

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta chemickej a potravinárskej technológie

Evidenčné číslo: FCHPT-5414-44230

Informačný systém ÚIAM

Diplomová práca

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta chemickej a potravinárskej technológie

Evidenčné číslo: FCHPT-5414-44230

Informačný systém ÚIAM

Diplomová práca

Študijný program: automatizácia a informatizácia v chémii a potravinárstve

Študijný odbor: 5.2.14. automatizácia

Školiace pracovisko: Ústav informatizácie, automatizácie a matematiky

Vedúci záverečnej práce: Ing. Ľuboš Čirka, PhD.

Bratislava 2016

Bc. Martin Krippel



ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Martin Krippel**
ID študenta: 44230
Študijný program: automatizácia a informatizácia v chémii a potravinárstve
Študijný odbor: 5.2.14. automatizácia
Vedúci práce: Ing. Ľuboš Čírka, PhD.

Názov práce: **Informačný systém ÚIAM**

Špecifikácia zadania:

Cieľom práce je vytvoriť nový a prerobiť už existujúce moduly v Informačnom systéme ÚIAM (IS). Nový modul umožní vložiť a editovať predmety v IS na základe údajov v akademickom informačnom systéme STU (AIS). Existujúce moduly (citácie, dochádzka, dovolenky, ...) budú prerobené s využitím nových technológií s cieľom jednoduchšieho ovládania.

Úlohy:

1. Naštudovať jazyky PHP, CSS a JavaScript (AJAX), zoznámiť sa s databázou MySQL.
2. Naštudovať architektúru IS ÚIAM.
3. Vykonať analýzu, špecifikovať požiadavky a navrhnúť štruktúru nového modulu.
4. Programovo realizovať modul v PHP, JavaScript a MySQL.
5. Implementovať vytvorenú aplikáciu do IS ÚIAM.
6. Vykonať analýzu existujúcich modulov.
7. Programovo realizovať zmeny modulov v PHP, JavaScript a MySQL.
8. Overiť funkčnosť modulov.
9. Vypracovať dokumentáciu.

Rozsah práce: 50

Zoznam odbornej literatúry:

1. Resig, J. *JavaScript a Ajax : Moderní programování webových aplikací*. Praha: Computer Press, 2007. 360 s. ISBN 978-80-251-1824-5.
2. Odell, D. *JavaScript. Průvodce programováním ajaxových aplikací*. Praha: Computer Press, 2010. 368 s. ISBN 978-80-251-2733-9.
3. Darie, C. – Brinzarea, B. – Chereches-Tosa, F. – Bucica, M. *AJAX a PHP tvoříme vysoce interaktivní webové aplikace : Profesionálně*. Brno: Zoner Press, 2006. 320 s. ISBN 80-86815-47-1.
4. Gilmore, W J. *Beginning PHP and MySQL: From Novice to Professional*. New York, USA: Apress, 2010. 824 s. ISBN 978-1-4302-3114-1.

Riešenie zadania práce od: 15. 02. 2016

Dátum odovzdania práce: 22. 05. 2016

L. S.

Bc. Martin Krippel

študent

prof. Ing. Miroslav Fikar, DrSc.

vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.

garant študijného programu

Týmto spôsobom by som sa rád poďakoval svojmu vedúcemu diplomovej práce Ing. Ľubošovi Čírkovi, Phd. za odborné vedenie, cenné rady a postrehy, inšpiratívne návrhy , ktoré mi poskytoval počas celého obdobia vypracovávaní tejto práce. Moja vďaka patrí aj všetkým ostatným, ktorí mi ktorýmkoľvek spôsobom pomohli s jej vypracovaním.

Abstrakt

Práca sa zaoberá vylepšovaním informačného systému Ústavu informatizácie, automatizácie a matematiky (ÚIAM). Jej cieľom je vytvorenie nového modulu a úprava existujúcich. Nový modul bude slúžiť na úpravu údajov o predmetoch a možnosťou ich aktualizácie údajmi z akademického informačného systému univerzity (AIS). Existujúce moduly citácií a dochádzky budú doplnené o vhodnú funkcionality a upravené za účelom jednoduchšieho a intuitívnejšieho ovládania. Spomenutých webových aplikácií využívajú trojvrstvovú architektúru, pričom ich jadro je napísané v jazyku PHP, na ukladanie údajov sa využíva databáza MySQL a klientska časť aplikácií je napísaná v JavaScripte s využitím knižnice jQuery. Na komunikáciu medzi klientom a serverom sa často využíva AJAX.

Kľúčové slová: informačný systém; databáza; webová aplikácia; parsovanie údajov

Abstract

The master's thesis aims to improvement of the information system of Institute of Information Engineering, Automation and Mathematics. Its goal is to create new module for the system and to improve existing modules. The new module will be able to edit courses data and will also contain feature of indirect data updating from Academic Information System.of Slovak University of Technology. Existing modules (citations, attendance) will be extended by new suitable functionality and reworked to be simpler and more intuitive. The abovementioned modules are web applications with three-layer architecture, their core is written in PHP, MySQL database is used for storing data and the client-side of application is written in JavaScript with the usage of jQuery library. AJAX technique is often used for client-server communication.

Key words: information system; database; web application; parsing

Obsah

Zoznam obrázkov	11
Zoznam použitých skratiek	12
1 Úvod	13
2 Opis použitých programovacích prostriedkov a technológií	14
2.1 HTML	14
2.2 Apache HTTP Server	14
2.3 PHP	15
2.4 Relačný databázový systém	16
2.4.1 SQL	16
2.4.2 Databáza MySQL	17
2.5 JavaScript, AJAX, jQuery	17
2.6 JSON	19
2.7 jQuery UI Month Picker	19
3 Popis architektúry IS ÚIAM	20
4 Modul predmety	22
4.1 Analýza stavu a špecifikácia požiadaviek	22
4.2 Realizácia modulu	24
4.2.1 Funkcionalita na strane servera	27
4.2.2 Funkcionalita na strane klienta	33
4.2.3 Parsovanie údajov z AIS	34
4.2.4 Integrácia do informačného systému a jazykové verzie	36
4.3 Užívateľská dokumentácia	38
4.3.1 Spustenie	38
4.3.2 Vyhľadávanie predmetov	38
4.3.3 Editovanie predmetu	39
4.3.4 Získanie údajov z AIS	40
4.3.5 Vytvorenie nového predmetu	41
4.3.6 Vymazanie predmetu	41
5 Modul citácie	45
5.1 Analýza stavu a špecifikácia požiadaviek na vylepšenie	45
5.2 Realizácia vylepšení	46
5.2.1 Sprievodca vložení citácie z externého zdroja	46
5.2.2 Parsovanie formátu RIS	48
5.3 Užívateľský opis novej funkcionality	50

6	Modul dochádzka	53
6.1	Analýza stavu a špecifikácia požiadaviek na rozšírenie	53
6.2	Realizácia exportu údajov	54
6.2.1	Vstupné formuláre	55
6.2.2	Export údajov	57
6.3	Užívateľská dokumentácia	59
6.3.1	Prístup	59
6.3.2	Nastavenia exportu	59
6.3.3	Výber mesiaca	60
6.3.4	Výsledok exportu	61
7	Záver	64

Zoznam obrázkov

1	Schéma funkcionality modulu na úpravu predmetov	23
2	Náčrt architektúry aplikácie na úpravu predmetov	25
3	Entitno-relačný diagram časti databázy využívanej modulom na úpravu predmetov	26
4	Grafické znázornenie dotazu hľadajúceho predmety	32
5	Stránka predmetu v katalógu AIS s vyznačenými údajmi na parsovanie . .	35
6	Ukážka slovenskej verzie aplikácie	36
7	Ukážka anglickej verzie aplikácie	37
8	Spustenie modulu predmety	38
9	Vyhľadávanie predmetov	39
10	Získanie údajov predmete z AIS	40
11	Aktualizovanie elementu údajmi z AIS	41
12	Ostatné akcie modulu na úpravu predmetov	42
13	Úvodná časť formulára na editáciu predmetu	43
14	Najspodnejšia časť formulára na editáciu predmetu	44
15	Náčrt štruktúry formátu RIS	49
16	Sprievodca importu bibliografických údajov z <i>Web of Knowledge</i>	50
17	Vkladanie údajov zo súboru	51
18	Vkladanie údajov zo súboru	51
19	Chyba pri načítavaní súboru	52
20	Upozornenie po neúspešnej kontrole formátu dát	52
21	Úspešne načítané údaje	52
22	Formuláre pre export dochádzky pri administrátorskom prístupe	59
23	Formuláre pre export dochádzky pri bežnom prístupe	60
24	Špeciálny kalendár na výber mesiaca v roku (jQuery UI Month Picker) . .	61
25	Útržok tabuľky s údajmi o dochádzke (odpracované časy) za vybraný mesiac v HTML dokumente)	62
26	Útržok tabuľky s údajmi o dochádzke (odpracované časy, príchody, odchody) za vybraný mesiac v HTML dokumente)	62
27	Útržok tabuľky s údajmi o dochádzke za obdobie viacerých mesiacov v HTML dokumente)	63

Zoznam použitých skratiek

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
XHTML	Extensible HyperText Markup Language
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
ÚIAM	Ústav informatizácie, automatizácie a matematiky
PHP	HyperText Preprocessor
XML	Extensible Markup Language
PDF	Portable Document Format
SQL	Structured Query Language
AJAX	Asynchronous JavaScript and XML
JSON	JavaScript Object Notation
UI	User Interface
AIS	Akademický informačný systém Slovenskej technickej univerzity
FCHPT	Fakulta chemickej a potravinárskej technológie
STU	Slovenská technická univerzita
URL	Uniform Resource Locator
DOM	Document Object Model

1 Úvod

V súčasnej dobe sa už bez dobrého automatizovaného informačného systému dá len ťažko zaobiť. Nech sa jedná o takmer ktorékoľvek odvetvie hospodárstva, je potrebné za pomerne krátky čas spracovávať veľké množstvo prichádzajúcich informácií a zároveň vedieť ich aj poskytovať. K masívnemu urýchleniu šírenia údajov samozrejme prispela aj dostupnosť internetu, ktorý sa už bežne používa na každodennú výmenu údajov. Práve kvôli tomuto trendu je asi najvhodnejšie riešenie, aby bol informačný systém založený na webovej platforme a bežal na nejakom serveri. Takmer nepretržitá dostupnosť, prepojenie s databázou a možnosť paralelne spracovávať informácie prichádzajúce z rôznych zdrojov, je len časť výhod, ktorý takáto architektúra poskytuje.

Akademická oblasť samozrejme nie je výnimkou, čo sa týka potreby využívať informačný systém. Univerzity potrebujú registrovať prihlášky na štúdium, mať prehľad o aktuálnom stave študentov, predmetov. Tak ako v iných oblastiach, aj tuto je dôležité spracovávanie osobných údajov, evidencie transakcií, študijných výsledkov a podobne.

Vývoj informačného systému však nie je krátkodobá a v konečnom dôsledku ani jednorazová činnosť. Práve jeho každodenným využívaním sa najlepšie odhaľujú jeho nedostatky, horšie navrhnuté riešenia, chyby, ale na druhej strane aj jeho silné stránky. Taktiež je potrebné počítať s tým, že ako v dnešnej dobe rýchlo napreduje vývoj výpočtovej techniky, tak podobnou rýchlosťou sa posúva vpred aj ich softvérové vybavenie a celkovo využiteľné technológie.

V mojej práci sa zameriavam na vylepšovanie informačného systému ÚIAM. Systém slúži verejnosti, študentom, ale v prvom rade zamestnancom, ktorí ako jediní majú prístup do všetkých častí systému. Práve táto verejnosť neprístupná časť bola predmetom vylepšení. Po oboznámení sa s architektúrou systému sa zaoberám tvorbou nového modulu na úpravu predmetov uložených v databáze systému a snažím sa nájsť riešenie na problém, ktorý predstavuje umožnenie používateľovi aktualizovať údaje o predmete s údajmi, ktoré sa nachádzajú v databáze AIS, do ktorej nemám žiadny prístup. Okrem toho sa ešte budem zaoberať aj niektorými existujúcimi modulmi, ktoré sa budem snažiť vylepšiť za účelom jednoduchšieho ovládania alebo doplnenia vhodnej funkcionality. Pri práci na každom module najskôr zanalyzujem aktuálny stav a špecifikujem si požiadavky, následne z pohľadu programátora opíšem realizáciu špecifikovaných požiadaviek a nakoniec popíšem novú funkcionality z pohľadu budúceho užívateľa.

2 Opis použitých programovacích prostriedkov a technológií

V tejto časti práce opíšem technológie, programovacie jazyky a techniky, ktoré som používal pri svojej práci. Okrem ich opisu aj spomeniem, kde boli využité.

2.1 HTML

HyperText Markup Language je značkovací jazyk, ktorý sa už roky štandardne používa na vytváranie webových stránok. Zapisuje sa vo forme HTML elementov pozostávajúcich zo značiek, ktoré sú uzavreté v lomených zátvorkách. Webové prehliadače čítajú jednotlivé elementy HTML kódu a použijú ho na interpretáciu jednotlivých častí web stránky, ktorú následne zobrazia. Nejedná sa preto o programovací jazyk, ale značkovací. HTML (spolu s CSS) udržiava a štandardizuje World Wide Web Consortium. Prvýkrát vyšiel v roku 1993, je posledná verzia HTML5 vyšla v októbri 2014 po takmer pätnásťročnej prestávke, pretože pôvodne sa predpokladalo, že jazyk bude postupne úplne nahradený jazykom XHTML, k čomu nakoniec nedošlo. [1] V súčasnej dobe sa očakáva ďaleko viac, než len statické zobrazovanie stránok, takže sa predpokladalo, že HTML už zostane len nevyhnutý stavebný kameň webových stránok. Situácia sa ale trochu zmenila príchodom HTML5 a CSS3, ktoré značne rozšírili možnosti dizajnérom stránok. Okrem definovania mnohých nových elementov a umožnenia vytvárať animácie a iné dynamické prvky, pridali aj rôzne nové API a odstránili mnoho zastaralých prvkov. [2]

Bežné HTML aj spolu s CSS som využil pri exporte dochádzky ako jeden z výstupných formátov. V prípade citácií som však využil *File API* pre prácu s lokálnymi súborami.

2.2 Apache HTTP Server

Bežne nazývaný Apache, je najrozšírenejší softvér na obsluhu webového serveru [2]. Vyvíjaný od roku 1995, Apache zohral kľúčovú úlohu v počiatočnom raste celosvetového webu a od roku 1996 je najpopulárnejší HTTP server na svete. Ide o open-source software, ktorý je dostupný pre širokú škálu operačných systémov ako *Unix*, *Linux*, *Microsoft Windows*, *Solaris*, *OS X* a iné. Najbežnejšie je používaný na Unixových systémoch. Medzi jeho hlavné prednosti patrí spomínaná kompatibilita, cena (open-source), porovnateľné vybavenie ako u iných platených HTTP serverov, jednoduchá inštalácia a konfigurácia ako aj nízke systémové požiadavky. [3] Na tomto serveri beží aj informačný systém ÚIAM.

2.3 PHP

PHP je v súčasnosti veľmi populárny open-source skriptovací jazyk, ktorého skripty sú vykonávané na strane serveru. Začiatkom tisícročia sa stal asi najpoužívanejšou platformou na svete. Jazyk je určený najmä na vytváranie dynamických webových stránok, ale používa sa aj všeobecne na vývoj rôznych aplikácií a môže byť vnorený priamo do HTML. Skratka PHP je rekurzívny akronym pre PHP: Hypertextový Preprocesor, keďže pôvodný význam skratky bol Personal Home Page. Vývoj PHP začal v roku 1994, keď Rasmus Lerdorf vytvoril v jazyku C užívateľské rozhranie na správu svojej domácej webstránky [4]. Neskôr ho rozšíril o schopnosť práce s webovými formulármi a o modul, ktorý umožňoval komunikáciu s databázou. Túto implementáciu nazval Personal Home Page/Forms Interpreter alebo skratkou PHP/FI. Lerdorf nemal žiadny úmysel vytvoriť nový programovací jazyk, avšak jeho obľuba u programátorov začala rásť a začal sa formovať vývojársky tím, ktorý v roku 1997 vydal vylepšený PHP/FI 2. O rok na to vyšlo PHP 3, ktoré používalo prepísaný parser, ktorý vyvinuli Zeev Suraski s Andim Gutmansom. Táto verzia zároveň priniesla zmenu významu skratky jazyka na terajší PHP: Hypertext Preprocessor. Od verzie 4 začal jazyk používať nové kompletne prepísané jadro PHP, engine nazývaný Zend. Vo verzii PHP 5 bol pridaný Document Object Model modul, ktorý slúži na prácu s XML a HTML dokumentami. Od roku 2001 je PHP vyvíjané skupinou The PHP Group.

Keďže syntax jazyka PHP sa podobá na jazyky ako C, Perl a Java, nie je ťažké pre programátora začať pracovať v PHP. Jazyk je ešte obohatený o ďalšie vlastnosti, akou je napr. sada príkazov na integráciu a ovládanie databázy (ako napr. MySQL, SQLite a iné) cez PHP kód. Tieto spomenuté vlastnosti dávajú vývojárom do rúk mocné nástroje na tvorbu dynamických webových stránok. PHP je multiplatformový jazyk, ktorý bude bežať na väčšine HTTP serverov ako aj operačných systémov. Veľmi obľúbené je použitie PHP ako modulu na spomínanom web serveri Apache, najbežnejšie na systémoch Unix, ale aj Microsoft Windows. PHP beží na strane serveru a ponúka možnosť stiahnuť určité informácie zo servera a zobrazíť ich na stránke v prehliadači. Taktiež môže prijímať údaje zo strany užívateľa a následne ich spracovať na serveri. Tieto možnosti sú veľmi rozmanité, môže ísť napr. o komunikáciu cez užívateľský formulár, synchronizáciu času, operácie s databázami, prácu s PDF dokumentami alebo obrázkami, či pomocou spomínaného modulu vytvárať, načítavať a aj upravovať časti alebo celé HTML dokumentu alebo XML súboru. Ako bolo už spomenuté, PHP sa používa na tvorbu dynamických webových stránok. V praxi to funguje tak, že v aktuálnej HTML stránke sa odošle požiadavka na spustenie *php* skriptu. Prehliadač pošle túto požiadavku na server, keďže PHP sa nachádza na strane serveru a je vykonávaný priamo za chodu. Keď server zistí, že je potrebné vykonať skript, pošle požiadavku PHP interpreter, ktorý ho vykoná a server pošle výsledok (už HTML kód) späť prehliadaču. Tým pádom sa stránka zmení počas chodu a preto ju nazývame dynamická. PHP kód na stránke nie je viditeľný, užívateľ môže vidieť len jeho výsledok vo forme upraveného HTML kódu stránky. [4][5]

Na PHP je založená celá funkcionálnosť systému na strane servera, preto som aj ja jadro modulov písal v tomto jazyku.

2.4 Relačný databázový systém

Databáza je množina údajov uložená na pamäťovom médiu, ktoré sú väčšinou usporiadané v pevnej štruktúre. Databázy sa delia na hierarchicky zostavené, viacrozmerné a relačné. V súčasnosti sa najviac vyskytujú relačné databázy. [4] V relačnom modeli tabuľky predstavujú relácie, stĺpce sú atribúty, pod riadkami rozumieme usporiadané n-tice a konkrétne hodnoty uložené v tabuľkách sú tým pádom hodnoty atribútov. Do tabuľky sa zvyknú ukladať informácie o jednom druhu objektov. Aby sa dalo s tabuľkami pracovať, ich obsah musí byť jednoznačne identifikovaný. To znamená, že každý stĺpec musí niesť svoj jedinečný názov. Pri práci s databázou často potrebujeme získať alebo upraviť údaje len pre určitý počet záznamov (riadkov). V takom prípade potrebujeme jednoznačne identifikovať dané záznamy a preto je vhodné, aby tabuľka obsahovala primárny kľúč. Ide o atribút, ktorý je pre každý riadok tabuľky unikátny. Bežným príkladom primárneho kľúča sú identifikačné čísla (id). Okrem názvu musia mať atribúty definovaný ešte dátový typ. Ten určuje, aký typ údajov môžeme do daného stĺpca zapisovať. Najčastejšie sa používajú celé čísla, textové reťazce (*string*), reálne čísla, časový údaj, dátum alebo ich kombinácia, prípadne logické hodnoty (*boolean*). Databázy zvyčajne obsahujú viac tabuliek, pričom každá má definované svoje meno. Každá tabuľka obsahuje ucelené údaje, to ale nevylučuje možnosť, že údaje medzi jednotlivými tabuľkami nejako nesúvisia. Tieto vzťahy medzi tabuľkami sú relácie relačnej algebry a najbežnejšie sa používa vzťah 1:N, čo znamená, že jeden záznam v prvej tabuľke súvisí s viacerými záznamami v druhej tabuľke. V praxi je tento spôsob bežný, pretože, ak by sa tieto údaje uložili do jednej tabuľky, mohli by vzniknúť problémy s definovaním dátového typu (napr. jeden záznam by potreboval uložiť viac hodnôt do jedného stĺpca, čo nie je pri číselných typoch dát možné) alebo by veľkosť databázy zbytočne narastala v dôsledku opakujúcich sa záznamov. Tieto vzťahy sa teda riešia tak, že v prvej tabuľke si určíme tzv. cudzí kľúč, ktorý bude v druhej tabuľke definovaný ako primárny kľúč. Potom nebude problém priradiť všetky údaje (prípadne aj viac záznamov) z druhej tabuľky práve jednému záznamu z prvej tabuľky pomocou tohto cudzieho kľúča. Kým každá tabuľka môže mať viacero cudzích kľúčov a byť tak prepojená s viacerými tabuľkami naraz, primárny kľúč je v každej tabuľke vždy len jeden. [6]

2.4.1 SQL

Structured Query Language (SQL) je dotazovací jazyk slúžiaci na komunikáciu s databázou. Je podporovaný väčšinou databáz, pričom všetky podstatné príkazy používajú rovnakú alebo podobnú syntax vo všetkých databázach. SQL sa nepoužíva len na dota-

zovanie, ale aj k priamej manipulácii s dátami uloženými v databáze. Umožňuje s nimi operácie ako je ich upravovanie, mazanie alebo pridávanie nových dát. Pomocou tohto jazyka je možné aj vytvárať nové databázy a ich tabuľky alebo meniť štruktúru, prípadne vymazať existujúce. Databázy sa používajú v podstate už všade, kde sa pracuje s väčším počtom dát a jednou z najbežnejších operácií je načítanie potrebných dát z databázy. Na tieto operácie sa SQL používa bežne, v prípade veľkých databáz môžeme povedať, že prakticky bez výnimky. Výhoda SQL je, že ho možno použiť priamo v iných programovacích jazykoch, pokiaľ majú zabezpečenú schopnosť práce s databázou. V prípade PHP je táto vlastnosť už priamo zabudovaná, takže je hneď možné používať SQL dotazy cez PHP skript. SQL sa používa na prácu s relačnými databázami, kde sú spomínané spojenia viacerých tabuliek tvorené práve príkazmi jazyka SQL. [4]

2.4.2 Databáza MySQL

MySQL je najpopulárnejšia open source relačná SQL databáza (využívajúca SQL jazyk). Taktiež patrí aj medzi najpoužívanéjšie databázy na svete. MySQL je rýchla a spoľahlivá databáza umožňujúca spracovanie veľkého množstva dát. Pôvodne bola vyvinutá na rýchle ovládanie veľkých databáz, v dnešnej dobe ponúka aj široké spektrum užitočných funkcií. [4] Vlastnosti ako ľahká použiteľnosť pre užívateľa, rýchlosť a bezpečnosť je obľúbená voľba databázy pre použitie vo webových aplikáciach. Väčšina voľných alebo open source projektov, ktoré potrebujú využívať plne funkčný relačný databázový systém, používa ako riešenie MySQL. Okrem open source verzie existuje aj viacero platených verzií s licenciou, ktoré poskytujú dodatočné funkcie a je možné ich použiť na komerčné účely. Viacero známych aplikácií alebo veľkých web stránok využíva MySQL ako databázu. PHP jazyk priamo obsahuje modul pre prácu s MySQL databázou, ale je možné získať aj open source rozhranie pre iné jazyky, ako napr. C a to aj pre viaceré platformy. [7]

Systém používa tento typ databázy na ukladanie údajov, s databázou som komunikoval prostredníctvom SQL dotazov.

2.5 JavaScript, AJAX, jQuery

JavaScript je dynamický, objektovo orientovaný, skriptovací jazyk, ktorý je v súčasnosti súčasťou webových prehliadačov. JavaScript sa môže vkladať priamo do HTML kódu stránky, beží na klientskej strane aplikácie a umožňuje vytvárať dynamické akcie na stránke, asynchrónnu komunikáciu so serverom alebo úpravu už zobrazeného webového dokumentu. Medzi najbežnejšie využitie patrí ovládanie interaktívnych prvkov formulárov ako tlačidlá a textové polia, alebo potom tvorba grafických efektov alebo animácií na stránke. JavaScript vytvoril v roku 1995 Brendan Eich, ktorý vtedy pracoval pre spoločnosť Netscape, ktorá v tomto čase súperila s Microsoftom o priazeň užívateľov na

poli webových technológií a platforiem. JavaScript mal byť určený hlavne pre neprofesionálnych programátorov ako nenáročný interpretovaný programovací jazyk, ktorý mal konkurovať podobným jazykom ako napr. Visual Basic od Microsoftu. Jazyk začal nabrať na obľube a už v roku 1996 si ho osvojil aj Microsoft a implementoval do vtedajšej verzie prehliadača Internet Explorer pod názvom JScript (kvôli autorským právam). V roku 1997 bol štandardizovaný asociáciou ECMA a táto verzia dostala názov ECMAScript. JavaScript však nebol len to, čo definoval ECMAScript, okrem toho ešte obsahuje Document Object Model a Browser Object Model. Syntakticky je podobný jazykom tzv. C rodiny. Nie je to však úplný programovací jazyk a to z dôvodu, že beží až na strane užívateľa a preto má z bezpečnostných dôvodov obmedzenia ako napríklad neschopnosť pracovať so súborami. [8]

AJAX alebo celým názvom asynchrónny JavaScript a XML je skupina techník vývoja webu bežiaca na strane klienta za účelom vytvorenia asynchrónnych webových stránok. Takéto stránky si potom môžu vymieňať dáta so serverom na pozadí bez nutnosti načítať a prekresliť celú stránku. Medzi zásady používania tejto techniky patrí snaha o minimalizáciu prenášaných dát, príliš nerozptyľovať užívateľa prebiehajúcou požiadavkou a iné. Napriek svojmu názvu nemusí vždy využívať XML a nie všetky požiadavky musia byť výlučne asynchrónne. V skutočnosti sa nejedná o jednu technológiu, ale o celú skupinu, ktorá zahŕňa predovšetkým HTML s CSS na prezentáciu stránky, DOM na dynamické zobrazovanie, XML na vymieňanie dát, XMLHttpRequest (XHR) na asynchrónnu komunikáciu a ďalšie. JavaScript je použitý na vzájomné prepojenie týchto technológií. AJAX ako pojem prvýkrát verejne použil v roku 2005 Jesse James Garret, ale metódy na asynchrónnu komunikáciu vznikali už oveľa skôr, pričom v počiatkoch bola v tomto smere najväčším priekopníkom firma Google. [9]

jQuery je open source, JavaScript knižnica vyvinutá za primárnym účelom zjednodušiť tvorbu skriptov bežiacich na strane užívateľa najmä skráteným zápisom a normalizáciou rozdielov medzi prehliadačmi. V súčasnosti je to najpopulárnejšia JavaScript knižnica, ktorá je hojne využívaná aj na najväčších a najznámejších stránkach na svete. Medzi jej výhody patrí rýchlosť, malá veľkosť, dostupnosť dokumentácia a široká komunita vývojárov. Popularitu získala hlavne kvôli zjednodušenému zápisu navigácie cez DOM, ľahkej tvorbe animácií, výraznému uľahčeniu využívania AJAX technológie, možnosti hociktorú premennú alebo objekt zabaliť do jQuery objektu a ďalšie. Základná filozofia knižnice je spraviť viac s menej rozsiahlym kódom. [10]

Takmer celú klientskú časť aplikácií som robil v jQuery zápise, pričom na komunikáciu som serverom som pomerne často využíval AJAX techniku.

2.6 JSON

Ide o odľahčený formát určený na transfer dát. Pre ľudí je pomerne ľahko čitateľný a zapisovateľný, ale zároveň aj stroje ho dokážu ľahko generovať alebo analyzovať. Tým, že je to textový formát, ktorý nie je závislý na žiadnom programovacom jazyku, je vhodným kandidátom na prenos zložitejších štruktúr textových dát, ktoré sa bežne prenášajú pri webových aplikáciach. Je založený na dvoch štruktúrach, jednou je objekt, čo množina párov v tvare *názov: hodnota*. Druhá možná štruktúra je pole, čo je usporiadaná kolekcia hodnôt. Čo sa týka hodnôt, tie môžu byť buď ďalšie objekty a polia, prázdna (*null*), pravdivostná hodnota, reálne číslo alebo reťazec. Reťazec používa na uzavretie úvodzovky, takže v prípade ich použitia vnútri reťazca je potrebné použiť spätnú lomku ako únikovú sekvenciu.[11]

V práci som využil tento formát na prenášanie údajov o predmetoch cez AJAX požiadavky.

2.7 jQuery UI Month Picker

Je to open source projekt založený na existujúcom kalendáriku *jQuery UI Datepicker*. Cieľom projektu bolo vytvoriť element podobný kalendáru, ktorý by umožňoval používateľovi vyberať iba mesiace v roku. Po kliknutí na text s rokom sa miesto mesiacov budú zobrazovať roky a potom je možné sa tlačidlami posúvať o periódu 12 rokov. Umožňuje vývojárovi definovať pomerne široký počet nastavení akými sú definovanie hraníc z ktorých je možné vyberať mesiace, podporuje možnosť definovania si vlastnej jazykovej verzie, okrem toho obsahuje funkcie na validovanie dátumu alebo zachytávanie udalostí. Používa rovnaký štýl ako bežný kalendárik a v prípade potreby je možné ho zmeniť. Hlavným vývojárom tohto rozšírenia je Benjamin Albert. [12]

Toto rozšírenie som doplnil do informačného systému, pretože pri exporte dochádzky som potreboval element, pomocou ktorého budem môcť vyberať priamo mesiace v roku a klasický kalendár pre jQuery neumožňoval také nastavenie.

3 Popis architektúry IS ÚIAM

Informačný systém je možné chápať ako súbor viacerých technológií, ktoré slúžia na poskytovanie, získavanie a uskladnenie údajov prevažne o ústave. Práve na základe použitých technológií je možné načrtnúť základnú architektúru systému. Vo všeobecnosti môžeme povedať, že má klasickú trojvrstvovú architektúru, ktorá pozostáva z prezentačnej, logickej a dátovej vrstvy. Logická vrstva obojsmerne komunikuje so zvyšnými dvoma vrstvami.

Úlohou dátovej vrstvy je uschovávať všetky potrebné dáta a v prípade potreby ich poskytnúť logickej vrstve. V tomto systéme sú údaje ukladané do relačnej databázy MySQL, ktorá beží spolu s logickou vrstvou systému na Apache serveri. Systém používa desiatky tabuliek, ktoré sú pomenované podľa toho, na aký účel sú ich dáta využívané. Niektoré tabuľky slúžia ako číselníky obsahujúce možnosti, ktoré môže nejaký údaj nadobúdať. Údaje slúžia na najrozličnejšie účely, môžu to byť texty priamo zobrazované užívateľovi, údaje o osobách, informácie o predmetoch, programoch, bibliotéka, interné hodnoty ale aj prihlasovacie údaje alebo referencie na binárne súbory, ktoré sú uložené v nejakom adresári na serveri. Na prácu s MySQL databázou má systém v logickej časti vytvorené zdieľané funkcie, jednak na vyťahovanie dát a potom aj na zmenu akou môže byť vkladanie nových údajov, úprava alebo mazanie existujúcich. Toto umožňuje jednoduchú prácu s databázou v ktorejkoľvek časti systému pomocou SQL dotazov.

Logická vrstva je z hľadiska tejto práce najzaujímavejšia, pretože je jadrom celého systému. Je napísaná kompletne v jazyku PHP a má na starosti celú funkcionálnosť systému na strane servera a taktiež generuje majoritnú časť prezentačnej vrstvy systému. Systém je navrhnutý tak, že je možné ho rozdeliť na dve časti. Prvá je verejne dostupná časť, ktorá slúži predovšetkým na zobrazovanie rozličných informácií o ústave a dostane sa do nej každý návštevník webovej stránky. Do tejto časti sú aj zahrnuté študentské aplikácie akou je napr. prihlasovanie na semestrálne projekty alebo zobrazovanie rozvrhu hodín. Z väčšej časti sa teda táto časť na prvý pohľad javí ako statický web, avšak pokiaľ sa do systému prihlási užívateľ s dostatočnými oprávneniami, je schopný upravovať obsah týchto stránok, čo znamená, že aj táto časť webu je generovaná dynamicky. Druhá časť systému je editačná (nazývaná aj skrytá), ktorá je dostupná používateľovi len po prihlásení. Užívateľské účty majú len zamestnanci ústavy. Táto časť pozostáva z modulov, ktoré slúžia predovšetkým práve na pridávanie a úpravu údajov v systéme. Jedná sa napr. o citácie, publikácie, rozvrh hodín, evidenciu dovolení a ďalšie. Tieto moduly sú uložené v samostatných skriptoch, ktoré sú prilinkované k jadru systému. Každý platný užívateľ má priradený druh účtu, ktorý rozhoduje o jeho oprávneniach. Preto aj funkcionálnosť niektorých modulov môže byť rôzna pre bežného užívateľa a pre administrátora. Platnosť prihlásenia je realizovaná cez reláciu (*session*). Relácia má v PHP svoju super-globálnu premennú, z ktorej je potom možné získať práve prihláseného užívateľa a overiť tak jeho

oprávnenia. Relácia sa v tomto systéme používa aj uchovanie aktuálne nastavenej jazykovej verzie systému, čo sa tiež dá využiť, pokiaľ bude potrebné rozhodovanie na základe jazykov.

Prezentačná vrstva systému je zobrazovaná na strane klienta vo webovom prehliadači, keďže ide o webovú aplikáciu. Pozostáva zo statického HTML kódu, ktorý prijme prehliadač zo servera (logickej vrstvy) a z dynamickej časti, ktorá je v tomto systéme zabezpečená JavaScriptom. Ten v niektorých prípadoch aj mení štruktúru stránky, robí rôzne kontroly, vytvára pre užívateľa interaktívnejšie rozhranie a keď je to potrebné, vie komunikovať so serverom bez toho, aby bolo potrebné opätovne načítať celú stránku. Systém v sebe zahŕňa aj knižnicu jQuery, ktorá v prípade vývoja modulov v mnohom uľahčuje prácu pri tvorbe klientskej časti aplikácie.

Výstupy informačného systému sú teda webové stránky, na ktorých sú zobrazované želané údaje. Vstupy môžu prichádzať priamo od užívateľa prostredníctvom odoslania požiadavky cez webovú stránku, alebo aj automatizované, ktoré môže na server odoslať spomínaný JavaScript alebo aj zariadenia, ktoré komunikujú so systémom inými spôsobmi, napr. čítačka kariet používaná na registrovanie dochádzky zamestnancov.

4 Modul predmety

Tento modul informačného systému slúži na spravovanie a údržbu údajov o predmetoch zabezpečovaných ÚIAM. Pod údržbou rozumieme možnosť tieto údaje prezerať a upravovať, ale aj pridávať úplne nový predmet alebo vymazať existujúci. V neposlednom rade ponúka aj čiastočne automatizovanú možnosť získať informácie o aktuálne upravovanom predmete z AIS a následne ich vypísať používateľovi. Ten ich potom môže porovnať s aktuálnou hodnotou uloženou v informačnom systéme a v prípade potreby jednoducho nahradiť hodnotou získanou z AIS. Ako úložisko údajov o predmetoch slúži relačná databáza MySQL.

4.1 Analýza stavu a špecifikácia požiadaviek

Informačný systém ÚIAM doposiaľ neobsahoval žiadnu možnosť úpravy predmetov. Nedá sa však povedať, že by neobsahoval údaje o samotných predmetoch. Tieto údaje sa využívajú v iných častiach systému už dlhý čas, pretože bez nich by webová stránka nemohla poskytovať mnohé základné funkcionality. Hovorím najmä o verejne dostupnej časti stránky, ktorá je dostupná každému návštevníkovi webu. V tejto časti sa jedná predovšetkým o zobrazenie rozvrhu hodín alebo prehľad predmetov. V oboch sekciách sa tieto údaje filtrujú podľa rôznych kritérií, akými sú napríklad študijné programy, oddelenie zabezpečujúce daný predmet, ročník a iné. V editačnej časti webovej stránky údaje o predmetoch využíva modul na tvorbu rozvrhu. Pri tvorbe nového záznamu v rozvrhu je samozrejme nevyhnutné špecifikovať, o ktorý predmet sa jedná. Všetky spomenuté moduly však tieto údaje o predmetoch potrebujú z databázy len čítať, čo znamená, že predpokladajú správnosť uložených údajov. Ak by bolo treba nejaký údaj zmeniť, odstrániť alebo naopak pridať, bolo by nevyhnutné túto zmenu vykonať pomocou priameho prístupu do databázy.

Potreba zmeniť tieto údaje môže vzniknúť z viacerých príčin. Jedným z dobrých príkladov bola nedávna akreditácia nových programov, kedy vznikli nové predmety a mnohé staré boli zmenené alebo úplne zanikli. Takáto situácia si vyžiadala rozsiahlejšie úpravy v údajoch o predmetoch. Niekedy však môže ísť aj o úpravy v menšom rozsahu, napríklad sa pri konkrétnom predmete zistí nejaký preklep alebo iná nepresnosť, ktorú je potrebné opraviť. Ďalší príklad môže byť čiastková aktualizácia údajov, akými sú napríklad osnova predmetu alebo odporúčaná literatúra pre daný predmet.

Nech už bola potreba upraviť údaje o predmete rôzna, bolo potrebné ju nahlásiť administrátorovi, ktorý mal priamy prístup k databáze a ten následne v nej musel vykonať požadované zmeny. Takýto postup bol však pomerne zdĺhavý a nepraktický. Preto prvá požiadavka bola vytvoriť webovú aplikáciu, ktorá by umožňovala užívateľovi vyhľadávať predmety dostupné v databáze informačného systému ÚIAM a následne zobraziť údaje o

zvolenom predmete s možnosťou vykonať v nich úpravy a tie následne uložiť. Aplikácia by bola dostupná ako ďalší modul na stránke ústavu, čím by odpadala možnosť riešiť všetky úpravy cez administrátora.

Keďže celá univerzita vrátane FCHPT používa AIS, údaje o všetkých predmetoch fakulty a teda aj ústavu, sú v ňom uložené. Preto ďalšia požiadavka na modul bola, aby sa dalo podľa možnosti čo najjednoduchšie aktualizovať údaje o predmetoch údajmi z AIS. Katalóg predmetov STU je verejne dostupný na stránkach AIS, avšak do jeho databázy nemá ÚIAM žiadny prístup. Manuálne kontrolovanie a prepisovanie neprichádza do úvahy, pretože je to zdĺhavý proces čo i len pre jeden predmet, navyše vôbec by to nespĺňalo požiadavku aspoň čiastočnej automatizácie. Využijem teda skutočnosť, že stránky o predmetoch, ktoré obsahujú väčšinu požadovaných údajov, sú verejne dostupné a z nich sa pokúsím získať potrebné dáta.



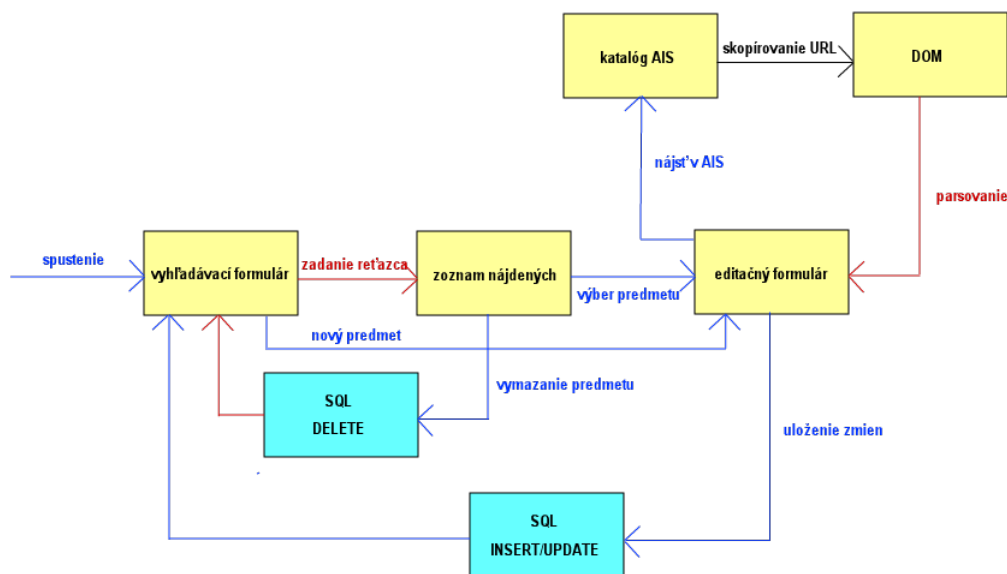
Obr. 1: Schéma funkcionality modulu na úpravu predmetov

Na obrázku 1 som znázornil požiadavky na funkcionality modulu. Aplikácia bude obojsmerne komunikovať s používateľom a s databázou informačného systému ÚIAM. Táto komunikácia je znázornená modrými šípkami. V prípade komunikácie s užívateľom sú možné vstupy a následné výstupy zobrazované prostredníctvom webového prehliadača, v prípade komunikácie s databázou informačného systému si aplikácia vyžiada konkrétne údaje pomocou SQL dotazov a údaje obdrží vo forme nejakého dátového typu, najčastejšie asociačného poľa v jazyku PHP. Prerušovaná šípka medzi aplikáciou a AIS znázorňuje, že tento druh komunikácie nie je plne automatizovaný a tiež to, že zdroj požadovaných informácií je externý a odpoveď nie je priama.

4.2 Realizácia modulu

Po zistení požiadaviek a návrhu funkcionality mohla začať druhá fáza vývoja a tou bola samotná realizácia, čiže naprogramovanie jednotlivých častí modulu. Bude sa jednať o aplikáciu typu klient-server. Pri tejto úlohe som sa snažil dodržať viaceré ciele. Prvý bol správne implementovať modul do existujúceho informačného systému tak, aby používal podobnú funkcionality ako z užívateľského, tak aj z programátorského hľadiska. Pod užívateľskou funkcionality sa rozumie to, že do aplikácie sa bude dať dostať rovnakým spôsobom ako do ostatných existujúcich modulov, ovládanie a výber akcií bude riešené v podobnom štýle ako pre iné moduly. Aplikácia samozrejme nesmie narušiť ani dizajn webových stránok. Programátorskou funkcionality som myslel to, že modul by mal podľa možností čo najviac využívať zdieľané funkcie a zdroje. Mojm ďalším cieľom bolo postupne upravovať kód modulu tak, aby bolo podľa možností čo najľahšie ho modifikovať v budúcnosti v prípade nových požiadaviek. Hovorím najmä o formulári, v ktorom sa nachádzajú elementy na úpravu jednotlivých údajov o predmete. Chcel som, aby pre správcu informačného systému bolo pomerne jednoduché pridať ďalší element, prípadne upraviť, presunúť alebo odstrániť existujúci bez toho, aby bolo nutné upravovať kód na mnohých miestach alebo bez toho, aby bolo potrebné prepisovať celú funkciu na generovanie tohto formulára. A mojm posledným cieľom bolo minimalizovať komunikáciu medzi serverom a klientom, aby sa používala len vtedy, keď je to naozaj nevyhnutné.

Pred samotným programovaním som si ešte vytvoril plánovanú architektúru aplikácie, ktorá špecifikuje jednotlivé časti modulu a ich vzájomné prepojenie. Náčrt architektúry sa nachádza na obrázku 2. Šípky predstavujú akcie, zvyčajne vyvolané používateľom. Akcia je popísaná textom pri šípke. Farba šípky rozlišuje, o aký typ akcie sa jedná. Modré šípky znázorňujú prechod na inú stránku prostredníctvom odkazu alebo odoslaním bežného formulára. Červené šípky znamenajú AJAX požiadavky odoslané JavaScriptom na server, pričom odpoveď je následne priamo vložená do aktuálne zobrazenej stránky. Čierna šípka hovorí o tom, že daná akcia musí byť manuálne vykonaná používateľom. Výsledky akcií sú znázornené obdĺžnikmi a obsahujú aj krátky popis výsledku. Obdĺžniky so žltou výplňou sú výsledky vo forme HTML stránok, kým tyrkysové predstavujú vykonanie SQL dotazu v databáze.



Obr. 2: Náčrt architektúry aplikácie na úpravu predmetov

Ako som už pri analýze pôvodného stavu spomenul, údaje o predmetoch v databáze už existovali, preto bolo ešte potrebné sa oboznámiť aj so štruktúrou databázy, resp. s časťou, ktorá obsahuje tieto údaje. Celú štruktúru spolu aj s relačnými vzťahmi, ktorú využíva táto aplikácia, som znázornil na obrázku 3. Z neho vidno, že primárna tabuľka s názvom *courses* obsahuje v sebe záznamy o všetkých predmetoch, pričom každý predmet je v tejto tabuľke je jeden riadok s unikátnym identifikátorom. Tento identifikátor je pomenovaný *courses_id* a ide o primárny kľúč tejto tabuľky. Tabuľka je vo vzťahu *N:1* k piatim ďalším tabuľkám, ktoré v podstate slúžia ako číselníky. Na základe číselnej hodnoty daného stĺpca v tabuľke *courses*, ktorý je zároveň primárnym kľúčom pre daný číselník je potom z neho možné vytiahnuť ďalšie údaje. Zvyčajne to bývajú textové reprezentácie zvolenej číselnej hodnoty. Týchto päť tabuľky vystupujúce ako číselníky sú *department_ais*, ktorá obsahuje zoznam vybraných oddelení na fakulte, *courses_option*, obsahujúca rozdelenie predmetov na základe voliteľnosti, *courses_form*, v ktorej sú možnosti, akými môže byť predmet vyučovaný, *courses_type_evaluation*, kde sú spôsoby ukončenia predmetu a posledná *courses_language*, v ktorej sa nachádzajú jazyky, v ktorých môže byť predmet vyučovaný. Okrem toho je tabuľka *courses* ešte v relačnom vzťahu k dvom ďalším tabuľkám *program_ais* a *person_ais*, ktoré sú v podstate tiež číselníky obsahujúce údaje o študijných programoch, resp. ľuďoch. Keďže jeden predmet môže byť vyučovaný vo viacerých programoch a teoreticky môže mať aj viac garantov (čo sa však v praxi nezvykne stávať a ani tento modul s tým neráta), tieto vzťahy sú typu *M:N*. Na takýto vzťah je potrebné vytvoriť ďalšiu asociačnú tabuľku, ktorá potom obsahuje primárne kľúče z oboch spájaných tabuliek. Pre programy je vytvorená pomocná tabuľka *courses_program*, ktorá

obsahuje primárne kľúče *courses_id* a *program_ais_id* zo spájaných tabuliek. Pre účely nového modulu som do nej ešte pridal jeden nový stĺpec s názvom *courses_semester*, ktorý bude hovoriť, v ktorom semestri sa bude predmet v danom programe vyučovať. Je to z toho dôvodu, že ten semester môže byť rôzny pre jednotlivé programy. Pre garanta predmetu je vytvorená pomocná tabuľka *courses_garant*, ktorá obsahuje primárne kľúče *courses_id* a *person_ais_id* zo spájaných tabuliek.



Obr. 3: Entitno-relačný diagram časti databázy využívanej modulom na úpravu predmetov

Samotnú realizáciu možno rozdeliť na dve časti, ktoré teraz bližšie opíšem v nasledujúcich častiach práce. Rozdelenie je založené na tom, v ktorej časti aplikácie sa dané operácie vykonávajú.

4.2.1 Funkcionalita na strane servera

Slúži predovšetkým na prácu s databázou a generovanie počiatočného obsahu a výzoru stránky. Jadro aplikácie sa nachádza v skripte s názvom *iam_function_course.php*. V ňom sa nachádza funkcia *courses_edit*, ktorá inicializuje modul a zobrazuje jeho základnú stránku. Funkcia sama o sebe neprijíma žiadne argumenty a jej návratová hodnota je HTML kód stránky. Rozhoduje sa však na základe parametrov poslaných metódou *POST*. Pokiaľ je zadefinovaný parameter *id_del* a na základe hodnoty identifikátora sa nájde príslušný predmet, bude vymazaný z tabuľky *courses* a pokiaľ nenastane chyba, tak následne aj všetky príslušné záznamy z vyššie spomínaných pomocných tabuliek *courses_program* a *courses_garant* pomocou SQL dotazu *DELETE*. V prípade, že sa nájde zadefinovaný parameter *send* s hodnotou *Zrušiť*, aplikácia sa vráti na úvodnú stránku. Tento prípad nastane, ak užívateľ opustí editačný formulár bez uloženia zmien. Ďalší prípad je, keď existuje zadefinovaný parameter *save* s hodnotou *1*. Znamená to, že užívateľ uložil zmeny v editačnom formulári. V tomto prípade by sa samozrejme medzi parametrami metódy *POST* mali nachádzať aj hodnoty jednotlivých polí formulára. Tieto parametre majú rovnaký názov ako stĺpce v tabuľke *courses*. Pri parametri *courses_date* sa použije aktuálny dátum a čas, keďže ide o dátum poslednej úpravy. Len parameter s názvom *courses_garant* bude uložený do rovnomennej tabuľky, pričom jeho hodnota sa uloží do stĺpca *person_ais_id* a do stĺpca *courses_id* sa vloží identifikátor riadku z tabuľky *courses* vrátený ako odpoveď MySQL modulom v prípade úspešného zápisu do tejto tabuľky. To znamená, že táto operácia sa nevykoná v prípade neúspešného zápisu do hlavnej tabuľky. To isté platí aj pre ukladanie do tabuľky *courses_program*, ktoré je však riešené trochu iným spôsobom. Táto odlišnosť je spôsobená práve tým, že predmet môže mať priradených viacero programov a v každom programe môže byť vyučovaný v odlišnom semestri. Najskôr sa teda oddelia parametre s názvami *courses_program_X* a *courses_semester_X*, kde *X* je celé číslo začínajúce od 0 a udáva, o koľký program a k nemu patriaci semester sa jedná. Keďže užívateľ môže programy vo formulári pridávať, odoberať a meniť, miesto nejakého zložitého vyhľadávania v tabuľke, ktoré by muselo pokryť všetky možné prípady, som zvolil prístup, že najskôr sa vymažú všetky záznamy pre upravovaný predmet a potom sa nanovo vložia aktuálne vybrané pomocou dotazu *INSERT*. Ukladanie do ostatných tabuliek predpokladá pridanie alebo úpravu jedného riadku, preto sa na základe prítomnosti parametra *edit_id* rozhodne, či sa použije dotaz *UPDATE* na úpravu existujúceho alebo dotaz *INSERT* na pridanie nového záznamu. Následne aplikácia prejde na svoju základnú stránku a vymaže tak parametre metódy *POST*. Tým sa zabráni opätovnému ukladaniu do databázy pri prípadnom obnovovaní stránky v prehliadači.

Táto prvá časť funkcie kontrolovala na základe prítomnosti určitých parametrov metódy *POST*, či má vykonať nejakú akciu na pozadí. Druhá časť funkcie má na starosti rozhodovanie, akú stránku vykresliť. V podstate existujú dve hlavné možnosti. Prvou je základná stránka aplikácie. Tá obsahuje okrem nadpisu a základného popisu formulár na výber predmetu. Pri úvodnom načítaní stránky tento formulár obsahuje element, ktorý spustí funkciu na tvorbu editačného formulára pre nový predmet a textové pole, ktoré je viazané na JavaScript funkciu, ktorá odošle AJAX požiadavku, ktorej výsledok bude HTML kód zoznamu nájdených predmetov spolu s elementmi na ich úpravu alebo vymazanie. Tento HTML kód sa vloží do dopredu vytvoreného *div* elementu s identifikátorom *found_courses*. Okrem toho formulár obsahuje dva skryté (*hidden*) elementy počiatočne bez hodnoty. Jeden má názov *id_del* a v prípade odstránenia predmetu sa do neho priradí jeho kód. Funkcionalita vymazania je už popísaná vyššie. Druhý má názov *courses_code* a jeho využitie popíšem na nasledujúcom odseku. Vránci výsledku vyššie spomínanej AJAX požiadavky môže vzniknúť ešte jeden skrytý element s názvom *search_courses*, ktorý ak obsahuje hodnotu, tak spôsobí, že sa automaticky vyplní hodnota spomínaného textového poľa a znovu prebehne spomínaná AJAX požiadavka. Je to použité pre prípad, že by sa užívateľ rozhodol zmazať nejaký predmet a následne akciu buď potvrdil alebo zrušil v zobrazenom modálnom okne. V prípade potvrdenia sa po vymazaní predmetu vďaka tejto funkcionalite znovunačíta zoznam nájdených predmetov a vymazaný medzi nimi už nebude. To isté sa v podstate stane aj v prípade zrušenia akcie, pre tento prípad to má taký zmysel, že užívateľovi ostanú zobrazené výsledky posledného vyhľadávania.

Druhá možnosť je, že funkcia vytvorí samotný editačný formulár. Tento prípad nastane, pokiaľ sa v parametroch metódy *POST* nenachádzajú parametre *id_del* ani *send* a zároveň je definovaný vyššie spomenutý parameter *courses_code*. Môže mať aj špeciálnu hodnotu *-1*, ktorá hovorí, že predmet ešte neexistuje a akcia bola zavolaná za účelom vytvorenia nového predmetu. Tá je následne spätne prevedená na prázdny reťazec, aby si systém nemyslel, že to bude kód nového predmetu. V prípade, že je kód predmetu platný, vytiahnu sa o predmete všetky údaje z tabuliek *courses*, *courses_garant*, *courses_program* a priradia sa do asociačného poľa. Najskôr sa vytvorí na stránke ešte jeden skrytý formulár, ktorého prvky obsahujú rovnaké identifikátory a hodnoty ako má vyhľadávacie formulár na úvodnej stránke katalógu predmetov v AIS. Jediný viditeľný prvok tohto inak skrytého formulára je tlačidlo na odoslanie. Tu som práve využil verejnú dostupnosť katalógu a spravil to tak, aby sa odoslali skryté údaje, pričom jeden z nich je kód upravovaného predmetu, metódou *POST* na stránku katalógu predmetov a výsledok dal zobrazíť na novej karte prehliadača. Tam by sa mala otvoriť stránka s údajmi predmetu. Pokiaľ by sa našlo viac predmetov s rovnakým kódom, AIS zobrazí ich zoznam. Keďže v tomto prípade ide o požiadavku odoslanú medzi dvoma doménami, z dôvodu ochrany užívateľov takéto požiadavky nevracajú žiadne dáta na stránku, odkiaľ boli odoslané. Preto musí užívateľ skopírovať URL adresu stránky predmetu v katalógu a vložiť ju do

textového poľa, ktoré pod odosielačím tlačidlom vytvorila JavaScript funkcia naviazaná na neho. Dôvod všetkého tohto je umožnenie užívateľovi získať informácie o upravovanom predmete aj z AIS. Okrem toho ešte funkcia vytvorí kostru editačného formulára, ktorá pozostáva z dvoch tlačidiel na odoslanie, pričom prvé slúži na uloženie zmien a druhé na opustenie formulára bez uloženia zmien. Obe posielajú parameter s názvom *send*. Ďalej ešte vytvorí tlačidlo na resetovanie formulára, na ktoré je naviazaná JavaScript funkcia, ktorá nastaví na pôvodnú hodnotu komplexnejšie prvky formulára. A ešte vytvorí dva skryté elementy, jeden má názov *edit_id*, kam sa nastaví kód upravovaného predmetu, a názov druhého je *save*, ktorý je automaticky nastavený na hodnotu 1. Po odoslaní formulára tlačidlom pre uloženie tak vie funkcia rozlíšiť, že má uložiť údaje o predmete a vie aj to, či sa jedná o nový alebo existujúci predmet. Vytvorenie jednotlivých viditeľných elementov editačného formulára s údajmi je už realizované ďalšou funkciou, ktorú opíšem v nasledujúcom odseku.

Funkcia má názov *create_course_form*, ako vstupný argument berie spomínané asociačné pole s údajmi o predmete a návratová hodnota je HTML kód tela editačného formulára. V pôvodnej testovacej verzii modulu bola táto funkcia pomerne komplikovaná, jej kostrou bol zložitý cyklus opakujúci sa pre každý prvok v poli s údajmi o predmetoch, avšak prijímala ako vstupné argumenty aj ďalšie pomocné polia obsahujúce rôzne dáta ako zobrazovaný text, numerické hodnoty, pomocou ktorých sa uprostred cyklu rozhodovalo, či sa daný element bude dať doplniť o údaje z AIS, jeho poradie vo formulári, typ elementu a iné. Z praktického hľadiska by sa táto časť kódu veľmi zle udržiavala, ale aspoň poskytovala bežným užívateľom pracovať s údajmi o predmetoch. Keďže som už predtým spomínal, že jedným z mojich cieľov bolo postupne spraviť kód generujúci formulár čo najľahšie modifikovateľný pre prípad, že by správca informačného systému potreboval nejaký element upraviť alebo inak zmeniť formulár. Preto som sa funkciu rozhodol prerobiť tak, že všetky podstatné nastavenia sú obsiahnuté v poli s názvom *\$settings*, ktoré je definované na začiatku funkcie. Prvky tohto poľa musia byť asociačné polia s určitou štruktúrou, ktoré reprezentujú jednotlivé elementy formulára. Keďže po novom je cyklus generujúci HTML kód formulára závislý od počtu prvkov poľa s nastaveniami, už viac nie je problém vynechať nejaký nepotrebný alebo nechcený údaj z databázy. Z tohto vyplýva aj skutočnosť, že pozícia elementu v poli nastavení odpovedá jeho pozícii vo formulári. Preto už nie je žiadny problém s presúvaním elementov. Ako som spomínal, vlastnosti samotného elementu musia byť definované v asociačnom poli, ktoré podporuje nasledujúce kľúče (vlastnosti):

- **title** - jeho hodnota definuje popis elementu, ktorý sa zobrazuje nad ním
- **data** - názov musí byť totožný s príslušným stĺpcom v databáze, slúži na zobrazenie pôvodnej hodnoty ako aj identifikátor pri ukladaní údajov

- **type** - určuje, aký druh elementu sa má vytvoriť. Podporované sú nasledovné hodnoty:

- **text** - obyčajné jednoriadkové textové pole
- **number** - to isté ako predchádzajúce, len hodnota je pred načítaním a pri ukladaní prevedená na číslo
- **radio** - skupina prepínačov, z ktorých je možné vybrať len jednu možnosť
- **select** - výberový zoznam s viacerými položkami, pričom vybrať sa dá len jedna
- **editor** - vytvorí viacriadkové pole na ktoré je následne naviazaný externý modul CKEditor, ktoré umožňuje pokročilé formátovanie dlhšieho textu. Pôvodná hodnota je v HTML kóde vpísaná aj priamo do elementu textového poľa a ostáva nezmenená až do uloženia, preto som ju využil na opakované načítanie pôvodnej hodnoty do editoru v prípade resetovania formulára, pretože táto štandardná HTML akcia na neho nemá efekt.
- **programs** - jedná sa o špeciálnu skupinu prvkov zoskupených v jednom elemente typu *div* a slúži na definovanie študijných programov a príslušných semestrov pre daný predmet. Skladá sa z jedného alebo viacerých riadkov obsahujúcich dve výberové zoznamy, jeden pre program a druhý pre semester, pričom dostupné položky sú závislé od aktuálne zvoleného stupňa štúdia. Ďalej ešte obsahuje tlačidlo pre odoberanie programov, ktoré sa zobrazí, keď má predmet aktuálne vybrané aspoň dve programy a tlačidlo na pridanie nového programu. Toto je zobrazené dovtedy, kým je možné pridať aspoň jeden ďalší program. Aj výberový zoznam na aktuálne poslednom riadku ponúka iba doposiaľ nevybrané programy. Z toho dôvodu sú aj výberové zoznamy na predchádzajúcich riadkoch deaktivované, aby sa jednotlivé zoznamy ich položiek pri prehadzovaní nestali neaktuálne. Pri prvotnom vytvorení formulára sa vytvorí aj skrytý element, ktorý obsahuje počiatočné dáta pre túto skupinu elementov vo formáte *JSON*. Tieto dáta sú hneď po načítaní stránky na strane klienta JavaScriptom transformované do objektu a tento element je odstránený z HTML kódu stránky. Vyriešil som týmto spôsobom z toho dôvodu, že počiatočné údaje sú nemenné a preto som nechcel pri resetovaní formuláru, ktoré prebieha za normálnych okolností len na strane klienta, zbytočne posielať požiadavku na server.

- **width** - nie je povinný, pokiaľ je definovaný, tak určuje šírku elementov typu *text* a *number*, na ostatné typy nemá žiadny efekt

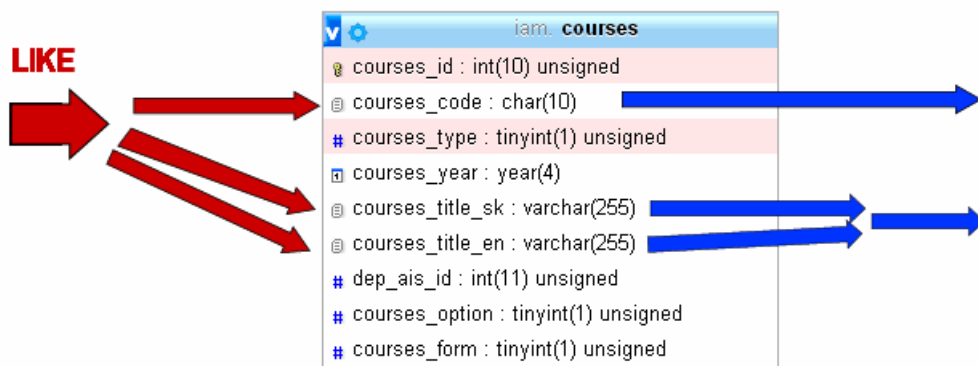
Okrem týchto vlastností existujú ešte ďalšie tri, ktoré definujú možnosti, resp. položky pre výberové elementy (typy *radio*, *select* a aj *programs*). Tieto vlastnosti sa navzájom vylučujú.

čujú a v prípade, že sú definované viaceré naraz, tak sa použije tá s najvyššou prioritou. Ich zoznam je usporiadaný od najnižšej po najvyššiu prioritu:

- **options** - musí obsahovať asociačné pole, pričom hodnota každého kľúča bude použitá ako názov položky výberového prvku a kľúč samotný ako jej hodnota
- **db** - pomocou jednoduchého SQL dotazu definovaného v tejto možnosti sa z databázy načíta zoznam dostupných možností. Dotaz musí vracať pre každý nájdený riadok v databáze dve hodnoty, prvá bude použitá ako hodnota a druhá ako názov položky výberového prvku. Vďaka tomuto prístupu sú dáta o predmetoch dostupné presne tam kde treba, miesto toho, aby sa musela tabuľka *courses* načítať pospájaná a obsahovať tak nadbytočné údaje.
- **func** - na načítanie dát sa použije PHP funkcia s názvom definovaným v tejto vlastnosti. Do funkcie bude odoslané pole s údajmi o predmete ako vstupný argument a ako návratovú hodnotu musí vracať asociačné pole s položkami výberového elementu. Túto možnosť som vytvoril pre prípad, že predchádzajúce dve sú nedostatočné a na vytvorenie položiek je potrebný zložitejší proces. Momentálne je to použité len pre element *courses_garant*, ktorý používa funkciu *get_garant_list*, ktorá vytiahne potrebné údaje z databázy a poskladá meno osoby z krstného mena, priezviska a titulov. Do budúcnosti môže mať však táto možnosť pestré uplatnenie, pretože je možné pomocou nej zavolať ľubovoľnú funkciu, ktorú niekto vytvorí.

V novej funkcii teda podstatne zjednodušený cyklus nájde v poli nastavení všetky definované elementy, zistí ich nastavenia a následne ich vytvorí. Na záver funkcia ešte vytvorí riadok s dátumom poslednej úpravy predmetu, ktorý je v tabuľke *courses* uložený v stĺpci *courses_date*.

Aplikácia využíva na strane servera ešte druhý skript s názvom *ajax_courses.php*, ktorý automaticky spúšťa rovnomennú funkciu (bez koncovky). Tento skript je volaný AJAX požiadavkami, ktoré prichádzajú zo strany klienta. Tieto požiadavky používajú metódu *POST*. Na základe prijatých parametrov potom funkcia rozhodne, ktorú z dvoch činností má vykonať. V prípade, že je definovaný parameter *course_str*, tak ide o už vyššie spomenuté vyhľadávanie predmetov na základnej stránke aplikácie. Parameter obsahuje vyhľadávaný reťazec, ktorý najskôr zbaví znakov s diakritikou a potom pošle SQL dotaz typu *SELECT*, v ktorom je použitý operátor *LIKE*, zabezpečujúci že hľadaný reťazec môže byť aj súčasťou iného reťazca. Vyhľadávanie prebieha v anglickom alebo slovenskom názve predmetu ako aj v jeho kóde. Grafické znázornenie vyhľadávacieho dotazu sa nachádza na obrázku 4. Z nájdených údajov potom funkcia vytvorí spomínaný zoznam predmetov a vráti jeho HTML kód.



Obr. 4: Grafické znázornenie dotazu hľadajúceho predmety

Druhý prípad nastane, pokiaľ je definovaný *POST* parameter s názvom *parse*. V tomto parametri by mala byť URL adresa predmetu v katalógu AIS. Keď z nej vytiahne potrebné údaje a podarí sa mu ju načítať, tak prebehne parsovanie údajov o predmete. Tomuto procesu sa budem venovať v osobitnej časti práce. Po ukončení parsovanie zo získaných údajov vytvorí asociačné pole a následne ho prekonvertuje *JSON* formátu a tento odošle späť na klientsku stranu aplikácie. Základná štruktúra *JSON* objektu po úspešnom parsovaní vyzerá nasledovne:

```
{
  ais_date: "15. 12. 2015",
  data: {Object},
  success: 1
}
```

Pokiaľ funkcia úspešne nedokončí všetky operácie, vrátený *JSON* objekt vyzerá takto:

```
{
  success: 0
}
```

Parameter *success* teda hovorí JavaScriptu o tom, či bolo parsovanie úspešné. Ak áno, v parametri *ais_date* nájdeme dátum poslednej úpravy údajov o predmete v AIS a parameter *data* je objekt, ktorý v sebe obsahuje údaje o predmete, pričom názvy kľúčov v tomto objekte odpovedajú názvom jednotlivých elementov formulára. Hodnoty týchto kľúčov sú ďalšie objekty, ktoré majú vždy kľúč s názvom *value*, ktorý obsahuje získanú hodnotu. Môže obsahovať aj kľúč *text*, ktorý bude vo formulári zobrazený namiesto hodnoty (napr. meno osoby a nie jej osobné číslo). Výnimku tvoria dáta pre zložený element obsahujúci programy a semestre. V tomto prípade sú použité polia objektov.

4.2.2 Funkcionalita na strane klienta

Okrem toho, že pre túto aplikáciu je pomocou JavaScriptu použité globálne nastavenie, aby sa formulár neodosiela po stlačení klávesy *Enter*, táto aplikácia používa viacero funkcií napísaných v JavaScripte, resp. v drivej väčšine prípadov v jQuery zápise. Nasledujúci zoznam stručne popisuje ich funkcionality:

- *SetInputListener* - k textovému poľu na vyhľadávanie predmetov pripojí zachytávanie *oninput* udalostí, resp. *onkeyup* v prípade starého prehliadača, pri zachytení udalosti zavolá funkciu *GetCourse*
- *SetParseInputListener* - k textovému poľu na URL adresu predmetu v katalógu AIS pripojí zachytávanie *oninput* udalostí, resp. *onchange* v prípade starého prehliadača, pri zachytení udalosti zavolá funkciu *StartParsing*
- *SetFormListeners* - nastaví vzájomnú závislosť medzi jednotlivými elementmi skupiny pre úpravu programov a semestrov predmetu ako aj stupňom štúdia, nastaví volanie resetovacích funkcií pre tieto elementy ako aj textové editory
- *GetCourse* - odošle AJAX požiadavku na skript *ajax_courses.php* s vyhľadávaným reťazcom, pokiaľ obsahuje aspoň 3 znaky a vrátený HTML kód zoznamu vloží na stránku
- *ShowAISAddressForm* - vytvorí textové pole na vkladanie URL adresy predmetu v AIS
- *StartParsing* - odošle AJAX požiadavku na skript *ajax_courses.php* s URL adresou predmetu v AIS a v prípade úspešného parsovania zobrazí hodnoty z AIS pod príslušnými elementami a vytvorí k nim tlačidlá na aktualizáciu elementu
- *CopyToForm* - na základe stlačeného tlačidla nastaví hodnotu príslušného elementu na hodnotu získanú z AIS
- *SetSemesters* - nastaví dostupné semestre na základe aktuálne vybraného stupňa štúdia
- *ResetEditors* - nastaví pôvodný text všetkých textových editorov na stránke
- *ResetPrograms* - nastaví pôvodný stav programov a príslušných semestrov
- *AjaxLoadingImage* - zobrazí alebo schová ikonu a text informujúcich o prebiehajúcej AJAX požiadavke

4.2.3 Parsovanie údajov z AIS

Keďže jednou z požiadaviek na modul bola možnosť mať počas úpravy predmetu k dispozícii dáta o predmete z AIS a v prípade potreby ich vložiť do formulára bez toho, aby si užívateľ všetko musel sám hľadať na stránkach AIS, avšak zároveň žiadne kopírovanie databázy nie je možné, nakoľko prístup k databáze AIS STU jednoducho nemáme k dispozícii, bolo potrebné získať tieto údaje iným spôsobom. Pravdepodobne najschodnejšia cesta, ako to spraviť, je parsovanie. Ide o proces, pri ktorom zo vstupu, ktorý najčastejšie býva nejaký textový reťazec alebo aj iný objekt a zvyčajne obsahuje veľké množstvo údajov, získavame na základe určitých pravidiel len želané časti vstupu vo forme viacerých reťazcov, objektov alebo iných dátových typov. Na tento účel je v PHP možné použiť viacero riešení. Ja som sa rozhodol využívať DOM modul priamo od PHP, ale použiť sa dajú aj rozšírenia vytvorené inými osobami, napr. voľne dostupný PHP Simple HTML DOM Parser, ktorý funguje na veľmi podobnom princípe ako štandardný modul od PHP a tým je vytvorenie DOM objektu načítaného dokumentu. Softvér je voľne dostupný na <http://simplehtmldom.sourceforge.net>, kde je dostupná aj dokumentácia. Hlavné príčiny, prečo som sa rozhodol pre štandardný modul sú dve, jedna spočíva v tom, že pri štandardne dodávaných moduloch je väčšia istota, že bude udržiavané aj v budúcnosti, druhú výhodu vidím v tom, že nie potrebné na server pridávať ďalší softvér, keď sa tam porovnateľný už nachádza.

Aby som mohol zadať podmienky vyhľadávania v DOM objekte stránky predmetu v katalógu AIS, bolo potrebné najprv získať informácie o štruktúre stránky predmetu v AIS (zobraziť zdrojový kód cez prehliadač). Najskôr som si vyhlíadal na bežnej stránke potrebné údaje, čo znázorňuje obrázok 5. Po analýze zdrojového kódu stránky som zistil, že všetky potrebné údaje sú zobrazené v jednej veľkej tabuľke. Preto som najskôr v DOM objekte našiel všetky uzly *table*), následne vybral prvý (hlavná tabuľka obsahujúca všetky potrebné elementy a údaje stránky). Následne som cyklom prehládaval všetky jej riadky a porovnával, či v prvom stĺpci je reťazec, ktorý potrebujem (napr. Názov predmetu). Ak áno, tak som z druhého stĺpca vybral jeho textový obsah a uložil ho do premennej. Týmto jednoduchým spôsobom som bol schopný získať väčšinu potrebných údajov, v niektorých prípadoch bolo potrebné v prípade nálezu potrebného reťazca skočiť na ďalší riadok tabuľky pomocou metódy *NextSibling* alebo v prípade garanta, kde bolo potrebné získať jeho identifikačné číslo, nájsť v danom riadku tabuľky element *a*, následne jeho atribút *href*, pretože *id* osoby sa nachádza v URL adrese tohto odkazu. Z tejto adresy som pomocou rozbíjania reťazca získal potrebné *id*. Niektoré prípady potrebovali ešte zložitejšie operácie ako napr. získavanie všetkých dcérskych elementov, odstraňovanie atribútov alebo dodatočné dopĺňanie údajov z databázy informačného systému ÚIAM. Môžem však konštatovať, že definícia každej podmienky parsovania priamo vyplýva zo štruktúry aktuálnej HTML stránky a pokiaľ by sa v budúcnosti zmenila, bolo by potrebné prerobiť tieto podmienky, aby parsovanie fungovalo správne aj naďalej.

Vysoká škola:	Slovenská technická univerzita												
Fakulta:	Fakulta chemickej a potravinárskej												
Kód predmetu:	422T1_4I												
Názov predmetu:	Teória automatického riadenia I												
Druh, rozsah a metóda vzdelávacích činností:													
prednáška	2 hod. týždenne (prezenčná metóda)												
laboratórne/konštrukčné cvičenie	3 hod. týždenne (prezenčná metóda)												
Počet kreditov:	6												
Odporúčaný semester/trimester:	automatizácia a informatizácia												
Stupeň štúdia:	2.												
Podmieňujúce predmety:	žiadne												
Podmienky na absolvovanie predmetu:													
Hodnotenie predmetu je založené 30% na práci počas semestra, 30% z písomnej príkladovej skúšky a													
Výsledky vzdelávania:													
Študent získa vedomosti zo základov teórie automatického riadenia, vie navrhnúť späťoväzbové riadi													
Stručná osnova predmetu:													
Lineárne dynamické systémy Stavové modely procesov Ljapunovova stabilita Prenosové modely procesov Časové odozvy Frekvenčná analýza Stavový regulátor s pozorovačom Štruktúra spätnej väzby													
Odporúčaná literatúra:													
Základné: MIKLEŠ, J. -- FIKAR, M. <i>Process Modelling, Identification, and Control</i> . Berlin Heidelberg: Springer B													
Jazyk, ktorého znalosť je potrebná na absolvovanie predmetu:	slovenský jazyk alebo anglický												
Poznámky:													
Hodnotenie predmetov:													
Celkový počet hodnotených študentov: 107													
<table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>FX</td></tr><tr><td>20,6 %</td><td>23,4 %</td><td>29,9 %</td><td>15,9 %</td><td>8,4 %</td><td>1,8 %</td></tr></table>		A	B	C	D	E	FX	20,6 %	23,4 %	29,9 %	15,9 %	8,4 %	1,8 %
A	B	C	D	E	FX								
20,6 %	23,4 %	29,9 %	15,9 %	8,4 %	1,8 %								
Vyučujúci:	prof. Ing. Miroslav Fikar, DrSc. Ing. Ayush Sharma (cvičiaci) Ing. Anna Vasičkaninová, PhD.												
Dátum poslednej zmeny:	14. 12. 2015												
Schválil:	prof. Ing. Miroslav Fikar, DrSc.												

Obr. 5: Stránka predmetu v katalógu AIS s vyznačenými údajmi na parsovanie

Väčšinu údajov som parsoval z anglickej verzie stránky, pre položky, kde bolo treba aj slovenský preklad som nanovo vytvoril DOM objekt pre stránku s upravenou URL adresou pre slovenskú verziu a rovnakým spôsobom, len znovu s mierne odlišnou URL, získal aj údaje zo záložky pre syllabus predmetu, odkiaľ som získal jeho obsah a typ ukončenia.

4.2.4 Integrácia do informačného systému a jazykové verzie

Pod integráciou sa rozumie sprístupnenie modulu cez webové stránky ako aj využívanie aktuálnych možností informačného systému. Modul bol zaradený do editačnej (chránenej) časti systému, ktorá je dostupná len pre prihlásených užívateľov stránky, čiže zamestnancov ústavu. Aplikáciu nie je teda možné spustiť bez prihlásenia ani keby niekto priamo zadal jej presnú URL adresu v systéme. Čo sa týka zdieľaných funkcií, tak aplikácia využíva hlavne funkcie na komunikáciu s databázou alebo úpravu dát poslanými metódou *POST*.

Informačný systém ponúka možnosť výberu medzi dvoma jazykovými verziami, slovenskou a anglickou. Ako som však spomínal, dvojязыčné verzie stránok sú len vo verejne dostupnej časti systému, kde sú dostupné informácie pre všetkých návštevníkov stránky. Nakoľko som ale tento modul vytváral od úplného začiatku, rozhodol som sa pridať aj anglickú verziu stránky, keďže aj údaje o predmetoch sú dostupné v oboch jazykoch. Dvojязыčné verzie údajov ako napr. osnova predmetu som síce v editačnom formulári zobrazil ako dva elementy a to z praktických dôvodov, aby užívateľ videl všetky dostupné údaje a nemusel prepínať kvôli tomu jazyk stránky, ale všetko ostatné je dostupné v oboch jazykoch. Toto okrem dvoch verzií zobrazovaného textu znamená aj to, že aj výberové prvky vo formulári načítavajú svoje možnosti v príslušnom jazyku a aj vyhľadávanie predmetov pracuje v aktuálnej jazykovej verzii, čo znázorňujú obrázky 6 a 7.





















Obr. 6: Ukážka slovenskej verzie aplikácie

Course editing

Application allows you to add a new course or edit an existing one.

 [New course](#)

After entering at least three characters, automatic search will find all matching courses. Search pattern can be any part of course title or code.

	422T1_4I	Automatic Control Theory I	
	422T3_4I	Automatic Control Theory II	
	N422T0_4I	Automatic Control Theory III	
	N424V1_4B	General Theory of Economy	
	N424M3_4B	Mathematics I	
	N424M2_4B	Mathematics II	
	N427N0_4B	Numerical Mathematics	
	N427V0_4I	Selected Topics in Mathematics	
	N427S0_4B	Seminar of Mathematics	

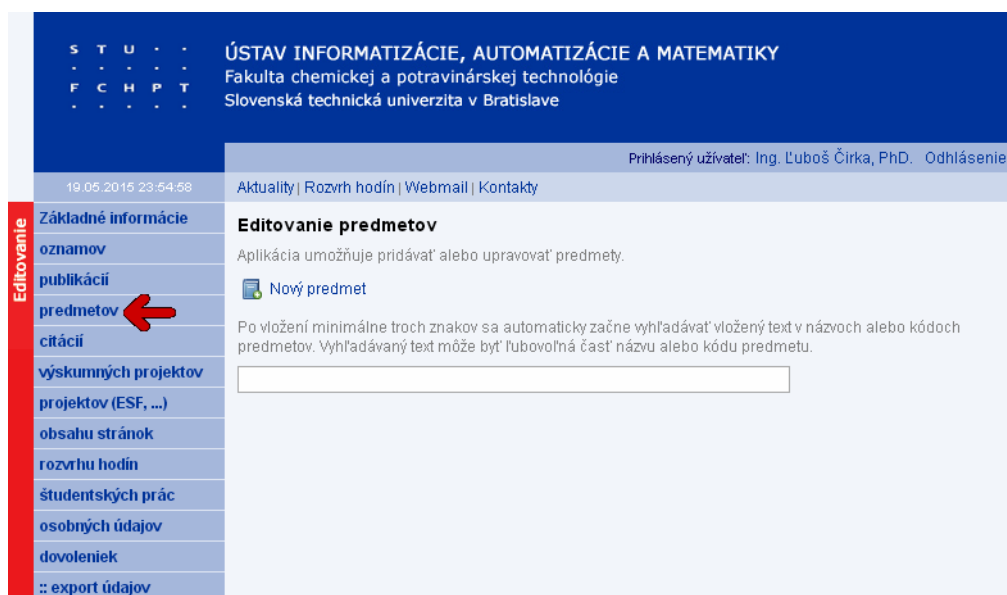
Obr. 7: Ukážka anglickej verzie aplikácie

4.3 Uživatelská dokumentácia

Slúži na predstavenie modulu budúcemu používateľovi, opisuje a znázorňuje prácu s ním, zahŕňa dostupné akcie a funkcionality z pohľadu užívateľa.

4.3.1 Spustenie

Modul na úpravu predmetov sa spúšťa priamo z bočného menu webovej stránky predmetu a to v sekcii *Editovanie* pomocou položky *predmetov* (obrázok 8). Používateľ musí byť prihlásený na to, aby sa mu zobrazila editačná časť menu.



Obr. 8: Spustenie modulu predmety

4.3.2 Vyhľadávanie predmetov

Na úpravu predmetu je najskôr potrebné zvoliť, ktorý predmet to bude. Preto aplikácia má v sebe zabudované vyhľadávanie predmetov. Keď používateľ vloží do textovej poľa (obrázok 9) aspoň 3 platné znaky, spustí sa vyhľadávanie. Za platné znaky sa považujú malé a veľké písmená (vrátane diakritiky), číslce a podtržník. Všetky ostatné sú odfiltrované, preto aplikácia nezobrazí žiadne výsledky ani keď má používateľ zadané tri znaky. Vyhľadávaný text môže byť časť alebo celý kód predmetu alebo jeho názov, pričom pri názve záleží od aktuálnej jazykovej verzie aplikácie. Nájdene predmety sú zobrazené pod textovým poľom.





Editovanie predmetov

Aplikácia umožňuje pridávať alebo upravovať predmety.

 [Nový predmet](#)

Po vložení minimálne troch znakov sa automaticky začne vyhľadávať vložený text v názvoch alebo kódoch predmetov. Vyhľadávaný text môže byť ľubovoľná časť názvu alebo kódu predmetu.



	422T1_4I	Teória automatického riadenia I	
	422T3_4I	Teória automatického riadenia II	
	N422T0_4I	Teória automatického riadenia III	
	N424V1_4B	Všeobecná ekonomická teória	

Obr. 9: Vyhľadávanie predmetov

4.3.3 Editovanie predmetu

Pre editovanie predmetu je potrebné kliknúť na editačnú ikonku prislúchajúcu pre daný predmet v zozname. Ikonka sa nachádza vľavo od kódu požadovaného predmetu (viď obrázok 9). Následne sa otvorí editačný formulár a načítajú sa do neho údaje pre vybraný predmet. Nakoľko týchto údajov je dosť veľa, aj formulár obsahuje väčší počet prvkov. Podľa typu údajov by ho bolo možné naoko rozdeliť na dve časti. Prvá časť obsahuje jednoriadkové textové polia a výberové prvky. V tejto časti sa nachádzajú najmä základné informácie o predmete, akými sú názov predmetu a jeho kód, forma výučby, týždenná výmera, počet kreditov za predmet, stupeň na ktorom sa vyučuje a iné. Za zmienku stojí element na výber odporúčaného semestra. Predmet sa totiž môže vyučovať aj na viacerých študijných programoch a to aj v rôznych semestroch. Preto tento element dovoľuje používateľovi vybrať viac programov a pre každý aj semester. Pokiaľ chce užívateľ pridať ďalší program, môže tak urobiť kliknutím na tlačidlo *Pridať*, ktoré dostupné dovtedy, kým nie sú vybraté všetky programy na aktuálnom stupni štúdia. Pokiaľ chce naopak užívateľ program odobrať, dosiahne to kliknutím na tlačidlo so symbolom krížika. Odstrániť je vždy možné len aktuálne posledný program a prvý nie je možné odstrániť nikdy, lebo musí byť definovaný aspoň jeden program pre predmet. Spomínanú prvú časť formulára je vidieť na obrázku 13 na strane 43, pričom spomínané tlačidlá na pridávanie a odoberanie predmetov sú označené v červených rámečkoch. Druhá časť formulára pozostáva hlavne z textových editorov, pretože obsahuje obsiahnejšie a formátovateľné údaje ako osnova, syllabus, cieľ alebo podmienky na absolvovanie predmetu. Všetky tieto údaje sa vyskytujú aj v slovenskej aj v anglickej verzii, čo počet elementov ešte zdvojnásobí. Okrem toho ešte na úplnom konci formuláru môže mať predmet definované čísla kurzov v Moodle, ak nejaké má, a potom je tu ešte vypísaný dátum poslednej úpravy predmetu a 3 tlačidlá, z ktorých prvé zľava uloží zmeny vykonané vo formulári a vráti užívateľa na stránku s vyhľadávaním, stredné nastaví hodnoty všetkých elementov vo formulári na pôvodnú

hodnotu a posledné vráti užívateľa na vyhľadávanie bez uloženia zmien. Na obrázku 14 na strane 44 je zobrazená najspodnejšia časť formulára, zobrazujúca textové editory pre úpravu slovenskej a anglickej verzie sylabu predmetu, polia na definovanie čísel kurzov v Moodle a spomínané tlačidlá, ktoré sú označené červeným rámečkom.

4.3.4 Získanie údajov z AIS

Pokiaľ chce používateľ porovnať aktuálne údaje o predmete s údajmi v AIS, na začiatku formulára sa nachádza tlačidlo (na obrázku 10 v červenom rámečku), ktoré po stlačení otvorí na novej karte prehliadača stránku predmetu v katalógu AIS. Pokiaľ by existovalo viac predmetov s rovnakým kódom, zobrazí sa stránka so zoznamom týchto predmetov a užívateľ si v nich musí sám nájsť ten správny. Potom stačí skopírovať URL adresu tejto stránky v katalógu a vložiť ju do textového poľa, ktoré sa medzičasom vytvorilo v editačnom formulári (tiež znázornené na obrázku 10).

The screenshot shows a web form titled "Editovanie predmetov". Below the title is a descriptive text: "Aplikácia umožňuje upravovať predmety. Ďalej umožňuje kopírovať údaje o predmete uložené v AIS do IS ÚIAM." Below this text is a button labeled "Nájsť v AIS" which is highlighted with a red rectangular border. Underneath the button is a text instruction: "Do textového poľa vložte URL adresu predmetu v katalógu AIS". Below this instruction is a large, empty text input field, also outlined with a red border. To the right of this input field is a red arrow pointing left towards the field, with the text "URL" next to it. Below the input field is a label "Kód predmetu" followed by a text input field containing the value "N422Z2_4B".

Obr. 10: Získanie údajov predmete z AIS

Následne sa do formuláru doplnia údaje získané z AIS a zobrazia sa pod príslušnými elementmi. Naľavo od údajov sa vytvorí tlačidlo so symbolom šípky, ktoré po stlačení nastaví hodnotu alebo pozíciu elementu tak, aby odpovedala údaju získanému z AIS. Na obrázku 11 je znázornený tento prípad pre element slúžiaci na úpravu študijných programov a príslušných semestrov. Obrázok znázorňuje stav po získaní údajov z AIS (časť nad čiernou čiarou) a stav po stlačení tlačidla na aktualizáciu elementu.

Odporúčaný semester

Študijný program: Potraviny, výživa, kozmetika

Semester: 1.

Študijný program: Chemické inžinierstvo

Semester: 5.

Študijný program: Chémia, medicínska chémia a chemické materiály

Semester: 2.

✕

Pridať

↑ AIS:

Automatizácia, informatizácia a manažment v chémii a potravinárstve (1)

Chémia, medicínska chémia a chemické materiály (3)

Odporúčaný semester

Študijný program: Automatizácia, informatizácia a manažment v chémii

Semester: 1.

Študijný program: Chémia, medicínska chémia a chemické materiály

Semester: 3.

✕

Pridať

↑ AIS:

Automatizácia, informatizácia a manažment v chémii a potravinárstve (1)

Chémia, medicínska chémia a chemické materiály (3)

Obr. 11: Aktualizovanie elementu údajmi z AIS

4.3.5 Vytvorenie nového predmetu

Pokiaľ by chcel používateľ vytvoriť nový predmet, ktorý sa ešte nenachádza v databáze ÚIAM, môže otvoriť prázdny editačný formulár kliknutím na ikonu alebo odkaz napravo od nej v hornej časti stránky na vyhľadávanie predmetov (viď obrázok 12). V tomto formulári môže potom všetky údaje vyplniť ručne, alebo keď vie o tom, že predmet sa nachádza v katalógu AIS, môže použiť postup popísaný v sekcii 4.3.4 s tým rozdielom, že tlačidlo zobrazené na obrázku 10 otvorí v novej karte prehliadača úvodnú stránku katalógu predmetov s vyhľadávaním a užívateľ si predmet musí nájsť sám.

4.3.6 Vymazanie predmetu

V prípade, že by používateľ potreboval vymazať niektorý predmet, tak po jeho vyhľadani treba kliknúť na ikonu nachádzajúcu sa napravo od názvu daného predmetu (obrázok 12) v zozname nájdených predmetov. Po kliknutí na ňu sa zobrazí modálne okno, ktorým je možné túto akciu buď potvrdiť alebo zrušiť. Po jej potvrdení je predmet nenávratne vymazaný z databázy a zmizne aj zo zobrazeného zoznamu nájdených predmetov.





41

Editovanie predmetov

Aplikácia umožňuje pridávať alebo upravovať predmety.

 [Nový predmet](#)

Po vložení minimálne troch znakov sa automaticky začne vyhľadávať vložený text v názvoch alebo kódoch predmetov. Vyhľadávaný text môže byť ľubovoľná časť názvu alebo kódu predmetu.

-  422T1_4I Teória automatického riadenia I 
-  422T3_4I Teória automatického riadenia II 
-  N422T0_4I Teória automatického riadenia III 
-  N424V1_4B Všeobecná ekonomická teória 

Obr. 12: Ostatné akcie modulu na úpravu predmetov

Editovanie predmetov

Aplikácia umožňuje upravovať predmety. Ďalej umožňuje kopírovať údaje o predmete uložené v AIS do IS ÚIAM.

Nájsť v AIS

Kód predmetu

N422Z2_4B

Rok zavedenia

2015

Názov predmetu

Základy Matlabu

Anglický názov predmetu

Fundamentals of Matlab

Stupeň štúdia

☒ bakalársky ☐ inžiniersky

Zabezpečuje

Oddelenie informatizácie a riadenia procesov ▼

Garant

Ing. Ľuboš Čírka, PhD. ▼

Voliteľnosť predmetu

Povinné predmety ▼

Druh vzdelávacích činností

laboratórne cvičenie ▼

Týždenný rozsah prednášok

0

Týždenný rozsah cvičení

0

Týždenný rozsah laboratórnych/konštrukčných cvičení

2

Počet kreditov

2

Odporúčaný semester

Študijný program: Automatizácia, informatizácia a manažment v chémii ▼ Semester: 1. ▼

Študijný program: Chémia, medicínska chémia a chemické materiály ▼ Semester: 3. ▼

Pridať

Spôsob ukončenia

Z ▼

Jazyk, v ktorom sa predmet vyučuje

slovenský, anglický ▼

Obr. 13: Úvodná časť formulára na editáciu predmetu

5 Modul citácie

Tento už existujúci modul systému sa používa na pridávanie nových alebo editovanie existujúcich citácií k publikáciám, ktoré sa nachádzajú v databáze informačného systému. Aplikácia používa BibTeX formát a dokáže aj spracovávať textové údaje obsahujúce citácie v tomto formáte.

5.1 Analýza stavu a špecifikácia požiadaviek na vylepšenie

Modul sa spúšťa priamo z menu editačnej časti systému. Úvodná stránka obsahuje textové pole na vyhľadávanie dostupných publikácií. Po zadaní aspoň troch znakov do tohto poľa sa spustí vyhľadávanie, ktoré hľadá názvy publikácií v slovenskom aj anglickom jazyku. Používateľ má prístup len k svojim vlastným publikáciám. Nájdene publikácie sa zobrazia pod textovým poľom. V tomto kroku si užívateľ volí citovanú publikáciu a keď si niektorú vyberie, má možnosť pridať k nej novú citáciu alebo potom upraviť či zmazať existujúce, ktoré sú zobrazené v spodnej časti stránky v zozname usporiadanom podľa roku publikovania, pokiaľ nejaké existujú. Pri úprave sa otvorí formulár s jednotlivými položkami citácie, ktoré je možné upravovať. Tieto položky odpovedajú použitým typom bibliografických údajov definovaným formátom BibTeX. V prípade, že sa užívateľ rozhodne pridať novú citujúcu publikáciu, najskôr si musí vybrať z ponuky jeden zo spôsobov zadávania citujúcej publikácie. Zvoliť sa dá manuálny spôsob, kedy všetky údaje musí užívateľ vyplniť sám, ďalej je možné vytvoriť citáciu priamo výberom citujúcej publikácie z informačného systému ÚIAM, pričom publikácia sa môže nachádzať buď v zozname publikácií alebo citácií. Posledný a zrejme aj najčastejšie používaný spôsob je import citácií z niektorej z celosvetovo najobľúbenejších elektronických knižníc, ku ktorým má škola prístup. Tieto knižnice sa zaoberajú okrem iného aj zberom citácií v oblasti vedy, techniky a výskumu v celosvetovom merítku. Po zaplatení poplatku ponúkajú potom prístup k databáze a služby ako vyhľadávanie a export bibliografických údajov do viacerých formátov. Modul ponúka sprievodcov na pridanie citácií z vyexportovaných dát vo formáte BibTeX. Tieto údaje môžu byť vyexportované z jednej z troch nasledujúcich zdrojových databáz:

- *Web of Knowledge*
- *Scopus*
- *IEEE Xplore*

Následne sa musia vyexportované údaje ručne skopírovať a vložiť do textového okna a po spracovaní sa pod ním zobrazia jednotlivé položky všetkých citácií, ktoré je v prípade potreby ešte možné upraviť pred uložením do databázy. Údaje sú parsované pomocou funkcií, ktoré boli vytvorené zvlášť pre každý zdroj, pretože ani BibTeX formát nie je striktné štandardizovaný.

Z hľadiska funkčnosti bol tento modul navrhnutý dobre, postup vkladania nových citujúcich publikácií je zrozumiteľný a modul obsahuje aj pomocné texty. Preto sa v tomto prípade vo vylepšovaní zameriam na import citácií zo spomínaných externých zdrojov, pretože na dnešnú dobu už pôsobí pomerne ťažkopádne. Taktiež som sa chcel pokúsiť rozšíriť podporované formáty bibliografických údajov, pretože z praktických skúseností sa zistilo, že dostať citácie vo formáte BibTeX nemusí byť vždy jednoduché a očividné. Keďže tabuľka v databáze je navrhnutá práve pre formát BibTeX a ukázalo sa, že ostatné formáty nie sú konzistentné, hlavne na Web of Science, kde z iných formátov nebolo možné určiť typ citujúcej publikácie ako aj priradiť názov do správneho poľa. Väčšinou aj tak išlo o rôzne neštandardizované textové formáty. Preto som sa rozhodol pridať aspoň formát RIS, ktorý je narozdiel od ostatných štandardizovaný a spolu s BibTeX-om aj najčastejšie využívaný. Export v tomto formáte podporuje *Scopus* aj *IEEE Xplore*. Ani v tomto prípade nemusí byť konverzia na BibTeX stopercentne presná, ale vo väčšine prípadov bola úspešná a prípadné chyby je možné upraviť ručne. Stále je to časovo efektívnejšie ako používať externý softvér na konverziu.

5.2 Realizácia vylepšení

Počas práce na tomto module som spravil drobné úpravy, akými boli napríklad úprava pomocných textov alebo pridanie *label* elementov pre popisky prepínačov a zaškrŕavacích políčk, vďaka ktorým je možné potom označiť prepínací element aj po stlačení na text vedľa neho. Toto malé vylepšenie môže byť praktické najmä v častiach aplikácie, kde sa môže vyberať z dlhých zoznamov publikácií, pričom text zaberá ďaleko väčšiu plochu ako samotný prepínač. Okrem toho som ešte opravil pár nedostatkov v parsovacích funkciách pre formát BibTeX. Prvý sa týkal parsovania atribútu *number* (číslo vydania, používané hlavne pri článkoch), ktorý vyparsovalo aj v prípade, že slovo *number* nebol celý názov atribútu, čo v niektorých prípadoch viedlo k získaniu hodnoty úplne iného atribútu. Druhá úprava bola rozlišovanie pôvodu dokumentu z toho dôvodu, aby sa funkcia pre iný zdroj nesnažila parsovať údaje, pretože niekedy to viedlo k neúplným alebo prázdnyim hodnotám atribútov.

5.2.1 Sprievodca vložením citácie z externého zdroja

V tomto prípade som sa zameriaval hlavne na klientsku časť aplikácie za účelom zjednodušiť prácu so sprievodcom importu externých údajov a spraviť ho viac interaktívneho. Hovorím najmä o možnosti vložiť obsah súboru do textového okna priamo pomocou presunutia ikony tohto súboru kurzorom do okna (*drag and drop*). Export údajov môže totiž prebehnúť aj formou uloženia textového súboru v príslušnom formáte, je menej praktické, keď potom užívateľ musí otvárať súbor, označovať v ňom obsah, ten následne skopírovať a vložiť do textového okna. Ja som navyše použil prístup, ktorý prináša ešte ďalšie výhody.

Využil som *File API*, ktoré prinieslo HTML5, ktoré umožňuje načítavať lokálne súbory na strane klienta. Prvá výhoda je v rýchlosti načítavania obsahu, keďže súbor sa načítava iba do pamäti lokálneho zariadenia používateľa, ktorú používa webový prehliadač. To umožňuje pracovať so súborom vo webovej aplikácii pomocou JavaScriptu bez potreby čokoľvek odolať na server. Taktiež tento prístup nevyžaduje žiadne miesto navyše z úložnej kapacity servera. V tomto prípade to použijem tak, že na server budú odoslané údaje v úplne rovnakej forme ako keby boli ručne skopírované do textového okna. Užívateľ sa nemusí starať ani o kódovanie textu v súbore, lebo aj to bude po vložení do okna prispôbené kódovaniu webovej stránky. Najskôr som v JavaScripte vytvoril globálne nastavenie, ktoré zabráni, aby prehliadač otváral vo svojej karte obsah súboru, ktorý je do neho vložený pomocou udalosti *drag and drop*. Tým dosiahnem to, že ak by užívateľ prenášaný súbor uvoľnil mimo textového okna, stránka na to nebude reagovať. Následne pri načítaní stránky so sprievodcom na import citácii spustím funkciu *initialize_filer*. Tá najskôr overí, či prehliadač podporuje takéto lokálne načítavanie súborov. Pokiaľ nie, oznámi to užívateľovi prostredníctvom modálneho okna (štandardná funkcia *alert*). V opačnom prípade na textové okno naviaže tri funkcie pre čiastkové udalosti, ktoré patria do skupiny udalostí pri vkladaní súborov prostredníctvom kurzora (spomínaný *drag and drop*):

- *fnDragOver* - je volaná vždy, keď sa užívateľ dostane so súborom nad textové okno. Táto funkcia zmení farbu okna a tvar kurzora, aby užívateľ vedel, že sem môže vložiť súbor.
- *fnDragLeave* - vykoná sa vtedy, ak užívateľ opustí s kurzorom plochu textového okna bez toho, aby uvoľnil tlačidlo a tým súbor vložil do okna, táto funkcia vráti farbu textového okna na pôvodnú.
- *fnSelectedFile* - zavolá sa, ak užívateľ vloží súbor do okna, načíta jeho obsah a pokiaľ zistí, že súbor nie je prázdny, vloží jeho obsah do textového okna. Popri tom aj zmení farbu okna na pôvodnú.

Okrem toho som vytvoril ešte ďalšie JavaScript funkcie, ktorých účel popíšem v nasledujúcom zozname:

- *FilerMsg* - vypíše správu pod textovým oknom, text správy je prvým argumentom funkcie a typ zobrazenia druhým. Správa sa môže zobraziť ako varovná, chybová alebo s ikonkou, ktorá znázorňuje priebeh načítavania dát. Ako upozornenie sa používa v prípade, že sa nepodarí rozpoznať požadovaný formát dát, ako chyba v prípade, že načítavanie obsahu zlyhá alebo užívateľ vloží prázdny súbor do okna.
- *HideFilerMsg* - odstráni správu z obrazovky, táto funkcia je vždy volaná aj na začiatku funkcie *FilerMsg*, pretože niekedy môže nastať prípad, že sa vypíše nová správa bez toho, aby predchádzajúcu nejaká udalosť predtým odstránil

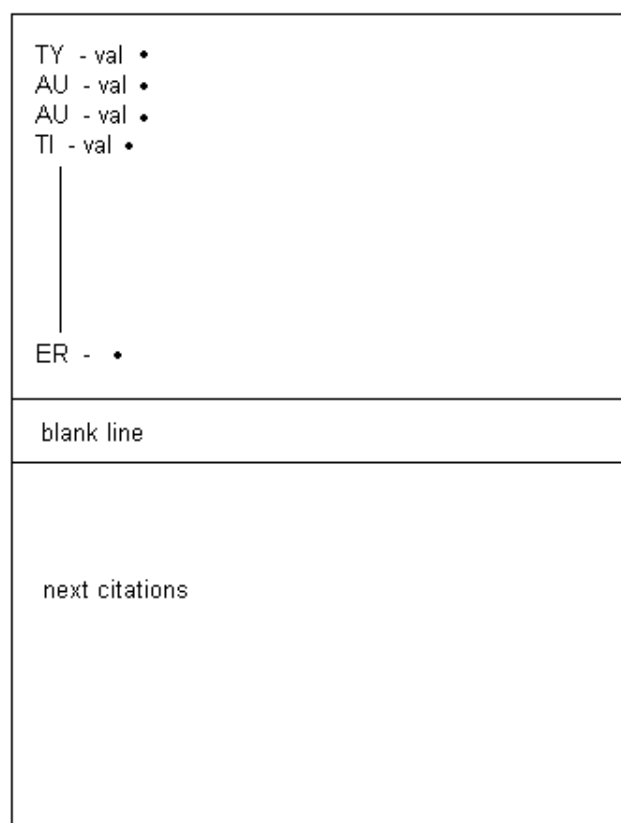
- *CheckSourceData* - na základe typu zdroja citujúcej publikácie volá funkcie na overovanie formátu dát v okne. Pre *Scopus* a *IEEE Xplore* sa kontrolujú formáty BibTeX a RIS, v prípade *Web of Knowledge* len *BibTeX*. Pokiaľ sa povolený formát nepodarí identifikovať, zobrazí sa varovná správa. V opačnom prípade nastaví, ktorú parsovaciu funkciu má server použiť.
- *isBIB_WoS* - funkcia, ktorá zistí, či sú vložené dáta vo formáte BibTeX a či pochádzajú z *Web of Knowledge*
- *isBIB_Sco* - obdobná funkcia, len zisťuje, či zdroj údajov je *Scopus*
- *isBIB_IEEE* - tiež obdobná funkcia, zdroj však musí byť *IEEE Xplore*
- *isRIS* - funkcia, ktorá overí, či sú údaje v okne vo formáte RIS. Pokiaľ je druhý argument nastavený na pravdivú hodnotu, sleduje, či zdroj údajov je *Scopus*.

Ako bolo spomenuté v popisoch funkcií, v prípade, že nebol rozoznaný podporovaný formát, systém na to užívateľa upozorní varovnou správou. Užívateľ môže odoslať aj napriek tomu údaje na spracovanie, pokiaľ si myslí, že rozpoznávanie bolo nejakým spôsobom oklamané. Ešte spomeniem, že som pridal aj tlačidlo na vymazanie obsahu textového okna, čo môže byť praktické najmä pri väčšom množstve údajov.

5.2.2 Parsovanie formátu RIS

Ako som už spomínal, jedná sa o štandardizovaný formát vyvinutý za účelom vymieňania si údajov medzi programami pracujúcimi s citáciami. Každý riadok s údajom začína značkou atribútu, potom nasledujú dve medzery, pomlčka, ďalšia medzera a hodnota atribútu. V prípade, že publikácia má viacerých autorov, nachádza sa tam riadok so značkou pre autora (*AU*) viackrát. Povinné atribúty sú len *TY*, ktorým vždy začína záznam a hovorí o jej type a atribút *ER*, ktorý je vždy posledný, neobsahuje žiadnu hodnotu a označuje koniec záznamu. Obrázok 15 na strane 49 obsahuje jednoduchý náčrt štruktúry tohto formátu.

Parsovacia funkcia najskôr cez atribút *ER* oddelí jednotlivé citácie, potom odstráni prípadné prázdne riadky a nakoniec pomocou deliaceho vzoru (spomínaná pomlčka a medzery) oddelí do poľa značku atribútu a jeho hodnotu. Ešte sa pomocou vyhľadávania všetkých riadkov so značkou pre autora získa kompletný zoznam autorov publikácie. Následne sa už vyhľadávajú len potrebné atribúty, funkcia sa potom prevedie typ publikácie na hodnotu kompatibilnú s formátom BibTeX, v prípade potreby overí, či je zdrojová databáza *Scopus* a ich hodnoty sa uložia do poľa, ktoré funkcia vráti na ďalšie spracovanie.



Obr. 15: Náčrt štruktúry formátu RIS

5.3 Uživateľský opis novej funkcionality

Modul umožňuje načítavať obsah súborov s bibliografickými dátami priamo vloženíím ikony súboru uloženého na užívateľovom zariadení. Najskôr je potrebné zvoliť si citovanú publikáciu a potom vybrať sprievodcu na pridanie citujúcej publikácie z nejakého externého zdroja (napr. *Web of Knowledge*). Sprievodca po spustení vyzerá ako na obrázku 16.

Citácia/citácie z Web of Knowledge
Skopírujte údaje z Web of Knowledge do textového poľa. Údaje exportujte vo formáte BibTeX.
☒ Postup pri vytváraní BibTeX výstupu z Web of Knowledge
1. Adresa stránky: <http://apps.webofknowledge.com/>;
2. Po vyhľadaní (napr. podľa mena autora) sa zobrazí tabuľka s publikáciami, kde v riadku začínajúcom textom **Times Cited:** je uvedený počet citácií;
3. Kliknutím na uvedené číslo sa zobrazí nová tabuľka, ktorá obsahuje iba citujúce publikácie;
4. Kliknutím na výberový prvok (Checkbox – na začiatku každého riadku) označte publikácie, ktoré chcete exportovať. V prípade, že chcete všetky zobrazené publikácie, zvolte **All records on page** v prvom kroku exportu;
5. V dolnej časti stránky sa nachádza export citácií (**Output Records**);
6. V prvom kroku zvolte, ktoré citácie sa majú exportovať;
7. V druhom kroku zvolte **Authors, Title, Source** bez Abstraktu;
8. V treťom kroku zvolte **Save to BibTeX** a kliknite na tlačidlo **Save**;
9. Výsledok vložte do okna (**CTRL+V** alebo potiahnutím lokálneho súboru myšou).

Spracovať informácie

Vymazať údaje

Dátum uloženia: (v prípade, že nechcete, aby sa citácie uviedli v aktuálnom roku, zmeňte dátum na skorší termín).

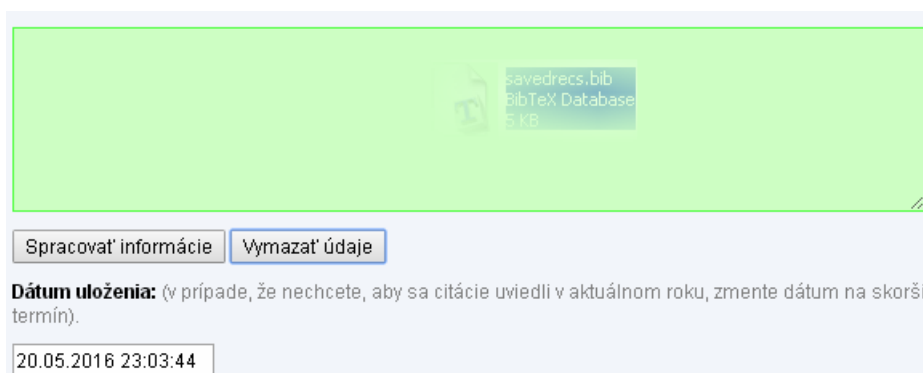
20.05.2016 23:03:44

Uložiť

Zrušiť

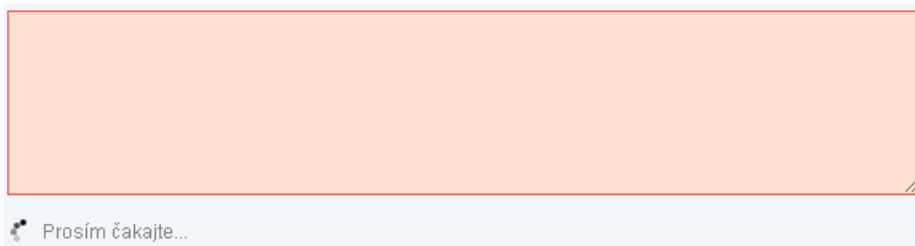
Obr. 16: Sprievodca importu bibliografických údajov z *Web of Knowledge*

Teraz už užívateľ môže začať myšou presúvať ikonu želaného súboru s údajmi až kým sa kurzorom nedostane na textové okno (obrázok 17). Zmena farby indikuje, že je sem možné vložiť súbor.



Obr. 17: Vkladanie údajov zo súboru

Následne sa začne obsah súboru načítavať, o čom informuje ikonka pod oknom (viď obrázok 18). Zvyčajne tento proces trvá len malý moment, pretože súbor sa nemusí nahrávať na server.



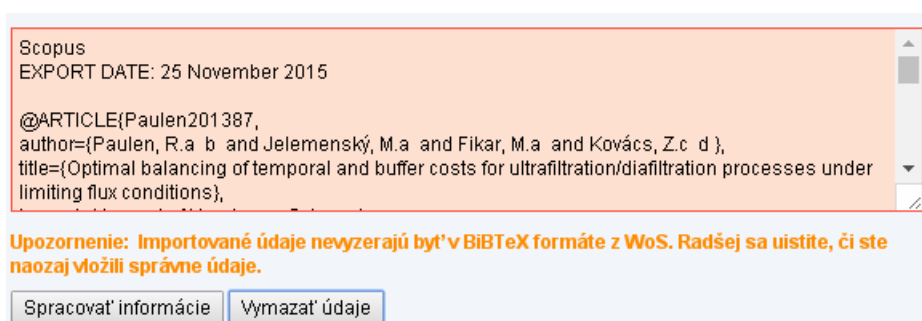
Obr. 18: Vkladanie údajov zo súboru

Po dokončení tejto operácie môže aplikácia vrátiť tri rôzne odpovede:

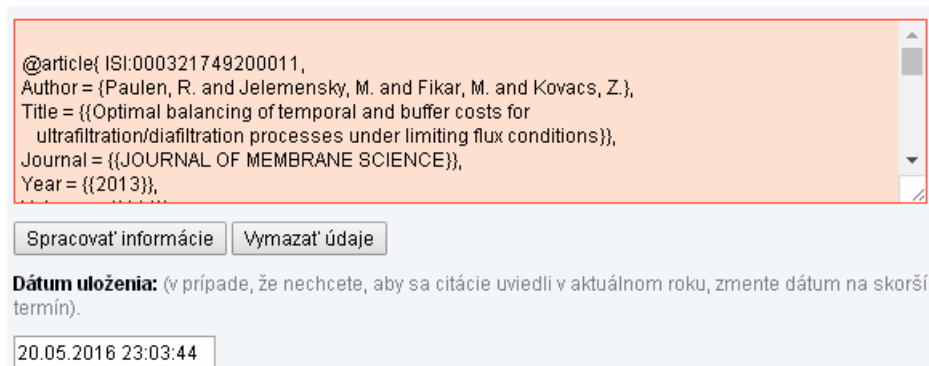
- vypíše sa chybová hláška, príčiny môžu byť buď prázdny súbor alebo zlyhanie načítavania (obrázok 19)
- údaje sa podarí načítať, ale interná kontrola údajov zistí, že údaje nie sú pravdepodobne správny formát pre import, o čom systém bude informovať varovnou správou (obrázok 20)
- všetko prebehne úspešne, údaje je možné dať spracovať pomocou tlačidla *Spracovať informácie* alebo vymazať obsah okna kliknutím na tlačidlo *Vymazať údaje* (obrázok 21)



Obr. 19: Chyba pri načítavaní súboru



Obr. 20: Upozornenie po neúspešnej kontrole formátu dát



Obr. 21: Úspešne načítané údaje

6 Modul dochádzka

Jedná sa o modul, ktorý je prepojený s elektronickým dochádzkovým systémom oddelenia. Služi zamestnancom na sledovanie vlastnej dochádzky a umožňuje im aj dodatočné doplnenie údajov.

6.1 Analýza stavu a špecifikácia požiadaviek na rozšírenie

Do modulu sa dá dostať priamo z menu chránenej sekcie a aplikácia má len jednu stránku. Na nej aktuálne prihlásený zamestnanec oddelenia vidí svoju dochádzku od dátumu spustenia dochádzkového systému až po aktuálny dátum vo forme tabuľky, kde každý riadok predstavuje jeden deň. V tomto riadku je vypísaný príslušný dátum, časy príchodu a odchodu zaznamenané v databáze a odpracovaný čas v daný deň zobrazený v hodinách a minútach. Okrem týchto údajov sa tam ešte nachádza možnosť pridať dochádzkový záznam do evidencie. Po kliknutí sa zobrazí modálne okno, v ktorom zamestnanec musí definovať čas príchodu, odchodu a príčinu dodatočného vloženia záznamu. Najskorší možný čas príchodu je 5:00 a najneskorší čas odchodu je 20:59. To ale platí iba pre prípad, že zamestnanec v danom dni nemá evidovaný žiadny záznam v systéme. Pokiaľ už v tento deň bol zamestnanec evidovaný v práci, čas príchodu, resp. odchodu už nie je možné nastaviť tak, aby zasahoval do vnútra časového intervalu existujúceho záznamu. Ako som už spomenul, pre takto dodatočne definované záznamy je treba udať aj dôvod. Ten je možné potom následne zobrazíť po kliknutí na čas príchodu alebo odchodu záznamu. Dôvod si zamestnanec vyberá z výberového zoznamu, kde nájde zoznam dostupných možností.

Aplikácia sama o sebe funguje dobre, je pomerne prehľadná a práca s ňou intuitívna. Služi však len čisto pre potreby zamestnancom, avšak dochádzka sa robí predovšetkým pre zamestnávateľa. V aktuálnom stave však neumožňuje žiadnemu užívateľovi prezerat' si dochádzku niekoho iného. Vznikla teda požiadavka na rozšírenie tohto modulu tak, aby toto umožňoval. Najvhodnejším spôsobom je umožniť export dochádzky do nejakého ľahko použiteľného formátu, pretože je zas nežiaduce, aby niekto iný mohol upravovať cudzie záznamy. Keďže pôjde teda o export údajov, rozhodol som sa pridať prístup k tejto akcii do sekcie *export údajov*, ktorá je dostupná priamo cez menu editačnej časti systému (najspodnejšia položka). K dispozícii budú dva formáty, do ktorých bude možné údaje exportovať. Prvým bude HTML formát, ktorý sa vytvorí a zobrazí v novej karte prehliadača a tým pádom v prípade potreby uložiť ho ako lokálny súbor nebude obsahovať nič iné okrem údajov zapísaných v HTML kóde. Druhým bude formát tabuľkového procesora Microsoft Excel (súbor s príponou *.xlsx*), ktorý sa bude dať uložiť na disk počítača a následne otvoriť a prípadne ďalej spracovať v príslušnom softvéri.

6.2 Realizácia exportu údajov

Samotné programovanie rozšírenia sa dá rozdeliť do dvoch základných častí. Prvou z nich je vytvorenie formuláru, pomocou ktorého používateľ zadá a špecifikuje svoju požiadavku systému. Druhá časť potom pozostáva z funkcionality, ktorá má na starosti spracovanie takejto požiadavky a následne vytvorenie požadovaného výstupu. Najskôr som sa potreboval oboznámiť s tým, ako vyzerá štruktúra tej časti databázy, ktorú budem potrebovať na získavanie potrebných údajov. Časy príchodov a odchodov sú zaznamenané v tabuľke s názvom *person_attendance*, z ktorej som potreboval nasledujúce stĺpce:

- *person_id* - identifikačné číslo osoby
- *attend_date* - dátum a čas udalosti
- *attend_po* - typ udalosti, pričom 1 znamená príchod a 2 odchod

Keďže vo výpise je dobré mať aj mená osôb, ešte som použil aj tabuľku *person*, na ktorú sa táto tabuľka odkazuje pomocou cudzieho kľúča *person_id*. Z nej stačilo vytiahnuť meno a priezvisko osoby. Keďže neskôr vznikla ešte požiadavka, aby sa v údajoch uvádzal dôvod neprítomnosti pre dni, kedy osoba nemá evidovaný príchod do práce, ešte som aj používal tabuľku *holidays*, v ktorej som využíval nasledovné stĺpce:

- *person_id* - opäť ten istý cudzí kľúč odkazujúci na osobu
- *date_from* - dátum začiatku
- *to_date* - dátum konca
- *type* - dôvod dlhodobej neprítomnosti - dovolenka, služobná cesta a iné

V neposlednom rade aj toto rozšírenie modulu bolo treba spraviť tak, aby podľa možnosti čo najviac využíval zdieľané zdroje informačného systému. Hovorím najmä o funkciách a rozširujúcich moduloch (*angl. plugins*). Zistil som, že na generovanie hárkov, ktoré používa Microsoft Excel, je v informačnom systéme dostupné rozšírenie *PHPExcel*, ktoré umožňuje prácu s týmito súbormi vrátane vytvárania nových, čo som využil v tomto module. Toto rozšírenie je PHP trieda, ktorá obsahuje rôzne metódy a atribúty, ktoré som si samozrejme musel najskôr naštudovať, aby som ich mohol využiť. Na generovanie druhého spomínaného formátu (HTML) som si vystačil s definovaním obyčajného HTML dokumentu, v ktorom som aj definoval niekoľko CSS tried a vlastností, ktorými som upravil vzhľad tabuľky s údajmi. Jednalo sa predovšetkým o farby a okraje. Obsah tohto dokumentu som samozrejme vygeneroval dynamicky pomocou vytvorenej PHP funkcie.

6.2.1 Vstupné formuláre

Ako som už vyššie spomínal, formulár na export dochádzky som umiestnil do chránenej sekcie s názvom *export údajov*. Táto sekcia obsahuje v sebe viacero formulárov na export údajov z rozličných modulov systému. Celá stránka je generovaná funkciou *export*, ktorá sa nachádza v skripte s názvom *iam_functions.php*. Táto funkcia vracia HTML kód stránky a preto na koniec procesu, v ktorom vytvára tento kód, som pridal kód na vytvorenie formuláru. Aby som to upresnil, jedná sa až o dva formuláre. Prvý slúži na export údajov dochádzky za zvolený mesiac a dá sa vypísať v dvoch formách. Jedna vypíše len odpracovaný čas pre jednotlivé dni v danom mesiaci a druhá detailnejšia ešte navyše k tomu aj čas príchodov a odchodov v každom dni. Obe formy ešte k tomu vypočítajú a zobrazia odpracovaný čas za jednotlivé týždne a aj za celý mesiac. Odpracovaný čas sa počíta v minútach a je zobrazovaný na celé hodiny a zvyškové minúty. Druhý formulár exportuje údaje pre zvolený počet mesiacov, pričom toto obdobie sa definuje zadaním intervalu, ktorého hranice tvoria zvolené mesiace. Teraz stručne popíšem rutinu vytvárania prvého formuláru, pričom pod pojmom prvok na nasledujúcich riadkoch budem myslieť element formulára a jeho popis, ktorý je v skutočnosti hlavička tabuľky bez okrajov a element sa nachádza v stĺpci pod svojim popisom:

1. pomocou zdieľanej funkcie *GetUserPermission* sa zistí, či užívateľ prihlásený v danej relácii má administrátorské práva
2. pokiaľ je užívateľ administrátor, vytvorí sa prvok s popisom *Osoba*, čo je výberový zoznam, ktorého jednotlivé položky sú zamestnanci. Tento zoznam sa získa zo spomínanej tabuľky *person*, pričom ako vyhľadávacie kritérium je také, že osoby musia byť z oddelenia automatizácie a informatizácie a musia mať v stĺpci tabuľky *active* hodnotu 1, čo odfiltruje zo zoznamu ľudí, čo už na oddelení nepracujú. Zoznam položiek výberového poľa obsahuje v sebe aj položku s názvom *Všetci*. Pokiaľ je zvolená táto možnosť, exportovať sa budú údaje o dochádzke všetkých osôb v tomto zozname. Pokiaľ užívateľ nie je administrátor, tento element sa nevytvorí a exportovať môže len svoje vlastné údaje.
3. vytvorí sa prvok s popisom *Mesiac*, je to textové pole s vlastnosťou *readonly*
4. vytvorí sa prvok s popisom *Formát*, čo je skupina dvoch prepínačov, pomocou ktorej sa zvolí formát, v ktorom budú údaje exportované
5. vytvorí sa prvok s popisom *Detail*, ide o zaškrŕávacie políčko, pokiaľ je zaškrŕnuté, tak sa údaje budú exportovať v detailnejšej forme (s časmi príchodov a odchodov), v opačnom prípade sa použije kratšia forma len s odpracovanými časmi.

Následne sa vygeneruje aj druhý formulár veľmi podobným spôsobom, len s troma rozdielmi:

- pre administrátora sa vo výberovom poli *Osoba* nenachádza položka *Všetci*
- miesto prvku *Mesiac* sa vytvorí prvok s popisom *Obdobie*, v tomto prípade sa vytvoria dve textové polia s vlastnosťou *readonly*
- prvok *Detail* sa nevytvorí

Ku každému formuláru je samozrejme vytvorené odosielacie tlačidlo s textom *Export*. Po vytvorení stránky je ešte volaná JavaScript funkcia s názvom *ExportMonthPickers*, do ktorej je zo strany servera poslaný aktuálny dátum a premenná, ktorá má hodnotu 1, ak sa používa slovenská verzia systému a 0, pokiaľ anglická. Táto funkcia iniciuje rozšírenie *jQuery UI Month Picker* (ďalej len *monthpicker*) a naviaže ho na vytvorené textové polia v spomínanom prvku *Mesiac*, resp, *Obdobie*. Keďže tieto polia boli vytvorené s vlastnosťou *readonly* a tým pádom užívateľ do nich nemôže priamo vložiť hodnotu, je zabezpečené, že sa na server bude odosielať mesiac len požadovanom formáte, ktorý je v tvare *mm/yyyy*. Je to preto, že hodnoty do týchto polí bude vkladať len *monthpicker*. Keďže dochádzka sa bude exportovať pre jednotlivé mesiace, bolo potrebné zabezpečiť, aby si užívateľ mohol vyberať medzi mesiacmi v nejakom roku. Výber musel byť obmedzený tak, že spodná hranica musí byť mesiac spustenia dochádzkového systému a horná hranica aktuálny mesiac, pretože nemá zmysel sa pokúšať exportovať dochádzku z obdobia, ktoré ešte nenastalo. Jednou z možností, ako toto zabezpečiť by bolo vytvoriť klasické výberový zoznam, pričom jeho položky by boli všetky mesiace medzi spodnou a hornou hranicou. Tento spôsob sa mi však zdá už zastaralý a postupom času by sa stal aj nepraktický a neprehľadný, pretože možnosti na výber by pribúdali. Druhou možnosťou by mohol byť v systéme dostupný kalendár (*jQuery UI Datepicker*), nastavený tak, aby vracal dátum len v mesiacoch a rokoch, avšak to by podľa môjho názoru užívateľa mýlilo, pretože kalendár by si od neho pýtal vybrať konkrétny dátum, hoci ten by bol následne konvertovaný. Preto som hľadal dostupné riešenia až som narazil na tento *monthpicker*, ktorého hlavná výhoda je, že užívateľ si z rozbaleného kalendárika vyberá priamo mesiace v danom roku a v prípade potreby môže aj meniť roky. Ďalšia výhodou je, že bol vyvinutý zo spomínaného kalendára pre jQuery. To znamená, že používa aj rovnaký dizajn, čo je žiaduce, pretože ako som spomínal, takéto kalendáriky sú už použité na webových stránkach. Tento *monthpicker* som však musel doplniť do informačného systému. V tomto prípade som použil pri inicializácii nasledovné nastavenia:

- mesiac spustenia dochádzkového systému je minimum, ktoré sa dá zvoliť
- najviac ide vybrať súčasný mesiac, ktorý je navyše aj prednastavenou hodnotou
- *monthpicker* nezobrazuje potvrdzovacie tlačidlo, čo znamená, že sa zatvorí hneď po vybratí mesiaca

- v prípade, že sa používa slovenská verzia systému, načíta sa objekt so slovenským prekladom *monthpicker*-a, ktorý som vytvoril

Tieto nastavenia platia pre prvý formulár, kde sa vyberá mesiac, pre ktorý bude exportovaná dochádzka. Ako som už spomínal, v druhom formulári som vytvoril dve textové polia, na ktoré som naviazal tiež *monthpicker*. Tieto polia slúžia ako interval, pomocou ktorého sa určí, ktoré všetky mesiace má export dochádzky obsahovať. Preto aj pri inicializácii som musel použiť mierne odlišné nastavenia oproti tým, čo som uviedol pre prvý formulár. Prednastavená hodnota prvého *monthpicker*-a (*Od*) je január súčasného roku, kým druhého (*Do*) súčasný mesiac. Navyše majú v sebe zadané funkcie, ktoré sa spustia vždy po výbere mesiaca. Pri výbere mesiaca z prvého funkcia nastaví druhému že nepôjde vybrať skorší mesiac ako ten, čo práve užívateľ zvolil. A naopak pri výbere z druhého zase funkcia nastaví prvému, že nepôjde vybrať neskorší mesiac ako aktuálne zvolený. Týmto sa zabráni prípadu, že by sa hranice intervalu prekrížili. Ostatné nastavenia vrátane jazyka sú rovnaké ako pre prípad v prvom formulári.

6.2.2 Export údajov

Po odoslaní prvého formulára sa zavolá metódou *POST* skript *export_attendance.php*, ktorý najskôr zistí zvolený formát exportu. V prípade, že ide o hárok *Microsoft Excel*, zavolá sa skript *export_attendance_excel.php*, ktorý zastaví ďalšie prebiehanie pôvodného skriptu. Funkcionalita oboch skriptov je inak rovnaká rozdiel spočíva v tom, že kým prvý skript údaje zapisuje do tabuľky v HTML kóde, verzia pre *Excel* používa metódy triedy *PHPExcel*, pomocou ktorých vytvorí tabuľku v hárku. Taktiež nepoužíva CSS ale vlastné metódy na zafarbovanie polí tabuľky. A najvýznamnejší rozdiel spočíva v tom, že *Excel* označuje polia tabuľky na základe súradníc stĺpca a riadku, pričom stĺpce sú označené jedným alebo viacerými písmenami abecedy a riadky prirodzenými číslami. Použil som preto zdieľanú funkciu *get_column_names*), ktorá vráti pole s menami pre stĺpce. Z tohto poľa už nebol problém na základe indexov aktuálneho riadku a stĺpca vytvoriť potrebnú súradnicu poľa tabuľky. Nasledujúci popis algoritmu je už zhodný pre obe skripty. Najskôr sa overí, či je definované identifikačné číslo osoby. Pokiaľ áno, použije sa, ak má hodnotu *-1*, znamená to, že bola vybraná možnosť *Všetci* a zoznam aktuálnych zamestnancov oddelenia vytiahne z databázy. Pokiaľ číslo osoby nie je definované, znamená to, že exportovať údaje dal užívateľ bez administrátorských oprávnení a v tomto prípade sa jeho číslo zoberie z aktuálnej relácie. Následne sa z odoslaných údajov zistí žiadaný mesiac a rok, zavolá sa funkcia *sviatky*, ktorá vráti zoznam dní pracovného pokoja pre daný rok a následne sa overí, či sa v danom mesiaci nejaký vyskytuje. Ak áno, čísla dní sa vložia do pomocného poľa. Následne sa z údajov vytvorí a odošle SQL dotaz, ktorý z tabuľky *person_attendance* spojenej s tabuľkou *person*, vytiahne meno osoby, číslo dňa záznamu, typ (príchod alebo odchod) a čas v minútach pre všetky záznamy v určenom mesiaci a roku pre dané číslo osoby a tieto údaje usporiada primárne podľa mien zamest-

nancov, sekundárne podľa dátumu a času, nakoniec ešte terciárne zostupne podľa typu záznamu (kvôli výpočtu časových rozdielov, aby najskôr bol odchod a potom príchod). Vytiahnuté údaje v poli sú potom doplnené o prázdné dni v mesiaci a začína fáza tvorby tabuľky. Najskôr sa vytvorí hlavička a následne začne prebiehať hlavný cyklus, ktorý tvorí telo tabuľky. V skutočnosti ide o dva vnorené cykly, pričom vonkajší iteruje cez záznamy jednotlivých osôb a vnútorný cez všetky dni v mesiaci. Prvé dva stĺpce obsahujú identifikačné číslo a meno osoby, ďalšie vypočítaný odpracovaný čas pre každý deň, priebežne sa počítajú aj hodnoty odpracovaného času za týždeň a mesiac. Týždenný je zobrazovaný po každom piatku, mesačný v poslednom stĺpci tabuľky. Víkendové dni sa do tabuľky nezapisujú, štátne sviatky sa označia pomlčkou. Pokiaľ v nejaký deň zamestnanec nemal evidovanú dochádzku, zavolá sa funkcia *attendance_check_holiday*, ktorá zistí, či v daný deň mal dovolenku, služobnú cestu alebo iný typ neprítomnosti. Ak áno, nulová hodnota sa nahradí príslušným písmenom, ktorého význam je v legende pod tabuľkou. Ak sa ani takýto záznam nenájde, nulová hodnota sa vyznačí červenou farbou. V prípade, že užívateľ vybral detailnejší výpis dochádzky, po vnútornom cykle prebehne ďalší cyklus, ktorý pre každý deň doplní farebne odlíšené časy príchodov a odchodov.

V prípade, že užívateľ odošle druhý formulár, metódou *POST* sa zavolá skript s názvom *export_attendance_period.php*, resp. ak užívateľ nezvolil formát HTML, tak jeho verzia pre *Excel* s názvom *export_attendance_period_excel.php*. V tomto skripte sú rozdiely v tom, že dáta sa načítavajú z databázy vždy len pre jednu osobu, ale všetky ktoré majú dátum väčší alebo rovný ako prvý deň mesiaca určeného ako dolná hranica intervalu a zároveň menšie alebo rovné ako posledný deň mesiaca zvoleného ako horná hranica intervalu. Vonkajší cyklus iteruje cez mesiace v tomto intervale (v prvom prípade to bol zoznam osôb) a aj tabuľka vyzerá odlišne. V prvom stĺpci je uvedený mesiac a rok, v ďalších odpracované časy v jednotlivých dňoch mesiaca. V tomto prípade sa nevynechávajú víkendy, ale sú označené rovnako ako štátne sviatky. Tabuľka vždy obsahuje stĺpce pre 31 dní, pokiaľ ich daný mesiac má menej, nadbytočné polia sú vyplnené šedou farbou. V poslednom stĺpci je uvedený odpracovaný čas za daný mesiac. Dovolenky a ostatné druhy neprítomnosti sú riešené rovnakým spôsobom ako v prvom prípade.

6.3 Uživateľská dokumentácia

V tomto prípade sa jedná o opis postupu, pomocou ktorého je užívateľ schopný exportovať údaje o dochádzke do externého súboru a vysvetlenie jednotlivých nastavení formulárov.

6.3.1 Prístup

Pokiaľ chce užívateľ exportovať údaje o dochádzke, musí byť prihlásený v systéme a kliknúť na najspodnejšiu položku menu editačnej časti s názvom *export údajov*. Otvorí sa stránka s rôznymi formulármi, ktoré sú zoradené do číslovaného zoznamu. Na spodnom konci stránky sa ako deviaty člen zoznamu nachádzajú dva formuláre slúžiacie na export dochádzky.

6.3.2 Nastavenia exportu

Dostupné možnosti závisia od toho, či má prihlásený užívateľ administrátorské práva. Ak áno, formuláre vyzerajú ako je znázornené na obrázku 22.

9. Export dochádzky

Pre daný mesiac:

Osoba	Mesiac	Formát	Detail
Čirka Ľuboš ▾	05/2016	<input type="radio"/> MS Excel <input checked="" type="radio"/> HTML	<input type="checkbox"/>

Pre zvolené obdobie:

Osoba	Obdobie	Formát
Čirka Ľuboš ▾	Od 01/2016 do 05/2016	<input checked="" type="radio"/> MS Excel <input type="radio"/> HTML

Obr. 22: Formuláre pre export dochádzky pri administrátorskom prístupe

Vrchný formulár slúži na export údajov o dochádzke za zvolený mesiac. Vo výberovom zozname označenom *Osoba* je možné si zvoliť zamestnanca oddelenia, ktorého dochádzka sa bude exportovať. Je možné vybrať aj všetkých zamestnancov naraz pomocou prvej položky zoznamu s označením *Všetci*. V poli *Mesiac* sa definuje pomocou špeciálneho kalendárika konkrétny mesiac v želanom roku, z ktorého sa budú exportovať dáta, pomocou prepínačov je možné vybrať formát, do ktorého sa údaje uložia, na výber je hárok programu *Microsoft Excel* alebo jednoduchá HTML stránka, ktorá sa otvorí v novej karte prehliadača. Zaškrtnutím políčka *Detail* si užívateľ určuje, či chce exportovať len údaje o počte odpracovaných hodín alebo aj detailný výpis časov príchodu a odchodu pre každý deň. Spodný formulár slúži na export dochádzky za obdobie viacerých mesiacov, ktoré je možné nastaviť pomocou polí *Od* a *Do*. Kalendárik si automaticky sledujú aktuálne

zvolenú hodnotu a neponúknú na výber mesiac, ktorý bol skôr, resp. neskôr než aký je vybraný v druhom poli. Tento druh exportu dovoľuje vybrať len jednu osobu.

Pokiaľ prihlásený užívateľ nemá administrátorské práva, môže exportovať len svoje vlastné údaje, preto uvidí formuláre v skrátenej forme, ktorá je na obrázku 23.

9. Export dochádzky

Pre daný mesiac:

Mesiac	Formát	Detail
05/2016	<input checked="" type="radio"/> MS Excel <input type="radio"/> HTML	<input type="checkbox"/>

Export

Pre zvolené obdobie:

Obdobie		Formát
Od 01/2016	do 05/2016	<input checked="" type="radio"/> MS Excel <input type="radio"/> HTML

Export

Obr. 23: Formuláre pre export dochádzky pri bežnom prístupe

6.3.3 Výber mesiaca

Ako už bolo spomenuté v sekcii 6.3.2, mesiace sa vyberajú pomocou špeciálnych kalendárikov, ktoré sa zobrazia po kliknutí na pole s údajom o mesiaci. Tento kalendárik obsahuje na výber priamo mesiace (viď obrázok 24). Kliknutím na tlačidlo so želaným mesiacom sa mesiac automaticky vyberie a kalendárik zatvorí. V záhlaví je zobrazený aktuálne nastavený rok, pre pohyb medzi rokmi slúžia tlačidlá so šípkami naľavo a napravo od textu s rokom. Po kliknutí na tento text sa namiesto mesiacov zobrazí výber rokov, zvolením nejakého sa užívateľ dostane späť na výber mesiacov. Kalendárik nedovolí vybrať skorší mesiac ako ten, v ktorom bol dochádzkový systém uvedený do prevádzky a neskorší ako je súčasný mesiac v súčasnom roku.

Export Resetovať

8. Generovanie údajov pre /

Annual report za rok: 201

Generuj

9. Export dochádzky

Pre daný mesiac:

Osoba

Čirka Ľuboš

02/2016

MS Excel HTML

Export

Pre zvolené obdobie:

Osoba	Obdobie	Formát
Čirka Ľuboš	Od 01/2016 do 05/2016	MS Excel HTML

Export

Obr. 24: Špeciálny kalendár na výber mesiaca v roku (jQuery UI Month Picker)

6.3.4 Výsledok exportu

Závisí od zvoleného formátu a ostatných nastavení, ale v každom prípade sa jedná o tabuľku s odpracovaným časom pre každý deň okrem štátnych sviatkov a dní, kedy nebol zamestnanec prítomný na pracovisku z jedného z dôvodov uvedených v legende pod tabuľkou. V prípade, že si užívateľ zvolí formát HTML a vyexportuje údaje za určitý mesiac bez zaškrtnutia možnosti *Detail*, vygeneruje sa mu tabuľka, ktorej začiatkové a posledné stĺpce sú znázornené na obrázku 25.

Dochádzka - 04/2016

Osobné číslo	Priezvisko a meno	1	Týždenný súčet	4	5	28	29	Týždenný súčet	Mesačný súčet
402727	Čírka Euboš	12:23	12:23	10:30	10:36	12:32	13:08	60:47	248:02

Vysvetlivky:

- D - dovolenka
- Sc - služobná cesta
- PN - práceneschopnosť
- Md - materská dovolenka
- Rp - rodičovský príspevok
- Nv - neplatené voľno
- Pv - náhradné (platené) voľno
- Pa - návšteva u lekára (paragraf)

Obr. 25: Útržok tabuľky s údajmi o dochádzke (odpracované časy) za vybraný mesiac v HTML dokumente)

Prípád so zobrazenými časmi príchodov a odchodov (zaškrtnutá možnosť *Detail*) je na obrázku 26

Dochádzka - 04/2016

Osobné číslo	Priezvisko a meno	1	Týždenný súčet	4	5	28	29	Týždenný súčet	Mesačný súčet
402727	Čírka Euboš	12:23	12:23	10:30	10:36	12:32	13:08	60:47	248:02
Príchody/odchody		06:37 19:00		06:10 16:40	06:17 16:53	06:45 19:17	06:22 19:30		

Vysvetlivky:

- D - dovolenka
- Sc - služobná cesta
- PN - práceneschopnosť
- Md - materská dovolenka
- Rp - rodičovský príspevok
- Nv - neplatené voľno
- Pv - náhradné (platené) voľno
- Pa - návšteva u lekára (paragraf)

Obr. 26: Útržok tabuľky s údajmi o dochádzke (odpracované časy, príchody, odchody) za vybraný mesiac v HTML dokumente)

V prípade, že používateľ použije druhý formulár na export dochádzky pre obdobie niekoľkých mesiacov, tabuľka bude vyzerat' odlišne (obrázok 27).

Dochádzka - Čírka Ľuboš

Mesiac	1	2	3	4	5	6	7	8	9				
1/2016	-	-	-	D	D	-	D	D	-	29	30	31	Mesačný súčet
2/2016	8:05	10:04	11:04	8:07	D	-	-	10:45	Sc	10:22	-	-	138:35
3/2016	10:48	11:31	12:17	6:47	-	-	11:02	10:12	11:06	11:58			201:55
4/2016	12:23	-	-	10:30	10:36	12:27	12:30	12:38	-	D	10:40	12:10	196:06
5/2016	-	13:28	9:33	13:13	12:42	13:04	-	-	10:10	13:08	-		248:02
										-	0:00	0:00	81:10

Vysvetlivky:

- D - dovolenka
- Sc - služobná cesta
- PN - práceneschopnosť
- Md - materská dovolenka
- Rp - rodičovský príspevok
- Nv - neplatené voľno
- Pv - náhradné (platené) voľno
- Pa - návšteva u lekára (paragraf)

Obr. 27: Útržok tabuľky s údajmi o dochádzke za obdobie viacerých mesiacov v HTML dokumente)

Pokiaľ bol zvolený export do formátu *Excel*, prehliadač ponúkne možnosť uložiť súbor na zariadenie užívateľa. Preddefinovaný formát názvu súboru je v prípade prvého formuláru *Dochadzka_m_yyyy_[id osoby].xlsx*, v prípade druhého s obdobím viacerých mesiacov je formát názvu súboru *Dochadzka_[id osoby]_m_yyyy-m_yyyy.xlsx*.

7 Záver

Prvým cieľom diplomovej práce bolo vytvoriť nový modul pre informačný systém ÚIAM. Modul bude slúžiť na úpravu údajov o predmetoch uložených v databáze systému. Tá obsahuje predmety zabezpečované ústavom. Modul bol plne integrovaný do zabezpečenej, verejnosti neprístupnej časti systému. Výber predmetov bol vyriešený cez vyhľadávanie, ktoré filtruje zobrazené predmety na základe zadanej časti názvu alebo kódu predmetu. Úprave údajov o predmete bola vyriešená prostredníctvom formuláru, pričom každý jeho element je jeden údaj. Kód na vytváranie formulára som sa snažil robiť čo najviac modulárne, aby prípadná definícia, úprava lebo premiestnenie elementu neznamenal dlhé prepisovanie kódu. Možnosť o aktualizáciu predmetu údajmi z AIS som vyriešil pomocou parsovania údajov zo stránky verejného katalógu predmetov, pričom riešenie sa mi podarilo zautomatizovať do takej miery, že jediná akcia, ktorú musí užívateľ spraviť sám, je skopírovanie URL adresy stránky s predmetom a jej vloženie do textového poľa. Okrem tohto som ešte do modulu pridal akcie na vytvorenie nového predmetu a potom ešte aj na zmazanie existujúceho. Modul som vytvoril v dvoch jazykových verziách, slovenskej a anglickej, pričom aktuálna verzia vplýva aj na vyhľadávanie v názvoch predmetov. V budúcnosti by sa malo hlavne sledovať, aké výsledky dáva parsovanie údajov z AIS a v prípade potreby urobiť potrebné zmeny.

Druhým cieľom práce bolo vylepšiť existujúce moduly informačného systému, hlavne išlo o vylepšenia z hľadiska ovládateľnosti, prípadne doplnenie vhodnej funkcionality. Mne sa podarilo vylepšiť dva moduly. Prvým bol modul na pridávanie a úpravu citácií. V ňom som sa hlavne venoval vylepšeniu ovládania sprievodcu na import citácií z externých databáz. Hlavné vylepšenie spočívalo v možnosti načítavania bibliografických údajov priamo zo súboru metódou *drag and drop*, pričom táto funkcionality prebieha na strane klienta a nie je potrebné, aby bol súbor prenášaný na server. Okrem toho som ešte pridal podporu RIS formátu údajov a kontrolu načítaných dát ešte pred odoslaním na server. Druhý modul, ktorý som vylepšoval, bola evidencia dochádzky. V tomto prípade som pridal funkcionality na export údajov o dochádzke. Dostupné sú dva formáty, jeden je *HTML* dokument otvorený v novej karte prehliadača a druhý je hárok programu *Microsoft Excel* (*xlsx* formát). Údaje je možné exportovať buď za jeden zvolený mesiac alebo obdobie viacerých mesiacov. Pri exporte mesačnej dochádzky je na výber aj možnosť detailnejšieho výpisu, kedy sa okrem odpracovaných časov pre konkrétne dni vypíšu aj časy príchodov a odchodov. V budúcnosti by bolo možné pokračovať s vylepšeniami ďalších modulov, či už z hľadiska lepšej ovládateľnosti alebo vylepšenej funkcionality.

Použitá literatúra

- [1] Wikipedia. HTML. [online], [cit. 2016-05-18] Dostupné na internete:
<<http://en.wikipedia.org/wiki/HTML>>
- [2] W3C. HTML5. [online], [cit. 2016-05-18] Dostupné na internete:
<<https://www.w3.org/TR/html5/>>
- [3] The Apache Software Foundation. Apache HTTP Server Project. [online], [cit. 2016-05-18] Dostupné na internete: <<http://httpd.apache.org/>>
- [4] Leiss O. - Schmidt J. PHP v praxi pro začátečníky a mírně pokročilé, 1. vyd. Praha: Grada Publishing, a.s., 2010. 13,14,103,104,105,107 s. ISBN: 978-80-247-3060-8
- [5] Gutmans, A. - Bakken S. - Rethans D. PHP 5 Power Programming, 1. vyd. Indianapolis: Prentice Hall PTC, 2004. 226-227 s. ISBN: 0-131-47149-X
- [6] Szabó Peter. 2005. Databázové a informačné systémy - relačný databázový systém. [online], [cit. 2016-05-18] Dostupné na internete:
<<http://www.spseke.sk/tutor/prednasky/dbs/rdbms.html>>
- [7] Wikipedia. MySQL. [online], [cit. 2016-05-18] Dostupné na internete:
<<http://en.wikipedia.org/wiki/MySQL>>
- [8] Zakas, Nicholas C. Professional JavaScript for Web Developers, 2. vyd. Indianapolis: Wiley Publishing, Inc., 2009. 1-7 s. ISBN: 978-0-470-22780-0
- [9] Zakas, N.C. - McPeak J. - Fawcett J. Professional Ajax, 2. vyd. Indianapolis: Wiley Publishing, Inc., 2007. 1-7 s. ISBN: 978-0-470-10949-6
- [10] Lindley Cody. jQuery Cookbook: Solutions & Examples for jQuery Developers. 1. vyd. Sebastopol: O'Reilly Media, Inc., 2009. 2-4 s. ISBN: 978-0-596-15977-1
- [11] ECMA-404, ECMAScript® Language Specification, 1. edícia (Október 2013)
- [12] Albert B. - Segura R. jQuery UI Month Picker Plugin. [online], [cit. 2016-05-22]. Dostupné na internete
<<https://github.com/KidSysco/jquery-ui-month-picker/wiki>>.