

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WEBOVÝ PLÁNOVACÍ SYSTÉM

BAKALÁŘSKÁ PRÁCE

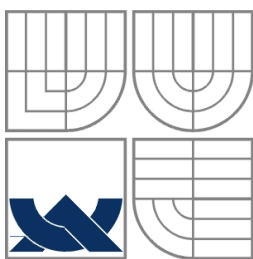
BACHELOR'S THESIS

AUTOR PRÁCE

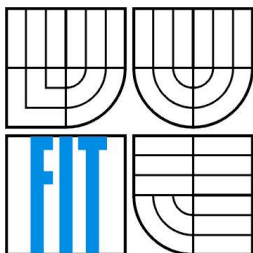
AUTHOR

TOMÁŠ MÁTL

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

WEBOVÝ PLÁNOVACÍ SYSTÉM

WEB PLANNING SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ MÁTL

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MARTIN STRAKA

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Mátl Tomáš**
Obor: Informační technologie
Téma: **Webový plánovací systém**
Kategorie: Web

Pokyny:

1. Seznamte se s principy tvorby dynamických www stránek a jazykem HTML, PHP nebo Javascript případně databází MySQL.
2. Prostudujte a navrhnete úplný webový plánovací systém pro manažerské i osobní potřeby.
3. Vytvořte detailní návrh systému a modelujte jej pomocí UML.
4. Plánovací systém realizujte a pečlivě zdokumentujte (on-line nápověda).
5. Zhodnoťte dosažené výsledky a diskutujte možné pokračování a rozšíření projektu.

Literatura:

- Welling, L., Thomsonová, L.: PHP a MySQL - rozvoj webových aplikací, Softpress 2003. 910 s. ISBN 80-86497-60-7.
- Kosek, J.: HTML, tvorba dokonalých www stránek, Grada 1998. 296 s. ISBN 80-7169-608-0.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních 3 bodů zadání

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: **Straka Martin, 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

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Tomáš Mátl**
Id studenta: 89058
Bytem: Újezd 462, 783 96 Újezd u Uničova
Narozen: 19. 06. 1984, Šternberk
(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: Webový plánovací systém
Vedoucí/školitel VŠKP: Straka Martin, 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 tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel



.....

Autor

Abstrakt

Tato práce se zabývá tvorbou informačního systému webového plánování. Jsou zde popsány jednotlivé fáze vývoje tohoto systému, od počáteční specifikace požadavků a analýzy požadavků na takovýto systém až po návrh systému, modelování a implementaci. Systém byl navržen za použití technologií HTML, CSS, PHP, Javascript a databáze MySQL na platformě Apache.

Klíčová slova

Plánovací systém, informační systém, databáze, HTML, CSS, PHP, Javascript, MySQL

Abstract

This work deals with creation of an information system for web planning. The work involves each phase of system development from the first requirements specification and analysis to system planning, modeling and implementation. System was implemented using technologies such as HTML, CSS, PHP, Javascript, MySQL database and Apache web server.

Keywords

Planning system, information system, database, HTML, CSS, Javascript, MySQL

Citace

Tomáš Mátl: Webový plánovací systém, bakalářská práce, Brno, FIT VUT v Brně, 2007

Webový plánovací systém

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Martina Straky
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
Datum

© Tomáš Mátl, 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	1
Úvod	3
1 Analýza požadavků na webový plánovací systém	4
1.1 Specifikace požadavků na systém	4
1.2 Úlohy jednotlivých uživatelů systému	4
1.2.1 Běžný uživatel.....	5
1.2.2 Správce systému.....	5
2 Návrh systému	6
2.1 UML.....	6
2.2 Diagram případů užití.....	6
2.3 ER diagram.....	7
2.4 Transformace konceptuálního modelu na tabulky relační databáze.....	7
2.4.1 Normální formy	8
2.5 Popis tabulek	9
2.5.1 Uživatel	9
2.5.2 Událost	9
2.5.3 Kontakt.....	10
2.5.4 Poznámka.....	10
3 Implementace systému.....	11
3.1 Použité technologie	11
3.1.1 Apache	11
3.1.2 MySQL	11
3.1.3 HTML a CSS	12
3.1.4 PHP	12
3.1.5 Javascript	13
3.2 Instalace a konfigurace serveru	13
3.2.1 Instalace komponent	13
3.2.2 Konfigurace PHP	14
3.2.3 Konfigurace MySQL	14
3.3 Implementace uživatelského rozhraní.....	14
3.3.1 Registrace.....	14
3.3.2 Přihlašování.....	15
3.3.3 Rozhraní uživatele	15
3.3.4 Rozhraní administrátora.....	17

3.3.5	Automatické zasílání upomínek.....	18
3.3.6	Vzhled aplikace.....	18
3.4	Problémy při implementaci.....	19
4	Náměty na rozšíření systému.....	20
4.1	Zabezpečení systému pomocí HTTPS.....	20
4.2	Rozšíření funkcí.....	20
4.3	Import a export.....	20
4.4	Instalace a konfigurace.....	20
5	Závěr.....	22
	Literatura.....	23
	Seznam příloh.....	24

Úvod

Plánování činností je užitečné prakticky pro každého člověka. Plánovat lze jakékoli události ať už pracovního nebo osobního rázu. V dnešní hektické době navíc roste počet různých úkolů, schůzek, poznámek a pamatovat si vše je někdy velmi obtížné.

Plánování času bývalo, a stále ještě bývá, řešeno pomocí klasických papírových plánovacích kalendářů. V dnešní době však rozvoj informačních technologií velmi pokročil, a na jejich místo se dostávají různé elektronické formy plánování. Ve firmách většinou dominuje plánování pomocí programu Microsoft Outlook, který má komplexní správu emailů a plánování. V mém plánovacím systému jsem se zaměřil spíše na plánování pro jednotlivce a domácí uživatele, kteří nepotřebují tolik funkcí, nebo kteří si nemohou placené plánovací programy dovolit.

Zaměření na plánování na webu má několik výhod. Především systém je dostupný odkudkoliv, kde se lze připojit na internet. Systém je také nezávislý na operačním systému uživatele a pro jeho zobrazení postačuje jakýkoli modernější grafický internetový prohlížeč s podporou javascriptu a cookies. Nevýhodou může být nutnost připojení k internetu, nebo relativně nižší bezpečnost dat, než u lokálního programu.

První kapitola práce se zaměřuje na analýzu požadavků na takovýto informační systém. Od požadavků pro manažerské i osobní potřeby. Dále jaké události mohou očekávat uživatelé od systému.

Druhá kapitola se zabývá zpracováním těchto požadavků a návrhu výsledné aplikace. Návrh je řešen pomocí UML diagramů, je zde také popsán jejich význam a popis tvorby. Je zde uvedena také transformace těchto diagramů na výsledné tabulky v databázi a příslušná teorie.

Třetí kapitola popisuje implementaci daného systému. Jsou zde uvedeny prostředky použité při implementaci systému. Je zde také popsána instalace systému a jeho nastavení, řešení zajímavých částí systému a nastíněno uživatelské prostředí aplikace. Rovněž jsou nastíněny problémy, které jsem řešil při implementaci.

V předposlední čtvrté kapitole je popsán nástin budoucího možného vývoje aplikace, různé vylepšení a funkce.

Na závěr je shrnutí práce, jejích výsledků a zhodnocení zkušeností.

1 Analýza požadavků na webový plánovací systém

1.1 Specifikace požadavků na systém

Základním požadavkem na tento systém je podpora více uživatelů. Uživatele budeme muset nějak rozlišit, proto zde bude nutná databáze uživatelských účtů a přihlašování do aplikace. Z důvodu aby nebyl administrátor příliš zatížen požadavky, měli by uživatelé mít možnost vytvářet účty sami.

Administrací těchto uživatelů by se měla zabývat pověřená osoba správce systému. Tento správce by měl mít možnost blokovat, mazat, vytvářet nové správce, atd. Rovněž by měl řešit různé problémy uživatelů.

Dalším důležitým požadavkem bude bezpečnost daného systému. Protože systém bude umístěn na internetu, může se snadno stát cílem různých útoků. Bezpečnost je o to více důležitá, jelikož v systému budou uloženy osobní data uživatelů a další důvěrné informace. Proto by měl být systém chráněn zabezpečeným přihlašováním pomocí uživatelského jména a hesla.

Samozřejmým požadavkem na systém jsou plánovací funkce. Sem můžeme zahrnout zobrazování kalendáře, zadávání událostí, poznámek, schůzek, kontaktů atd. Zadané informace by měly být lehce dostupné a možno v nich vyhledávat popřípadě filtrovat obsah. Zadávané údaje by mělo být možno snadno upravovat a mazat.

Na události by mělo být patřičným způsobem upozorňováno. Systém by měl proto disponovat různými druhy automatických upomínek a mít možnost je nastavovat. Například odesílání na email, nebo na mobilní telefon formou SMS.

Systém by měl být snadný na používání, přehledný a s decentním grafickým designem. Samozřejmostí by měla být přehledná nápověda ke všem jednotlivým funkcím systému.

Protože systém bude vystaven na internetu, měl by splňovat normy pro jednotlivé použité internetové technologie. Samozřejmostí by tak měl být validní HTML a CSS kód aplikace. Aplikace také musí být bez problémů použitelná na nejvíce používaných internetových prohlížečích jako je Internet Explorer, Mozilla Firefox, Opera, nebo Konqueror.

1.2 Úlohy jednotlivých uživatelů systému

Systém bude rozlišovat dvě hlavní skupiny uživatelů. Běžný uživatel bude mít přístup k vlastním plánovacím funkcím systému. Na běžné uživatele budou dohlížet správci systému, kteří tvoří druhou

skupinu. V této kapitole budou popsány funkce, které daní uživatelé v systému budou moci vykonávat.

1.2.1 Běžný uživatel

- Uživatel si vytvoří vlastní účet. Pro vytvoření účtu bude muset zadat informace o sobě a zvolit si přihlašovací jméno a heslo. Po vytvoření účtu jej bude muset aktivovat formou aktivačního odkazu zasláného emailem.
- V případě zapomenutí přihlašovacího hesla má uživatel možnost zažádat o zaslání nového hesla.
- Uživatel má poté možnost se libovolně přihlašovat do systému a odhlašovat se z něj.
- V systému má možnost změnit údaje o sobě, například změnit email, zadat nové heslo atd.
- Dále může používat vlastní plánovací funkce systému. Zadávat události, zobrazovat, třídit, nebo tisknout. Může dále zadávat, zobrazovat nebo tisknout události a kontakty. V uvedených informacích může samozřejmě kontextově vyhledávat.
- Na zadávané informace má možnost zasílat upomínky.
- V případě problémů s ovládáním aplikace má možnost zobrazit si nápovědu.

1.2.2 Správce systému

- Správce systému má za úkol spravovat účty běžných uživatelů a ostatních správců.
- Správce může zablokovat problémového uživatele, nebo ho rovnou smazat.
- Při problémech při ověření uživatelského účtu může tento účet aktivovat ručně.
- Dále má možnost vytvořit nový správcovský účet, popřípadě povýšit uživatele na správce. Může také samozřejmě zrušit práva správce u uživatele.
- Správce má přístup k základním uživatelským údajům jako je jméno, příjmení, nebo email, aby mohl s uživateli komunikovat. Nemá ovšem přístup k privátním datům uživatelů, jako jsou jejich události, kontakty, datum narození atd.
- Správce má také přístup k vlastním plánovacím funkcím, zde po tom může používat funkce popsané v kapitole 1.2.1.

2 Návrh systému

Základem pro návrh systému, byla analýza požadavků zpracovaných v kapitole 1. Pro návrh modelů byl použit dle specifikace zadání jazyk UML.

2.1 UML

UML je zkratka pro Unified Modeling Language. Tento jazyk má za cíl sjednotit používané prostředky pro modelování aplikací. Vývoj jazyka UML začal v roce 1994 na základě principu objektově orientovaného návrhu. V roce 1997 pak byl tento jazyk přijat za standart a postupně vytlačil ostatní modelovací jazyky.

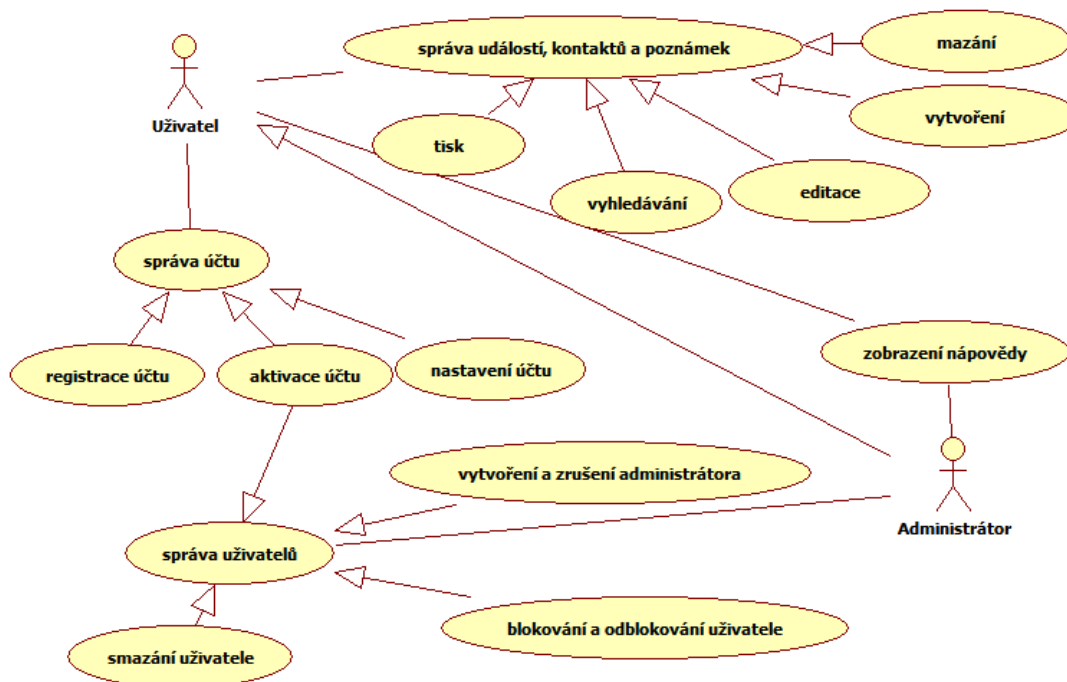
Jazyk UML obsahuje 9 různých modelovacích diagramů. Pro modelování mého systému jsem zvolil modelování pomocí diagramu případů užití, neboli use case diagram. Dále diagram entitně-vztahový neboli ER diagram. K modelování diagramu případů použití byl použit program StarUML, pro ER diagram pak Toad Data Modeler ve freeware verzi.

2.2 Diagram případů užití

Na základě požadavků na daný systém popsaných v kapitolách 1.2 a 1.3 jsem vytvořil model případů užití viz obrázek 2.1.

Diagram se skládá z aktérů a jejich rolí v systému. Jak je z obrázku patrné, tento systém má dva aktéry – uživatele a administrátora. Dále jsou v diagramu role jednotlivých aktérů. Tyto role vycházejí ze specifikace požadavků na tento systém. Některé role jsou pak společné pro oba aktéry, jako například aktivace uživatelského účtu. Zároveň je zde naznačen generalizovaný vztah administrátor – uživatel. Tento vztah znamená, že administrátor je v podstatě také uživatel aplikace, takže může používat stejné funkce systému jako uživatel.

Z tohoto diagramu jsem pak vycházel při implementaci uživatelského prostředí systému popsaného v kapitole 4.



Obrázek 2.1 – diagram případů užití

2.3 ER diagram

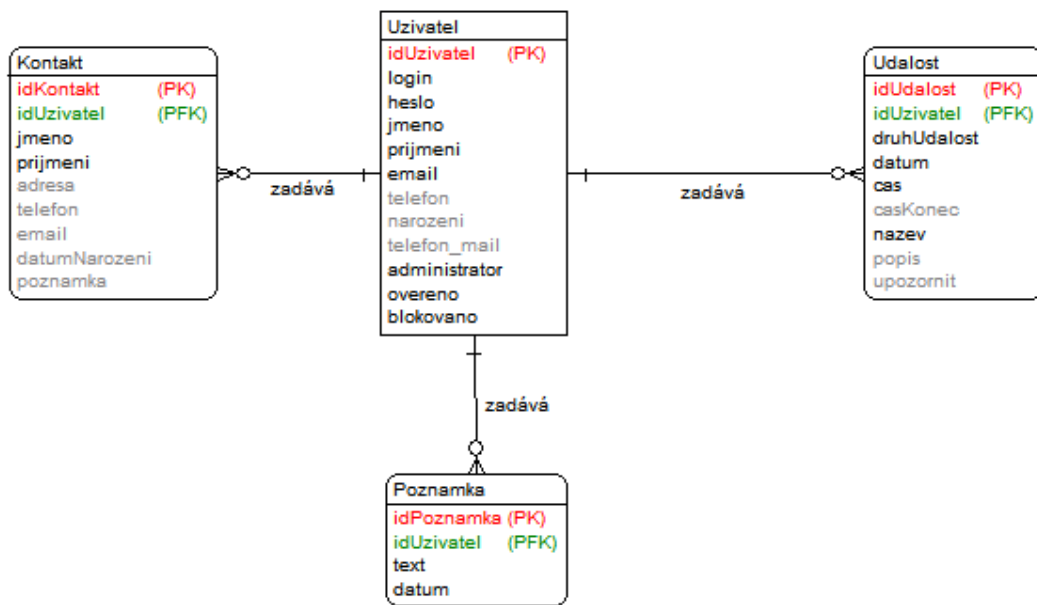
Tento diagram popisuje entity a jejich vzájemné vztahy, ale nepopisuje jaké operace je možno s daty provádět. Podle požadavků popsaných v kapitole 3, jsem vytvořil ER diagram viz obrázek 2.2

V tomto diagramu jsou popsány jednotlivé entity a každá entita má seznam vlastností neboli atributů. Z tohoto diagramu jsem potom vytvořil návrh databáze popsaný v kapitole 2.4.

Tento návrh není příliš složitý, máme zde entitu uživatel, na kterou navazují ostatní entity poznámka, kontakt a událost. Každá entita má vlastní primární klíč označený hodnotou PK, navazující entity pak také cizí klíč FK, kterým se odkazují na entitu uživatel.

2.4 Transformace konceptuálního modelu na tabulky relační databáze

Pro převod ER diagramu, nebo diagramu tříd, se používá standardní proces zvaný normalizace. Proces normalizace nám pomůže odstranit nedostatky chybného návrhu databáze. Cílem normalizace tak je postupné upravování tabulek, aby ve výsledku splňovaly normální formy určitého stupně. Tyto formy jsou popsány v podkapitole 2.4.1.



Obrázek 2.2 – ER diagram

2.4.1 Normální formy

Normální formy zavedl E. F. Codd, aby sloužily k ohodnocení kvality návrhu databázových tabulek. V čím vyšší normální formě tabulky jsou, tím je jejich návrh kvalitnější. Celkem je 6 normálních forem, většinou však postačuje, aby tabulka splňovala 3. normální formu, popřípadě Boyce-Coddovu normální formu.

Popis normálních forem:

- **Nultá normální forma**
Relace je v nulté normální formě právě tehdy, existuje-li alespoň jedno pole, které obsahuje více než jednu hodnotu. Tato forma odpovídá nenormalizovanému modelu.
- **První normální forma**
Relace je v první normální formě právě, když jsou všechny atributy relace atomické, tedy dále nedělitelné.
- **Druhá normální forma**
Relace je v druhé normální formě právě, když je v první normální formě a každý její neklíčový atribut je plně funkčně závislý na attributech primárního klíče.
- **Třetí normální forma**
Relace je v třetí normální formě právě, když je v druhé normální formě a žádný atribut, který není primárním klíčem, není tranzitivně závislý na žádném klíči.

2.5 Popis tabulek

V této kapitole jsou popsány jednotlivé tabulky výsledné databáze. Pokud bereme adresu jako atomickou hodnotu, splňují tyto tabulky třetí normálovou formu popsanou v kapitole 2.4.1.

2.5.1 Uživatel

V této tabulce jsou popsány informace o každém uživateli tohoto systému. Jsou zde informace potřebné pro přihlášení do systému, dále osobní informace o uživateli, kontaktní informace a práva uživatele. Každý uživatel je jednoznačně identifikován pomocí primárního klíče *idUzivatel*.

Pro přihlášení do systému je zde položka *login*, neboli přihlašovací jméno. Toto jméno je jedinečné a slouží k identifikaci uživatele při přihlašování. Dále je zde atribut *heslo*, sloužící k zabezpečení účtu. Toto heslo je v databázi pro vyšší bezpečnost uloženo zakódované pomocí algoritmu MD5. MD5 je jednocestná šifra, což má nevýhodu v případě zapomenutí hesla, které již pak nelze z databáze získat. Výhodou je bezpečnost v případě neoprávněného přístupu do databáze, kdy z MD5 otisku hesla nelze již získat původní heslo.

Dále jsou zde osobní informace uživatele, jako třeba jméno, příjmení, telefon, email, atd. Tyto položky slouží ke komunikaci s daným uživatelem v případě problému, například s uživatelským účtem. Položka *email* slouží také k aktivaci uživatelského účtu a pro zasílání automatických upomínek. Pro zasílání upomínek slouží rovněž položka *mobilní email*, kde jsou tyto upomínky zasílány formou SMS.

Dále jsou zde položky *administrator*, *overeno* a *blokovano*. Tyto položky slouží pro identifikaci práv uživatele, zda je účet aktivován, nebo blokován.

Atributy *login*, *jmeno*, *prijmeni* a *email* mají nastaven index typu FULLTEXT. Tento index je pak použit pro vyhledávání při administraci uživatelů.

2.5.2 Událost

Tato tabulka popisuje jednotlivé události daného uživatele. Každá událost je označena primárním klíčem *idUdalost* a obsahuje cizí klíč *idUzivatelFK*, jenž identifikuje vlastníka této události.

V této tabulce jsou atributy *datum*, *cas* a *casKonec* označující začátek a konec události. Dále jsou zde atributy *nazev* a *popis*, které udávají titulek události a její podrobnější popis.

Poslední atributy jsou *druhUdalost* a *upozornit*, které označují typ dané události a počet hodin před událostí, kdy se má zaslat upomínka.

Pro snazší vyhledávání mají atributy *nazev* a *popis* nastaven FULLTEXT index.

2.5.3 Kontakt

Tabulka popisuje jednotlivé kontakty daného uživatele. Kontakt je identifikován jednoznačným primárním klíčem *idKontakt* a je vázán k tabulce *uzivatel* pomocí cizího klíče *idUzivatelFK*.

Tato tabulka obsahuje pouze atributy obsahující osobní a kontaktní informace o jednotlivých kontaktech. Například jméno, příjmení, email, datum narození atd. Pro snazší vyhledávání je zde opět nastaven FULLTEXT index pro atributy *jmeno*, *prijmeni*, *adresa*, *telefon*, *email* a *poznamka*.

2.5.4 Poznámka

Poslední tabulka obsahuje poznámky uživatelů. Každá poznámka je opět jednoznačně identifikovaná pomocí primárního klíče *idPoznamka* a cizího klíče *idUzivatelFK*.

Poznámka obsahuje pouze atribut *text*, obsahující vlastní text poznámky a dále atribut *datum*, obsahující datum vložení poznámky. Atribut *datum* má rovněž nastaven FULLTEXT index pro vyhledávání.

3 Implementace systému

V této kapitole popíšu podrobněji jednotlivé technologie použité při implementaci systému, dále jejich instalaci a konfiguraci. Dále je zde popsáno uživatelské prostředí této aplikace. Výslednou podobu aplikace je možné nalézt na adrese <http://webplan.php5.cz/>.

3.1 Použité technologie

Při implementaci tohoto systému jsem použil následující technologie.

- webový server Apache verze 2.0.59
- databázový server MySQL verze 5.0.27
- jazyk HTML a kaskádové styly CSS
- skriptovací jazyk PHP verze 5.2.0
- skriptovací jazyk Javascript

3.1.1 Apache

Projekt webového severu Apache vznikl v roce 1993 na Illinoiské univerzitě. První veřejná verze pak byla k dispozici od roku 1995. Od roku 1996 se pak stal tento server nejpoužívanějším na internetu. Tomuto rozšíření jistě napomohla cena, jelikož Apache je zdarma distribuován pod open source licencí. Další výhodou je, že je tento systém multiplatformní, tudíž dostupný pro velkou škálu operačních systémů. Rovněž je používán na většině webhostingových službách, ať už placených, nebo zdarma. Z těchto důvodů jsem si jej také zvolil jako základ pro tento informační systém.

3.1.2 MySQL

MySQL je relační databázový server, vytvořený švédskou firmou MySQL AB v roce 1995. Postupem času se vyvíjel a v dnešní době se řadí mezi nejpoužívanější databázové servery. Tento server je mimo jiné šířen pod GNU licencí a tudíž je k dispozici zdarma za předpokladu, že výsledný projekt bude také splňovat GNU licenci. Velkou výhodou je, že lze tuto databázi používat u většiny webhostingových služeb.

Tento systém byl od počátku stavěn především na rychlost a to za cenu toho, že nepodporoval některé pokročilejší funkce. Postupem času byla ale většina funkcí přidána jako cizí klíče, transakce, nebo poddotazy. Aktuální verze MySQL 5.0 již podporuje také uložené procedury, pohledy, nebo trigger.

Nejpoužívanějším typem tabulek v MySQL jsou tabulky typu MyISAM a InnoDB. Velkou výhodou tabulek typu MyISAM je jejich rychlost. Další výhodou je pak podpora fulltextového

vyhledávání. Nevýhodou je, že nepodporují transakce a cizí klíče. InnoDB je naopak pomalejší, ale podporuje i transakce i cizí klíče. Velkou nevýhodou pak pro mě byla chybějící podpora fulltextového vyhledávání. Podporu cizích klíčů u tabulek typu MyISAM by měla vyřešit MySQL verze 5.2, ovšem tato verze je teprve ve vývoji. Z těchto důvodů jsem si nakonec vybral pro tabulky mé databáze typ MyISAM.

3.1.3 HTML a CSS

3.1.3.1 HTML

HTML je zkratka pro Hyper Text Markup Language, neboli značkovací jazyk pro hypertext. Tento jazyk byl vyvinut na základech jazyka SGML a jeho vývoj začal v roce 1989 ve výzkumném centru CERN. Postupem času měl tento jazyk velký úspěch a roku 1994 bylo založeno mezinárodní konsorcium W3C (World Wide Web Consortium), které mělo dohlížet na sjednocení standardů a rozvoj tohoto jazyka. Poslední verzí jazyka HTML byla verze 4.01, poté byl nahrazen jazykem XHTML.

Jazyk XHTML vznikl s cílem sjednocení zápisu s jazykem XML a aby zároveň byla zachována zpětná kompatibilita. Hlavním rozdílem oproti HTML je, že všechny prvky jazyka jsou párové, a tudíž musí být uzavřeny. Hodnoty atributů musejí být také psány v uvozovkách a jejich názvy psány malými písmeny.

Tento systém je psán v jazyce XHTML 1.0 Strict a všechny stránky byly zkontrolovány a jsou validní, pomocí validačního nástroje na adrese <http://validator.w3.org/>.

3.1.3.2 CSS

CSS je zkratka pro Cascading Style Sheet, neboli tabulky kaskádových stylů. Je to jazyk, který popisuje zobrazení jednotlivých elementů jazyka HTML, XHTML, nebo XML. Hlavní účel CSS je oddělení popisu vzhledu dokumentu od jeho obsahu a struktury. Toto přináší vyšší přehlednost a snadnou změnu designu výsledných stránek. Nevýhodou je pak rozdílná implementace CSS v jednotlivých prohlížečích, především v prohlížeči Internet Explorer. Toto se poté musí různými metodami obcházet.

3.1.4 PHP

PHP je skriptovací programovací jazyk určený především pro tvorbu dynamických webových stránek. Tento jazyk vytvořil v roce 1994 Rasmus Lerdorf pro osobní potřebu na základě skriptovacího jazyka Perlu a jeho přepsáním do jazyka C. Od té doby byl jazyk několikrát přepracován a v současné době je k dispozici již verze 5.2.1.

Syntaxe tohoto jazyka obsahuje prvky různých programovacích jazyků jako je Perl, C, Java a další. Kód jazyka PHP se na rozdíl od jazyka Javascript vykonává na straně serveru. Tam je spuštěn

interpret jazyka PHP, který překládá kód do vnitřního bajtového kódu. Ten je pak následně pomocí Zend Engine vykonáván. Výsledkem je tedy dynamická webová stránka generovaná na serveru. Výhodou tohoto řešení je, že k PHP zdrojovým kódům nemá přístup uživatel dané aplikace. Ten dostane až výsledný vygenerovaný HTML kód.

Tento systém je z velké části napsán pomocí PHP. Jazyk PHP má totiž široké možnosti, například pro obsluhu HTML formulářů, práci s texty, emaily, nebo provázání s databází MySQL. Velkou výhodou je opět open source licence a široké nasazení u hostingových služeb.

3.1.5 Javascript

Javascript je objektově orientovaný skriptovací jazyk, který se používá především jako doplněk v HTML stránkách. Javascript vytvořil Brendan Eich a byl poprvé představen v roce 1995 jako součást prohlížeče Netscape Navigator.

Narozdíl od jazyka PHP je jazyk Javascript přímo součástí HTML kódu dané stránky. Skript se tedy vykonává přímo v prohlížeči daného klienta a ne na serveru jako u PHP. Toto má výhodu v možnostech reagovat na různé události uživatele, jako je například sledování pohybu myši. Nevýhodou je ovšem nemožnost zadávat jakékoli privátní informace, například hesla, z důvodu transparentnosti kódu. Také by přes tento jazyk neměly být řešeny bezpečnostní funkce z důvodu, že zpracování jazyka Javascript lze snadno vypnout. Proto se tento jazyk používá především na řešení navigace, popřípadě kontrolu zadávaných údajů atp.

3.2 Instalace a konfigurace serveru

3.2.1 Instalace komponent

Pro úspěšné zprovoznění systému je potřeba nejdříve nainstalovat jednotlivé komponenty. Instalaci je možno provádět buď ručně, nebo využít nějakého balíku typu vše v jednom. Pro zdárný běh je potřeba nainstalovat Apache web server, dále MySQL server alespoň ve verzi 4.0 a PHP verze 4.0 a vyšší. Dále doporučuji nainstalovat aplikaci phpMyAdmin, kterou je možné nalézt zde <http://www.phpmyadmin.net/>. Tato open source aplikace slouží pro komplexní správu databáze uložené v MySQL.

Z důvodu úspory času při instalaci, mohu doporučit open source projekt WAMP5 <http://www.wampserver.com/en/>. Tento balík obsahuje všechny potřebné aplikace v jejich nejnovějších verzích a usnadňuje tak jejich instalaci a konfiguraci.

3.2.2 Konfigurace PHP

PHP se konfiguruje pomocí textového konfiguračního souboru *php.ini*. Popíšu zde pouze odlišnosti, které jsou potřeba nastavit oproti standardnímu nastavení.

- *short_open_tag = on* Umožňuje používání krátkých otevíracích PHP značek v kódu.
- *max_execution_time = 600* Umožňuje zpracovávat skripty delší dobu, důležité například při hromadném odesílání mailů.
- *display_errors = off* Zabraňuje vypisování chybových zpráv. Což je důležité pouze při ladění systému.
- *magic_quotes_gpc = on* Pro vyšší bezpečnost aplikace při zadávání údajů.
- *SMTP = smtp.server.cz* Zde nastavte adresu Vašeho SMTP serveru. Důležité pro odesílání emailů.
- *sendmail_from = administrator@mail.cz* Zde nastavte emailovou adresu, která se bude zobrazovat v odesílaných emailech.
- *session.use_cookies = 1* Session se bude přenášet jen pomocí cookies. Vyšší bezpečnost.
- *register_globals = Off* Opět vyšší bezpečnost systému. Systém tyto proměnné nepoužívá.

3.2.3 Konfigurace MySQL

MySQL se konfiguruje pomocí souboru *my.ini*. Zde není potřeba měnit standardní konfiguraci. Je však vhodné nastavit hodnotu *ft_min_word_len = 3*. Tato hodnota určuje minimální počet písmen ve slově, které bude zahrnuto do vyhledávání.

Pro snadné načtení tabulek je součástí zdrojových kódů aplikace soubor *databaze.sql*. Tento soubor obsahuje strukturu potřebných tabulek. Stačí jej pouze importovat například v aplikaci phpMyAdmin. Tento soubor již obsahuje vytvořený administrátorský účet. Přihlašovací jméno je *admin* a heslo také *admin*. Tyto údaje doporučuji změnit po prvním přihlášení v nastavení uživatelského profilu.

3.3 Implementace uživatelského rozhraní

3.3.1 Registrace

Protože tuto aplikaci budou používat různí uživatelé, kteří do něj budou zadávat důvěrné informace, bylo potřeba zavést jejich registraci. Registrace probíhá plně v režii uživatele a není zde nutný zásah administrátora.

Snažil jsem se, aby zadávání jednotlivých údajů bylo intuitivní. Všechna zadávaná data jsou kontrolována na správnost pomocí kontrolní třídy. Nesprávně zadaná data jsou pokud je to možné, převedena na správný formát. Na případné chyby v zadaných datech je případně upozorněno.

Pro ověření platnosti emailu a jako ochrana před automatickou registrací do systému, je vygenerován aktivační odkaz a ten je poté zaslán na emailovou adresu uživatele. Účet je tak zablokován, dokud neproběhne toto ověření. V případě problémů má možnost uživatel zaslat aktivační odkaz znovu, nebo požádat o pomoc administrátora.

Při zapomenutí hesla má uživatel možnost nechat si zaslat nové heslo. Toto heslo je náhodně vygenerováno a zasláno na uživatelův email.

3.3.2 Přihlašování

Přihlašování může být ve webových aplikacích řešeno více způsoby. Nejpoužívanější jsou v podstatě dva způsoby a to HTTP autentizace a formulářová autentizace.

HTTP autentizace využívá standardů protokolu HTTP. Uživateli pošle zprávu 401 neautorizovaný přístup a prohlížeč poté vyzve uživatele k zadání přihlašovacích údajů. Výhodou je, že při tomto přihlašování není potřebná podpora cookies a jednodušší správa přihlašování přímo serverem. Jméno a heslo se poté posílají stále s každým požadavkem na server. Nevýhodou je prakticky nemožnost takto přihlášeného uživatele odhlásit.

Pro tento informační systém jsem si zvolil formulářovou autentizaci. Tato autentizace získává přihlašovací údaje z formuláře přímo začleněného do stránky. Následně jsou data odeslána na server a to nejlépe metodou POST. Server poté ověří údaje porovnáním s databází a danému uživateli přidělí session. Tato session obsahuje uživatelské přihlašovací jméno a identifikátor daného uživatele.

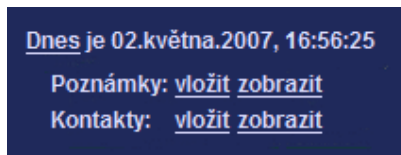
Session je pak kontrolována před každou zabezpečenou stránkou a v případě že nesedí, je načtena jen stránka informující o odhlášení. U tohoto způsobu autentizace je rovněž snadné uživatele odhlásit, v tomto systému je to řešeno metodou zrušení dané session.

3.3.3 Rozhraní uživatele

Po úspěšném přihlášení uživatele do systému se mu zobrazí hlavní stránka aplikace, obsahující navigaci, kalendář a události aktuálního dne.

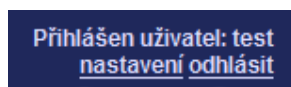
3.3.3.1 Navigace

Hlavička aplikace obsahuje navigaci po jednotlivých částech systému viz obrázek 3.1. V levé horní části je odkaz *dnes* který uživatele kdykoliv přivede na zobrazení událostí aktuálního dne. Dále je zde zobrazen aktuální datum a běžící čas. Tento čas je generován pomocí funkce napsané v Javascriptu. Pod zobrazením času se nachází odkazy pro zadávání a zobrazování kontaktů a poznámek.



Obrázek 3.1 – navigace levá část

Pravá část hlavičky pak obsahuje informaci o přihlášeném uživateli. Dále je zde odkaz vedoucí k administraci uživatelského účtu a odkaz pro odhlášení ze systému.



Obrázek 3.2 – navigace pravá část

3.3.3.2 Hlavní stránka aplikace

Hlavní stránka aplikace (viz obrázek 3.3) se zobrazí po úspěšném přihlášení uživatele do systému. Tato stránka zobrazuje hlavní plánovací funkce.

Levá část obsahuje navigaci, řešenou v podobě měsíčního kalendáře. Kalendář je generován pomocí PHP a zobrazuje aktuální měsíc, týden a rok. Samozřejmostí je přepínání měsíců pomocí šipek. Kliknutím na den v měsíčním kalendáři se zobrazí aktuální události tohoto dne. Tyto události se pak zobrazují v pravé části stránky.

Pod měsíčním kalendářem je zobrazen rychlý přehled pro aktuálně zvolený týden. Jsou zde zobrazena jména svátků a ikony označující události v jednotlivých dnech.

Pravá část pak slouží k zobrazení, zadávání, mazání a editaci událostí. V levé horní části je formulář umožňující zobrazení všech událostí mezi zadanými daty. V pravé horní části pak formulář pro vyhledávání událostí dle názvu nebo popisu. Pod tímto formulářem je odkaz *zobrazit vše*, umožňující zobrazit všechny zadané události daného uživatele.

Dále je zde tabulka zobrazující události. Událost je možno editovat kliknutím na ní, popřípadě označit, nebo smazat. Pod událostí je editační formulář. Pokud je kliknuto na událost v tabulce, je pomocí Javascriptu vyplněn a slouží pro editaci. Prázdný pak slouží pro zadávání nové události.

The screenshot shows a web application interface for task management. On the left, there is a navigation bar with a calendar for the month of May 2007, highlighting the current date, Tuesday, May 15. Below the calendar is a 'Rychlý přehled pro daný týden:' (Quick overview for the given week) section, showing a vertical list of dates from 14 to 20 with corresponding icons and names (e.g., 14. Po Bonifác, 15. út Žofie, 16. St Přemysl, 17. Čt Aneta, 18. Pá Nataša, 19. So Ivo, 20. Ne Zbyšek).

The main content area is titled 'plán na úterý 15.května.2007' (plan for Tuesday, May 15, 2007). At the top, there are search filters for 'Od: 03.5.2007' and 'Do: 03.5.2007', along with buttons for 'zobrazit', 'vyhledat', 'zobrazit vše', and 'zobrazit aktuální den'. Below this is a table of tasks:

čas od	čas do	upozornit	typ	název	popis
12:00:00	00:00:00	12	důležité	Odevzdání BP	poslední možnost odevzdání bakalářské práce
20:00:00	24:00:00	0	schůzka	Oslava	oslava úspěšného ukončení BP

Below the table, it shows 'Počet akcí: 2' and a button 'Označené odstranit'. At the bottom, there is a form for adding or editing tasks with fields for 'čas od:', 'do:', 'název:', 'popis:', and 'upozornit:'. The 'typ:' field has a list of radio buttons: ostatní, schůzka, důležité, telefonát, škola, lékař, and narozeniny. There are also 'Uložit' and 'Reset' buttons.

Obrázek 3.3 – hlavní stránka aplikace

3.3.3.3 Poznámky

Tento oddíl systému slouží k zadávání a zobrazování poznámek. Poznámky jsou zadávány a zobrazovány do formulářového okna typu textarea. Tento způsob byl zvolen z důvodu zachování formátování poznámky.

Poznámky jsou zobrazeny dle data vložení a je možno je mazat a tisknout. Samozřejmostí je pak fulltextové vyhledávání v textech poznámek.

3.3.3.4 Kontakty

Tato sekce slouží k zadávání zobrazování a editaci kontaktů. Kontakty jsou vypisovány do tabulky a je možno v nich také vyhledávat. Dále je zde možnost z označených kontaktů vytvořit tiskovou sestavu.

3.3.3.5 Nastavení uživatelského účtu

Tato část systému slouží ke změně uživatelských údajů. Uživatel má zde možnost veškeré změnit údaje, které zadal při registraci, kromě přihlašovacího jména.

3.3.4 Rozhraní administrátora

Administrátor je speciální typ uživatele s administrátorskými právy. Má tudíž stejné možnosti pro práci se systémem jako uživatel, popsán v kapitole 3.3.3. Navíc má právo dohlížet na uživatele a spravovat jejich účty.

Po přihlášení administrátora se v hlavičce aplikace zobrazí odkaz *administrátorské menu*. Toto menu obsahuje nástroje pro administraci uživatelů viz obrázek 3.4. Administrátor tak má právo zablokovat, nebo odblokovat libovolného uživatele. Dále potvrdit uživateli registraci v případě problémů, nebo zaregistrovat nového administrátora systému. Také má možnost povýšit uživatele na administrátora, popřípadě tato práva odebrat.

Samozřejmostí je opět vyhledávání uživatelů dle přihlašovacího jména, příjmení, jména, nebo emailu.

login	příjmení	jméno	email	ověřen	blokován	administrátor	
admin	Mátl	Tomáš	xmatit00@stud.fit.vutbr.cz	ano	ne	ano	<input type="checkbox"/>
novak	Novák	Jan	efx@atlas.cz	ne	ne	ne	<input type="checkbox"/>
redl	Redl	František	efrax@centrum.cz	ano	ne	ano	<input type="checkbox"/>
test	Testovič	Testor	efrax@tiscali.cz	ano	ne	ne	<input type="checkbox"/>

Označené uživatele:

Obrázek 3.4 – administrátorské menu

3.3.5 Automatické zasílání upomínek

O automatické zasílání upomínek se stará stránka *upominky.php*. Tato stránka po otevření kontroluje databázi a zasílá upomínky dle nastaveného času. Pro spouštění této stránky je vhodné využít program Cron, pokud to hosting umožňuje. Další možností pro pravidelné spouštění této stránky je možnost využít webovou službu WebCron na adrese <http://www.webcron.org/>.

3.3.6 Vzhled aplikace

Vzhled aplikace je definován pomocí CSS. CSS má výhodu pro snadnou změnu prakticky všech prvků na stránce.

Barevné schéma je laděno do nepříliš výrazné modré barvy. Barevné schéma bylo zvoleno pokud možno ne moc pestré, aby neodrazovalo uživatele aplikace. Vzhled aplikace byl vytvořen pro rozlišení obrazovky 800x600 a vyšší.

3.4 Problémy při implementaci

Při implementaci systému jsem se setkal s několika menšími problémy, které popíšu v této kapitole.

Jedním z problémů byla složitá kontrola zadávaných údajů. Údajů se zadává do aplikace velké množství ať už při registraci, nebo zadávání kontaktů či událostí. Tyto údaje bylo potřeba kontrolovat a upravovat jejich formát. Toto jsem nakonec vyřešil kontrolní třídou napsanou v PHP. Třída dostane pole se vstupními údaji a pole s parametry, které je třeba kontrolovat, popřípadě převádět. Tyto údaje třída zpracuje a vrátí upravené pole výsledků, nebo pole s popisem chyb v zadání.

Další problémy vyplývaly z přirozené vlastnosti protokolu HTTP a to jeho bezstavovosti. To znamená, že protokol si nepamatuje údaje, dotazy, formuláře, které uživatel zadal v minulosti. Pamatuje si pouze aktuálně odesílanou informaci na server. Tento problém se nejvíce projevoval například při označení více položek a následné práci s nimi. Problém jsem vyřešil v odesílání takovýchto méně významných informací pomocí pole GET jako součást HTTP adresy.

Dalším problémem bylo sjednocení grafického designu aplikace. Aplikace byla od počátku psána jako validní XHTML 1.0 Strict kód a to včetně validního CSS kódu. Se zobrazením na prohlížečích Firefox, nebo Opera nebyl žádný problém. Webový prohlížeč Internet Explorer 6.0 má však odlišnou interpretaci CSS kódu a tak zobrazil design stránek rozdílně. Problém jsem vyřešil úpravami v CSS kódu speciálně pro Internet Explorer.

4 Náměty na rozšíření systému

Systém v aktuální podobě zvládá v podstatě všechny důležité funkce, které lze od plánovacího systému očekávat. Byly splněny všechny požadavky stanovené při návrhu v kapitole 1.1. Při implementaci mě napadla různá rozšíření systému, která popíši v této kapitole.

4.1 Zabezpečení systému pomocí HTTPS

V systému se pracuje s velkým množstvím citlivých údajů zadaných uživateli. Proto pro jejich vyšší ochranu by mohl být použit protokol HTTPS namísto protokolu HTTP.

HTTPS je nadstavba protokolu HTTP popsaného v kapitole 3.1.3. Narozdíl od HTTP však HTTPS přenáší veškerá data šifrovaně pomocí SSL popřípadě TLS. Tato data jsou pak zabezpečena před odposloucháváním a dalšími typy útoku.

4.2 Rozšíření funkcí

Systém by také mohl obsahovat více různých funkcí. Například pro použití pro manažerské potřeby by mohl obsahovat plánování dovolené, plánování projektů, nebo služebních cest. Dále by zde mohlo být oddělení pracovních a osobních událostí, kontaktů a poznámek. Student používající tento systém by také mohl uvítat funkci pro školní rozvrh hodin. Dalším rozšířením funkcí by mohlo být například odesílání informací mezi jednotlivými uživateli aplikace.

4.3 Import a export

V současné podobě systém umožňuje exportovat nebo importovat data pouze hlavnímu administrátorovi. Ten tak může například pomocí aplikace PHPMyAdmin exportovat, nebo importovat celou databázi, popřípadě jednotlivé tabulky.

Rozšířením systému by mohla být podpora pro export a import jednotlivých dat uživatelem aplikace. Ten by pak měl možnost ukládat a načítat jednotlivé poznámky, události nebo kontakty do zvoleného formátu. Vhodným formátem pro tyto data by mohlo být například XML.

4.4 Instalace a konfigurace

Pro jednodušší zprovoznění systému by mohl být vytvořen systém instalace a konfigurace. V současné podobě je nutno například importovat databázi a nastavovat jednotlivé konfigurační soubory ručně viz kapitola 3.2. Toto by tak mohl obstarávat systém naprogramovaný v PHP.

Například by automaticky vytvořil při prvním spuštění tabulky v databázi, administrátorský účet, nakonfiguroval jednotlivé parametry služeb atd.

5 Závěr

Na závěr bych se rád zmínil o splnění zadání jednotlivých bodů této bakalářské práce a zhodnotil výsledek této práce.

Zpočátku práce byla provedena podrobná specifikace všech požadavků a funkcí důležitých pro plánovací systém. Analýza těchto požadavků pak byla podrobně popsána v kapitole 1.2. Dle počáteční specifikace jsem pak připravil UML diagramy a to konkrétně diagram případů užití a ER diagram.

Další částí bakalářské práce pak bylo navrhnout strukturu databáze. Návrh databáze samozřejmě vycházel z předcházející analýzy požadavků a to především z ER diagramu. Výslednou strukturu databáze jsem pak kompletně popsal v kapitole 2.5.

Poslední částí pak byla samotná implementace systému. Při implementaci jsem postupoval dle návrhu a její výsledky pak detailně popsal v kapitole 3. Tato fáze bakalářské práce byla nejvíce časově náročná.

Systém byl nakonec uveden do plně funkčního stavu a byly splněny veškeré specifikované požadavky. Při implementaci mě napadaly různé náměty na rozšíření systému, na které už však při implementaci nezbyl čas. Tyto náměty jsem pak popsal v kapitole 4.

Výslednou podobu systému je možno nalézt na adrese <http://weblan.php5.net/>. Tento systém pak byl testován na skupině uživatelů s různou zkušeností při ovládání počítače. Při zpětné vazbě pak nebyly zaznamenány žádné větší problémy při obsluhování systému, nebo s jeho funkčností.

Na konec bych se rád zmínil o přínosu této bakalářské práce. Na začátku této práce jsem v podstatě neměl žádné velké zkušenosti s návrhem a implementací webové aplikace. Během měsíců práce na tomto systému jsem se však hodně nového naučil. Rozšířil jsem si znalosti především o skriptovacím jazyku PHP, který jsem začal používat ve velkém až při této práci. Proto bych rád prohlásil, že tato práce pro mě byla velkým přínosem a určitě nabyté poznatky využiji při návrhu případných dalších webových aplikací.

Literatura

- [1] Ullman, Larry, PHP a MySQL. Computer Press, 2004. ISBN 80-251-0063-4.
- [2] J. Zendulka, přednášky k předmětu Databázové systémy. Dostupné na adrese <https://www.fit.vutbr.cz/study/courses/IDS/private/>
- [3] R. Burgert, přednášky k předmětu Tvorba webových stránek. Dostupné na adrese <https://www.fit.vutbr.cz/study/courses/ITW/private/>
- [4] články na serveru <http://www.jakpsatweb.cz/>
- [5] články na serveru <http://www.interval.cz/>
- [6] články na serveru <http://www.root.cz/>
- [7] články na serveru <http://www.wikipedia.org/>

Seznam příloh

Příloha 1. CDROM se zdrojovými kódy aplikace včetně elektronické nápovědy