nad tabulkou budou umístěny souhrnné informace a dialog pro výběr měsíce. Dialog dovolí výběr pouze z měsíců, pro které existují nějaké záznamy. Souhrnné informace budou obsahovat údaje o počtu návštěv, počtu jednotlivců, kteří akci navštívili a celkový počet hodin, které v klubu (akci) za měsíc strávili. Tabulka bude obsahovat v řádcích seznam návštěvníků, kteří v daném měsíci provedli alespoň

jednu návštěvu. Pro každý záznam bude zobrazeno číslo člena, identifikátor, počet návštěv a celkový strávený čas na akci.

Funkce správa uživatelů zobrazí tabulku uživatelů (včetně neaktivních) a umožní běžné operace se seznamem. Ke každému záznamu bude evidován identifikátor a stav. Stav je reprezentován hodnotami aktivní a neaktivní. Číslo návštěvníka bude generováno automaticky.

4.2.2 Výkaz

Aktuálně je na všech pracovištích veden papírový výkaz práce. Způsoby jeho generování zahrnují metody od vyplnění předtištěného dokumentu, přes tisk excelového listu až po jednoduché webové formuláře. Ač došlo k domluvě o jednotném znění formuláře, byly postupně vytvořeny modifikace a tak kromě různého způsobu distribuce vznikají časté problémy s tiskem¹², pozdním odevzdáváním a chybami v datech. Vše je řešeno opětovným vypracováním formuláře a tiskem, čímž dochází ke zbytečným ztrátám na spotřebním materiálu.

Modul umožní evidenci výkazů práce pro jednotlivé uživatele. Každý zaměstnanec bude mít zaregistrované až tři různé smlouvy, pro které bude veden nezávislý formulář odpracovaných hodin. K modulu bude možno navíc nastavit speciální oprávnění pro zobrazení archivu výkazů a zjištění stavu jednotlivých zaměstnanců.

Funkční požadavky

Při vstupu do modulu uvidí uživatel až tři tlačítka pro zobrazení svých výkazů. Tlačítka budou označena funkcí zaměstnance, pro kterou je výkaz veden. Po vstupu do výkazu bude zobrazen samotný formulář. V horní části bude umístěn list pro výběr starých výkazů. Pod formulářem budou tlačítka pro uložení formuláře a pro uzamčení (virtuální podepsání) formuláře. Formulář bude obsahovat záhlaví dle dodaného vzoru a datovou část se záznamy činnosti. Víkendy budou barevně vyznačeny. Pro každý den bude formulář schopen uchovat čas příchodu a odchodu (z těchto hodnot počítat odpracovanou dobu s respektováním zákonné pauzy), dále budou dostupná pole pro zadání času suplování, výuky, práce na akci, přesčasu, dovolené, nemoci a náhradního volna. Pro automatizaci výpočtu bude dále možno zadat dělenou směnu a přerušení

¹² V případě použití webového rozhraní nelze s různými prohlížeči zaručit jednotný vzhled tištěné stránky.

pracovní doby. Uživatel bude moci libovolný den označit popiskem, stejně jako celý formulář. Vklad času příchodu a odchodu bude realizován vhodným výběrovým seznamem.

Výběrem funkce Archiv bude oprávněnému uživateli zobrazen seznam uživatelů, pro které je zaznamenán nějaký formulář. Zobrazení se bude provádět výběrem z listu, každá smlouva každého zaměstnance bude mít samostatný list.

Funkce uzávěrka bude sloužit ke kontrole stavu vyplnění výkazů. Při vstupu bude privilegovanému uživateli zobrazen seznam uživatelů, kteří v daném měsíci mají vyplněné nějaké smlouvy, a k těmto smlouvám bude analyzováno, v jakém stavu se nachází výkaz práce. Systém bude zobrazovat stavy dokončeno, rozepsáno, prázdné. Uživatelům, kteří nebudou mít formulář ve stavu dokončeno, bude možné odeslat hromadný mail s informací o blížícím se konci měsíce.

4.3 Správa

4.3.1 Servis

Jako reakce na silné snížení množství povolených pracovních úvazků byl v mnoha objektech zřízen jednoduchý kamerový systém, který ve většině případů sestává z IP kamer. Je využit pro kontrolu vstupu do objektu (náhrada za vrátnici), popř. problematických prostor, jako jsou šatny, veřejné počítačové učebny atd. Dále pak začíná být pravidlem sdílení jednoho pracovníka pro správu informační techniky nejen mezi budovami, ale dokonce i mezi organizacemi, což silně komplikuje systém rozdělení a kontroly jeho práce.

Funkční požadavky

Cílem modulu bude nabízet servisní služby. Při vstupu do modulu uvidí uživatel záznamy práce, provedené technikem. Záznamy budou seskupeny po dnech, každý bude obsahovat název problému, popis a kategorii. Kategorie bude charakterizovat, zda šlo o řešení softwarového, nebo hardwarového problému.

Přidávání záznamů bude nastavitelné jako první funkce modulu. Další funkce zobrazí uživateli výstup ze všech interních kamer na jedné obrazovce. Žádné funkce nahrávání či procházení záznamů nejsou plánovány. Dále bude modul nabízet funkci detekce zapnutého zařízení v organizaci (počítače, síťové tiskárny, IP kamery, servery...). Jednotlivé položky budou načítané z příslušných číselníků, po zpracování

výsledků bude zobrazena tabulka, obsahující pro každou položku název stroje, IP adresu, fyzické umístění a informaci o dostupnosti.

4.3.2 Vedoucí – správa učeben

V případě, kdy organizace nabízí výuku spojenou s výpočetní technikou, dochází k tvorbě tzv. počítačových učeben. Ty obsahují zpravidla mezi 10 – 20 počítači pro návštěvníky a počítač vybavený dataprojektorem pro pedagoga. Správa počítačových učeben s sebou nese několik činností, které jsou obecně řešeny manuálně a jejich automatizace by přinesla nemalé časové úspory.

Pro správu uživatelských kont je aktuálně vždy použita doména Windows (viz kapitola 4.5), což přináší otázku metodiky údržby jejího provozu. Zpracováním webového rozhraní pro přístup k uživatelským kontům v pevně daném umístění by byla možná editace uživatelů, listování záznamy pedagogem či např. reset zapomenutého hesla (který je aktuálně častým důvodem pro kontaktování administrátora). Další automatizovatelnou činností je vypínání učebny při ukončení výuky.

Funkční požadavky

Pro všechny funkce modulu bude možno nastavovat oprávnění. Modul bude komunikovat s doménovým řadičem a umožní obsluhu vkládání a editaci záznamů v konfiguraci daném umístění. Dále dovolí reset hesla libovolného uživatele (jeho změnu bez nutnosti znát heslo předešlé) a výpis všech uživatelů spolu s datem posledního přihlášení.

Dále bude nabízet funkce Vypnutí počítače/učebny a Odvolání vypnutí počítače. Po jejich volbě se uživateli zobrazí dialog pro výběr zvolené oblasti a nastavení parametrů. OS Windows vypisuje při vynuceném vypnutí informace o jeho důvodu. Nastavit bude možno čas do vypnutí a informační zprávu.

4.3.3 Administrace zaměstnanců

Modul umožní správu číselníku zaměstnanců. Evidované položky budou vycházet z analýzy ERA modelu. Modul bude disponovat samostatnou funkcí pro výmaz zaměstnance. Vizually budou rozlišeni interní a externí pracovníci.

Další funkcí modulu bude výpis záznamů z přihlašování zaměstnanců do systému.

4.4 Nápoředné a navigační prostředky

Protože má být aplikace ovladatelná bez přímého školení personálu, je nutno zpracovat dokumentaci, která by v plné míře obsahla navigaci a použití funkcí portálu. Byly proto provedeny sociální experimenty s přístupem zaměstnanců k různým způsobům nápoředy, jejichž výsledkem je snaha minimalizovat rozsah tištěného manuálu. Takový materiál je zcela nevyhovující, protože lineární čtení dokumentu nepřinese představu o ovládání systému a dohledávání jednotlivých informací není praktické, protože většinou v daný moment nepřipadá v úvahu.

Nejvhodnějším řešením je zpracovat do aplikace množství navigačních a nápoředných prvků, které by uživatele dostatečně vedly při používání.

4.4.1 Dynamická menu

Rozhraní aplikace bude uživateli nabízet pouze funkce, na které má oprávnění. Hlavní i vedlejší menu budou proto dynamicky řešena dle aktuálně přihlášeného uživatele. Aby uživatel věděl, jaké funkce a moduly systém nabízí a mohl o ně zažádat, bude aplikace vypisovat po přihlášení tabulku, zobrazující všechny funkce systému a aktuální nastavení přístupu k nim.

4.4.2 Formulářové a tabulkové prvky

Součástí portálu bude množství tabulek a formulářů, které jsou různou měrou komplikované na přesné a jednotné vyplnění. Pro zjednodušení tohoto úkonu a eliminování nutnosti vést odděleně metodiku způsobu zápisu, budou implementovány následující pomocné prvky.

Počet zbývajících znaků

Každé vstupní pole formuláře zobrazí po aktivaci počet znaků, které je ještě možno zadat do vyčerpání jeho rozsahu. Počítadlo bude dynamicky přepisováno při editaci pole. V případě využití celé kapacity bude pole graficky výrazně označeno. Po deaktivaci pole bude počítadlo opět schováno, čímž se zabezpečí přehlednost formuláře.

Bublinové nápoředy

Pro popis způsobu vyplnění formulářových polí bude implementována bublinová nápoředa. Prvek bude zobrazen při najetí kurzoru nad vstupní pole. Skryt bude buď změnou pozice kurzoru, nebo aktivací pole (kliknutím do pole pro vyplnění). Nápoředa

bude obsahovat jako nadpis název pole, pro které je určena. Následovat bude krátký popis významu a cíle. Pokud to charakter pole dovolí, bude uveden příklad vyplnění.

Záložky

Tabulky, zobrazující číselníky a jiná data budou zobrazována jednotnou formou. V případě možnosti editovat data budou ovládací tlačítka zobrazena jako první sloupec tabulky. Tabulky budou používat barevné prokládání řádků a pro zvýšení přehlednosti budou zvýrazňovat označenou řádku. Jednotlivé buňky umožní vkládání netextového obsahu, např. obrázků. Data v tabulce bude možno řadit dle specifikovaných sloupců.

V případě, že bude jedna funkce obsahovat více tabulek, bude jejich oddělení realizováno záložkami. Pokud to implementace umožní, bude přepínání záložek i řazení řádků řešeno bez nutnosti znovu načíst obsah. Tabulky budou opatřeny počítadlem záznamů.

4.4.3 Lokalizace

Aplikace bude zahrnovat systém přepínání českého a anglického jazyka pro navigační prvky (menu) a statické popisky (nadpisy formulářových polí, tlačítka...). Změna jazyka bude realizována prostřednictvím volby v záhlaví stránky. Jako výchozí jazyk bude použita čeština. Roleta přepínače bude zobrazovat aktuálně použitý jazyk formou ikony vlajky.

4.4.4 Oznámení o provedení akce

Aplikace bude uživatele informovat o výsledku provedených akcí formou informačního dialogu. Styl a vzhled dialogu bude upraven dle celkového rozložení grafiky aplikace. Systém oznamování bude definovat dvě různá nastavení, pro oznámení kladného i záporného výsledku činnosti.

4.5 Active Directory

Při nasazení OS Windows Server dochází obecně k zásadnímu problému, kdy je tento systém použit pro přihlašování uživatelů do domény (což je smysl jeho použití), a správu uživatelů zajišťuje IT technik. Tato situace má dva zásadní nedostatky. Z ekonomického hlediska je nevhodné využití pracovníka na tento typ činnosti, protože jde o rutinní činnost, která „by nemusela“ vyžadovat kvalifikaci. Za druhé pak začátkem školního roku dochází k zápisu účastníků do zájmových útvarů (tedy i vytváření ekvivalentních záznamů v AD) a při vyšším počtu může vytváření účtů

znamenat vyčerpání technika na několik pracovních dní, kdy nemůže dělat práci, která by jindy měla vyšší prioritu. V níže uvedeném modulu Vedoucí je proto analyzována možnost převést část správy uživatelských kont na pedagoga.

Active Directory je distribuovaná adresářová služba a zastřešuje celý systém správy počítačové sítě Microsoft, který je obsažen v OS Windows Server 2000, 2003 a 2008. AD zahrnuje řadu služeb, primární rolí je však autentizace a autorizace, tedy správa oprávnění. Ukládá a spravuje informace o objektech a jejich vztazích. Po instalaci AD se server stává doménovým řadičem [21].

Nejpoužívanějším protokolem pro komunikaci s AD je *LDAP*, což je aplikační protokol navržený pro dotazování a modifikaci adresářových služeb. LDAP vznikl zjednodušením skupiny standardů *X.500* (DOP, DSP, DISP...) a pracuje nad *TCP/IP* s porty 389 a 636 (SSL¹³). Pro výměnu informací využívá standardizovaný textový formát *LDIF*.

LDAP data jsou objekty, které jsou uloženy ve stromové struktuře *DIT*. Model je založen na záznamech obsahujících informace o konkrétním objektu (počítač, uživatel, organizační jednotka). Atributy vždy obsahují typ a jednu či více hodnot a charakterizují vlastnosti objektů. Standardní atributy používané v AD jsou uvedeny v tabulce 4.1.

Atribut	Význam
sAMAccountName	uživatelské jméno podporované staršími verzemi OS
sAMAccountType	typ účtu
userPrincipalName	přihlašovací jméno tvar: <uživatel>@<jméno domény>
displayName	zobrazované jméno (například v aplikaci Exchange)
givenName	jméno
sn	příjmení
description	popisek
mail	adresa elektronické pošty
company	společnost (firma)
department	oddělení
location	místo
streetAddress	ulice
memberOf	skupiny, do kterých účet patří

Tabulka 4.1: Standardní atributy používané v AD.

¹³ SSL je vrstva vložená mezi aplikační a transportní vrstvu, která se stará o zabezpečení komunikace šifrováním.

Příslušnost atributů, ke které třídě objektů popisuje implementace informačního modelu označovaná jako schéma. Výchozí schéma Active Directory lze dále modifikovat a rozšiřovat pro použití s dalšími službami, jako například Microsoft Exchange.

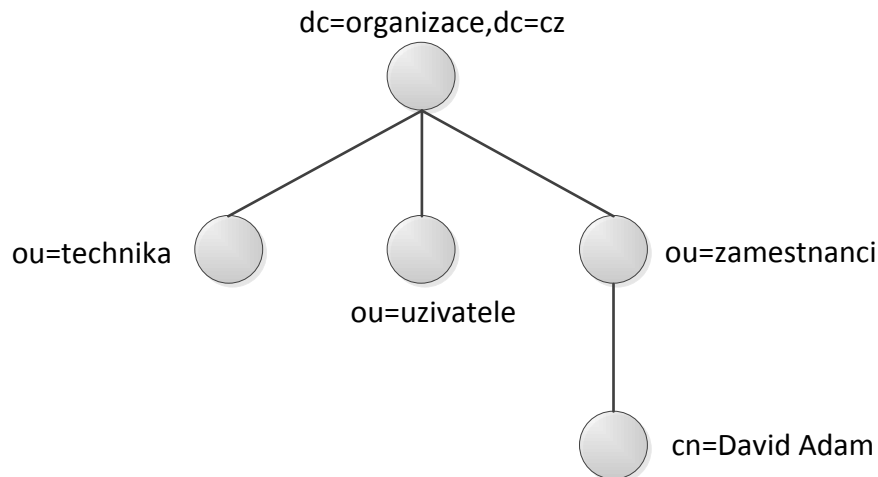
Jmenný model – Distinguished Name

Identifikace objektů je realizována pomocí *Distinguished Name* (dále jen DN). Tento identifikátor obsahuje celou cestu k objektu (jména jednotlivých kontejnerů a domén) a jméno objektu, vše oddělené čárkami. Položky obsahují vždy název atributu (viz tabulka 4.2) a hodnotu.

AD atribut	Jméno
CN	Common Name
OU	Organization Unit
DC	Domain Component

Tabulka 4.2: Jmenné atributy AD.

Obrázek 4.1 ilustruje příklad adresáře AD pro doménu organizace.cz. DN objektu David Adam je **DN=cn=David Adam,ou=zamestnanci,dc=organizace,dc=cz**



Obrázek 4.1: Příklad adresáře AD.

5 Programátorská dokumentace

5.1 Architektura aplikace

Pro lepší správu kódu a přehlednost je aplikace organizována do tří samostatných vrstev.

- **Databázová vrstva**

Tato vrstva je reprezentována třídami nazývanými *data access objects* (dále jen DAO), které tvoří abstraktní rozhraní pro přístup k datům a oddělují zbytek aplikace od funkcí specifických pro použitou databázi. Spojení je realizováno prostřednictvím *Java Database Connectivity API* (dále označován jako JDBC). Výhodou tohoto řešení je možnost změny databázového systému bez nutnosti přepsání kódu logiky aplikace a skrytí detailů databázového schématu.

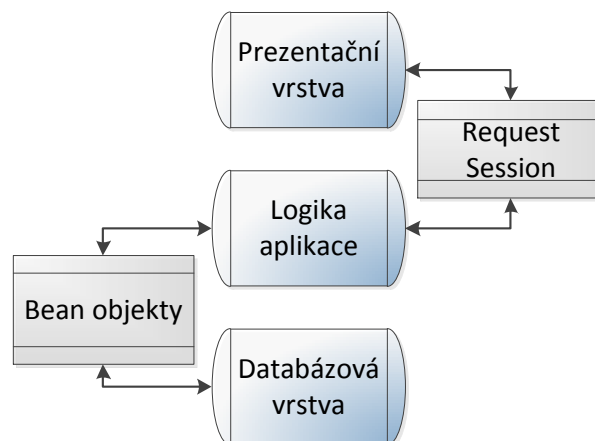
- **Logika aplikace**

Tato vrstva se stará o zpracování požadavků a přípravu všech potřebných dat pro další obrazovku. Tvořena je třídami typu servlet, které si obsah předávají a podle svého určení jej doplňují.

- **Prezentační vrstva**

Prezentační část aplikace je řešena šablonami JSP s využitím JSTL (viz kapitola 3.6.1), výstupem je XHTML kód.

Jak ilustruje obrázek 5.1, nekomunikují mezi sebou jednotlivé vrstvy přímo. Pro komunikaci mezi databázovou vrstvou a logikou aplikace jsou předávány jako parametry a návratové hodnoty metod objekty typu bean. Po zpracování jsou tyto objekty předávány prezentační vrstvě, která díky JSTL dokáže efektivně extrahovat jejich obsah pro vizualizaci. Předání je realizováno nejčastěji přes objekt *request*, nebo uložením do *session*.



Obrázek 5.1: Vrstvy aplikace.

5.2 Adresářová struktura projektu

Základní členění adresářové struktury je dáno výchozím rozložením projektu v IDE Netbeans. Dále jsou složky děleny dle logického zaměření. V případě zdrojových kódů podle typu objektů, které obsahují.

- web
 - design
 - buttons obrázky ovládacích prvků
 - default výchozí grafiku prostředí
 - ico specifické ikony prostředí
 - icons ikony z knihovny CrystalClear
 - js JavaScriptové skripty a knihovny
 - files grafické podklady dle nasazení
 - mailer soubory JSP registrace odebrání zpráv
 - stag soubory JSP administrace portálu
- test testy *JUnit*
- src zdrojové kódy aplikace v jazyce Java
 - beans třídy typu bean
 - core třídy s nižšími funkcemi systému
 - db DAO objekty
 - servlets třídy typu servlet
 - translations soubory překladů typu properties
- build strukturu projektu s přeloženými třídami
- dist balíček *WAR* k distribuci

5.3 Objekty a metody

Tato kapitola obsahuje popis důležitých objektů systému a jejich metod. Projekt dále obsahuje aktuálně 22 tříd servletů, 31 DAO objektů a 30 Bean.

5.3.1 ADAdm

Třída zapouzdřuje práci s LDAP služby Active Directory. Její metody využívá modul vedoucí, který s ní přímo pracuje. Navíc poskytuje autentifikaci pro přihlášení do aplikace.

boolean deleteUser(String userName)

Výmaz zadaného uživatele z AD. Vrací kladnou hodnotu pro úspěšné odebrání, v opačném případě zápornou.

boolean connect()

Připojení k serveru, nastavení potřebných parametrů, vytvoření kontextu. Metoda na základě deploy descriptoru nastaví adresu AD řadiče, typ autentifikace a umístění úložiště klíčů. V základním nastavení se připojuje na port 636 s použitím SSL protokolu. Vrací kladnou hodnotu pro úspěšné vytvoření kontextu, zápornou pro chybu.

Map validateUser(String userName, String pass)

Metoda slouží k autentifikaci uživatele. O úspěchu rozhoduje na základě úspěšného vytvoření základního kontextu. Spojení navazuje na portu 389. Po vytvoření kontextu vyčte data o zadaném záznamu, která vrací formou objektu Map¹⁴. V případě navrácení null nebyl uživatel ověřen.

boolean createUser(String userName, String name, String pass, String group)

Metoda vytvoří nového uživatele na základě zadaných parametrů. Z důvodů technických omezení AD je nejprve vytvořen deaktivovaný účet s nastavenými parametry pro povolení změny hesla a bez expirace. Poté je účtu nastaveno heslo a je aktivován.

Pokud jsou všechny kroky úspěšné, je účet zařazen do skupiny spravované portálovou aplikací a vrácena kladná hodnota. V případě selhání kteréhokoliv kroku je navrácena hodnota záporná.

boolean changePass(String userName, String pass)

Metoda modifikuje heslo zadanému uživateli. Pro operaci je využit administrátorský kontext, proto není pro změnu vyžadováno heslo staré. Metoda vrací logickou hodnotu dle výsledku operace.

ArrayList listAll()

Metoda vyhledá všechny dostupné záznamy v celém stromě a vrací je jako objekty *UserBean* v poli.

¹⁴ Datový typ počítaný mezi Java kolekce. Někdy též nazývaný slovník, nebo asociativní pole. [30]

ArrayList listGroup(String groupName)

Metoda vyhledá všechny záznamy v zadané skupině a vrátí je jako záznamy *UserBean* v poli.

String getUserDN(String username)

Privátní metoda sloužící pro převod uživatelského jména na *Distinguished Name*.

String getUserUID(String userDN)

Privátní metoda sloužící pro převod *Distinguished Name* na uživatelské jméno.

String convertWinTime(String winTime)

Privátní metoda, sloužící pro převod času užívaného ve Windows Server do čitelné podoby v českém prostředí.

Realizace konverze jmen je stejným postupem implementována též pro skupiny.

5.3.2 Hwutils

Třída zapouzdřuje funkce spojené s hardware. V aktuální verzi nabízí možnost vypnout PC, vypnout učebnu, zjistit stav PC a několik vývojových a testovacích funkcí.

PcBean ping(PcBean target)

Metoda zjišťuje síťovou dostupnost zařízení, které je zadáno jako parametr typu *PcBean*. Pro svou činnost využívá volání aplikace ping OS Windows a jeho návratovou hodnotu. Ta je po zpracování dosazena do objektu jako atribut *status*.

int shutdown(String param)

Privátní metoda, sloužící k volání aplikace *shutdown*, která je součástí OS Windows. Vstupem metody je řetězec zformátovaných parametrů. Metoda vrací návratovou hodnotu aplikace *shutdown*.

boolean abortShutdown(String pcName)

Metoda slouží ke zrušení vypínání počítače, u kterého byl zadán čas do vypnutí. Po zformátování požadavku využívá ke své funkci metodu *shutdown*.

boolean shutdownClass(String className)

Metoda slouží k hromadnému vypnutí daného umístění.

5.3.3 MailUtils

Třída je určena ke správě metod souvisejících s mailovými službami.

boolean sendMail(String to, String message, String subject)

Metoda slouží k jednoduchému odesílání emailové zprávy prostřednictvím objektu `mail`, který je součástí prostředí Java. Vstupy metody jsou označení příjemce, text zprávy a předmět. Metoda vrací návratovou hodnotu reprezentující úspěch akce.

5.3.4 Gallery a GalleryAdmin

Třídy obsahují metody pro práci s obrazovou galerií, která je součástí prezentační části webu.

String galSize(ServletContext cx)

Metoda vrací textový řetězec s informací o celkové velikosti galerie a dostupném prostoru na úložišti ve formátu „použitá velikost z dostupné místo“. K výpočtu velikosti galerie využívá metodu `sizeOfDirectory()` ze třídy `FileUtils`, pro výpočet zbývajících místa metodu `getUsableSpace()`.

String humanReadableSize(long bytes)

Metoda je určena k formátování informací o velikosti souborů. Vstupní parametr je velikost souboru v bytech, výstupem je řetězec s velikostí opatřený příslušnou jednotkou.

boolean deleteGalleryFiles(File path)

Metoda rekurzivně projde zadané umístění a vymaže všechny obsah.

ArrayList breadList(int gal)

Metoda slouží ke generování drobečkového menu pomocí rekurzivního procházení cesty ke kořeni úložiště.

ArrayList dirList()

Metoda sumarizuje data o aktuálně prohlíženém albu, která jsou využita pro výpis detailů při procházení galerie.

void resize(File originalFile, File resizedFile, int newHeight, float quality)

Metoda je součástí třídy *GalleryAdmin* a slouží ke tvorbě náhledů nahrávaných souborů. Vstupními parametry jsou zdrojový soubor, výsledné soubory a úroveň JPG komprese. Metoda také opatří exportované soubory vodoznakem, který je shodný s adresou webu zadanou v Deploy Descriptoru.

5.4 Formuláře

Součástí aplikace je množství formulářů pro vstup uživatelských dat. Ty jsou tvořeny standardní HTML konstrukcí:

```
<form action="mailer" method="post">
    //jednotlivá vstupní pole
</form>
```

kde parametr *action* představuje cílovou adresu a parametr *method* definuje způsob přenosu dat (*get* nebo *post*).

5.4.1 Multipart

Pro realizaci formulářů, které obsahují vstupní pole typu *file*, je třeba nastavit parametry *enctype="multipart/form-data"* *accept-charset="utf-8"*. Pro příjem dat z takto nastaveného formuláře je však standardní přístup v servletu nedostatečný a je třeba k realizaci využít další knihovnu. Použita byla knihovna *Apache commons-fileupload*, která nabízí všechnu potřebnou funkcionalitu. Přesto je třeba dodatečné řešení dvou komplikací:

- **Kódování vstupů**

Knihovna *commons-fileupload* provádí pro své operace změnu znakové sady. Přijaté vstupy jsou tedy s původním nastavením poškozené. Řešením je změna sady při příjmu konstrukcí:

```
text = new String(item.getString("UTF-8").trim());
```

- **Zjišťování typu vstupních polí**

Knihovna *commons-fileupload* vrátí data formuláře jako list objektů. Kromě vyčítání hodnot cyklem je třeba rozhodovat, zda je dané pole formulářový prvek, nebo soubor. To zajistí konstrukce:

```
if(item.isFormField()) {
    if(item.getFieldName().equals("enable")) {
        //zpracování vstupu
    }
}
```

5.4.2 Poslední zapsaný index

Pokud formulář odesílá soubory, jež se budou vztahovat k záznamu v databázi, který je daty ze stejného formuláře vytvořen, je vhodné tyto soubory umístit do složky, jejíž název bude shodný s klíčem záznamu – jeho *ID*.

Zde nastává problém, kdy *ID* je v MySQL generováno automaticky při kladu záznamu (vlastnost *autoincrement*). Situace je často nevhodně tvořena konstrukcí zápisu záznamu, uložením dat do dočasného umístění, přečtením posledního zapsaného *ID* a přesun souborů na správné místo. Slabina této konstrukce je v souběžných zápisech jednotlivých instancí aplikace. Proběhne-li zápis dalšího záznamu v době práce se soubory, celý algoritmus selže.

Řešením je SQL dotaz

```
SELECT LAST_INSERT_ID()
```

který zavolaný uvnitř transakce vrátí poslední zapsané *ID*.

5.5 Řízení oprávnění

Portál byl vytvořen tak, aby svou funkcionalitou rozšiřoval možnosti různých oddělení organizace. Proto je samozřejmostí, že musel být vymyšlen způsob, jakým by aplikace řešila oprávnění uživatelů k užívání jednotlivých funkcí.

5.5.1 Autentizace uživatele

Autentizací rozumíme proces, který vede k jednoznačnému určení identity uživatele, který do systému přistupuje. Přihlášení k aplikaci je realizováno dvojicí údajů, jménem a heslem. Vstupní pole jsou umístěna v záhlaví každé stránky, proto je uživatel schopen se přihlásit z libovolného místa webové prezentace. Pro lepší pochopení smyslu jsou kromě popisků pole opatřena předvyplněným textem, který je automaticky odstraněn při vybrání daného pole.

Při autorizaci uživatele aplikace testuje shodnost zadaných údajů s dvěma nezávislými systémy (viz tabulka 5.1). Tento způsob byl zvolen z důvodu možného samostatného běhu portálu bez nutnosti propojení s databází LDAP. Pro úspěšné ověření uživatele stačí získat oprávnění jedním způsobem.

	LDAP	MySQL
Autentizace uživatele	Aplikace se pokusí o vytvoření kontextu, které je možné pouze s platnými údaji.	Nalezení záznamu uživatele a porovnání vygenerované MD5 hashe s uloženou.
Ukládání oprávnění	Neukládáno	Ukládáno (viz kapitola 0)
Správa hesla	Funkcemi MS Windows i pomocí správce LDAP v portálu.	V modulu správy uživatelů v portálu.

Tabulka 5.1: Použití jednotlivých databází v systému.

5.5.2 Autorizace uživatele

Autorizaci chápeme jako proces, ve kterém jsou uživatelé přidělena definovaná oprávnění pro služby informačního systému. Autorizace probíhá jako následující krok po autentizaci. Portál dostává z autentizační metody atributy, které nesou informace o přihlášeném uživateli. Na jejich základě:

- zjistí aktuální moduly,
- zjistí funkce načtených modulů,
- zpracuje oprávnění pro uživatele,
- sestaví tabulku oprávnění.

Všechny tabulky jsou uloženy v MySQL databázi a nejsou klonovány do LDAPu. Řešit duplikaci dat do obou databází není potřeba, protože aplikace ke svému chodu potřebuje MySQL a tudíž je schopna vždy data získat odtud. Relace mezi tabulkami a jednotlivé atributy jsou popsány ERA modelem (viz kapitola 5.8)

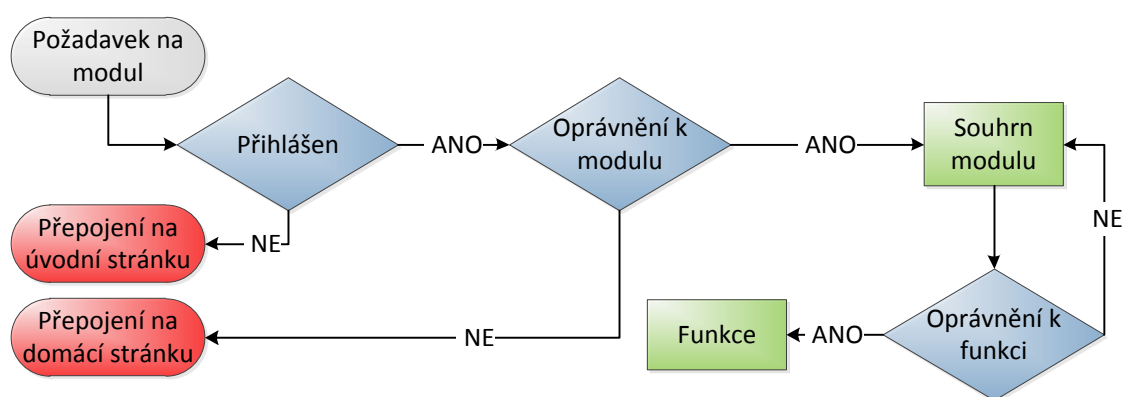
Údaje o přihlášeném uživateli i tabulka oprávnění jsou uloženy do session. Protože aplikace uzpůsobuje prostředí tak, aby uživatelé nabídla pouze funkce, které může použít (viz kapitola 5.5.3), musí být tabulka ve snadno použitelném formátu. Indexací sloupců jako funkcí a řádků jako modulů bylo docíleno tabulky, kde kladná hodnota v buňce znamená přiřazená oprávnění pro korespondující úkon.

Aby se zjednodušilo generování hlavního menu, bylo využito prázdného nultého indexu (data v tabulce jsou indexována od čísla jedna) a tento řádek je nastavován na kladnou hodnotu ve chvíli, kdy uživatel má k dispozici alespoň jednu funkci z modulu.

5.5.3 Proces vyhodnocení oprávnění

Při každém požadavku na určitou funkci systému je vyhodnocováno, zda uživatel má oprávnění ji využít. Kromě dynamicky generovaného menu je tak zajištěno, že se uživatel neoprávněně nedostane do modulu ručně zadanou adresou.

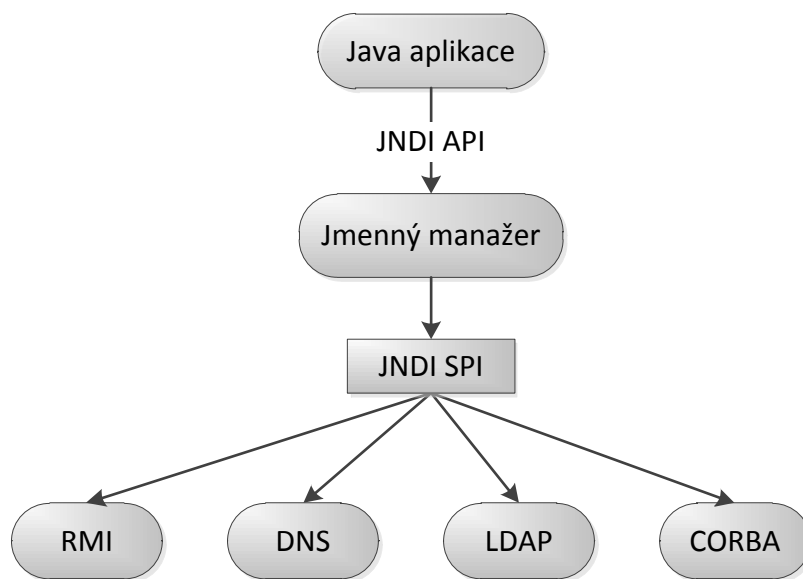
Obrázek 5.2 ilustruje, jak portál zpracovává oprávnění při požadavku na funkci. Červená pole znamenají stavy, ve kterých portál zaznamenal neautorizovaný vstup, zeleně jsou zakresleny stavy, ve kterých uživatel obdržel žádanou funkci.



Obrázek 5.2: Princip zpracování oprávnění.

5.6 Java Naming and Directory Interface

Pro komunikaci s adresářovými službami disponuje Java rozhraním *Java Naming and Directory Interface* (dále JNDI). Rozhraní nabízí objektově orientovaný přístup ke jmenným a adresářovým službám stejným způsobem, jako rozhraní JDBC k databázím. JNDI disponuje dvěma API a jedním SPI (Service-Provider Interface). Pomocí těchto API je možné přistupovat k mnoha souborovým systémům, distribuovaným objektovým systémům (CORBA, Java RMI) a adresářovým službám (LDAP, Novell NetWare, NIS+). SPI umožňuje JNDI připojit se ke konkrétní implementaci jmenné služby tím, že zaobalí její proprietární rozhraní tím, které poskytuje JNDI. Architektura JNDI je zobrazena na obrázku 5.3.



Obrázek 5.3: Architektura JNDI.

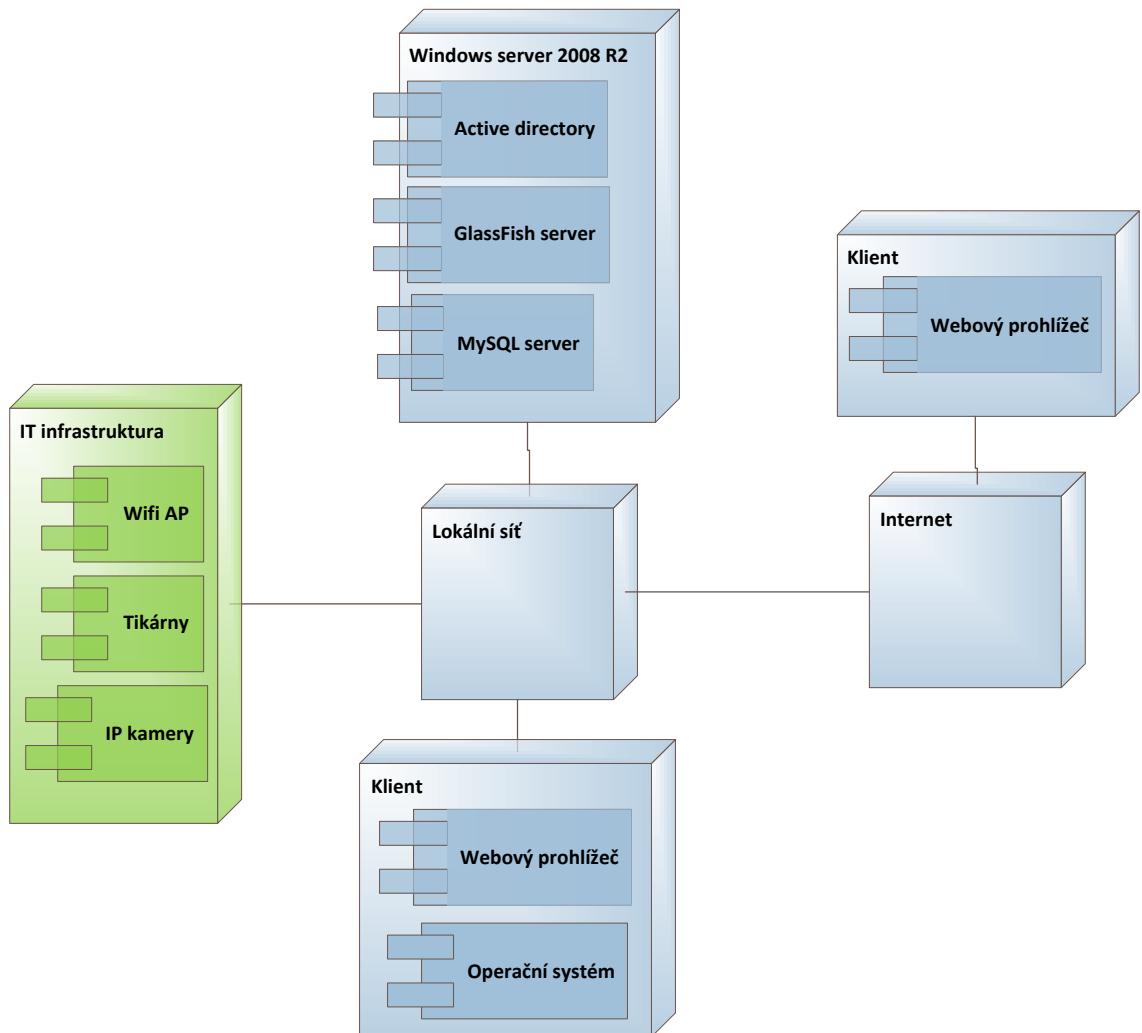
V tomto projektu je JNDI použito pro komunikaci s Active Directory. Jednotlivé funkce jsou implementovány ve třídě ADAdm popsané v kapitole 5.3.1. Pro vytvoření kontextu je použita konstrukce:

```
Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, hostName);
env.put(Context.SECURITY_PRINCIPAL, " CN=Users,DC=test,DC=local");
env.put(Context.SECURITY_CREDENTIALS, "test");
env.put(Context.SECURITY_PROTOCOL, "ssl");
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put("java.naming.ldap.factory.socket", "javax.net.ssl.SSLSocketFactory");
env.put("java.naming.ldap.version", "3");
System.setProperty("javax.net.ssl.keyStore", "c://cacerts");
System.setProperty("javax.net.ssl.keyStorePassword", "changeit");
System.setProperty("javax.net.ssl.trustStore", "c://cacerts");

ctx = new InitialDirContext(env);
```

5.7 Diagram nasazení

Obrázek 5.4 zachycuje předpokládané rozložení sítě, ve které bude IS nasazen.



Obrázek 5.4: Diagram nasazení.

5.8 ERA model

Základem modelu je tabulka *staff*, která drží informace o registrovaném uživateli. Kromě jiných údajů drží uživatelské jméno, které aplikaci propojuje s Active Directory systémem Windows Server. Dále pak obsahuje jednotlivé funkce, jako cizí klíče tabulky funkcí, které jsou využity pro propojení s modulem docházky. Je zde uveden též primární klíč *id*, který slouží k identifikaci v systému webové aplikace. Tento identifikátor je cizím klíčem v tabulce *rights* (oprávnění), která je rozkladovou tabulkou pro tabulku modulů (jeden uživatel může k více funkcím, jedna funkce má více uživatelů). Dále je uvedena tabulka *submodules*, která je číselníkem jednotlivých funkcí a slouží k interaktivnímu vypisování funkcí a menu. Dále je pak *id* cizím klíčem v tabulce *staff_logins*. Ta jednak sleduje využívání systému a jednak se na základě dat z ní budou v budoucnu uživatelé též odstraňovat. Nad tabulkou *staff* je pověšen trigger (viz níže), který při vložení nového řádku vloží základní sadu oprávnění.

Samostatně jsou pak jednotlivé tabulky spravovaných funkcí. Tábory, akce, soutěže, zájmové útvary, novinky a evidovaná docházka. První zmiňované slouží k ukládání dat, která jsou poté dostupná nepřihlášeným uživatelům jako webová prezentace. Docházkový systém naopak pracuje s již dodanými daty a na základě vydaných karet může obsluha aplikace nastavovat příchod a odchod, zjišťovat aktuální obsazení apod. Model dále obsahuje tabulky pro funkci galerie, která udržuje data o virtuální adresářové struktuře a informace o nahraných fotografiích (tabulky *gallery_photos* a *gallery_dirs*).

Několik tabulek v databázi je využito též na držení dat pro podporu jiných funkcí, jako je virtualizace infrastruktury organizace pro potřeby sledování dostupnosti a vypínání počítačů (tabulky *classes* a *pcs*), nebo jednoúčelových číselníků.

Všechny tabulky a relace jsou uvedeny na obrázku 5.5.

Trigger

Trigger je typ uložené procedury, která se spouští v souvislosti s prováděním akčních dotazů (INSERT, UPDATE, DELETE). Může obsahovat smyčku, podmínku, matematický výpočet či lokální proměnné pro dílčí operace. Na rozdíl od uložené procedury není možné triggeru nastavit vstupní parametr a též nevrací zpět sady záznamů. Pro svou činnost má trigger k dispozici obsah řádků, které se mění voláním dotazu spouštějícího proceduru.

Triggery se v databázích používají z důvodů zajištění konzistence dat, s jejich údržbou či správou. Lze jejich pomocí přenést část aplikační logiky na databázi.

Pomocí uvedeného spouštěče je zavolána níže uvedená metoda při vkladu záznamu do tabulky *staff* a předána jako parametr hodnota *id* vkládaného záznamu.

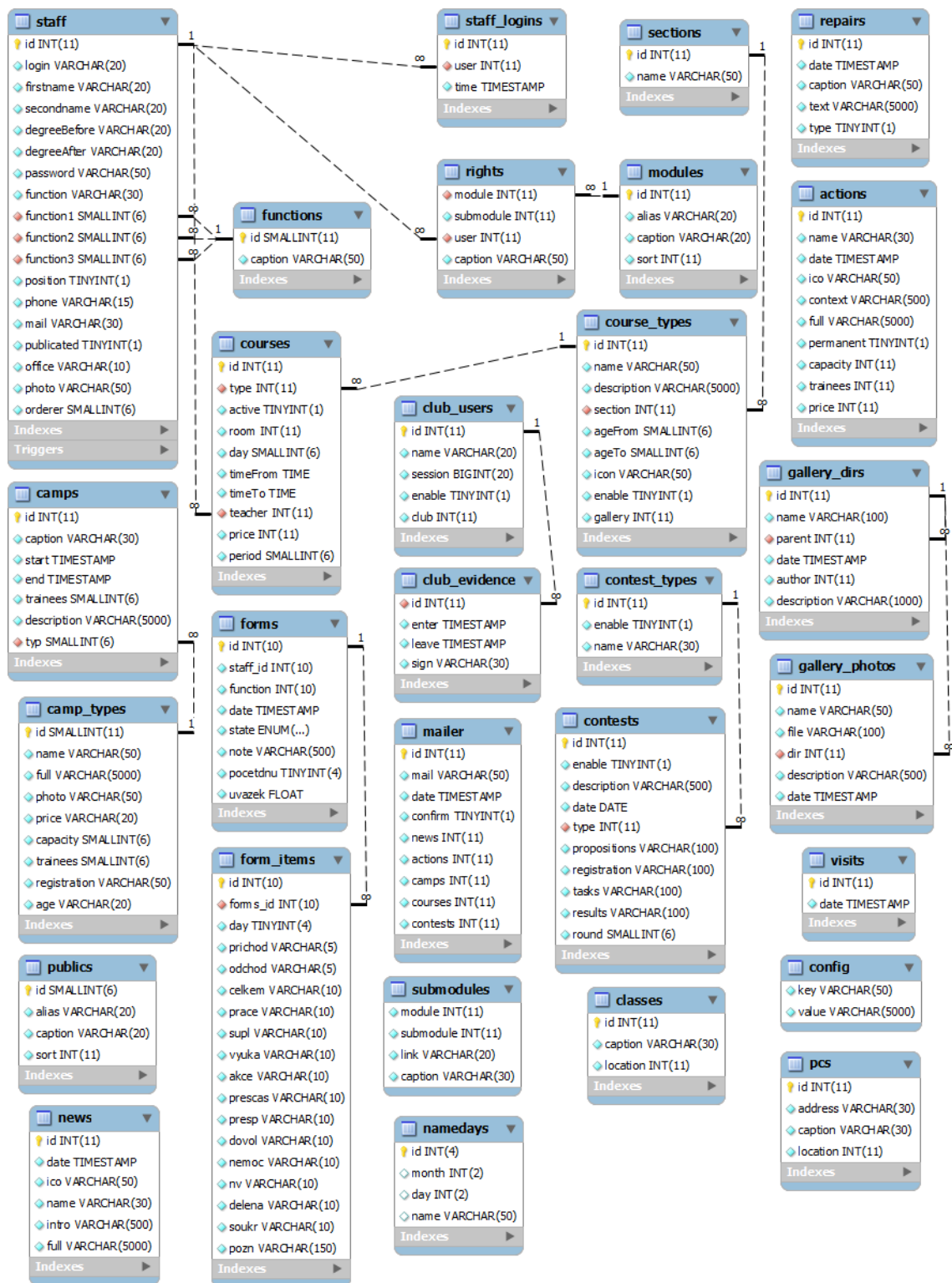
```
DROP TRIGGER IF EXISTS `membership`;  
CREATE TRIGGER `membership` AFTER INSERT ON `staff`  
    FOR EACH ROW BEGIN  
        CALL dright(NEW.id);  
    END
```

Uložené procedury

Uložené procedury jsou sadou SQL příkazů, které jsou umístěné na databázovém serveru. Důvodem pro vznik tohoto mechanismu je možnost předkompilování procedury dle potřeb databázového serveru a tím výrazné urychlení zpracování dávky. Stejně jako u triggeru slouží uložené procedury k unifikaci požadavků od klientů, usnadnění správy databázové vrstvy aplikací a přispívají také k zabezpečení serveru.

Zde uvedená metoda slouží k automatickému přidělení výchozí sady oprávnění pro zadaného uživatele. Je tedy možno nastavit, která oprávnění uživatel získá v okamžiku tvorby účtu. Použitím uložené procedury je minimalizováno množství přenášených dat, protože jde o skupinu vkladů do tabuly M:N relací.

```
DROP PROCEDURE `dright`;  
CREATE DEFINER=`root` PROCEDURE `dright`(IN userID INT)  
    BEGIN  
        INSERT INTO rights VALUES('2','8',userID,'');  
        //další oprávnění  
    END
```



Obrázek 5.5: ERA model.

5.9 Monitorování sítě

Pro zajištění přehledu o zapnutých zařízeních v síti byl zrealizován systém hromadného zjištění dostupnosti. Tuto funkci lze v prostředí Javy implementovat mnoha způsoby. Pro dosažení co nejlepších výsledků byl proto proveden rozbor několika konstrukcí, které jsou popsány v následujících kapitolách. Pro jednoduchost implementace, výborné výsledky i rychlost byla zvolena první možnost, tedy implementací pomocí aplikace ping, která je součástí OS Windows

5.9.1 Volání funkce ping přes příkazový řádek Windows jako runtime

- Paměťově náročné
- V základě pomalé
- Snadno parametrizovatelné

```
Runtime rt = Runtime.getRuntime();
Process pr;
Pr = rt.exec("cmd /c ping 127.0.0.1");
int errorCode = pr.waitFor();
target.setStatus(errorCode);
return target;
```

Každá aplikace psaná v Javě má po spuštění jednu instanci objektu *Runtime*, který aplikaci dovoluje komunikovat s prostředím, ve kterém běží. Výše uvedenou konstrukcí je možné docílit spuštění jiného programu, v tomto případě aplikace *ping* systému Windows. Toto řešení není v základě použitelně rychlé, umožňuje však efektivní optimalizaci. Zadáním vhodných parametrů lze získat rychlejší odezvu aplikace (snížení počtu opakování, velikost přenášeného packetu, snížení času čekání na odpověď). Dalšího urychlení bylo docíleno paralelizací pomocí objektu *ExecutorService*.

5.9.2 Využití funkce `isReachable()`

- Jednoduché
- Pomalé
- Sporná funkčnost

```
boolean reachable = InetAddress.
getByName(target.getAddress()).
isReachable(1500);
```

Teoreticky jde o ideální řešení, které v praxi bohužel dobře nefunguje. Metoda v závislosti na právech aplikace používá ke zjištění dostupnosti buď klasický ICMP packet¹⁵, nebo ověřuje dostupnost navázáním spojení na portu 7 (TCP handshake). Od toho se též odvíjí doba, kterou metoda pro svou práci potřebuje. Experimentálně bylo zjištěno, že pod 2000ms je navracený výsledek s vysokou pravděpodobností na více zatížené infrastruktuře chybný. Java touto funkcí disponuje od verze 1.5.

¹⁵ Posílání zpráv ICMP „Echo request“ a příjem „Echo reply“ pro zjištění dosažitelnosti cíle.

5.9.3 Implementace ICMP

- Rozsáhlé
- Platformě závislé

Poslední testované řešení spočívá ve využití knihoven pro zachytávání a tvorbu paketů, např. Jpcap [22]. Algoritmus zjištění stavu používá pro svou činnost dvě fáze. Nejprve je sestaven a odeslán ARP požadavek k získání cílové MAC adresy. Pokud není obdržena odpověď, je testovaný host považován za nedostupný a algoritmus vrací informaci o nedostupnosti. V opačném případě je zpracována druhá fáze, ve které je sestaven a poslán ICMP packet s využitím získané MAC adresy. Při obdržení odpovědi vrací algoritmus informaci o dostupnosti [23].

Ač řešení na této bázi fungují obecně dobře, jejich nevýhodou je nutnost instalace příslušné použité knihovny do OS, což komplikuje instalaci portálu i následnou údržbu systému.

5.9.4 Porovnání výkonnosti

V tabulce 5.2 jsou uvedeny naměřené rychlosti zjišťování dostupnosti zařízení pomocí výše popsaných metod. Údaje jsou průměrnou hodnotou ze sta vzorků.

Aplikace ping má nastaveny parametry: count = 1, size = 1, timeout = 3

Metoda isReachable má nastaven parametr: timeout = 2000

Parametr	Aplikace ping	Funkce isReachable()	Vlastní implementace
Odezva [ms] (spuštěno sériově)	12903	27305	18200
Odezva [ms] (spuštěno paralelně)	811	4012	1013
Navracený stav	Správný	Chybný	Správný

Tabulka 5.2: Rychlost metod skenování sítě.

5.10 Deploy descriptor

Deployment descriptor (dále jen DD) je konfigurační soubor, který je součástí archivu, nasazovaného na webový server. Je umístěn v adresáři `WEB-INF` a v J2EE je povinné jméno `web.xml`. Obsah souboru definuje mapování jednotlivých servletů, konfigurační informace, nastavení zabezpečení atd. Následuje popis vybraných částí:

Příklad definování chybových stránek. V aplikaci může dojít k chybám, na které webový server reaguje generováním stránky s chybovým kódem. Tyto stránky je možné upravit (většinou po designové stránce) nahrazením automaticky generované stránky vlastním souborem.

```
<error-page>
  <error-code>404</error-code>
  <location>/WEB-INF/errorPages/404.jsp</location>
</error-page>
```

V konfiguraci je možno nastavit délku platnosti `session`. Ovlivnit tím lze například dobu, po kterou je uživatel přihlášen. Hodnota je udávána v minutách (na příkladu nastaveno 6 hodin).

```
<session-config>
  <session-timeout>
    360
  </session-timeout>
</session-config>
```

Při mapování servletů je možné definovat i anotace potřebné pro různé knihovny. V případě uvedeném níže jde o konfiguraci přenosu z formulářů typu *multipart*. Definována je cesta, kam mají být soubory ukládány, maximální velikost souboru, maximální velikost požadavku a údaj o tom, po jak velkých částech má být soubor zapsán na disk.

```
<servlet>
  <servlet-name>GalleryAdminServlet</servlet-name>
  <servlet-class>servlets.stag.GalleryAdminServlet</servlet-class>
  <multipart-config>
    <location>c:\webfiles\files\gallery</location>
    <max-file-size>15728640</max-file-size>
    <max-request-size>786432000</max-request-size>
    <file-size-threshold>1048576</file-size-threshold>
  </multipart-config>
</servlet>
```

Celý DD je umístěn na příloženém CD.

6 Nasazení systému

Následující kapitola popisuje kroky nutné pro nasazení systému do provozu. Hardwarové nároky jsou uvedeny v kapitole 7.1.

6.1 Instalace potřebných aplikací

Pro nasazení portálu je třeba provést instalaci a nastavení níže uvedených aplikací. V kapitole 7.2 jsou uvedeny verze, se kterými byla kompatibilita software odzkoušena.

- **MySQL**
Vytvoření databáze prostřednictvím dodaného souboru s příkazy SQL.
- **Java SE, Java EE**
- **GlassFish**
Deploy aplikace např. pomocí webového interface serveru Glassfish, který standardně běží na adrese `http://localhost:4848`.

6.2 Instalace certifikátu

Nejsložitější částí instalace je konfigurace JRE a serveru pro komunikaci pomocí LDAP. Komunikaci se serverem Active Directory je možné provést bez šifrování na portu 389, nebo pomocí SSL/TSL na portu 636. Samozřejmě, pro operace změny hesla apod. je vyžadováno SSL, z toho důvodu je třeba provést tyto kroky:

- vytvoření certifikační autority na serveru AD,
- export certifikátu,
- import certifikátu do JRE a nastavení cest v aplikaci.

Na první dva body existuje v OS Windows 2003 Server příslušný průvodce, který uživatelé provede potřebnými kroky. Postup je následující:

1. Otevřít dialog Start -> Ovládací panely -> Přidat a odebrat programy.
2. Zvolit záložku Přidat/odebrat komponenty Windows.
3. Vybrat a instalovat komponentu Certifikačních služeb.
4. Zobrazí se dialog, ve kterém je třeba vybrat *Enterprise root CA* typ certifikátu, vyplnit jméno serveru a dobu platnosti certifikátu.
5. Potvrdit dokončení.
6. V certifikačním úřadu OS Windows provést export vytvořeného certifikátu.

Import certifikátu do úložiště JRE je možné provést příkazem:

```
keytool -import -keystore .\jre\lib\security\cacerts -file  
server-certificate.crt
```

Provedení operace bude vyžadovat zadání hesla k úložišti. Úspěšné zpracování je potvrzeno výpisem informací o importovaném certifikátu. Příklad výstupu:

```
Owner: CN=akemi  
Issuer: CN=akemi  
Serial number: 42e5b3c4  
Valid from:  
Mon Jul 25 22:53:45 MDT 2012 until: Sun Oct 23 22:53:45 MDT 2012  
  
Certificate fingerprints:  
MD5:  
99:8F:14:C5:BB:21:86:77:D2:CF:56:DE:98:DD:74:62  
SHA1:  
EC:59:92:E9:1F:8A:A6:0A:85:54:EC:76:47:DB:5F:3F:D2:15:78:77  
Trust this certificate? [no]: yes  
  
Certificate was added to keystore
```

Nakonec je nutné upravit cesty v aplikaci tak, aby ukazovali na použité úložiště. Umístěny jsou ve třídě ADAdm a v DD.

6.3 Vytvoření potřebných adresářů, úprava Deploy Descriptoru

Pro ukládání souborů portál využívá adresářovou strukturu, která je součástí dodané distribuce. Po jejím nasazení je třeba nastavit platné cesty do DD.

Protože lze řešení nasadit i na OS Linux, je třeba pamatovat při úpravě cest na rozdílnou adresářovou strukturu.

6.4 Omezení funkcí na OS Linux

Přestože je aplikace určena pro OS Windows Server, byla její funkcionality v průběhu vývoje testována také pod OS Linux (viz kapitola 3.1.4). Navržené řešení má při nasazení na OS Linux tato omezení:

- nelze zjišťovat dostupnost síťových zařízení,
- nelze síťová zařízení vypnout.

Pro zajištění funkce monitorování sítě by bylo třeba použít v implementaci místo aplikace ping další z postupů popsaných v kapitole 5.9. Pro vzdálené vypínání OS Windows není alternativa dostupná.

7 Testování

Pro potřeby testování je aplikace v průběhu vývoje nasazena v organizaci:

*Středisko volného času dětí a mládeže Plzeň,
pracoviště Sady pětatřicátníků – Stanice mladých techniků. [24]
(data jsou uvedena v tabulce 7.1).*

Jako aplikační server je zvolen Glassfish (viz kapitola 3.4.3), instalovaný na jednotlivé operační systémy (viz kapitola 3.1). S výjimkou OS Windows Server 2003 jsou systémy virtualizovány nástrojem Oracle VM VirtualBox [25] s přidělenými prostředky uvedenými v kapitole 7.1.1.

Prostředí nasazení odpovídá modelu obecného DDM, v tomto případě:

- 70 PC, z toho 50 ve třech samostatných učebnách a 20 rozmístěných samostatně
- 5 tiskáren, z toho 2 síťové
- Kamerový systém, 4 IP kamery + aplikace Motion
- 3 bezdrátové přístupové body (WIFI AP)
- 8 aktivních interních zaměstnanců, 10 externích

Parametr	Hodnota
Datum nasazení	1. 1. 2011
Návštěvníků	12 000
V provozu	500 dní

Tabulka 7.1: Doba testování.

7.1 Hardwarové nároky

V této kapitole jsou uvedeny systémové nároky pro provoz serverové i klientské části aplikace. Vzhledem k faktu, že jde o webovou aplikaci, nelze přesně definovat kritické parametry. Proto uvedené údaje odpovídají prostředkům, na kterých byl systém bez problémů odzkoušen. Pro zvýšení zátěže byly také prováděny simulované požadavky. Cílové zatížení aplikace nelze též předpovědět z důvodu nízkého využití aktuálních systémů¹⁶ pracujících ve výše uvedené organizaci.

¹⁶ Webový server vykazuje průměrnou hodnotu 800 návštěv denně. V tomto čísle je započteno i využití intranetem a návštěvy robotů.

7.1.1 Server

Dlouhodobě je portál provozován na OS Windows Server 2003 s hardware:

- Intel® Core™2 Quad Processor Q9550 (12Mb, 2.83 GHz, 1333 MHz FSB)
- 2048 MB DDR2 RAM (800 MHz)

Dále byl systém testován ve virtualizovaných OS s hw parametry:

- Intel® Core™ i7-950 Processor (8MB, 3.06 GHz, 4.80 GT/s Intel® QPI)
(OS přidělena dvě jádra procesoru)
- 4096 MB DDR3 RAM (1333 MHz)

Bez zátěže je procesy alokováno 35 MB pro MySQL a 285 MB pro server GlassFish.

7.1.2 Klient

Protože výstup aplikace tvoří webové rozhraní, jsou nároky na provoz klientské části odvozeny od nároků použitého prohlížeče. Prostředí je optimalizováno pro zobrazovač s rozlišením 1024 x 768 pixelů.

7.2 Verze software

Tabulka 7.2 udává verze jednotlivých prostředků, které jsou využity pro implementaci projektu a se kterými je nasazený portál testován.

Software	Verze
GlassFish	3.1.2
MySQL	5.5.20
Java	1.6.31

Tabulka 7.2: Verze software.

Přestože je výsledný HTML kód validní, dochází k běžným problémům s kompatibilitou starších prohlížečů a k chybám v zobrazení. Z těchto důvodů je výstup optimalizován pro nejrozšířenější prohlížeče uvedené v tabulce 7.3.

Prohlížeč	Verze
Internet Explorer	9
Opera	> 9
Firefox	> 10
Safari	5.1.7
Google Chrome	18.0

Tabulka 7.3: Kompatibilní verze prohlížečů.

7.3 Testování kompatibility

Testování kompatibility slouží k zajištění stability provozu s různými systémy, aplikacemi, prohlížeči apod. Výsledky testů definují problematická místa aplikace, což vede buď k jejich upravení pro zajištění kompatibility, nebo vytvoření záznamu o problému. K testování kompatibility lze využít automatizovaných nástrojů, nebo může být prováděno manuálně jako součást mimofunkčního testování.

V tomto projektu je testována kompatibility:

- prohlížečů,
- verzí,
- operačních systémů.

Testování OS a verzí použitého podpůrného software je prováděno manuálně, pro testování kompatibility s prohlížeči je využíváno nástroje BrowserShots [26].

7.4 Testování použitelnosti

Použitelnost webových stránek určuje, jak jsou pochopitelné pro návštěvníky. Dobře použitelné webové stránky jsou na první pohled přehledné, ovládání je intuitivní a požadovaná data jsou dosažitelná bez komplikovaného vyhledávání.

Protože lze použitelnost webových stránek předem odborně určit pouze částečně, je prováděno tzv. „*uživatelské testování použitelnosti*“. Základem testu je tvorba scénářů, které na konkrétním webu definují cíl akce. Tyto scénáře jsou předloženy vzorku uživatelů, na nichž jsou sledovány reakce a dokumentován průběh plnění úkolu. [27]

Pro tento projekt byly tvořeny tematické úkoly:

- vyhledání informací z webu a celkový dojem,
- orientace v administrativním prostředí,
- pedagogická obsluha funkcí portálu,
- práce s formuláři.

Úlohy byly řešeny týmem ve funkčním složení:

- čtyři pedagogičtí pracovníci,
- dvě organizátorky akcí,
- systémový administrátor,
- dva externisté.

Celé uživatelské prostředí bylo poté přepracováno dle výsledků a zjištění jednotlivých testů. Pro maximalizaci výsledku je tento test pravidelně opakován a kombinován s testováním kompatibility (viz výše).

7.5 Metriky

Do této kapitoly je zařazeno několik metrik, které sice žádným způsobem necharakterizují kvalitu kódu nebo projektu, je však pomocí nich možno nastínit rozsah projektu (viz tabulka 7.4)

Metrika	Hodnota
Počet souborů	185
Počet řádek	25 000
Poměr kódů/komentářů	73%
Počet obrazovek	cca 50
Počet formulářů	cca 30

Tabulka 7.4: Metriky projektu.

7.6 Zhodnocení výsledků

Z rozsahu definice požadavků je patrné, že hlavní důraz při vývoji informačního systému byl kladen na snadnou obsluhu a okamžité výsledky v úspore času. Toto bylo zohledněno též při testování, proto nejvíce využívanými funkcemi byla správa učeben a vedení docházky do klubu, který v testovacím prostředí funguje.

Modul správy učeben se stal součástí ukončení výuky každého počítačového zájmového útvaru, jejichž činnost prodloužil cca o 5 minut. Dále je součástí procesu ukončení činnosti večer, kdy centrální kontrola IT ušetří každý den cca 20 minut času.

Vedení statistik docházky zjednodušilo systém zadávání a vedení dat, nejpodstatněji se však úspora projevuje při generování statistiky na konci roku. Manuální výpočet zabral dříve dva pracovní dny jednoho zaměstnance (cca 16 hodin), portálem jsou data k dispozici za libovolné období trvale a s množstvím užitečných informací navíc.

Dále jsou nejvíce oceňovány funkce galerie, která přináší automatizaci v podobě tvorby alb a práce s miniaturami. Ty jsou v ostatních případech vždy tvořeny ručně.

U ostatních funkcí nelze udat konkrétní časové úspory či přínos, protože přináší funkce, které ani jedno z analyzovaných pracovišť do této chvíle nevyužívalo.

8 Závěr

První část práce se zabývá analýzou prostředí, ve kterém organizace pracující s dětmi a mládeží provozují své informační systémy, a služeb, jejichž implementace by přinesla úspory času a usnadnění používání technologií zaměstnanci. Na základě prostudování technologických nároků a dostupných možností je jako nejvhodnější platforma pro vývoj vybrána Java Enterprise Edition, provozována na OS Windows Server spolu s databází MySQL.

Protože se dnes žádný větší projekt neobejde bez širokého spektra knihoven, které usnadní dílčí části vývoje, je součástí práce kromě popisu použitých frameworků též rozbor nejčastěji používaných licencí. Z výsledku vyplývá, že lze slušně spoléhat na produkty licencované MIT, BSD či AL, jelikož lze snadno zaručit splnění licenčních podmínek.

Další části se již zabývají samotnou specifikací jednotlivých funkcí navrhovaného řešení dle analyzovaných potřeb organizací. Mimo samozřejmého navržení prezentačního webu zahrnuje portál funkce monitorování zapnuté techniky v učebnách, služby vzdáleného vypnutí této techniky, správu doménových účtů návštěvníků a systém agendy zaměstnanců. Pro jednotlivé moduly i funkce lze individuálně nastavovat oprávnění k přístupu. Nasazením portálu je tak dosaženo úspor na čase, materiálu a v konečném důsledku financí. Protože jde o prakticky uplatnitelnou realizaci, jsou navíc zpracovány další funkční i mimofunkční požadavky, jako webový design, optimalizace docházkového systému apod.

Pro podporu interakce s uživatelem a rozšíření možností HTML integruje řešení množství JavaScriptových knihoven a skriptů. Použitím knihovny jQuery bylo docíleno intuitivnějšího chování formulářů, použití vybraných částí YUI pak podporuje práci se záložkami, tabulkami a též jsou využity její části z oblasti výběru a přenosu velkého množství souborů najednou.

Přestože jde pouze o prototyp, byl nasazen v rámci testovacího prostředí do ostrého provozu a nejdůležitější funkce byly pro ověření spolehlivosti používány po celou dobu vývoje. K tomuto účelu byla jako součást práce vytvořena graficky vhodná šablona dle cílového nasazení. V závěru práce je proto uveden popis testovacího prostředí, metodiky, hardwarové nároky aplikace i výsledná kompatibilita s prohlížeči.

Použité zkratky a slovník pojmů

Zkratka	Význam
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
ASP	Active Server Pages
CSS	Cascading Style Sheets
DAO	Data access object
DB	DataBase
DDM	Dům dětí a mládeže
DIT	Directory Information Tree
DN	Distinguished Name
HTML	Hypertext Markup Language
HQL	Hibernate Query Language
IDE	Integrated Development Environment
IS	Information System
JDBC	Java Database Connectivity
JNDI	Java Naming and Directory Interface
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
MVC	Model-view-controller
OOP	Object- oriented programming
ORM	Object-relational mapping
OS	Operating System
SSI	Server-Side Includes
SSL	Secure Sockets Layer
SVN	Subversion
XHTML	Extensible HyperText Markup Language

Tabulka 1: Zkratky.

Výraz	Význam
Deploy	Nasazení verze aplikace na server.
Framework	Programový celek, sloužící jako podpora vývoje v určité oblasti.
Session	U protokolu HTTP se jedná o schopnost serveru držet kontext.

Tabulka 2: Slovník pojmů.

Literatura

1. *Bouška, Petr*: **Komponenty AD** [Online][Citace: 12. 11 2011]
<http://samuraj-cz.com/clanek/active-directory-komponenty-domain-tree-forest-site>
2. *Red hat inc.*: **Red Hat Enterprise Linux**. [Online][Citace: 26. 3 2011]
<http://www.redhat.com/products/enterprise-linux>
3. *Oracle Corp.*: **Oracle Database 11g Express Edition**. [Online][Citace: 27. 4 2012]
<http://www.oracle.com/technetwork/products/express-edition/overview/index.html>
4. **NetBean**. [Online][Citace: 17. 11 2011]
<http://netbeans.org/about/index.html>
5. **Eclipse**. [Online][Citace: 17. 11 201.]
<http://eclipse.org/org/>
6. *Lang, Jean-Philippe*: **Redmine**. [Online][Citace: 12. 11 2011]
<http://www.redmine.org>
7. *Corel*: **CorelDRAW Graphics Suite Tutorials**. [Online][Citace: 22. 9 2011]
<http://www.corel.com/corel/pages/index.jsp?pgid=800382&storeKey=ca>
8. *Adobe* : **Photoshop Elements 7 tutorial**. [Online][Citace: 22. 9 2011]
<http://adobe.com/designcenter-archive/photoshopelements/>
9. **Coelho, Everaldo**. [Online][Citace: 17. 11 2011]
<http://www.everaldo.com/>
10. *Hall, Marty*: **Core Servlets and JavaServer Pages (JSP)**
Praha : Neocortex spol. s. r. o., 2001. ISBN 80-86330-06-0.
11. *Písek, Slavoj*: **JAVASCRIPT efektní nástroj oživení www stránek**.
Praha: Grada Publishing, spol. s.r.o, 2001. ISBN 80-247-0014-X.
12. *Seznam.cz a.s.*: **Jak - javascriptová knihovna**. [Online][Citace: 26. 3 2012]
<http://jak.seznam.cz/>

13. **The jQuery Project** [Online][Citace: 1. 5 2012]
<http://jquery.com/>
14. *Miles, David*: **AD Gallery – a jQuery gallery plugin.** [Online] [Citace: 1. 5 2012]
<http://coffeescripiter.com/2009/07/ad-gallery-a-jquery-gallery-plugin/>.
15. **Fancybox** [Online][Citace: 1. 5 2012]
<http://fancybox.net/>
16. *Potts, Kevin*: **Website Language Dropdown With jQuery. *graphicPush*.**
[Online][Citace: 1. 5 2012]
<http://www.stjerneman.com/demo/maxlength-with-jquery>
17. *Stjerneman, Emil*: **Maxlength with jQuery.** [Online][Citace: 1. 5 2012]
<http://www.stjerneman.com/demo/maxlength-with-jquery>
18. *Zaefferer, Jörn*: **jQuery plugin: Tooltip.** [Online][Citace: 1. 5 2012]
<http://bassistance.de/jquery-plugins/jquery-plugin-tooltip/>
19. *Yahoo! Inc.*: **YUI Library.** [Online][Citace: 10. 5 2012]
<http://developer.yahoo.com/yui/>
20. *Creative Commons Česká republika*: **Licenční prvky.** [Online] [Citace: 27. 3 2012]
<http://www.creativecommons.cz/zakladni-informace-o-cc/licencni-prvky/>
21. *Cafourek, Bohdan a Kačmář, Dalibor*:
Microsoft Windows Server 2003 Nové funkce, upgrade, implementace.
Brno : Computer Press, 2003. ISBN 80-7226-582-2.
22. *Fujii, Keita*: **Introduction. *Jpcap*.** [Online][Citace: 10. 5 2012]
<http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/>
23. *Netter, Michael*: **ICMP Ping in Java.** [Online][Citace: 10. 5 2012]
<http://www.0x13.de/index.php/about-me.html>
24. *Středisko volného času dětí a mládeže Plzeň*: **O nás.** [Online] [Citace: 6. 4 2012]
<http://svcpl.cz/>
25. *Oracle Corp.*: **VirtualBox.** [Online] [Citace: 11. 5 2012]
<https://www.virtualbox.org/wiki/VirtualBox>

26. **Browser Compatibility Test** [Online][Citace: 10. 5 2012]
<http://browsershots.org>
27. **8 guidelines for usability testing** [Online][Citace: 10. 5 2012]
<http://webcredible.co.uk/user-friendly-resources/web-usability/usability-testing.shtml>
28. *The Apache Software Foundation: Apache Tomcat* [Online][Citace: 1. 5 2012]
<http://tomcat.apache.org/>
29. *Red Hat, Inc.: JBoss Enterprise Application Platform* [Online][Citace: 1. 5 2012]
<http://www.redhat.com/products/jbossenterprisemiddleware/application-platform/>
30. *Doc. Ing. Pavel Herout, Ph.D.: Java - bohatství knihoven.*
České budějovice : KOOP, 2006. ISBN 80-77232-288-5.
31. *Malina, Patrik: Microsoft Windows Server 2003 Hotová řešení.*
Brno : Computer Press, a.s., 2006. ISBN 80-251-1096-6.
32. *Osif, Michal: Windows Server 2003 PORADCE EXPERTA.*
Praha : Grada Publishing, a.s., 2003. ISBN 80-247-0396-3.

Seznam příloh

Příloha A	Uživatelská dokumentace
Příloha B	Obrazovky a uživatelské rozhraní
Příloha C	Obsah CD ROM

A. Uživatelská dokumentace

Obecné

Zde naleznete nejdůležitější obecné poznámky k ovládání portálové aplikace. Aplikace je dostupná přes Váš internetový prohlížeč na adrese, kterou Vám sdělí systémový administrátor.

Přihlášení

Pro přihlášení je v záhlaví každé stránky umístěn přihlašovací formulář. Pro vstup do systému zadejte Vaše uživatelské jméno a heslo. Tyto údaje jsou shodné s těmi, které užíváte pro přihlášení k počítači. Pokud se k vašemu počítači nepřihlašujete, požádejte o vytvoření účtu vašeho administrátora.

Moduly a funkce

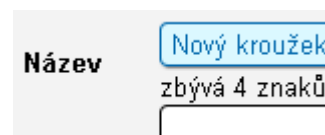
Portál je dělen na tzv. moduly a funkce. Modulem je nějaký větší celek, zastřešující více podobných funkcí. Např.: modul *Web* obsahuje funkce správy prezentací novinek, akcí, nahrávání fotografií do galerie apod.

Formuláře

Součástí aplikace je množství formulářů pro vkládání potřebných dat. Pro zjednodušení jejich obsluhy je implementováno množství mechanismů, které Vám usnadní práci s nimi.

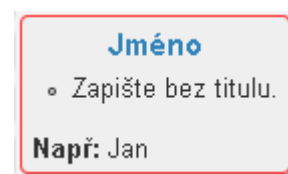
- **Délka textu**

Každý textový vstup je opatřen ukazatelem počtu znaků, které je ještě možno zadat. Pro přehlednost je počítadlo zobrazeno až při editaci daného pole. Při vyčerpání celkové délky je zápis dalšího textu blokován a tento stav je též indikován vizuálně, změnou barvy pole na červenou.



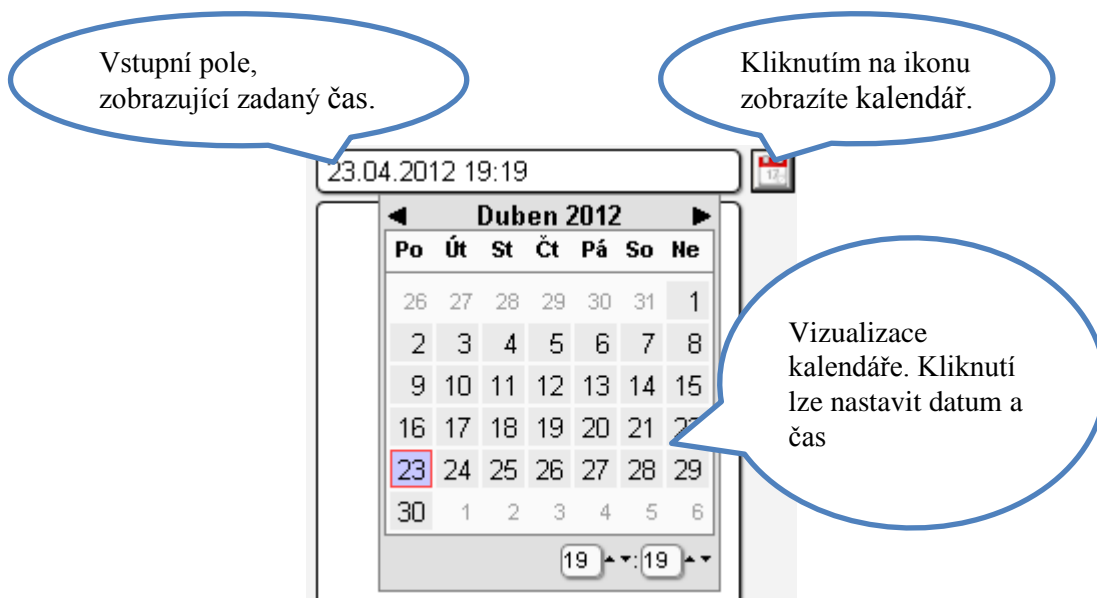
- **Bublinová nápověda**

Pro zjednodušení vyplňování údajů jsou formuláře vybaveny systémem bublinových nápověd, které Vám pomohou s přesným zadáním údaje. Bublina se



zobrazí při najetí myši nad editační pole. Zmizí při přejetí kurzorem jinam, nebo při kliknutí do pole.

- **Datum a čas**



Vklad data a času je řešen vizuálním kalendářem. Tento způsob zadávání Vám umožní jednoduché a přehledné vložení a zároveň zabezpečuje správnost vstupu, protože výsledný řetězec je pouze generován a nelze jej editovat.

Rozložení okna

Informace
Přihlášen/a: David Adam
Svátek má: Pankrác
Odhlásit

Zaměstnanci Klub Vedoucí Servis Web Docházka Administrace

Tábory
Nový běh tábora Nový druh tábora

Běhy	Druhy	Akce	Název	Začátek	Konec	Obsazenost
		X	1. běh - Darwinova přírodovědná expedice.	11.07.2011	15.07.2011	30
		X	2. běh - Hledání zlatého pokladu.	18.07.2011	22.07.2011	30
		X	3. běh - Cesta kolem Plzně.	25.07.2011	29.07.2011	24
		X	4. běh - Malý turista.	01.08.2011	05.08.2011	7
		X	5. běh - Pat a Mat ve Stanici.	08.08.2011	12.08.2011	27
		X	6. běh - Digitální odyssea.	15.08.2011	19.08.2011	17
		X	1. běh - Na kvildě.	11.07.2011	20.07.2011	10

Zobrazeno 7 záznamů.

Novinky Akce Tábory Soutěže Galerie Kroužky Kontakty Upozornění mailem

1. Přihlašovací formulář

Dialog pro přihlášení a odhlášení. V prezentační části webu zobrazuje pole pro přihlášení, v administrační části zobrazuje jméno aktuálně přihlášeného uživatele, údaj o tom, kdo má příslušný den svátek a tlačítko na odhlášení.

2. Hlavní menu

Slouží k přepínání mezi moduly aplikace. V prezentační části portálu slouží jako menu pro jednotlivé aktivity, po přihlášení jsou v něm uvedeny moduly portálu. Aktivity jsou společné pro všechny uživatele a rozsah jejich zobrazení nelze nastavit. Přístup do jednotlivých modulů (a tím i jejich viditelnost v menu) je dán nastavením Vašich oprávnění.

3. Kontextové menu

Pokud modul obsahuje další funkce a máte k nim oprávnění přistupovat, jsou zobrazeny v tomto menu.

4. Tlačítka

Prostředí obsahuje množství odkazů, navigačních prvků apod., mezi které patří i tlačítka. Pro lepší orientaci v aplikaci jsou vždy umístěna nad tabulkami, ke kterým se vztahují a ve většině případů slouží k zobrazení formuláře pro zadání nového údaje do tabulky. Naopak ve formulářích jsou tlačítka zobrazena pod editačními políčky.

5. Záložky

Pokud funkce obsahuje více než jednu tabulku, je výpis opatřen záložkami, sloužícími pro přechod mezi tabulkami.

6. Tabulky

Data jsou vypisována formou tabulek. Každá tabulka obsahuje zcela vlevo sloupec s tlačítky pro úpravy a mazání jednotlivých záznamů. Každý sloupec je opatřen záhlavím, které kromě nadpisu sloupce slouží k seřazení tabulky (seřazení provedete kliknutím).

7. Přepínač jazyků

Zobrazená ikona značí, jaký jazyk zrovna aplikace zobrazuje. Po kliknutí se Vám zobrazí seznam dostupných lokalizací.

Moduly a jejich funkce

Zaměstnanci

Zde můžete nastavit účty zaměstnanců. Kromě osobních údajů je zde vyplněna funkce v organizaci, kterou pracovník zastává. Ta je následně využita ke generování jeho docházkových formulářů. Po vyplnění je zaměstnanec schopen se k aplikaci přihlásit a využívat funkce, které mu administrátor definuje v modulu administrace. Modul Vám také vypisuje informace o přihlašování do portálu.

Klub

V modulu máte k dispozici administraci klubové činnosti Vaší organizace. Můžete zde spravovat registrované uživatele a následně evidovat jejich přítomnost v klubu. Modul Vám automaticky vygeneruje přehledné statistiky o návštěvnosti a využívání služeb jednotlivými návštěvníky.

Vedoucí

Modul je určený pro vedoucí technických zájmových útvarů. Můžete zde spravovat konta jednotlivých uživatelů pro přihlašování do systému Windows, resetovat hesla a zjišťovat poslední přihlášení uživatelů. Modul Vám umožní také vzdálené vypnutí počítačů po výuce.

Servis

Tento modul slouží k evidenci servisních úkonů prováděných v organizaci. K dispozici je zde i výstup kamerového systému (pokud jej využíváte). V modulu můžete také zjistit, kde máte zapnutá která zařízení (počítače, kamery, servery...).

Web

Modul slouží pro úpravy údajů v prezentační části webu. Samostatně můžete nastavovat záznamy pro novinky, akce, zájmové útvary, soutěže a tábory. Dále je zde konfigurace údajů o organizaci, správa fotografií v galerii a rozesílání mailových zpráv.

Docházka

Modul slouží k evidenci docházky. Pro každou smlouvu definovanou v profilu zaměstnance Vám generuje každý měsíc příslušný formulář. Dále Vám umožní přístup k archivním údajům. Pro kontrolu disponuje modul funkcí uzávěrky, která Vám zobrazí přehled stavu plnění formulářů jednotlivými zaměstnanci.

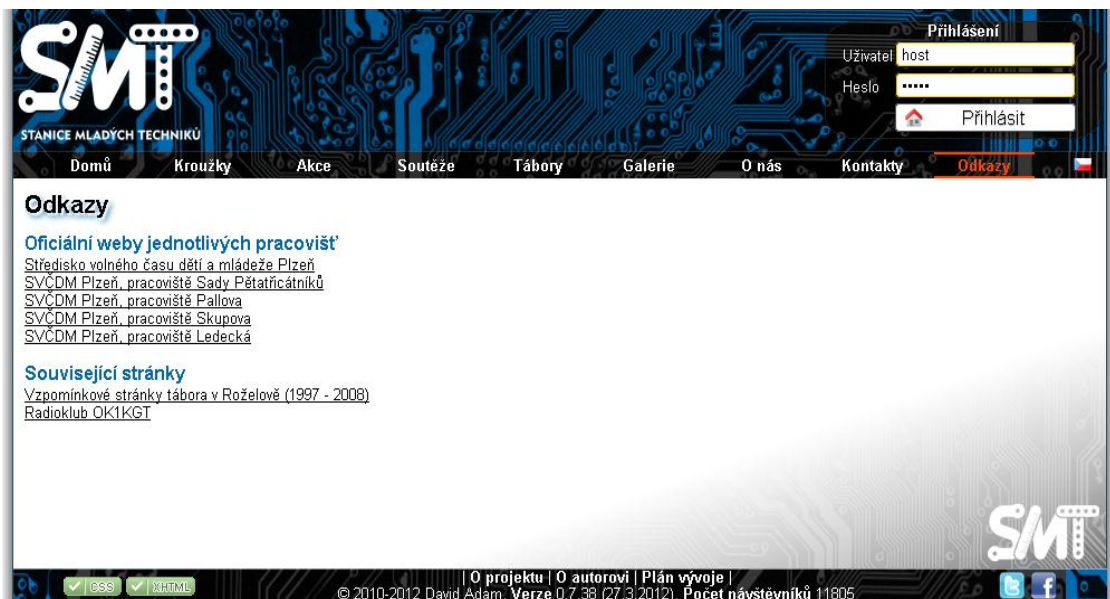
Administrace

Modul je dostupný pouze pro administrátora systému. Po vstupu Vám zobrazí jednotlivé zaměstnance a tabulky s nastavením jejich oprávnění. Kliknutím na funkce systému u jednotlivých uživatelů lze oprávnění přidávat a odebírat. Změny se projeví až po novém přihlášení upravovaného uživatele do systému.

B. Obrazovky a uživatelské rozhraní

Celkové rozložení stránky

Na obrázku 1 je zobrazeno základní rozložení portálu. V záhlaví je umístěn přihlašovací dialog a logo organizace. Mezi záhlavím a obsahem je hlavní menu, ve kterém je trvale vizuálně vyznačena část webu, na které se návštěvník právě nachází. Pod obsahem je zápatí s obecnými informacemi o projektu a zkratkami na další informační zdroje.



Obrázek 1: Rozvržení stránky, design testovacího nasazení.

Docházkový výkaz

Obrázek 2 zobrazuje příklad formuláře pro výkaz práce. V horní části je umístěn výběrový seznam, který umožní se přepnout na předešlé výkazy. Následuje samotný formulář. Dále je zobrazen dialog vstupu pro zadání času u příchodů a odchodů.

Vyberte datum 2012 duben Vybrat

EVIDENCE PRACOVNÍ DOBY										Stanice mladých techniků																																				
Pracovní zařazení Správce sítě SVČ					Příjmení, jméno, titul					David Adam																																				
Měsíc	duben				Počet pracovních dnů	20		Pracovní úvazek	0.5		Počet hodin za měsíc	80.0																																		
DAT	PŘÍCHOD HOD	ODCHOD HOD	CELKEM HOD	PŘÍMÁ PRÁCE	SUPL.	VÝLUKA	AKCE	PŘESČAS	PŘESP. HODIN	DOVOL.	NEMOC	NV	DĚLENÁ SMĚNA	SOUKR.	POZNÁMKA																															
1	07:00	12:00	5.0	0.0	0	0	0	0	0	0	0	5.0	0	0																																
2	08:00	13:00	5.0	0.0	0	0	0	0	0	0	0	1.0	0	0																																
3	07:00	12:00	5.0	0.0	0	0	0	0	0	0	0	1.0	0	0																																
4	08:00	11:00	3.0	0.0	0	0	0	0	0	0	0	-1.0	0	0																																
5	07:00	12:00	5.0	0.0	0	0	0	0	0	0	0	1.0	0	0																																
6	<div style="display: flex; justify-content: space-between;"> Hodiny Minuty </div>																																													
7	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="width: 20px;"></td> <td style="width: 20px;">00</td> <td style="width: 20px;">01</td> <td style="width: 20px;">02</td> <td style="width: 20px;">03</td> <td style="width: 20px;">04</td> <td style="width: 20px;">05</td> <td style="width: 20px;">00</td> </tr> <tr> <td>8</td> <td>06</td> <td style="background-color: #ffff00;">07</td> <td>08</td> <td>09</td> <td>10</td> <td>11</td> <td>15</td> </tr> <tr> <td>9</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> <td>30</td> </tr> <tr> <td>10</td> <td>18</td> <td>19</td> <td>20</td> <td>21</td> <td>22</td> <td>23</td> <td>45</td> </tr> </table>															00	01	02	03	04	05	00	8	06	07	08	09	10	11	15	9	12	13	14	15	16	17	30	10	18	19	20	21	22	23	45
	00	01	02	03	04	05	00																																							
8	06	07	08	09	10	11	15																																							
9	12	13	14	15	16	17	30																																							
10	18	19	20	21	22	23	45																																							
11	07:00	12:00	5.0	0.0	0	0	0	0	0	0	0	1.0	0	0																																
12	07:00	12:00	5.0	0.0	0	0	0	0	0	0	0	1.0	0	0																																
13	07:00	12:00	5.0	0.0	0	0	0	0	0	0	0	1.0	0	0																																

Obrázek 2: Formulář docházkového výkazu.

Tabulky a záložky

Obrázek 3 zobrazuje příklad výpisu číselníku běhu táborů. Nad tabulkou jsou tlačítka pro vklad nových záznamů a záložky pro přepínání mezi souvisejícími číselníky. Data v tabulce lze řadit dle libovolného sloupce, a záložky přepínat bez nutnosti znovu načíst stránku.

+ Nový běh tábora
+ Nový druh tábora

Běhy
Druhy

Akce	Název	Začátek	Konec	Obsazenost
X	1. běh - Darwinova přírodovědná expedice.	11.07.2011	15.07.2011	30
X	2. běh - Hledání zlatého pokladu.	18.07.2011	22.07.2011	30
X	3. běh - Cesta kolem Plzně.	25.07.2011	29.07.2011	24
X	4. běh - Malý turista.	01.08.2011	05.08.2011	7
X	5. běh - Pat a Mat ve Stanici.	08.08.2011	12.08.2011	27
X	6. běh - Digitální odyssea.	15.08.2011	19.08.2011	17
X	1. běh - Na kvildě.	11.07.2011	20.07.2011	10

Zobrazeno 7 záznamů.

Obrázek 3: Tabulky a záložky.

Galerie

Obrázek 4 zobrazuje informování uživatele o stavu nahrávání souborů na server. V horní části jsou umístěny dialogy pro výběr fotografií a tvorbu nové galerie, ve spodní, tabulka vybraných fotografií spolu s ukazatelem stavu jednotlivých souborů.

Galerie

Upload souborů do aktuální galerie

Obsazené místo: 257,1 MB z 46,1 GB

Vybrat soubory Nahrát

Hotovo: 0/3

Tvorba nové podgalerie

Vytvořit zde galerii

Soubory k nahrání

Soubor	Velikost	Stav
DSC_0129.jpg	4864135	<div style="width: 100%; height: 10px; background-color: #ccc; position: relative;"><div style="width: 100%; height: 100%; background-color: #f00;"></div></div>
DSC_0127.jpg	5606614	Připraven
DSC_0128.jpg	5436386	Připraven

Obrázek 4: Proces nahrávání souborů na server.

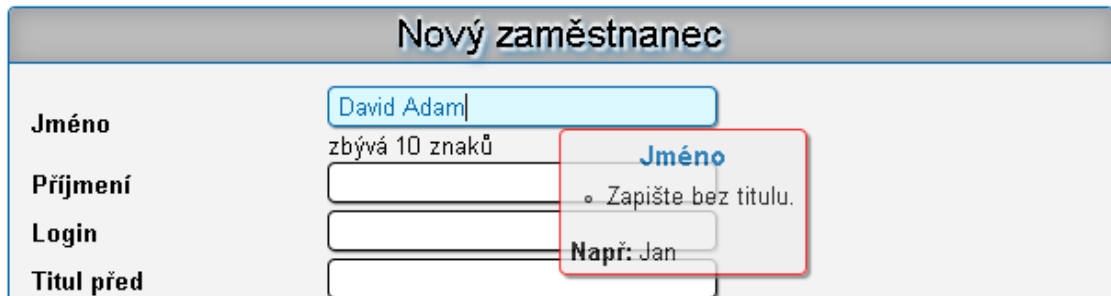
Na obrázku 5 je zobrazen dialog práce se složkami galerie. V levé části je seznam alb, umístěných v aktuální složce, uprostřed informace o aktuálním albu. Vpravo je umístěn přepínač režimů zobrazení.

<p>Alba</p> <p>Akce</p> <p>Tábory</p>	<p>Galerie</p> <p>Zde naleznete fotografie z činností, které pravidelně (i nepravidelně) provozujeme. Prohlédnout si můžete naše jednotlivé akce, tábory či jen projít činnost na zdokumentovaných kroužcích.</p> <p>Vytvořil: Bc. David ADAM , 25. říjen 2011 fotografií: 262 velikost: 257,1 MB (včetně vnořených alb)</p>	<p>Zobrazení</p> <p></p> <p></p>
--	--	---

Obrázek 5: Dialog procházení alby.

Formuláře, nápovědné systémy

Obrázek 6 zobrazuje systém bublinové nápovědy a počítadlo zbývajících znaků vstupního pole.

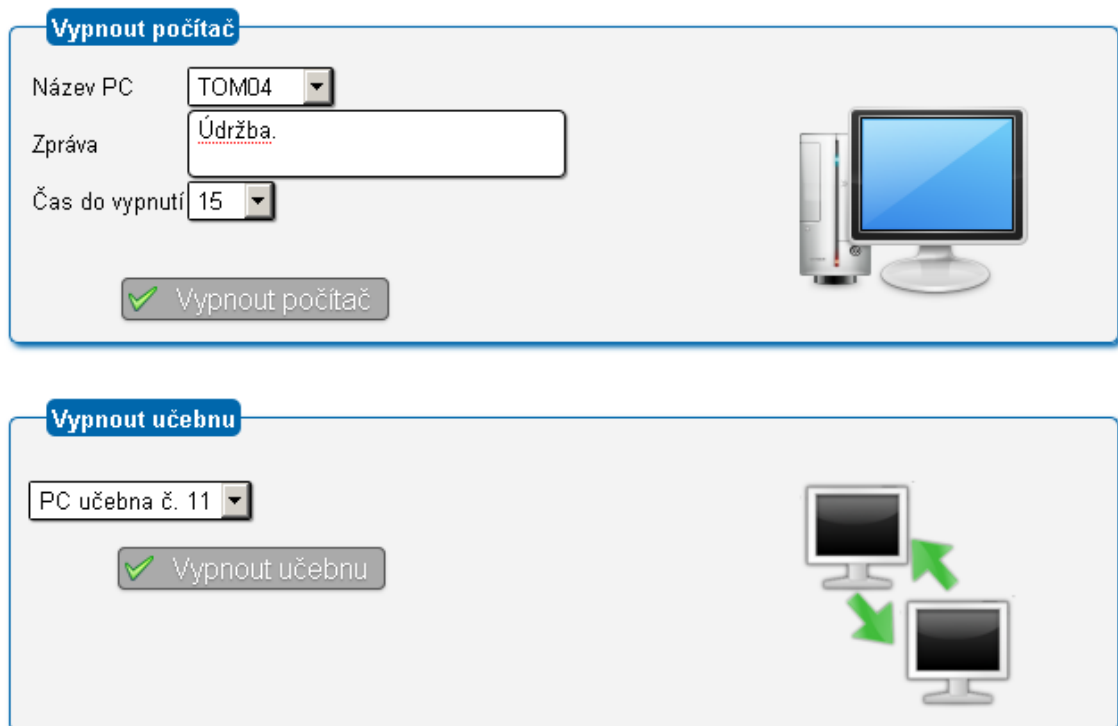


The image shows a web form titled "Nový zaměstnanec" (New employee). It has four input fields: "Jméno" (Name) containing "David Adam", "Příjmení" (Surname), "Login", and "Titul před" (Title prefix). A tooltip is displayed over the "Jméno" field, containing the text "Jméno" and a bullet point "• Zapište bez titulu." (Write without title). Below the tooltip, the text "Např: Jan" (Example: Jan) is visible. A character count "zbývá 10 znaků" (10 characters remaining) is shown below the "Jméno" field.

Obrázek 6: Nápovědné prostředky.

Vypínání PC

Obrázek 7 zobrazuje dialogy příkazů na vypnutí PC nebo učebny.




The image shows two dialog boxes. The first, titled "Vypnout počítač" (Shut down PC), has a dropdown menu for "Název PC" (PC name) set to "TOM04", a text input for "Zpráva" (Message) containing "Údržba." (Maintenance.), and a dropdown for "Čas do vypnutí" (Time to shut down) set to "15". A "Vypnout počítač" button with a green checkmark is at the bottom. An illustration of a PC tower and monitor is on the right. The second dialog, titled "Vypnout učebnu" (Shut down classroom), has a dropdown for "PC učebna č." (PC classroom no.) set to "11". A "Vypnout učebnu" button with a green checkmark is at the bottom. An illustration of two monitors with green arrows between them is on the right.


Obrázek 7: Dialog vypínání PC a učebny.


Servisní záznamy


Na obrázku 8 se nachází příklad výstupu servisních záznamů.

17. květen 2011


 **Web JTV**
Přidání jednoduchého captcha systému, snad pomůže proti SPAMu.


 **Paloova - demontáž starého PC**
Rozebrání a likvidace PC po uživateli Pokorová.

 **Instalace PC.**
Uživatel Goesl - likvidace původního PC (TOM95) - kompletně KO. Nahrazení novým z dodávky ČEZ.

 **Paloova - obnova certifikátů.**
Generování nových osobních certifikátů a záloha privátních klíčů pro uživatele Poncarová Anna a Pavlendová Miloslava.

06. květen 2011

 **Certifikáty**
Prodlužování certifikátu - Monika Fremrová.

 **PC uč. 3**
Konfigurace a zařazení do domény tří prototypů v učebně č. 3. Tvorba výzvy k volbě nových aplikací.

Obrázek 8: Servisní záznamy.

Web

Na obrázku 9 je uveden příklad výpisu dialogu v prezentační části webu. Dialog vypisuje jeden typ tábora se všemi jeho běhy. Kromě souhrnných informací obsahuje dynamické části, jako zobrazení aktuálního obsazení tábora nebo odpočet do jeho začátku.



Středisko volného času dětí a mládeže Plzeň
pracoviště sady Pětatřicátníků 3
Stanice mladých techniků

Příměstské tábory **200,- Kč/den**
6 - 13 let

Příměstský tábor se Stanicí mladých techniků je určen pro děti, které nechtějí být doma sami. V letošním roce je každý běh jiný. Každý den se začíná vždy v 8.00 hod. a končí v 16.00 hod. Přijď a nesed' doma.

Běh	Termín	Obsazenost	Do tábora zbývá	Popisek
1. běh	11.07 - 15.07	100% 	je to tu	Darwinova přírodovědná expedice.
2. běh	18.07 - 22.07	100% 	je to tu	Hledání zlatého pokladu.
3. běh	25.07 - 29.07	80% 	je to tu	Cesta kolem Plzně.
4. běh	01.08 - 05.08	23% 	je to tu	Malý turista.
5. běh	08.08 - 12.08	90% 	je to tu	Pat a Mat ve Stanici.
6. běh	15.08 - 19.08	56% 	je to tu	Digitální odyssea.

Obrázek 9: Prezentace organizace - dialog tábora.

C. Obsah CD ROM

/app	potřebné aplikace
/app/netbeans	použité vývojové prostředí NetBeans (verze pro Windows)
/app/java	Java SE, Java EE + GlassFish
/dist	nasaditelný balíček ve formátu war, struktura databáze ve formě sql souboru.
/doc	elektronická verze textu diplomové práce
/prj	projekt pro IDE NetBeans