



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

INFORMAČNÍ SYSTÉM SKUPINY ZPRACOVÁNÍ PŘIROZENÉHO JAZYKA

NLPGROUP INFORMATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MILAN SEITLER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV DYTRYCH

BRNO 2013

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací rozšíření pro informační systém KNOTIS. Výsledkem práce je funkční nástroj pro tvorbu výkazů z programů pro sledování času a nová verze automatického plánovače.

Abstract

This bachelor thesis deals with analysis and implementation of extensions for information system KNOTIS. As its result there has been created tool for importing work reports from time tracking software and new version of automatic work scheduler.

Klíčová slova

Informační systém, programy pro sledování času, pracovní výkazy, pracovní plán, plánovač, web, HTML, CSS, PHP, Javascript.

Keywords

Information system, time tracking software, work reports, work schedule, scheduler, web, HTML, CSS, PHP, Javascript.

Citace

Milan Seitler: Informační systém Skupiny zpracování přirozeného jazyka, bakalářská práce, Brno, FIT VUT v Brně, 2013

Informační systém Skupiny zpracování přirozeného jazyka

Prohlášení

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

.....

Milan Seitler
10. května 2013

Poděkování

Rád bych poděkoval Ing. Jaroslavu Dytrychovi za aktivní přístup k vedení této práce.

© Milan Seitler, 2013.

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

1 Úvod	3
2 Informační systém Skupiny zpracování přirozeného jazyka	4
2.1 Skupina zpracování přirozeného jazyka	4
2.2 Informační systém KNOTIS	4
2.2.1 Denní a týdenní výkazy	5
2.2.2 Plán práce	6
3 Analýza požadavků	8
3.1 Vykazování práce	8
3.1.1 Programy pro sledování času	8
3.2 Plánování práce	9
3.3 Další požadavky	9
4 Použité technologie	10
4.1 HTML	10
4.2 CSS	10
4.3 DOM	11
4.4 Javascript	11
4.5 jQuery	11
4.6 PHP	11
4.7 MySQL	12
4.8 CSV	12
5 Návrh	13
5.1 Tvorba výkazů	13
5.1.1 Výběr souboru	13
5.1.2 Zpracování souboru	14
5.1.3 Úprava denních výkazů	14
5.1.4 Tvorba týdenních výkazů	15
5.1.5 Detekce duplicitních výkazů	16
5.2 Automatický plánovač	16
5.2.1 Rozdělení do období	16
5.2.2 Režimy plánovače	17
5.2.3 Situace při plánování	18
5.2.4 Činnost plánovače	20
5.3 Další vylepšení systému	23

6 Implementace	24
6.1 Import výkazů	24
6.1.1 Zpracování vstupního souboru	24
6.1.2 Seznam denních výkazů	24
6.1.3 Týdenní výkazy	25
6.1.4 Označení duplicitních výkazů	25
6.2 Automatický plánovač	27
6.3 Použití apostrofu	27
7 Testování	29
7.1 Import - data obsahující chyby	29
7.2 Plánovač – plán nového uživatele	30
7.3 Plánovač – režim přepřelánování budoucnosti	31
7.4 Napadení systému pomocí SQL Injection	32
8 Závěr	33
Literatura	34
Přílohy	37
Seznam příloh	38
A Obsah CD	39

Kapitola 1

Úvod

Tématem bakalářské práce je analýza a rozšíření informačního systému Skupiny zpracování přirozeného jazyka, především pak v oblasti vykazování a plánování práce.

Ve výzkumné skupině jsem pracoval více než rok a informační systém jsem pravidelně používal, rozhodl jsem se tedy pro toto téma, abych se blíže seznámil s vnitřní strukturou a prvky systému. O činnosti skupiny a úkolech informačního systému KNOTIS pojednává kapitola **Informační systém Skupiny zpracování přirozeného jazyka**.

Požadavkům na rozšíření systému KNOTIS jsem se věnoval v kapitole **Analýza požadavků**. Z této analýzy vyplynuly vhodné technologie, jejichž přiblížení je obsahem kapitoly **Použité technologie**.

Návrh řešení v kapitole **Návrh** analyzuje jednotlivá rozšíření KNOTIS a obsahuje grafický návrh uživatelského rozhraní. Zabývá se také problémovými situacemi, na které systém musí umět reagovat.

Programová část řešení je popsána v kapitole **Implementace**, ukázky testů jsou pak předmětem kapitoly **Testování**.

Obsahem poslední kapitoly **Závěr** je shrnutí výsledků celé bakalářské práce.

Kapitola 2

Informační systém Skupiny zpracování přirozeného jazyka

2.1 Skupina zpracování přirozeného jazyka

Skupina se během listopadu roku 2013 přejmenovala na Skupinu znalostních technologií [3], dále ji tedy v textu budu označovat zkratkou KNOT. Je jednou z řady výzkumných skupin působících na Fakultě informačních technologií VUT v Brně. KNOT se zaměřuje na zpracování přirozeného jazyka, mezi typické činnosti patří získávání informací z textových dat o velkém objemu či práce se slovníky.

Skupina se zapojuje také do mezinárodních projektů, na kterých spolupracují přední evropské univerzity a instituce.

2.2 Informační systém KNOTIS

Mezi hlavní úkoly informačního systému skupiny KNOT patří:

- správa jednotlivých projektů
- evidence pracovníků KNOT
- pracovní výkazy jednotlivých osob
- automatická e-mailová upozornění
- správa prostředků na serverech KNOT

Vedoucí projektů mohou prostřednictvím KNOTIS sledovat průběh prací. Každý pracovník je povinen informovat o své činnosti na projektech prostřednictvím denních a týdenních výkazů. KNOTIS pro každou osobu umožňuje vytvořit plán práce, ve kterém lze nastavit předpokládaný počet odpracovaných hodin v budoucích týdnech.

Systém vyhodnocuje data získaná z výkazů a pracovního plánu a podle nich pak označuje tzv. „hříšníky“, tedy osoby, které nepracují či neodesílají výkazy. V těchto případech KNOTIS automaticky příslušným vedoucím a pracovníkům odesílá upozornění e-mailem. E-maily se posílají také například při vytvoření výkazu či změně pracovního plánu.

Projekt: 415. OPaBP: Plánování a vykazování práce v NLPIS *

Týden: 16. (15.04.2013 - 21.04.2013) * předminulý minulý současný Nastavit

Zadejte informace o novém denním výkazu:

Projekt: 415. OPaBP: Plánování a vykazování práce v NLPIS

Datum: 16.04.2013 * **Čas:** od [] do 13:03 **Odpracováno:** 0 : 0 *







Činnost:

Činnost nesouvisí přímo s projektem:

Uložit Zpět

Obrázek 2.1: Rozhraní pro vytvoření denního výkazu.

Pracovní výkazy:

Datum	Hodin	Od	Do	Činnost	Operace	SSP
31.03.2013	1:20			Plánovač - úpravy	 	A
26.03.2013	1:30			Konzultace	 	A
25.03.2013	6:30			Plánovač - vytvoření nového plánovače	 	A
Celkem	9:20					

Obrázek 2.2: Přehled denních výkazů.

2.2.1 Denní a týdenní výkazy

Základním typem výkazu je denní výkaz (obrázky 2.1 a 2.2). Ten obsahuje jednotlivé činnosti společně s jejich datem a časem. Denní výkazy se seskupují do jednoho týdne, jehož souhrn pracovník provede v týdenním výkazu (obrázek 2.3). Zde popíše celkové práce na projektu za uplynulý týden, plán do dalšího týdne a případné problémy či omluvy. Stránka s rozhraním pro správu výkazů umožňuje nastavit týden a projekt, podle kterého se řídí zobrazení a úprava výkazů. Je zde možné přidávat a upravovat denní i týdenní výkazy zároveň. Vytvořené denní výkazy lze přearadit k jinému projektu.

Týdenní výkaz:

Projekt: 415. OPaBP: Plánování a vykazování práce v NLPIS

Týden: 13. (25.03.2013 - 31.03.2013)

Souhrn:

Plán:

Problémy a nejasnosti: (odesláno)

Omluvy a výmluvy:

Obrázek 2.3: Rozhraní pro vytvoření týdenního výkazu.

2.2.2 Plán práce

Každý pracovník má zadán celkový počet hodin k odpracování spolu s termínem, do kdy je třeba hodiny odpracovat. Tyto hodiny si v rozhraní plánu práce (obrázek 2.4) rozdělí do jednotlivých týdnů. Součástí systému je i automatický plánovač, který rozdělení hodin provádí například při vytvoření nového pracovníka či změně počtu hodin k odpracování. Plánovač rozlišuje mezi zkouškovým obdobím zimního semestru a zbytkem roku. Pro jednotlivé typy pracovníků (student bakalářského programu, doktorand, zaměstnanec a další) jsou týdenní konstanty přednastaveny ve zdrojovém kódu, na základě tohoto nastavení plánovač postupně do každého týdne vkládá počet hodin k odpracování. Plánovač kontroluje uložené výkazy a do počtu hodin, které rozděluje, zahrnuje i neodpracované či nadpracované hodiny z minulosti.

Plán práce:

Číslo týdne	Pondělí	Neděle	Plánováno hodin	Zbude hodin
16	15.04.2013	21.04.2013	11 : 25 *	20:55
17	22.04.2013	28.04.2013	11 : 9 *	9:46
18	29.04.2013	05.05.2013	9 : 46 *	0:00
19	06.05.2013	12.05.2013	0 : 0 *	0:00
20	13.05.2013	19.05.2013	0 : 0 *	0:00
21	20.05.2013	26.05.2013	0 : 0 *	0:00
22	27.05.2013	02.06.2013	0 : 0 *	0:00
Celkem			32:20	0:00

[Uložit změny](#)[Zpět](#)**Historie plánu práce:**

Číslo týdne	Pondělí	Neděle	Plánováno hodin	Odpracováno hodin	Rozdíl	Zbývalo
4	24.01.2011	30.01.2011	5:4	0:00	5:04	521:00
5	31.01.2011	06.02.2011	5:38	5:39	-0:01	515:21
6	07.02.2011	13.02.2011	2:0	2:00	0:00	513:21
7	14.02.2011	20.02.2011	5:52	4:30	1:22	508:51
8	21.02.2011	27.02.2011	6:8	0:00	6:08	508:51

Obrázek 2.4: Rozhraní pracovního plánu.

Kapitola 3

Analýza požadavků

3.1 Vykazování práce

Při práci na počítači používá řada lidí programy pro sledování času (viz kapitola 3.1.1). Rozšíření KNOTIS nabídne pracovníkům možnost importu dat z programů pro sledování času ve formátu CSV. Rozšíření bude podporovat různou strukturu dat a umožní z importovaných údajů vytváření denních a týdenních výkazů se stejnými možnostmi, jako při vykazování přes klasické rozhraní. Ukázka CSV souboru je obsahem výpisu 3.1.

```
1 "Date","Start Time","End Time","Minutes","Title"
2 "1/3/13","14:00","15:00","60","IBP - návrh tříd"
3 "1/3/13","16:00","17:20","80","IBP - jednotkové testy"
4 "1/4/13","10:30","12:00","90","rss2 - úprava repozitáře"
5 "1/10/13","08:10","08:30","20","rss2 - tvorba výkazu"
```

Výpis 3.1: Ukázka CSV souboru

3.1.1 Programy pro sledování času

Tyto programy sledují a zaznamenávají činnost prováděnou na počítači a umožňují zjistit, kolik času uživatel strávil s jednotlivými programy či webovými stránkami. Jednotlivé programy se od sebe odlišují mimo jiné i mírou automatizace. Například webový nástroj Rescue Time ¹ dokáže sledovat veškeré aktivity prováděné na počítači včetně stránek zobrazených v internetovém prohlížeči. Jednotlivým činnostem automaticky přiřazuje ohodnocení jejich produktivity od -2 do +2 (ohodnocení lze ručně upravit). Na základě těchto údajů pak poskytuje grafy a statistiky produktivity, viz obrázek 3.1. Existují i komplexnější programy, například český Inspectime ². Ten umožňuje sdružovat jednotlivé činnosti do projektů a nabízí také správu týmových aktivit. Činnosti jsou zde zadávány ručně, opět jsou k dispozici grafy a statistiky.

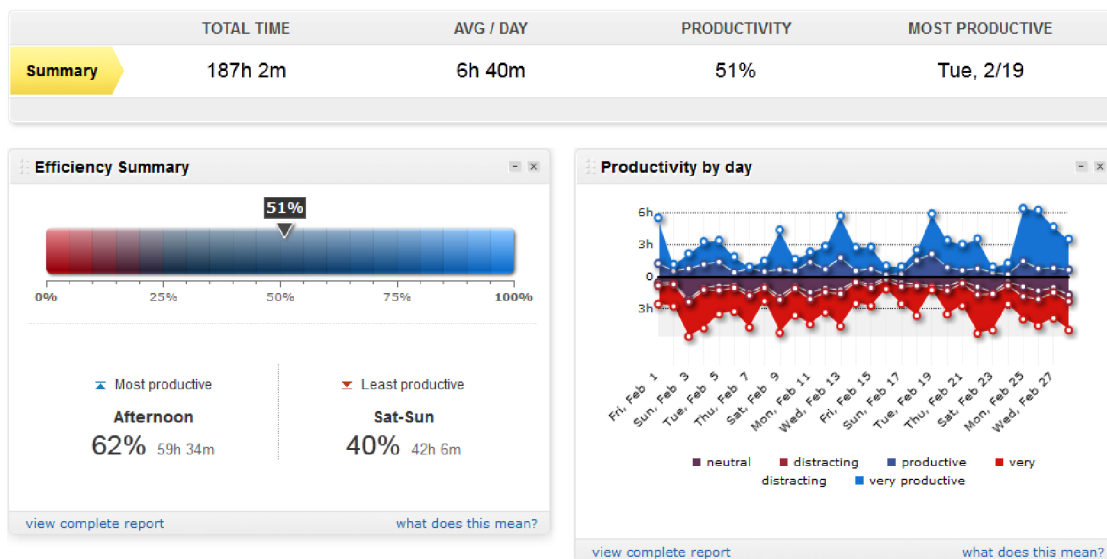
Nástroje pro sledování času obvykle kromě grafického rozhraní poskytují také výstup do souboru, například ve formátech XLS ³, XML ⁴ nebo CSV (viz 4.8).

¹<http://rescuetime.com>

²<http://inspectime.com>

³Přípona souborů vytvořených v MS Excel.

⁴Extensible Markup Language - značkovací jazyk používaný pro serializaci dat.



Obrázek 3.1: Statistika programu Rescue Time.

3.2 Plánování práce

Dalším vylepšením informačního systému KNOTIS bude nová verze automatického plánovače. Nejpodstatnější změnou bude možnost rozdělit rok do libovolného počtu období tak, aby v každém období bylo možné určit jiný základní počet hodin k odpracování týdně. Bude tak možné rozlišit například mezi zkouškovým obdobím a běžným semestrem, kdy lze očekávat, že člověk bude schopen odpracovat více hodin. Nová verze umožní uživatelům automatickou změnu plánu bez nutnosti měnit všechny položky ručně. Součástí plánovače je také automatické odesílání e-mailových upozornění o změně plánu.

3.3 Další požadavky

V rámci analýzy KNOTIS jsem zjišťoval, jaké funkce systém poskytuje, a zkoušel jsem práci s jednotlivými stránkami systému, především s rozhraním pro správu výkazů a rozhraním pro úpravu pracovního plánu. Mimo funkcí souvisejících s novými rozšířeními jsem se zaměřil také na uživatelskou přívětivost a použitelnost systému. Narazil jsem tak na několik problémů, které uživatelům komplikují práci. Při pokusu zadat do výkazu text v anglickém jazyce jsem zjistil, že systém nepovoluje použití apostrofu. Také se ukázalo, že uživatel má možnost přesunout denní výkaz k jinému projektu, nelze jej však přesunout do jiného týdne. Při zadání špatného datumu tak bylo potřeba celý výkaz odstranit a vytvořit znovu. Na další problém jsem narazil při úpravě denního výkazu, který zasahoval do dvou dní (začínal před půlnocí a končil následující den). Když jsem změnil počáteční nebo koncový čas, datum výkazu se posunulo o den zpět.

Z pohledu pracovníka se mi tyto situace jevily jako zdržující a rozhodl jsem se je konzultovat s dalšími uživateli systému. Z diskuze vyplynulo, že by uvítali zejména využití apostrofů a možnost přesunout výkaz do jiného týdne. Mezi cíle bakalářské práce jsem tak zařadil i odstranění těchto problémů.

Kapitola 4

Použité technologie

4.1 HTML

HTML (HyperText Markup Language) je značkovací jazyk používaný pro tvorbu webových stránek. Jazyk HTML je tvořen množinou značek, které jsou většinou párové a uzavírají mezi sebou části zdrojového textu. Některé značky mohou být nepárové, v tom případě nemají obsah a slouží například k vložení obrázků či vytváření formulářů. Každá značka může obsahovat atributy, například unikátní identifikátor značky či titulek, který se zobrazí po najetí myši na prvek uvozený značkou.

První verze jazyka vznikla v roce 1991 v návaznosti na požadavky vědců, kteří chtěli sdílet výsledky svých výzkumů. S šířením internetu bylo nutné unifikovat tvorbu HTML dokumentů, vznikly tak standardy HTML 2.0 [12] a HTML 3.0 [24]. Na dlouhou dobu posledním standardem byl HTML 4.0 [25], potažmo vylepšená verze 4.01 [26]. Tuto verzi používají tvůrci webových stránek dodnes. V současnosti vzniká nová verze standardu, označovaná jako HTML 5 [11], která přináší nové dynamické prvky a mj. také podporu zvuku a videa.

4.2 CSS

Zkratka CSS v překladu označuje Kaskádové styly, což je jazyk, který slouží k popisu vzhledu dokumentů vytvořených značkovacími jazyky. Může být tedy použit nejen s HTML ale například i s XML. Slovo „kaskádové“ označuje vrstvení jednotlivých stylů, kde prvek může být popsán více pravidly, platí však vždy to poslední uvedené (není-li jinému přiřazena vyšší priorita).

Jednotlivé prvky dokumentu jsou v CSS vybírány prostřednictvím tzv. selektorů, získat lze například všechny prvky určitého typu či prvky označené třídou, případně pouze jeden prvek pomocí jeho identifikátoru. Takto vybraným prvkům pak lze přiřazovat jednotlivé vlastnosti, tedy barvu, velikost a řez písma, pozici prvku, rozměry prvku a další.

První verze CSS 1 [22] byla vydána v roce 1996, aktuálně je oficiálně doporučenou verzí CSS 2.1 [13]. Ve vývoji je verze 3 [1], která prozatím přinesla nové selektory či rozšířené možnosti stylování blokových prvků. I přes standardizaci CSS je však problémem kompatibilita mezi prohlížeči, kde zatím stále nelze dosáhnout univerzálního CSS zápisu, který by byl zobrazován ve všech prohlížečích stejně.

4.3 DOM

DOM (Document Object Model [20]) je aplikační programové rozhraní, které slouží k dynamickému přístupu k dokumentu a jeho modifikaci. DOM využívá stromové struktury dokumentu a umožňuje tak přístup k jednotlivým prvkům dokumentu jako k objektům. S pomocí DOM lze také vkládat nové objekty. Specifikace DOM se dělí do několika úrovní, kde každá další úroveň postupně rozšiřuje úroveň předchozí. Aktuálně nejvyšší úrovní je DOM Level 3 [21] (podporováno většinou prohlížečů).

4.4 Javascript

Javascript (JS [16]) je interpretovaný, objektově orientovaný jazyk, který původně vznikl jako součást webových prohlížečů. JavaScript je standardizován jako ECMAScript, nejnovější verze 1.8.5 odpovídá ES verze 5 [9]. Jazyk je dynamicky a slabě typovaný a jeho syntaxe je podobná jazyku C. Skripty v JS se spouštějí na straně klienta a umožňují vytváření dynamických webových stránek. Objektová orientace zde není realizována pomocí tříd, nýbrž tzv. prototypováním [4]. S tím souvisí také odlišný princip dědičnosti. Nejnovější verzí jazyka je verze 1.8.5.

Javascript umožňuje prostřednictvím DOM měnit dokument po jeho vytvoření, nabízí například možnost animovat jednotlivé prvky či vytvářet dialogová okna. Uživatel se také může často setkat s JS validací webových formulářů. Javascript umožňuje odesílání informací na server bez nutnosti obnovení stránky, tato technologie je souhrnně označována zkratkou AJAX [17] [27].

4.5 jQuery

jQuery [5] [14] je knihovna pro Javascript, která zjednodušuje syntaxi a urychluje tvorbu skriptů. Umožňuje snadněji pracovat s událostmi prvků, aniž by bylo potřeba chování specifikovat přímo v HTML zápisu prvku. jQuery pro výběr prvků používá podobné selektory jako CSS, oproti tomuto jazyku jsou však možnosti jQuery širší. Knihovna je také využívána k tvorbě různých efektů a animací. jQuery je v neustálém vývoji, během roku vychází několik nových verzí (aktuálně 2.0.), je však nutné důsledně testovat kompatibilitu jednotlivých vydání.

Rozšířením základního jQuery je jQuery UI [6], které usnadňuje vytváření uživatelských rozhraní a poskytuje předpřipravené moduly jako například kalendář či posuvná okna.

4.6 PHP

PHP (PHP Hypertext Preprocessor [8] [19]) je programovací jazyk používaný zejména pro tvorbu dynamických webových stránek. Lze v něm vyvíjet například i konzolové aplikace. Tento skriptovací jazyk je interpretovaný a jednotlivé skripty se provádí na straně serveru. PHP skript vytváří výslednou stránku, která se zobrazí uživateli, velmi časté je propojení s jazykem HTML. Poslední vydanou verzí je PHP 5.4, existuje také zkušební verze 5.5.

Syntaxe jazyka je podobná jazyku C, umožňuje použití třídního objektově orientovaného programování. Jazyk PHP je dynamicky typovaný. Výhodou PHP je jeho rozšířenost,

je podporován většinou hostingových služeb. PHP skripty často spolupracují s databázovými systémy, podporovány jsou například databáze MySQL, PostgreSQL [2] či Firebird [29].

4.7 MySQL

MySQL [7] je relační databázový systém využívající jazyk SQL ¹. MySQL je, podobně jako PHP, podporován většinou hostingových služeb. MySQL je multiplatformní a podporuje řadu programovacích jazyků, mezi nejčastěji používané patří PHP, Python či Java. Poslední verzí systému je MySQL 5.6.

4.8 CSV

Zkratka CSV (Comma separated values [28]) označuje souborový formát, který reprezentuje tabulková data. CSV soubor obsahuje jednotlivé řádky s hodnotami oddělenými čárkou. Toto rozdělení značí jednotlivé sloupce tabulky. Sloupce mohou být odděleny i jinými znaky, například středníkem či tabulátorem. Soubory ve formátu CSV jsou používány k výměně informací mezi různými zdroji, řada programů podporuje import a export ve formátu CSV. Výhodou tohoto formátu je jednoduché zpracování, pro které lze často použít knihovny daného jazyka.

¹Structured Query Language - dotazovací jazyk používaný v relačních databázích

Kapitola 5

Návrh

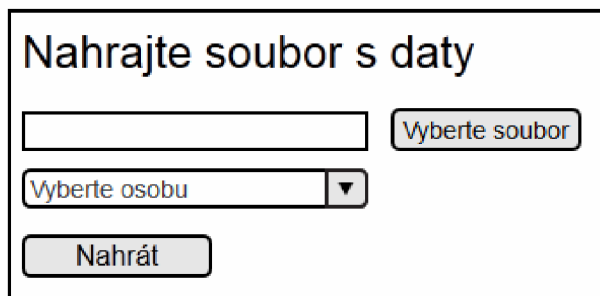
5.1 Tvorba výkazů

Postup, jakým lze z dat programů pro sledování času vytvořit výkazy, je rozdělen do tří kroků:

1. Výběr a odeslání souboru s daty
2. Zpracování souboru a úprava jednotlivých denních výkazů, přiřazení výkazů k týdnům a projektům
3. Doplnění týdenních výkazů

5.1.1 Výběr souboru

Na obrázku 5.1 je zobrazen návrh rozhraní pro první krok importu. Kromě výběru souboru je zde také výběr osoby, který umožňuje vedoucím importovat výkazy zaslané e-mailem. Seznam všech pracovníků se zobrazí pouze vedoucím, kteří k němu mají udělen přístup. Nástroj podporuje soubory ve formátu CSV. Tento formát byl zvolen jako výchozí díky své jednoduché struktuře. Dalším vhodným formátem by mohl být XML, jeho vnitřní stavba se však v různých případech může oproti CSV značně lišit. Pro použití XML souborů lze doplnit předzpracování souboru. Jednou z možností je například využití XSLT [15] k převodu souboru ve formátu XML do formátu CSV.



The image shows a web form titled "Nahrajte soubor s daty". It contains the following elements:

- A text input field for the file name.
- A button labeled "Vyberte soubor" (Choose file).
- A dropdown menu labeled "Vyberte osobu" (Choose person) with a downward arrow.
- A button labeled "Nahrát" (Upload).

Obrázek 5.1: Návrh rozhraní pro výběr souboru.

5.1.2 Zpracování souboru

Po nahrání souboru následuje jeho zpracování. Na prvním řádku se nachází hlavička, která označuje jednotlivé sloupce. Pod ní se nacházejí řádky s jednotlivými činnostmi. Při zpracování souboru je nutné nejprve identifikovat jednotlivé sloupce. Aby mohla být data zpracována, musí povinně obsahovat datum činnosti, dále počáteční a koncový čas činnosti, nebo délku trvání činnosti. Nepovinným údajem je popis činnosti. Protože se výstupy jednotlivých programů mohou lišit, nástroj musí být schopen detekovat sloupce s různými názvy a různým pořadím. K tomu využívá možné názvy sloupců, např. date, hours, minutes, total, start, title, podporovány jsou i názvy v češtině (den, datum, hodiny, popis atd.). Po identifikaci sloupců lze zpracovat jednotlivé činnosti a přejít na další krok procesu, ve kterém se importované činnosti zobrazí uživateli.

5.1.3 Úprava denních výkazů

Jednotlivé činnosti ze vstupního souboru tvoří denní výkazy. Protože uživatel může současně pracovat na více projektech, je nutné každý denní výkaz přiřadit k projektu. Aby uživatel nemusel všechna přiřazení provádět ručně, má možnost definovat vlastní zkratky projektů, které pak může uvést jako součást popisu činnosti. Nástroj tyto zkratky automaticky rozpozná a daný výkaz přiřadí k odpovídajícímu projektu. Obrázek 5.2 zachycuje návrh rozhraní, ve kterém uživatel může upravit jednotlivé denní výkazy.

Činnosti jsou podle data automaticky přiřazeny do týdne, případně podle zkratky i k projektu. Pokud importované činnosti obsahovaly neúplné údaje, uživatel je zde musí doplnit včetně přiřazení k projektu, poté může přejít k poslednímu kroku, kterým je doplnění týdenních výkazů.

Seznam denních výkazů								
Týden	Datum	Čas od	Čas do	Délka	Činnost	Projekt 1	Projekt 2	Projekt 3
1. 1.1.2013-7.1.2013	3.1.2013	14:00	15:00	1:00	IBP - návrh tříd	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
1. 1.1.2013-7.1.2013	3.1.2013	16:00	17:20	1:20	IBP - jednotkové testy	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
1. 1.1.2013-7.1.2013	4.1.2013	10:30	12:00	1:30	rss2 - úprava repozitáře	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
2. 8.1.2013-14.1.2013	10.1.2013	08:10	08:30	0:20	rss2 - tvorba výkazu	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Obrázek 5.2: Návrh rozhraní pro úpravu denních výkazů.

Týdenní výkazy
1/3

Týden 1.1.2013 - 7.1.2013
Projekt IBP
« »

Zahodit
Uložit
Uložit a odeslat

Denní výkazy

Datum	Čas od	Čas do	Délka	Činnost
3.1.2013	14:00	15:00	1:00	návrh tříd
3.1.2013	16:00	17:20	1:20	jednotkové testy

Obrázek 5.3: Návrh rozhraní pro tvorbu týdenních výkazů.

5.1.4 Tvorba týdenních výkazů

Upravené denní výkazy se podle týdne a projektu rozdělí do skupin, příklad tohoto rozdělení je uveden v následujícím výpisu:

```
Projekt IBP , Týden 1
Projekt rss2 , Týden 1
Projekt rss2 , Týden 2
```

Výpis 5.1: Ukázka kombinací týdne a projektu

Pro každou tuto skupinu se pak musí vytvořit příslušný týdenní výkaz. Návrh rozhraní se nachází na obrázku 5.3.

V tabulce denních výkazů se zobrazí i ty výkazy, které se pro příslušný týden a projekt v databázi KNOTIS již nacházejí. Uživatel zde může upravit jednotlivé denní výkazy. Do týdenního výkazu se ukládá souhrn práce, plán práce na další týden, problémy a omluvy k aktuálnímu týdnu. Uživatel může s týdenním výkazem provádět tyto operace:

- *Zahodit* – systém smaže importované denní výkazy pro danou kombinaci týdne a projektu. Tuto možnost lze aplikovat pouze v případě, že týdenní výkaz ještě nebyl uložen.
- *Uložit* – systém uloží denní výkazy včetně případných změn a uloží také týdenní výkaz.
- *Uložit a odeslat* – systém provede stejné operace jako v případě *Uložit*, navíc týdenní výkaz odešle na e-mail vedoucího projektu, dotyčného pracovníka, případně dalším příjemcům.

Všechny tyto operace po svém provedení zobrazí v pořadí následující skupinu týden – projekt.

5.1.5 Detekce duplicitních výkazů

Při tvorbě výkazů pomocí importu může nastat situace, kdy uživatel použije aktualizovaný soubor, který kromě nových činností obsahuje i činnosti již jednou vykázané (uživatel například upraví popis některých činností). V takovém případě je v rámci uživatelské přívětivosti žádoucí, aby byly duplicitní výkazy automaticky detekovány a uživatel je nemusel ručně odstraňovat. Tato detekce proběhne při posledním kroku, tedy úpravě týdenních výkazů. Aby dva výkazy mohly být označeny jako shodné, musí obsahovat stejné datum, stejný čas začátku činnosti a stejný čas konce činnosti či délku trvání činnosti. Nástroj pak rozliší mezi dvěma typy duplicity - zcela shodné výkazy a výkazy lišící se popisem činnosti. V prvním případě se nově importované výkazy automaticky odstraní a původní výkazy se v tabulce označí, aby bylo zřejmé, že došlo ke smazání duplicitního výkazu. U výkazů, které se liší popisem činnosti, se zobrazí a zvýrazní oba výkazy tak, aby uživatel mohl rozhodnout, který z nich odstraní, a provedl toto smazání ručně.

5.2 Automatický plánovač

Plánovač má za úkol vytvoření pracovního plánu na základě počtu hodin k odpracování a rozsahu týdnů, ve kterých má být práce vykonána. Při vytváření plánovače je potřeba uvážit různá roční období a situace, které mohou nastat.

5.2.1 Rozdělení do období

Jak již bylo zmíněno v kapitole 3.2, rok se dělí do několika období, ve kterých lze očekávat rozdílné počty odpracovaných hodin týdně. Na obrázku 5.4 je zobrazeno ukázkové rozdělení školního roku. Rozlišujeme mezi běžným semestrem, obdobím půlsesternálních zkoušek a zkouškovým obdobím.



Obrázek 5.4: Přehled ročních období.

Součástí rozšíření KNOTIS tedy bude rozhraní, které umožní definovat libovolný počet ročních období a pro každé nastavit požadovaný počet hodin k odpracování týdně. Pracovníci ve skupině KNOT se dělí do několika skupin, proto se budou hodiny nastavovat

Seznam pracovních období

Od	do	popis	
1.1.	31.1.	Zkoušky	upravit
1.2.	5.5.	LS	upravit

Přidat období

Typy osob

BIT	upravit
MIT	upravit
Zam	upravit

Typ BIT

	hod.	navyš.
1.1. - 31.1.	<input type="text" value="1"/>	<input checked="" type="checkbox"/>
1.2. - 5.5.	<input type="text" value="5"/>	<input checked="" type="checkbox"/>

Obrázek 5.5: Návrh rozhraní pro správu období.

pro každou skupinu zvlášť. Návrh rozhraní pro správu období je ukázán na obrázku 5.5. Okno pro nastavení hodin pro konkrétní skupinu uživatelů obsahuje také položku *navýšit*, která určuje, zda plánovač v tomto období může v případě potřeby navýšit již vyplněné položky plánu. Toto nastavení může například zajistit, aby se prioritně navyšovaly týdny během semestru, nikoliv během zkouškového období. V případě, že pro určitou část roku není vytvořeno žádné období, plánovač pro dané týdny uvažuje týdenní konstantu rovnu nule. Předpokládá se tak, že v nedefinovaných obdobích není vyžadována žádná práce.

5.2.2 Režimy plánovače

Plánovač pracuje ve třech různých režimech:

- běžný režim
- režim zahození budoucnosti
- režim přeplánování budoucnosti

Základním plánovacím režimem je **běžný režim**. Ten nejprve zjistí, zda existují neaplánované hodiny. Pokud ano, rozdělí je do zbývajících týdnů tak, že hodiny podle nastavených konstant doplní do týdnů, které se nacházejí za posledním vyplněným týdnem. Tento princip respektuje předchozí vytvořený plán a snaží se do něj nezasahovat. Pokud po tomto rozdělení ještě zbývají hodiny k naplánování, rozdělí se podle nastavení rovnoměrně do týdnů, ve kterých lze navyšovat, případně do všech týdnů. Ukázka běžného režimu je na obrázku 5.6.

Režim zahození budoucnosti pracuje stejně jako běžný režim, před samotným rozdělením hodin však kompletně vymaže již existující plán pro budoucí týdny. Tento režim lze tedy použít v situacích, kdy je plán vytvořen špatně a běžný režim by jej nedokázal vylepšit. Ukázka takového plánu je zachycena na obrázku 5.7.

5	0	10	0	0	0
5	0	10	4	4	4

Obrázek 5.6: Běžný režim.

10	53	0	19	60	0
23	23	23	23	23	23

Obrázek 5.7: Režim zahození budoucnosti.

Režim přepřánování budoucnosti, na rozdíl od zahození budoucnosti, nemaže celý vytvořený plán. Hodnoty, které přesahují konstantu příslušející danému týdnu, se oříznou na hodnotu konstanty a přebytek se převede na nenaplánované hodiny. Dále se opět použije běžný režim. Jak lze vidět na obrázku 5.8, režim přepřánování má své využití v situacích, kdy daná osoba opakovaně nevykazuje žádnou práci, následkem čehož se neodpracované hodiny navyšují směrem k termínu odpracování. V tomto okamžiku vedoucí projektu provede oddálení termínu, běžný režim by však na tuto změnu nedokázal zareagovat, neboť všechny hodiny jsou již naplánovány. Režim přepřánování budoucnosti se s touto situací dokáže vypořádat.

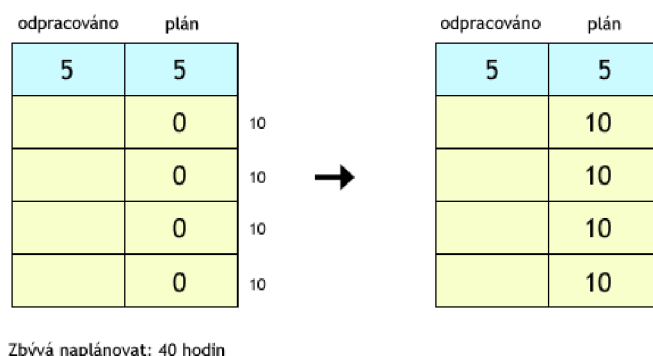
5	4	100			
5	4	10	33	33	33

Obrázek 5.8: Režim přepřánování budoucnosti.

5.2.3 Situace při plánování

Při plánování mohou nastat některé neobvyklé situace, které je potřeba řešit podle určitých pravidel.

Ještě před problémovými situacemi uvedu ukázkou běžného stavu na obrázku 5.9. Modrou barvou jsou označeny týdny v minulosti, žlutou barvou týdny v budoucnosti (včetně aktuálního týdne). Čísla vpravo od tabulky označují základní počet hodin k odpracování pro daný týden. V první ukázce stačí 40 hodin k odpracování postupně vložit do jednotlivých týdnů po deseti. Protože předtím nebyl vyplněn žádný řádek v budoucnosti, začíná se vkládat do prvního týdne, jinak by se hodiny vkládaly až za poslední vyplněný týden.



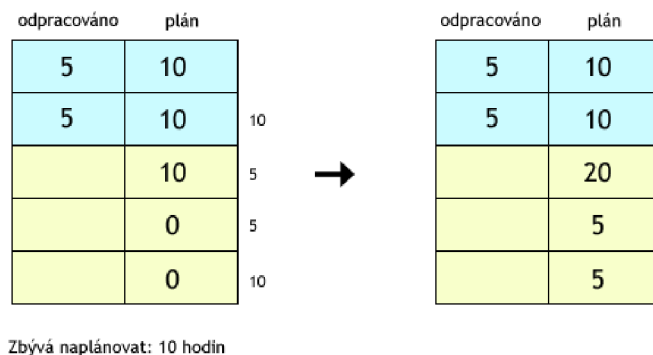
Obrázek 5.9: Normální stav.

Prvním problémem, který může nastat, je překročení základních konstant, jak je ukázáno na obrázku 5.10. Jednotlivé týdny se nejprve opět naplní podle zadaných konstant, nicméně po tomto naplnění stále zbývá ještě 10 hodin k naplánování. Dalším krokem je tedy navýšení již vyplněných týdnů. Uvažujme, že v tomto případě lze navýšit všechny týdny, 10 hodin vydělíme 4-mi týdny, ve výsledku se tedy každý týden navýší o 2,5 hodiny.



Obrázek 5.10: Překročení základních konstant.

Při plánování je potřeba zohlednit také plán v minulosti a uložené pracovní výkazy. Na obrázku 5.11 lze vidět, že v minulosti bylo plánováno celkem 20 hodin, osoba však odpracovala pouze 10 hodin. Současně zbývá naplánovat ještě dalších 10 hodin. Nejprve je tedy nutné zjistit počet hodin, které se převedou z minulosti, v tomto případě je to neodpracovaných 10 hodin. Tyto hodiny se následně navýší do vyplněných řádků v budoucnosti, které povolují navýšení. Pokud žádné vyplněné řádky navýšení nepovolují, rozdělí se do všech vyplněných. V případě, že v budoucnosti neexistují žádné vyplněné řádky, tyto hodiny se automaticky připočítají k hodinám, které je potřeba naplánovat. Hodiny k naplánování se pak do plánu doplní již podle běžných pravidel.



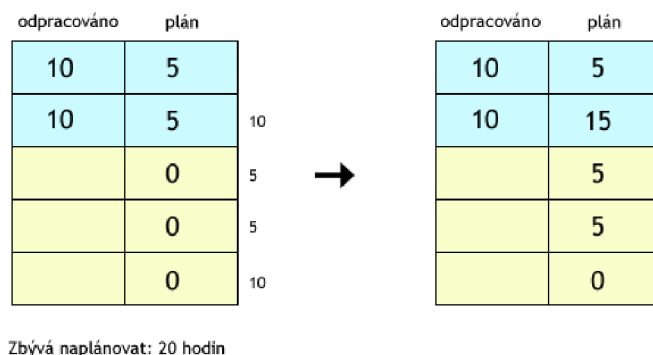
Obrázek 5.11: Navýšení o minulost.

Nastat může i opačná situace, ve které osoba odpracuje více hodin, než kolik v minulosti plánovala. Viz obrázek 5.12. Dohromady bylo odpracováno o 10 hodin více. Je potřeba plán zpětně navýšit, aby celkový počet naplánovaných hodin nebyl nižší než počet hodin k odpracování. Do posledního týdne v minulosti se tedy přidá potřebný počet hodin, v tomto případě 10. Zbývajících 10 hodin, které je potřeba naplánovat, se pak v budoucnosti rozloží podle výše uvedených pravidel.

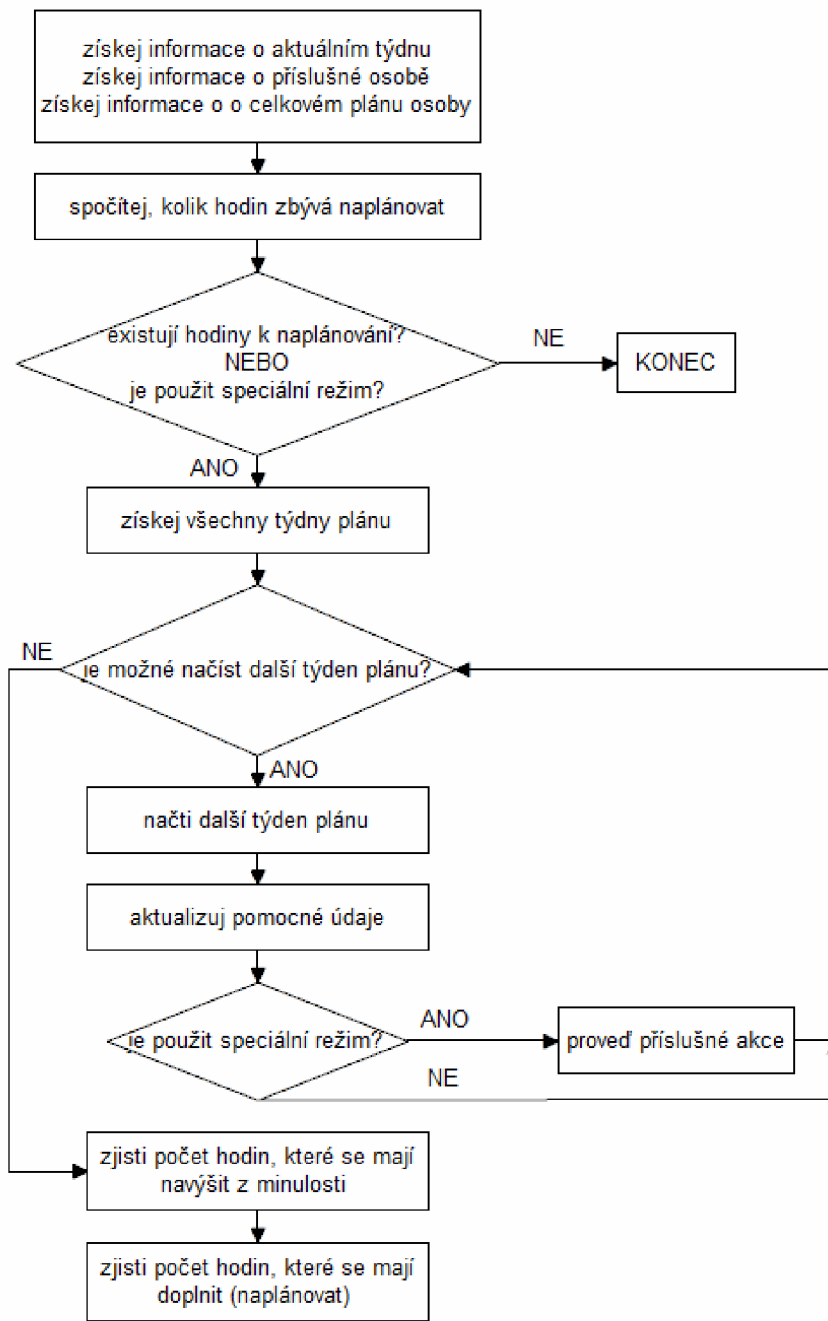
5.2.4 Činnost plánovače

Zjednodušený vývojový diagram, který popisuje postup automatického plánovače, je zachycen na obrázcích 5.13 a 5.14. Nejprve je nutné získat podklady k plánování, podle kterých se určí, zda je potřeba naplánovat nějaké hodiny. Režimy zahození a přeplánování budoucnosti lze použít i v případě, že nezbývají žádné hodiny k naplánování. Po načtení všech týdnů plánu se tyto týdny jednotlivě procházejí a skript zaznamenává potřebné údaje (počet týdnů v budoucnosti, počet vyplněných týdnů v budoucnosti a další). Zde se také provedou příslušné akce, pokud je aktivován speciální režim.

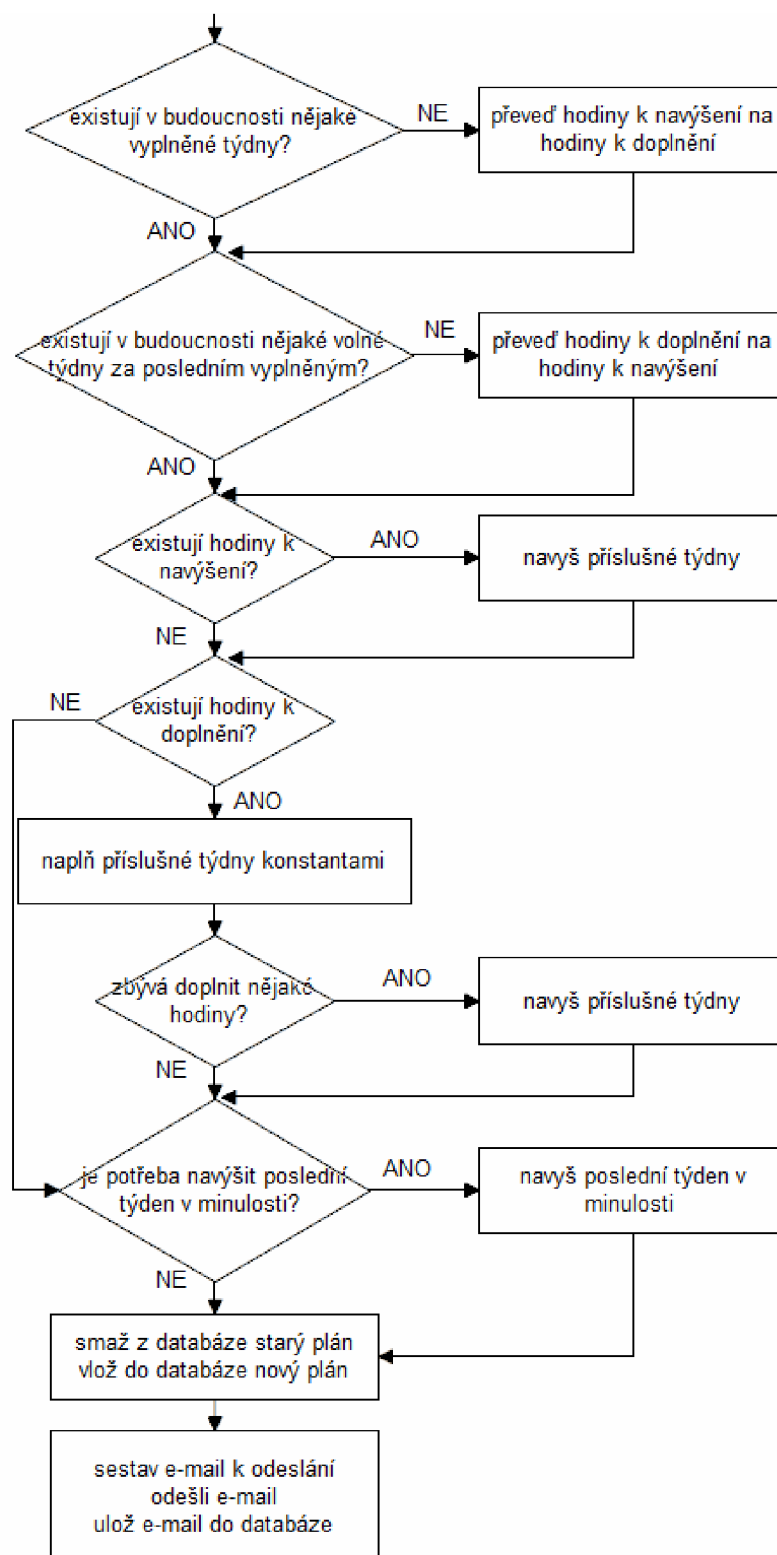
Po zpracování všech týdnů získá plánovač počet hodin, které je potřeba navýšit a doplnit. Tyto hodiny pak následně podle uvedených pravidel doplní do plánu. Skript také upraví



Obrázek 5.12: Nadpracované hodiny.



Obrázek 5.13: Vývojový diagram plánovače 1/2.



Obrázek 5.14: Vývojový diagram plánovače 2/2.

poslední týden v minulosti, je-li to potřeba. Po kompletním vytvoření nového plánu se starý plán v databázi nahradí novým. Poslední činností plánovače je sestavení a odeslání e-mailu o změně plánu příslušné osobě. Tento e-mail se zálohuje i do databáze, odkud jej lze zobrazit prostřednictvím KNOTIS.

5.3 Další vylepšení systému

Při analýze systému jsem narazil na další problémy, které komplikovaly práci se systémem, viz kapitola 3.3.

U výkazů, jejichž čas zasahuje přes půlnoc, je automatické posunutí data zamýšleným efektem. Důvodem pro tuto funkcionalitu je řazení výkazů v tabulce s denními výkazy. Je požadováno, aby tabulka byla seřazena od nejnovějších výkazů. Pokud však uživatel v jednom dni dokončil a vykázal práci, kterou začal vykonávat předchozího dne, a později ve stejném dni vykázal další hodiny, pořadí výkazů bylo obrácené. Bylo to způsobeno počátečním časem činnosti, kdy například 23:00 (první činnost) je „novější“ než 13:00 (druhá činnost) a proto se první činnost zobrazovala výše. Správného řazení je docíleno automatickou změnou data výkazu, který se rozkládá přes půlnoc, na předchozí den. Tuto změnu je však potřeba provést pouze při vytváření nového výkazu. V případě, že uživatel upravuje existující výkaz, automatická změna již není nutná.

Dalším požadavkem je možnost přesunout výkaz do jiného týdne. Pokud uživatel vytvoří výkaz v určitém týdnu, systém mu při úpravě výkazu neumožní zadat datum, které do daného týdne nepatří. Novou součástí rozhraní pro úpravu denního výkazu tak bude tlačítko, které „odblokuje“ ostatní týdny a při uložení výkazu jej přesune do týdne příslušejícího novému datu.

Nejdůležitějším vylepšením je povolení použití apostrofu na vhodných místech. Tento znak byl z bezpečnostních důvodů v celém systému zakázán. Po analýze KNOTIS jsem sestavil seznam položek, kde by bylo vhodné apostrof povolit. Jedná se nejen o popis činnosti denního výkazu a pole týdenního výkazu, ale například i o název projektu či zadání úkolu k řešení. Detekce apostrofu se provádí v rámci kontroly všech řetězců, které jsou získány z formulářů. Existuje několik úrovní kontroly, mezi sebou se liší sadou povolených znaků. Pro podporu apostrofu bude vytvořena nová úroveň, která zahrne i tento znak. Úprava však přináší bezpečnostní riziko v podobě SQL Injection ¹, je tedy potřeba všechny řetězce, které mohou obsahovat apostrof, před kontaktem s databází důkladně ošetřit.

¹Napadení databáze vsunutím vlastního dotazu do neošetřeného existujícího dotazu.

Kapitola 6

Implementace

Při implementaci obou rozšíření jsem využil volně dostupné knihovny pro Javascript - jQuery, jQuery UI (kapitola 4.5), Tablesorter (kapitola 6.1.2) - a vestavěné funkce jazyka PHP. Snažil jsem se také co nejvíce dělit kód do krátkých funkcí, kde má každá funkce pouze jeden úkol [23]. Každá funkce je popsána komentářem pro Doxygen ¹.

6.1 Import výkazů

6.1.1 Zpracování vstupního souboru

Pro načtení řádků souboru ve formátu CSV jsem použil nativní PHP funkci `fgetcsv`, která vrací pole obsahující hodnoty jednotlivých sloupců, vždy pro jeden aktuální řádek. První řádek se zpracuje jako hlavička a skript identifikuje jednotlivé sloupce podle pravidel uvedených v kapitole 5.1.2. U ostatních řádků se pak jednotlivé sloupce zpracovávají podle hodnoty, kterou obsahují. Ke zpracování data, času a délky činnosti je použita třída `DateTime`. Délku činnosti lze zadat počtem minut i počtem hodin. U popisu činnosti se skript podle definovaných zkratkou pokusí identifikovat, zda činnost patří k nějakému konkrétnímu projektu. U každé činnosti se během zpracování provede automatické dopočítání času, které na základě dvou hodnot (jsou-li vyplněny) spočítá hodnotu třetí (počáteční čas, koncový čas, délka činnosti). Poté jsou všechny činnosti uloženy do session.

6.1.2 Seznam denních výkazů

Denní výkazy uložené v session se v dalším kroku zobrazí v přehledné tabulce, viz obrázek 6.1. Jednotlivé položky je možné editovat, v pravé části rozhraní lze vidět seznam projektů, na kterých osoba pracuje a kde je potřeba denní výkaz přiřadit k jednomu z nich. Řádky tabulky lze dynamicky řadit podle data, času a čísla řádku, k tomuto účelu jsem použil jQuery knihovnu *Tablesorter* [10].

Použil jsem také komponentu *Datepicker* [18] knihovny jQuery UI, která umožňuje snadný výběr data (obrázek 6.2) a poskytuje funkce pro zpracování řetězce s datem.

Při změně některého z údajů probíhá validace celého řádku, která umožňuje i automatické dopočítání odpracovaného počtu hodin na základě počátečního a koncového času. Při validaci se také kontroluje, zda zadané datum spadá do rozsahu práce na projektu, který je k dané činnosti přiřazen.

¹<http://www.doxygen.org>

Výkazy pro Uživatel Testovací

Týden	Datum	Čas od	Čas do	Počet hodin	Činnost	zproj...	rss2...
1. (31.12.2012 - 06.01.2013)	03.01.2013	14:00	15:00	01:00	IBP - návrh tříd, dokumentace	<input checked="" type="radio"/>	<input type="radio"/>
1. (31.12.2012 - 06.01.2013)	03.01.2013	16:00	17:20	01:20	IBP - jednotkové testy	<input checked="" type="radio"/>	<input type="radio"/>
1. (31.12.2012 - 06.01.2013)	04.01.2013	10:30	12:00	01:30	rss2 - úprava repozitáře	<input type="radio"/>	<input checked="" type="radio"/>
2. (07.01.2013 - 13.01.2013)	10.01.2013	08:10	08:30	00:20	rss2 - tvorba výkazu	<input type="radio"/>	<input checked="" type="radio"/>

Uložit Přidat řádek

Obrázek 6.1: Rozhraní pro úpravu denních výkazů.

Datum	Čas od	Čas do	Počet hodin	Činnost
03.01.2013	14:00	15:00	01:00	IBP - návrh tříd
				IBP - jednotkov
				rss2 - úprava r
				rss2 - tvorba vý

leden 2013						
po	út	st	čt	pá	so	ne
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Obrázek 6.2: Výběr data.

6.1.3 Týdenní výkazy

Na obrázku 6.3 lze vidět rozhraní pro úpravu týdenního výkazu. V případě, že týdenní výkaz již existuje, zobrazí se v polích texty, které lze editovat. Kromě akčních tlačítek jsem implementoval také tlačítka na posun mezi jednotlivými týdenními výkazy. Příslušné denní výkazy lze upravit, není zde však již libovolné řazení, výkazy se řadí automaticky dle data. K tomu jsem využil nativní JS funkci `sort`. I v této tabulce je implementována validace pomocí Javascriptu. Výkazy mohou být označeny různými barvami, světle modrou jsou označeny nově importované výkazy, žlutou barvu mají ty výkazy, které již byly vytvořeny dříve a byly získány z databáze. Další barvy výkazů jsou popsány v kapitole 6.1.4.

Aby nástroj pro import poskytoval stejné možnosti, jako klasické rozhraní, nachází se u každého denního výkazu pole NSP, které umožňuje výkaz označit jako „nesouvisí s projektem“. Díky této možnosti lze vykázat činnosti, které se nezobrazí v seznamu činností vykonaných na určitém projektu. Není tak například nutné vytvářet „pomocné“ projekty určené speciálně pro netypické činnosti (školení, krátká účast na jiném projektu).

6.1.4 Označení duplicitních výkazů

Uvažujme CSV soubor z výpisu 3.1, který uživatel použije pro vytvoření výkazů. Postup jeho importu je zachycen na obrázcích v této kapitole (6.1, 6.3).

Uživatel následně importuje soubor s tímto aktualizovaným obsahem:

```

1 "Date","Start Time","End Time","Minutes","Title"
2 "1/3/13","14:00","15:00","60","IBP - návrh tříd, dokumentace"
3 "1/3/13","16:00","17:20","80","IBP - jednotkové testy (10 testů)"
4 "1/4/13","10:30","12:00","90","rss2 - úprava repozitáře"
5 "1/10/13","08:10","08:30","20","rss2 - tvorba výkazu"

```

Výsledek druhého importu je ukázán na obrázku 6.4. Světle růžovou je obarven výkaz, který byl označen jako zcela duplicitní a byl automaticky odstraněn, v seznamu tak zůstala pouze původní činnost. Výkaz s tmavou barvou je označen jako částečně shodný (liší se popisem činnosti). Je tedy na uživateli, kterou ze dvou podobných činností odstraní.

Týdenní výkaz:

Projekt: 1. zproj: Překladač slovník

Týden: 1. (31.12.2012 - 06.01.2013)

Souhm:

Plán:

Problémy a nejasnosti:

Omluvy a výmluvy:

Zahodit Uložit Uložit a odeslat << >> 1/3

Pracovní výkazy:

Datum	Od	Do	Počet hodin	Činnost	NSP	
03.01.2013	14:00	15:00	01:00	IBP – návrh tříd, dokumentace	<input type="checkbox"/>	odstranit
03.01.2013	16:00	17:20	01:20	IBP – jednotkové testy	<input type="checkbox"/>	odstranit

Přidat řádek

Obrázek 6.3: Rozhraní pro tvorbu týdenních výkazů.

Pracovní výkazy:

Datum	Od	Do	Počet hodin	Činnost	NSP	
03.01.2013	14:00	15:00	01:00	IBP – návrh tříd, dokumentace	<input type="checkbox"/>	odstranit
03.01.2013	16:00	17:20	01:20	IBP – jednotkové testy	<input type="checkbox"/>	odstranit
03.01.2013	16:00	17:20	01:20	IBP – jednotkové testy (10 testů)	<input type="checkbox"/>	odstranit

Obrázek 6.4: Označení duplicitních výkazů.

6.2 Automatický plánovač

Při implementaci plánovače jsem postupoval podle vývojového diagramu uvedeného v kapitole 5.2.4. Využil jsem také některé části kódu z předchozí verze plánovače (odeslání e-mailu).

Aby bylo možné určit, kolik hodin se má plánovat, je potřeba získat z tabulky *osoba* informace o rozsahu práce, odpracovaných a naplánovaných hodinách. Pro zjištění údajů o odpracovaných a plánovaných hodinách jsem použil vnořené dotazy. U těchto dotazů může dojít k situaci, že v tabulkách *plan* a *vykaz* neexistují žádné odpovídající záznamy a výsledkem dotazu je hodnota `NULL`. Proto jsem použil podmínku `IFNULL`, která umožňuje prázdnou hodnotu nahradit nulou. Stejným způsobem jsem vytvořil také dotaz pro získání plánu osoby, který vybírá data z tabulky *tyden*. Ke každému týdnu pak vnořenými dotazy připojí týdenní konstantu a údaj o tom, zda lze v tomto týdnu navyšovat hodiny.

Plánovač postupně prochází týdny plánu získané z databáze a ukládá je do pole, aby s nimi mohl dále pracovat. Poté, co je vytvořena nová verze plánu pro danou osobu, skript odstraní starý plán z databáze a nahradí jej plánem novým. Na závěr své činnosti plánovač zašle upozornění o změně plánu na e-mailovou adresu pracovníka.

Rozhraní pro správu plánovacích období se nachází na obrázku 6.5. Při úpravě jednotlivých období systém kontroluje, zda jsou zadána validní data a zda je počáteční datum menší než koncové datum. KNOTIS také dokáže detekovat překrývající se období a zobrazit upozornění v případě, že některá ze skupin pracovníků nemá vyplněna žádná plánovací období.

6.3 Použití apostrofu

Detekci nepovolených znaků provádí funkce `kontrolaRetezce`, která přijímá vstupní řetězce z formulářových polí a pomocí regulárních výrazů zjišťuje, zda řetězec odpovídá nastavené úrovni kontroly. Na řetězce, které mohou obsahovat apostrof, se před použitím v databázovém dotazu aplikuje funkce `mysql_real_escape_string`, jejíž úkolem je upravit řetězec tak, aby nemohl způsobit SQL Injection. Při implementaci jsem zjistil, že řetězce s apostrofem se do databáze ukládají nejen v dotazech upravujících příslušné údaje, ale i v dotazech, které slouží k ukládání e-mailů (ty mohou obsahovat řetězce s apostrofy). Bylo tedy potřeba důkladně ošetřit všechna místa, kde by potenciálně mohlo dojít k útoku.

Seznam plánovacích období

Od	Do	Popis	
<input type="text" value="01.01"/>	<input type="text" value="03.02"/>	<input type="text" value="zimní zkouškové"/>	<input type="button" value="Smazat"/>
<input type="text" value="04.02"/>	<input type="text" value="25.04"/>	<input type="text" value="letní semestr"/>	<input type="button" value="Smazat"/>
<input type="text" value="26.04"/>	<input type="text" value="10.06"/>	<input type="text" value="letní zkouškové"/>	<input type="button" value="Smazat"/>
<input type="text" value="11.06"/>	<input type="text" value="19.09"/>	<input type="text" value="prázdniny"/>	<input type="button" value="Smazat"/>
<input type="text" value="20.09"/>	<input type="text" value="20.12"/>	<input type="text" value="zimní semestr"/>	<input type="button" value="Smazat"/>
<input type="text" value="21.12"/>	<input type="text" value="31.12"/>	<input type="text" value="zimní zkouškové"/>	<input type="button" value="Smazat"/>

Přidat období

Typy osob

BIT	upravit
MIT	upravit
DVI	upravit
ExD	upravit
Spol	upravit
Zam	upravit

Obrázek 6.5: Správa plánovacích období.

Kapitola 7

Testování

Při testování nových rozšíření pro systém KNOTIS jsem se u nástroje pro import zabýval především chováním systému v případech, kdy uživatel použije soubor obsahující chybná data. U automatického plánovače jsem testoval použití všech plánovacích režimů na různé výchozí situace. Některé z testů jsem uvedl v následující kapitole.

7.1 Import - data obsahující chyby

Tento test má za cíl ověřit chování systému v případě, že uživatel při importu použije soubor, ve kterém chybí některé údaje, případně jsou neplatné. Při testu jsem použil uživatele *xtest00*. Obsah CSV souboru je následující:

```
1 "Date","Start Time","End Time","Minutes","Title"
2 "neplatná hodnota","12:30","13:30","","zp - činnost 1"
3 "15.2.2013","","","","zp - činnost 2"
4 "15.2.2013","","","90","zp - činnost 3"
5 "15.2.2013","","","1:30","zp - činnost 4"
6 "15.2.2013","neplatná hodnota","","90","činnost 5"
```

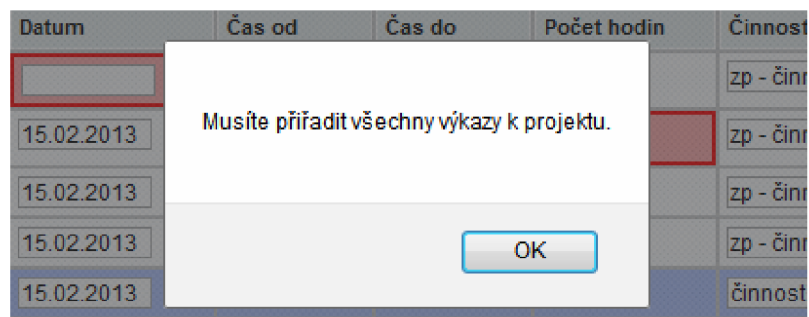
Výpis 7.1: Obsah importovaného souboru ve formátu CSV

Data obsahují neplatné hodnoty ve sloupci *Date* a *Start Time*. Ve třetím řádku také chybí upřesnění času. KNOTIS by měl „zahodit“ neplatné hodnoty a upozornit na chybějící údaje, bez kterých nelze přejít do další fáze importu. Popis činnosti u prvních čtyř výkazů obsahuje uživatelem definovanou zkratku projektu, systém by tedy tyto činnosti měl automaticky přiřadit k příslušnému projektu. Výsledek importu je ukázán na obrázku 7.1. Modře zbarvený řádek označuje výkaz, který je nutno přiřadit k projektu. V importu nelze pokračovat, dokud nejsou všechny výkazy přiřazeny k projektu, viz obrázek 7.2.

Nástroj pro import obsahuje validaci i na straně serveru, ukázka je na obrázku 7.3.

Týden	Datum	Čas od	Čas do	Počet hodin	Činnost	zproj...
1 16. (15.04.2013 - 21.04.2013)	<input type="text"/>	<input type="text" value="12:30"/>	<input type="text" value="13:30"/>	<input type="text" value="01:00"/>	<input type="text" value="zp - činnost 1"/>	<input checked="" type="radio"/>
2 7. (11.02.2013 - 17.02.2013)	<input type="text" value="15.02.2013"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="zp - činnost 2"/>	<input checked="" type="radio"/>
3 7. (11.02.2013 - 17.02.2013)	<input type="text" value="15.02.2013"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="01:30"/>	<input type="text" value="zp - činnost 3"/>	<input checked="" type="radio"/>
4 7. (11.02.2013 - 17.02.2013)	<input type="text" value="15.02.2013"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="01:30"/>	<input type="text" value="zp - činnost 4"/>	<input checked="" type="radio"/>
5 7. (11.02.2013 - 17.02.2013)	<input type="text" value="15.02.2013"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="01:30"/>	<input type="text" value="činnost 5"/>	<input type="radio"/>

Obrázek 7.1: JS validace souboru.



Obrázek 7.2: Uložení výkazů bez přiřazení k projektu.

Chyba: Nebyly zadány minuty ani hodiny.

Výkazy nebyly uloženy, protože obsahovaly chyby.

Výkazy pro Uživatel Testovací

	Týden	Datum	Čas od	Čas do	Počet hodin	Činnost	zproj...
1	16. (15.04.2013 - 21.04.2013)	<input type="text"/>	12:30	13:30	01:00	zp - činnost 1	<input type="radio"/>
2	7. (11.02.2013 - 17.02.2013)	15.02.2013	<input type="text"/>	<input type="text"/>	<input type="text"/>	zp - činnost 2	<input type="radio"/>
3	7. (11.02.2013 - 17.02.2013)	15.02.2013	<input type="text"/>	<input type="text"/>	01:30	zp - činnost 3	<input type="radio"/>
4	7. (11.02.2013 - 17.02.2013)	15.02.2013	<input type="text"/>	<input type="text"/>	01:30	zp - činnost 4	<input type="radio"/>
5	7. (11.02.2013 - 17.02.2013)	15.02.2013	<input type="text"/>	<input type="text"/>	01:30	činnost 5	<input type="radio"/>

Obrázek 7.3: Validace souboru na straně serveru.

7.2 Plánovač – plán nového uživatele

Jednou ze situací, kdy je použit automatický plánovač, je vytvoření nového uživatele. Vytvořil jsem nového uživatele s těmito údaji:

- *Typ:* Stud. spolupracovník
- *Login:* xtest01
- *Aktivní od:* 15.04.2013
- *Odpracovat:* 50
- *Odpracovat od:* 15.04.2013
- *Odpracovat do:* 09.06.2013

Na obrázku 7.4 se nachází nastavení týdenních konstant pro typ Spolupracovník. Na obrázku 7.5 pak lze vidět, jak plánovač na základě tohoto nastavení rozdělil hodiny k odpracování.

Nastavení období pro typ Stud. spolupracovník

Období	Popis	Počet hodin	Lze navýšit
01.01 - 03.02	zimní zkouškové	2	<input type="checkbox"/>
04.02 - 05.05	letní semestr	5	<input checked="" type="checkbox"/>
06.05 - 10.06	letní zkouškové	3	<input type="checkbox"/>
11.06 - 19.09	prázdniny	4	<input checked="" type="checkbox"/>
20.09 - 20.12	zimní semestr	5	<input checked="" type="checkbox"/>
21.12 - 31.12	zimní zkouškové	2	<input type="checkbox"/>

Obrázek 7.4: Hodnoty plánovacích týdenních konstant pro typ Stud. spolupracovník.

Číslo týdne	Pondělí	Neděle	Plánováno hodin	Zbude hodin
16	15.04.2013	21.04.2013	11 : 40 *	38:20
17	22.04.2013	28.04.2013	11 : 40 *	26:40
18	29.04.2013	05.05.2013	11 : 40 *	15:00
19	06.05.2013	12.05.2013	3 : 0 *	12:00
20	13.05.2013	19.05.2013	3 : 0 *	9:00
21	20.05.2013	26.05.2013	3 : 0 *	6:00
22	27.05.2013	02.06.2013	3 : 0 *	3:00
23	03.06.2013	09.06.2013	3 : 0 *	0:00
Celkem			50:00	0:00

Obrázek 7.5: Plán pracovníka vytvořený automatickým plánovačem.

7.3 Plánovač – režim přeplánování budoucnosti

Tato ukázka testuje režim přeplánování budoucnosti. Plán uživatele jsem upravil podle obrázku 7.6, který simuluje situaci, kdy se všechny hodiny postupnou činností plánovače nahromadily do posledních týdnů před termínem odpracování. Vedoucí v tomto případě prodloužil termín odpracování, pro změnu plánu je však potřeba použít režim přeplánování, který vytvoří přijatelnější verzi plánu podle pravidel uvedených v kapitole 5.2. Výsledek přeplánování je ukázán na obrázku 7.7.

Číslo týdne	Pondělí	Neděle	Plánováno hodin	Zbude hodin
22	27.05.2013	02.06.2013	25 : 0 *	25:00
23	03.06.2013	09.06.2013	25 : 0 *	0:00
24	10.06.2013	16.06.2013	0 : 0 *	0:00
25	17.06.2013	23.06.2013	0 : 0 *	0:00
26	24.06.2013	30.06.2013	0 : 0 *	0:00
27	01.07.2013	07.07.2013	0 : 0 *	0:00
28	08.07.2013	14.07.2013	0 : 0 *	0:00
29	15.07.2013	21.07.2013	0 : 0 *	0:00
Celkem			50:00	0:00

Obrázek 7.6: Plán pracovníka před změnou.

Číslo týdne	Pondělí	Neděle	Plánováno hodin	Zbude hodin
22	27.05.2013	02.06.2013	3 : 0 *	47:00
23	03.06.2013	09.06.2013	3 : 0 *	44:00
24	10.06.2013	16.06.2013	3 : 0 *	41:00
25	17.06.2013	23.06.2013	8 : 12 *	32:48
26	24.06.2013	30.06.2013	8 : 12 *	24:36
27	01.07.2013	07.07.2013	8 : 12 *	16:24
28	08.07.2013	14.07.2013	8 : 12 *	8:12
29	15.07.2013	21.07.2013	8 : 12 *	0:00
Celkem			50:00	0:00

Obrázek 7.7: Plán pracovníka po činnosti režimu přeplánování.

7.4 Napadení systému pomocí SQL Injection

Jak již bylo zmíněno v kapitole 5.3, povolení apostrofů v řetězcích znamená bezpečnostní riziko a je potřeba systém ochránit proti SQL Injection. Následující ukázka zachycuje pokus o napadení databázové vrstvy systému. Útočník se pokouší uložit týdenní výkaz, který obsahuje souhrn s tímto textem: `';DELETE FROM vykaz;SELECT * FROM vykaz WHERE id = '`. Systém by v tomto případě měl ošetřit řetězec tak, aby při použití v databázovém dotazu neznamenal žádné nebezpečí. V opačném případě by došlo ke smazání všech dat z tabulky *vykaz*. Následující výpis zachycuje výsledný databázový dotaz:

```
1 UPDATE tydenni_v SET souhrn='\'';DELETE FROM vykaz;\'',plan='',
  problemy='',omluy='' WHERE id_osoby='1' AND id_vykazu='2';
```

Lze vidět, že systém vkládaný řetězec ošetřil správně a ten tak nepředstavuje bezpečnostní riziko.

Kapitola 8

Závěr

Před samotnou analýzou a implementací požadavků jsem se musel blíže seznámit s informačním systémem KNOTIS. Ten jsem znal z pohledu uživatele, nyní jsem však zkoumal vnitřní strukturu. Při analýze jsem se zaměřil především na možnosti plánování a vykazování práce.

V rámci semestrálního projektu jsem se zabýval analýzou požadavků na rozšíření systému KNOTIS. Navrhl jsem nástroj pro tvorbu výkazů z dat programů pro sledování času a novou verzi automatického plánovače.

Cílem bakalářské práce pak bylo obě rozšíření implementovat a nasadit do systému KNOTIS. První rozšíření nabízí uživatelům možnost vytvářet denní a týdenní výkazy na základě činností uložených v CSV souborech. Bylo potřeba vyřešit možné odlišnosti mezi výstupy jednotlivých programů a také validaci importovaných hodnot. Rozšíření umožňuje rozdělit denní výkazy podle týdne a projektu a následně k nim doplnit informace týdenních výkazů. Tento doplněk také dokáže detekovat duplicitní výkazy a automaticky je odstranit. Nástroj pro tvorbu výkazů by bylo možné dále rozšířit o podporu dalších formátů jako je XML či XLS.

Nová verze plánovače nyní zahrnuje rozhraní pro definici vlastních plánovacích období, které umožňuje jemnější a přesnější rozložení hodin. Lze tak například určit, že ve zkouškovém období je výchozí počet hodin k odpracování nižší než během semestru. Tyto hodiny je možné nastavit zvlášť pro jednotlivé typy pracovníků. Plánovač také poskytuje různé režimy činnosti, které usnadňují úpravu již existujících plánů či vytvoření plánu zcela nového.

V rámci analýzy systému jsem objevil několik problémů, které komplikovaly práci se systémem a po konzultaci s uživateli KNOTIS jsem se rozhodl pro jejich odstranění. Umožnil jsem tak přesun denního výkazu do jiného týdne, upravil jsem automatickou změnu data u výkazů přesahujících půlnoc a umožnil jsem uživatelům použití apostrofu na vhodných místech, díky kterému lze snadněji tvořit výkazy v anglickém jazyce.

Literatura

- [1] CSS3 Tutorial. W3Schools, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3schools.com/css3/>
- [2] PostgreSQL: The world's most advanced open source database. The PostgreSQL Global Development Group, [Online; navštíveno 15. 4. 2013].
URL <http://www.postgresql.org/>
- [3] Výzkumná skupina znalostních technologií. Fakulta Informačních Technologií VUT v Brně, [Online; navštíveno 15. 4. 2013].
URL <http://www.fit.vutbr.cz/research/groups/knot/>
- [4] What does it mean that Javascript is a prototype based language? [Online; navštíveno 15. 4. 2013].
URL <http://stackoverflow.com/questions/186244/what-does-it-mean-that-javascript-is-a-prototype-based-language>
- [5] jQuery. jQuery Foundation, 2013, [Online; navštíveno 15. 4. 2013].
URL <http://jquery.com/>
- [6] jQuery UI. jQuery Foundation, 2013, [Online; navštíveno 15. 4. 2013].
URL <http://jqueryui.com/>
- [7] MySQL: The world's most popular open source database. Oracle Corporation, 2013, [Online; navštíveno 15. 4. 2013].
URL <http://www.mysql.com/>
- [8] PHP: Hypertext Preprocessor. The PHP Group, 2013, [Online; navštíveno 15. 4. 2013].
URL <http://php.net/>
- [9] ECMAScript Language Specification. ECMA International, Červen 2011, [Online; navštíveno 15. 4. 2013].
URL <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
- [10] Bach, C.: Tablesorter documentation. [Online; navštíveno 16. 4. 2013].
URL <http://tablesorter.com/docs/>
- [11] Berjon, R.; aj.: HTML 5.1 Nightly. World Wide Web Consortium (W3C), 4. května 2013, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/html/wg/drafts/html/master/>

- [12] Berners-Lee, T.; Conolly, D.: *Hypertext Markup Language - 2.0*. World Wide Web Consortium (W3C), Listopad 1995, [Online; navštíveno 15. 4. 2013].
URL <http://www.ietf.org/rfc/rfc1866.txt>
- [13] Bos, B.; aj.: Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. World Wide Web Consortium (W3C), 7. června 2011, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/TR/CSS2/>
- [14] Chaffer, J.; Swedberg, K.: *Learning jQuery, Third Edition*. Packt Publishing, 2011, ISBN 978-1849516549.
- [15] Clark, J.: XSL Transformations (XSLT). World Wide Web Consortium (W3C), 16. listopadu 1999, [Online; navštíveno 16. 4. 2013].
URL <http://www.w3.org/TR/xslt>
- [16] Eich, B.; aj.: JavaScript Language Resources. Mozilla Developer Network, 2. dubna 2013, [Online; navštíveno 15. 4. 2013].
URL https://developer.mozilla.org/en-US/docs/JavaScript/Language_Resources
- [17] Garrett, J. J.: Ajax: A New Approach to Web Applications. 18. února 2005, [Online; navštíveno 15. 4. 2013].
URL <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>
- [18] Grabanski, M.; aj.: Datepicker. jQuery Foundation, [Online; navštíveno 16. 4. 2013].
URL <http://jqueryui.com/datepicker/>
- [19] Gutmans, A.; Rethans, D.; Bakken, S. S.: *Mistrouství v PHP 5*. Computer Press, 2005, ISBN 80-251-0799-X.
- [20] Hégarret, P. L.: Document Object Model (DOM) Technical Reports. World Wide Web Consortium (W3C), 2. května 2012, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/DOM/DOMTR>
- [21] Hors, A. L.; aj.: Document Object Model (DOM) Level 3 Core Specification. World Wide Web Consortium (W3C), 7. dubna 2004, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/TR/DOM-Level-3-Core/>
- [22] Lie, H. W.; Bos, B.: Cascading Style Sheets, level 1. World Wide Web Consortium (W3C), 17. prosince 1996, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/TR/REC-CSS1/>
- [23] Martin, R. C.: *Čistý kód*. Computer Press, 2009, ISBN 978-80-251-2285-3.
- [24] Ragget, D.: HyperText Markup Language Specification Version 3.0. World Wide Web Consortium (W3C), 28. března 1995, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/MarkUp/html3/html3.txt>
- [25] Ragget, D.; Hors, A. L.; Jacobs, I.: HTML 4.0 Specification. World Wide Web Consortium (W3C), 24. dubna 1998, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/TR/1998/REC-html40-19980424/html40.txt>

- [26] Ragget, D.; Hors, A. L.; Jacobs, I.: HTML 4.01 Specification. World Wide Web Consortium (W3C), 24. prosince 1999, [Online; navštíveno 15. 4. 2013].
URL <http://www.w3.org/TR/REC-html40/html40.txt>
- [27] Resig, J.: *JavaScript a Ajax*. Computer Press, 2007, ISBN 978-80-251-1824-5.
- [28] Shafranovich, Y.: Common Format and MIME Type for Comma-Separated Values (CSV) Files. Network Working Group, Říjen 2004, [Online; navštíveno 16. 4. 2013].
URL <http://tools.ietf.org/html/rfc4180>
- [29] Yemanov, D.; aj.: Firebird: True universal open source database. Firebird Foundation, 2013, [Online; navštíveno 15. 4. 2013].
URL <http://www.firebirdsql.org/>

Přílohy

Seznam příloh

A Obsah CD

39

Příloha A

Obsah CD

- Soubory informačního systému KNOTIS
- Ukázkové soubory s obsahem databáze systému
- Ukázkové CSV soubory
- Návod k instalaci systému
- Elektronická verze bakalářské práce