

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Petr Pokorný

Webový konferenční kalendář

Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Tomáš Vyskočil

Studijní program: Informatika

Studijní obor: Programování

Praha 2012

Děkuji vedoucímu práce Mgr. Tomáši Vyskočilovi za jeho nasazení i za oceánem a všem, kteří mě podporovali v práci.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne

Petr Pokorný

Název práce: Webový konferenční kalendář

Autor: Petr Pokorný

Katedra (ústav): Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Tomáš Vyskočil

e-mail vedoucího: Tomas.Vyskocil@mff.cuni.cz

Abstrakt: Vědecký článek lze přihlásit jen na jednu konferenci. Cílem této práce je vytvořit webovou aplikaci, která pomůže získat přehled v tom, kam lze ještě článek přihlásit v případě neúspěchu, a to hlavně pomocí grafického diagramu. Vlastní text nejprve podrobně specifikuje problém, následně diskutuje volbu jednotlivých technologií a přístup k implementaci. Další část pak stručně popisuje některé implementační detaily a navrhuje možnosti pro rozšíření projektu. Poslední kapitola se pak krátce věnuje technickým požadavkům a licencím. Výsledkem práce je funkční webová aplikace určená do akademických kruhů k vlastní úpravě.

Klíčová slova: web, konference, kalendář, vykreslování

Title: Web Conference Calendar

Author: Petr Pokorný

Department: Department of Applied Mathematics

Supervisor: Mgr. Tomáš Vyskočil

Supervisor's e-mail address: Tomas.Vyskocil@mff.cuni.cz

Abstract: A scientific article can only be submitted to one conference. The purpose of this work is to create a web application which will aid in determining where a given article can still be submitted, using mainly a graphical diagram. The text itself first provides a detailed specification of the problem, then it discusses the choice of individual technologies and the general implementation approach. The following section then briefly describes some implementation details and suggests areas for expansion of the project. The last chapter is dedicated to technical and licensing considerations. The result of the work is a working web application designed for use in academic circles and for further refinement.

Keywords: web, conference, calendar, drawing

Obsah

Obsah	5
Úvod	7
Struktura této práce	7
1 Specifikace	8
1.1 Problém a cílová skupina	8
1.2 Předpoklady a požadavky	9
1.3 Diagram	10
2 Technologie	11
2.1 HTML 5 vs. HTML 4.01	11
2.2 Rastrová vs. vektorová grafika	12
2.3 Programovací jazyk na straně serveru	12
2.4 Frameworky	12
2.4.1 PHP framework	12
2.4.1.1 Nette	12
2.4.1.2 Zend Framework	13
2.4.1.3 CodeIgniter	13
2.4.2 Javascript framework	13
3 Implementace	14
3.1 Architektura	14
3.1.1 MVC	14
3.1.2 MVP	14
3.2 Rozvržení aplikace	15
3.2.1 Modely	15
3.2.2 Controllery	15
3.2.3 Views	16

3.2.4	Vykreslování diagramu	16
4	Známé problémy a náměty pro další rozšíření	18
4.1	Známé problémy	18
4.1.1	Podpora pro více jazyků	18
4.2	Správa uživatelů	19
4.2.1	System práv	19
4.2.2	Skupiny uživatelů	19
4.3	Prohlížení konferencí	19
4.3.1	Další možnosti zobrazení	19
4.3.2	Osobní správa vlastních konferencí	20
4.3.3	Export kalendářových dat	20
5	Technické požadavky a licence	21
5.1	Technické požadavky	21
5.1.1	Technické požadavky na straně serveru	21
5.1.2	Technické požadavky na straně klienta	21
5.2	Licenční ujednání	22
	Závěr	23
	Literatura	24
	Seznam použitých zkratk	26
	Přílohy	27

Úvod

Struktura této práce

Práce čtenáře postupně provádí projektem webového konferenčního kalendáře chronologicky tak, jak vznikal. V úvodu se podrobně věnuje zadání práce, popisuje jednotlivé požadavky na aplikaci a představuje tak řešený problém. V následujících oddílech pak rozebírá volbu technologií pro implementaci, především programovací jazyky a frameworky. Čtenář se dále dozví o návrhu webové aplikace, jednotlivých zvažovaných variantách a výsledné implementaci. Podrobněji se pak práce věnuje vykreslování přehledového diagramu. Následuje zpětná analýza implementace, a návrhy pro další rozšíření aplikace. Krátce se práce věnuje technickým požadavkům na aplikaci a licenčnímu ujednání. V závěru pak práce hodnotí projekt a představuje možnosti pro jeho praktické využití.

Kapitola 1

Specifikace

Tato kapitola představuje podrobné zadání softwarového projektu se všemi předpoklady a požadavky na jeho funkčnost.

1.1 Problém a cílová skupina

Tato práce si klade za cíl vytvořit webový kalendář akademických konferencí. Nevyklučuje se její použití i v jiných sférách, primárně je však určena do akademických kruhů. Vědečtí pracovníci potřebují prezentovat výsledky svých prací na k tomu určených konferencích. Vzhledem k poměru vědecky činných akademiků k počtu konaných konferencí je přirozené, že dochází k nezbytné selekci přihlášených článků, díky čemuž si konference zachovává svůj věhlas a prestiž. Kvůli výše zmíněné selekci však může být obtížné článek prosadit, proto se nabízí otázka využití možnosti vícenásobného přihlášení článku v případě neúspěchu. Právě pro snadnou orientaci v časových souslednostech vědeckých konferencí by měl sloužit webový kalendář konferencí.

1.2 Předpoklady a požadavky

Pro další řešení problému uvažujme následující předpoklady:

- U každé konference lze dohledat následující údaje:
 - Datum, do kterého je třeba přihlásit vědecké články
 - Datum, do kterého přihlášený obdrží rozhodnutí o článku
 - Datum začátku vlastní konference
 - Datum konce vlastní konference
- Každý článek může být současně přihlášen na nejvýše jedné konferenci

Webová aplikace musí nabízet především uživatelský systém a nástroje pro správu jednotlivých konferencí. Zároveň je třeba vytvořit administrátorský účet pro schvalování konferencí. Požadavky na práva uživatelů nechtě jsou tyto:

- Každý může prohlížet kalendář konferencí
- Jen přihlášený uživatel může přidat konferenci
- Jen administrátor může schválit konferenci
- Jen přihlášený uživatel může navrhnout změnu konference
- Jen administrátor může schválit změnu konference
- Jen přihlášený uživatel může navrhnout odstranění konference
- Jen administrátor může odstranit konferenci

Pro přehled konferencí pak bude aplikace nabízet:

1. Standardní kalendářový přehled konaných konferencí pro vybraný měsíc
2. Grafický diagram s přehledem závislostí termínů pro odeslání článků a rozhodnutí o jejich přijetí

1.3 Diagram

Přehledový grafický diagram konferencí by měl především ukazovat závislosti mezi termíny pro odeslání a termíny pro rozhodnutí o přijetí článků. Diagram by měl být dvojrozměrný, s daty daného časového rozpětí v chronologickém pořadí odshora dolů na svislé ose. Jednotlivé konference budou vykresleny jako spojnice mezi jednotlivými daty tak, aby byly co nejlépe rozlišitelné. Spojnicí se rozumí obdélník bez vykreslené levé strany takový, že jeho levé dva vrcholy leží na levé hraně diagramu ve výšce odpovídající spojovaným dnům, s nenulovou šířkou. Každá spojnice bude mít navíc vlastní popisek. Konkrétní požadavky na rozlišitelnost spojnic jsou:

- Barevná odlišnost sousedících konferencí
- Co nejmenší možný počet křížení a překryvů, tzn.
 - je-li interval jedné konference celý uvnitř intervalu jiné, spojnice budou vykresleny bez křížení (spojnice konference s kratším intervalem bude mít ostře menší šířku, než konference s větším)
 - setkávají-li se spojnice v jednom dni, budou jejich části vykresleny pod sebou a to tak, aby nedocházelo k jejich křížení
 - svislé části spojnic se nikde nebudou překrývat

Kapitola 2

Technologie

Následující kapitola stručně popisuje technologie zvažované a použité při vývoji webového konferenčního kalendáře.

2.1 HTML 5 vs. HTML 4.01

Při volbě verze HTML pro tvorbu webového kalendáře byly zvažovány dvě možnosti: HTML 4.01 strict a HTML 5.

První volba zaručuje širokou kompatibilitu v drtivé většině dnes používaných prohlížečů. HTML 5 naproti tomu nabízí nové možnosti, které se pro vývoj webového kalendáře jeví jako velmi užitečné, především jde o podporu SVG, kterou lze velmi výhodně využít při generování přehledového diagramu.[1] Při použití starší verze HTML by bylo třeba generovat diagram jako rastrovou grafiku, což přináší značné nevýhody, viz. níže. Další možností, kterou HTML 5 nabízí je použití standardů CSS 3 pro stylování dokumentu. Tento argument však při konečném rozhodování nebyl nejdůležitější, neboť CSS 3 bylo potřeba pouze na jediném místě, kde by bylo snadno nahraditelné starší normou.

Podpora HTML 5 v prohlížečích v době publikování práce ještě není úplná, předně proto, že ani kompletní specifikace HTML 5 ještě není uzavřená. Nicméně základní novinky HTML 5, které se nabízely k využití (především zmíněné SVG) jsou už nyní podporované nejnovějšími verzemi nejrozšířenějších prohlížečů.[2] S přihlédnutím k tomu, že v dané oblasti probíhá masivní vývoj a existuje reálný předpoklad, že v blízké době dojde k rozšíření HTML 5 do prohlížečů většiny uživatelů, přiklonil jsem se nakonec k modernější variantě.

2.2 Rastrová vs. vektorová grafika

Přehledový diagram je možné generovat buď rastrově nebo vektorově. Zde však kromě výše popsaných problémů s podporou prohlížečů, které se ukázaly jako nepříliš relevantní, hrají všechny karty do rukou vektorové grafiky. Z těch největších výhod uveďme alespoň následující:

- Minimalizace datového přenosu
- Možnost vytvářet hypertextové odkazy přímo na jednotlivých elementech grafiky
- Škálovatelnost

2.3 Programovací jazyk na straně serveru

Pro naprogramování aplikace na straně serveru byl zvolen jazyk PHP především ze dvou důvodů. Prvním je masivní podpora tohoto skriptovacího jazyka u poskytovatelů hostingových služeb, na rozdíl od jiných jazyků (ASP, Ruby, Java), případně jednoduchost instalace na vlastním serveru. Druhým důvodem je osobní praxe autora s tímto jazykem.

2.4 Frameworky

2.4.1 PHP framework

Jedním z logických kroků při programování webového projektu v PHP je jistě použití patřičného frameworku. Programátorovi takový framework usnadní práci, neboť nemusí programovat to, co již bylo mnohokrát naprogramováno, zajistí vyšší bezpečnost, neboť integritu frameworku zpravidla hlídá početná komunita a vývoj trvá již delší dobu, a zároveň jej udrží v konvencích slušného programování a přispěje ke kvalitě výsledného kódu.

PHP frameworků existuje celá řada, každý s jiným zaměřením, jiným přístupem. Do užšího výběru pro použití na projektu patří následující:

2.4.1.1 Nette

Nette je PHP framework českého původu. Používá vlastní šablonovací jazyk, staví na architektuře MVP (Model-View-Presenter, viz 3.1.2). Framework je v ČR velmi

rozšířený a používaný, nicméně jeho naučení je bez předchozího absolvování školení vcelku náročné.[3]

2.4.1.2 Zend Framework

Zend framework je pravděpodobně světově nejrozšířenějším PHP frameworkem vůbec. Disponuje obrovskou základnou ve vlastní komunitě, což mu dodává spolehlivost a dobrou použitelnost. Pro projekt webového kalendáře je však Zend Framework pomyslný „kanón na komára,” zvláště bez jeho předchozí znalosti.[4]

2.4.1.3 CodeIgniter

CodeIgniter je lightweight PHP framework, který nabízí vlastní jednoduchý šablonovací jazyk, k jehož využívání však nenutí. Díky jeho lightweight provedení je rychlý, snadno instalovatelný, přesto však příjemně použitelný. CodeIgniter je ideální pro uživatele, který se nechce učit komplexní framework a šablonovací jazyk pro nenáročný projekt. Tento framework zároveň podporuje architekturu MVC (Model-View-Controller, viz 3.1.1). Právě pro tyto vlastnosti byl nakonec zvolen pro webový konferenční kalendář.[5]

2.4.2 Javascript framework

Při výběru javascriptového frameworku byl zvažován Dojo Toolkit[6], ovšem pro jeho rozsáhlost a předchozí neznalost nakonec nebyl použit. Použitými javascriptovými frameworky tak zůstaly dnes všeobecně rozšířené a populární

- jQuery[7]
- jQuery UI[8]

Kapitola 3

Implementace

Následující kapitola představuje implementační podrobnosti projektu, především zvolené metodiky a další detaily.

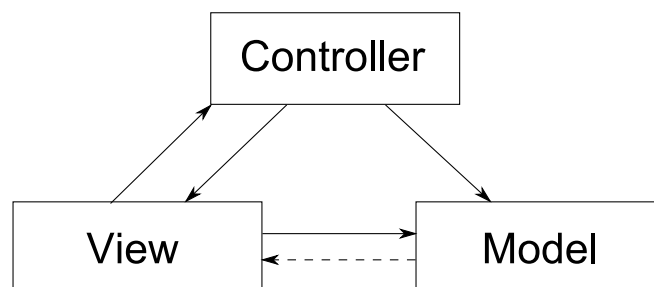
3.1 Architektura

3.1.1 MVC

Architektura MVC, tedy Model-View-Controller, je technika vývoje (nejen) webových prezentací. Její základní myšlenkou je snaha o oddělení uživatelského rozhraní a aplikační logiky. Modely tak představují nástroje (zpravidla třídy) pro manipulaci s daty (soubory, databázové systémy apod.), zatímco views jsou určené pro prezentaci a komunikaci s uživatelem a nabízí jednotlivé pohledy (stránky, widgety, komponenty) na vlastní datový objekt. Controller je pak hlavní část, která zprostředkovává interakci mezi těmito dvěma.[9] V praxi se právě v controlleru odehrává nejvíce práce, view pak bývá jen jakousi šablonou pro vykreslení dat (ačkoli i zde může dojít na pokročilé renderování), model nabízí základní operace pro práci s daty. Tato architektura tak při správném použití zajišťuje dobrou modularitu a rozšiřitelnost projektu a zároveň přirozeně dělí úkoly mezi jednotlivé specialisty (databázový specialista, programátor, grafik). Jednoduchost idey architektury i dobrá rozšiřitelnost výsledného projektu byly důvodem, proč jsem zvolil právě MVC pro webový konferenční kalendář.

3.1.2 MVP

Architektura MVP, tedy Model-View-Presenter je v podstatě variantou MVC a také se jí hodně podobá. Společným rysem je rozdělení do třech částí a také přítomnost



Obrázek 3.1: Model-View-Controller

modelu a view. Rozdíl je zde ovšem v chápání úlohy presenteru (controlleru v MVC). Zatímco v MVC je za příjem a zpracování vstupů zodpovědný controller, v MVP jsou vstupy od uživatele zpracovávány přímo ve view, zatímco presenter nabízí v podstatě aplikační vrstvu pro práci s modelem. Existuje více variant MVP, které se od sebe vcelku výrazně liší. Na variantě MVP je postaven také český Nette framework pro vývoj PHP aplikací.[9]

3.2 Rozvržení aplikace

3.2.1 Modely

Celou aplikaci tvoří pouze dva modely. Model User a model Calendar. Modely zároveň odpovídají databázovým tabulkám a nabízejí základní operace pro práci nad nimi. Jak název napovídá, v případě Modelu se tak jedná o správu uživatelů, jejich přidávání, ověřování apod. Calendar se naproti tomu stará o správu veškerých konferencí, přidávání, úpravu, mazání apod. Také nabízí metody pro získávání dat několika různými způsoby.

3.2.2 Controllery

Shodně s modely jsou v projektu dva odpovídající controllery, které v příslušném z modelů manipulují s daty. Controller User se stará o registraci, přihlašování a odhlašování uživatelů, Calendar pak obsluhuje model Calendar, slouží tak pro veškerou manipulaci s konferencemi a dotazy nad modelem.

3.2.3 Views

Jednotlivých views je pak mnoho, typicky pro každou uživatelskou akci jeden, navíc některé další pro chybové nebo potvrzující hlášení apod. Views se starají o prezentaci dat pro uživatele. Nejrozsáhlejším a nejdůležitějším view v celém projektu je `calendar_diagram`, který se stará o rendering přehledového diagramu konferencí v SVG.

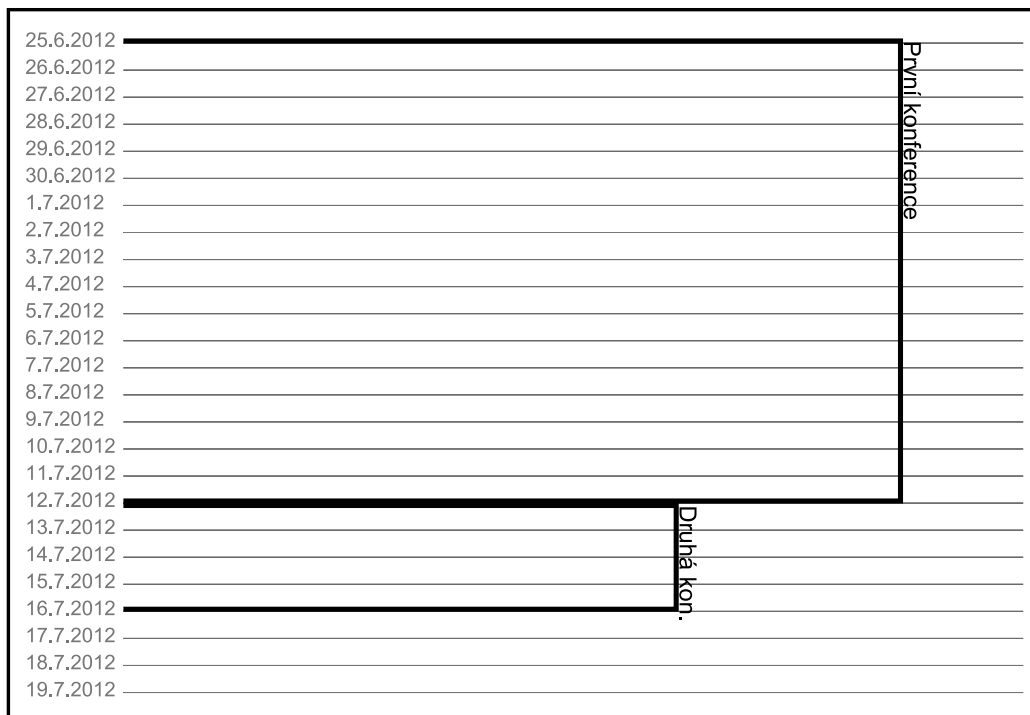
Diagramové view dostane od controlleru, potažmo modelu, data o jednotlivých konferencích připravená v poli, které umožní lineární průchod v jednoduchém `foreach` cyklu. Díky uspořádání dat v poli lze snadno dosáhnout všech požadavků na vykreslování diagramu (viz výše).

3.2.4 Vykreslování diagramu

Pro vykreslování diagramu jsem se rozhodl použít SVG, z výše uvedených důvodů. Problém tak spočíval pouze v sestavení kódu na bázi XML, tedy elementů odpovídajících v podstatě jen čtvercům, čarám a krátkým textům. Pro splnění všech požadavků na diagram (viz 1.3) a zároveň jednoduché a rychlé vykreslení jsem navrhl vlastní strukturu dat, předávanou modelem. Předává se tak pole, jehož indexy odpovídají všem dnům, kde některá spojnice buď končí, nebo začíná, tedy všem dnům, na které připadá termín odeslání článků nebo rozhodnutí o jejich přijetí. Hodnoty pole jsou pak opět poli, znovu indexovanými dny, tentokrát dny, se kterými je daný den spojen. Je-li v tento den termín odeslání článků některé konference, pak je v poli její den rozhodnutí o přijetí a vice versa. Jednotlivé dny jsou navíc uspořádány tak, aby jejich pořadí odpovídalo pořadí spojnic odshora dolů v daném dni. Uveďme příklad pole a vykresleného obrázku:

Algoritmus 3.1 Datové pole

```
$data = array(  
    '25.6.2012' => array(  
        '12.7.2012' => array('id' => 1, 'name' => 'První konference')  
    ),  
    '12.7.2012' => array(  
        '25.6.2012' => array('id' => 1, 'name' => 'První konference'),  
        '16.7.2012' => array('id' => 2, 'name' => 'Druhá kon.')    ),  
    '16.7.2012' => array(  
        '12.7.2012' => array('id' => 2, 'name' => 'Druhá kon.')    )  
)
```



Obrázek 3.2: Vykreslený diagram

Kapitola 4

Známé problémy a náměty pro další rozšíření

Při práci na projektu vyvstalo mnoho námětů pro další rozšíření aplikace nad rámec specifikace, které nebyly implementovány z různých důvodů. Ty nejzajímavější uvádím pro inspiraci případným zájemcům o modifikaci aplikace o další užitečné části. V úvodu kapitoly také uvádím známé nedostatky práce.

4.1 Známé problémy

Při zpětné analýze projektu se ukázalo, že jsem se dopustil některých návrhových nebo implementačních chyb, jejichž krátký přehled uvádím.

4.1.1 Podpora pro více jazyků

Při návrhu aplikace nebyla brána v potaz možnost rozšíření aplikace o další jazyky, její implementace tomu také odpovídá. Veškeré texty, které se v aplikaci objevují, jsou většinou vepsány přímo v šablonách jednotlivých view, v případě chybových hlášek občas v controllerech. To ovšem neumožňuje jazykovou modularitu, zaručenou extrakcí řetězců do zvláštních jazykových souborů. To je škoda, protože by to případnému zájemci o překlad usnadnilo práci a navíc framework CodeIgniter pro takovouto možnost nabízí nástroje.

4.2 Správa uživatelů

Jedním ze směrů, kde je velký potenciál v případném rozšiřování aplikace, je správa uživatelů. Uvedme několik příkladů:

4.2.1 Systém práv

Současný uživatelský systém tak, jak je navržen, zavádí pouze dva typy uživatelských rolí - administrátora a běžného uživatele, jejichž práva jsou staticky daná a neměnná. Bylo by dobré vypracovat důmyslnější systém práv na bázi relační databáze, kdy by každý uživatel mohl mít unikátní práva k jednotlivým konferencím a to jak pro čtení, tak pro přidávání/úpravu. V takovém případě by navíc bylo třeba vytvořit rozhraní pro administraci jednotlivých uživatelských kont. V přímé souvislosti se systémem práv je pak další část:

4.2.2 Skupiny uživatelů

Sdružování uživatelů do skupin by mohlo přinést zajímavé možnosti v otázce přidělování práv. Bylo by například možné oddělit jednotlivé vědní obory tak, aby mohlo aplikaci využívat více uživatelů a zároveň se vědeckým pracovníkům v oblasti např. neurologie nezobrazovaly konference týkající se jaderné fyziky. Rovněž by tak bylo možné určit správce pro jednotlivé skupiny, případně vytvořit stromovou strukturu a obecně tak delegovat správu jednotlivých oblastí.

4.3 Prohlížení konferencí

Prohlížení konferencí uživatelem je další část s potenciálem pro nabídnutí většího uživatelského komfortu.

4.3.1 Další možnosti zobrazení

Vzhledem k tomu, že autor sám nepřihlašuje své články na vědecké konference a není tak potenciálním přímým uživatelem aplikace, ani nemá s danou problematikou praxi, je zřejmé, že nabízené možnosti prohlížení konferencí (měsíční kalendář a diagram) nemusí být optimálním zobrazením pro přehlednou orientaci. Předpokládá se proto, že vlastní uživatelé navrhnou jiná zobrazení, pro jejichž doplňující implementaci je v aplikaci přirozený prostor.

4.3.2 Osobní správa vlastních konferencí

Další funkcí, kterou by potenciální uživatel aplikace pravděpodobně rád využil je osobní správa konferencí, o které se zajímá. Například evidence těch, na které se přihlásil či těch, kde uvažuje nad účastí. Také by si mohl uživatel sám zvolit, které konference chce, aby se mu zobrazovaly v jednotlivých přehledech.

4.3.3 Export kalendářových dat

Mnoho dnešních uživatelů používá pro plánování svého času různé kalendářové aplikace, které zpravidla umožňují import dat v zavedených formátech nebo přímo připojení ke kalendářovému serveru. Podobné exportovací funkce by po prostudování jednotlivých známých formátů (např. .csv, .cal, .ics) nebylo příliš těžké naimplementovat a uživatelům by to přineslo možnost prohlížet data ve svém oblíbeném klientovi spolu s dalšími osobními či pracovními událostmi.

Kapitola 5

Technické požadavky a licence

5.1 Technické požadavky

5.1.1 Technické požadavky na straně serveru

Pro správné fungování aplikace je třeba zajistit na serveru následující požadavky:

- PHP verze 5.1.6 nebo novější
- Připojení k databázi, podporované jsou: MySQL 4.1 a novější, MySQLi, MS SQL, Postgres, Oracle, SQLite a ODBC.
 - Aplikace byla testována a vyvíjena na databázi MySQL[10]
 - Pro daný typ databáze je potřeba mít nainstalováno odpovídající rozšíření PHP[11]

5.1.2 Technické požadavky na straně klienta

Pro správné zobrazování a fungování aplikace na straně klienta je potřeba internetový prohlížeč, který podporuje standard HTML5, především pak dokáže vykreslovat grafiku SVG. Testovány byly tyto prohlížeče:

- Google Chrome 20
- Mozilla Firefox 14
- Opera 12
- Internet Explorer 10

Testovány byly i některé verze novějších mobilních prohlížečů, většinou s úspěchem. Na dotykových zařízeních může být problém s prokliknutím odkazu umístěného na spojnici v diagramu, vzhledem k jeho tenké šířce.

5.2 Licenční ujednání

Projekt používá několik různých technologií publikovaných pod různými licencemi.

- PHP používá licenci PHP License v3.01[12]
- CodeIgniter používá vlastní licenci[13]
- jQuery dovoluje používat licenci MIT[15] nebo GPL v2[16][14]

Projekt lze provozovat/rozšiřovat pod libovolnou licenci, která je kompatibilní s těmito.

Závěr

Výsledkem práce je funkční webová aplikace umožňující snadnou tvorbu přehledu nadcházejících konferencí. Aplikace nabízí všechny základní prostředky pro řešení daného problému a otevírá řadu možností pro další rozvoj. Vzhledem k velkému množství variant rozšíření a omezenému času však projekt nabízí pouze nejdůležitější funkcionalitu s předpokladem, že různí uživatelé mají různé doplňující požadavky, a spíše než nabídnout monolitickou všeobjímající aplikaci se snaží poskytnout jednoduché jádro, které může být snadno upraveno a doplněno dle uživatelských nároků. Více než přímý finální produkt, ačkoliv je samostatně funkční tak, jak je, chce tento webový konferenční kalendář sloužit jako základ pro další kustomizaci a rozšíření dle aktuálních požadavků uživatelů.

Literatura

- [1] HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C. W3C [online]. Editor's Draft, revision 1.5697. 26 June 2012 [cit. 2012-07-26]. Dostupné z: <http://dev.w3.org/html5/spec/single-page.html>
- [2] SIGHTS. The HTML5 test: HOW WELL DOES YOUR BROWSER SUPPORT HTML5? [online]. version 3.0. April, 2012 [cit. 2012-07-26]. Dostupné z: <http://html5test.com>
- [3] NETTE FOUNDATION. Nette Framework [online]. © 2008, 2012 [cit. 2012-07-26]. Dostupné z: <http://nette.org/>
- [4] ZEND TECHNOLOGIES LTD. Zend Framework [online]. © 2006 - 2012 [cit. 2012-07-26]. Dostupné z: <http://framework.zend.com/>
- [5] ELLISLAB, Inc. CodeIgniter [online]. (c) 2001- 2012 [cit. 2012-07-26]. Dostupné z: <http://codeigniter.com/>
- [6] THE DOJO FOUNDATION. Dojo [online]. [cit. 2012-07-26]. Dostupné z: <http://dojotoolkit.org/>
- [7] THE JQUERY FOUNDATION. JQuery: write less, do more. [online]. © 2012 [cit. 2012-07-26]. Dostupné z: <http://jquery.com/>
- [8] JQUERY FOUNDATION, JQUERY UI TEAM. JQuery: user interface [online]. © 2012 [cit. 2012-07-26]. Dostupné z: <http://jqueryui.com/>
- [9] Interactive Application Architecture Patterns. Aspiring Craftsman [online]. August 25, 2007 [cit. 2012-07-26]. Dostupné z: <http://aspiringcraftsman.com/2007/08/25/interactive-application-architecture/>
- [10] ORACLE CORPORATION AND/OR ITS AFFILIATES. MySQL [online]. © 2012 [cit. 2012-07-26]. Dostupné z: <http://www.mysql.com/>

- [11] Database Extensions. THE PHP GROUP. Php.net [online]. © 2001-2012, Last updated: Thu Jul 26 12:41:07 2012 UTC [cit. 2012-07-26]. Dostupné z: <http://cz.php.net/manual/en/refs.database.php>
- [12] PHP Licensing. Php.net [online]. © 2001-2012, Last updated: Thu Jul 26 01:41:19 2012 UTC [cit. 2012-07-26]. Dostupné z: <http://www.php.net/license/index.php>
- [13] CodeIgniter License Agreement. ELLISLAB, Inc. CodeIgniter [online]. (c) 2008 - 2011 [cit. 2012-07-26]. Dostupné z: http://codeigniter.com/user_guide/license.html
- [14] License - jQuery Project. JQUERY FOUNDATION. JQuery project. [online]. © 2012 [cit. 2012-07-26]. Dostupné z: <http://jquery.org/license/>
- [15] Open Source Initiative OSI - The MIT License (MIT):Licensing. OPEN SOURCE INITIATIVE. Open Source Initiative [online]. [1988] [cit. 2012-07-26]. Dostupné z: <http://opensource.org/comment/935>
- [16] GNU General Public License: Všeobecná veřejná licence GNU. FREE SOFTWARE FOUNDATION, Inc. GNU Operating System [online]. (C) 1996, 1997, 1998, 1999, Updated: 2007/06/28 20:04:38 [cit. 2012-07-26]. Dostupné z: <http://www.gnu.org/licenses/gpl-2.0.html>

Seznam použitých zkratek

ASP Active Server Pages

CSS Cascading Style Sheet

HTML Hypertext Markup Language

MVC Model-View-Controller

MVP Model-View-Presenter

PHP Hypertext Preprocesor

SVG Scalable Vector Graphics

XML eXtended Markup Language

Přílohy

Na přiloženém CD jsou tyto součásti práce:

- Elektronická verze tohoto textu
- Zdrojové soubory aplikace (conference_calendar)
- Návod k instalaci