VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

INTERNETOVÝ PORTÁL VÝZKUMNÉ SKUPINY

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE DAVID BLAHETA
AUTHOR

BRNO 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

INTERNETOVÝ PORTÁL VÝZKUMNÉ SKUPINY

THE OFFICIAL WEBSITE OF RESEARCH GROUP

BAKALÁŘSKÁ PRÁCE

BACHELOR´S THESIS

AUTOR PRÁCE — DAVID BLAHETA
AUTHOR

VEDOUCÍ PRÁCE — ING. MILOŠ OHLÍDAL
SUPERVISOR

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Blaheta David**
Obor: Informační technologie
Téma: **Internetový portál výzkumné skupiny**
Kategorie: Web

Pokyny:

1. Seznamte se s principy tvorby dynamických www stránek.
2. Seznamte se detailně s programovacími technikami PHP a databází MySQL.
3. Seznamte se důkladně s požadavky kladenými na internetový. portál výzkumné skupiny. Podle výsledků analýzy navrhněte koncepci portálu.
4. Navrženou koncepci realizujte.
5. Zhodnoťte dosažené výsledky a diskutujte možné pokračování v projektu.

Literatura:

- http://cz.php.net/
- http://www.mysql.com/

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

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

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

Vedoucí: **Ohlídal Miloš, Ing.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

LICENCNI SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **David Blaheta**
Id studenta: 85930
Bytem: E.Rošického 1072, 721 00 Ostrava
Narozen: 21. 11. 1984, Moravská Ostrava
(dále jen "autor")

a

2. Vysoké učení technické v Brně

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

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

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

Název VŠKP: Internetový portál výzkumné skupiny
Vedoucí/školitel VŠKP: Ohlídal Miloš, Ing.
Ústav: Ústav počítačových systémů
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

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

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

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☒ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

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

V Brně dne: .....

.....	Blaulta .....
Nabyvatel	Autor

Abstrakt

Obsahem bakalářské práce je vytvoření portálu vědecké skupiny dle konkrétních požadavků zadavatele. Vytvořené webové stránky jsou pojaty dynamicky a umožňují uživateli značný komfort nejen v oblasti získávání informací, ale také v oblasti vlastní administrátorské práce. V práci jsou popsány informační technologie, které byly při realizaci využity, PHP, MySQL, XHTML, CSS a JavaScript. Dále jsou uvedeny funkce, kterými portál disponuje a je popsán způsob jejich implementace. Jedná se o funkce kalendář akcí, správa publikací, vytváření animací, zasílání novinek na email, spolupráce s programem *Mathcad* a další. Součástí práce je manuál pro uživatele a administrátory, který obsahuje obrázky výsledné aplikace. Jako příloha je elektronické médium se zdrojovými kódy celé aplikace.

Klíčová slova

Vědecká skupina, portál, informační systém, webová aplikace, PHP, MySQL, XHTML, CSS, JavaScript, Mathcad, databáze

Abstract

The contents of this bachelor‘s thesis consists in a development of a research group website according to the specified requirements of the proposer. The created web pages are constructed dynamically and provide the user a lot of amenities not only in the field of information acquisition but also in the field of the administrator’s work. The thesis describes information technologies which was utilized for realization, PHP, MySQL, XHTML, CSS and JavaScript. Moreover, it describes functions implemented in the portal and the way of their implementation. Namely, these functions are the following: diary of events, administration of publications, creation of animations, sending news on an e-mail address, joint work with *Mathcad* etc. A part of the thesis contains user‘s and administrator‘s manuals with pictures from the programmed application. As an enclosure, an electronic medium is provided with the source codes of the entire application.

Keywords

Scientific group, portal, information system, Web application, PHP, MySQL, XHTML, CSS, JavaScript, Mathcad, database

Citace

David Blaheta: Internetový portál výzkumné skupiny, bakalářská práce, Brno, FIT VUT v Brně, 2007

Internetový portál výzkumné skupiny

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Miloše Ohlídala. Další informace mi poskytl doc. RNDr. Miloslav Ohlídal, CSc.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
David Blaheta
15. 5. 2007

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Miloši Ohlídalovi a rovněž zadavateli, doc. RNDr. Miloslavu Ohlídalovi, CSc., za připomínky a konzultace při realizaci mé bakalářské práce.

© David Blaheta, 2007
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah 8
Úvod 9
1 Seznámení s programovými nástroji 10
1.1 Vývoj prezentace na internetu 10
1.2 Jazyk pro popis dokumentů 10
1.3 Jazyk PHP 13
1.4 Databáze MySQL 15
1.5 Jazyk JavaScript 16
2 Vlastní řešení 17
2.1 Analýza problému 17
2.2 Návrh řešení 18
2.3 Struktura databáze 20
2.4 Implementace jednotlivých částí 22
2.5 Propojení s programem *Mathcad* 29
3 Zhodnocení práce 33
3.1 Dosažené výsledky 33
3.2 Vlastní přínos 33
3.3 Další vývoj 34
Literatura 35
Seznam příloh 36

Úvod

Rozvoj informačních technologií v posledních letech umožňuje nejen rychlou a snadnou komunikaci, ale také dostupnost nejrůznějších informací o státech, institucích, firmách, ale také o jenotlivcich. Mít svou vlastní prezentaci na internetu je dnes již běžnou záležitostí. Vlastní Web je totiž prostor, kde jednotlivec může prezentovat své výsledky nebo informovat kohokoli na světě o tom, čím se zabývá. Proto je zájem o umístěni informačních, publikačních nebo třeba jen zábavných webových stránek na celosvětovou síť dnes velmi žádaný.

Způsob prezentace na internetu se poměrně často měnil a mění se stále. Je to způsobeno hlavně tím, že Web prošel za svou krátkou historii už velkým množstvím změn a často se mění také trendy, jak se na internetu prezentovat. Původně internet sloužil jen pro vědecké účely a informace byly statické, uživatel tedy nemohl obsah Webu ovlivnit. To se dnes ale od základu změnilo a velké procento internetových prezentací se dnes snaží o nějakou interakci s uživatelem.

Vytvoření interaktivní prezentace je cílem mé bakalářské práce. Jedná se o vytvoření portálu vědecké skupiny, která musí přesně splňovat požadavky zadavatele. Dnes je mnoho prostředků, které se dají k vytvoření takového portálu použít. V mé bakalářské práci jsem použil jazyk PHP a databázový systém MySQL, které se k tomuto účelu výborně hodí.

Nástroje, kterými byla bakalářská práce vytvořena jsou popsány v první kapitole, která se snaží uvést základní poznatky a také přiblížit vývoj, kterým prošly. Druhá kapitola se zabývá vlastním řešením práce. Jsou v ní popsány funkce programu a popis implementace jednotlivých částí. Kapitola tři poté obsahuje plány do budoucna a možnosti dalšího rozšíření. Přílohou práce je uživatelský manuál.

1 Seznámení s programovými nástroji

Prezentace, kterou jsem se snažil vytvořit musí být dynamická. Z tohoto požadavku vyplývá, že na její vytvoření bylo třeba použít několik nástrojů.

První kapitola je věnována vývoji, kterým internetové prezentace prošly od svého vzniku. Dále obsahuje popis jednotlivých nástrojů, pomocí kterých byl Web realizován.

1.1 Vývoj prezentace na internetu

Začátek internetových prezentací můžeme datovat od roku 1990. Tento rok byla spuštěna služba World-Wide Web ve výzkumném centru ve švýcarském CERNu. Služba WWW se asi nejvíce zasloužila o rozvoj internetu a proto v devadesátých letech minulého století přestal být Internet doménou vědeckých kruhů, ale rozšířil se dále mezi populaci.

V době nástupu WWW byly používány jen 3 technologie, které zajišťovaly prezentaci na Internetu. Byly to přenosový protokol HTTP (*Hyper Text Transfer Protocol*). Adresace pomocí URL (*Uniform Ressource Locator*) a nakonec technologie HTML (*HyperText Markup Language*), který sloužil a stále slouží k zápisu webových stránek. Z těchto tří technologií se budu zabývat jen HTML, protože stále patří mezi hlavní prostředky pro tvorbu webových prezentací [1].

V následujících letech se internet mohutně rozvinul a dnes si bez něj často nedovedeme představit jediný den. Jak se internet a služba WWW rychle rozvíjela, tak se také velmi rychle vyvíjel způsob prezentování na internetu. Z původně statických dokumentů se hlavně díky vstupu komerční sféry začaly stávat dokumenty dynamické. Ty se snaží o interakci s uživatelem a dynamickou změnu dat podle určitých událostí. Tyto technologie už ale nejsou dost dobře realizovatelné jen za pomocí HTML, a do hry vstoupilo daleko více nástrojů pro vývoj Webu. V této kapitole představuji skriptovací jazyk PHP a JavaScript, databázi MySQL a jazyky pro popis dokumentů HTML a CSS.

1.2 Jazyk pro popis dokumentů

1.2.1 HTML

HyperText Markup Language (dále jen HTML) vznikl roku 1990 na půdě výzkumného centra CERN. Za jeho vznikem stáli především **Tim Berners-Lee** a **Robert Caillau**. Jazyk byl původně vytvořen pro popis dokumentů, v té době převážně vědeckých. Byl navržen tak, aby byl jednodušší než TeX, *postscript* a jiné jazyky, které se v té době hojně využívaly. Ve stejné době a na stejném místě také vznikl protokol, který umožňoval jeho přenos v síti a to HTTP. V roce 1993 po vzniku prohlížeče *Mosaic*, což byl první prohlížeč HTML s grafickým uživatelským rozhranním, začala popularita

HTML rychle růst. Od té doby se jazyk HTML stále vyvíjí a nabízí stále nové možnosti, čímž se přizpůsobuje potřebám dnešního moderního publikování na internetu [12].

Jazyk HTML je značkovací jazyk, to znamená, že jeho zdrojový kód obsahuje zároveň jak vlastní text, tak instrukce pro jeho zpracování. Ty se zpracovávají pomocí značek (*tags*). Zdrojový text je ASCII soubor, takže je snadno editovatelný i v jednoduchých textových editorech.

HTML jak jej známe dnes, je HTML verze 2.0. Je charakterizován množinou značek, které mají definované určité atributy. Mezi tyto značky je umístěn text, který určuje význam daného dokumentu. Značky jsou většinou párové, kdy rozlišuje počáteční a konečnou značku. Ta začíná znakem lomítko. Značky jsou uzavřeny mezi úhlové závorky. Jednotlivé značky se do sebe dají vnořovat a tím vzniká struktura HTML dokumentu. Příklad dokumentu zapsaného pomocí jazyka HTML by vypadal asi takto.

```
<html>
      <body>
            <center><h1>Nadpis</h1></center>
      </body>
</html>
```

Značky HTML dokumentu můžeme rozlišit na tři základní skupiny.

1. **Strukturální značky:** rozvrhují strukturu dokumentu(např. odstavce, nadpisy)
2. **Popisné značky:** popisují povahu elementu (např. `<title>`)
3. **Stylistické značky:** utvářejí vzhled dokumentu

Pro zobrazení HTML dokumentu slouží prohlížeč. Ten je vybaven *parserem,* který dokument rozloží na jednotlivé elementy. Každý prohlížeč obsahuje tabulku značek, které podporuje. Tuto tabulku můžeme omezit zapsáním typu HTML dokumentu na začátku zdrojového kódu. Každé značce je poté přiřazen styl . Ten definuje, jak bude text mezi dvěma značkami vypadat [2, 4].

Vývoj jazyka HTML je již nyní zastaven. Byl ukončen verzí 4.01 a dalším vývojem pro tvorbu prezentací je jazyk XHTML.

1.2.2 XHTML

Jazyk XHTML je nástupce jazyka HTML verze 4.01. Jedná se také o značkovací jazyk a vychází ze stejných standardů. XHTML konkrétně ve verzi 1.0, byl použit pro tvorbu stránek v mé bakalářské práci.

Byl vyvinut proto, aby vyhovoval podmínkám tvorby XML dokumentů a přitom, aby zůstala zachována kompatibilita s jazykem HTML. Existuje ve třech verzích s*trict*, t*ransitional* a f*rameset*.

V mé bakalářské práci byla použita verze *trasitional*, která neklade tak jednoznačné podmínky pro popis dokumentu, a proto umožňuje programátorovi při tvorbě programu větší volnost.

Popisem XHTML se zde nebudu šířeji zabývat, protože platí vše, co již je uvedeno v kapitole o jazyku HTML. Jediné rozdíly jsou snad v tom, že téměř všechny značky v dokumentu jsou párové a musí být uzavřeny, a to i značky původně nepárové (např. `<br />`). Rozdíl je také v tom, že všechny značky musí být napsány malými písmeny a atributy musí být zapsány v uvozovkách.

Aktuální verzí XHTML je verze 1.1, ale tento jazyk je dále vyvíjen konsorciem W3C (http://www.w3c.org) [13].

1.2.3 CSS

CSS neboli *Cascading Style Sheet* je jazyk pro popis zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Jazyk CSS je také použit v mé bakalářské práci a s jeho pomocí jsou definovány skoro všechny prvky vzhledu internetové prezentace.

Tento jazyk byl taktéž navržen konsorciem W3Ca je nyní ve verzích CSS1, CSS2. Pracuje se na verzi CSS3. Hlavním cílem jazyka CSS je oddělení vzhledu dokumentu od jeho obsahu a struktury. Dříve byla prezentace specifikována společně s obsahem, což mělo své nevýhody. Stránky byly ale špatně udržovatelné. Změna vzhledu Webu znamenala změnu všech dokumentů. Další nevýhodou bylo, že při změně vzhledu byla nutná také změna programu. Řešením bylo odebrání prostředků vzhledu značkám HTML a vznik jazyka CSS [15].

Vývoj jazyka CSS začíná v roce 1996 příchodem verze CSS1. V té době ale ještě nebyla úplně zajištěna podpora ze stran prohlížečů. To se ovšem změnilo s příchodem verze CSS2. Od té doby je většina nových stránek tvořena za pomoci jazyka CSS a mnoho stávajících stránek je do něj převáděno.

CSS je většinou externí dokument, ve kterém jsou definovány vlastnosti jednotlivých elementů. Jazyk CSS může být ale obsažen také přímo v hlavičce (X)HTML dokumentu.

Syntax jazyka vypadá následovně:

```
selektor { deklarace vlastností }
```

kde `selektor` určuje, na které elementy se dané pravidlo použije. Toto pravidlo se poté použije na všechny elementy v (X)HTML dokumentu. `Vlastnosti` určují jednotlivé vizuální vlastnosti daného elementu [7].

1.3 Jazyk PHP

V předchozí kapitole byly popsány dva jazyky pro tvorbu prezentací na internetu. Byl to jazyk HTML a modernější jazyk XHTML a také jazyk pro popis vzhledu CSS. V moderních internetových aplikacích si ale nevystačíme jen se statickým dokumentem. V dnešní době jsou standardem stránky, které nabízejí jistou míru interakce s uživatelem, nebo stránky, které vykonávají některé činnosti automaticky, bez vnějšího zásahu. K těmto úkonům nám ovšem jednoduchý jazyk (X)HTML nestačí a musíme použít jiný nástroj. V mé bakalářské práci je pro tvorbu dynamického obsahu použit skriptovací jazyk PHP.

1.3.1 Historie PHP

Jazyk PHP je nástupcem staršího jazyka PHP/FI, který vznikl v roce 1995. Byl vytvořen Rasmusem Lerdorfem původně jako jednoduchý soubor skriptů napsaných v jazyce *Perl* pro zpracování záznamů o přístupech k jeho Webu. Jelikož neustálé spouštění interpretu jazyka *Perl* velmi zatěžovalo WWW server, byl systém přepsán do jazyka C [1].

Původně byl celý systém vytvořen jen pro Lerdorfovo použití. Systém se ale návštěvníkům Webu zalíbil a začali ho používat. Aplikace se stávala mnohem rozsáhlejší a přibyly některé funkce pro práci s databázemi a umožňovala také uživatelům vyvíjet jednoduché dynamické aplikace pro Web. Autor poté zdrojové kódy uvolnil pod názvem *Personal Home Page Tools*, později přejmenováno na *Personal Home Page Construction Kit.* Takže kdokoliv mohl do programu přidávat nové funkce a tak se program rozšiřoval a stával se stále více populární.

Další rozšíření, které Lerdorf vytvořil, byl *Form Interpreter.* Toto rozšíření obsahovalo zpřístupnění databází na Webu. Po spojení těchto dvou programů vzniklo PHP/FI 2.0, který už si získal celosvětovou proslulost. V té době se jednalo o jednoduchý programovací jazyk, který se zapisoval do HTML stránek.

Další verzí bylo PHP 3.0. Byla to první verze, která se podobá PHP jak ho známe dnes. Verze vznikla kompletním přepsáním PHP/FI 2.0, protože tento jazyk už nevyhovoval potřebám e-komerce. Hlavní výhodou PHP 3.0 byly jeho možnosti rozšíření o spolupráci s mnoha databázemi, protokoly a dalšími, které webovým programátorům usnadňovaly život. S touto verzí přišel také nový název a PHP/FI vystřídalo pouze 'PHP', což je rekurzívní akronym - PHP: Hypertext Preprocessor. PHP 3.0 bylo uvolněno v roce 1998 po devíti měsíčním veřejném testování.

Hned po uvedení se ale začalo pracovat na přepsání jádra PHP, což mělo zvýšit výkon pro složité aplikace. V polovině roku 1999 bylo jádro přepracováno a nový *Zend Engine* byl vydán. Zbývalo jen přidat některé rozšiřující funkce a v roce 2000 bylo vydáno PHP verze 4.0. K podstatně zvýšenému výkonu této verze, přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

Po dlouhém vývoji bylo poté vydáno PHP verze 5 s novým jádrem *Zend Engine 2.0* a spoustou nových funkcí. Tato verze PHP se více orientuje na objektové programování a zavádí nové moderní přístupy, jako například propojení s platformou .NET a jiné [14].

1.3.2 Popis jazyka

PHP je skriptovací programovací jazyk. Je to jazyk interpretovaný, což znamená, že se vyhodnocuje až za běhu a nepřevádí se do podoby binární spustitelné aplikace. PHP běží na straně serveru a k prezentaci ve webových aplikacích potřebuje webový server. Syntaxe jazyka kombinuje hned několik programovacích jazyků (*Perl, C, Pascal a Java*). PHP je nezávislý na platformě, skripty fungují bez úprav na mnoha různých operačních systémech.

Velkou výhodou PHP a také jeden z hlavních důvodů jeho značné oblíbenosti je jeho velký seznam všech možných funkcí. PHP obsahuje velké množství funkcí pro práci se soubory, grafikou, přístup k velkému počtu databází (např. MySQL, PostgreSQL aj.), zpracování textových dat, hojně jsou také zastoupeny funkce pro práci s různými protokoly, které se na internetu vyskytují. Ať už jsou to protokoly pro odesílání elektronické pošty (*SMTP, IMAP, POP3*), protokol http a jiné. Funkcí, které značně usnadňují práci vývojářům je v PHP celá řada . Tyto funkce jsou buď standardně obsaženy už v jádru PHP, ale mnoho z nich je volně dostupných jako jakési rozšíření ve formě dynamických knihoven (*extension*).

S verzí pět se také výrazně zlepšil postoj k objektově orientovanému programování a PHP se v tomto směru trochu přiblížilo jazyku Java.

Syntaxe jazyka PHP je velmi podobná jazyku C, ale na rozdíl od něj nabízí programátorovi výrazně větší svobodu při psaní programů. Jak vypadá syntaxe jazyka si můžeme ukázat na jednoduchém příkladě programu, který na obrazovku vypíše „bakalarska prace“.

```
<?
 echo "bakalarska prace";
?>
```

Na tomto jednoduchém příkladě je vidět, že syntaxe jazyka je opravdu jednoduchá. Z příkladu je patrné, že každý program v jazyce PHP musí být uzavřeny mezi značky `<?` a `?>`. Tím je PHP odděleno od zbytku dokumentu a můžeme ho proto jednoduše kombinovat se značkami HTML.

Mezi typické vlastnosti jazyka patří to, že je dynamicky typový. To znamená, že datový typ proměnné se určí v okamžiku přiřazení hodnoty [1].

1.4 Databáze MySQL

V předchozí kapitole byl stručně uveden jazyk PHP. Bylo tam řečeno, že jeho značná obliba pramení také z toho, že je velmi dobře uzpůsoben ke spolupráci s databázemi. Mezi podporované databázové programy patří také MySQL. Stručné seznámeni s databází MySQL bude obsahem této kapitoly. Je zaprvé proto, že MySQL je dnes jeden z nejpoužívanějších databázových systémů, který se ve webovém vývoji používá, ale hlavně proto, že za pomocí databáze MySQL je vytvořena má bakalářská práce.

1.4.1 Historie MySQL

Databáze MySQL vychází ze starší databáze *mSQL*. Byla vytvořena švédským programátorem Michaelem Wideniem pro firmu *TcX*. Začátek vývoje se datuje do roku 1994, kdy se firma *TcX* začala zabývat vývojem webových aplikací. Databáze *UNIREG*, kterou Widenius vytvořil, ale nebyla pro tyto potřeby dostačující.

Michael Widenius se tedy zkontaktoval s autorem databáze *mSQL*, která byla v té době oblíbená a navrhnul mu spojení *mSQL* a *UNIREG*u. To byl základ pro vznik databáze MySQL. Původně se mělo jednat jen o verzi 2 databáze *mSQL*, ale společnost *TcX* chtěla databázi, která by přesně vyhovovala jejich potřebám na tvorbu internetových aplikací.

Bylo vytvořeno aplikační rozhranní, které bylo velmi podobné jako u *mSQL,* ale lépe vyhovovalo potřebám společnosti. To mělo za následek, že v již existujícím kódu napsaném pro SQL stačilo udělat jen pár nepatrných změn.

V roce 1995 po tlaku obchodního partnera *TcX*, byla databáze MySQL vydána na internetu. Výsledkem tohoto vydání bylo, že se databáze MySQL stala více přístupná než *mSQL* a proto se stala více populární a známější, než její předchůdce. Poté byla databáze převedena na více platforem jako například *UNIX, WIN32* nebo *OS/2*.

Ke jménu MySQL se pojí několik dohadů. Předpona My byla ve společnosti *TcX* používána pro mnoho knihoven a aplikací, které tam vznikaly, ale My bylo také jméno dcery tvůrce databáze. Dodnes není zcela jisté, jak jméno MySQL vlastně vzniklo.

Dnes aktuální je verze 5.1 a MySQL patří mezi nejpoužívanější databázové systémy na internetu. Díky své dostupnosti, základní verze je distribuována zdarma, je populární jak pro osobní použití, ale také pro rozsáhlejší projekty [10, 11].

1.4.2 Popis MySQL

MySQL je databáze pro mnoho platforem. Komunikaci zajišťuje dotazovací jazyk SQL. SQL je standardizovaný jazyk, vytvořený pro práci s daty v relačních databázích. Jazyk SQL je v této databázi

používán ještě s několika specifickými rozšířeními. Základ ale stejně vychází ze standardního jazyka SQL.

Hlavní výhody MySQL je její značná rychlost. Už od počátku se tato databáze snažila orientovat na rychlost zpracování dat. Proto je použitelná i pro vývoj složitějších systémů. Oblibu si získala také díky jednoduché přenositelnosti přes různé platformy. Díky tomu byla její podpora integrována do jazyka PHP a společně se serverem *Apache* tvoří základ většiny dnešních webových aplikací.

Syntaxe jazyka pro komunikaci vychází ze standardního dotazovacího jazyka SQL. V následujícím příkladu je uveden jednoduchý dotaz pro vybrání všech dat z databázové tabulky.

```
SELECT * FROM nazev_tabulky
```

Tato jednoduchá ukázka slouží k přiblížení, jak jednoduchá je díky jazyku SQL komunikace s MySQL [6, 11].

1.5 Jazyk JavaScript

Poslední nástroj, který v této kapitole popíšu je JavaScript. Uvádím ho zde, protože byl použit v mé bakalářské práci, ale také proto, že rozhodně patří mezi nástroje, které se používají pro tvorbu moderních internetových prezentací.

Jazyk JavaScript je stejně jako PHP skriptovacím jazykem a z toho plynou jisté podobné rysy. Zásadně se tyto jazyky ale liší tím, kde se zpracovávají. Zatímco PHP se zpracovává na straně serveru, JavaScript se zpracovává na straně klienta. Interpretuje ho přímo webový prohlížeč. Je zřejmé, že jazyk JavaScript je odvozen od jazyka Java firmy *Sun*, ale pravda je, že tyto jazyky mají jen podobnou syntaxi. Jinak se od sebe výrazně odlišují.

JavaScript není jazyk příliš univerzální, je použitelný pouze pro vývoj webových aplikací. Zapisuje se přímo do HTML dokumentu nebo může být uložen v externím souboru. Je spustitelný pouze přes prohlížeč. Je to jazyk objektový. Využívá objektů prohlížeče a zabudovaných objektů (např. objekt `window`). Funguje pouze v prohlížeči, z čehož pramení některé nevýhody. Uživatel může spouštění JavaScriptu zakázat, ale poté některé aplikace nefungují správně. Různé prohlížeče JavaScript různě interpretují, což může vést k chybám.

I přes tyto nevýhody má ale JavaScript své místo v internetových prezentacích. Díky své jednoduchosti a pohodlnosti zápisu je velmi často využíván k vytvoření jednoduchých aplikací, které ale mohou uživateli stránek značně pomoci [3, 5].

2 Vlastní řešení

V této kapitole se budu zabývat vlastním řešením a vytvářením aplikace. Internetový portál, který jsem vytvořil má dvě základní úrovně. Uživatelská část, která je viděna běžným uživatel a administrátorská část, ke které má přístup pouze administrátor systému.

V této kapitole budou uvedeny způsoby vytvoření jednotlivých částí portálu od počáteční analýzy, až po konečnou implementaci řešení.

2.1 Analýza problému

Mým úkolem bylo vytvořit internetový portál vědecké skupiny. Tento portál byl vytvořen pro konkrétního zadavatele, kterým je doc. RNDr. Miloslav OHLÍDAL, CSc. ze strojní fakulty VUT Brno. Jedná se vytvoření portálu, kde bude uživatel seznámen se základními problémy, které se na pracovišti řeší. Obsahem musí být také prezentace výsledků a samozřejmě prezentace osoby doc. Ohlídala.

Požadavky na portál byly takové, aby každý uživatel měl jednoduchý přístup k získávání nových informací a aby se mu po zaregistrování do systému umožnilo ještě více možností. Dalším požadavkem bylo vytvoření administrátorského rozhraní, aby správce systému mohl velkou část portálu zpracovávat sám, a to bez nutnosti znát zdrojový kód nebo do něj něco doplňovat. Požadavkem bylo implementovat následující části portálu.

Kalendář akcí. Tato část má sloužit jako informování návštěvníka o připravovaných akcích nebo akcích, které již proběhly. Musí mít uživatelské a administrátorské prostředí.

Správa publikací. Jelikož je zadavatel práce vědecký pracovník, jeho seznam publikací je poměrně rozsáhlý a je třeba vytvořit rozhraní, které umožní jednoduché vkládání nových, ale také udržování informací o starých publikacích. Opět bylo požadavkem vytvořit administrátorské a uživatelské prostředí.

Zasílání novinek na email. Tato část portálu umožňuje správci systému zaslat novinku na email, z databáze přihlášených uživatelů, kteří si to přejí. Součástí musí být jak uživatelské, tak administrátorské prostředí.

Stránka pro studenty. Ta má sloužit ke komunikaci se studenty. Zadavatel práce také vyučuje na strojní fakultě a potřebuje nástroj, kterým sdělí studentům důležité informace o výuce a jiných podobných věcech spojených se studiem.

Vyhledávání musí být také součástí stránek. Jedná se o vyhledávání podle různých kritérií v nejdůležitějších částech Webu.

Animace obrázků. Tato část portálu má sloužit k prezentaci výsledků z laboratorního výzkumu. Pomocí animace se musí ukázat průběh fyzikálních procesů.

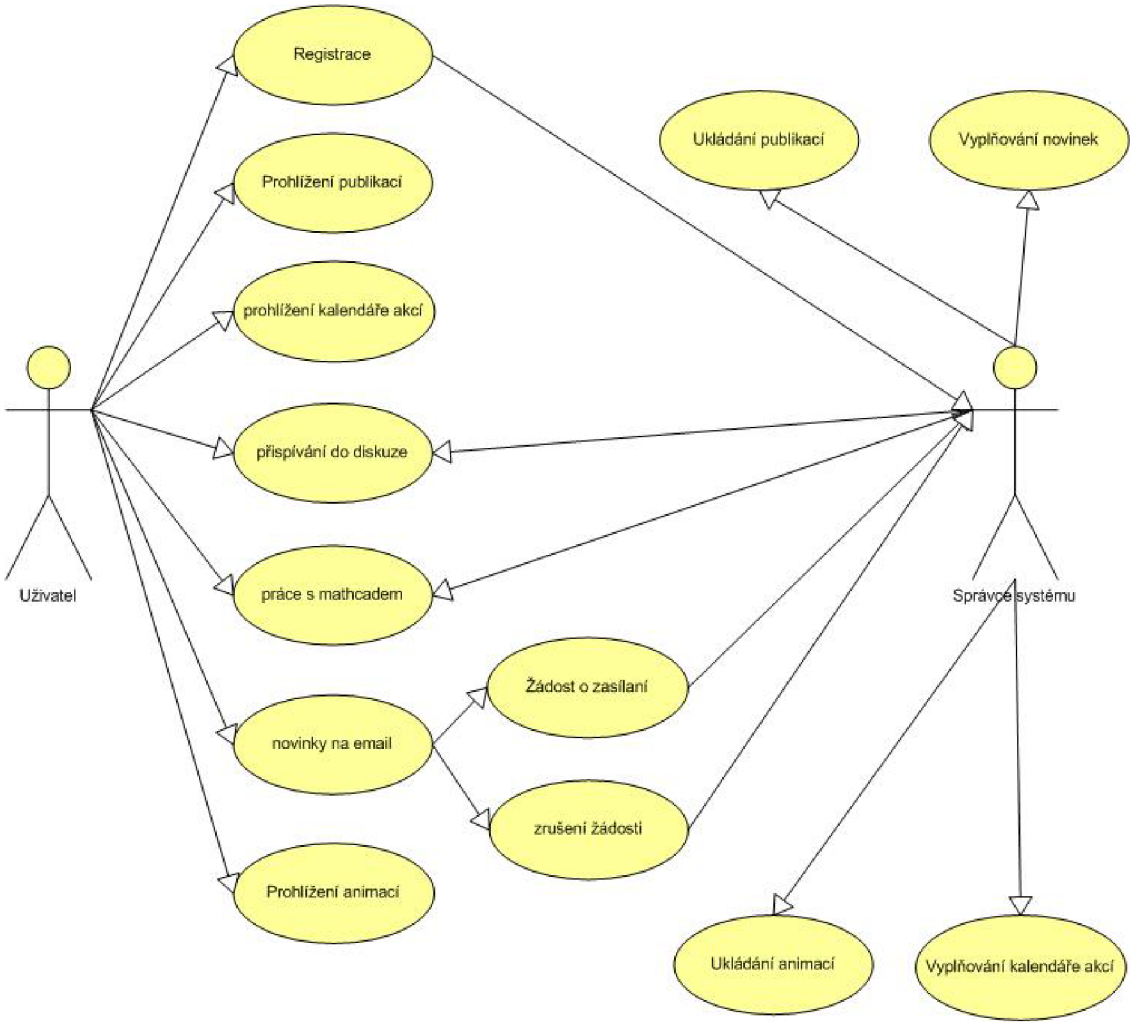
Propojení s programem *Mathcad*. Výsledky pracoviště jsou často zpracovávány pomocí programu *Mathcad*. Tato část aplikace má umožnit uživateli jednoduchou demonstraci výsledků za použití tohoto matematického programu.

Se zadavatelem práce byly upřesněny základní vlastnosti, které musí výsledný portál splňovat. Aplikace musí být napsána v jazyku PHP. Jako databáze se musí použít MySQL. To zaručuje velkou přenositelnost na jiné systémy a také značnou dostupnost, protože oba tyto nástroje jsou šířeny zdarma.

2.2 Návrh řešení

Aplikace byla rozdělena na jednotlivé části. Ty vycházejí z analýzy a se zadání od zadavatele. Tyto části byly na závěr spojeny do jednoho celku.

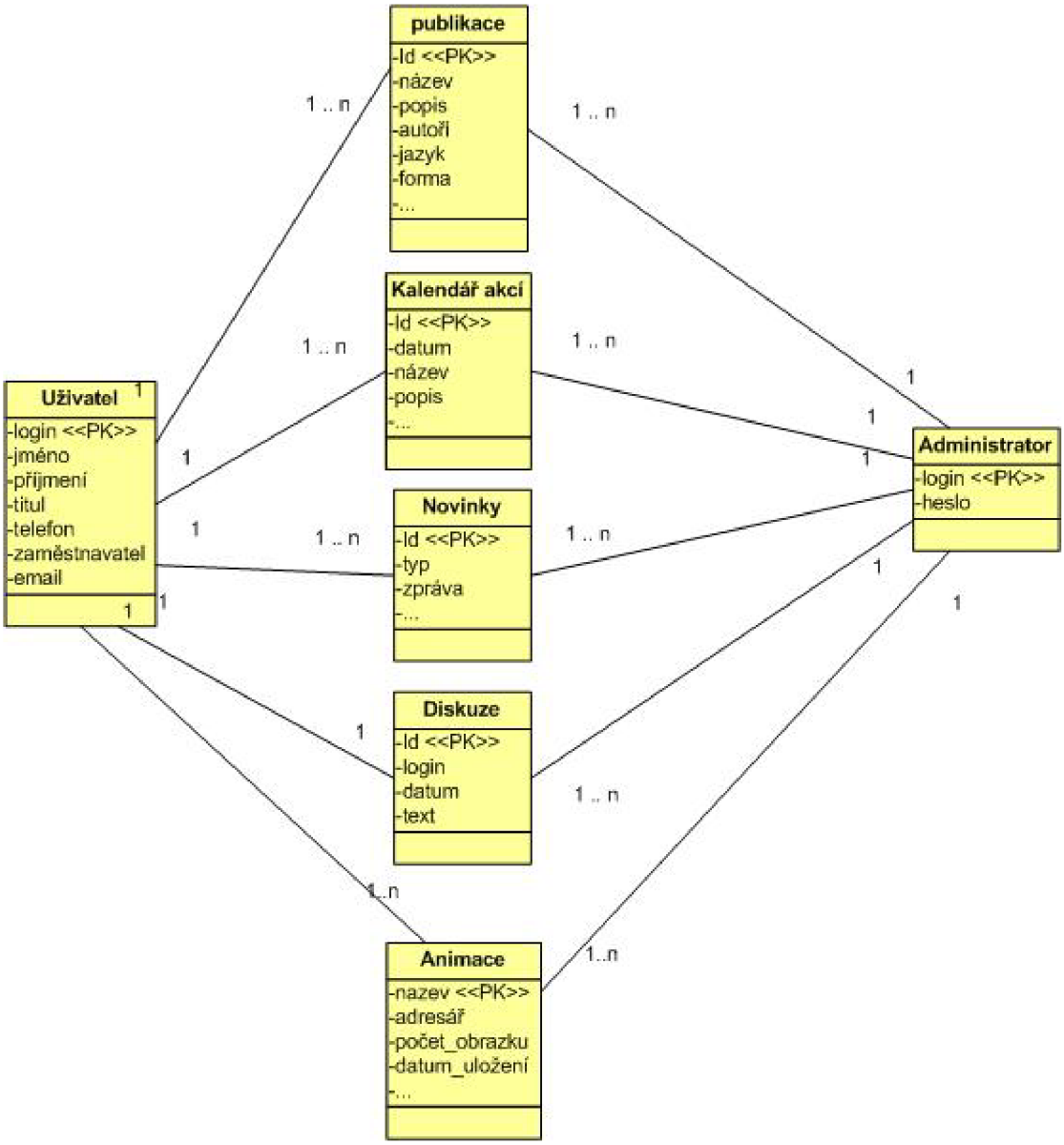
Vzhledem k tomu že aplikace bude nasazena do běžného použití byl vytvořen *use case diagram* pro základní představu o rozsahu projektu.



Obrázek 1: Use case diagram

Z *use case diagramu* je patrné rozdělení aplikaci na uživatelskou a administrátorskou část.

Pro jednotlivé součásti aplikace, jako například kalendář akcí, správa publikací a uživatelských účtů, ale i pro další části portálu, byla použita databáze. V té se uchovavávají prakticky všechny dynamicky vytvářená data. Výhodou je poté jednoduché zpracování výsledků. Možnost tvorby různých statistik, ale také poměrně jednoduchá implementace vyhledávání. Pro návrh struktury databáze byl vytvořen *er diagram.* V něm je naznačeno jak vypadá výsledná struktura databáze a databázových tabulek.



Obrázek 2: ER diagram

Pro každou část systému bylo navrženo administrátorské a uživatelské rozhraní, tak aby co nejvíce splňovalo podmínky zadání.

Kalendář akcí v administrátorském režimu byl navržen jako forma diáře, do kterého administrátor k příslušnému datu zapíše čas, název a popis akce. Pro uživatele je potom viditelný obsah kalendáře a může jím být listováno podobně jako jsme zvyklí z klasického papírového kalendáře.

Správa publikací byla vytvořena tak, aby splňovala podmínky pro dnešní popis vědeckých publikací. V administrátorském režimu se vyplní potřebné informace, které jsou uloženy do databáze. Z té se uživateli objeví citace publikace ve správném formátu včetně abstraktu nebo případně její celé znění.

Zasílání novinek na email bylo navrženo tak, aby si administrátor mohl vybrat, která novinka bude uživatelům rozeslána na email. Na výběr jsou buď všichni uživatelé nebo jen studenti. Zaregistrovanému uživateli, který si přál na email novinky zasílat, je poté zaslán email se zformátovaným textem novinky.

Stránka pro studenty je vlastně statickou stránkou obsahující informace pro studenty. Administrátor má možnost vyplnit ve svém režimu její obsah se základními možnostmi formátování.

Animace jsou navrženy tak, že administrátor vloží na server přes vytvořené rozhraní jednotlivé obrázky. Uživatel si potom po vstupu na stránky může animaci spustit s možnostmi jednoduché modifikovatelnosti rychlosti.

Propojení s programem *Mathcad.* Tato část aplikace byla navržena následovně. Administrátor vloží zdrojový soubor *mathcadu* a uloží počet a názvy proměnných, které se v něm budou modifikovat. Uživatel bude mít na stránkách připraven formulář, kde vyplní modifikovatelné proměnné. Ty se na serveru zpracují a výsledkem bude obrázek, který *Mathcad* vyrobí.

Pro zápis statických dat byl vybrán jazyk XHTML verze 1.0 ve spojení s CSS pro popis vzhledu dokumentu. Tyto dvě technologie nám zaručují, že Web bude snadno udržovatelný a díky dodržování standardů by měl být správně zobrazitelný na většině moderních prohlížečů.

2.3 Struktura databáze

Databáze hraje v mé bakalářské práci velmi důležitou roli, jelikož jsou do ní ukládána data, která administrátor uvede. Tato data jsou poté zobrazeny na stránkách běžnému uživateli. Struktura databáze vychází z ER diagramu.

Každá část Webu obsahuje vlastní databázovou tabulku, nad kterou se vykonávají jednotlivé dotazy. Struktura jednotlivých databázových tabulek je ukázána v následující tabulce.

Název tabulky	Sloupec	Typ	Nulový	Výchozí	Extra	Privátní klíč
administrator	admin_login	VARCHAR(20)	ne			ano
	admin_heslo	VARCHAR(60)	ne			
animace	nazev	VARCHAR(100)	ne			ano
	adresar	VARCHAR(20)	ne			
	pocet	INT	ne	0		
	zobrazit_animaci	ENUM('ano','ne')	ne	ano		
	datum	DATE	ne	0000-00-00		
	popis	TEXT	ano	NULL		
kalendar	id	INT	ne	0	auto_increment	ano
	datum_vlozeni	DATE	ne	0000-00-00		
	datum_udalosti	VARCHAR(20)	ano	NULL		
	nadpis	VARCHAR(160)	ano	NULL		
	popis	text	ano	NULL		
	den	INT	ne	0		
	mesic	INT	ne	0		
	rok	INT	ne	0		
	cas	VARCHAR(20)	ano	NULL		
novinky	id	INT	ne		auto_increment	ano
	kategorie	VARCHAR(255)	ne			
	novinka	text	ne			
	na_email	ENUM('ano','ne')	ne	ne		
	pro_studenty	ENUM('ano','ne')	ne	ne		
	datum	DATE	ne	0000-00-00		
	pro_reg	ENUM('ano','ne')	ne	ne		
publikace	id	INT	ne		auto_increment	ano
	nazev	VARCHAR(200)	ano	NULL		
	autori	VARCHAR(255)	ano	NULL		
	publikovano	VARCHAR(255)	ano	NULL		
	popis	text	ano	NULL		
	publikace	VARCHAR(100)	ano	NULL		
	zobrazit	ENUM('ano','ne')	ne	ne		
	forma	VARCHAR(150)	ne			
	jazyk	VARCHAR(150)	ne			
uzivatele	login	VARCHAR(20)	ne			ano
	heslo	VARCHAR(20)	ne			
	jmeno	VARCHAR(50)	ne			
	prijmeni	VARCHAR(50)	ne			
	titul	VARCHAR(15)	ano	NULL		
	telefon	VARCHAR(30)	ano	NULL		
	mobil	VARCHAR(30)	ano	NULL		
	zamestnavatel	VARCHAR(60)	ano	NULL		
	email	VARCHAR(100)	ne			
	novinky	ENUM('ano','ne')	ne			

Tabulka 1: Struktura databáze

2.4 Implementace jednotlivých částí

2.4.1 Uživatelská část

2.4.1.1 Statická prezentace

Část prezentace, která bude viditelná pro návštěvníky Webu, byla implementována pomocí jazyka XHTML 1.0. A pro vzhled dokumentu byl použit jazyk CSS.

Základní stránka se skládá z hlavičky, levé části, která obsahuje menu a kontakt, střední části, která obsahuje samotný text stránky a patičky, která obsahuje copyright. Tímto způsobem byla vytvořena základní kostra, ze které vycházejí další stránky. Tato kostra byla vytvořena pomocí stylů, které byly definovány pro element `<div>`. Styly jsou uloženy v externím souboru, což umožňuje jednoduché vkládání do všech stránek, kde je to zapotřebí.

Základem je selektor *main*, ve kterém je definována barva pozadí levé části, také okraje obsahové části Webu a pomocí vlastnosti `margin: auto` je definováno zarovnání celého Webu na střed stránky. Rozlišení webové prezentace bylo upraveno tak, že jeho šířka je 800 pixelů. Z této skutečnosti plyne, že Web by měl být dobře zobrazitelný i na monitorech s menším rozlišením.

Dalším důležitým prvkem je hlavička. Hlavička je opět uzavřena do elementu `<div>` . O její styl se stará selektor *hlavicka*, v něm je definováno pozadí, které je tvořeno obrázkem, a styly pro písmo.

Důležitou částí Webu je levá část, kde se nachází menu a kontaktní informace. Levá část je nastylována pomocí selektoru *leva*. Z tohoto selektoru poté vycházejí další, které definují odsazení a styl písma jeho barvu a další vzhledové vlastnosti.

Obsah Webu je uložen ve střední části prezentace. Text je vložen mezi elementy `<div>` které jsou nastylovány selektorem *stred*. Základním požadavkem byla dobrá čitelnost textu, proto je zvoleno bílé pozadí a černý text o velikosti 12 bodů. Pro lepší čitelnost bylo použito bezpatkové písmo *Verdana*.

Na konec Webu je umístěna patička, ve které jsou uloženy informace o copyrightu a kontakt na tvůrce Webu.

Tato statická část prezentace byla vytvořena tak, aby odpovídala dnešním standardům publikování na Webu. Snaha byla také vytvořit stránky, na které si uživatel nebude muset zvykat, ale pohyb pro něj bude již od počátku intuitivní. Z této kostry vychází celá uživatelská část Webu. Kromě úvodní stránky jsou informace do stránky vkládány pomocí PHP funkce `include()`. Díky tomu se nemusí každá stránka načítat společně s hlavičkou a dalšími částmi popsanými výše, ale načítá se jen obsah umístěn ve střední části Webu.

2.4.1.2 Registrace uživatelů

Některé funkce na Webu jsou přístupné jen uživatelům, kteří se na stránkách zaregistrovali. K registraci slouží registrační formulář, na který se návštěvník dostane po kliknutí na příslušnou položku v menu.

Registrace je tvořena šesti poli pro povinné údaje(*login, heslo, potvrzení hesla, jméno, příjmení a email)* a čtyřmi poli, které nemusí být vyplněny (*titul, telefon, mobil, zaměstnavatel*). Pro zasílání novinek slouží element typu *checkbox*, který v případě souhlasu musí být zaškrtnut.

Po vyplnění údajů je stránka odeslána ke zpracování. O zpracování se stará skript *vloz.php*. Nejprve se skript připojí k databázi MySQL a databázi pošle dotaz, kterým se vybere zadaný login. Jelikož login je privátním klíčem tabulky uživatelé, kde se budou data ukládat je třeba, aby návštěvník zadal login, který se v databázi ješte nevyskytuje. Výsledek dotazu je předán PHP funkci `mysql_num_rows()`, ta vrací počet záznamů, které vyhovují dotazu. Pokud je počet roven nule, je vše v pořádku a může se pokračovat dále. Pokud je počet větší než nula skript se ukončí s chybovým hlášením.

V další části skriptu jsou provedeny jednotlivé kontroly, zda uživatel vyplnil povinné údaje a pokud telefon nebo mobil neobsahují nepovolené znaky. Při každé chybě se generuje chybové hlášení a skript je ukončen.

Pokud se projde v pořádku přes všechny kontroly, jsou údaje uloženy v superglobálním poli `$_POST[]`, vloženy do databáze a skript přesměruje stránku na přihlášení.

2.4.1.3 Přihlášení

Přihlášení uživatelů je vytvořeno jako ověření uživatelského loginu a hesla. Uživatel je vyzván k zadání loginu a hesla a po správném vyplnění je přesměrován na první stránku po přihlášení.

Login a heslo je uloženo do pole jako `$_SESSION["login"]` a `$_SESSION["heslo"]`. Na každé stránce, kde je nutné přihlášení jsou proměnné `$_SESSION["login"]` `$_SESSION["heslo"]` kontrolovány ve skriptu *session.php*. Ten je do každé stránky chráněné heslem vložen. Údaje se vždy srovnávají s údaji z databáze. Když se údaje shodují a login i heslo jsou ověřeny, pokračuje se. Jinak je stránka přesměrována na stránku *vystraha.html*. Zároveň je při každém vstupu na zabezpečenou stránku vytvořena proměnná `$_SESSION["cas"]`, která slouží k automatickému odhlašování. Do této proměnné je vložena hodnota aktuálního času pomocí PHP funkce `time()` a je srovnána pokud je menší než 1 hodina. Po hodině nečinnosti je uživatel automaticky odhlášen.

2.4.1.4 Kalendář akcí

Uživatelská část kalendáře akcí je implementována jako listování kalendářem. Údaje jsou uloženy v databázi v tabulce *kalendar*.

Po vstupu na stránku jsou z databáze zobrazeny záznamy na aktuální měsíc. Záznamy jsou vypisovány v cyklu pomocí PHP funkce `mysql_fetch_array()` ve zformátované podobě.

Listování kalendářem probíhá kliknutím na názvy měsíců. Vždy se dá kliknou o jeden měsíc dopředu nebo dozadu. Pro tyto potřeby byla vytvořena funkce `nazevMesice()`, která má jako parametr číslo měsíce a vrací jeho název. Po kliknutí na název měsíce se vytvoří nový dotaz na databázi a jsou vybrány záznamy pod číslem vybraného měsíce. Tím se dá v databázi listovat.

Na stránce se zobrazuje vždy jen 10 záznamů na daný měsíc. To zajišťuje dotaz na databázi s limitem 10 prvků. Na další záznamy se poté odkazuje přes URL, ve kterém je specifikováno, od kterého záznamu se bude zobrazovat.

Procházet se dají také jednotlivé roky. Do připraveného formuláře se vyplní číslo roku, na který chceme přejít a po odeslání se tato hodnota uloží do proměnné `$_SESSION["rok"]`. Je to proto, že během procházení kalendáře je třeba si tuto proměnnou podržet v paměti.

2.4.1.5 Publikace

Stránka pro zobrazení publikací je v uživatelském režimu není složitá. Nejprve se provede dotaz na databázi a z tabulky publikace jsou vybrány veškeré záznamy pomocí standardní funkce `mysql_fetch_array()`. Poté jsou v cyklu procházeny jednotlivé záznamy tabulky a jejich hodnoty jsou vypisovány ve zformátované podobě. Formát se snaží dodržovat standardy pro vypsání citace publikace.

Ke každé publikaci je možno zobrazit abstrakt případně její celé znění. Celé znění je zobrazeno jen přihlášeným uživatelům, což se ověří testem, pokud je inicializovaná proměnná `$_SESSION["login"]`, která vzniká po přihlášení.

2.4.1.6 Novinky

Novinky jsou standardně vypisovány z databáze novinek. Dělí se do kategorií podle toho jak je uvedeno v databázi.

Pokud má být některá novinka zobrazena jen pro registrované uživatele, je proveden opět test na existenci proměnné `$_SESSION["login"]` a podle toho je také upraven dotaz na databázi.

2.4.1.7 Animace obrázků

Základem animace obrázků jsou funkce v jazyku *JavaScript*. Nejprve jsou do formulářového pole *select* načteny názvy všech animací, které se v databázi vyskytují. Po zvolení animace se už pracuje jen s touto animací.

Vytvoří se dotaz na databázi, který vybere veškeré záznamy o dané animaci. Potom jsou za pomoci standardních funkcí PHP pro práci s adresáři (`opendir()`, `readdir()` a `closedir()`) vybrány obrázky, které se budou animovat. Obrázky jsou předány *JavaScriptu* a jsou uloženy do pole obrázků.

Samotná animace probíhá jako záměna obrázků v jistých časových intervalech. Tuto činnost obstarává funkce `zamen()`. Záměna v časových intervalech se vykonává ve funkci `animuj()`, kdy se zaměňují postupným procházením pole jednotlivé obrázky. Časový interval je určen implicitně na 1 vteřinu pomocí JavaScript funkce `window.setTimeout('animuj()',pauza)`. Jak je vidět, rekurzivně se stále volá funkce `animuj()` v časovém intervalu určeném proměnnou `pauza`.

Rychlost animace má uživatel možnost ovlivnit pomocí JavaScript funkcí `zrychli()`, `zpomal()`. Tyto funkce zkracují nebo prodlužují intervaly mezi střídáním obrázků tak, že se dělí nebo násobí proměnná pauza. Nejpomalejší animace je jeden obrázek za vteřinu nejrychleji potom animace může běžet 64 obrázků za vteřinu.

2.4.1.8 Vyhledávání

Vyhledávat se dá v publikacích, kalendáři akcí, animacích a novinkách. Základem vyhledávání je funkce `hledani()`, která má jako parametry výraz, který se bude vyhledávat a v jaké části databáze se bude vyhledávat.

V této funkci se podle vstupních parametrů upraví SQL dotaz na databázi. Pomocí tohoto dotazu je poté v jednotlivých databázových tabulkách vyhledáváno. Je možné vyhledávat buď v jedné a nebo ve více tabulkách. Při vyhledávání nezáleží na velikosti písmen v dotazu.

Výsledky hledání jsou zobrazovány ve zformátované podobě HTML tabulek. Kam jsou z databáze vypisovány v cyklu pomocí funkce `mysql_fetch_array()`.

2.4.1.9 Spolupráce s programem Mathcad

Tato problematika bude popsána dále ve zvláštní kapitole.

2.4.2 Administrátorská část

Základní kostra administrátorské části Webu je jednoduchá a snaží se hlavně o účelnost a rychlé načítání. Je tvořena hlavičkou a střední částí. Hlavní nabídka je umístěna pod hlavičkou, kde se nachází veškerá navigace na Webu.

Kostra administrátorské části je také napsána v jazyku XHTML 1.0 a vzhled je vytvořen pomocí stylů CSS, které jsou uloženy v externím souboru *style.css*.

K přístupu do administrátorské části slouží přihlašovací formulář. Administrátor musí zadat login a heslo a po ověření, jestli se tyto údaje nacházejí v databázi je umožněno pokračovat dále. O ověření se stará PHP skript *session.php*, který je prakticky shodný jako v uživatelské části. Teprve po ověření je administrátor přesměrován na stránky, kde má možnost ke všem možnostem.

2.4.2.1 Kalendář akcí

Základní činností kalendáře akcí je v administrátorském režimu uložení záznamu do databáze. Pro administrátora je připraveno jednoduché a intuitivní rozhraní.

Základem je HTML tabulka, ve které je umístěn aktuální kalendář. O vykreslení kalendáře se starají čtyři PHP funkce.

První je funkce, která zjišťuje zda-li zadaný rok je rokem přechodným. Je to funkce `JePrechodnyRok()` a jako parametr je předáváno číslo roku. Pokud je rok přechodý, vrací funkce `TRUE` v opačném případě vrací `FALSE`. Zjištění přechodného roku v těle funkce zajišťuje standardní PHP funkce `Date()`. Ta s parametrem "L" vrací, jestli je rok přestupný.

Další funkcí potřebnou pro vytvoření kalendáře je funkce `PocetDnu()`. Tato funkce má dva parametry. Prvním je číslo měsíce v roce a druhým je rok. Poté se opět pomocí standardní funkce `Date()` spočítá počet dní v daném měsíci.

K sestavení kalendáře je zapotřebí ještě zjistit první den v měsíci. Tento údaj nám poskytne vytvořená funkce `PrvniDen()`, jelikož ale PHP pracuje s anglickým formátem času a data, je třeba výsledek, funkce `Date()` s parametrem "w", upravit. Pro neděli totiž funkce vrací číslo 0. Vše je ošetřeno jednoduchou podmínkou. Pokud je číslo dne 0 vrací se číslo 7, což odpovídá našemu pořadí.

Tím jsou připraveny všechny potřebné funkce a kalendář může být sestaven. O sestavení kalendáře do podoby HTML tabulky se stará funkce `Kalendar()` s parametry číslo měsíce a roku. Na začátku funkce jsou provedeny kontroly, jestli vstupní parametry jsou čísla a je vypočten počet sloupců, které bude kalendář obsahovat. Kalendář je sestaven ve dvou *for* cyklech, kdy jeden jde po řádcích, kterých bude 7 a druhý po sloupcích. V tomto cyklu jsou také provedeny dotazy na databázi, protože pokud už na určitý den máme v databázi akci, je buňka tabulky podbarvena.

Na dny v kalendáři se dá kliknout a poté se zobrazí nabídka vkládání do kalendáře akcí. Vkládá se vždy na daný den. Vkládání zajišťuje skript *vloz.php*, jeho funkcí jsou kontroly vkládaných údaju a poté samotné vložení do databáze pomocí SQL dotazu INSERT.

Záznamy může uživatel prohlížet. Prohlížení záznamu je uloženo ve skriptu *prohlidni.php*. V tomto skriptu jsou z databáze vybrány všechny záznamy a jsou zobrazeny v tabulce. V tabulce je zobrazeno vždy 12 záznamů. To je provedeno pomocí databázového dotazu s limitem 12 záznamů. Záznamy se v tabulce dají řadit podle různých kritérií to umožňuje změna SQL dotazu. K dotazu je přidáno slovo DESC pro změnu řazení záznamů.

V tomto skriptu lze také záznamy editovat a mazat. Editace záznamu probíhá po kliknutí na ikonu pro editaci. Skriptu je v URL předána hodnota id záznamu který se má editovat. Tento záznam je vybráz z databáze a jeho prvky jsou zobrazeny do formulářových polí. Samotná změna záznamu je prováděna opět ve skriptu *vloz.php*, kde je po úspěšné kontrole údajů záznam změněn. Mazání záznamu je prováděno ve skriptu *delete.php*. Po kliknutí na ikonu mazání je administrátor pomocí *JavaScript* funkce `confirm()` dotázán, zda-li si přeje záznam smazat. Pokud je mazání potvrzeno je

zavolán skript *dalete.php* a v URL je uvedena hodnota *id* záznamu, který se bude mazat. Ve skriptu je záznam vymazán z databáze pomocí SQL dotazu DELETE.

2.4.2.2 Správa publikací

V této části může administrátor do systému vkládat, mazat a prohlížet publikace.

Vkládání publikace je vytvořeno pomocí HTML formuláře. Do něj se zadávají jednotlivé údaje o publikaci. Nejzajímavější je vkládání do pole pro abstrakt. Je to vlastně `TEXTAREA`, která ovšem obsahuje základní prvky pro formátování textu. Text je formátován pomocí pseudo HTML značek. Ty byly inspirovány značkami jaké se používají v *miniBB*. Jsou to značky `[b][/]` pro začátek a konec psaní tučného písma, `[i][/i]` pro začátek a konec psaní kurzívou, `[url='odkaz'][/url]` pro začátek a konec odkazu a `[big][/big]` pro začátek a konec psaní větším písmem. Tyto značky jsou do pole pro text vkládány vždy za kurzor a stará se o to JavaScript funkce `vlozText()`, která je uložena v externím souboru *java.js*. Tato funkce má dva parametry, a to jméno formuláře kam se bude text vkládat, a dále text, který se bude vkládat.

Ostatní informace o publikaci jsou vkládány do standardních formulářových polí. Administrátor může také uložit plné znění publikace v externím souboru, který se vkládá přes formulářový prvek `<input type="file">`.

Po odeslání formuláře jsou data předány skriptu *upload.php*. V něm jsou provedeny základní kontroly vložených dat. Pokud jsou všechny kontroly provedeny, nahradí se PHP funkcí `str_replace()` pseudo HTML kód abstraktu na HTML značky. Poté je záznam o publikaci uložen do databáze. Pokud byl vložen soubor s publikací, je tento uložen pomocí funkce `move_uploaded_file()` do uživatelské části Webu do adresáře publikace.

Také část správa publikací obsahuje prohlížení všech publikací, jejich editaci a vkládání, ale princip je shodný jako u prohlížení kalendáře, a proto není třeba jej zde znovu uvádět.

2.4.2.3 Správa novinek

Administrátorský režim umožňuje pohodlné vkládání novinek do databáze. V této části Webu je také možno novinky mazat, popřípadě je editovat a prohlížet.

Novinky se vkládají pomocí HTML formulářů. Stejně jako v části o publikacích je umožněno vkládat novinky s jednoduchými možnostmi formátování. Dále je na výběr pomocí zatrhávacích polí *checkbox* určit, jestli bude novinka zaslána na email a vybrat skupinu, komu bude zaslána.

O vkládání novinek do databáze se stará PHP skript *vloz.php*. Zde jsou provedeny kontroly vyplněných údajů a také převedení pseudo HTML značek na standardní značky HTML. Následuje vložení do databáze pomocí SQL dotazu INSERT.

Ve skriptu *vloz.php* se také novinky odesílají na emaily těm uživatelům, kteří si to přáli při registraci. Emaily jsou odesílány pomocí PHP funkce `mail()`. V textu mailu je úvodní hlavička a poté text novinky. Uživatelé, kterým bude email poslán, se vyberou z databáze a v cyklu se uloží jejich

emailové adresy do pole. Prvky pole s emaily je rozděleno pomocí funkce `implode()` na jednotlivé adresy oddělené čárkou. Takto uložené adresy jsou předány funkci `mail()` v hlavičce jako skrytí adresáti. To zaručuje, že uživatel, který dostane email s novinkou, nezná emailové adresy ostatních uživatelů, kterým byly zaslány kopie tohoto emailu.

Jako všechny administrátorské části i správa novinek obsahuje prohlížení, editaci a mazání všech záznamů z databázové tabulky *novinky*. To je ale opět velmi podobné skriptu *prohlidni.php* z části o kalendáři akcí, takže to není třeba znovu uvádět.

2.4.2.4 Správa animací

Hlavní části správy animací je její uložení do databáze a uložení obrázků, které se budou animovat do zvoleného adresáře.

Vkládání animací probíhá ve skriptu *vlozit_novou.php*. V tomto skriptu je formulářové pole pro vkládání animace obrázků a také výpisy chybových hlášení. Chyby se vypisují JavaScript funkcí `alert()`. Po zadání potřebných údajů jsou informace metodou post předány skriptu *vloz.php*. V něm jsou provedeny kontroly zadaných údajů. Kontroluje se, zda-li je vložený soubor obrázek, pomocí funkce `getimagesize()`. Tato funkce vrací rozměry obrázku, ale pokud vložený soubor není obrázek, vrací prázdný řetězec. Další kontrolou je, zda-li je obrázek vhodného typu. Ukládat se smí obrázky ve formátu gif, jpeg nebo png. Tato kontrola je prováděna pomocí superglobálního pole `$_FILE["jmeno"]["type"]`, které obsahuje informace o typu uploadovaného souboru. Z funkce `getimagesize()` se vezmou rozměry obrázku a jsou porovnány s hodnotou 467 pixelů na šířku a 350 pixelů na výšku. Pokud jsou rozměry větší, zahlásí se varování. Poté se pomocí funkcí PHP `is_uploaded_file()` a `move_uploaded_file()` uloží první obrázek do zvoleného adresáře. Adresář pro ukládání obrázků animace se vytvoří pomocí funkce `mkdir()` v uživatelské části Webu v adresáři *animace/images*. Od zadaného počtu obrázků se poté odečte jedna a je přesměrováno zpět na skript *vlozit_novou.php*. Tam je uživatel vyzván pro vložení dalšího obrázku a informace se odešlou zase na skript *vloz.php*, kde se opět provede uložení obrázku do zvoleného adresáře. Takto se pokračuje až do doby, kdy už není třeba vkládat další obrázky.

Prohlížení animací je zajištěno ve skriptu *prohlidni.php*. Animace nelze editovat, ale jinak je prohlížení podobné jako u předcházejících částí administrátorského rozhraní. V prohlížení konkrétní animace jsou také zobrazeny náhledy jednotlivých obrázků ve zmenšené podobě. Obrázky jsou zmenšovány v poměru a jejich výška nebo šířka je vždy maximálně 100 pixelů. Mazání animací je realizováno skriptem *delete.php*. Mazání animace probíhá tak, že je nejprve vymazán pomocí SQL dotazu záznam o animaci z databáze a potom jsou pomocí funkcí `unlink()` a `rmdir()` smazány soubory animace a adresář.

2.4.2.5 Správa uživatelů

Tato část je realizována poměrně jednoduchým způsobem. Nejprve jsou pomocí SQL dotazu vybrány z databázové tabulky *uživatelé* vybráni všichni uživatelé. Jednotlivé záznamy jsou uloženy do tabulky.

Poté můžeme záznamy prohlížet tak, že klikneme na login uživatele, kterého chceme zobrazit. Tyto údaje jsou zobrazeny pomocí SQL dotazu specifikovaného loginem, jež je předán v adresu URL, a v cyklu vypisovány pomocí funkce `mysql_fetch_array()`.

Jednotlivé uživatele můžeme odebírat. Po kliknutí na odkaz *odebrat* je zavolán skript *delete.php* a tam je vybraný záznam vymazán z databáze.

2.4.2.6 Editace sekce pro studenty

Tato část prezentace se bude často měnit, proto je pro administrátora připraveno rozhraní, ve kterém může obsah sekce editovat.

Základem je formulář, do kterého se zapisuje text, který bude zobrazen na stránkách. V tomto textu je možné použít editovací tlačítka. Ty jsou naprogramovány stejně jako ve správě publikací. Zformátovaný text je uložen do souboru *obsah*, který je potom do stránek vložen pomocí PHP funkce `include()`. Zároveň je text uložen do souboru *obsahtemp*, kde ale nejsou pseudo HTML značky převedeny na HTML. Tento soubor je poté zobrazen v poli `<TEXTAREA>`, aby měl administrator možnost text znovu editovat bez přítomnosti kódu HTML.

Možností administrátora je zobrazení náhledu. Náhled je zobrazen v rámu `<iframe>`. V tomto rámu je zobrazena HTML stránka *nahled.html*, která se generuje po odeslání formuláře pomocí PHP funkce `fputs()`.

2.5 Propojení s programem *Mathcad*

2.5.1 Úvod do problému

Jedním z úvah zadavatele bylo vytvoření uživatelského rozhraní pro program *mathcad*. Mathcad je matematický program od firmy *Mathsoft*. *Mathcad* je program podobný programům *Maple* nebo *Matlab*. Slouží pro matematické výpočty, které jsou zapisovány do souboru, který je obdobou dokumentu psanému ve *MS Word,* takže uživatel nemusí znát žádný programovací jazyk.

Webová prezentace by měla obsahovat rozhraní, kde by uživatel editoval pomocí HTML formulářů proměnné v dokumentu *Mathcadu*. Po změně proměnných by se zavolal *Mathcad*, ve kterém by se podle nových hodnot proměnných přepočítal výsledek. Výsledek by byl poté zobrazen na stránce.

Tento problém jsem se ve své bakalářské práci snažil řešit a při jeho řešení jsem narazil na několik závažných problémů, které implementaci značně komplikují.

2.5.2 Propojení s programem, OLE/COM architektura

Program *Mathcad* je vybaven nástroji pro propojení s jinými aplikacemi. Buď se dá *Mathcad* do aplikace přidat jako objekt vložená aplikace (*embedded application*). Druhou možností je propojení přes OLE/COM architekturu, což nabízí od verze 2001.

OLE/COM architektura je typická pro systémy *MS Windows*. COM (*Component Object Model*) je programovací technika, kdy je pomocí vytvořené komponenty umožněno sdílení objektů, funkcí a vlastností jedné aplikace jinými aplikacemi. COM jsou většinou používány vývojáři aplikací k vytvoření znovupoužitelných komponent, které potom budou použity v jiných aplikacích nebo k vytváření nových aplikací za použití již vytvořených komponent. COM komponenty jsou často používány v aplikacích MS Windows jako například MS Word nebo MS Excel, kde slouží k dynamickému propojování těchto aplikací. Lze si to představit jako rozšířený mechanismus vzdáleného volání procedur (RPC) se základy objektů. COM odděluje implementaci od rozhraní.

V dnešní době je ale architektura COM nahrazována novější .NET architekturou a od COM modelů se již upouští. I přesto je tato technologie v mnoha aplikacích pro MS Windows stále používaná.

Program *Mathcad* je také vybaven COM modelem, díky kterému je možné ovládat *Mathcad* z jiných aplikací, které podporují tuto architekturu. Díky *mathcadovským* komponentám je možné ovládat spouštění programu, provádět změny proměnných programu, získání výsledku a rekalkulace výsledku. Je také možné z jiné aplikace ukládat soubory v různých formátech, které *Mathcad* podporuje [8, 9].

2.5.3 COM architektura a PHP

PHP od verze 5 podporuje vytváření COM objektů a tím umožňuje propojení s jinými aplikacemi. K tomu slouží funkce `COM()`, ta má jako parametr název nebo id komponenty. Tato funkce vytvoří objekt pomocí kterého potom můžeme k jednotlivým metodám a vlastnostem dané aplikace přistupovat. Bohužel tato funkce není zcela dobře podporována a není příliš dobře zdokumentována.

Propojení s *Mathcadem* za pomoci PHP je možné zapsat následujícím způsobem.

```
$mathcad = COM("Mathcad.Application");
$verze = $mathcad->Version;
echo $verze." verze mathcadu";
```

Tento příklad ukazuje jak funguje připojení na aplikaci *Mathcad*. Příklad vypíše verzi *Mathcadu* za pomoci vlastnosti `Version` definované v komponentě programu.

Podobným způsobem je možné zapsat také složitější programy, které budou s *mathcadem* spolupracovat.

2.5.4 Předpokládaná implementace

V uživatelském režimu uživatel vyplněním formulářových polí změní proměnné programu a potom se pomocí funkce `COM("Mathcad.Application")` připojí na serveru na aplikaci *Mathcad*. Díky COM architektuře se do *Mathcadu* předají nové hodnoty proměnných pomocí *mathcadovské* funkce `SetValue()` nastaví nová hodnota proměnné a provede se přepočítání výsledku. Toto přepočítání zajistí funkce `recalculate()`.

Výsledek se poté z dokumentu *Mathcadu* vezme pomocí funkce `GetValue()`. V zadání bylo požadováno, aby byl na stránce zobrazen obrázek, který *Mathcad* vygeneruje. V *Mathcadu* není v COM komponentě žádná možnost pro uložení samotného obrázku, ale dokument se dá uložit jako HTML stránka. Tím jsou všechny obrázky z dokumentu vygenerovány. Dokument jako HTML stránku můžeme uložit pomocí funkce `SaveAs()` a jako druhý parametr je zadána konstanta pro HTML dokument `MCFileFormat.mcHtml`.

V administrátorském režimu administrátor vkládá informace o počtu proměnných a jejich názvech. To se ukládá do databáze. Také jsou zde vloženy zdrojové soubory *Mathcadu*, které potom uživatelé můžou spouštět.

2.5.5 Problémy při řešení

Při řešení této části se vyskytlo několik problémů. Vytvoření COM objektů v *Mathcadu* nebylo tak jednoznačné jako například pro aplikace rodiny MS Office. Starší verze jazyka PHP nedokázaly vytvořit objekt `Mathcad.Application` a tím nemohla být zahájena práce s touto aplikací. Tento problém vyřešilo nainstalování novější verze jazyku PHP, konkrétně verze 5.2.2. V této verzi se podařilo vytvořit COM objekt a pracovat s programem.

Dalším a asi největším problémem je zobrazení informací na serveru. V mé bakalářské práci je použit jako webový server server *Apache*, ale pokud je PHP spouštěno přes tento server, nejde připojení k *Mathcadu* vytvořit. Problémy mohou být následující. Špatná konfigurace serveru. Tento problém byl ale vyloučen, jelikož aplikace rodiny MS Office na serveru normálně fungují. Druhou možností, která pravděpodobně tento problém způsobuje, jsou nedostatečná práva webového serveru pracovat s COM objekty aplikace *Mathcad*. Bohužel pro řešení problému je velmi málo informací, jelikož něco podobného ještě nebylo podle dostupných informací řešeno.

Vyzkoušeny byly i jiné webové servery, jako například server *IIS* od firmy *Microsoft*, ale bohužel výsledek byl stejný. Nepomohlo ani přeinstalování *Mathcadu* na vyšší verzi.

Protože má být tato práce nasazena do běžného užívání, je v současné době nevhodné tuto část Webu v prezentaci spouštět. Speciální nastavení serverů, které by vyžadovala hrozí totiž značnými bezpečnostními mezerami a případný útočník by díky tomu mohl převzít kontrolu na serverem.

2.5.6 Vlastní řešení

Problém připojení k *Mathcadu* byl zatím vyřešen tak, že je napsán PHP skript pro konkrétní *Mathcad* aplikaci. Tento skript může být spouštěn na lokálním počítači, kde je *Mathcad* nainstalován, ale nemůže být vzdáleně spouštěn uživatelem ze serveru.

Skript je napsán pro řešení 2D náhodné funkce, která vygeneruje 2 obrázky. Obrázky ukazují změnu povrchu při změně vstupní proměnné.

3 Zhodnocení práce

3.1 Dosažené výsledky

Výsledkem mé bakalářské práce je dynamický internetový portál pro vědeckou skupinu. Portál má sloužit k informovanosti návštěvníků o nových projektech a akcích, které se týkají daného pracoviště. Portál byl vytvořen podle požadavků zadavatele.

Základními výhodami vlastní internetové prezentace jsou snadná dostupnost ze všech míst planety a možnost doplňování nových informací. Díky registraci je umožněno také sledovat, kdo stránky navštěvuje a jaká komunita lidí se o daný problém zajímá.

Během tvorby práce bylo třeba seznámit se s novými trendy publikování na internetu a hlouběji proniknout do problematiky tvorby internetových portálů specificky zaměřených na jistou skupinu. Zjistil jsem, že hlavním požadavkem ve světě vědeckých prezentací není vizuální stránka, ale stránka obsahová a důležité je také jednoduché a intuitivní ovládání aplikace, které by uživatele neodvádělo od práce nad určitým problém. Tomuto byla také má práce uzpůsobena.

Mým hlavním cílem bylo tedy vytvořit internetovou prezentaci, která bude svým ovládáním intuitivní a uživatelé si nebudou muset zvykat na nové postupy. Dalším cílem bylo vytvořit systém, který bude snadno přenositelný na různé platformy. To bylo dosaženo dodržováním standardů jazyka XHTML a CSS pro popis webových dokumentů. Proto je prezentace dobře zobrazitelná na většině dnes používaných internetových prohlížečích. Přenositelnost aplikace je také dána použitím jazyka PHP a databáze MySQL. Výhodou těchto nástrojů je to, že jsou šířeny zdarma a pohybuje se kolem nich velká komunita uživatelů. To umožňuje mnoho problémů řešit v různých diskuzích na internetu nebo pomocí manuálů, které jsou k těmto aplikacím velmi dobře zpracované.

3.2 Vlastní přínos

Hlavním přínosem mé práce je skutečnost, že byla vytvořena prezentace, která se na internetu běžně nevyskytuje. Nejedná se jen o osobní stránky, ale o stránky, které mají pro vědeckou společnost určitě svůj význam, jelikož na nich budou prezentovány výsledky, které jsou v daném oboru nové.

Přínosem pro mne je také nastudování problematiky propojení PHP s jinými aplikacemi. Zjistil jsem, že toto propojení je možné a nové verze jazyka PHP pro toto nabízejí již standardní funkce. Ovšem mnoho z nich je teprve ve stádiu vývoje a běží jen na experimentální úrovni. Tady je k řešení ještě mnoho problémů.

Ověřil jsem si, že propojení s jinými programy nezávisí pouze na jazyku PHP, ale úzce souvisí také s nastavením serveru. Bohužel toto nastavení serveru je velmi atypické nepodařilo se mi najít

žádné věrohodné informace. Je ale zřejmé, že webový server musí k tomuto získat značné oprávnění, což ale není příliš vhodné z hlediska bezpečnosti.

Vzhledem k tomu, že zadání nebylo od počátku přesně specifikováno, věnoval jsem značné úsilí k získání informací a požadavků na vytvoření takovéto aplikace. Řada funkcí, které stránky umožňují byla mým vlastním nápadem.

3.3 Další vývoj

Pro další vývoj projektu je zajímavé propojení s programem *Mathcad*. Snaha bude o vytvoření uživatelského rozhraní pro spolupráci s tímto programem. K této části je zapotřebí, aby byly vydány nové verze programů, se kterými aplikace pracuje.

Vzhledem k dnešnímu vývoji aplikací a rozšiřování technologie .NET, která také umožňuje sdílení objektů, metod a vlastností jedné aplikace aplikacemi jinými, je celkem pravděpodobné, že nové verze programů budou k tomuto již lépe přizpůsobeny. Poté by mělo být vytvořeno kompletní uživatelské rozhraní pro připojení k aplikaci a práci s ní. Uživatel by tak měl možnost využívat výhody programu *Mathcad*, aniž by ho musel mít nainstalován na lokálním počítači. Toto rozhraní by mělo být vytvořeno značně obecně a uživatelům by měla být zpřístupněna možnost zdrojové soubory upravovat a poté s nimi dále pracovat.

Vývoj se bude také zaměřovat na zlepšení prostředí pro administrátora a přidání některých nových funkcí do programu. Jednou z nich je vytvoření plnohodnotné diskuze, kde by si uživatelé mohli vyměňovat své názory k problémům tohoto vědeckého odvětví.

Další rozšíření budou postupně diskutovány se zadavatelem práce a budou vytvářeny přesně podle jeho požadavků. Lze předpokládat, že po seznámení zadavatele se všemi možnostmi vytvořených stránek, se objeví další nápady, jak možnosti rozšířit.

Literatura

[1] Kosek, J.: PHP – tvorba interaktivních internetových aplikací. 1. vydání, Praha, Grada Publishing, spol. s.r.o., 1998, s. 492

[2] Kosek, J.: HTML – tvorba dokonalých www stránek. 1. vydání, Praha, Grada Publishing, spol. s.r.o., 1998, s. 296

[3] Flanagan, D.: JavaScript, kompletní průvodce. 2. aktualizované vydání, Praha, Computer Press, 2002, s. 842

[4] Hruška, T., Burget, R.: Internetové aplikace (wap) II. část SGML, HTML, CSS, DOM studijní opora. 2006.

[5] Hruška, T., Burget, R.: Internetové aplikace (wap) IV. část JavaScript a PHP studijní opora. 2006.

[6] Zendulka, J., Rudolfová, I.: Databázové systémy I. část. Brno, studijní opora 2006

[7] Burget, R.: Tvorba webových stránek, Brno, přednášky 2006

[8] Osovský, M.: Programování v .NET, Brno, přednášky 2006

[9] COM: Component Object Model Technologies [online]. 2007 [cit. 2007-05-10]. Dostupný z WWW: <http://www.microsoft.com/com/default.mspx>

[10] History of MySQL [online]. 2007 [cit. 2007-05-10]. Dostupný z WWW: <http://dev.mysql.com/doc/refman/5.0/en/history.html>.

[11] MySQL and mSQL : The History of MySQL [online]. 2001 [cit. 2007-05-10]. Dostupný z WWW: <http://www.unix.org.ua/orelly/linux/sql/ch01_04.htm>.

[12] HTML [online]. 2007 , 13:50, 25. 4. 2007 [cit. 2007-05-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/HTML>.

[13] XHTML [online]. 2007 , 12:51, 30. 4. 2007 [cit. 2007-05-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/XHTML>.

[14] Historie PHP a souvisejících projektů : Historie PHP [online]. 2007 , 24. 3. 2007 [cit. 2007-05-10]. Dostupný z WWW: <http://cz.php.net/manual/cs/history.php>.

[15] Cascading Style Sheets [online]. 2007 , 23:08, 29. 4. 2007 [cit. 2007-05-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Cascading_Style_Sheets>.

Seznam příloh

Příloha 1. Uživatelský manuál

Příloha 2. CD se zdrojovými texty programu

UŽIVATELSKÝ MANUÁL

Uživatelský manuál

Obsah

Uživatelský manuál ... 2
Obsah ... 2
Úvod ... 3
1 Uživatelská část ... 3
1.1 Seznámení s portálem ... 3
1.1.1 Hlavní okno portálu ... 3
1.1.2 Registrace nového uživatele ... 5
1.1.3 Přihlášení uživatele ... 5
1.1.4 Kalendář akcí ... 6
1.1.5 Publikace ... 7
1.1.6 Animace ... 8
1.1.7 Vyhledávání ... 9
1.2 Shrnutí ... 9
2 Administrátorský režim ... 10
2.1 Seznámení s režimem ... 10
2.1.1 Hlavní obrazovka ... 11
2.1.2 Kalendář akcí ... 12
2.1.2.1 Vkládání nového záznamu ... 12
2.1.2.2 Prohlížení všech akcí ... 13
2.1.3 Správa publikací ... 14
2.1.3.1 Vkládání nové publikace ... 14
2.1.3.2 Prohlížení všech publikací ... 15
2.1.4 Správa novinek ... 15
2.1.4.1 Vložení nové novinky ... 15
2.1.4.2 Prohlížení novinek ... 16
2.1.5 Správa animací ... 16
2.1.5.1 Vkládání animací ... 16
2.1.5.2 Prohlížení animací ... 17
2.1.6 Prohlížení uživatelů ... 18
2.1.7 Editace sekce pro studenty ... 18
2.2 Shrnutí ... 19

Úvod

Úkolem manuálu je seznámit uživatele s ovládáním všech funkcí programu. Manuál je rozdělen na 2 části, první popisuje část portálu pro běžného uživatele a druhá část bude sloužit jako příručka pro administrátory.

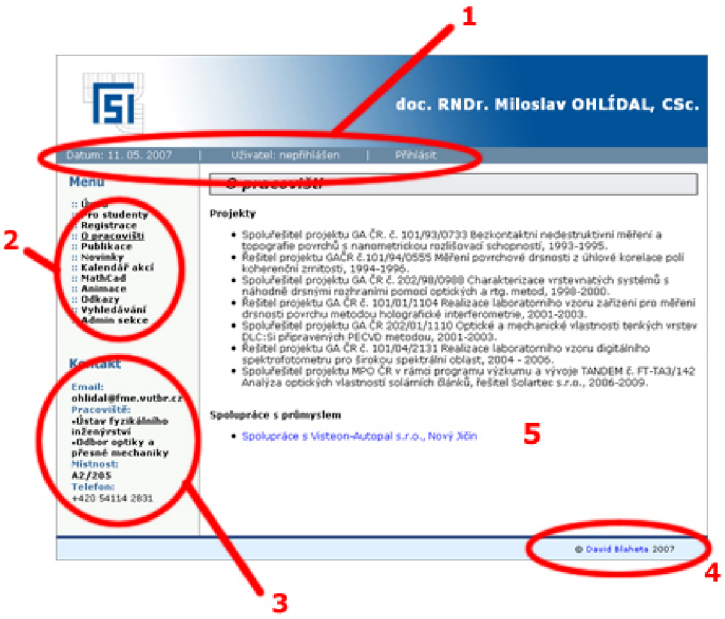
K ovládání programu nejsou zapotřebí žádné zvláštní znalosti, ale jen standardní ovládání PC a alespoň základní zkušenost s ovládáním internetových prezentací. Ani administrátor nemusí vstupovat do zdrojových kódů programů, ale jistá znalost HTML by mohla být výhodou. Není to však v žádném případě podmínka.

1 Uživatelská část

1.1 Seznámení s portálem

V této kapitole bude uživatel seznámen se základní strukturou a navigací po portálu. Portál je standardní webovou aplikací a uživatel, který s podobnými již pracoval, by se měl velmi rychle zorientovat.

1.1.1 Hlavní okno portálu



Obrázek 1: Hlavní okno portálu

Jak je vidět na obrázku hlavní okno portálu obsahuje 4 hlavní části a samozřejmě obsahovou část. V této kapitole budou popsány jednotlivé části hlavní stránky.

1. **Hlavička a informační pruh (bublina 1):**

   Hlavička obsahuje logo strojní fakulty a jméno zadavatele práce. Pro uživatele je ale důležitý informační pruh, který je na obrázku označen. Informační pruh obsahuje tři základní části. Nejprve informuje o aktuálním datu, poté informuje, zda-li je uživatel přihlášen a pod jakým loginem. Poslední část je odkaz, který po kliknutí přesměruje uživatele na stránku s přihlášením.

2. **Menu (bublina 2):**

   Jedná se o část na obrázku označena bublinou 2 a je to jedna z nejdůležitějších částí Webu. Menu zajišťuje veškerou navigaci na Webu. Obsahuje položky, které uživatele přesměrují na jednotlivé části Webu. Odkazem *úvod* se uživatel dostane zpět na úvodní stránku. Odkaz *admin sekce* slouží jen pro administrátory a bez platného loginu a hesla k této části nemá uživatel přístup. Jednotlivé části Webu budou popsány dále v manuálu.

3. **Kontakt (bublina 3):**

   Tato část obsahuje kontaktní informace zadavatele. Na jednotlivé položky se dá kliknout a budete přesměrováni na příslušné místo.

4. **Copyright (bublina 4):**

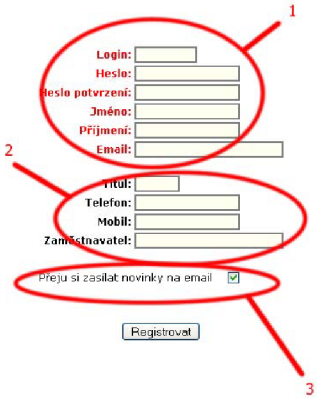
   Copyright tvůrce Webu je umístěn v patičce. Po kliknutí na jméno je možno odeslat mail.

5. **Střed (číslo 5):**

   Střed je částí webu, která vždy obsahuje všechny informace příslušné části.

1.1.2 Registrace nového uživatele

Aby uživatel získal přístup k veškerým možnostem, které portál nabízí, je nutné, aby se zaregistroval. Registraci uživatel provádí pomocí formulářového pole, kde vyplňuje osobní údaje.



Obrázek 2: Formulářové pole

1. **Povinné údaje (bublina 1):**

   Tyto údaje jsou nutné a uživatel je musí vyplnit, jinak registrace nebude provedena.

1. **Volitelné údaje (bublina 2):**

   Tyto údaje nejsou nutné pro registraci, ale uživatel je může vyplnit. Studenti FSI, kteří navštěvují některý z předmětů na němž se doc. Ohlídal podílí, mohou do kolonky zaměstnavatel uvést *student FSI,* poté mohou dostávat emaily, které se týkají výuky, nebo jiných důležitých informací ohledně školy.

2. **Novinky na email (bublina 3):**

   Zaškrtnutím tohoto pole uživatel souhlasí se zasíláním novinek na email. Novinky budou zasílány na adresu, kterou při registraci uvedl.

1.1.3 Přihlášení uživatele

Pro přihlášení je třeba, aby uživatel kliknul na odkaz *Přihlásit* v informační části hlavičky stránky. Poté se objeví formulář pro přihlášení.

Obrázek ukazuje formulář pro přihlášení, kde uživatel vyplní login a heslo. V těchto položkách záleží na velikosti písmen tzn. Heslo ≠ heslo. Po odeslání formuláře jsou údaje ověřeny a pokud vše bylo zadáno v pořádku, uživatel je přihlášen.

1.1.4 Kalendář akcí

Kalendář akcí je velmi užitečná aplikace, která slouží pro zisk informací a připravovaných nebo již proběhlých akcích. Uživatel může listovat kalendářem po měsících nebo po rocích a jsou mu zobrazovány akce na dané období.

Obrázek 3: Kalendář akcí

Obrázek ukazuje uspořádání hlavní navigace v kalendáři.

1. **Listování měsíci kalendáře (bublina 1):**

   Při vstupu na stránku je zobrazen aktuální měsíc. Měsíci se dá listovat tam a zpět kliknutím na jeho název.

2. **Přechod na jiný rok (bublina 2):**

   Implicitně je nastaven aktuální rok, ale je možno zadáním do pole *rok* přejít na jiný. Přechod se potvrdí klávesou *ENTER*.

3. **Obsah kalendáře (bublina 3):**

   V této části jsou vypsány akce naplánované na zvolené období.

1.1.5 Publikace

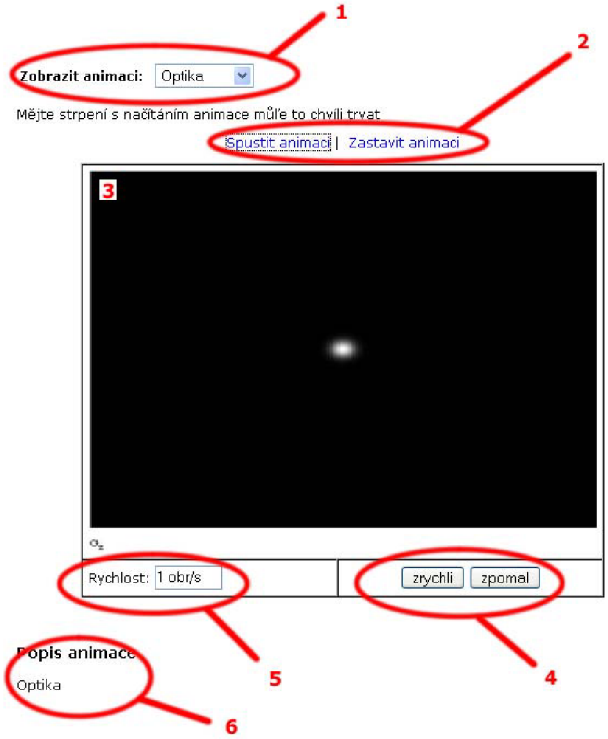
Seznam vybraných publikací je zobrazen v této části prezentace. Uživatel má možnost prohlédnout si abstrakt dané publikace nebo její celé znění, pokud je to vyplněno.

Obrázek 4: Seznam publikací

1. **Citace publikace (bublina 1):**
   Citace je vytvořena podle standardů a jsou v ní uvedeni autoři, její název a kde byla publikována.
2. **Abstrakt (bublina 2):**
   Pokud je odkaz aktivní je možné zobrazit abstrakt k dané publikaci.
3. **Zobrazení celé publikace (bublina 3):**
   Pokud je publikace uložena na serveru, je možno zobrazit její celé znění. Po kliknutí na odkaz bude publikace zobrazena. Tuto možnost mají pouze registrovaní uživatelé. Ostatní se musí zaregistrovat (viz. Kapitola 1.1.2).

1.1.6 Animace

Jednou ze služeb portálu je vytváření animací ze samostatných obrázků. Tato služba nejprve načte potřebné obrázky do paměti, což může být na pomalejších typech připojení časově náročné. Mějte proto strpení s rychlostí načítání animací.



Obrázek 5: Okno s animací

1. **Výběr animace (bublina 1):**

   Z formulářového pole je možno si vybrat jednu animaci, která bude po načtení zobrazena.

2. **Ovládání animace:**

   Uživatel má možnost animace ovládat, může jí spustit nebo zastavit (bublina 2). Běh animace je možno pomocí tlačítek (bublina 4) zrychlovat nebo zpomalovat. Výchozí je rychlost 1 obr/s a zrychlení nebo zpomalení je vždy dvojnásobné. Maximální rychlost animace je 64 obr/s. Rychlost animace se zobrazuje v poli 5.

3. **Vlastní animace (3):**

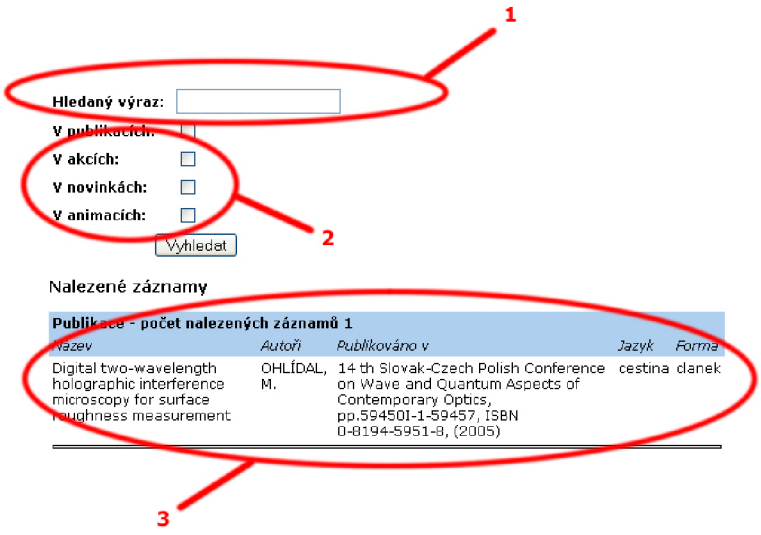
   V této části se budou jednotlivé obrázky za sebou střídat a tím bude animace vytvořena.

4. **Popis animace (bublina 6):**

   Pokud je k animaci připojen nějaký popis, bude zobrazen v této části.

1.1.7 Vyhledávání

Vyhledávání je pro uživatele obecně velmi využívanou funkcí. V této kapitole jsou popsány možnosti vyhledávání v publikacích, akcích, novinkách a animacích.



Obrázek 6: Vyhledávání

1. **Hledaný výraz (bublina 1):**
   Do tohoto pole bude zapsán hledaný výraz. V zadávání výrazu nezáleží na velikosti písmen.
2. **Kde se bude vyhledávat (bublina 2):**
   Specifikace místa, kde se bude vyhledávat. Může být vybrána jedna až všechny položky.
3. **Výsledky vyhledávání (bublina 3):**
   Výsledky hledání jsou zobrazeny v této části. Jsou uspořádány a zformátovány do přehledných tabulek. Kliknutím na záznam je možno přejít na místo, kde se nachází.

1.2 Shrnutí

V uživatelském režimu se uživateli nabízí standardní webová stránka, která mu umožní přístup ke všem částem webové prezentace.

Některé funkce Webu jsou však přístupné až po přihlášení. Aby se uživatel mohl přihlásit musí vyplnit registrační formulář a poté zadat uživatelské jméno a heslo.

Web byl navržen podle standardů publikování na internetu. Jeho obsah by měl být dobře zobrazitelný na všech běžných prohlížečích.

2 Administrátorský režim

2.1 Seznámení s režimem

V této části příručky nalezne administrátor návod potřebný pro všechny základní funkce, které portál nabízí.

Administrátorský režim je přístupný po zadání administrátorského jména a hesla. Pro vstup slouží tato přihlašovací obrazovka.

Obrázek 7: Přihlášení pro administrátory

2.1.1 Hlavní obrazovka

Po úspěšném přihlášení je administrátor přesměrován na hlavní okno administrátorského režimu.

Obrázek 8: Hlavní okno

Hlavní obrazovka obsahuje tyto části:

1. **Hlavní navigační panel (bublina 1):**

   Tento panel nabízí odkazy do všech částí administrátorského režimu. Po kliknutí je administrátor přesměrován do dané sekce. Je zde zároveň tlačítko *odhlásit* pro odhlášení.

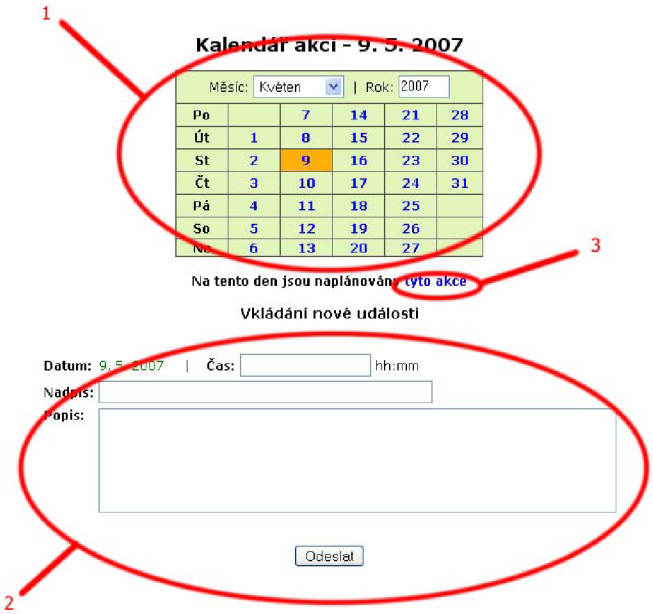
2. **Hlavní okno**

   V této části je zobrazen obsah dané sekce.

2.1.2 Kalendář akcí

Kalendář akcí umožňuje administrátorovi plánování akcí na určitý den pomocí jednoduchého a intuitivního rozhraní.

2.1.2.1 Vkládání nového záznamu



Obrázek 9: Vkládání do kalendáře akcí

1. **Aplikace kalendář (bublina 1):**

   Pomocí této tabulky vybírá administrátor den, na který je akce naplánována. Po výběru měsíce a roku, které potvrdí stiskem *ENTER,* se zobrazí kalendář na daný měsíc. Na všechny dny v kalendáři se dá kliknout. Tím se objeví formulář pro vkládání nové akce na tento den. Políčka v kalendáři vybarvená oranžově jsou dny, na které jsou již nějakou akcí obsazeny.

2. **Vkládání nové akce (bublina 2):**

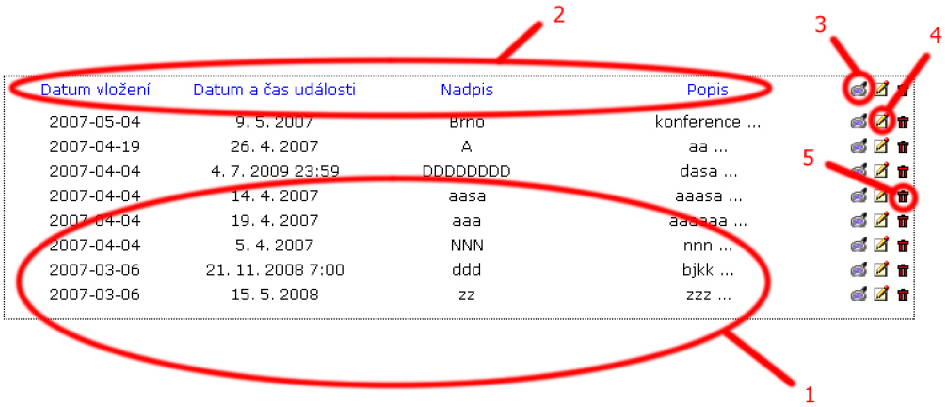
   Nová akce je vkládána do databáze přes vkládací formulář. Povinné je vyplnění názvu akce. Dále je možno zadat čas akce, který je nutné ve formátu hh:mm. Připojit se dá ještě popis akce a po odeslání je akce na zvolený den uložena do databáze.

3. **Prohlížení akcí na určitý den (bublina 3):**

   Pokud je již na vybraný den naplánovaná akce, je v této části možné tyto akce prohlížet, případně editovat nebo mazat. O editaci viz další kapitola.

2.1.2.2 Prohlížení všech akcí

Druhou částí kalendáře akcí je prohlížení všech záznamů.



Obrázek 10: Prohlížení všech záznamů

1. **Zobrazení všech akcí (bublina 1):**

   Tabulka obsahuje základní údaje o všech akcích. Vždy se zobrazuje 12 akcí na stránku. Další akce mohou být zobrazeny kliknutím na další stránku.

2. **Řazení akcí (bublina 2):**

   Akce je možné řadit vzestupně či sestupně podle různých kritérií. Kliknutím na příslušné kritérium jsou podle něj akce seřazeny. Po opětovném kliknutí se směr řazení změní. Implicitně jsou akce řazeny podle data vložení, tzn. že na prvním místě je akce vkládána jako poslední.

3. **Prohlížení vybrané akce (bublina 3):**

   Kliknutím na obrázek lupy je možné určitou akci prohlédnout. Zobrazí se všechny dostupné informace o ní.

4. **Editace akce (bublina 4):**

Obrázek 11: Editace akce

Editovat se dá akce pomocí změny hodnot ve formulářových polích. Ve formulářích jsou hodnoty, které byly původně zadány. Tyto hodnoty lze měnit a změněnou akci poté uložit stisknutím tlačítka *Změnit.*

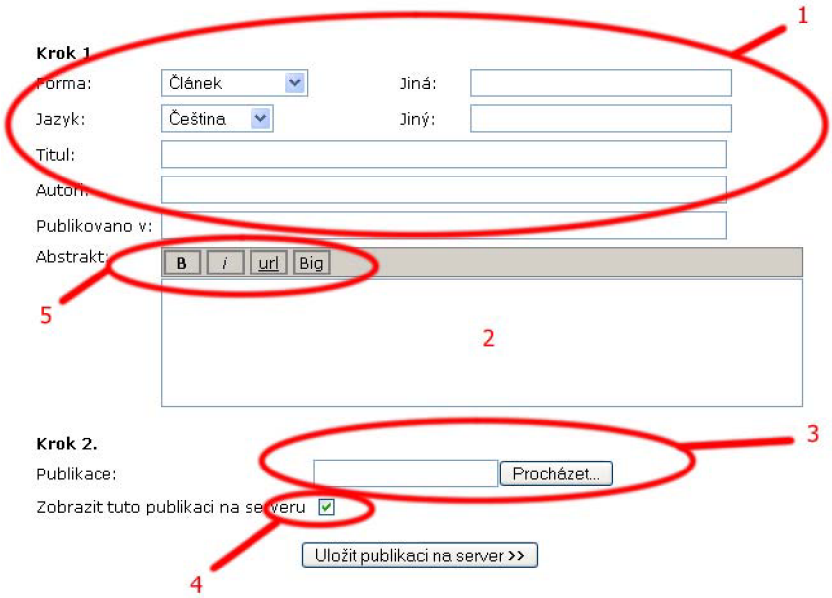
5. **Mazání akcí (bublina 5):**

   Akci můžete smazat kliknutím na ikonu odpadkového koše. Administrátor bude tázán, zda-li chce akci smazat. Po potvrzení je akce odstraněna z databáze.

2.1.3 Správa publikací

Správa publikací má jako všechny části administrátorského režimu 2 části. Jsou to vkládání nové publikace a prohlížení všech publikací.

2.1.3.1 Vkládání nové publikace



Obrázek 12: Vkládání nové publikace

1. **Vyplnění povinných údajů (bublina 1):**

   Pomocí formulářů musí být vyplněny základní informace o publikaci. Pokud je jazyk nebo forma publikace jiná, je třeba ve výběru zvolit položku *jiná* a poté toto specifikovat v příslušném poli.

   Příklad vyplnění:

   Jazyk: Angličtina

   Forma: Článek

   Titul: Digital two-wavelength holographic interference microscopy for surface roughness measurement

Autoři: OHLÍDAL, M.

Publikováno v: 14 th Slovak-Czech Polish Conference on Wave and Quantum Aspects of Contemporary Optics, pp.59450I-1-59457, ISBN 0-8194-5951-8, (2005)

2. **Napsání abstraktu (2):**

   Do tohoto pole může být vožen abstrakt. Abstrakt může být formátován pomocí formátovacích tlačítek (bublina 5). Po stisknutí např. tlačítka pro vkládání tučného textu jsou do pole vloženy 2 značky `[b]` a `[/b]`. Text napsaný mezi tyto značky bude zobrazen jako tučný. Vkládání odkazů je podobné. Po stisku tlačítka *url* budou vloženy značky `[url='neco']text odkazu[/url]`, kde "neco" nahradíme url adresou stránky, na kterou se odkazujeme, a jako text odkazu uvedeme text, který bude zobrazen.

3. **Vkládání publikace (bublina 3):**

   Na server může být vložen celý soubor s publikací. Vložení provádíme přes tento formulář.

4. **Zobrazení publikace (bublina 4):**

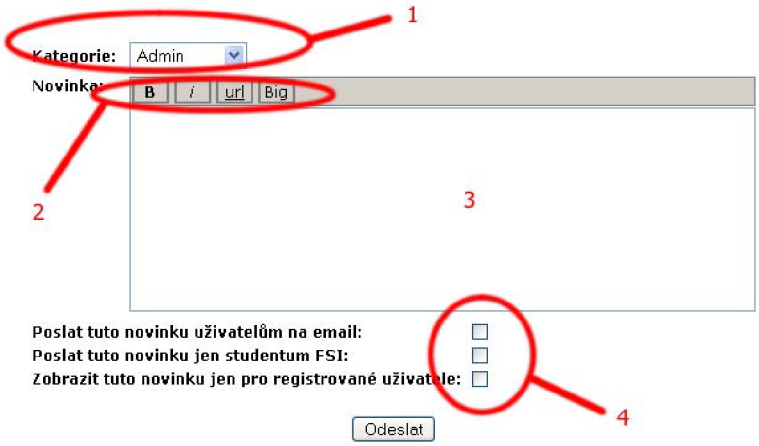
   Publikace nemusí být na serveru zobrazena, což specifikujeme pomocí tohoto pole.

2.1.3.2 Prohlížení všech publikací

Tato část je analogická jako prohlížení kalendáře akcí (viz. 2.1.2.2)

2.1.4 Správa novinek

2.1.4.1 Vložení nové novinky



Obrázek 13: Vkládání novinky

1. **Vybrání kategorie (bublina 1):**

   Výběr kategorie novinky probíhá přes formulářové pole select.

2. **Napsání novinky (3):**

   V tomto poli můžeme napsat novinku a využít základní formátovací tlačítka (bublina 2 viz. kapitola 2.1.3.1).

3. **Odesílání novinek na email (bublina 4):**

   V této části je možné specifikovat, jestli se má novinka odeslat na emaily uživatelů. Možnost je odeslat novinku všem uživatelům zatržením políčka *poslat všem* nebo *jen studentům FSI*, kteří to uvedli při registraci (viz. 1.1.2). Poslední je možnost, že novinka bude zobrazena jen pro registrované uživatele.

2.1.4.2 Prohlížení novinek

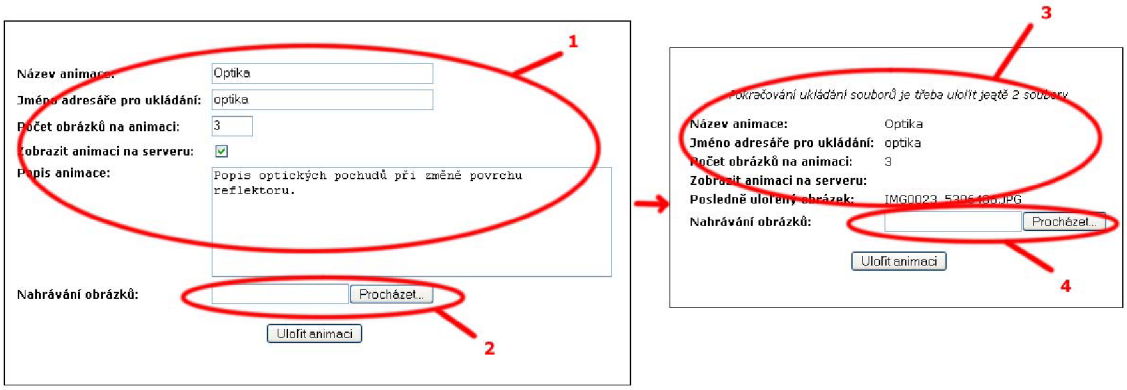
Prohlížení novinek je shodné jako prohlížení kalendáře akcí (viz. 2.1.2.2).

2.1.5 Správa animací

Administrátor má možnost vytvářet animace jako posloupnost obrázků. Opět jsou zde dvě části, a to vkládání nové animace a prohlížení animací.

2.1.5.1 Vkládání animací

Pro vkládání animací je připraveno administrátorské rozhraní. S jeho pomocí je vytvoření animace velmi jednoduché.



Obrázek 14: Vkládání animací

1. **Vyplnění základních údajů (bublina 1):**

   Do připravených formulářů je třeba uvést základní informace o animaci.

2. **Vložení souboru s animací (bublina 2):**

   Do tohoto pole je nutné zadat jméno prvního obrázku pro animaci. Po uložení se přejde na další obrazovku, kde ukládání souborů animace pokračuje.

3. **Zobrazení základních informací (bublina 3):**

   Díky těmto informacím má administrátor stále přehled o ukládané animace. Vidí také jméno posledně vkládaného souboru.

4. **Vkládání dalších souborů (bublina 4):**

   V tomto poli je uvedeno jméno dalšího souboru animace. Pokud je ještě co vkládat, okno pro pokračování vkládání se otevře znovu.

2.1.5.2 Prohlížení animací

Je podobné jako prohlížení kalendáře akcí a dalších částí Webu, ale animace nelze editovat, ale pouze mazat a prohlížet.

Obrázek 15: Prohlížení animací

1. **Prohlížení informací o animaci (bublina 1):**

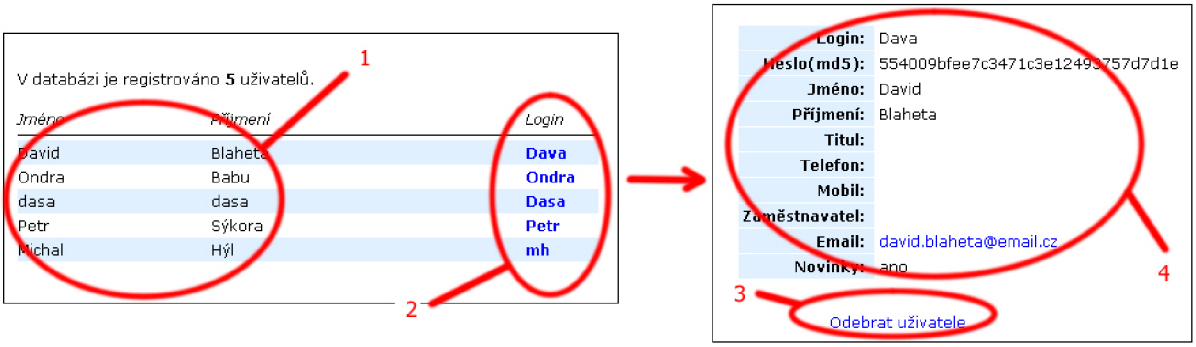
   Zde se zobrazují informace o animaci, počet obrázků a další.

2. **Náhledy obrázků (bublina 2):**

   V této části jsou náhledy obrázků, které jsou použité v animaci. Po kliknutí na obrázek se zobrazí jeho plná velikost.

2.1.6 Prohlížení uživatelů

Administrátor má možnost procházet uživatelské účty jednotlivých uživatelů a také uživatele mazat.

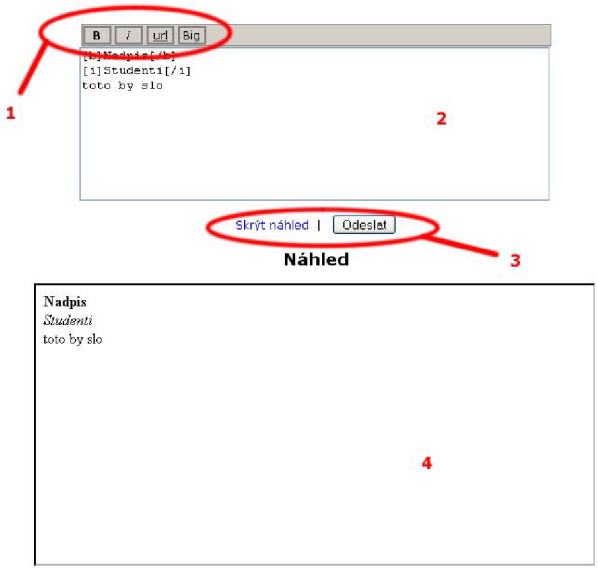


Obrázek 16: Prohlížení uživatelů

Přehled všech uživatelů (bublina 1). V tabulce je zobrazen přehled všech registrovaných uživatelů. Po kliknutí na login je možné vidět všechny informace o uživateli (bublina 2) a případně uživatele po kliknutí na odkaz smazat (bublina 3).

2.1.7 Editace sekce pro studenty

Pro studenty se informace často mění. Pro vkládání do této sekce slouží následující administrátorské rozhraní.



Obrázek 17: Editace sekce pro studenty

1. **Formátovací tlačítka (bublina 1):**
   Pro editaci sekce slouží formátovací tlačítka. Formátovaní probíhá stejným způsobem, jako v ostatních částech administrátorského režimu.
2. **Text sekce (2):**
   Zde administrátor napíše text, který bude poté na stránkách zobrazen.
3. **Zobrazení náhledu (4):**
   Pro lepší přehled má administrátor možnost zobrazit náhled na výsledný vzhled sekce. Pro zobrazení náhledu je třeba kliknout na příslušný odkaz (bublina 4). Tam se dá také náhled skrýt.

2.2 Shrnutí

Administrátorský režim umožňuje po přihlášení plnou kontrolu nad Webem. Administrátor má celou řadu možností jak editovat obsah stránek bez nutnosti zasahovat do zdrojového kódu. Nutností není ani znalost HTML i když je to výhodou.