

Univerzita Pardubice  
Fakulta ekonomicko-správní

System pro evidenci bakalářských a diplomových prací

Lada Pěničková

Bakalářská práce

2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lada PĚNIČKOVÁ**  
Osobní číslo: **E05714**  
Studijní program: **B6209 Systémové inženýrství a informatika**  
Studijní obor: **Regionální a informační management**  
Název tématu: **Systém pro evidenci bakalářských a diplomových prací**  
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

### Z á s a d y p r o v y p r a c o v á n í :

Identifikace a formulace požadavků na systém  
Návrh systému pro evidenci bakalářských a diplomových prací  
Implementace navrženého systému pro evidenci bakalářských a diplomových prací

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

**BRÁZA, J. PHP: začínáme programovat. Praha: Grada Publishing, 2005. 240 s. ISBN 80-247-1146-X.**

**HAROLD, E. R. XML Bible. 2. vydání, New York: Hungry Minds, 2001. 1151 s. ISBN 07-6454-760-7.**

**POKORNÝ, J., HALAŠKA, M. Databázové systémy. 2. přeprac. vyd. Praha: Vydavatelství ČVUT, 2003. 148 s. ISBN 80-01-02789-9.**

**SMIDEK, V., BERIT, J., MRÁZEK, L. PHP v příkladech. České Budějovice: Kopp, 2001. 120 s. ISBN 80-7232-162-5.**

*Miloslav Hub*

Vedoucí bakalářské práce:

**Ing. Miloslav Hub, Ph.D.**

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **5. října 2009**

Termín odevzdání bakalářské práce: **30. dubna 2010**

*Renáta Myšková*

doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.

*Jiří Křupka*

doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 5. října 2009

Prohlašuji:

Tuto práci jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 29. 6. 2010

Lada Pěničková

## PODĚKOVÁNÍ

Na tomto místě bych chtěla poděkovat vedoucímu bakalářské práce Ing. Miloslavu Hubovi, Ph.D. za jeho trpělivost a důvěru. Dále děkuji Ing. Lukáši Čeganovi a Ing. Lukáši Slánskému z Fakulty elektrotechniky a informatiky a celému kolektivu z Informačního centra Univerzity Pardubice za cenné rady a připomínky. Rovněž děkuji všem pracovníkům na vysokých školách a univerzitách po celé České republice, kteří odpověděli na mé dotazy ohledně vypisování kvalifikačních prací na jejich ústavech či katedrách.

Dále děkuji členům vývojářské komunity PRADO a členům četných diskusí za ochotu pomoci při řešení problémů. V neposlední řadě děkuji vývojářům i uživatelům sázecího systému  $\text{\LaTeX}$ , ve kterém je napsána tato práce, a Pavlu Tylovi za zapůjčení počítače a testování vytvářené aplikace. Největší dík patří mým rodičům za finanční i duševní podporu během mého dosavadního studia.

## **ANOTACE**

Výběr tématu bakalářské či diplomové práce je jedním z důležitých úkolů, které musí každý student vyřešit pro úspěšné dokončení svého studia. Výběr vhodné práce mu může do určité míry usnadnit centralizovaný systém pro evidenci bakalářských a diplomových prací. Stejně tak učiteli takový systém ulehčí práci s vypisováním a schvalováním prací. Cílem této bakalářské práce je navrhnout a implementovat takový systém, který bude užitečný jak pro studenty, tak pro pedagogy a zároveň bude plnit základní požadavky kladené na informační systémy – přehlednost, jednoduchost ovládání a bezpečnost.

## **KLÍČOVÁ SLOVA**

evidence, evidenční systém, návrh systému, bakalářské práce, diplomové práce

## **TITLE**

System for Evidence of Bachelor and Diploma Thesis

## **ANNOTATION**

Choosing of bachelor or diploma thesis theme plays an important role that every student must solve for successful graduation. Centralized system for evidence of bachelor and diploma thesis can measurably help with choosing of suitable work. The goal of this work is to propose such system that will be useful for both students and educators and that will realize base requirements posed for information systems – lucidity, ease of control and security.

## **KEYWORDS**

evidence, system for evidence, desing of system, bachleor thesis, diploma thesis

# OBSAH

<b>Seznam obrázků</b> . . . . .	<b>8</b>
<b>Seznam zdrojových kódů</b> . . . . .	<b>9</b>
<b>Seznam použitých zkratk</b> . . . . .	<b>10</b>
<b>Úvod</b> . . . . .	<b>12</b>
<b>1 Analýza současného stavu vypisování studentských prací</b> . . . . .	<b>13</b>
1.1 Současný stav v České republice . . . . .	13
1.2 Současný stav na Univerzitě Pardubice . . . . .	14
1.3 Současný stav na Fakultě ekonomicko-správní . . . . .	15
<b>2 Definice parametrů systému</b> . . . . .	<b>18</b>
2.1 Nejdůležitější funkce . . . . .	19
2.1.1 První etapa – Návrh témat . . . . .	19
2.1.2 Druhá etapa – Schvalování témat . . . . .	19
2.1.3 Třetí etapa – Publikování témat . . . . .	19
2.2 Ostatní parametry . . . . .	19
<b>3 Návrh systému</b> . . . . .	<b>21</b>
3.1 Kontextový diagram . . . . .	21
3.2 Diagram datových toků . . . . .	23
3.3 Entitně relační diagram . . . . .	24
<b>4 Implementace SEP</b> . . . . .	<b>26</b>
4.1 Výběr technologie . . . . .	26
4.2 Nastavení PRADO . . . . .	27
4.3 Tvorba stránek . . . . .	29
4.4 Práce s databází . . . . .	31
4.5 Role nepřihlášeného uživatele . . . . .	35
4.6 Autentizace a autorizace . . . . .	40
4.7 Role studenta . . . . .	41
4.8 Role pedagoga . . . . .	43
4.9 Role vedoucího ústavu . . . . .	46
4.10 Další společné vlastnosti systému . . . . .	46
4.10.1 Tisk . . . . .	46

4.10.2 Vyhledávání . . . . .	48
4.10.3 RSS kanál . . . . .	50
<b>Závěr . . . . .</b>	<b>53</b>
<b>Použitá literatura . . . . .</b>	<b>54</b>
<b>Seznam příloh . . . . .</b>	<b>56</b>



## SEZNAM OBRÁZKŮ

1	Vytvoření zadávacího listu . . . . .	15
2	Vytvoření zadání . . . . .	15
3	Odevzdání práce . . . . .	16
4	Vytvoření posudku . . . . .	16
5	Diagram datových toků současného stavu na FES . . . . .	17
6	Kontextový diagram SEP . . . . .	21
7	Entitně relační diagram databázového systému . . . . .	24
8	Adresářová struktura SEP . . . . .	28
9	Rezervace práce studentem . . . . .	42
10	Obsazení práce studentem . . . . .	44
11	Upozornění při uzavírání neobsazené práce . . . . .	46
12	Schválení práce vedoucím ústavu . . . . .	47
13	Zobrazený výsledek vyhledávání . . . . .	49
14	Diagram datových toků první úrovně . . . . .	i
15	Diagram datových toků – zobrazení prací . . . . .	iv
16	Diagram datových toků – editace údajů o práci . . . . .	vi
17	Diagram datových toků – editace nastavení . . . . .	ix

## SEZNAM ZDROJOVÝCH KÓDŮ

1	Základní varianta souboru index.php . . . . .	27
2	Nastavení domovské stránky a implicitního rozvržení stránek v souboru application.xml . . . . .	28
3	Nastavení přístupu na stránky pomocí config.xml . . . . .	29
4	Implicitní rozvržení stránek v souboru MainLayout.tpl . . . . .	30
5	Základní varianta souboru MainLayout.php . . . . .	31
6	Contact.page . . . . .	31
7	Contact.php . . . . .	31
8	Nastavení databáze v souboru application.xml . . . . .	32
9	Namapování tabulky prace pomocí PraceRecord.php . . . . .	32
10	Komponenta TRepeater a TPager v souboru Home.page . . . . .	36
11	Obsah rendereru HomePraceBPRenderer.tpl . . . . .	37
12	Načtení dat do datagridu v souboru TabListBPrace.php . . . . .	37
13	Ukázka použití komponenty TDataGrid v souboru TabListBPrace.page . . . . .	38
14	Validace NetID a hesla a vytvoření session v souboru SEPUser.php . . . . .	40
15	Nadefinování parametrů pro odeslání e-mailu v souboru RezervacePrace.php . . . . .	41
16	Posílání e-mailů nadefinované v souboru posliMail.php . . . . .	42
17	Funkce pro vyhledávání v souboru HledatPrace.php . . . . .	49
18	Vybrání všech pedagogů v tabulce prace – soubor HledatPedagog.php . . . . .	50
19	Načtení dat pro RSS kanál v souboru FeedProvider.php . . . . .	50
20	Tvorba XML souboru RSS kanálu v souboru FeedProvider.php . . . . .	51
21	Nastavení služby RSS kanálu v souboru application.xml . . . . .	51

## SEZNAM ZKRATEK A POJMŮ

<b>AJAX</b>	Asynchronous JavaScript and XML je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání.
<b>CAS</b>	Centrální autentizační systém.
<b>CSS</b>	Cascading Style Sheets – jazyk pro popis způsobu zobrazení webových dokumentů.
<b>DOMPDF</b>	Document Object Model PDF – PHP knihovna pro převod XHTML do PDF využívající objektový model dokumentu.
<b>FEI</b>	Fakulta elektrotechniky a informatiky.
<b>FES</b>	Fakulta ekonomicko-správní.
<b>FOP</b>	Formatting Object Processor – open-source formátovací procesor od sdružení Apache.
<b>FPDF</b>	Free PDF – PHP knihovna pro přímý přístup k PDF.
<b>HTML</b>	HyperText Markup Language je značkovací jazyk pro hypertext.
<b>I18N</b>	Internacionalizace, příprava aplikace na podporu různých kulturních zvyklostí, např. různého formátu datumu a času. Zkratka i18n vznikla z anglického slova internationalization, které má mezi i a n 18 písmen.
<b>IS</b>	Informační systém.
<b>L10N</b>	Lokalizace, proces, který následuje po I18N, spočívá především v přeložení textů do konkrétního jazyka.
<b>MySQL</b>	Multiplatformní systém řízení báze dat.
<b>NetID</b>	Identifikační číslo přidělené všem zaměstnancům a studentům Univerzity Pardubice.
<b>PDF</b>	Portable Document Format – přenosný formát dokumentů vyvinutý společností Adobe.
<b>PDO</b>	PHP Data Object – abstraktní vrstva pro komunikaci s databázemi.
<b>PHP</b>	Rekurzivní zkratka PHP: Hypertext Preprocessor, původně Personal Home Page. Označuje skriptovací programovací jazyk určený především pro programování dynamických webových stránek.

<b>PRADO</b>	PHP Rapid Application Development Object-oriented – rychlé objektově orientované programování aplikací v PHP.
<b>RSS</b>	Rich Site Summary – XML formát pro sledování webového obsahu.
<b>RTF</b>	Rich Text Format – značkovací jazyk k formátování textu vyvinutý společností Microsoft.
<b>SEP</b>	Systém pro evidenci bakalářských a diplomových prací.
<b>STAG</b>	IS/STAG je informační systém pro evidenci studijní agendy vysoké školy nebo univerzity.
<b>SŘBD</b>	Systém řízení báze dat, databázový server.
<b>TTF2UFM</b>	True Type Font to Unicode Font Metrics – utilita pro převod True Type fontů na Unicode metrický font.
<b>UFPDF</b>	Unicode Free PDF – rozšíření knihovny FPDF o kódování Unicode.
<b>WS STAG</b>	Webové služby IS/STAG.
<b>XHTML</b>	eXtensible HyperText Markup Language je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW vyvinutý W3C.
<b>XSL-FO</b>	Extensible Stylesheet Language – Formatting Objects.
<b>XSLT</b>	Extensible Stylesheet Language Transformations – slouží ke zformátování zdrojových dat v XML a převodu do libovolného jiného formátu.
<b>Xalan</b>	Open-source XSLT procesor od sdružení Apache.

## Úvod

O systematické shromažďování údajů o bakalářských a diplomových pracích na Univerzitě Pardubice se již několik let stará informační systém IS/STAG [13]. V současné době však neexistuje jednotný systém pro zadávání a editaci témat vyučujícími, schvalování navržených témat vedoucími ústavů a zápis studentů na jednotlivá témata, dále např. vyhledávání volných témat... Proto musí učitel navržené téma předložit schvalujícímu v tištěné podobě nebo formou e-mailu. Stejným způsobem probíhá i informování učitele o výsledku schvalování. Řešením pro studenty je pečlivé sledování nástěnek, obcházení vyučujících či psaní e-mailů s žádostí o informaci, zda je vybrané téma volné či ne.

Východiskem z tohoto stavu je vytvoření jednotného systému pro evidenci a správu bakalářských a diplomových prací (*SEP*). Tento systém si nebude klást za cíl nahradit modul z IS/STAG, ale usnadnit návrh a schválení nových témat vyučujícím a nalezení a rezervaci vhodného tématu studentům – to vše se splněním základních požadavků kladených na informační systémy – přehlednost, jednoduchost ovládání a bezpečnost.

Tato bakalářská práce je rozdělena do čtyř hlavních částí. První seznamuje se současným stavem dané problematiky. Následuje shrnutí cíle systému včetně požadavků, které jsou na systém kladeny. V další části je popsán postup při návrhu systému. Poslední část se soustředí na praktické problémy implementace a popis tvorby systému spolu se stručným návodem, jak systém používat. Závěr podává zhodnocení navrženého systému, shrnuje jeho výhody a zároveň omezení i možná úskalí, která je nutné odstranit (např. za pomoci testování systému). Bakalářská práce je doplněna výpisem nejzajímavějších částí zdrojového kódu s jejich popisem.

# 1 Analýza současného stavu vypisování studentských prací

## 1.1 Současný stav v České republice

Několik vysokých škol v České republice používá svůj vlastní informační systém (IS), kterým spravuje veškerou studijní agendu [7], [9], [11], [12], [15], [16], [17]. V rámci těchto IS mohou pedagogičtí pracovníci vypisovat témata bakalářských a diplomových prací a po schválení garantem se na ně mohou zapisovat studenti.

Na Ostravské univerzitě používají IS/STAG podobně jako na Univerzitě Pardubice. Pro vyhlášení a zadávání témat vytvořili svůj IS DIPL 2, který umožňuje [2]:

- **studentovi**

- vybírání tématu z volných témat,
- vytvoření volného tématu,
- generování zadání,
- vyplnění údajů o práci,
- generování licenční smlouvy,
- prohlížení posudků;

- **učiteli**

- prohlížení všech prací,
- prohlížení vlastních témat a prací,
- přidání volného tématu,
- vložení posudku;

- **sekretářce**

- prohlížení všech témat a prací,
- prohlížení volných témat,
- přidání volného tématu,
- vytváření tiskových výstupů,
- přihlášení se jako student.

Hlavním nedostatkem tohoto systému je absence schvalovacího procesu. Tedy možnost vedoucího ústavu/katedry ovlivnit podobu vypisovaných témat před jejich zveřejněním.

Ostatní vysoké školy a univerzity vypisují témata svých kvalifikačních prací na nástěnky kateder/ústavů nebo na webové stránky školy. Studenti se po zveřejnění témat buď zapisují přímo na nástěnky k vybrané práci, nebo kontaktují vedoucího práce, který dané téma studentovi přiřadí. Témata vypsaná učitelem často nepodléhají schválení vedoucím ústavu či katedry na rozdíl od témat, se kterými přijdou sami studenti.

## 1.2 Současný stav na Univerzitě Pardubice

Univerzita Pardubice je jednou z dvanácti vysokých škol a univerzit v České republice, která používá IS/STAG. Ze sedmi jejích fakult pouze jedna používá IS pro evidenci bakalářských a diplomových prací. Tou je Fakulta elektrotechniky a informatiky (FEI), která si vytvořila systém speciálně pro své potřeby. Jedná se o samostatnou aplikaci, jejíž běhové prostředí je webové rozhraní vytvořené v PHP napojené na databázi MySQL. Uživatelé se v aplikaci autentizují pomocí svého NetID, které je ověřováno přes centrální autentizační systém (CAS) [osobní konzultace].

Uživatelé k systému mohou přistupovat ve 4 rolích a podle role mají následující práva [osobní konzultace]:

- **vyučující**
  - prohlížení vypsaných prací ostatních,
  - editování vlastních prací,
  - archivování a odarchivování prací,
  - přidělení a odebrání tématu studentovi,
- **student**
  - prohlížení volných prací,
  - přihlášení a odhlášení k/od tématu,
- **recenzent**
  - schvalování témat,
  - označování témat k přepracování,
- **administrátor**
  - vytváření tiskových výstupů,
  - nastavování recenzentů.

O každé práci se eviduje její název, obor, pro který je určená, rozsah, termín vypracování (resp. obhajoby), zásady pro zpracování, doporučená literatura, vedoucí práce a přihlášený student.

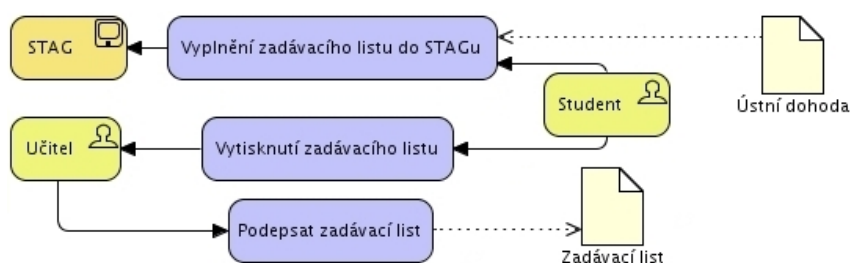
### 1.3 Současný stav na Fakultě ekonomicko-správní

Na Fakultě ekonomicko-správní (FES) se nepoužívá žádný IS pro evidenci kvalifikačních prací. Akademičtí pracovníci směřují vypsaná témata závěrečných prací podle svého zaměření a svých vlastních projektů. Vždy musí řešit aktuální problém. Podle zákona o vysokých školách si téma může navrhnout také student [18]. Musí ale platit, že k danému tématu má vedoucího z FES. Všechna navrhovaná témata musí obsahovat krátkou anotaci cíle práce (příp. obsahu) [14]. Vedoucí ústavu stanovuje přesný počet kvalifikačních prací (zvlášť bakalářské a zvlášť diplomové práce), které může vypsát jeden pedagog. Tento počet se liší u asistentů, odborných asistentů, docentů a profesorů.

Podmínkou zařazení témat do seznamu závěrečných prací je souhlas příslušného garantujícího ústavu. Následuje zveřejnění témat na nástěnkách ústavů a na webových stránkách univerzity. To se děje obvykle k prvnímu květnu.

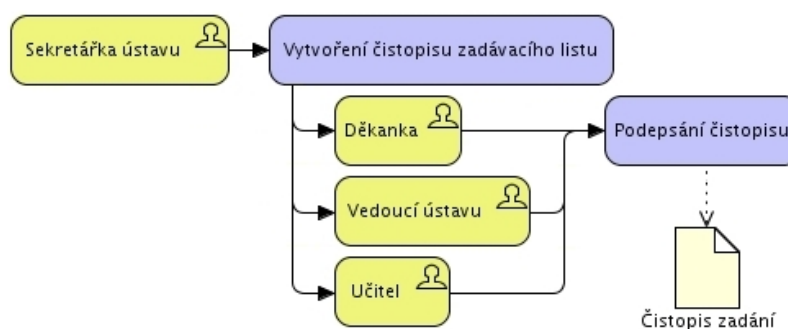
Student druhého ročníku bakalářského studia nebo prvního ročníku magisterského studia si nejprve vybere téma, dohodne přesný název práce, její osnovu a příslušnou literaturu se svým vedoucím závěrečné práce do konce května [14]. Vedoucí práce vyznačí na nástěnce, že je dané téma obsazené a informuje o tom i tajemníka ústavu. Ten potom předá podklady správci webových stránek, který je aktualizuje.

Povinností studenta je také vyplnit sjednané údaje do pracovní verze zadávacího listu ve



Obrázek 1: Vytvoření zadávacího listu

Zdroj: vlastní



Obrázek 2: Vytvoření zadání

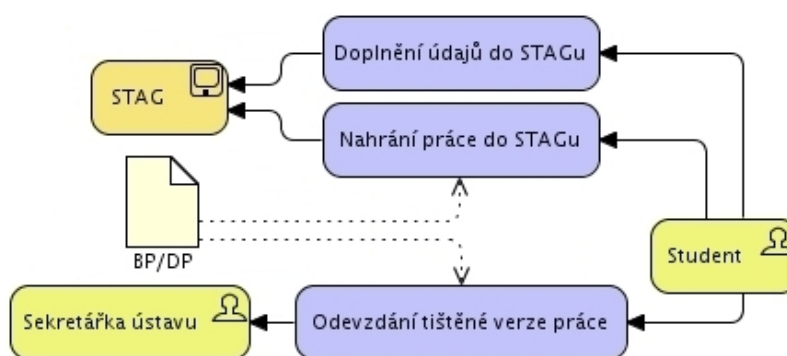
Zdroj: vlastní



STAGu, vytisknout formulář a nechat ho odsouhlasit a podepsat svému vedoucímu práce. Podepsaný formulář odevzdá na sekretariát příslušného ústavu nejpozději do konce června [14].

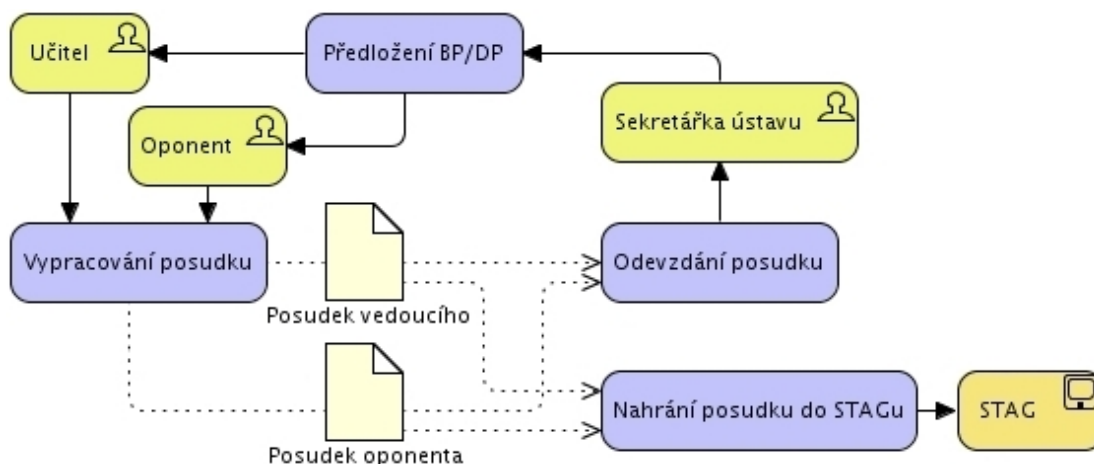
Na sekretariátech ústavů vytvoří čistopis zadávacího listu a nechají ho podepsat děkankou, vedoucím ústavu a vedoucím práce. Poté (do konce října) si jej student vyzvedne a do 1. května následujícího roku má čas na zpracování daného tématu a odevzdání práce. Hotovou závěrečnou práci odevzdává v požadovaném počtu tištěných verzí na příslušném ústavu fakulty. Jejich součástí bude i „Zadání BP/DP“ a prohlášení studenta. Zároveň nahraje plnou verzi své práce v elektronické podobě do STAGu do „Formuláře na doplnění údajů o DP/BP studentem“. Nesmí také zapomenout na doplnění dalších upřesňujících údajů k práci do STAGu, jinak nebude moci svázanou práci odevzdat [14].

Současný stav problematiky spojené se zadáváním závěrečných prací na FES znázorňuje DFD na obr. 5. Procesy vyžadující podrobnější popis jsou zvětšeny na obr. 1 – 4. Diagramy se snaží sekvenčně zobrazit činnosti od vymyšlení tématu po přečtení uložených posudků.



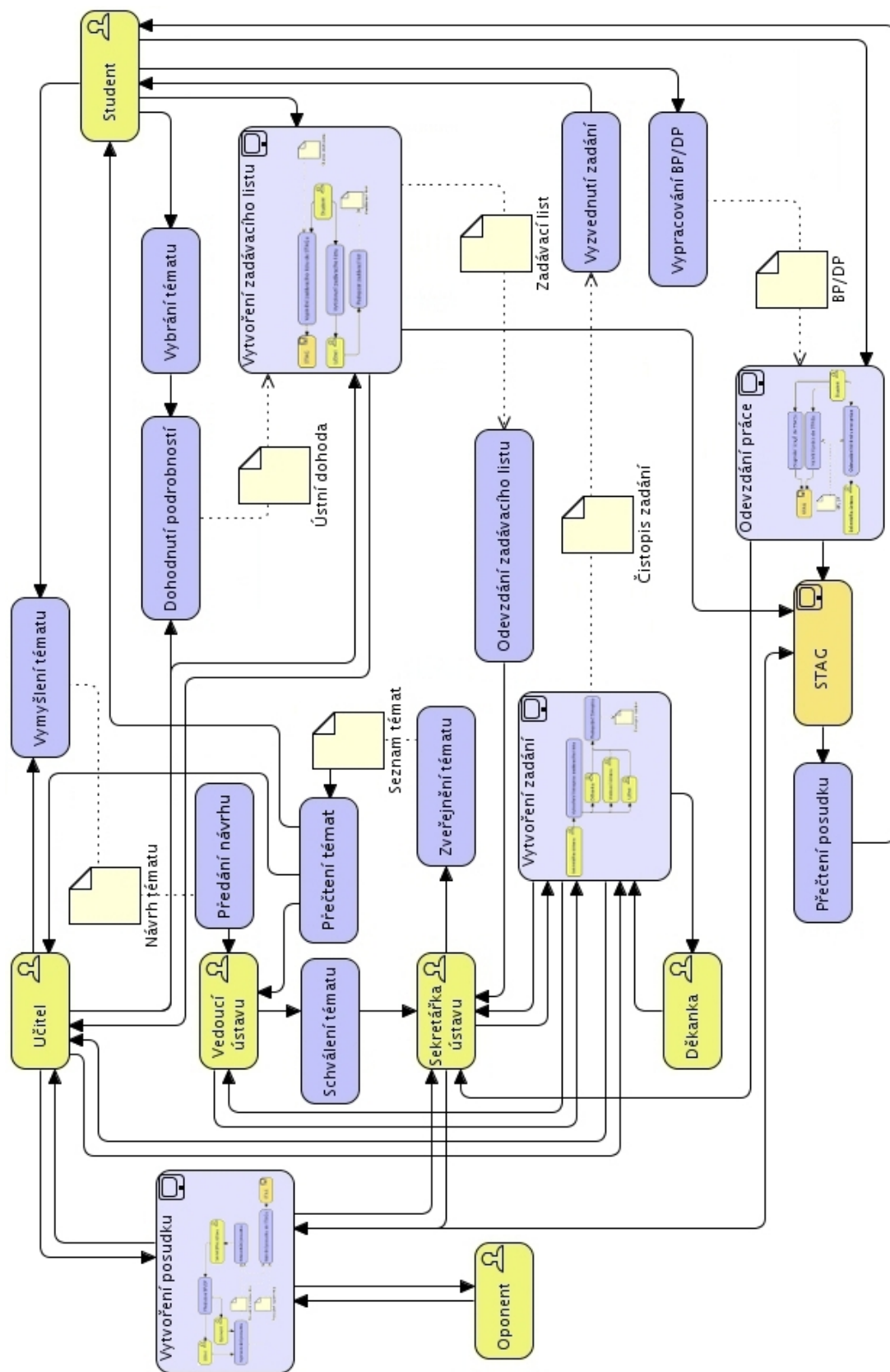
Obrázek 3: Odevzdání práce

Zdroj: vlastní



Obrázek 4: Vytvoření posudku

Zdroj: vlastní



Obrázek 5: Diagram datových toků současného stavu na FES

Zdroj: vlastní

## 2 Definice parametrů systému

Cílem této bakalářské práce je vytvořit systém pro evidenci bakalářských a diplomových prací. Nově vytvořená aplikace ponese stejný název (Systém pro evidenci bakalářských a diplomových prací) se zkratkou SEP.

Systém je určen především pro učitele a studenty Ústavu systémového inženýrství a informatiky Fakulty ekonomicko-správní Univerzity Pardubice. Učitelé budou mít přehled jednak o svých tématech a jednak o tématech vypsáných svými kolegy. Zároveň již nebude možné, aby se některý student zapsal na více témat najednou. Další přínos spočívá ve snížení papírování a urychlení práce s vypisováním témat.

Uživatelé se budou dělit do několika skupin podle role, kterou zastávají v systému. Jedná se o učitele, garanta (vedoucího ústavu/katedry), studenty a administrátora. Zvláštní rolí je nepřihlášený uživatel, který si bude moci pouze prohlížet témata, vyhledávat a tisknout.

### **Administrátor**

Administrátor se stará o správnou funkčnost systému. Jedná se o osobu s nejvyššími přístupovými právy – má veškerá práva ostatních uživatelů.

### **Vedoucí ústavu**

Jedná se o pedagoga s vyšším oprávněním. To znamená, že má stejné pravomoci jako pedagog, ale navíc schvaluje navržená témata. V případě, že některé téma neschválí, napíše k zamítnutí také důvod.

### **Pedagog**

Pedagog je uživatel, který bude plnit databázi novými záznamy. Již vypsaná témata může do termínu zveřejnění změnit nebo smazat. Jeho další činností je potvrzování rezervace studentů a tím dojde k obsazení práce. Jak rezervaci, tak obsazení práce může později zrušit. Dále bude moci prohlížet témata nová, čekající na schválení, schválená i neschválená. Ve schválených tématech může vyhledávat podle jména učitele nebo v názvu práce. V obhájených pracích může navíc vyhledávat i podle jména studenta. Má také možnost vytváření tiskových sestav.

### **Student**

Student může prohlížet pouze schválená témata. Z těchto témat si jedno vybere a rezervuje, tím dojde k zablokování tohoto tématu pro ostatní studenty. Zároveň si tento student už nemůže

rezervovat žádné další téma. Má možnost kontaktovat vedoucího práce pomocí e-mailu, který je u práce uveden. Může v tématech vyhledávat a vytvářet tiskové sestavy.

## **Nepřihlášený**

Uživatel, který se nepřihlásí, má možnost pouze prohlížet schválená témata po datu zpřístupnění. Může v nich vyhledávat a vytvářet tiskové sestavy.

## **2.1 Nejdůležitější funkce**

Systém pro evidenci bakalářských a diplomových prací má obsahovat tři fáze životního cyklu témat. První etapou je vypisování nových témat učitelem. Druhá etapa zahrnuje schvalovací proces a ve třetí etapě jsou schválená témata zpřístupněna studentům a nepřihlášeným uživatelům. Systém má podporovat vytváření tiskových sestav a vyhledávání podle zadaných kritérií.

### **2.1.1 První etapa – Návrh témat**

Učitel může vypisovat nová témata, u kterých musí zadat název v češtině a angličtině, popis a typ práce – zda se jedná o bakalářskou nebo diplomovou práci a určit, pro který rok je práce vypsána.

### **2.1.2 Druhá etapa – Schvalování témat**

Učiteli s vyšším oprávněním (vedoucí ústavu) jsou zobrazena již vypsána témata – zvláště bakalářské a diplomové práce. U každého může vybrat, zda ho schválí či neschválí. Pokud téma neschválí, má možnost připojit k dané práci důvod zamítnutí. Učitel následně vidí, která jeho témata jsou schválená a která nikoliv. Zamítnutá témata může upravit a znovu poslat ke schválení.

### **2.1.3 Třetí etapa – Publikování témat**

Schválená témata jsou průběžně zpřístupňována studentům a také nepřihlášeným uživatelům. Nepřihlášení mají oprávnění pouze témata prohlížet. Studenti mohou jednak témata prohlížet (zvláště bakalářské a zvláště diplomové práce) a jednak si mohou vybrané téma rezervovat a kontaktovat učitele. Každý student může mít rezervovanou pouze jednu práci. Učitel studentovi danou práci přidělí, ale toto přidělení může zrušit a tím se práce znovu objeví mezi volnými.

## **2.2 Ostatní parametry**

Systém pro evidenci bakalářských a diplomových prací bude napojený na dva další systémy. Jedním z nich je CAS, díky čemuž bude v SEPu fungovat centrální autentizace uživatelů. Druhým

systémem jsou webové služby IS/STAG (WS STAG), které budou sloužit pouze jako zdroj informací o zadavatelích prací a zapsaných studentech.

Vyvíjený systém není náročný na hardware, jelikož celá aplikace bude umístěna na školní server a přistupovat k ní bude možné přes webové rozhraní. Nutný je především přístup uživatelů k síti Internet. Co se týče softwarových požadavků, musí mít uživatel nainstalovaný webový prohlížeč (otestováno v nejběžnějších prohlížečích - Internet Explorer, Mozilla Firefox, Google Chrome, Opera). Pro správnou funkčnost je nutné mít v prohlížeči povolen JavaScript. Při vypnutém JavaScriptu je omezena funkčnost použitých komponent z frameworku PRADO (události OnClick, stránkování, validace, vyskakovací okna...), a proto se objeví upozornění „Nemáte povolený JavaScript“.

Každý uživatel musí být před vstupem do systému ověřen autentizací pomocí svého NetID. NetID pro pedagogy má tvar jmpr1234, kde jm jsou první dvě písmena křestního jména a pr jsou první dvě písmena příjmení. NetID pro studenty má tvar st123456. Přihlašovací údaje jsou odesílány na CAS, kde je uživatel ověřen autentizací.

Primárně je Systém pro evidenci bakalářských a diplomových prací určen uživatelům z Ústavu systémového inženýrství a informatiky Fakulty ekonomicko-správní Univerzity Pardubice. Možným rozšířením SEPu je jeho upravení pro další ústavy nebo i pro jinou fakultu. Bylo by ale nutné oddělit tyto části tak, aby se např. student z FES nemohl přihlásit na téma určené studentům jiné fakulty.

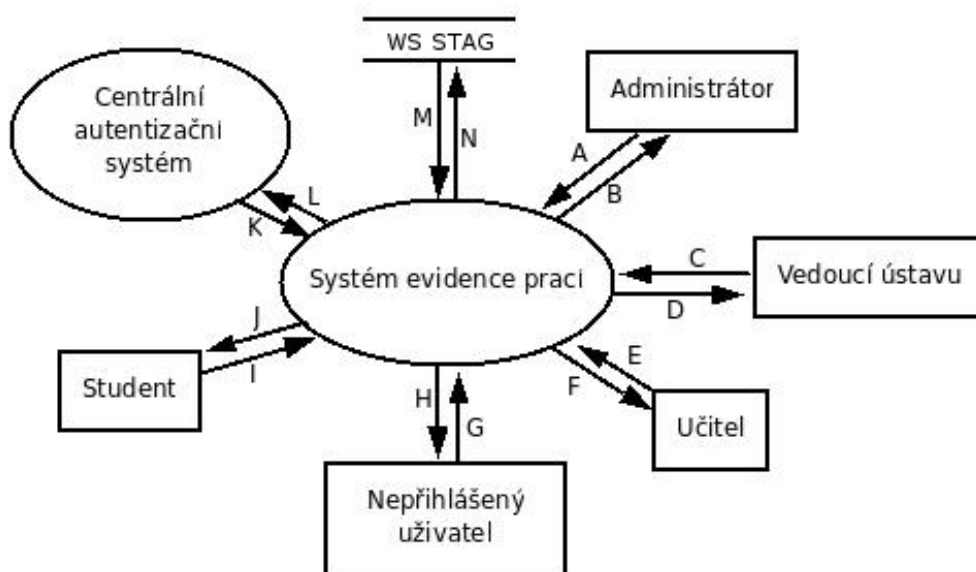
## 3 Návrh systému

### 3.1 Kontextový diagram

Tento model je výchozím modelem při modelování informačního systému. Řadí se mezi tzv. modely vnějšího chování systému. Charakteristika kontextového diagramu [1]:

- Reprezentuje hranici mezi vyvíjeným systémem a vnějším světem.
- Znázorňuje vyvíjený systém jako jediný proces, propojený datovými toky s okolím.
- Znázorňuje, kdo se systémem komunikuje, tj. vnější entity. Vnějšími entitami přitom mohou být lidé, organizace, jiné informační systémy apod.
- Zobrazuje vstupní a výstupní datové toky, tj.:
  - data vstupující do systému z okolí, která mají být systémem zpracována;
  - zpracovaná data vystupující ze systému do okolí.

Obrázek 6 zobrazuje SEP jako jeden proces a jeho vztah k okolí (vstupy a výstupy).



Obrázek 6: Kontextový diagram SEP

*Zdroj: vlastní*

#### Slovní popis významu jednotlivých externích entit

S podrobným popisem jednotlivých uživatelských rolí seznamuje kapitola 2.

- **Administrátor:** spravuje systém, má neomezený přístup
- **Vedoucí ústavu:** zobrazuje a modifikuje data v interní databázi
- **Učitel:** plní databázi daty, zobrazuje a modifikuje je
- **Nepřihlášený uživatel:** pouze zobrazuje data
- **Student:** zobrazuje a modifikuje data v interní databázi
- **CAS:** centrální autentizační systém, s jehož pomocí jsou uživatelé SEP autentizováni
- **WS STAG:** externí databáze s údaji o učitelích a studentech

### **Slovní popis významu a obsahu znázorněných datových toků**

- **A:** autentizace, správa systému, požadavek na smazání práce, prohlížení témat, vyhledávání, tisk
- **B:** povolení vstupu do systému, potvrzení změny, potvrzení smazání, výpis výsledků vyhledávání, vygenerování tiskové sestavy
- **C:** autentizace, schvalování témat, vrácení témat k upravení, vložení tématu, upravení tématu, smazání tématu, odeslání tématu ke schválení, přiřazení studenta k tématu, prohlížení témat, vyhledávání, tisk
- **D:** povolení vstupu do systému, potvrzení schválení tématu, potvrzení vrácení tématu k upravení, editace tématu, potvrzení smazání tématu, potvrzení odeslání témat ke schválení, editace obsazenosti tématu, výpis témat, výpis výsledků vyhledávání, vygenerování tiskové sestavy
- **E:** autentizace, vložení tématu, upravení tématu, smazání tématu, odeslání témat ke schválení, přiřazení studenta k tématu, prohlížení témat, vyhledávání, tisk
- **F:** povolení vstupu do systému, editace tématu, potvrzení smazání tématu, potvrzení odeslání témat ke schválení, editace obsazenosti tématu, výpis témat, výpis výsledků vyhledávání, vygenerování tiskové sestavy
- **G:** prohlížení témat, vyhledávání, tisk
- **H:** výpis témat, výpis výsledků vyhledávání, vygenerování tiskové sestavy
- **I:** autentizace, prohlížení témat, rezervování tématu, vyhledávání, tisk

- **J:** povolení vstupu do systému, výpis témat, editace obsazenosti tématu, výpis výsledků vyhledávání, vygenerování tiskové sestavy
- **K:** výsledek autentizace – povolení nebo zamítnutí přístupu
- **L:** přihlašovací údaje přistupujícího uživatele
- **M:** vypsání požadovaných údajů z databáze
- **N:** požadavek na vypsání údajů z databáze

### 3.2 Diagram datových toků

Účelem diagramu datových toků je [1]:

- zobrazit dynamický pohled na vyvíjený systém;
- znázornit místa, kde dochází k transformaci dat do jiné formy – tj. znázornit procesy;
- popsat, z jakých procesů a jejich návazností se systém skládá;
- znázornit a popsat datové toky sloužící jako vstupy a jako výstupy jednotlivých procesů.

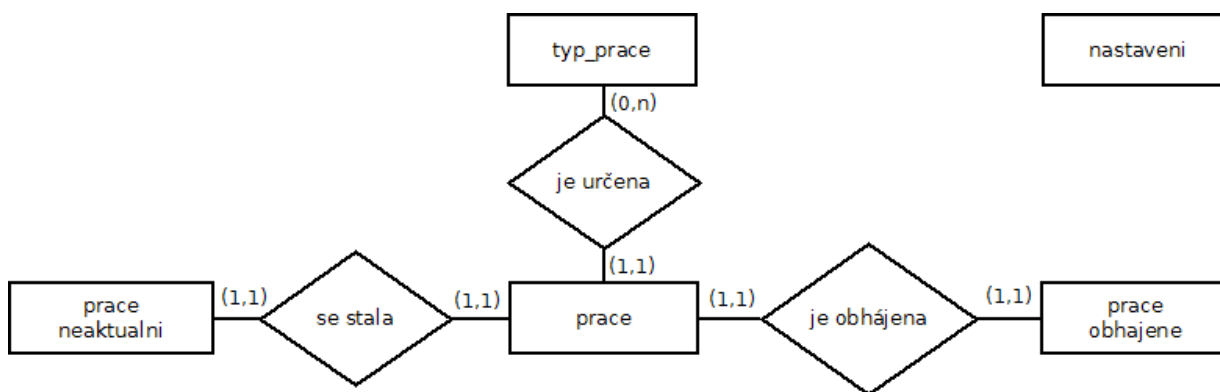
Výsledkem tvorby digramu datových toků je hierarchie dílčích modelů vyvíjeného systému, které znázorňují vyvíjený IS na různých rozlišovacích úrovních [1]. Na obrázku 14 v příloze A je systém znázorněn na nejvyšší rozlišovací úrovni, kde je zakresleno pět základních procesů (tmavě modrou), pět terminátorů (sytě žlutou) a šest skladů (červenou). Dále do systému vstupují webové služby systému STAG (WS STAG), ty ovšem nejsou přímou součástí SEP a proto je tento terminál zobrazen pouze světle žlutou barvou. Jejich účelem je poskytovat údaje o studentech a pedagogích. Posledním procesem je centrální autentizační systém (CAS) zobrazený světle modře. Do systému vstupuje také zvenčí a bude se starat o autentizaci a autorizaci. V testovací verzi bude CAS zastoupen tabulkou uzivatele, kterou bude možno před uvedením do provozu smazat. Datové toky jsou kvůli přehlednosti zakresleny obousměrně. Jediné jednosměrné toky propojují „Editace prací“ a „Odesílání e-mailů“ a dále „Odesílání e-mailů“ a „Student“.

Diagramy datových toků na nižších rozlišovacích úrovních znázorňují procesy „Zobrazení prací“ (viz obrázek 15 v příloze B), „Editace prací“ (viz obrázek 16 v příloze C) a „Editace nastavení“ (viz obrázek ?? v příloze D).



### 3.3 Entitně relační diagram

Pro snadnější pochopení vztahů mezi jednotlivými entitami databáze jsem vytvořila entitně relační diagram (viz obrázek 7). Atributy jednotlivých entit systému jsou více přiblíženy tabulkami 1 – 6 v příloze E. Z entitně relačního diagramu je patrné, že hlavní tabulkou databáze systému je **prace**, ve které jsou evidována jednotlivá témata. Každé téma má přidělené `id_prace`, které se automaticky vygeneruje u nového tématu. Atribut `id_typu` určuje, zda se jedná o bakalářskou nebo diplomovou práci. U každé práce je evidováno `NetID` učitele, který práci vypsál. Jelikož bude tento učitel v době vkládání nového tématu autentizován, doplní se do tabulky práce automaticky. Při vypisování může pedagog přiřadit práci studentovi a tím se uloží `student_netid` a `osobni_c` (`NetID` a osobní číslo studenta). To stejné je doplněno i v případě, že si téma rezervuje student sám. `Nazevcs` obsahuje název tématu v češtině a `nazeven` v angličtině. Do atributu `popis` je možné vložit zásady pro vypracování nebo popis práce. Stav může být buď *Volná*, *Rezervovaná*, nebo *Obsazená*. Pokud je zadáno *Volná*, nemůže být k práci přidělen student. `Datum_odevzdani` je předpokládaný rok odevzdání práce. Pokud je práce nově vypsána, bude v atributu `druh` *Nová*, při odeslání ke schválení se automaticky změní na *Čekající*, při schválení se uloží jako *Schválená* a při neschválení *Neschválená*. Tento atribut není uživatelům přístupný, mění se automaticky. Je jím pouze rozlišeno, kterým uživatelům se zobrazují jaká témata. Atribut `status` může nabývat hodnot *Aktuální*, *Neaktuální* nebo *Obhájená*.



Obrázek 7: Entitně relační diagram databázového systému

Zdroj: vlastní

Entity **prace\_neaktualni** a **prace\_obhajene** mají téměř stejné atributy jako předchozí entita. Rozdíl spočívá především v ukládání zadavatele a studenta. Předpokládá se totiž, že student, který obhájí práci, ukončí studium a v takovém případě by se již nezobrazily údaje hledané podle `NetID`. Stejně tak pedagog může ze školy odejít. Proto jsou v těchto entitách uloženy údaje přímo, nikoli přes `NetID`.

Entita **uzivatele** je pouze dočasná. Jakmile bude SEP napojen na centrální autentizační systém,

bude možné ji odstranit. V testovacím provozu kontroluje údaje při přihlašování a určuje roli pro přihlášeného uživatele. V entitě **typ\_prace** jsou pouze dva záznamy. Jedním z nich je *bakalářská práce* a druhým záznamem je *diplomová práce*. V budoucnu by bylo možné tento systém rozšířit o zadávání dalších druhů prací. Entitně relační diagram ukazuje, že typ práce nemusí určovat žádnou práci, ale může určovat n prací. Zatímco práce musí být vždy určena právě jedním typem práce. Poslení entitou je **nastaveni**, kde se budou ukládat údaje o datu zveřejnění a NetID vedoucího ústavu a administrátora systému.

## 4 Implementace SEP

### 4.1 Výběr technologie

Po vytvoření modelu a definování vztahů jednotlivých entit přichází na řadu samotná implementace systému. Již v minulosti jsem se setkala s PRADO (PHP Rapid Application Development Object-oriented – rychlé objektově orientované programování aplikací v PHP) a měla jsem možnost vyzkoušet si, co všechno dokáže. Líbí se mi řešení přístupu k databázi a také přehlednost kódu, kde je oddělena prezentace od logiky. Výběr technologie jsem tedy měla usnadněný. Při tvorbě kódu jsem nejčastěji hledala informace na stránkách *PRADO PHP Framework* [10].

PRADO je rozhraní pro vývoj webových aplikací v PHP 5. Je založené na komponentním a událostmi řízeném přístupu. Hlavním cílem PRADO je umožnit maximální znovupoužitelnost webového programování a to nejen znovupoužitím svých vlastních kódů, ale také snadné použití jinými programátory [10].

Následuje krátký souhrn nejdůležitějších vlastností PRADO [10]:

- Znovupoužitelnost – kód, který se řídí komponentním protokolem PRADO, je dobře znovupoužitelný. To přináší prospěch vývojářským týmům, jelikož mohou opětovně použít dřívější práci a snadno ji začlenit do jiných projektů.
- Programování řízené událostmi – aktivity uživatele, jako je kliknutí na odesílací tlačítko, jsou zachyceny jako serverové události, takže se vývojáři mohou více zaměřit na interakce uživatele.
- Oddělení prezentace a logiky – prezentace a logika jsou uloženy odděleně. Aplikace vytvořené pomocí frameworku PRADO podporují definici vlastních vzhledů.
- Výkonné webové prvky – PRADO přichází se sadou výkonných komponent pro práci s webovým uživatelským rozhraním. Vysoce interaktivní webové stránky mohou být vytvořeny na pár řádcích kódu. Například použitím komponenty *datagrid* lze v HTML vytvořit výpis prezentující tabulku v relační databázi, která umožňuje stránkování, třídění, úpravy jako odstranění dat apod.
- Výrazná podpora databáze – od verze 3.1. je framework vybaven kompletní databázovou podporou, jejíž použití je intuitivní. Podle složitosti celé aplikace si lze vybrat buď jednoduchý přístup k databázi prostřednictvím PDO (PHP Data Object) nebo hodně známý aktivní záznam (Active Record) nebo kompletní objektově mapované schéma SqlMap.
- Adaptace již existující práce – PRADO je všeobecný framework se zaměřením na prezentační vrstvu. Umožňuje vývojářům používání nejnovějších knihoven tříd a nástrojů.

- Výborné hlášení chyb a výjimek
- Přizpůsobitelná chybová hlášení
- Další vlastnosti – podpora AJAXu (bez nutnosti psaní kódu v JavaScriptu použitím tzv. aktivních komponent), dodržování předpisu pro XHTML (zdrojový XHTML kód webových stránek generovaných pomocí PRADO vyhovuje předpisům W3C), Podpora I18N a L10N (kompletní podpora pro tvorbu vícejazyčných aplikací), rozšiřitelná autorizace a autentizace, podpora cookies atd.

## 4.2 Nastavení PRADO

Minimální systémové požadavky PRADO jsou nainstalovaný webový server s podporou PHP 5.1.0 nebo vyšší. PRADO bylo testováno s webovým serverem Apache pod Windows i Linux, ale pravděpodobně poběží i s jinými webovými servery, které podporují PHP 5. Po instalaci PRADO, která je popsána na stránkách *PRADO PHP Framework* [10], je vhodné spustit PRADO Requirement Checker, který je k dispozici na adrese:

*http://web-server-address/prado/requirements/index.php*. Tento skript zobrazí, jaká máme nainstalovaná rozšíření a zda konfigurace našeho serveru vyhovuje požadovanému nastavení.

Dalším důležitým krokem je umožnit vznikající aplikaci přístup k PRADO. Toho docílím napsáním tohoto kódu do **index.php**:

```

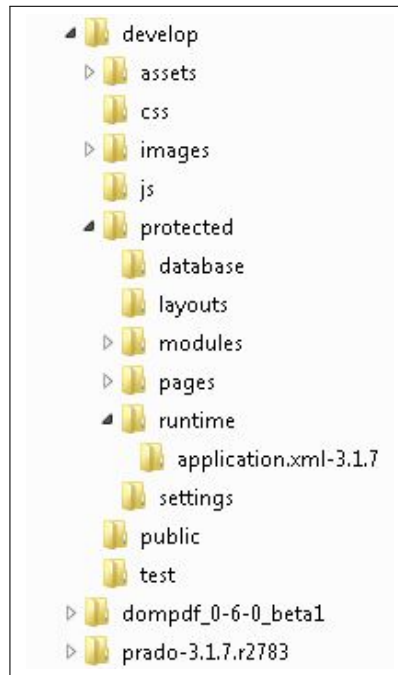
1 <?php
2 $frameworkPath='../prado-3.1.7.r2783/framework/prado.php';
3 require_once($frameworkPath);
4 $application=new TApplication;
5 $application->run();
6 ?>

```

Zdrojový kód 1: Základní varianta souboru index.php

PRADO aplikace má předepsanou adresářovou strukturu (viz obrázek 8), která vypadá následovně:

- assets – adresář, kde jsou uloženy publikované privátní soubory. Tento adresář musí být zapisovatelný pro procesy webového serveru.
- protected – základní cesta aplikace, kde jsou uložena data a privátní soubory skriptů aplikace. Tento adresář by měl být nastavený jako nepřístupný konečnému uživateli.
- runtime – provozní úložiště aplikace, kde jsou uloženy provozní informace aplikace, jako je stav aplikace, dočasná data atd. Tento adresář musí být zapisovatelný pro procesy webového serveru.



Obrázek 8: Adresářová struktura SEP

*Zdroj: vlastní*

- pages – základní úložiště pro všechny stránky PRADO.

Volitelný soubor **application.xml** obsahuje nastavení a další konfiguraci aplikace. Je možné zde například nastavit moduly, služby, připojení k databázi atd. Následující kód ukazuje nastavení jména domovské stránky (Home) a hlavní rozvržení všech stránek (MainLayout):

```
<services>
  <service id="page" class="TPageService" DefaultPage="Home">
    <pages MasterClass="Application.layouts.MainLayout" />
  </service>
</services>
```

Zdrojový kód 2: Nastavení domovské stránky a implicitního rozvržení stránek v souboru application.xml

SEP je určen jak pro přihlášeného, tak pro nepřihlášeného uživatele. Na jakou stránku bude mít uživatel přístup, nastavuji v souboru **config.xml** v elementu authorization (viz kód 3). V elementu allow povolují uživatelům (atribut users) určitých uživatelských rolí (atribut roles) přístup na stránky (atribut pages), jejichž názvy jsou oddělené čárkou. Atribut users může nabývat různých hodnot:

- konkrétní jméno uživatele (NetID),
- přihlášený uživatel (@),
- jakýkoli uživatel včetně anonymního (\*)

- a výlučně anonymní uživatel (?).

Atribute roles nabývá hodnot podle požadavků na systém, v případě SEPu jsou možné role čtyři: Admin, Vedoucí, Pedagog a Student. Nepřihlášený uživatel je označen jako Guest. Například na stránku RezervacePrace může vstoupit pouze uživatel přihlášený jako student nebo administrátor, naopak pro DetailPrace neplatí žádné omezení.

Role administrátora je v systému zavedena, ale není nutné ji používat. Administrátor nemá své odlišné administrátorské rozhraní. Jeho práva spočívají ve sloučení práv všech ostatních uživatelských rolí. Z tohoto důvodu nebudu dále tuto roli podrobněji popisovat.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <configuration>
3    <authorization>
4      <allow pages="NovaPrace,EditPrace,KopiePrace,MojePrace,OdeslaniPrace,SchvalPrace,
5        SchvalDetailPrace"
6        users="@@"
7        roles="Admin,Vedouci" />
8      <allow pages="NovaPrace,EditPrace,KopiePrace,MojePrace,OdeslaniPrace"
9        users="@@"
10       roles="Admin,Pedagog" />
11     <allow pages="RezervacePrace"
12       users="@@"
13       roles="Admin,Student" />
14     <allow pages=""
15       users="@@" />
16     <allow pages="ListBPrace,ListDPrace,TabListBPrace,TabListDPrace,DetailPrace" />
17     <deny users="" /></authorization></configuration>

```

Zdrojový kód 3: Nastavení přístupu na stránky pomocí config.xml

Po tomto nastavení se nemůže stát, že by některému uživateli byla přístupná stránka pro jinou roli. Například student, který se chce dostat na seznam prací ke schválení, může zkusit napsat do adresy SchvalPrace. Aplikace mu ji ale nezobrazí, místo toho ho přesměruje na přihlašovací formulář. Zde se očekává, že se přihlásí vedoucí ústavu nebo administrátor, kteří mají oprávnění tuto stránku vidět.

### 4.3 Tvorba stránek

Každá stránka v PRADO se skládá ze souboru, jehož název končí příponou *.page* a souboru s třídou stejného názvu, který končí příponou *.php*. V praxi to znamená, že stránka s detailem práce je popsána v souborech *DetailPrace.page* a *DetailPrace.php*, který zároveň obsahuje základní třídu *DetailPrace*.

Každá validní HTML stránka má definovanou strukturu. V případě SEP se do této struktury dá zahrnout i menu, patička apod. Abych se vyhnula neustálému opakování těchto částí, které jsou pro všechny stránky stejné, vytvořila jsem šablonu *MainLayout*. Jakým způsobem se toto

hlavní rozložení nastavuje, jsem již uvedla v kapitole 4.2. Skládá se ze souborů **MainLayout.tpl** a **MainLayout.php**. Soubor **MainLayout.tpl** obsahuje klasické HTML elementy a komponenty PRADO, jimiž jsou vkládány jednotlivé části stránky (viz kód 4). První komponentou je obdoba hlavičky HTML, komponenta THead s obsahem v souboru **Head.tpl**, vložená na řádku 4, která obsahuje například metadata (popis, klíčová slova, kategorie, autor, webmaster), dále odkazy na soubor se stylem<sup>1</sup>, s ikonou webu, informace pro prohlížeče, roboty. . .

Celý obsah elementu <body> je uzavřen komponentou TForm, která zobrazuje HTML formulář a je nepostradatelná pro funkčnost celé aplikace. Na řádku 10 vkládám do každé stránky hlavní menu, to stejné je vloženo i na řádku 19. Obsahuje odkazy domů, na bakalářské práce, diplomové práce, kontakt, přihlašovací formulář a zároveň je v něm informace o přihlášeném uživateli. Po přihlášení pedagoga zobrazuje navíc odkazy na jeho práce, formulář pro přidání nové práce, případně schvalování prací u vedoucího ústavu.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/
  xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
3   <com:THead>
4     <%include Application.layouts.Head %>
5   </com:THead>
6   <body>
7     <com:TForm>
8       <div id="container">
9         <div id="horni-menu">
10          <%include Application.layouts.Menu %>
11          <br style="clear: both;" />
12        </div>
13        <div id="content">
14          <com:TContentPlaceHolder ID="Obsah" />
15        </div>
16        <br style="clear: both;" />
17        <div id="dolni-menu">
18          <%include Application.layouts.DolniMenu %>
19          <%include Application.layouts.Menu %>
20        </div>
21        <br style="clear: both;" />
22        <div id="footer">
23          <%include Application.layouts.Footer %>
24        </div>
25      </com:TForm>
26    </body>
27  </html>
```

Zdrojový kód 4: Implicitní rozvržení stránek v souboru MainLayout.tpl

Důležitou komponentou v tomto souboru je TContentPlaceHolder na řádku 14, která re-

---

<sup>1</sup>PRADO nabízí pokročilé možnosti stylování (přepínání stylů apod.), ale v této práci se zaměřuji více na funkčnost systému, proto jsem použila jen jednoduché kaskádové styly CSS pro všechny obsažené stránky.

zervuje místo pro obsah konkrétní stránky, jenž sem bude vložen pomocí svého identifikátoru `ID="Obsah"`. Vkládaná stránka musí být potom uzavřená v komponentě `<com:TContent ID="Obsah">`.

Posledním vkládaným prvkem stránky je dolní menu, vložené příkazem na řádku 18. Menu obsahuje odkazy na RSS čtečky pro bakalářské práce a pro diplomové práce (viz kapitola 4.10.3). Do souboru `MainLayout.php` se vkládají veškeré funkce potřebné pro `MainLayout.tpl`, ale jelikož zde žádné nepoužívám, obsahuje `MainLayout.php` jen základní třídu (viz kód 5).

```
1 <?php
2     class MainLayout extends TTemplateControl {}
3 ?>
```

Zdrojový kód 5: Základní varianta souboru `MainLayout.php`

Do připravené šablony již stačí vložit samotnou stránku, například kontakt. V souboru `Contact.page` (viz kód 6) je vložen titulek stránky a komponenta `TContent`. Vše, co tato komponenta uzavírá, bude díky stejnému ID `Obsah` vloženo do hlavní šablony `MainLayout.tpl`. `Contact.php` zobrazuje kód 7. Rozdíl oproti `MainLayout.php` je v použití rozšiřující třídy `TPage` namísto `TTemplateControl`. `TPage` vyžaduje příponu `.page` u sdruženého souboru. `TTemplateControl` automaticky načítá a zpracovává šablony ze souboru se stejným názvem avšak s příponou `.tpl`.

```
1 <%@ Title="SEP &ndash; Kontakt" %>
2 <com:TContent ID="Obsah">
3     <h1>Kontakt</h1>
4     <ul>
5         <li>Lada Pěničková, SEP webadmin:
6             <a href="lada.penickova@upce.cz">lada.penickova@upce.cz</a>, +420 721 742 784</li>
7         <li>Informační centrum, správa UPCE:
8             <a href="ic@upce.cz">ic@upce.cz</a>, +420 466 036 633</li>
9     </ul>
10 </com:TContent>
```

Zdrojový kód 6: `Contact.page`

```
1 <?php
2     class Contact extends TPage {}
3 ?>
```

Zdrojový kód 7: `Contact.php`

## 4.4 Práce s databází

Jako většina informačních systémů používá i SEP k uložení dat o studentských pracích databázový systém. Jeho základem je systém řízení báze dat (SŘBD) MySQL<sup>2</sup>. Aby se aplikace vůbec mohla spojit s databází, musím upravit nastavení v souboru `application.xml` vložím modulů,

<sup>2</sup>V mém případě se jedná o testování na verzi MySQL 5.1.36.



jak ukazuje kód 8. V elementu database je připojovací řetězec k databázi se jménem databáze (dbname) a adresou databázového serveru (host). Dále uživatelské jméno a heslo k databázi a znaková sada, která určuje, v jakém kódování jsou data odesílána frameworkem do databáze<sup>3</sup>.

```
<modules>
  <module id="db" class="System.Data.TDataSourceConfig">
    <database ConnectionString="mysql:dbname=SEP;host=localhost" Username="root"
      Password="heslo" Charset="utf8" />
  </module>
  <module class="System.Data.ActiveRecord.TActiveRecordConfig" ConnectionID="db" />
</modules>
```

Zdrojový kód 8: Nastavení databáze v souboru application.xml

Po vytvoření databáze podle ER diagramu (viz Tabulka ??) následuje tzv. namapování všech používaných tabulek. Díky tomu bude každý záznam přístupný jako objekt Active Record a můžeme k němu přistupovat pomocí dotazů, ukládat a mazat bez nutnosti psaní SQL příkazů. Namapování tabulky prace v souboru **PraceRecord.php** zobrazuje kód 9. Pro každý atribut v tabulce prace má třída PraceRecord odpovídající datový prvek (řádky 5–18). Konstanta TABLE označuje název tabulky odpovídající třídě PraceRecord. Statická funkce finder() na řádku 20 dovoluje vykonávat dotazy (číst a zapisovat data pomocí objektů PraceRecord). Polem \$RELATIONS (řádek 24–27) definuji vztahy vycházející z ER diagramu, tedy používání cizích klíčů. Atribut typ\_prace je primárním klíčem tabulky TypPrace, která je s tabulkou prace ve vztahu 1:N (každá práce je buď bakalářská, nebo diplomová). Proto typ vztahu na řádku 25 je self::BELONGS\_TO. Naproti tomu v souboru **TypPraceRecord.php** bude uveden vztah 'prace' => array(self::HAS\_ONE, 'PraceRecord'), protože jeden typ práce může být uveden u více prací. Stejný typ vztahu (1:N) platí i pro datum odevzdání, který je nadefinován obdobným způsobem na řádku 26.

```
1 <?php
2 class PraceRecord extends TActiveRecord {
3     const TABLE='prace';
4
5     public $id_prace;
6     public $status;
7     public $nazevcs;
8     public $nazeven;
9     public $popis;
10    public $stav;
11    public $druh;
12    public $datum_vlozeni;
13    public $poznamka;
14    public $typ_prace;
15    public $zadavatel_netid;
16    public $student_netid;
17    public $osobni_c;
```

<sup>3</sup>Pro čitelnost dat v databázi i systému je potřeba, aby se kódování dat v databázi i znaková sada shodovaly.

```

18 public $datum_odevzdani;
19
20 public static function finder($className=__CLASS__) {
21     return parent::finder($className);
22 }
23
24 public static $RELATIONS=array (
25     'typ_prace' => array(self::BELONGS_TO, 'TypPraceRecord'),
26     'datum_odevzdani' => array(self::BELONGS_TO, 'RokRecord'),
27 );
28 }
29 ?>

```

Zdrojový kód 9: Namapování tabulky prace pomocí PraceRecord.php

V databázi v tabulce prace zbývá ošetřit, aby student nemohl mít rezervováno nebo obsazeno více prací. To je zabezpečeno pomocí SQL příkazu `CREATE UNIQUE INDEX [student] ON prace ({student_netid, osobni_c}) ;`. Nyní do databáze není možné k práci přiřadit kombinaci NetID a osobního čísla, která tam již existuje. Každý student má nejméně jedno osobní číslo, může jich mít i víc podle toho, kolik oborů studuje. Co obor, to osobní číslo. Student zapsaný do více oborů, může mít zapsáno také více prací, vždy však pro každý obor nejvýše jednu. Jestliže se student pokusí zarezervovat druhou práci pod stejným osobním číslem, systém zobrazí upozornění jako na obrázku 9.

SEP bude pravděpodobně používat spousta studentů i akademických pracovníků. Po přihlášení se každému zobrazí úvodní stránka, která čte informace z databáze. Stejně tak se stránky Bakalářské práce a Diplomové práce připojují k databázi. Kdyby SEP pokaždé načítal data o úplně všech pracích v databázi, minimálně by to aplikaci ztlačilo. Proto jsem se snažila při každém načítání dat z databáze dotaz co nejvíce zúžit. K tomu jsem použila pohledy, které jsem si vytvořila pro každé takové načítání. Mezi nimi jsou pohledy pro:

- aktuální práce,
- aktuální práce nové,
- aktuální práce čekající,
- aktuální práce neschválené,
- aktuální práce schválené,
- aktuální práce schválené volné,
- aktuální práce schválené volné bakalářské,
- aktuální práce schválené volné diplomové,

- aktuální práce schválené obsazené bakalářské,
- aktuální práce schválené obsazené diplomové,
- neaktuální práce,
- obhájené práce.

První návrh databáze počítal s tím, že SEP bude mít svoji vlastní tabulku uživatelů. U každého záznamu by se tak evidovalo NetID, jméno a příjmení, tituly, role a další atributy. Tím by ale nebyla zaručena aktuálnost dat. Poohlédla jsem se proto po již existujících úložištích dat o studentech a pedagozích, ze kterých by SEP mohl údaje čerpat. První variantou je LDAP, odkud je možné získat NetID, jméno, příjmení a e-mail. Už ale neviduje tituly, obor studenta nebo ústav pedagoga. Přístup k LDAP by byl navíc umožněn až hotové aplikaci a pro testování by bylo potřeba vytvořit testovací data.

Druhou možností čerpání údajů jsou webové služby IS/STAG. Zde jsou data přístupná bez omezení, stejné údaje umožňuje prohlížet i veřejná část serveru portal.upce.cz. WS STAG vznikly z potřeby čerpat informace z databáze IS/STAG a používat je například v takových systémech jako je SEP nebo jiné stránky kateder, to je jejich primární účel. Další výhodou je množství informací, které lze získat. Např. u pedagoga je to:

- jméno,
- příjmení,
- titul před jménem,
- titul za jménem a
- e-mailovou adresu, pokud je zadaná

a u studenta:

- jméno,
- příjmení,
- zda studuje, má přerušené studium, nebo daný obor již nestuduje,
- NetID,
- studijní program,
- fakulta,

- forma studia,
- místo výuky,
- ročník,
- název a zkratka oboru a
- zda se jedná o obor bakalářský, navazující nebo doktorandský.

Na základě těchto předností jsem se rozhodla získávat data o uživateli prostřednictvím WS STAG. Komunikace probíhá zcela jednoduchým způsobem. Stačí znát URL s konkrétní službou, kam SEP pošle dotaz a WS STAG vrátí výsledek ve formátu XML. V SEP jsou využívány celkem čtyři služby (. . . nahrazují ws/services/rest):

- <http://stag-ws.upce.cz/. . . /users/getOsobniCislaByExternalLogin?login=st16423>, kde vstup je NetID studenta (např. st16423) a výstupem jsou všechna osobní čísla studenta,
- <http://stag-ws.upce.cz/. . . /student/getStudentInfo?osCislo=E05714>, vstup je osobní číslo studenta (např. E05714) a výstup kompletní údaje o studentovi,
- <http://stag-ws.upce.cz/. . . /users/getUcitIdnoByExternalLogin?externalLogin=mihu0334>, vstup je NetID pedagoga (např. mihu0334) a výstup identifikační číslo pedagoga v IS/STAG,
- <http://stag-ws.upce.cz/. . . /ucitel/getUcitelInfo?ucitIdno=260>, identifikační číslo pedagoga v IS/STAG (260) a výstup kompletní údaje o pedagogovi.

Získaná data jsou převedena na pole pomocí funkcí sestavených v souboru **wsSTAG.php** a následně zobrazena jako součást stránky SEP [6].

## 4.5 Role nepřihlášeného uživatele

Uživatel, který se nepřihlásí, má v systému nejnižší oprávnění. Smí pouze prohlížet schválené práce, vyhledávat v názvu práce nebo podle jména pedagoga, může vyplnit a odeslat kontaktní formulář, nebo se přihlásit.

Na úvodní stránce se uživateli zobrazí výpis schválených prací, které nejsou obsazené ani rezervované a studenti si z nich mohou vybírat. Výpis je včetně vedoucího a popisu práce, odděleně pro bakalářské a diplomové práce. Každý název práce v češtině je zároveň odkazem na detail tématu.

Výpisy na úvodní stránce jsou vytvořeny pomocí komponenty TRepeater (řádek 14–20), což se dá přeložit jako opakovač. Jak název napovídá, tato komponenta zobrazuje opakovaně svůj

obsah, který získává z nějakého zdroje. U úvodní stránky je tímto zdrojem renderer („zobrazovač“). Z kódu 10 je patrné, že těchto rendererů může být hned několik – jeden základní (řádek 15), druhý alternativní (řádek 16) a třetí (řádek 17) se zobrazuje v případě, že nebyla načtena žádná data (např. aktuálně není žádná volná bakalářská práce). Alternativní renderer umožňuje například střídání vzhledů sudých a lichých výsledků repeateru. V SEP slouží pro střídání barvy pozadí.

```

1 <com:TContent ID="Obsah">
2 <div id="listpracehome">
3 <h1>Práce</h1>
4 <div id='BPhome' >
5 <h2>Bakalářské práce</h2>
6 <p class="pagertop">
7 <com:TPager ControlToPaginate="RepeaterPraceBP" OnPageIndexChanged="
8 pageChangedPraceBP" Mode="NextPrev" PageButtonCount="5" ButtonType="LinkButton"
9 PrevPageText="&nbsp;&lsaquo;&nbsp;&nbsp;" FirstPageText="&nbsp;&laquo;&nbsp;&nbsp;"
10 NextPageText="" LastPageText="" />
11 <com:TPager ControlToPaginate="RepeaterPraceBP" OnPageIndexChanged="
12 pageChangedPraceBP" Mode="Numeric" PageButtonCount="5"
13 PrevPageText="" FirstPageText="" NextPageText="" LastPageText="" />
14 <com:TPager ControlToPaginate="RepeaterPraceBP" OnPageIndexChanged="
15 pageChangedPraceBP" Mode="NextPrev" PageButtonCount="5" ButtonType="LinkButton"
16 PrevPageText="" FirstPageText="" NextPageText="&nbsp;&rsaquo;&nbsp;&nbsp;" LastPageText="
17 &nbsp;&raquo;&nbsp;&nbsp;" />
18 </p>
19 <com:TRepeater ID="RepeaterPraceBP"
20 ItemRenderer="Application.pages.homeRenderers.HomePraceBPRenderer"
21 AlternatingItemRenderer="Application.pages.homeRenderers.AltHomePraceBPRenderer"
22 EmptyRenderer="Application.pages.homeRenderers.EmptyHomePraceBPRenderer"
23 AllowPaging="true"
24 AllowCustomPaging="true"
25 PageSize="5" />
26 (...)
27 </div>
28 </div>
29 </com:TContent>

```

Zdrojový kód 10: Komponenta TRepeater a TPager v souboru Home.page

Důležitým údajem u repeateru je povolení stránkování (vlastnosti AllowPaging a AllowCustomPaging v kódu na řádcích 18 a 19). Vlastnost PageSize určuje, kolik záznamů se na stránku zobrazí (= parametr Limit v souboru php). Dalším parametrem funkce pro stránkování je Offset, který určuje odstup prvního záznamu zobrazené stránky od celkově prvního záznamu. Např. Limit = 30 a Offset = 40 znamená, že se zobrazí 30 záznamů od čtyřicátého. Při změně stránky se změní i Offset, ke kterému se po každém kliknutí na následující stránku přičte hodnota Limit. Limit se nemění.

Komponenta TPager definuje vzhled ovladačů stránkování jedním ze tří typů rozhraní:

- NextPrev – tlačítka na předchozí a následující stránku,

- Numeric – čísla stránek,
- DropDownList – roletový seznam s čísly stránek.

V kódu 10 je stránkování nastaveno na řádcích 7–12, přičemž NextPrev je použito dvakrát. Jednou pro šipky na první (FirstPageText) a předchozí (PrevPageText) stránku (řádek 7–8) a podruhé pro šipky na následující (NextPageText) a poslední (LastPageText) stránku (řádek 11–12). Mezi těmito komponentami TPager je vložena další v módu Numeric, která zobrazuje čísla stránek a omezuje jejich počet na pět (PageButtonCount).

Zdrojový kód v souboru **HomePraceBPRenderer.tpl** je zobrazen ve výpisu 11. Komponenty TLiteral a THyperLink zobrazují jednotlivé části výsledného rendereru, které jsou čtené z databáze. V souboru HomePraceBPRenderer.php jsou již pouze funkce getZnak(), getUrl() a zapisZadavatele().

```

1 <div class="prace-box">
2   <h3>
3     <com:TLiteral Text="<## $this->getZnak() %>" />
4     <com:TLiteral Text=" <## $this->Data->typ_prace %> &ndash; " Visible="False" />
5     <com:THyperLink Text="<## $this->Data->nazevcs %>" NavigateUrl="<## $this->getUrl() %>
      " />
6     <com:TLiteral Text="(<## $this->Data->nazeven %>)" />
7   </h3>
8   <p><b>Vedoucí práce:</b>
9     <com:TLiteral ID="Zadavatel" Text="<## $this->zapisZadavatele($this->Data->
      zadavatel_netid) %>" />
10  </p>
11  <p><b>Popis práce:</b></p>
12  <p>
13    <com:TLiteral Text="<## $this->Data->popis %>" />
14  </p>
15 </div>

```

Zdrojový kód 11: Obsah rendereru HomePraceBPRenderer.tpl

Stejným způsobem jako úvodní stránka jsou vytvořeny i výpisy bakalářských (ListBPrace) a diplomových prací (ListDPrace). U nich je ale možné přepnout do tabulkového zobrazení (TabListBPrace a TabListDPrace). Tento tabulkový přehled s řádky a sloupečky je vytvořen pomocí komponenty TDataGrid – co řádek, to téma práce. K jednotlivým řádkům se přistupuje pomocí jejich vlastnosti Item. Data načtená z databáze jsou uložena do datového zdroje (DataSource) komponenty TDataGrid (viz kód 12). Data se v datagridu zobrazí pomocí metody dataBind(). Jelikož nezobrazují všechny atributy z tabulky práce, nemohu použít automatické generování sloupců. Jednotlivé sloupce specifikuji ručně a určuji jejich typy.

```

public function onInit($param) {
    parent::onInit($param);
    if (!$this->isPostBack) {
        $SchvaleneBpVolne=$this->getDataVolne('nazevcs');
    }
}

```

```

        $this->SchvaleneBpVolnePraceGrid->DataSource=$SchvaleneBpVolne;
        $this->SchvaleneBpVolnePraceGrid->dataBind();
    }
}
protected function getDataVolne($ordersby) {
    $criteria = new CActiveRecordCriteria;
    $criteria->OrdersBy[$ordersby]='asc';
    return PraceSchvaleneVolneBpRecord::finder()->findAll($criteria);
}

```

Zdrojový kód 12: Načtení dat do datagridu v souboru TabListBPrace.php

PRADO poskytuje několik typů sloupečků, přičemž v SEPu jsou použity:

- TBoundColumn – spojuje buňky s určitým polem dat, např. obsazenost může nabývat hodnot volná, rezervovaná nebo obsazená,
- THyperLinkColumn – zobrazuje v buňkách odkazy,
- TTemplateColumn – zobrazuje buňky s volitelnou šablonou, např. u zadavatele musím nejprve z NetID uloženého v databázi získat za pomoci webových služeb IS/STAG jméno a tituly.

Obdobně jako je tomu u rendereru, může mít i datagrid alternativní styl. V souboru **TabListBPrace.page** (viz kód 13) je tento styl nedefinován pouze pro barvu pozadí (řádek 9). Stejně tak může být určeno, co se zobrazí v případě, že je DataSource prázdný (viz EmptyTemplate, řádek 10). Název práce je klasický hypertextový odkaz a odkazuje na detail práce. Jak již bylo řečeno, atribut zadavatel se zobrazuje ve sloupečku typu TTemplateColumn. Ten se používá v případě, že před samotným zobrazením dochází k úpravě dat. V databázi SEPu ukládám pouze NetID pedagoga a všechny jeho ostatní údaje čtu z WS STAG. O to se stará funkce udajeUcitel(\$netid) volaná funkcí jmenozadavatele() na řádku 26. Jejich výstupem je celé jméno zadavatele včetně titulů před i za jménem a ústav pedagoga. Pokud takový pedagog neexistuje, vypíše se upozornění na tuto skutečnost.

```

1  <h2><a name="BPvolne"></a>Bakalářské práce &ndash; volné</h2>
2  <com:TDataGrid ID="SchvaleneBpVolnePraceGrid"
3      DataKeyField="id_prace"
4      AutoGenerateColumns="false"
5      AllowSorting="True"
6      OnSortCommand="sortSchvaleneBpVolnePraceGrid"
7      OnItemCreated="schvaleneBpVolnePraceGridCreated"
8      ItemStyle.BackColor="#ffffef"
9      AlternatingItemStyle.BackColor="#eee"
10     EmptyTemplate="<p><b>Žádné vypsané bakalářské práce nejsou volné.</b></p>">
11     <com:THyperLinkColumn
12         HeaderText="Název"
13         DataTextField="navezcs"
14         SortExpression="navezcs"

```

```

15     DataNavigateUrlField="id_prace">
16     <prop:DataNavigateUrlFormatString>#
17         $this->Service->createUrl('prace.DetailPrace', array('id' => {0}))
18     </prop:DataNavigateUrlFormatString>
19 </com:THyperLinkColumn>
20 <com:TTemplateColumn
21     HeaderText="Zadavatel"
22     SortExpression="zkratkaprijmeni">
23     <prop:ItemTemplate>
24         <%%
25             if ($this->Parent->DataItem->zadavatel_netid) {
26                 echo MojePrace::jmenozadavatele($this->Parent->DataItem->zadavatel_netid,
27                     True);
28             } else {
29                 echo ' &mdash;';
30             }
31         <%>
32     </prop:ItemTemplate>
33 </com:TTemplateColumn>
34 <com:TBoundColumn
35     HeaderText="Rok"
36     DataField="datum_odevzdani"
37     SortExpression="datum_odevzdani"
38     ItemStyle.HorizontalAlign="Center"
39     ItemStyle.Width="50px" />
40 <com:TBoundColumn
41     HeaderText="Obsazenost"
42     DataField="stav"
43     SortExpression="stav"
44     ItemStyle.HorizontalAlign="Center"
45     ItemStyle.Width="100px" />
46 </com:TDataGrid>

```

Zdrojový kód 13: Ukázka použití komponenty TDataGrid v souboru TabListBPrace.page

Webové služby systému STAG používám i pro čtení informací o studentovi. Celkem jsem v souboru **wsSTAG.php** sestavila tři funkce s dotazy na WS STAG. První funkce `osobniCislaStudent($netid)` vypíše všechna osobní čísla studenta s daným NetID. Funkce `udajeStudent($osCislo)` vrací pole se všemi údaji o studentovi s konkrétním osobním číslem. Z vracených údajů používám pro SEP jen titul před a za jménem, jméno a příjmení, které jsem si vložila do jedné proměnné `celejmeno`, a dále zkratku oboru, kterou vypisují obvykle do závorky za celé jméno. Poslední funkce `udajeUcitel($netid)` se již týká pedagoga a vypíše opět celé pole údajů, např. titul před a za jménem, jméno, příjmení, katedru nebo e-mail.

Soubor **wsSTAG.php** je globální třídou, která se do souborů, kde jsou potřeba její funkce, vkládá příkazem `Prado::Using('Application.modules.wsSTAG');`. Ve skutečnosti se jedná o PHP příkaz `include_once`. Soubor **wsSTAG.php** musí být v tomto případě umístěn ve složce `protected\modules`.



## 4.6 Autentizace a autorizace

V současné době je SEP v testovacím provozu na serveru mimo Univerzitu Pardubice. Přihlašování proto neprobíhá pomocí centrálního autentizačního systému, ale přes tabulku uživatelů, kde jsou uložena NetID, smyšlená hesla a role jen vybraných jedinců. Pro nasazení do ostrého provozu je nutné připojit SEP k CAS a zároveň upravit tři soubory – `SEPUser.php`, `LoginUser.php` a `LoginUser.page`.

Kód 14 ukazuje obsah souboru `SEPUser.php`, kde je na řádce 2 začleňován soubor `TDbUserManager.php`, který definuje třídu `TDbUser`. Nově vytvořená třída `SEPUser` je jejím rozšířením a reprezentuje data o uživateli potřebná pro uchování v session. Základní provedení uchovává informace o NetID a oprávnění (roli). Vytvořený objekt `SEPUser` je založený na konkrétním NetID a kontroluje v databázi, zda daný uživatel skutečně existuje. Pokud ano, vytvoří session a nastaví počáteční hodnoty. Vstupním parametrem funkce `createUser()` je NetID a výstupem pak objekt `SEPUser` nebo hodnota `null`, pokud NetID není správné. Funkce `validateUser()` kontroluje, zda je zadaná kombinace NetID a hesla správně. Vstupními parametry jsou NetID a heslo a vrací hodnotu `boolean`, zda jsou zadané údaje validní. Obě zmíněné funkce jsou ve třídě `TDbUser` povinné. Funkce `getIsStudent()` ukládá informaci o roli přihlášeného studenta. Stejným způsobem jsou nadefinovány i role pro administrátora, vedoucího ústavu i pedagoga, které již v kódu 14 nejsou zobrazeny.

```
1 <?php
2 Prado::using('System.Security.TDbUserManager');
3
4 class SEPUser extends TDbUser {
5     public function createUser($netid) {
6         // use UzivateleRecord Active Record to look for the specified netid
7         $userRecord = UzivateleRecord::finder()->findByPk($netid); // NEBUDE V
            PRIHLASOVANI PRES CAS
8         if ($userRecord instanceof UzivateleRecord) {
9             $user = new SEPUser($this->Manager);
10            $user->Name=$netid;
11            switch ($userRecord->id_opravneni) { // NEBUDE V PRIHLASOVANI PRES CAS
12                case 1: $user->setRoles('Student'); break;
13                case 2: $user->setRoles('Pedagog'); break;
14                case 3: $user->setRoles('Vedoucí'); break;
15                case 4: $user->setRoles('Admin'); break;
16            }
17            $user->IsGuest=False; // the user is not a guest
18            return $user;
19        }
20        else return null;
21    }
22    public function validateUser($netid,$heslo) {
23        // use UzivateleRecord Active Record to look for the (netid, heslo) pair.
24        return UzivateleRecord::finder()->findBy_netid_AND_heslo($netid,md5($heslo))!==
            null; // NEBUDE V PRIHLASOVANI PRES CAS
25    }
```

```

26     public function getIsStudent() {
27         return $this->isInRole('Student');
28     }
29 }
30 ?>

```

Zdrojový kód 14: Validace NetID a hesla a vytvoření session v souboru SEPUser.php

Při autentizaci proti LDAP, který je využíván pro CAS, je nutné získat NetID a uložit ho do `$this->user->name`. Dále je potřeba zjistit, jakou roli uživatel v SEP bude mít a uložit ji do `$this->isInRole()`. To by se muselo získávat buď využitím struktury NetID, kde u studenta začíná na „st“ následované čísly a u pedagoga jsou na začátku čtyři písmena a pak teprve čísla. NetID vedoucího ústavu a administrátora by musela být uložena např. v tabulce nastavení nebo v nějakém souboru s nastavením. Druhou možností se jeví zjišťovat, v jaké větvi struktury LDAP je dané NetID uloženo. Nejsem si ovšem jistá, zda je to technologicky možné.

## 4.7 Role studenta

Student, který se přihlásí pomocí svého NetID a hesla, uvidí vše stejně jako nepřihlášený uživatel. Funkcí navíc je možnost rezervovat volné práce. V detailu práce, která není obsazená ani rezervovaná, se mu zobrazuje roletové menu s osobními čísly přihlášeného studenta. Student studující pouze v jednom oboru zde nalezne jen jedno osobní číslo, které vybere a tím se mu zobrazí jméno studenta a obor, k němuž osobní číslo náleží. Zároveň se uvolní tlačítko „Rezervovat“ a student si jím práci může rezervovat. Pokud student studuje více oborů, má ke každému přiřazeno unikátní osobní číslo. Při rezervaci tématu si student vybere konkrétní osobní číslo podle toho, ke kterému jeho oboru se práce vztahuje. Poté se zobrazí jméno studenta, daný obor a je možné kliknout na „Rezervovat“. Při změně vybraného osobního čísla se změní i zobrazený obor.

```

$mail = new posliMail();
$komu = 'lada.penickova@gmail.com'; // $praceRecord->student_netid . '@student.upce.cz';
$odkoho = 'sep@upce.cz';
$predmet = 'Potvrzení registrace práce v systému SEP';
$text = '<html>Dobrý den,<br />potvrzujeme Vaši rezervaci práce v systému <a href="http
: //147.230.72.169/SEP/develop">SEP</a>.<br />S pozdravem<br />Team SEP</html>';
$mail->sendemail($komu, $odkoho, $predmet, $text);

```

Zdrojový kód 15: Nadefinování parametrů pro odeslání e-mailu v souboru RezervacePrace.php

Může nastat situace, kdy student již má nějakou práci rezervovanou nebo obsazenou a pokusí se rezervovat si jinou. V takovém případě se po stisku tlačítka „Rezervovat“ zobrazí upozornění, které je vyobrazeno na obrázku 9. Toho je docíleno pomocí funkce `checkNetidOsobni()`, která kontroluje, zda se kombinace osobního čísla a netid přihlášeného studenta již v tabulce práce vyskytuje. Pokud ano, zobrazí komponenta `TCustomValidator` již zmíněné upozornění. Jinak je student přiřazen k rezervované práci a přesměrován na detail práce. V detailu rezervované práce

## REZERVACE PRÁCE

### Systém pro evidenci bakalářských a diplomových prací

Název v angličtině:	System for Evidence of Bachelor and Diploma Thesis
Typ práce:	BP
Popis:	Cílem práce je navrhnout a implementovat takový systém, který bude užitečný jak pro studenty, tak pro pedagogy. Práce je vhodná pro obor RIM a IVS.
Stav:	Volná
Zadavatel práce:	Ing. Miloslav Hub, Ph.D.
Jméno studenta:	— (-)
Rok odevzdání:	2011
Osobní čísla:	
<input type="text" value="E08454"/>	Nelze rezervovat tuto práci, kombinace NetID a osobního čísla je již v databázi u jiné práce. Pokud je předchozí rezervace neplatná, musí ji její vedoucí označit jako neaktuální, obhájenou nebo volnou.
<input type="button" value="Rezervovat"/>	Nalezen student <b>Bc. Marie AUGUSTINOVÁ</b> (Regionální a informační management)

Obrázek 9: Rezervace práce studentem

*Zdroj: vlastní*

vidí student větu „Tuto práci máte aktuálně rezervovanou, kontaktujte vedoucího na“ a e-mail vedoucího práce.

Po úspěšném rezervování práce odešle systém studentovi potvrzení také na jeho e-mail. K nadefinování paramterů e-mailu slouží kód 15 v souboru **RezervacePrace.php**. Musí být určeno komu a od koho má být e-mail poslán, jaký text bude v předmětu a jaký v těle zprávy.

O samotné odeslání e-mailu se stará modul nazvaný **phpmailer** a umístěný v adresáři **modules**. Jeho dva soubory **class.phpmailer.php** a **class.smtp.php** jsou načítány na řádcích 2 a 3 kódu 16, který zobrazuje obsah souboru **posliMail.php**. Ten je rovněž umístěný v adresáři **modules** a slouží k nadefinování proměnných potřebných k odeslání e-mailu, např. kódování (řádek 10), SMTP server (řádek 12), přihlašovací údaje k SMTP serveru (řádky 20 a 21) apod. V současné době SEP neběží na serveru Univerzity Pardubice a nemá nastavený SMTP server. Proto k odesílání e-mailů používám SMTP server mé soukromé pošty a slouží tedy jako ukázka, co je možné při ostrém provozu nastavit a používat. E-maily by následně mohly být posílány i studentovi při obsazování práce zadávané pedagogem, pedagogovi při rezervaci studentem nebo při schválení vedoucím ústavu.

```
1 <?php
2 require_once('./protected/modules/phpmailer/class.phpmailer.php');
3 include("./protected/modules/phpmailer/class.smtp.php"); // optional, gets called from
  within class.phpmailer.php if not already loaded
4
5 class posliMail {
6     function sendemail($komu, $odkoho, $predmet, $text) {
7         $email          = new PHPMailer();
8         $body           = $text;
9
10        $email->CharSet  = "utf-8";
11        $email->IsSMTP() // telling the class to use SMTP
```

```

12     $email->Host      = "smtp.gmail.com"; // SMTP server
13     $email->SMTPDebug = 2;                // enables SMTP debug information (for testing)
14                                         // 1 = errors and messages
15                                         // 2 = messages only
16     $email->SMTPAuth  = true;              // enable SMTP authentication
17     $email->SMTPSecure = "ssl";           // sets the prefix to the servier
18     $email->Host      = "smtp.gmail.com"; // sets GMAIL as the SMTP server
19     $email->Port       = 465;              // set the SMTP port for the GMAIL server
20     $email->Username   = $komu ;           // GMAIL username
21     $email->Password   = "heslo";         // GMAIL password
22
23     $email->SetFrom('sep@upce.cz', 'SEP (NO-REPLY)');
24     $email->Subject    = $predmet;
25     $email->MsgHTML($body);
26
27     $address = "lada.penickova@gmail.com";
28     $email->AddAddress($address, "Lada Pěničková");
29
30     if(!$email->Send()) {
31         echo "Mailer Error: " . $email->ErrorInfo;
32     } else {
33         echo "Message sent!";
34         echo $body;
35     }
36 }
37 }
38 ?>

```

Zdrojový kód 16: Posílání e-mailů nadefinované v souboru posliMail.php

Odkaz v hlavním menu, který se zobrazuje pouze studentovi, je „Vlastní práce“. Pod ním se skrývá detail prací, které má student rezervované nebo obsazené. U každé je vypsán název v češtině a angličtině, vedoucí práce, popis, osobní číslo, pro které je obsazena, a dále zda se jedná o práci bakalářskou nebo diplomovou, zda je rezervována nebo obsazena a zda je práce aktuální nebo obhájená. Student může mít pro každý obor, který studuje, pouze jednu aktuální práci.

## 4.8 Role pedagoga

Po přihlášení pedagoga se zobrazí úvodní stránka shodná s úvodní stránkou pro studenta i nepřihlášeného uživatele. Stejně tak má možnost si prohlédnout všechny schválené práce, které vypsali i jeho kolegové. Změnilo se ale menu, kde přibýly odkazy na „moje práce“ a „přidat práci“. Pedagog, který je v SEP poprvé, uvidí na stránce Moje práce pouze nadpisy a upozornění, že nemá žádné práce vypsané. Jeho nejdůležitější funkcí v SEP je vypisování nových prací pomocí formuláře pro novou práci. Zde vyplní český a anglický název práce, její popis a typ (zda se jedná o bakalářskou nebo diplomovou práci). Bez těchto údajů nelze formulář odeslat. Dále může změnit rok odevzdání, pokud práci nechce vpsat pro aktuální rok. Pokud již při zadávání

nové práce zná studenta, který ji bude zpracovávat, klikne na výběrové tlačítko „Rezervovaná“ či „Obsazená“. Tím se zobrazí pole, kam pedagog napíše NetID studenta a tlačítkem „Zkontroluj NetID“ ověří, zda tento student existuje a zobrazí jeho celé jméno a obor, který studuje. Dále se zobrazí roletové menu, kde je nutné vybrat osobní číslo zobrazeného studenta. Obvykle mají studenti jen jedno osobní číslo, ale může se stát, že studují ve více oborech a pro každý obor mají přidělené unikátní osobní číslo. Pokud pedagog nezná osobní číslo obsazovaného studenta, je mu nápovědou obor, který se zobrazuje za jménem studenta při výběru konkrétního osobního čísla. Názorný příklad výše popsaného je na obrázku 10. Dokud není vybráno osobní číslo studenta a je zatrženo, že se jedná o rezervovanou nebo obsazenou práci, formulář není možné odeslat. Po úspěšném odeslání dat tlačítkem „Vytvoř“, je pedagog přesměrován na detail nově vypsané práce, kde může zkontrolovat správnost zadaných údajů, případně editovat, uzavřít či celou práci smazat.

**PŘIDÁNÍ NOVÉ PRÁCE**

**Stav:**  
 Volná  Rezervovaná  Obsazená

**NetID studenta:**  
st16502  Nalezen student **Bc. Pavel ŠPAČEK** (Pojistné inženýrství)

**Osobní čísla:**  
▼  
E090506  
D09722

**V práci:**

**Anglický název práce:**

Obrázek 10: Obsazení práce studentem

*Zdroj: vlastní*

Pokud již pedagog má nějaké práce vypsané, zobrazí se mu na stránce *Moje práce* tabulkový seznam všech aktuálních prací, které vypsál. Práce jsou rozdělené na nově vypsané, čekající na schválení, neschválené a schválené (atribut status v tabulce práce). V první tabulce je seznam všech těchto prací dohromady. Tyto seznamy jsou vytvořeny pomocí komponenty *TDataGrid*, kterou jsem popsala v kapitole 4.5. Každá tabulka musí být v samostatném datagridu. Název každé práce odkazuje na její detail, kde je možné práci editovat, uzavřít nebo smazat. V detailu nově vypsané práce se pedagogovi zobrazuje tlačítko „Odeslat ke schválení“, čímž práci přesune do čekajících na schválení.

Nejpoužívanější funkcí bude jistě editace práce, která se používá i pro obsazení práce studentem. Zde je několik možností, které mohou nastat:

- práce je nová nebo čekající na schválení,

- práce je neschválená,
- práce je schválená a pedagog chce změnit pouze obsazení práce,
- práce je schválená a pedagog chce změnit i atributy práce.

Pokud je práce nová nebo čekající na schválení, může ji zadavatel měnit bez omezení a to jak obsazení, tak samotné atributy práce. Podobné je to u neschválené práce, kde se navíc zobrazuje poznámka schvalujícího pedagoga (vedoucího ústavu). Jestliže je však práce schválená, může pedagog libovolně měnit pouze obsazení práce. Všechny ostatní atributy jsou nepřístupné. V takovém případě pedagog vybere, zda je práce volná, rezervovaná nebo obsazená a případně doplní NetID studenta, kterému práci přiřazuje. Následuje stejný postup jako u zadávání studenta k nově vypisované práci. Opět je nutné NetID zkontrolovat a vybrat správné osobní číslo. Poté je možné změny uložit.

Jestliže je práce schválená a pedagog chce změnit i atributy práce, musí zatrhnout políčko „Editovat vlastnosti práce“. Tím se atributy zpřístupní a zároveň s tím se zobrazí upozornění, že po editaci vlastností musí pedagog práci znovu odeslat ke schválení. V případě, že je práce označena jako rezervovaná nebo obsazená, je nutné před uložením ověřit existenci zadaného NetID kliknutím na tlačítko „Zkontroluj NetID“ a vybrat osobní číslo studenta.

Práce, které jsou již zadané a studenti je i obhájili u státnic, je potřeba vyřadit z aktuálních témat. Může se také stát, že dané téma ještě neobhájené práce již není v danou chvíli aktuální, ale třeba někdy později by se ještě mohlo hodit. Pedagog takovou práci nemusí rovnou smazat, ale má možnost ji přesunout do neaktuálních. V obou případech se nabízí odkaz „uzavřít“ v detailu práce. Na stránce Uzavření práce si pedagog vybere, zda práce byla obhájena nebo se jedná o neaktuální téma. Aby mohla být práce přidána mezi obhájené, musí se jednat o práci:

- schválenou a
- obsazenou studentem.

Jestliže některá z těchto podmínek není splněna, je dopředu zaškrtnuto „Práce je neaktuální“ a tuto volbu nelze změnit (viz obrázek 11). K uzavření práce lze připojit poznámku, např. známku z obhajoby nebo důvod neaktuálnosti, kterou vidí pouze pedagog – zadavatel.

Pro přidání práce mezi neaktuální žádné omezení neplatí. Takovou práci uvidí pouze pedagog, který ji vypisoval. Ostatním se skryje i přesto, že byla schválená. Pokud byl u práce přiřazený student, přiřazení se zruší (student si bude moci rezervovat jinou práci), ale pedagog u práce stále uvidí jeho jméno. Seznam neaktuálních prací je přístupný z menu „Moje práce“ odkaz „Neaktuální práce“. Je vytvořen pomocí datagridu a u každého záznamu je odkaz pro přesunutí práce mezi aktuální. V takovém případě bude práce nová a volná. Pedagog ji bude moci libovolně

# UZAVŘENÍ PRÁCE

## System pro evidenci bakalářských a diplomových prací

- Práce byla obhájena.
- Práce je neaktuální – tuto práci již uvidíte jenom vy. Můžete ji však později znovu aktivovat.

Neobsazenou práci nelze přidat k obhájeným. Můžete ji uzavřít pouze jako neaktuální.

### Poznámka k uzavření práce:

---

Uložit

Obrázek 11: Upozornění při uzavírání neobsazené práce

*Zdroj: vlastní*

upravovat, smazat nebo opět přesunout mezi neaktuální, může ji odeslat ke schválení nebo přiřadit studentovi.

## 4.9 Role vedoucího ústavu

Vedoucí ústavu má všechna práva jako pedagog včetně vypisování nových prací, obsazování prací studenty, editaci, uzavření i mazání. Čím se vedoucí ústavu liší, je funkce schvalování témat. V hlavním menu přihlášeného vedoucího ústavu přibyl odkaz „Schvalování prací“, který odkazuje na přehled prací ke schválení. Práce jsou rozděleny na čekající na schválení, neschválené a schválené. Každá z těchto tabulek je vytvořena pomocí komponenty TDataGrid podobně jako v přehledu vlastních aktuálních prací. Všechny názvy neschválených a schválených prací jsou odkazy na detail. Názvy prací čekajících na schválení odkazují na schválení. Vedoucí ústavu se zde podle zobrazených atributů rozhodne, zda práci schválí či zamítne. Při zaškrtnutí volby „Zamítnout“ se zobrazí pole, kam schvalovatel připiše poznámku pro zadavatele práce (viz obrázek 12). Po potvrzení volby dojde k přesměrování na detail práce.

## 4.10 Další společné vlastnosti systému

### 4.10.1 Tisk

Požadavek na tisk lze řešit několika způsoby. Prvním a nejjednodušším způsobem je připravit kaskádové styly jednotlivých stránek pro tisk. Tuto možnost jsem zavrhl hned v počátku, neboť nedovoluje například přidávání dalších informací k těm, které jsou aktuálně zobrazeny na stránce. Rovněž není možné jednotlivé části přeskupovat či skrývat a dále je problém se strán-

# SCHVÁLENÍ PRÁCE

## System pro evidenci bakalářských a diplomových prací

<b>Název v angličtině:</b>	System for Evidence of Bachelor and Diploma Thesis
<b>Typ práce:</b>	BP
<b>Popis:</b>	Cílem práce je navrhnout a implementovat takový systém, který bude užitečný jak pro studenty, tak pro pedagogy. Práce je vhodná pro obor RIM a IVS.
<b>Stav:</b>	Volná
<b>Zadavatel práce:</b>	Ing. Miloslav Hub, Ph.D.
<b>Rok odevzdání:</b>	2011

Schválit  Zamítnout

Výhrady k zadání:

---

---

---

Potvrdit volbu

Obrázek 12: Schválení práce vedoucím ústavu

*Zdroj: vlastní*

kovými zlomy. Další možností, zřejmě nejuniverzálnější, je využití XSLT (Extensible Stylesheet Language Transformations), což dovoluje validní XML výstup z PRADO. Slouží pro libovolné zformátování výstupního textu (vzhledové i obsahové) pomocí XSLT procesoru vestavěného v internetovém prohlížeči nebo na serveru. Takovým procesorem může být např. Xalan (open-source XSLT procesor od sdružení Apache). Tato možnost lze využít i pro formátování do XSL-FO (Extensible Stylesheet Language – Formatting Objects), z něhož lze nějakým formátovacím procesorem na serveru (například FOP) vytvořit PDF (Portable Document Format), RTF (Rich Text Format) či další podporované formáty.

Mnou navržené řešení využívá pro transformaci textu do PDF přímo PHP. V první testovací verzi tisku jsem použila knihovny FPDF (Free PDF) pro přímý převod PHP na PDF. Tato varianta má několik úskalí – prvním jsou české fonty pro kódování UTF-8. Tento problém se mi podařilo vyřešit tvorbou vlastních fontů pomocí knihoven UFPDF (Unicode Free PDF) a open-source nástroje tt2ufm (True Type Font to Unicode Font Metrics) vyřešit. Dalším a pro mě podstatným problémem bylo zpracování tabulek (zalamování textu v buňkách, zalomení stránek atd.). K nalezení nejvhodnějšího řešení mě přivedla diskuse přímo na serveru Pradosoft.com. Je jí knihovna DOMPDPDF [5] (Document Object Model PDF), která dokáže transformovat do PDF přímo vstupní zdrojový kód (X)HTML. DOMPDPDF podporuje rovněž stylování pomocí CSS. Pro



tvorbu záhlaví a zápatí stránek (stránkování, loga, ...) je také možné použít předchozí variantu s přímým přístupem k objektům PDF. Klíčové pro použití DOMPDF bylo rovněž vytvoření vlastních českých fontů z volně šiřitelného fontu DejaVu<sup>4</sup>(postup na [8]), které jsou přiloženy na CD s bakalářskou prací v adresáři s DOMPDF (dompdf\_0-6-0\_beta1). Tisk je nyní prováděn v několika základních krocích:

1. Po kliknutí na odkaz „Tisk do PDF“ je zavolána funkce, v níž se sestaví část vstupního XHTML kódu – to znamená, že dojde k načtení požadovaných dat z databázové tabulky či pohledu (například s využitím Active Records kritérií a `FindAll($criteria)` nebo pomocí `FindAllBySql($sql)`, podobně jako u vyhledávání), potom (pokud je nalezen nějaký záznam) se vkládají jednotlivé jeho části do buněk tabulky prostým spojováním řetězců.
2. Je zavolána funkce `tiskPDF` z modulu `tiskniPDF` se třemi parametry – vstupní HTML, nadpis textu v souboru PDF, název souboru:

```
tiskniPDF::tiskPDF($html,  
'Aktuální práce -- ' . $this->User->Name,  
'aktualni');
```
3. Modul `tiskniPDF` se postará o připojení HTML hlavičky, kaskádových stylů a ukončení HTML dokumentu.
4. Je zavolána knihovna DOMPDF:

```
$dompdf = new DOMPDF();  
$dompdf->load_html($html);  
$dompdf->render();  
$dompdf->stream($file . ".pdf", array("Attachment" => 1));
```
5. Prohlížeč pak podle nastavení otevře dokument *aktualni.pdf* přímo v sobě jako PDF (například v Adobe Acrobat Readeru) nebo nabídne stažení či otevření mimo prohlížeč.

#### 4.10.2 Vyhledávání

Všichni uživatelé mají stejné možnosti při hledání v pracích. Vyhledávání je možné buď v aktuálních, nebo obhájených pracích. Dále si uživatelé mohou vybrat, zda budou vyhledávat podle názvu práce v češtině a angličtině nebo podle pedagoga (viz obrázek 13).

Zadat lze minimálně tři znaky, jinak bude uživatel upozorněn a vyhledávání neproběhne. Funkce, která provádí samotné vyhledávání, je nazvána `getNalezenePrace($offset,$limit)` a na-

---

<sup>4</sup>není možné použít fonty například ze systému Windows, které jsou chráněny autorskými právy (např. Arial)

## VYHLEDÁVÁNÍ

Podle názvu práce
  Podle pedagoga

**Stav:**  
 Aktuální
  Obhájené

**Zadejte text v názvu práce:**

---

**Systém pro evidenci bakalářských a diplomových prací (System for Evidence of Bachelor and Diploma Thesis)**  
**Vedoucí práce:** Ing. Miloslav Hub, Ph.D. (USII)  
**Práce je volná**  
**Popis práce:**  
 Cílem práce je navrhnout a implementovat takový systém, který bude užitečný jak pro studenty, tak pro pedagogy.  
 Práce je vhodná pro obor RIM a IVS.

Obrázek 13: Zobrazený výsledek vyhledávání

*Zdroj: vlastní*

chází se v souboru `HledatPrace.php` (viz kód 17). Vytváří nový objekt třídy `TActiveRecordCriteria` s parametry podmínka (`Condition`), maximální počet vrácených záznamů na stránce (`Limit`) a odstup prvního záznamu zobrazené stránky od celkově prvního záznamu (`Offset`). V proměnných `$this->_navezcs` a `$this->_nazeven` je uložen hledaný řetězec. Dalšími parametry jsou `OrdersBy[]`, které nastavují třídění. V kódu 17 se jedná o vzestupné třídění podle příjmení vedoucího práce a poté podle názvu v češtině. Jelikož u aktuálních témat není uloženo přímo příjmení pedagoga, vytvořila jsem pohled na tabulku práce. V něm ukládám zkratku příjmení, kterou získám z NetID odstraněním prvních dvou písmen představujících křestní jméno. V tabulce práceobhajene již je příjmení pedagoga jako samostatný atribut a není nutné ho více upravovat. Výsledky vyhledávání jsou zobrazovány pomocí repeateru obdobně jako na úvodní stránce.

```

protected function getNalezenePrace($offset, $limit) {
    $criteria=new TActiveRecordCriteria;
    $criteria->Condition = 'navezcs LIKE \'' . $this->_navezcs . '\'' OR nazeven LIKE \'' .
        $this->_nazeven . '\'' ';
    $criteria->Limit=$limit;
    $criteria->Offset=$offset;
    if ($this->DruhPrace->Text=='Aktuální') {
        $criteria->OrdersBy['zkratkaprijmeni']='asc';
        $criteria->OrdersBy['navezcs']='asc';
        return PraceAktualniSchvaleneRecord::finder()->findAll($criteria);
    } else {
        $criteria->OrdersBy['zadavatel_prijmeni']='asc';
        $criteria->OrdersBy['navezcs']='asc';
        return PraceObhajeneRecord::finder()->findAll($criteria);
    }
}

```

Zdrojový kód 17: Funkce pro vyhledávání v souboru `HledatPrace.php`

Při vyhledávání podle pedagoga již uživatel nepíše hledaný výraz, ale vybírá jméno peda-

goga z roletového menu. To je plněno při inicializaci stránky, jak ukazuje kód 18. V pohledu pedagogove jsou uložena všechna NetID pedagogů, kteří se vyskytují v pohledu s aktuálními schválenými pracemi. Z pohledu pedagogove se nyní načte a uloží každé NetID do proměnné \$ucitele[\$polozka->zadavatel\_netid] v poli \$ucitele. Pomocí funkce jmenozadavatele() se z WS STAG doplní celé jméno včetně titulů a zkratky ústavu a následně je seznam pedagogů připraven k zobrazení v komponentě TDropDownList s ID="PedagogNetid".

```
public function onInit($param) {
    parent::onInit($param);
    $pedagogove = PedagogoveRecord::finder()->findAll();
    $ucitele = Array();
    foreach ($pedagogove as $polozka) {
        $ucitele[$polozka->zadavatel_netid] = funkcePrace::jmenozadavatele($polozka->
            zadavatel_netid, True);
    }
    $this->PedagogNetid->DataSource=$ucitele;
    $this->PedagogNetid->DataBind();
}
```

Zdrojový kód 18: Vybrání všech pedagogů v tabulce prace – soubor HledatPedagog.php

### 4.10.3 RSS kanál

RSS (Rich Site Summary<sup>5</sup>) je jedním z XML formátů určených pro čtení novinek na webových stránkách a obecněji sledování obsahu. Technologie RSS umožňuje uživatelům Internetu přihlásit se k odběru novinek z webu, který nabízí RSS zdroj (RSS feed, též RSS kanál, RSS channel). Tento zdroj se většinou vyskytuje na stránkách, kde se obsah mění a často přidává [4].

Software určený k práci s RSS kanály se označuje jako RSS čtečka. Může se jednat o samostatný specializovaný program nebo o plug-in do jiného programu (typicky webového prohlížeče). Tato funkce může být v jiném programu rovnou vestavěna, případně se může jednat o webovou aplikaci poskytující tuto funkčnost (např. Google Reader). Pokud některý webový server nabízí RSS kanály, obvykle to indikuje ikonkou, která vede přímo na URL příslušného zdroje, který uživatel zadá do čtečky a ta poté zobrazí seznam všech takto zpřístupněných informací či odkazů. Čtečka pak automaticky pravidelně kontroluje toto URL a zobrazuje nové položky [4].

Užitečnost RSS kanálu mě přivedla na myšlenku jej realizovat i v SEP. Na Pradosoft.com se mi podařilo nalézt kostru funkčního modulu pro RSS kanál od Andrease Bullinga [3] využívající PRADO třídu TFeedService, který jsem upravila tak (viz kód 19), aby načítal data z databázových pohledů stejně jako úvodní stránka SEP (modules\FeedProvider.php).

```
switch( $this->_type ) {
    case 'PraceB' :
```

<sup>5</sup>Samotná zkratka má mimo uvedeného i několik dalších výkladů – Resource Description Framework Site Summary (stvořitelem je organizace W3C) a Really Simple Syndication.

```

$records = PraceSchvaleneVolneBpRecord::finder()->findAll();
$channel_title = 'Bakalářské práce';
$channel_desc = 'Nově vypsané bakalářské práce v systému SEP';
$odkaz = '?page=prace.DetailPrace&id=';
break;
}

```

Zdrojový kód 19: Načtení dat pro RSS kanál v souboru FeedProvider.php

RSS kanál v SEP nabízí sledování volných schválených bakalářských a diplomových prací. V jednom záznamu RSS je momentálně téměř celé zadání práce – jméno zadavatele (využití modulu wsSTAG.php), název a typ práce, popis práce a datum vypsaní práce (datum schválení vedoucím). Zobrazen je rovněž odkaz na detail práce. Tyto informace jsou posílány jako XML soubor na klientský počítač.

Kód 20 ukazuje, jak se do XML souboru RSS kanálu uloží název práce s odkazem na její detail.

```

switch( $this->_type ) {
case 'PraceB':
foreach( $records as $record ) {
$i = 0;
$channelItems = new TXmlElement('item');

$itemTitle = new TXmlElement('title');
$title = $record->nazevcs;
$itemTitle->setValue($title);
$channelItems->getElements()->insertAt($i++, $itemTitle);

$itemLink = new TXmlElement('link');
$itemLink->setValue($projectweb.'/index.php'.$odkaz.$record->id_prace);
$channelItems->getElements()->insertAt($i++, $itemLink);

$channel->getElements()->insertAt(9, $channelItems);
}
break;
}

```

Zdrojový kód 20: Tvorba XML souboru RSS kanálu v souboru FeedProvider.php

Jelikož se odkaz na RSS kanál zobrazuje na všech stránkách SEP u, bylo vhodné ji přidat jako službu do souboru **Application.xml** (viz kód 21).

```

<service id="feed" class="System.Web.Services.TFeedService">
<feed id="PraceD" class="Application.modules.FeedProvider" />
<feed id="PraceB" class="Application.modules.FeedProvider" />
</service>

```

Zdrojový kód 21: Nastavení služby RSS kanálu v souboru application.xml

Studenti, ale i pedagogové, mohou být pak bez přímého připojení k SEP upozorněni na nové práce. RSS kanál by v budoucnu mohl být rozšířen o zabezpečený kanál (Secure RSS), který by

mohl využít přihlášený pedagog například pro to, aby byl upozorněn, který student si rezervoval jím vypsanou práci, že jsou jeho práce schváleny apod.

## Závěr

Systém je navržen tak, aby pedagogovi usnadnil práci při vypisování a zadávání témat. Zároveň mu ale neubírá možnost rozhodovat o tom, zda student, který si práci rezervuje, bude také skutečným řešitelem daného tématu. Pedagog může studenta odmítnout, práci uvolnit a obsadit ji vhodnějším uchazečem.

Zadáním této bakalářské práce bylo identifikovat a formulovat požadavky na systém pro evidenci studentských prací, navrhnout tento systém a řešení implementovat. Tyto body zadání se mi podařilo splnit. Aplikace běží v testovacím provozu na serveru mimo Univerzitu Pardubice a nasazení do provozu brání jen úpravy týkající se centrální autentizace a autorizace. Použitý framework i vytvořená aplikace jsou dostatečně flexibilní s oddělením vzhledu od aplikační logiky a je možné systém snadno upravovat a rozšiřovat.

Aplikace je podle mého názoru velice přehledná a její chování se nemění v majoritních webových prohlížečích. Jednou z velice důležitých funkcí systému je ošetření vstupních hodnot, které zadávají uživatelé při vyplňování formulářů. Mimo základní funkčnost je otestována možnost:

- posílání e-mailů,
- RSS čteček,
- tisku do PDF,
- vyhledávání,
- třídění záznamů tabulky.

Na vytvořeném systému lze mnoho věcí dále vylepšovat. Například je možné ještě lépe využít webové služby systému STAG, výše zmíněné funkce aplikovat na všechny části systému (rozšířit možnost tisku, přidat zabezpečené RSS kanály i pro přihlášené uživatele), či optimalizovat vyhledávání atd.

Nejdůležitější však bude systém pořádně otestovat, testovací režim může probíhat paralelně s dosavadním vypisováním studentských prací. Mnoho oprav v SEP bylo nutné udělat v důsledku anomálií ve webových službách STAG (všichni učitelé nemají zadaný e-mail, pomocí platného NetID nelze najít žádné osobní číslo studenta apod.), které jsou rovněž ještě v testovacím provozu.

Zdrojový kód této práce je napsán v téměř 130 souborech (PHP, HTML, XML, JS, CSS), které zabírají přes 340 kB a je mým prvním pokusem o vytvoření „smysluplné“ webové aplikace. Mimo faktické použití vytvořeného systému považuji za přínosné i to, že text práce a zdrojové kódy mohou dobře posloužit jako návod studentům, kteří chtějí začít s objektovým programováním v PHP nebo se rozhodli používat framework PRADO.

## POUŽITÁ LITERATURA

- [1] KAJZAR, Dušan, POLÁŠEK, Ivan. *Tvorba informačních systémů I: strukturovaný a objektový přístup*. 1. vyd. Opava: Slezská univerzita v Opavě, 2003. 219 s. ISBN 80-7248-214-9.
- [2] Ostravská univerzita v Ostravě. *Směrnice rektora č. 120/08 o vysokoškolských kvalifikačních pracích*.
- [3] BULLING, Andreas. *PRADO PHP Framework* [online]. Jan. 28, 2009, 02:03:49 AM [cit. 2010-05-21]. RSS Feed. Dostupné z WWW: <<http://www.pradosoft.com/forum/index.php?topic=11447.0>>.
- [4] BUREŠ, Jiří. *Interval.cz* [online]. 16. 09. 2004 [cit. 2010-05-02]. RSS 2.0. Dostupné z WWW: <<http://interval.cz/clanky/rss-20/>>.
- [5] CARSON, Benj. *Dompdf : The PHP 5 HTML to PDF Converter* [online]. c2004-2008 [cit. 2010-02-14]. Dostupné z WWW: <<http://www.digitaljunkies.ca/dompdf/>>.
- [6] CIV, Západočeská univerzita v Plzni – *Webové služby nad IS/STAG* [online]. 2008 [cit. 2010-06-26]. Dostupné z WWW: <<https://stag-ws.upce.cz/ws/help>>.
- [7] České vysoké učení technické v Praze – *KOS, Informační systém, komponenta studium* [online]. [2008], 17. 09. 2008 [cit. 2008-11-13]. Dostupný z WWW: <<http://cs.felk.cvut.cz/~halaska/navody/indexkos.htm>>.
- [8] *Dompdf* [online]. 2009 [cit. 2010-02-14]. Dostupné z WWW: <<http://groups.google.com/group/dompdf>>.
- [9] Masarykova univerzita – *Informační systém Masarykovy univerzity: Podrobnosti o našem systému* [online]. [2009] [cit. 2008-11-13]. Dostupný z WWW: <[http://is.muni.cz/nas\\_system](http://is.muni.cz/nas_system)>.
- [10] PRADO Group – *PRADO PHP Framework* [online]. c2004-2008 [cit. 2009-12-03]. Dostupné z WWW: <<http://www.pradosoft.com/>>.
- [11] Univerzita Hradec Králové – *ISIT: Informační systém FIM* [online]. c2004, 21.09.2008 [cit. 2008-11-13]. Dostupný z WWW: <<http://isit.uhk.cz>>.
- [12] Univerzita Karlova v Praze – *Informační systém UK* [online]. c2008 [cit. 2008-11-14]. Dostupný z WWW: <<http://is.cuni.cz/IS-1.html>>.

- [13] Univerzita Pardubice – *IS STAG* [online]. [2008] [cit. 2008-11-13]. Dostupný z WWW: <<https://portal.upce.cz>>.
- [14] Univerzita Pardubice – *Organizační pokyny pro vypisování a zpracování závěrečných prací v ak.r. 2008/2009* [online]. 2008 [cit. 2008-11-20]. Dostupný z WWW: <<http://genesis.upce.cz/fakulty/fes/fes-studium/fes-studium-prezencni/fes-studium-prezencni-statnice/FES120407.html>>.
- [15] Vysoká škola báňská – Technická univerzita Ostrava – *Univerzitní informační systém VŠB-TU Ostrava* [online]. c2004-2008 [cit. 2008-11-14]. Dostupný z WWW: <<http://unis.vsb.cz>>.
- [16] Vysoká škola ekonomická v Praze – *Integrovaný studijní informační systém VŠE* [online]. [2008] [cit. 2008-11-14]. Dostupný z WWW: <<http://isis.vse.cz>>.
- [17] Vysoké učení technické v Brně – *Informační systém: ApolloVUT* [online]. c2004-2008 [cit. 2008-11-13]. Dostupný z WWW: <<https://www.vutbr.cz/index.php?page=is&wapp=webcis&parent=1&tail=1&lang=0>>.
- [18] *Zákon č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů* [online]. [cit. 2008-10-27]. Dostupný z WWW: <<http://www.msmt.cz/vzdelavani/uplne-zneni-zakona-c-111-1998-sb-o-vysokych-skolach-text-se-zpracovanymi-novelami>>. § 62 odst. 1 písm. f).



## **SEZNAM PŘÍLOH**

**Příloha A – Diagram datových toků první úrovně**

**Příloha B – Diagram datových toků – zobrazení prací**

**Příloha C – Diagram datových toků – editace prací**

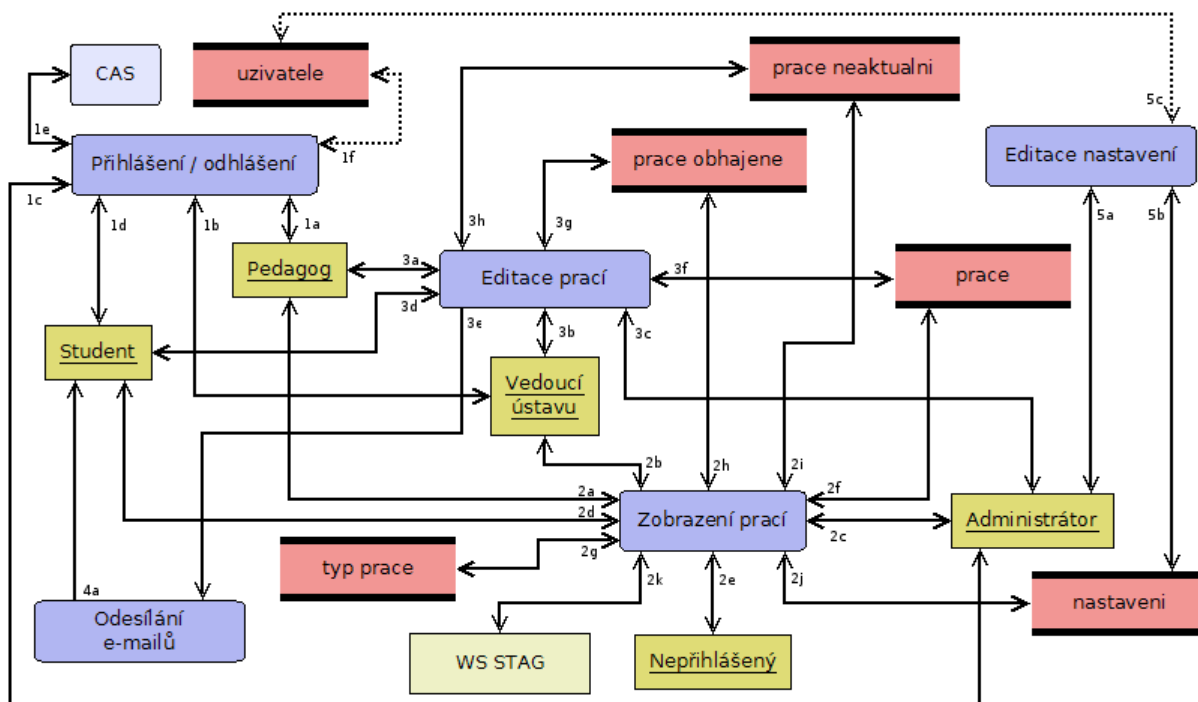
**Příloha D – Diagram datových toků – editace nastavení**

**Příloha E – Entity entitně relačního diagramu**

**Příloha F – Obsah CD**

**Příloha G – CD Systém pro evidenci bakalářských a diplomových prací**

## Příloha A – Diagram datových toků první úrovně



Obrázek 14: Diagram datových toků první úrovně

Zdroj: vlastní

### Slovní popis významu a obsahu znázorněných datových toků na obrázku 14

#### Proces 1 – Přihlášení/odhlášení:

- **1a:** požadavek pedagoga na přihlášení/odhlášení; výsledek přihlášení/odhlášení pedagoga
- **1b:** požadavek vedoucího ústavu na přihlášení/odhlášení; výsledek přihlášení/odhlášení vedoucího ústavu
- **1c:** požadavek administrátora na přihlášení/odhlášení; výsledek přihlášení/odhlášení administrátora
- **1d:** požadavek studenta na přihlášení/odhlášení; výsledek přihlášení/odhlášení studenta
- **1e:** požadavek na autentizaci; výsledek autentizace
- **1f:** požadavek na ověření existence přihlašovaného uživatele v tabulce uzivatele; výsledek ověření z tabulky uzivatele

## **Proces 2 – Zobrazení prací:**

- **2a:** požadavek pedagoga na zobrazení prací, detailu práce, vyhledávání, tisk; zobrazení prací, detailu práce, výsledek vyhledávání, náhled tisku pro pedagoga
- **2b:** požadavek vedoucího ústavu na zobrazení prací, detailu práce, vyhledávání, tisk; zobrazení prací, detailu práce, výsledek vyhledávání, náhled tisku pro vedoucího ústavu
- **2c:** požadavek administrátora na zobrazení prací, detailu práce, vyhledávání, tisk; zobrazení prací, detailu práce, výsledek vyhledávání, náhled tisku pro administrátora
- **2d:** požadavek studenta na zobrazení prací, detailu práce, vyhledávání, tisk; zobrazení prací, detailu práce, výsledek vyhledávání, náhled tisku pro studenta
- **2e:** požadavek nepřihlášeného uživatele na zobrazení prací, detailu práce, vyhledávání, tisk; zobrazení prací, detailu práce, výsledek vyhledávání, náhled tisku pro nepřihlášeného uživatele
- **2f:** požadavek na výpis z tabulky "prace"; informace z tabulky "prace"
- **2g:** požadavek na výpis z tabulky "typ\_prace"; informace z tabulky "typ\_prace"
- **2h:** požadavek na výpis z tabulky "prace\_obhajene"; informace z tabulky "prace\_obhajene"
- **2i:** požadavek na výpis z tabulky "prace\_neaktualni"; informace z tabulky "prace\_neaktualni"
- **2j:** požadavek na informace z tabulky "nastaveni"; informace z tabulky "nastaveni"
- **2k:** požadavek na informace z WS STAG; informace z WS STAG

## **Proces 3 – Editace údajů o práci:**

- **3a:** požadavek pedagoga na přidání nové práce, smazání, rezervaci, odeslání ke schválení, uzavření, editaci vlastností práce, zaktualnění neaktuální práce; potvrzení o přidání nové práce, smazání, rezervaci, odeslání ke schválení, uzavření, editaci vlastností práce, zaktualnění neaktuální práce pro pedagoga
- **3b:** požadavek vedoucího ústavu na přidání nové práce, smazání, rezervaci, odeslání ke schválení, uzavření, editaci vlastností práce, zaktualnění neaktuální práce, schválení / zamítnutí; potvrzení o přidání nové práce, smazání, rezervaci, odeslání ke schválení, uzavření, editaci vlastností práce, zaktualnění neaktuální práce, schválení / zamítnutí pro vedoucího ústavu

- **3c:** požadavek administrátora na přidání nové práce, smazání, rezervaci, odeslání ke schválení, uzavření, editaci vlastností práce, zaktualnění neaktuální práce, schválení / zamítnutí; potvrzení o přidání nové práce, smazání, rezervaci, odeslání ke schválení, uzavření, editaci vlastností práce, zaktualnění neaktuální práce, schválení / zamítnutí pro administrátora
- **3d:** požadavek studenta na rezervaci práce; potvrzení rezervace
- **3e:** požadavek na odeslání e-mailu
- **3f:** editace tabulky "prace"; potvrzení editace tabulky "prace"
- **3g:** editace tabulky "prace\_obhajene"; potvrzení editace tabulky "prace\_obhajene"
- **3h:** editace tabulky "prace\_neaktualni"; potvrzení editace tabulky "prace\_neaktualni"

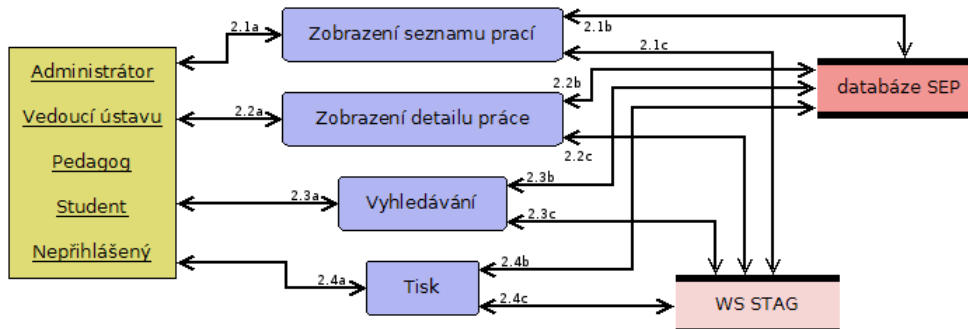
#### **Proces 4 – Odeslání e-mailů:**

- **4a:** odeslání e-mailu studentovi

#### **Proces 5 – Editace nastavení:**

- **5a:** požadavek administrátora na editaci nastavení; potvrzení editace nastavení
- **5b:** editace tabulky "nastaveni"; potvrzení editace tabulky "nastaveni"
- **5c:** editace tabulky "uzivatele"; potvrzení editace tabulky "uzivatele"

## Příloha B – Diagram datových toků – zobrazení prací



Obrázek 15: Diagram datových toků – zobrazení prací

Zdroj: vlastní

### Slovní popis významu a obsahu znázorněných datových toků na obrázku 15

#### Proces 2.1 – Zobrazení seznamu prací:

- **2.1a:** požadavek administrátora, vedoucího ústavu, pedagoga, studenta nebo nepřihlášeného uživatele na zobrazení seznamu prací; zobrazení seznamu prací
- **2.1b:** požadavek na zobrazení seznamu prací z databáze; zobrazení seznamu prací z databáze
- **2.1c:** požadavek na zobrazení informací z WS STAG pro seznam prací; zobrazení informací z WS STAG pro seznam prací

#### Proces 2.2 – Zobrazení detailu práce:

- **2.2a:** požadavek administrátora, vedoucího ústavu, pedagoga, studenta nebo nepřihlášeného uživatele na zobrazení detailu práce; zobrazení detailu práce
- **2.2b:** požadavek na zobrazení detailu práce z databáze; zobrazení detailu práce z databáze
- **2.2c:** požadavek na zobrazení informací z WS STAG pro detail práce; zobrazení informací z WS STAG pro detail práce

#### Proces 2.3 – Vyhledávání:

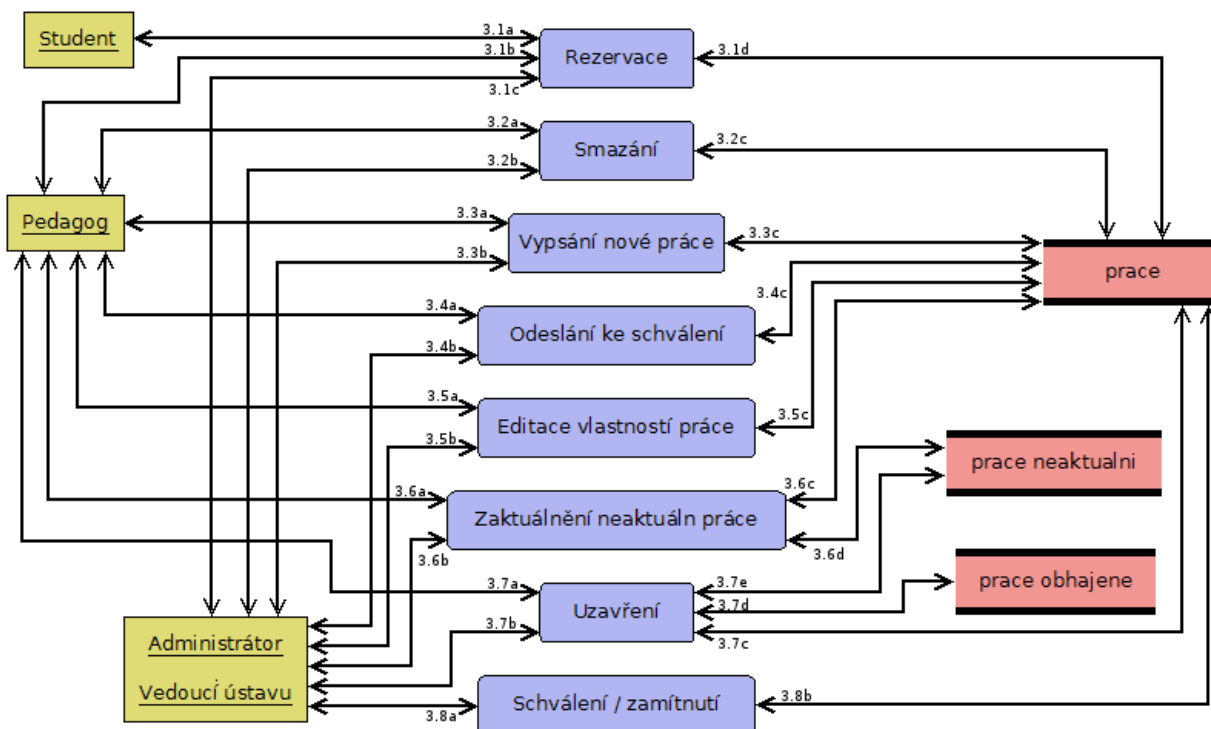
- **2.3a:** požadavek administrátora, vedoucího ústavu, pedagoga, studenta nebo nepřihlášeného uživatele na vyhledávání; vyhledávání
- **2.3b:** požadavek na vyhledávání v databázi; zobrazení výsledku vyhledávání v databázi

- **2.3c:** požadavek na informací z WS STAG pro vyhledávání; zobrazení informací z WS STAG pro vyhledávání

**Proces 2.4 – Tisk:**

- **2.4a:** požadavek administrátora, vedoucího ústavu, pedagoga, studenta nebo nepřihlášeného uživatele na tisk; zobrazení tiskové sestavy
- **2.4b:** požadavek na tisk informací z databáze; zobrazení informací z databáze pro tiskovou sestavu
- **2.4c:** požadavek na informací z WS STAG pro tisk; zobrazení informací z WS STAG pro tiskovou sestavu

## Příloha C – Diagram datových toků – editace prací



Obrázek 16: Diagram datových toků – editace údajů o práci

Zdroj: vlastní

### Slovní popis významu a obsahu znázorněných datových toků na obrázku 16

#### Proces 3.1 – Rezervace:

- **3.1a:** požadavek studenta na rezervaci práce; potvrzení rezervace práce studentem
- **3.1b:** požadavek pedagoga na rezervaci práce; potvrzení rezervace pro pedagoga
- **3.1c:** požadavek vedoucího ústavu nebo administrátora na rezervaci práce; potvrzení rezervace pro vedoucího ústavu nebo administrátora
- **3.1d:** požadavek na editaci rezervace v tabulce "práce"; editace rezervace v tabulce "práce"

#### Proces 3.2 – Smazání:

- **3.2a:** požadavek pedagoga na smazání práce; potvrzení smazání pro pedagoga
- **3.2b:** požadavek vedoucího ústavu nebo administrátora na smazání práce; potvrzení smazání pro vedoucího ústavu nebo administrátora

- **3.2c:** požadavek na smazání práce v tabulce "prace"; smazání práce v tabulce "prace"

### **Proces 3.3 – Vypsání nové práce:**

- **3.3a:** odeslání údajů o nové práci pedagogem; potvrzení přijetí údajů o nové práci od pedagoga
- **3.3b:** odeslání údajů o nové práci vedoucím ústavu nebo administrátorem; potvrzení přijetí údajů o nové práci od vedoucího ústavu nebo administrátora
- **3.3c:** odeslání údajů o nové práci do tabulky "prace"; potvrzení o vytvoření nové práce v tabulce "prace"

### **Proces 3.4 – Odeslání ke schválení:**

- **3.4a:** odeslání práce pedagogem ke schválení; potvrzení odeslání ke schválení pro pedagoga
- **3.4b:** odeslání práce vedoucím ústavu nebo administrátorem ke schválení; potvrzení odeslání ke schválení pro vedoucího ústavu nebo administrátora
- **3.4c:** požadavek na odeslání práce ke schválení; potvrzení odeslání práce ke schválení

### **Proces 3.5 – Editace vlastností práce:**

- **3.5a:** požadavek pedagoga na editaci vlastností práce; potvrzení editace vlastností práce pro pedagoga
- **3.5b:** požadavek vedoucího ústavu nebo administrátora na editaci vlastností práce; potvrzení editace vlastností práce pro vedoucího ústavu nebo administrátora
- **3.5c:** požadavek na editaci tabulky "prace"; potvrzení editace vlastností práce v tabulce "prace"

### **Proces 3.6 – Zaktuálnění neaktuální práce:**

- **3.6a:** požadavek na zaktuálnění práce od pedagoga; potvrzení zaktuálnění práce pro pedagoga
- **3.6b:** požadavek na zaktuálnění práce od vedoucího ústavu nebo administrátora; potvrzení zaktuálnění práce pro vedoucího ústavu nebo administrátora
- **3.6c:** požadavek na zaktuálnění práce v tabulce "prace; potvrzení zaktuálnění práce z tabulky "prace"



- **3.6d:** požadavek na smazání práce z tabulky "prace\_neaktualni"; potvrzení smazání práce z tabulky "prace\_neaktualni"

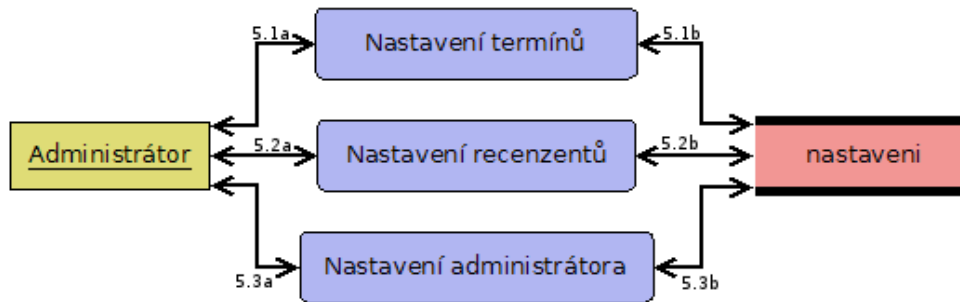
#### **Proces 3.7 – Uzavření:**

- **3.7a:** požadavek na uzavření práce od pedagoga; potvrzení uzavření práce pro pedagoga
- **3.7b:** požadavek na uzavření práce od vedoucího ústavu nebo administrátora; potvrzení uzavření práce pro vedoucího ústavu nebo administrátora
- **3.7c:** požadavek na uzavření práce v tabulce "prace; potvrzení uzavření práce z tabulky "prace"
- **3.7d:** požadavek na vytvoření práce v tabulce "prace\_neaktualni"; potvrzení vytvoření práce v tabulce "prace\_neaktualni"
- **3.7e:** požadavek na vytvoření práce v tabulce "prace\_obhajene"; potvrzení vytvoření práce v tabulce "prace\_obhajene"

#### **Proces 3.8 – Schválení / zamítnutí:**

- **3.8a:** požadavek vedoucího ústavu nebo administrátora na editaci schválení; potvrzení editace schválení pro vedoucího ústavu nebo administrátora
- **3.8b:** požadavek na editaci schválení v tabulce "prace"; potvrzení editace schválení v tabulce "prace"

## Příloha D – Diagram datových toků – editace nastavení



Obrázek 17: Diagram datových toků – editace nastavení

*Zdroj: vlastní*

### Slovní popis významu a obsahu znázorněných datových toků na obrázku 17

#### Proces 5.1 – Nastavení termínů:

- **5.1a:** požadavek administrátora na nastavení termínů; potvrzení nastavení termínů
- **5.1b:** požadavek na nastavení termínů v tabulce "nastavení"; potvrzení nastavení termínů v tabulce "nastavení"

#### Proces 5.2 – Nastavení recenzentů:

- **5.2a:** požadavek administrátora na nastavení recenzentů; potvrzení nastavení recenzentů
- **5.2b:** požadavek na nastavení recenzentů v tabulce "nastavení"; potvrzení nastavení recenzentů v tabulce "nastavení"

#### Proces 5.3 – Nastavení administrátora:

- **5.3a:** požadavek administrátora na nastavení administrátora; potvrzení nastavení administrátora
- **5.3b:** požadavek na nastavení administrátora v tabulce "nastavení"; potvrzení nastavení administrátora v tabulce "nastavení"

## Příloha E – Entity entitně relačního diagramu

Tabulka 1: Entita prace

*Zdroj: vlastní*

Typ entity	Primární klíč (podtržený) a atributy entity	Příklad výskytu entity
prace	<u>id_prace</u> typ_prace nazevc nazeven popis stav datum_odevzdani druh datum_vlozeni zadavatel_netid student_netid osobni_c status poznamka	23 bp Systém pro evidenci bakalářských a diplomových prací System for Evidence of Bachelor and Diploma Thesis Cílem práce je navrhnout a implementovat systém, který bude užitečný jak pro studenty, tak pro pedagogy. Volná 2011 Čekající 2010-02-15 20:36:12 urno1234 Aktuální

Tabulka 2: Entita prace\_neaktualni

Zdroj: vlastní

Typ entity	Primární klíč (podtržený) a atributy entity	Příklad výskytu entity
prace_neaktualni	<u>id_neaktualni_prace</u> id_prace typ_prace nazevcs nazeven popis datum_odevzdani druh datum_vlozeni datum_vlozeni_neaktualni zadavatel_netid student_cejmeno status poznamka	4 20 dp Informační systém veřejné správy Information System of Public Service Analýza současného stavu, návrh a implementace sys- tému veřejné správy 2011 Schválená 2010-03-30 11:41:10 2010-04-12 18:15:35 jali1234 Lada Pěničková Neaktuální

Tabulka 3: Entita prace\_obhajene

Zdroj: vlastní

Typ entity	Primární klíč (podtržený) a atributy entity	Příklad výskytu entity
prace_obhajene	<u>id_obhajene_prace</u> id_prace typ_prace nazevcs nazeven popis  datum_odevzdani datum_vlozeni datum_vlozeni_obhajene zadavatel_netid zadavatel_jmeno zadavatel_prijmeni zadavatel_titul_pred zadavatel_titul_za student_jmeno student_prijmeni student_titul_pred student_titul_za status poznamka	8 13 dp Informační systém Information System Analýza, návrh a implementace 2011 2010-03-30 11:41:10 2010-06-25 14:22:33 adbe1234 Adam Beran Ing. Ph.D. Jiřina Nováková Bc.  Obhájená Znamka: Výborně

Tabulka 4: Entita uzivatele

Zdroj: vlastní

Typ entity	Primární klíč (podtržený) a atributy entity	Příklad výskytu entity
uzivatele	<u>netid</u> heslo id_opravneni	luho1234 b02079c1ada232d3169a7b12b052fdbf 3

Tabulka 5: Entita typ\_prace

*Zdroj: vlastní*

Typ entity	Primární klíč (podtržený) a atributy entity	Příklad výskytu entity
typ_prace	<u>id_typu</u> typ	BP Bakalářská práce

Tabulka 6: Entita nastaveni

*Zdroj: vlastní*

Typ entity	Primární klíč (podtržený) a atributy entity	Příklad výskytu entity
nastaveni	<u>id_nastaveni</u> nazev hodnota	1 Vedoucí abcd1234

## **Příloha F – Obsah CD**

**/bp/LaTeX/** — adresář s bakalářskou prací v  $\text{\LaTeX}$ .

**/bp/bp.pdf** — bakalářská práce ve formátu PDF.

**/SEP/** — adresář se zdrojovými soubory SEP.

**/SEP/dompdf\_0-6-0\_beta1/** — adresář s knihovnou a fonty DOMPDF.