

Univerzita Pardubice  
Fakulta ekonomicko – správní

System pro evidenci bakalářských a diplomových prací

Martin Tulačka

Bakalářská práce  
2009

Univerzita Pardubice  
Fakulta ekonomicko-správní  
Ústav systémového inženýrství a informatiky  
Akademický rok: 2008/2009

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin TULAČKA**  
Studijní program: **B6209 Systémové inženýrství a informatika**  
Studijní obor: **Informatika ve veřejné správě**

Název tématu: **Návrh a implementace systému pro evidenci závěrečných prací**

### Zásady pro vypracování:

Formulace požadavků na systém pro evidenci závěrečných prací. Navržení systému pro evidenci závěrečných prací. Implementace systému pro evidenci závěrečných prací.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:


HERNANDEZ, Michael J. Návrh databází. 1. vyd. Praha: Grada, 2005. 408 s. ISBN 80-247-0900-7

KOSEK, Jiří. PHP - tvorba interaktivních internetových aplikací PHP - tvorba interaktivních internetových aplikací. 1. vyd. Praha: Grada, 1999. 490 s. ISBN 80-7169-373-1

ŘEPA, Václav. Analýza a návrh informačních systémů. 1. vyd. Praha: Ekopress, 1999. 403 s. ISBN 80-86119-13-0

WILLIAMS, Hugh E. Web database applications with PHP and MySQL. 1st edition. Sebastopol: O'Reilly, 2002. 563 s. ISBN 0-596-00041-3

Vedoucí bakalářské práce:

  
Ing. Miloslav Hub, Ph.D.  
Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce:

6. října 2008

Termín odevzdání bakalářské práce:

1. května 2009

  
doc. Ing. Renata Myšková, Ph.D.  
děkanka

I.S.

  
doc. Ing. Jiří Křepka, Ph.D.  
vedoucí ústavu

V Pardubicích dne 6. října 2008

Prohlašuji:

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

Byl jsem seznámen 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 24.08.2009

Martin Tulačka

## **Poděkování**

Na tomto místě bych chtěl poděkovat Ing. Miloslavu Hubovi, Ph.D. za poskytnuté rady, připomínky a pomoc v průběhu tvorby bakalářské práce. Dále bych chtěl poděkovat rodičům za poskytnutí možnosti studovat a všem, co pomohli s vypracováním bakalářské práce.

## **Souhrn**

Tato práce na téma „System pro evidenci bakalářských a diplomových prací“ se v úvodní části zabývá postupem vypisování bakalářských a diplomových prací na Univerzitě Pardubice. Hlavní část práce je věnována vývoji objektově orientované aplikace, podle metodiky unifikovaný proces vývoje aplikací. Pozornost je věnována všem etapám životního cyklu aplikace . V práci je dále okrajově zmíněn systém Moodle.

## **Klíčová slova:**

Moodle, php, vývoj systému

## **TITLE**

System for filing bachelor works and diploma thesis

## **ABSTRACT**

The thesis "System for filing bachelor works and diploma thesis" is focused on bachelor and masters theses submitting at the University of Pardubice in the introduction. The main part addresses build up of object oriented application in accordance to the unified application development process. Emphasis is put on all phases of the application life cycle. The Moodle system is briefly mentioned in the thesis.

## **KEYWORDS**

Moodle ,php, system development

## Obsah

Úvod .....	7
1 Současná situace vypisování témat prací na Fakultě Ekonomicko-správní Univerzity Pardubice .....	8
2 Požadavky na systém .....	11
2.1 Metody zjišťování požadavků .....	11
2.2 Funkční požadavky .....	12
2.3 Nefunkční požadavky .....	14
3 Analytické modelování .....	15
3.1 Použité diagramy .....	15
3.1.1 Use case diagram – diagram případu užití .....	15
3.1.2 Sekvenční diagram .....	16
3.1.3 Class diagram .....	16
3.2 Vybrané modely případu užití .....	17
3.2.1 Přihlášení uživatele do systému .....	17
3.2.2 Vedoucí práce .....	19
3.2.3 Student .....	24
3.2.4 Vedoucí ústavu .....	26
3.2.5 Administrátor – požadavky .....	28
3.3 Class diagram – model tříd .....	28
4 Návrh .....	30
4.1 Class diagram .....	30
4.2 Systém Moodle .....	31
4.2.1 Charakteristika systému Moodle: .....	32
4.2.2 Oprávnění a role v systému Moodle .....	32
4.3 ER diagram – entitně relační diagram .....	33
5 Implementace .....	35
5.1 Databáze .....	35
5.2 Technologie .....	35
5.3 Uživatelské rozhraní .....	35
Závěr .....	37
Literatura .....	38
Seznam zkratk .....	39
Seznam příloh .....	42

## Úvod

Vývoj informačního systému je náročná činnost. Mnoho projektů skončí dříve, nežli začne, jenom z důvodu nedostatečné přípravy. Proto je důležité před zahájením vývoje systému, vypracovat tuto aplikaci „nanečisto“ na papír. Tuto etapu mají na starosti softwaroví inženýři, kteří postupně specifikovali různé metodiky a postupy, jak úspěšně vytvářet aplikace.

Jednou z těchto metodik je Unifikovaný vývoj aplikací (dále UP) . Metodika je založená na iterativním a přírůstkovém procesu. Hlavní myšlenkou metodiky je rozložení komplexního projektu na menší „podprojekty“. Kde je každý „podprojekt“ považován za iteraci. Každá iterace obsahuje následující prvky softwarového projektu – plánování, analýza, návrh a implementace .

Tato práce nemá za cíl detailně popsat metodiku UP, ale s její pomocí navrhnout vytvořit **Systém pro evidenci bakalářských a diplomových prací** .

Důvodem vytvoření tohoto systému je neexistence aplikace, která by sjednotila postup vypisování a schvalování prací na Fakultě Ekonomicko-správní Univerzity Pardubice. Která by umožňovala vybrat studentům téma jejich absolventské práce.

Struktura práce je uzpůsobena podle této metodiky. V první kapitole je popsán postup vypisování prací na Fakultě Ekonomicko-správní. Další kapitoly jsou věnovány způsobům získávání požadavků, analytickému modelování systému, fázi návrhu systému a implementaci systému.

Pokud by se čtenář dověděl o metodice UP více informací, podrobněji se touto tematikou zabývá kniha „UML2 a unifikovaný proces vývoje aplikace“ od autora Jim Arlow.



# 1 Současná situace vypisování témat prací na Fakultě Ekonomicko-správní Univerzity Pardubice

## Postup vypisování prací

1. Vedoucí ústavu, nebo zástupce vedoucího ústavu vytvoří požadavek na počet témat, která budou vypsána pro konkrétní ústav a akademický rok. Množství témat je závislé na počtu studentů.
2. Vedoucí práce podle požadavku vytvoří návrhy témat bakalářských nebo diplomových prací (dále již jen BP , případně DP).
3. Tyto návrhy jsou předány vedoucímu ústavu, který je má za úkol vyhodnotit. V případě že jsou shledána jako vyhovující, jsou předána k dalšímu zpracování. Pokud je některé z témat nevyhovující je navracené vedoucímu práce k přepracování.
4. Z témat která byla odsouhlasena vedoucím konkrétního ústavu jako vyhovující, je následně vytvořen seznam prací. Tento seznam je vyvěšen na informační nástěnce katedry a internetových stránkách UPCE.
5. Z tohoto seznamu si následně studenti 2. ročníku bakalářského, resp. 1. navazujícího ročníku vybírají 1 téma které se rozhodnou zpracovat ve své závěrečné práci.
6. Poté navštíví student vedoucího práce vybraného tématu a pokud dojde ke shodě je tento student přiřazen jako řešitel práce. Dále je nutné vytvořit zadávací list, který obsahuje údaje spojené s tématem práce (Název práce, anotace, základní literatura, vedoucí práce). Zadávací list lze vytvořit pomocí formuláře v systému STAG. Poté je list předán vedoucímu ústavu k podpisu.

Počet a typ prací, který mají za úkol vypsát jednotliví akademičtí pracovníci se liší podle jejich odborné způsobilosti. V tabulce 1 jsou shrnuty tyto informace. Jak můžete vidět „nejméně“ prací má za úkol vypsát a vést pracovník s akademickým titulem profesor. Pro akademický rok 2009/2010 to jsou 3 maximálně 4 diplomové práce. Důvodem je vytíženost tohoto pracovníka vědeckou činností.

Dalším důležitým omezením vypisování prací je termín vyvěšení témat. Podle současné vyhlášky mají být práce vyvěšeny k 1.5.2009.

Tabulka 1: Počty vypsaných prací podle stupně odbornosti (Zdroj : vlastní)

	Bakalářské práce	Diplomové práce
Asistent	10-12	X
Odborný asistent	8-9	2-3
Docent	2-3	7-8
Profesor	X	3-4

Kompletaci seznamu prací a aktualizace obsazení má na starosti tajemník ústavu. Předávání dat mezi jednotlivými aktéry není oficiálně stanovené. Nejčastěji komunikace probíhá elektronickou formou. Velkým nedostatkem tohoto postupu, je dle mého názoru, nejednotná forma a vzhled návrhů prací. Tento nedostatek zvláště komplikuje závěrečnou kompletaci prací, kdy je poměrně časově náročné tuto formu ujednotit.

### **Některá omezení a nevýhody plynoucí ze současného stavu :**

1. **Nejednotná úprava prací** – jak již bylo zmíněno tento nedostatek zpomaluje samotný proces a zvyšuje jeho časovou náročnost .
2. **Nepřehlednost** – velký počet témat úměrné počtu studentů, a z toho vyplývající počet odděleně evidovaných dokumentů s tématy a záznamy přiřazení témat studentům.
3. **Ztráta dat** – například během závěrečné kompletace témat (přehlédnutí témata a tím jeho nezveřejnění) nebo ztráta dat během samotného procesu schvalování.
4. **Omezené možnosti aktualizace a změn** – například je-li již některé téma zabráno, jedinou možností jak o tom informovat studenty je přeškrtnout (případně jinak označit) toto téma. V případě opomenutí hrozí přiřazení tématu více studentům najednou.
5. **Zvýšené režijní náklady** – např. náklady spojené s tiskem témat v průběhu fází 1 a 2 vypisování témat (viz výše).
6. **Informovanost studentů** – omezená možnost získání informací o pracích a jejich obsazení. V současné době to je možné jen na nástěnce ústavu nebo na webu UPCE. V obou případech jsou takto získané informace neaktuální, protože aktualizace seznamu prací probíhá 1x týdně.
7. **Neexistence systému**, kde by byl podchycen schvalovací proces a uceleně spravoval

potřebné údaje s tímto postupem spojené.

Tyto, a některé další důvody, které zde nejsou zahrnuty, vedly k potřebě vytvořit samostatný systém, který bude evidovat témata BP a DP. Dále zefektivní postup jejich schvalování.

Tento systém nebude mít za cíl úplně vynechat osobní komunikaci mezi jednotlivými aktéry, ale pouze ulehčit a zpřehlednit samotný postup výběru a zadávání prací.

## 2 Požadavky na systém

Zjišťování požadavků budoucího systému (případně jakékoli aplikace) patří mezi nejdůležitější počítačnické úkony ve vývoji softwaru. Pokud je tato část nedostatečně zpracovaná nebo zcela zanedbána, může být budoucí vývoj softwaru ohrožen. Protože požadavky, které nejsou specifikované během této fáze, nebudou zahrnuty v budoucím systému. Případné zjištění nových a pro systém důležitých požadavků v pozdějších fázích návrhu, může vést až k ukončení procesu vývoje softwaru. Kdy jediným řešením této situace může být zcela nový návrh systému. Požadavky na systém se podle metodiky UP dělí na *funkční* a *nefunkční* [1]. U funkčních požadavků nás zajímá co má budoucí systém umět, ale není přitom řečeno jak toho má být dosaženo. Mezi ostatní (nefunkční) požadavky patří například na jaké platformě má budoucí aplikace běžet (Unix, Win), nebo zda s ní bude moci uživatel pracovat pomocí webového rozhraní.

### 2.1 Metody zjišťování požadavků

Mezi nejčastěji využívané metody sběru požadavků patří [7] :

- **Rozhovor s klíčovými osobami** – nejčastěji zvolená metoda sběru požadavků. Do rozhovorů by měli být zahrnuti zástupci všech skupin uživatelů, nejenom zástupci managementu. Protože management nemusí mít vždy ucelené vědomosti o zjišťované oblasti. Výhodou této metody může být získání odpovědí na nevyřešené otázky. Hlavní nevýhoda spočívá v časové náročnosti.
- **Dotazníkové šetření** – během zjišťování požadavků je potenciálním uživatelům předkládán dotazník s nejdůležitějšími otázkami. Hlavní výhodou šetření je rychlé získání informací od velké skupiny lidí, kdy každý účastník šetření má poleženy stejné otázky. Nevýhodou je nízká dotazníková návratnost.
- **Pozorování** – tato technika je založená na pozorování budoucích uživatelů, případně firmy při běžných pracovních úkonech. Kdy lze zjistit odchylky pracovního postupu od formálního. Při pozorování lze odhalit i události, které by nemusely být zjištěny při použití jiných technik získávání požadavků. Nevýhodou této techniky může být změna chování sledovaných osob, které vědí že je někdo sleduje a tím získání zkresleného

obrazu o pracovních postupech ve firmě. Dále pak časová a finanční náročnost.

## 2.2 Funkční požadavky

Ke zjištění požadavků na tento systém byla vybrána metoda rozhovorů s klíčovými uživateli. Rozhodnutí vzniklo na základě nedostatku časových rezerv a faktu, že se jedná o menší projekt.

Pro nový systém byly zjištěny následující požadavky, které by měly být zajištěny :

- Možnost přiřadit jednotlivým uživatelům roli a s tím spojené práva a možnosti akcí v systému.
- Předem určené role v systému :
  - Student,
  - Vedoucí práce,
  - Vedoucí ústavu,
  - Administrátor.
- Možnost nastavit časový milník (deadline) pro tyto akce :
  - otevření možnosti výběru témat studentům,
  - uzavření možnosti výběru témat studentům,
  - doba rezervace práce.
- Požadavky na vytvořenou práci :
  - práce obsahuje název a anotaci,
  - kontakt na vedoucího práce.
- Vedoucí práce má možnost vytvářet návrhy témat.
- Tvůrce práce bude moci své návrhy upravovat nebo případně smazat, mimo následujících výjimek :
  - práci nelze již dále upravovat jestliže byla již uznaná vedoucím ústavu,
  - práci nelze smazat v případě že je obsazená studentem.
- Tvůrce práce má možnost přiřadit k práci řešitele, ale pouze v případě uznání dané práce vedoucím ústavu.
- Student může mít rezervované, případně přiřazené nejvýše 1 téma.

- Studentům se zobrazí pouze uznaná témata a ty jsou rozdělená podle oboru, pro které jsou určena.
- Studentům se zobrazí obsazené nebo rezervované práce, tyto práce si už ale nemohou zapsat.
- Vedoucí práce má možnost uznat zapsaného studenta, případně zrušit rezervaci.
- Vedoucí ústavu má možnost změnit práci z návrhu na odsouhlasenou práci.
- V případě vrácení k přepracování/zrušení práce, bude mít možnost Vedoucí ústavu připojit k návrhu důvod rozhodnutí.
- Přehledné dělení uznaných témat :
  - podle stavu obsazení – volná, rezervovaná, přiřazená studentu,
  - podle studijního oboru a vedoucího práce.
- V případě rezervace práce studentem během víkendu (soboty, neděle) je prodloužena doba rezervace o 2 resp. 1 den oproti běžné době.
- V ostatních případech doba rezervace je dána nastavením systému (viz milníky výše).
- V případě zrušení rezervace práce systém studentovi umožní vybrání jiného tématu.
- Zrušení rezervace práce může nastat pouze v těchto případech :
  - vyprší platnost rezervace,
  - Student má možnost se kdykoli odhlásit z rezervované práce, v případě již uznané práce může zrušit přiřazení pouze vedoucí konkrétní práce,
  - pokud je uživatel s rolí Vedoucí práce tvůrcem práce, má možnost zrušit přiřazení studenta kdykoli. Bez ohledu jestli je student již uznaný, nebo má práci rezervovanou.
- Systém automaticky kontroluje vypršení platnosti rezervace u obsazených (rezervovaných) prací :
  - v případě kdy tato doba vyprší, zruší rezervaci a umožní opětovnou rezervaci studentům,
  - Systém informuje vedoucího práce a studenta o zrušení rezervace práce.
- Systém bude zaznamenávat změny tzv. „logování změn“.
- Systém bude informovat vedoucí práce o obsazení práce = pošle informační email.
- Uživatelé mají možnost tisku výstupních sestav.

### **2.3 Nefunkční požadavky**

- Zajistit základní bezpečnost systému, nepřihlášený uživatel nemá žádný informace ze systému.
- Systém bude provozován v prostředí WWW.
- Systém bude napsán pomocí jazyka PHP.
- Systém bude podporovat přihlašování uživatelů pomocí LDAP protokolu.
- Ukládání dat bude zajištěno pomocí relační databáze, například MySQL.
- Systém bude uživatelsky přívětivý – tzv. „User friendly“

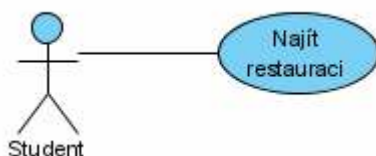
### 3 Analytické modelování

Další fází ve vývoji softwaru je analýza systému. Hlavním cílem analýzy je vytvoření analytického modelu, který ukazuje co má budoucí systém umět. Přičemž ale není určeno jakým způsobem se toho má dosáhnout. Tím se zabývá až fáze návrhu. K samotnému modelování se využívají diagramy a scénáře, které popisují daný problém a funkčnost systému.

#### 3.1 Použité diagramy

##### 3.1.1 Use case diagram – diagram případu užití

James Rumbaugh v [12] definoval Use case jako „specifikaci posloupností činností, včetně proměnných posloupností a chybových posloupností, které systém, podsystém nebo třída může vykonat prostřednictvím interakce s vnějšími aktéry“. Znamená to tedy, že případ užití je akce nebo jakýkoli jiný úkon, který uživatel očekává od systému. Každý případ užití je vždy iniciován některým aktérem. Například čas nebo systém může být považován za aktéra některého případu užití.



Obrázek 1: Grafické značení v jazyku UML (Zdroj : vlastní)

Na tomto obrázku je znázorněn grafický zápis pomocí jazyka UML. Kde se aktér značí jako postavička a případ užití je ohraničen oválem. V tomto případě máme v hlavní roli uživatele Studenta, který se snaží vyhledat v nějakém systému restaurační zařízení.

Diagram případu užití bývá ještě rozšířen o scénář (popis) případu užití. Struktura scénáře je následující :

- název případu užití,
- aktéři,



- vstupní podmínky které musí být splněny,
- hlavní úspěšný scénář,
- rozšiřující scénář.

### 3.1.2 Sekvenční diagram

Sekvenční diagram patří mezi diagramy interakce a znázorňuje interakci mezi čárami života jako časově uspořádanou posloupnost událostí. Čára života zastupuje jednoho účastníka (instanci) interakce a roli kterou v dané interakci hraje. Jednotlivé instance mezi sebou komunikují pomocí zpráv. Zpráva mezi čárami života může být například typu volání operace, tvorba instance, odstranění instance a návratová zpráva[2].

### 3.1.3 Class diagram

Dalším z významných analytických modelů je Class diagram. Před samotnou definicí co tento diagram modeluje je potřeba nejdříve vysvětlit pojmy Třída a Objekt.

- **Třída** je podle [1] „Deskriptor množiny objektů, které sdílejí stejné atributy, operace, metody, relace a chování“. Třída tedy popisuje vlastnosti , atributy a relace které charakterizují nějakou množinu objektů.
- **Objekt** je podle [1] „Diskrétní entita s jasně definovaným rozhraním, které zapouzdřuje stav a chování“. Jedná se tedy o instanci konkrétní třídy. Každá třída může mít více objektů, které se liší v attributech ale množinu operací mají vždy stejnou.

Class diagram tedy zobrazuje statický model systému, jeho třídy a vztahy mezi třídami. Tyto vazby se modelují pomocí relací a vznikají pokaždé, když je mezi třídami spojení. Vazby se dělí na asociace, agregace, kompozice a generalizace. Další důležitou částí popisující vazby mezi třídami je násobnost a průchodnost. Násobnost označuje interval objektů, které lze zahrnout do relace v určitém okamžiku. Průchodnost se značí šipkou a ta určuje směr posílání zpráv mezi třídami a zda jednotlivé třídy o sobě vědí. Průchodnost může být jednosměrná, obousměrná nebo úplně neprůchodná.

**Vymezení rolí v kterých mohou uživatelé vystupovat vůči systému a jejich požadavky:**

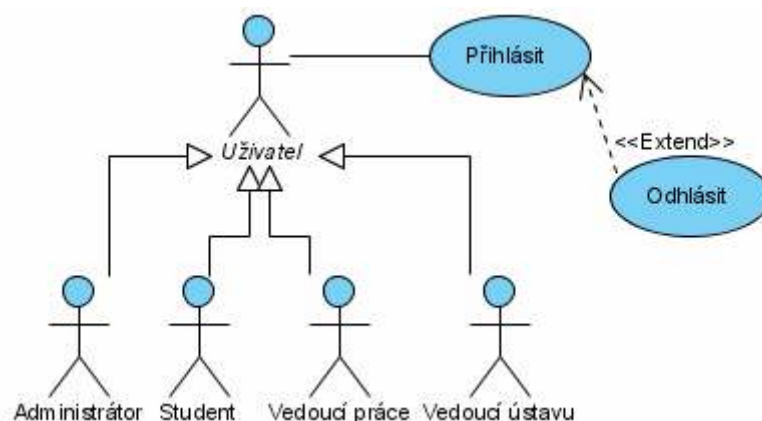
- **Administrátor** – jeho hlavním požadavkem je přehledně zpracované administrátorské rozhraní a nenáročnost systému na dodatečné nastavení jeho funkcí.
- **Vedoucí ústavu** - očekává přehlednost, jednoduchost a základní editační možnosti. Uživatele zajímají návrhy témat určené k uznání. Dále různé přehledy , jako seznam témat rozdělené podle jednotlivých vedoucích práce a oborů, obsazení prací atd.
- **Vedoucí práce** – očekává přehlednost, jednoduchost a základní editační možnosti. Pak dále přehledný soupis jím vytvořených prací, kde bude viditelná obsazenost prací studenty a zda již byla práce uznána Vedoucím ústavu.
- **Student** – tento uživatel od systému očekává přehlednost, jednoduchost a na pohled rozlišitelná témata z kterých může vybírat a u každé práce kontakt na vedoucího práce.

#### Hlavní cíle jednotlivých uživatelů :

- **Administrátor** – nenáročná správa celého systému
- **Vedoucí ústavu** – vybrat a odsouhlasit téma vhodné k vypracování studenty
- **Vedoucí práce** – vložit do systému návrhy témat, která si mohou po odsouhlasení vedoucím ústavu zapsat studenti a dále tyto studenty odsouhlasit jako řešitele.
- **Student** – jeho cílem je vybrat si z nabídky prací téma a zapsat si některou.

## 3.2 Vybrané modely případu užití

### 3.2.1 Přihlášení uživatele do systému



Obr. 1: Přihlášení uživatele diagram případu užití (Zdroj:vlastní)

Na prvním diagramu případu užití je zachycena společná část pro všechny uživatele. Jedná se o přihlášení a odhlášení uživatele do systému. Tento diagram lze popsat následujícím scénářem.

**Primární aktér:**

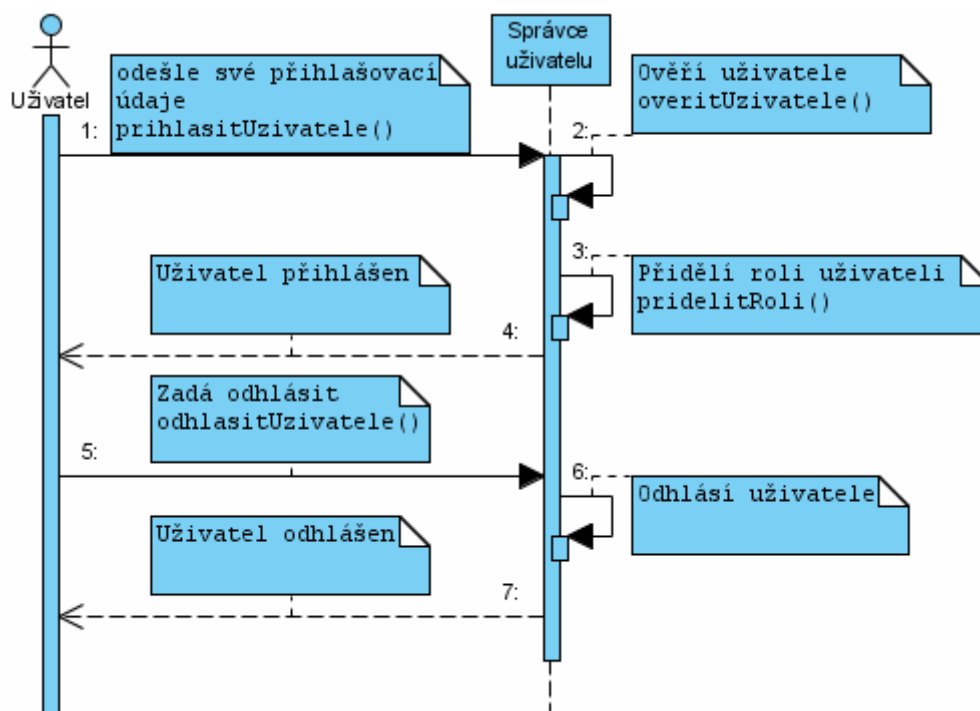
- Vedoucí práce,
- Vedoucí ústavu,
- Student,
- Administrátor.

**Hlavní úspěšný scénář**

1. Systém – zobrazí úvodní stranu kde může uživatel zadat své ověřovací údaje.
2. Uživatel – zadá potřebné údaje a potvrdí.
3. Systém – ověří údaje v centrální správě uživatelů.
4. Systém – přidělí roli uživateli.
5. Systém – umožní přístup do systému a přidělí akce podle role kterou uživatel zastává.
6. Uživatel – po ukončení všech prací uživatel zvolí volbu odhlásit.
7. Systém – odhlásí uživatele.

**Rozšiřující scénář**

- 3 Systém – nepodařené ověření údajů.
  - 3.1 uživatel nenalezen.
  - 3.2 chyba databáze, neplatné údaje.
- 4 Systém – znovu zobrazí úvodní stranu.

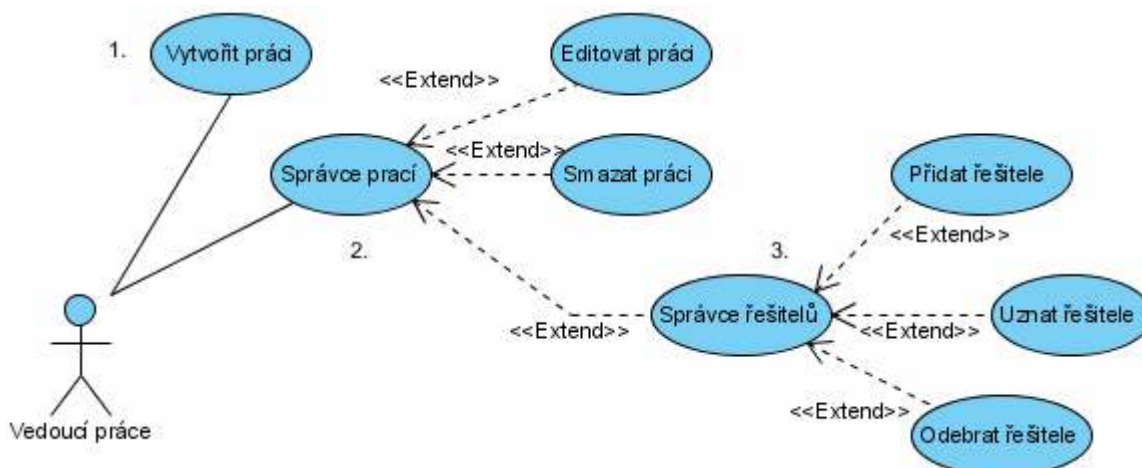


Obrázek 2: Přihlášení uživatele - sekvenční diagram (Zdroj:vlastní)

Na tomto obrázku je zachycen výše popsáný scénář. Po vyplnění a odeslání uživatelských údajů, systém nahlédne do Centrální správy uživatelů a v případě shody přidělí uživateli roli a práva s tím spojená. Pokud shoda nenastane systém požádá uživatele o opětovné vyplnění údajů. Nepřihlášení uživatele může nastat jedním z důvodů popsáných v rozšiřujícím scénáři. Pro práci v systému musí být uživatel k němu přihlášen. Případ užití končí poté co uživatel klikne na tlačítko „Odhlásit“. Systém odhlásí uživatele a zobrazí opět přihlašovací obrazovku.

### 3.2.2 Vedoucí práce

Na obrázku 4 je zachycena část požadavků uživatele Vedoucí práce na systém. Pro jednodušší a přehlednější namodelování požadavků, jsou případy užití rozděleny na 3 samostatné části. Každá z těchto částí bude popsána svým scénářem a sekvenčním diagramem. Případy užití Správce prací, Správce řešitelů jsou pouze informativním a abstraktním vyobrazením uživatelského rozhraní. Ve fázi analýzy systému není účelem tato rozhraní namodelovat, proto byla takto nahrazena.



Obrázek 3: Požadavky uživatele Vedoucí práce na systém - diagram případu užití (Zdroj:vlastní)

## 1. část

### Případ užití – Vytvořit práci

#### Primární aktér:

- Vedoucí práce.

#### Vstupní podmínky

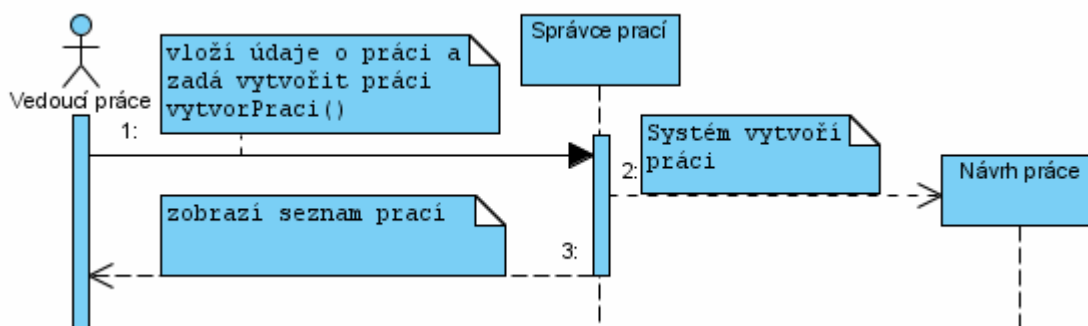
- Uživatel musí být přihlášen do systému.

#### Hlavní úspěšný scénář

- 1 Vedoucí práce – zadá příkaz Vytvořit práci.
- 2 Systém – zobrazí formulář pro vkládání návrhů témat.
- 3 Vedoucí práce – vloží do formuláře základní údaje a klikne na uložit.
- 4 Systém – uloží změny a vytvoří seznam témat určených ke schválení.
- 5 Systém – zobrazí seznam návrhu témat uživatele.

#### Rozšiřující scénář

- 4 Systém – Neuloží změny.
  - 4.1 Téma nesplňuje minimální požadavky.
  - 4.2 Chyba systému při ukládání.
  - 4.3 Databáze nenalezena.
  - 4.4 Text obsahuje nepovolené znaky.



Obrázek 4: Vytvořit práci - sekvenční diagram (Zdroj:vlastní)

Na sekvenčním diagramu je zachycen postup vypisování prací Vedoucím práce. Nejprve systém zobrazí formulář do kterého uživatel vyplní potřebné údaje. Poté zadá příkaz k uložení práce. Systém tuto práci uloží a zobrazí seznam prací uživatele.

## 2. část - Správa prací

### Případ užití – Upravit práci

#### Primární aktér:

- Vedoucí práce

#### Vstupní podmínky

- Uživatel musí být přihlášen do systému
- Uživatel má v systému vytvořenou nějakou práci

#### Hlavní úspěšný scénář

- 1 Vedoucí práce – zadá zobrazit práce
- 2 Systém – zobrazí práce uživatele
- 3 Vedoucí práce – Když upraví vybranou práci a zadá uložit změny
  - 3.1 Systém – uloží změny a zobrazí práce uživatele
- 4 Vedoucí práce – Když zadá odstranit práci
  - 4.1 Systém – odstraní práci a zobrazí práce uživatele

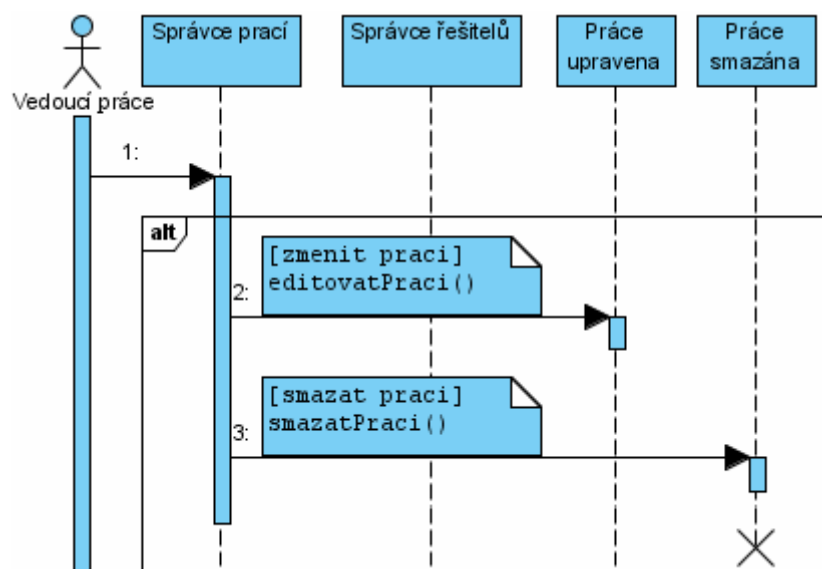
#### Rozšiřující scénář

- 2 Systém – Nezobrazí práce uživatele
  - 2.1 práce nenalezena

- 3 Systém – neuloží změny
  - 3.1 chyba databáze
  - 3.2 práce byla již uznána Vedoucím ústavu
- 4 Systém – neodstraní práci
  - 4.1 práce je obsazená studentem
  - 4.2 chyba databáze

Následující sekvenční diagram zobrazuje interakci mezi uživatelem a systémem při editaci, nebo úplném odstranění práce ze systému. Nutnými podmínkami pro možnost vykonat tyto akce je přihlášení uživatele do systému a práce, která byla vytvořena přihlášeným uživatelem.

Po přihlášení do systému uživatel vybere příkaz pro zobrazení jeho prací. Když systém najde alespoň jednu práci zobrazí jejich seznam. Uživatel se poté rozhodne jestli některou z prací odstraní, nebo upraví a uloží tyto změny.



Obrázek 5: Správce prací uživatele Vedoucí práce - sekvenční diagram (Zdroj:vlastní)

### 3. část - Správa řešitelů

Případ užití – Uznat studenta

Primární aktér:

- Vedoucí práce
- Student

### **Vstupní podmínky**

- Uživatel musí být přihlášen do systému
- Uživatel Vedoucí práce má v systému vytvořenou a uznanou práci
- Uživatel Student je zapsaný jako řešitel k práci

### **Hlavní úspěšný scénář**

- 1 Vedoucí práce – zadá zobrazit práce
- 2 Systém – zobrazí práce uživatele
- 3 Vedoucí práce – Když zadá vyhledat studenta
  - 3.1 Systém – vyhledá v centrální správě uživatelů studenty odpovídající hledanému výrazu a zobrazí jejich seznam
  - 3.2 Vedoucí práce – zadá přidělit studenta práci jako řešitele
- 4 Vedoucí práce – Když zadá uznat zapsaného studenta
  - 4.1 Systém – změní statut studenta z „neuznaný“ na „uznaný“ student
- 5 Vedoucí práce – Když zadá odebrat zapsaného studenta
  - 5.1 Systém – odebere řešitele práce a umožní opětovný výběr práce studentovi

### **Rozšiřující scénář**

- 3 Systém – nenalezne žádné studenty podle hledaného výrazu
  - 3.3 chyba databáze
  - 3.4 neplatný hledaný výraz
- 4 Systém – zobrazí opět vyhledávací formulář

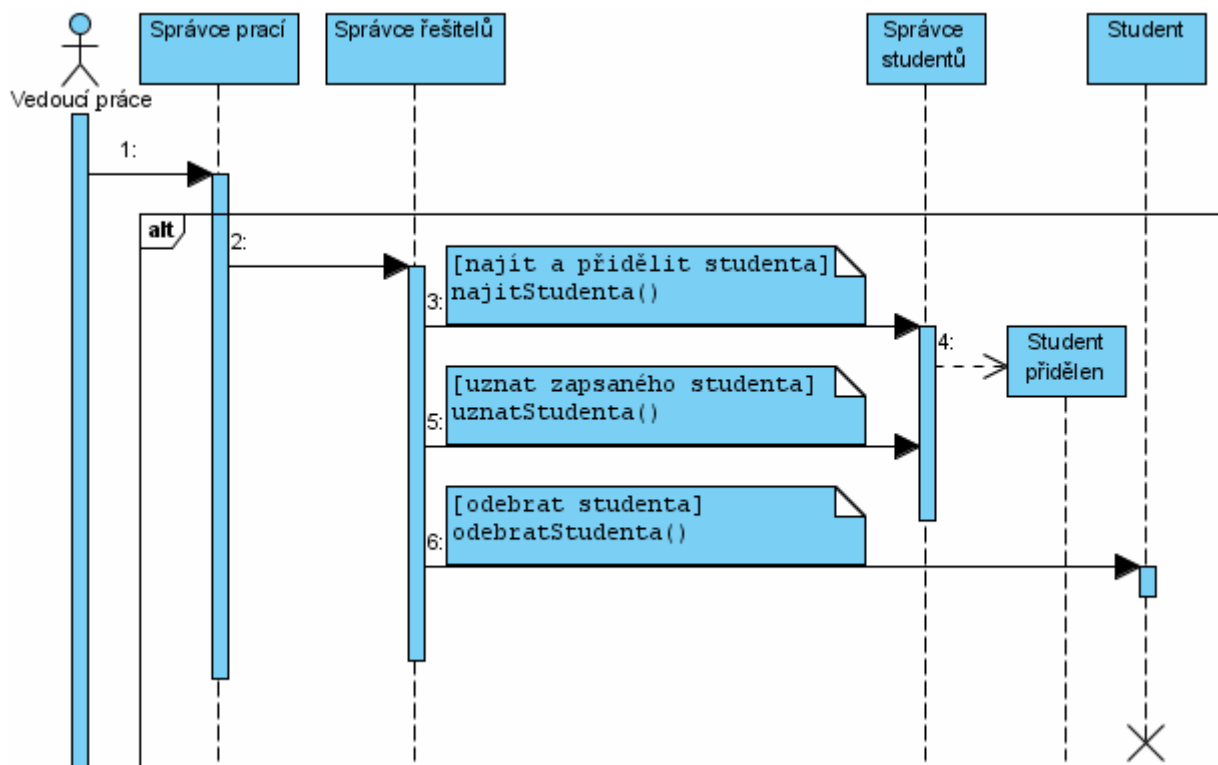
Na následujícím obrázku je namodelován sekvenční diagram který zachycuje část požadavků uživatele Vedoucí práce. Tento případ užití může nastat ve dvou případech. V prvním případě zapsáním práce některým studentem nebo rozhodnutím Vedoucího práce k této práci přiřadit studenta. V obou případech je nutné pro dokončení procesu uznat studenta jako skutečného řešitele nebo zrušit přiřazení práce.

Když chce vedoucí práce přidělit studenta práci dodatečně, musí nejprve zadat příkaz aby systém zobrazil práce které vytvořil . Poté do formuláře pro vyhledávání studentů zadá celé příjmení studenta, nebo jen jeho část a systém vyhledá v Centrální správě uživatelů osoby



odpovídající hledanému výrazu. Poté vedoucí práce zvolí zapsat určeného uživatele k práci jako řešitele.

Jestliže je k práci zapsaný student, vedoucí práce po zobrazení volby „Řešitel“ bude mít na výběr ze dvou možností. Když zvolí volbu uznat studenta, systém uzná studenta jako řešitele. Jinak systém zruší přiřazení k práci.



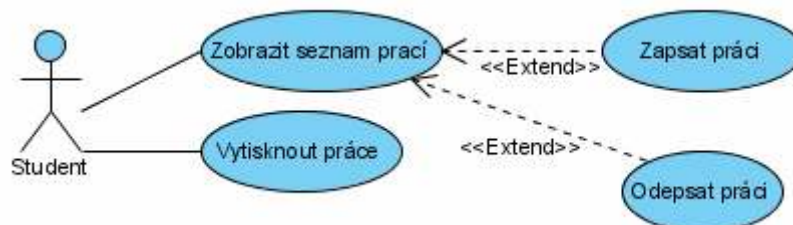
Obrázek 6: Správce řešitelů uživatele Vedoucí práce - sekvenční diagram (Zdroj:vlastní)

### 3.2.3 Student

#### Případ užití - Zapsání a odepsání práce

Tento případ užití začíná v okamžiku přihlášení uživatele v roli Student do systému. Po zobrazení nabídky příkazů uživatel vybere volbu Zobrazit seznam prací. Jestliže systém najde požadované práce zobrazí jejich seznam. Z tohoto seznamu si uživatel vybere práci a zadá volbu Zapsat. Pokud systém úspěšně zapíše k práci studenta jako řešitele, neumožní studentovi se přihlásit k jiné práci dokud se z této práce neodepíše volbou Odepsat. Poslední

příkaz který má možnost uživatel zvolit je Vytisknout práce. Pokud uživatel zvolí tento příkaz, systém se zeptá zda má vytisknout jednu konkrétní práci nebo všechny určené pro daný obor. Podle této volby připraví výstupní sestavu vhodnou pro tisk.



Obrázek 7: Požadavky uživatele Student - diagram případu užití (Zdroj:vlastní)

#### Primární aktér:

- Student

#### Vstupní podmínky

- Uživatel musí být přihlášení do systému
- Student nesmí mít zapsanou žádnou práci
- V systému musí být alespoň 1 uznaná práce Vedoucím ústavu

#### Hlavní úspěšný scénář

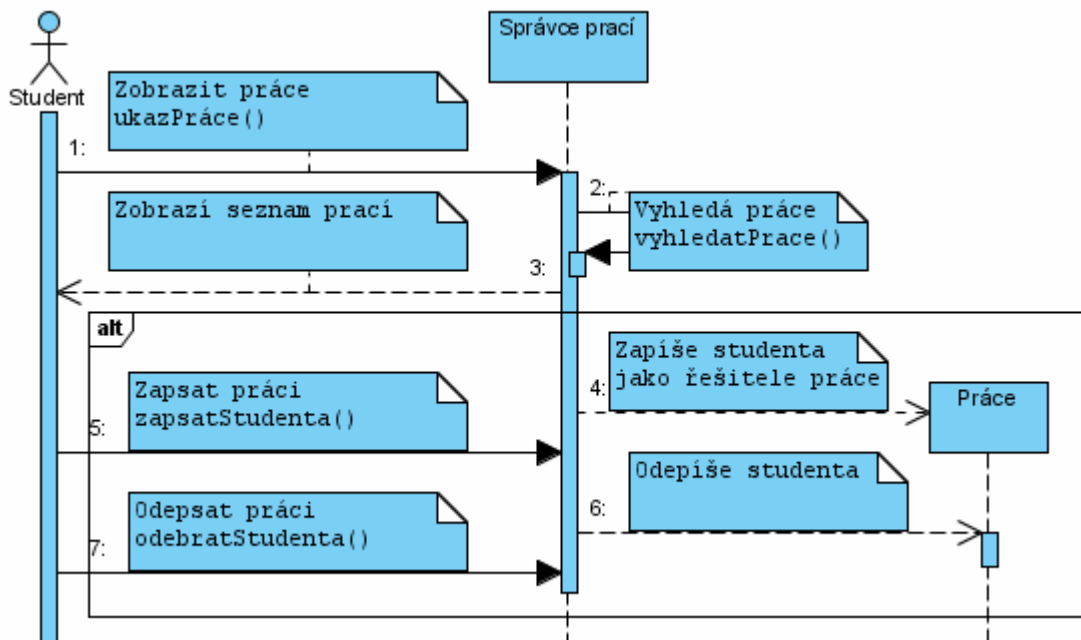
- 1 Student – zadá zobrazit práce
- 2 Systém – vyhledá práce a zobrazí jejich seznam
- 3 Student – Když vybere volbu zapsat
  - 3.1 Systém – zapíše studenta jako řešitele práce a práci nastaví jako rezervovaná práce
- 4 Student – Když vybere volbu odepsat
  - 4.1 Systém – odebere studenta jako řešitele práce a práci nastaví jako volná práce

#### Rozšiřující scénář

- 3 Zapsání neproběhlo
  - 3.3 Student má zapsanou jinou práci => povolena pouze 1 práce
  - 3.4 Práce byla obsazená jiným studentem
  - 3.5 Chyba databáze

3.6 Termín zápisu studentů ještě nenastal, nebo byl překročen

4 chyba databáze

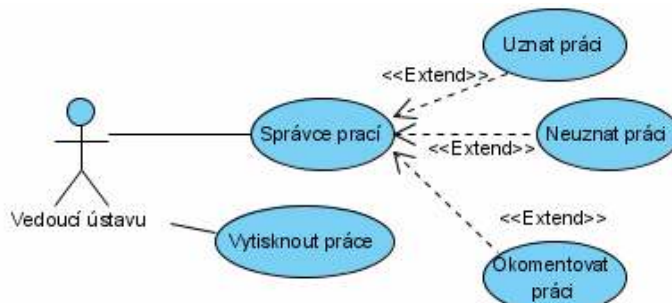


Obrázek 8: Výběr práce studentem – sekvenční diagram (Zdroj:vlastní)

### 3.2.4 Vedoucí ústavu

Jak již bylo řečeno, hlavním cílem tohoto uživatele je uznat návrh práce a umožnit tak vybrání této práce studentem. S tímto úkolem by měl uživateli napomocť správce prací, který přehledně zobrazí práce určené k uznání. Tuto funkcionalitu popisuje následující případ užití.

### 4. Případ užití – Uznat práci



Obrázek 9: Požadavky uživ. Vedoucí ústavu- diagram případu užití (Zdroj:vlastní)

Primární aktér:

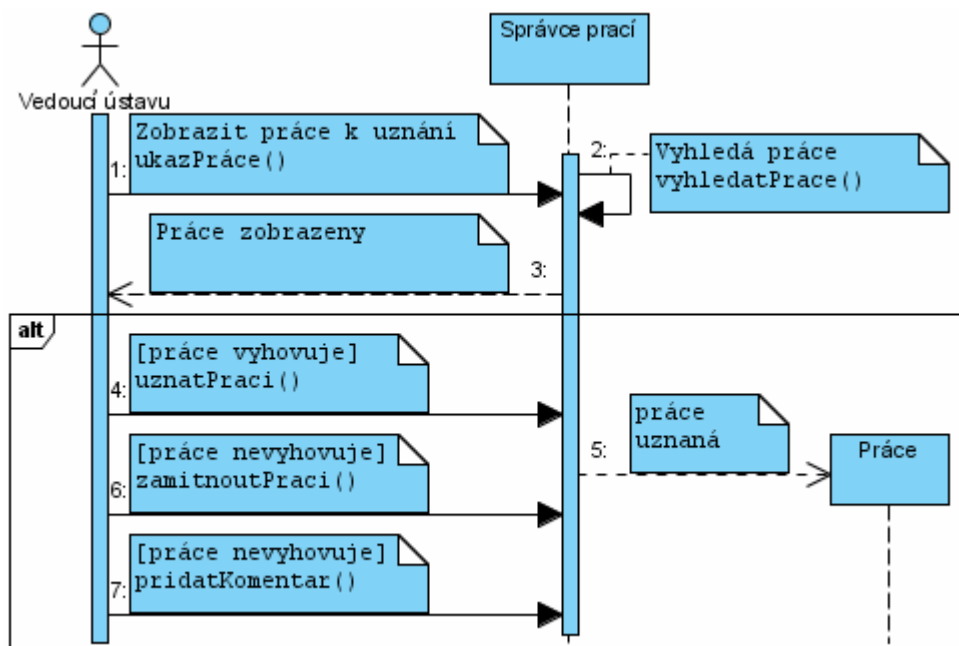
- Vedoucí ústavu

### Vstupní podmínky

- Uživatel musí být přihlášení do systému
- V systému musí být alespoň 1 práce určená k uznání Vedoucím ústavu

### Hlavní úspěšný scénář

- 1 Vedoucí ústavu – zadá zobrazit práce
- 2 Systém – vyhledá a zobrazí práce
- 3 Vedoucí ústavu – Když vybere uznat práci
  - 3.1 Systém – zobrazí formulář kde je možnost připojit k práci komentář
  - 3.2 Systém – změní práci z „neodsouhlasené“ na „odsouhlasenou“ a uloží změny
- 4 Vedoucí ústavu – Když vybere zamítnout práci
  - 4.1 Systém – zobrazí formulář kde je možnost připojit k práci komentář



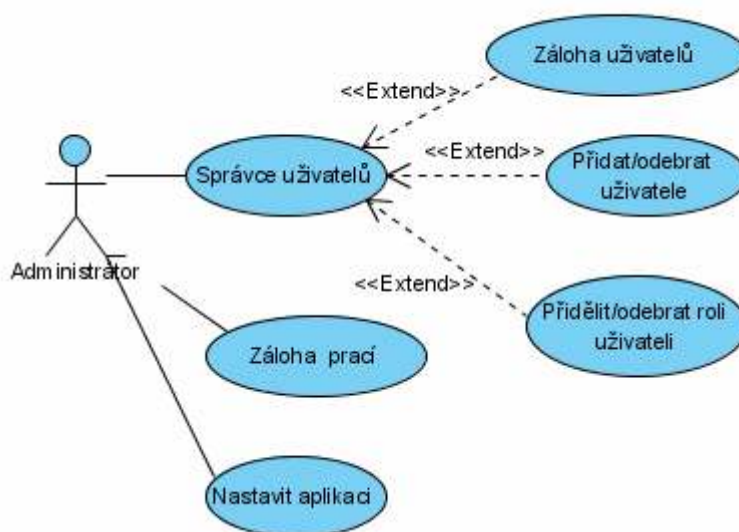
Obrázek 10: Uznání práce uživatelem Vedoucí ústavu (Zdroj:vlastní)

### 3.2.5 Administrátor – požadavky

Posledním uživatelem, který bude se systémem pracovat je Administrátor. Od systému

požaduje jednoduchou a přehlednou údržbu. K tomuto účelu by měly sloužit následující případy užití :

- *Správce uživatelů* – tento případ užití v sobě shrnuje jednotlivé úkony potřebné ke správě. Například vytvoření uživatele a následné přidělení role.
- *Nastavit aplikaci*
- *Záloha prací* – případ užití popisuje funkci zálohování prací, tedy jejich export a import



Obrázek 11: Rozhraní uživatele v roli Administrátor  
(Zdroj:vlastní)

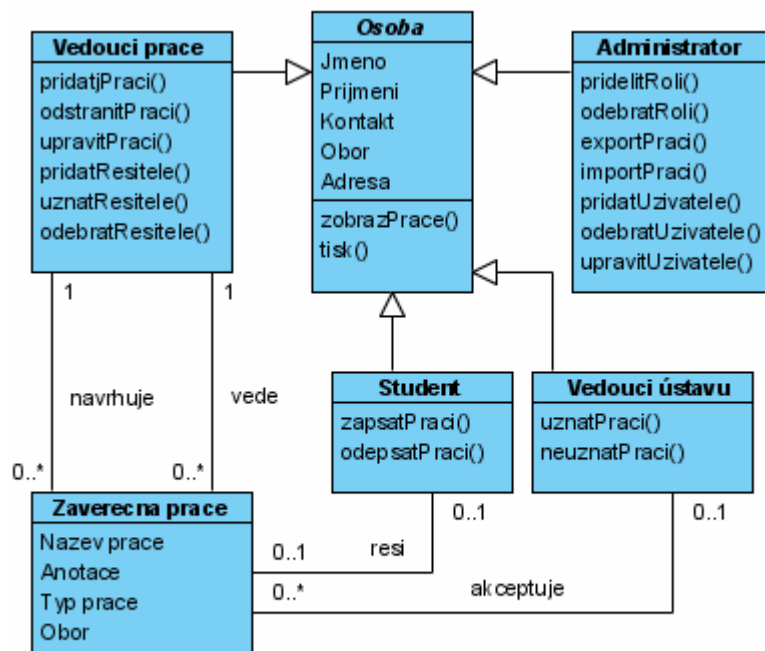
### 3.3 Class diagram – model tříd

Na obrázku 13 můžete vidět model analytických tříd, které byly určeny podle požadavků na systém. Model je zatím velmi zjednodušený a nejspíš neobsahuje třídy, které mohou vzniknout v dalších iteracích vývoje. V této iteraci je pouze naznačeno, jaké mají mezi sebou třídy interakce, ale zatím není řečeno jak se toho má dosáhnout. Násobnosti mezi třídami byly určeny podle požadavků na systém. Třída Osoba je abstraktní a vznikla generalizací následujících tříd Administrátor, Vedoucí ústavu, Vedoucí práce a Student. Třída Osoba v sobě shrnuje atributy a operace které jsou společné pro její podtřídy. Tyto podtřídy zastupují jednotlivé uživatele, resp. role které mohou mít přiděleny. Jednotlivé třídy obsahují operace které budou pro jejich práci nejdůležitější. Například mezi operace třídy Student patří

odepsatPraci(), zapsatPraci() a dále operace společné pro všechny třídy. Jsou to operace generalizované třídy Osoba – tisk() a zobrazPrace(). Poslední třída Závěrečná práce je tvořena pouze atributy Název práce, Anotace, Obor a jakého je typu.

Relace (vazby) a násobnost mezi třídami:

- Vedoucí práce může navrhnout 0 až M prací a práce má pouze jednoho tvůrce.
- Vedoucí práce může vést 0 až M prací a práce vede pouze jeden Vedoucí práce.
- Vedoucí ústavu může akceptovat 0 až M prací a Závěrečná práce může být uznána právě jedním vedoucím ústavu.
- Student může řešit právě 1 práci a práce může být přiřazena pouze jednomu studentu.



Obrázek 12: Analytický model tříd (Zdroj:vlastní)

## 4 Návrh

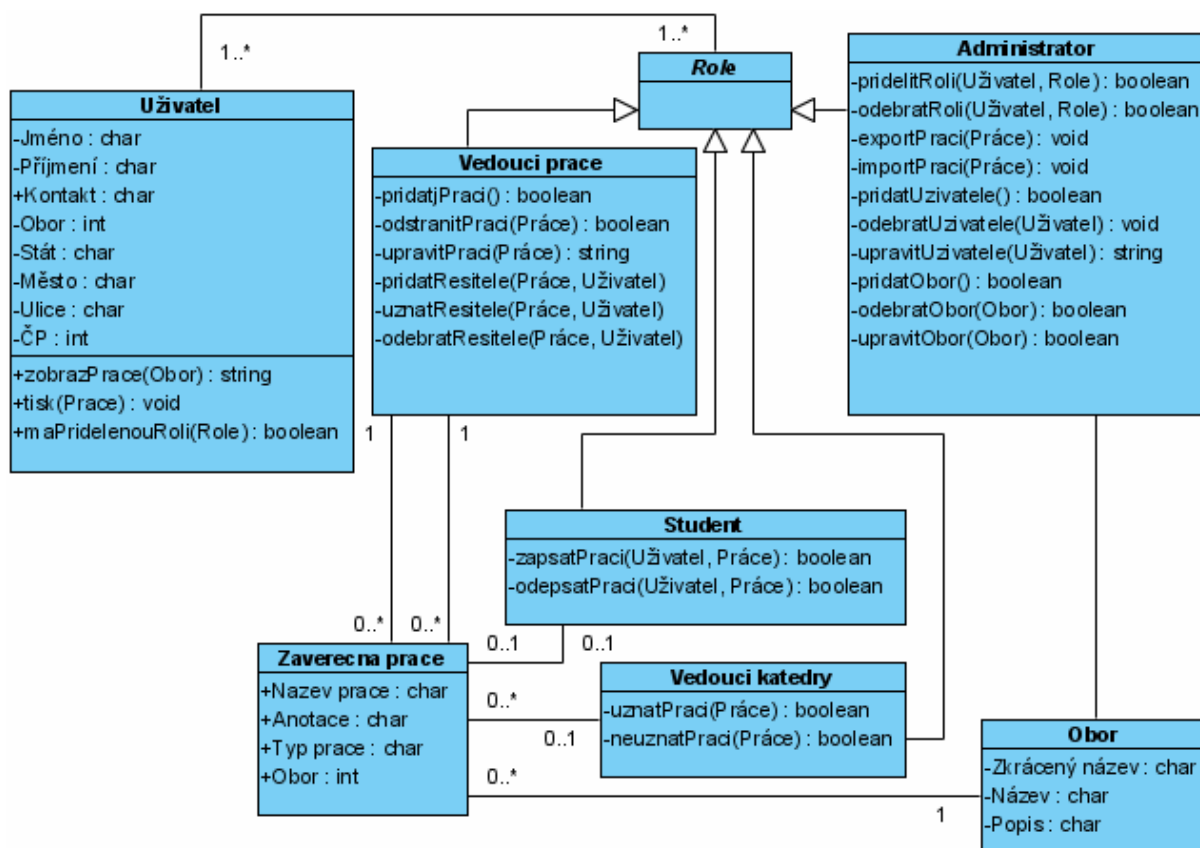
V této fázi se využívají výsledné analytické modely, kdy se upraví takovým způsobem, aby bylo možné návrhový model již implementovat. Mimo tohoto modelu je vytvořen ještě entitně relační model.

### 4.1 Class diagram

Návrhová třída se od analytické třídy liší ve stupni specifikace, kdy programátor by měl být schopen podle návrhové třídy vytvořit zdrojový kód. Sada operací z analytických tříd se během návrhu transformuje na metody. Metoda je detailně specifikovaná operace. Správně formulovaná třída by měla splňovat následující požadavky [2] :

- *úplnost* - třída by měla poskytovat služby a operace klientům které plynou již z názvu a které lze předpokládat,
- *dostatečnost* - dostatečnost určuje, zda všechny metody jsou zaměřeny na plnění účelu této třídy,
- *jednoduchost* – metody mají poskytovat nedělitelnou službu ,
- *vysoká soudržnost* – třída by měla obsahovat jen metody, které potřebuje ke splnění svého účelu,
- *minimalizace vazeb* – každá třída by měla být propojena pouze s nezbytně potřebnými třídami.

Na diagramu tříd ve fázi návrhu, lze již na první pohled rozeznat rozdíly oproti analytickému modelu. Kdy metodám (dříve operacím) byly přidány i vstupní proměnné, resp. objekty konkrétních tříd s kterými budou pracovat. K atributům byly doplněny informace o jejich typu. Dále byla vytvořena třída Obor, která vznikla oddělením od třídy Závěrečná práce. Kdy třída Závěrečné práce nesplňovala předpoklad o soudržnosti. Třída Osoba byla nahrazena třídami Uživatel a Role. Třídě Uživatel byla navíc přidána metoda maPridelenouRoli(Role), která je potřebná pro zjištění oprávnění uživatele. Třída Role v sobě spojuje typy uživatelů, které se systémem budou pracovat a jejich metody které potřebují ke splnění svého účelu.



Obrá

zek 13: Class diagram – návrh (Zdroi:vlastní)

## 4.2 Systém Moodle

Předtím než bude vysvětlen obrázek 15 je nutné popsat systém Moodle a důvod proč byl tento systém vybrán. Při rozhodování jak implementovat evidenční systém byly na výběr následující možnosti :

- Vytvořit zcela nového systému, kde bude nejobtížnějším úkolem zajistit alespoň základní bezpečnost dat a ověřování uživatelů pomocí LDAP protokolu u Centrální správy uživatelů Univerzity Pardubice.
- Vytvořit tento systém jako doplňkový balíček, nebo modul k některé již zavedené aplikaci na Univerzitě Pardubice.

Během výběru aplikace, pro kterou bude tento balíček vytvořen, jsem byl upozorněn na systém Moodle. Tento systém je na UPCE úspěšně zaveden a využíván již několik let. Proto byla vybrána druhá možnost. Dalšími výhodami tohoto systému je poměrně rozsáhlá česká



komunita, která se podílí na jeho dalším vývoji a fakt, že je tento systém distribuován pod veřejnou GNU/GPL (General Public License) licenci. Která umožňuje komukoli upravovat, distribuovat a rozvíjet tento systém bez porušení autorských práv.

#### 4.2.1 Charakteristika systému Moodle:

- je softwarový balík určený pro podporu prezenční i distanční výuky prostřednictvím online kurzů dostupných pomocí webového rozhraní.
- umožňuje, případně podporuje snadnou publikaci studijních materiálů, zakládání diskusních fór, sběr a hodnocení elektronicky odevzdávaných úkolů, tvorbu online testů a řadu dalších činností sloužících pro podporu výuky.
- je volně šiřitelný software s otevřeným kódem.
- běží pod Unix, Linux, Windows, Mac OS X, Netware a na jakémkoliv dalším systému, který podporuje PHP.
- data jsou ukládána v databázi MySQL, PostgreSQL, MS SQL nebo Oracle.[8]

#### 4.2.2 Oprávnění a role v systému Moodle

Základní rozdělení oprávnění je zajištěno přiřazením rolí jednotlivým uživatelům. Každá role má jinou sadu oprávnění a díky tomu se nemůže stát, že například student bude moci upravovat a nebo mazat údaje o jiných uživatelích. Tohoto stavu jde dosáhnout pouze nevhodným nastavením oprávnění administrátorem.

##### Role:

1. **Správce** – může manipulovat libovolně s obsahem stránek a kurzů
2. **Tvůrci kurzů** – mohou vytvářet nové kurzy a učit v nich
3. **Učitel** – učitelé mohou dělat v rámci kurzu cokoli, měnit aktivity a známkovat studenty
4. **Učitel bez práva upravovat** – uživatelé s touto rolí mohou v kurzu učita známkovat studenty, ale nemohou měnit obsah a činnosti kurzů
5. **Student** – mají v rámci kurzu omezené oprávnění
6. **Host** – mají minimální práva a nemohou systém jakkoli měnit, ani například přispívat na fóru

## 7. **Registrovaný uživatel** – je každý uživatel, který má v systému účet

Takto předdefinované role, jsou pro evidenční systém zcela dostatečné a budou se moct využít .

### 4.3 **ER diagram – entitně relační diagram**

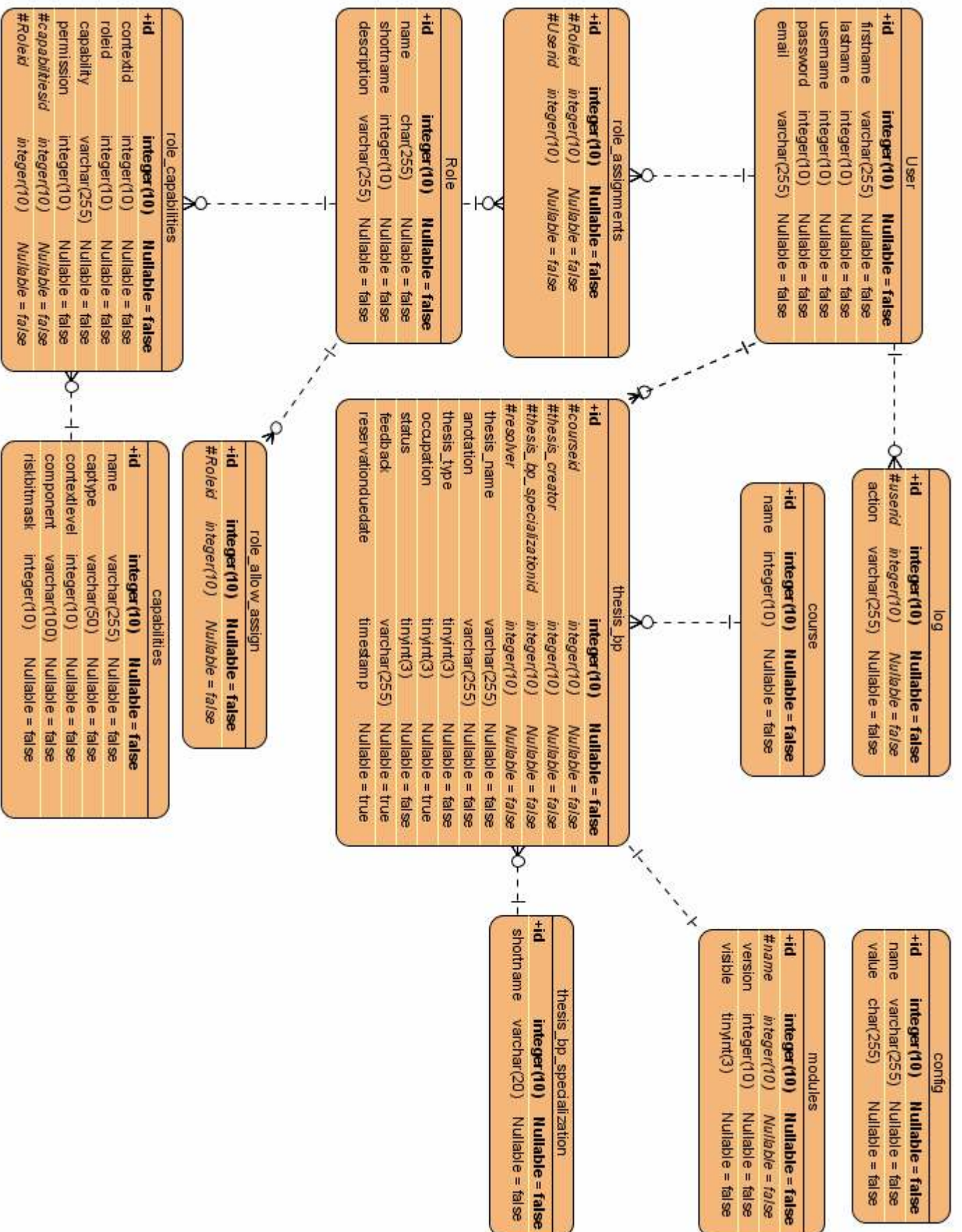
Entitně<sup>1</sup> relační diagram vznikl sloučením diagramu navrhovaného systému a diagramu systému Moodle. Moodle diagram byl získán analýzou již vytvořených tabulek v databázovém stroji MySQL. Protože nebyla k dispozici původní dokumentace systému. Entity zobrazené na obrázku nemají vypsány všechny atributy, ale jen ty nejdůležitější.

Důležité entity :

- **User** – popisuje základní informace o užívatelích. Nejdůležitějšími atributy jsou jméno, příjmení, uživatelské jméno a kontaktní email.
- **Role** – entita role popisuje základní role, které mohou být přiřazeny uživatelům.
- **Role\_capabilities** – tato entita uchovává informace o přiřazení oprávnění jednotlivým rolím.
- **Thesis\_bp** – tato entita popisuje absolventskou práci, mimo základních atributů o práci (název, anotace, typ práce) obsahuje i identifikátory (jako cizí klíče) dalších entit. Konkrétně identifikátor řešitele a tvůrce práce, v obou případech se jedná o identifikační atribut entity User. Dále pak identifikační atribut entity Thesis\_bp\_specialization.
- **Thesis\_bp\_specialization** – tato entita uchovává informace o studijních oborech.

---

1 - Entita je identifikovatelný a rozlišitelný objekt reality. Je předmětem našeho zájmu a má smysl o tomto objektu uchovávat informace [13]



Obrázek 14: Entitně relační diagram (Zdroj:vlastní)

## 5 Implementace

### 5.1 Databáze

O ukládání a správu dat se stará relační databáze MySQL. Důvodem této volby jsou následující výhody relačních databází všeobecně a výhody MySQL, mezi které je nutno počítat i distribuce pod GPL/GNU licencí. Jako Storage Engine byl zvolen InnoDB. Důvodem této volby je podpora transakcí.

**Výhody databáze MySQL [2] :**

- rychlost,
- přenositelnost,
- snadné ovládání.

**Výhody relační databáze [5] :**

- 1 **Zabudovaná víceúrovňová integrita** - integrita dat je zbudovaná přímo do modelu na úrovni položek, aby se zajistila přesnost dat.
  - *úroveň tabulek* – RD zajišťuje, že záznamy nebudou duplicitní a detekuje chybějící hodnoty primárních klíčů,
  - *úroveň vztahů* - RD zajišťuje, že vztah mezi dvojicí tabulek je platný.
- 2 **Garantovaná konzistence a přesnost dat** – data jsou konzistentní a přesná pomocí různých úrovním integrity, které jsou v databázi vynutitelné.
- 3 **Snadné získávání dat** – pomocí jednoduchých příkazů lze jednoduše získat z jedné, nebo více tabulek mezi kterými existuje vztah požadovaná data. Toto umožňuje zobrazit stejné informace mnoha způsoby.

### 5.2 Technologie – PHP

Pro napsání samotné aplikace byl vybrán jazyk PHP. PHP je skriptovací jazyk, který umožňuje psaní dynamických webových stránek které jsou interpretovány na straně serveru a uživateli je poslán samotný HTML kód. Výhodou tohoto jazyka je podpora objektově orientovaného programování, multiplatformita a distribuce pod open source. Nevýhodou je neexistence open source ladícího nástroje.

**Historie PHP** sahá až do roku 1994, kdy první verzi vytvořil Rasmus Reldorf pod názvem Personal Home Page Tools, tento skriptovací jazyk měl mít původně využití jen na jeho webových stránkách. V roce 1995 uvolnil tento nástroj jako open source pod názvem PHP/FI - Personal Home Page / Forms Interpreter . V roce 1997 pak byl uvolněn PHP/FI 2.0.

V roce 1997 vyšla verze 3.0 která se začala podobat současným verzím. Největší novinkou bylo zavedení objektivě orientované syntaxe. Změněn byl i název tohoto jazyka na PHP: Hypertext Preprocessor.

Verze 4.0 přinesla podstatné zvýšení výkonu, které umožnilo vytvářet složitější aplikace.

Verze 5.x rozšiřuje možnosti objektivě orientovaného jazyka, zavedením konstruktorů, volání odkazem atd. Dále zavádí mechanismus výjimek.

### **5.3 Uživatelské rozhraní**

Následující text se zabývá uživatelským rozhraním vytvořeného modulu. Pro lepší představu je toto rozhraní ukázáno v přílohách A-F. Všechny výstupy v přílohách byly mírně upraveny, protože se jedná o výřezy nejdůležitějších částí uživatelského rozhraní.

V příloze A je zobrazen formulář pomocí kterého může vedoucí práce přidat návrh práce do systému. Ve formuláři je potřeba vybrat z nabídky možností typ práce, pro jaký obor bude práce určena a doplnit zbývající údaje. V případě nevyplnění názvu práce, případně anotace systém neuloží tento návrh do databáze a označí chybějící pole.

Příloha B ukazuje seznam prací které vytvořil konkrétní vedoucí práce. Tento seznam lze nalézt pod záložkou „Mé práce“. Práce jsou řazeny podle typu práce, oboru a názvu. Na první pohled lze odlišit zda je práce obsazena studentem, nebo zda byla uznána vedoucím ústavu.

V další příloze je zřejmě nejdůležitější část požadavků vedoucího ústavu na systém. Jedná se o uznání návrhu práce. Základní údaje o práci jsou seříděny do tabulky. V levé části této tabulky se zobrazí fotka tvůrce práce. Pokud by tento uživatel neměl v systému vloženou fotku, je jako v tomto případě použit základní obrázek. Pod tabulkou je formulář kam může vedoucí ústavu připsat případný komentář.

Příloha D obsahuje detailní popis práce zobrazený studentům. Mimo základních

informací o práci je k popisu přiřazena fotka vedoucího práce. Pokud práce není obsazená zobrazí se tlačítko určené k zapsání/odepsání práce. Podle požadavků systém neumožní studentovi zapsat si více prací najednou. Pokud se student úspěšně zapíše, je ještě nutné, aby vedoucí této práce uznal studenta jako skutečného řešitele. Dokud tak nenastane, má student práci pouze rezervovanou.

Uznání studenta vedoucím práce jako řešitele je ukázáno v horní části přílohy E. Hlavní změnou tohoto rozhraní je potvrzující formulář pod tabulkou se souhrnem informací o práci. Pokud uživatel zvolí možnost Ano a uloží změnu, systém potvrdí studenta jako řešitele. V opačném případě bude studentu odebrána rezervace a práce se ihned zpřístupní k výběru ostatním studentům. Další změnou prošla i samotná informační tabulka. Pokud je práce obsazená studentem, vedoucímu práce se zobrazí fotka řešitele. V dolní části této přílohy je výstupní sestava prací určená k tisku, nebo exportu ve formátu PDF. Vzor této struktury byl získán na informační nástěnce Ústavu systémového inženýrství UPCE. Pro tento výstup byl využit skript autora Jan Slabon dostupný z [3].

## Závěr

První část práce byla zaměřena na popis postupu vypisování bakalářských a diplomových prací na Fakultě Ekonomicko-správní Univerzity Pardubice. Byly popsány nevýhody a omezení, které z tohoto postupu vyplývají.

Druhá kapitola se věnuje požadavkům na systém. Čtenář se zde seznámí se základními metodami získávání požadavků. O každém způsobu získávání informací se doví jejich výhody a nevýhody. Tato kapitola zaměřená na první etapu vývoje systému je velmi důležitá, protože požadavky které se zde neuvedou, nebudou do pozdějších fází vývoje zahrnuty.

Třetí kapitola je věnována analytickému modelování systému. Cílem této fáze bylo namodelovat systém jako „černou skříňku“. Kdy je řečeno o systému co bude umět, ale ještě se neví, jak toho má být dosaženo. Pro snadnější pochopení, jak má budoucí systém pracovat, se využívají diagramy. Nejvýznamnějším diagramem je Use Case diagram neboli diagram případu užití. Tento diagram se obvykle doplní o popisující slovní scénář. Dalšími významnými diagramy jsou sekvenční diagram a diagram tříd (Class diagram).

Čtvrtá kapitola je zaměřena na fázi návrhu systému. Tato fáze využívá výsledky z analytického modelování. V tomto případě byl využit diagram tříd, který byl transformován z analytického modelu do modelu návrhového. Při transformaci došlo k pozměnění některých tříd a upřesnění jejich metod. Dalším pomocným diagramem této etapy je Entitně-relační diagram, který zachycuje vztahy a propojení tabulek budoucí databáze. Poslední zmiňovanou problematikou této kapitoly je systém Moodle. Výsledkem této fáze by měl být natolik specifikovaný model systému, který lze již naprogramovat.

V páté a poslední kapitole je na několika příkladech popsána finální podoba systému, který byl vytvořen jako rozšiřující modul do systému Moodle.

Na závěr by se dalo říci, že tato práce byla splněna podle zadaných požadavků. Dále pak názorným způsobem ukazuje vývoj informačního systému ve všech vývojových etapách. Tedy sběr požadavků, analýza, návrh a závěrečná implementace.

## Literatura

- 1) ARLOW, Jim, NEUSTADT, Ila. *UML2 : a unifikovaný proces vývoje aplikací*. 2. aktualiz. vyd. Brno : Computer Press, 2008. 567 s. ISBN 978-80-251-1503-9.
- 2) DUBOIS, Paul. *Mysql : profesionálně*. 1. vyd. Brno : Mobil Media, 2003. 1071 s. ISBN 80-86593-41-X.
- 3) *FPDF : Table with multi-page columns* [online]. [2006] [cit. 2009-08-19]. Dostupný z WWW: <<http://www.fpdf.org/>>.
- 4) *Historie PHP a souvisejících projektů* [online]. nevedeno [cit. 2009-08-18]. Dostupný z WWW: <<http://jonatan.spse.pilsedu.cz/doc/php-man/history.html>>.
- 5) HERNANDEZ, Michael.J. *Návrh databází*. Praha : Grada, 2006. 408 s. ISBN 80-247-0900-7.
- 6) *Introduction to Moodle Programming* [online]. 2004 [cit. 2009-08-21]. Dostupný z WWW: <<http://dev.moodle.org/course/view.php?id=2>>.
- 7) KANISOVÁ, Hana, MÜLLER, Miroslav. *UML : srozumitelně*. 2. aktualiz. vyd. Brno : Computer Press, 2006. 176 s. ISBN 80-251--1083-4.
- 8) MOODLE.CZ [online]. nevedeno [cit. 2009-08-09]. Dostupný z WWW: <<http://moodle.cz/>>.
- 9) MOODLE.ORG [online]. nevedeno [cit. 2009-08-09]. Dostupný z WWW: <<http://docs.moodle.org/cs/Licence/>>.
- 10) OPPEL, Andrew. *Databáze : bez předchozích znalostí*. 1. vyd. Brno : Computer Press, 2006. 318 s. ISBN 80-251-1199-7.
- 11) *PHP Cross Reference of Moodle* [online]. 2004 [cit. 2009-08-21]. Dostupný z WWW: <<http://xref.moodle.org/nav.html?index.html>>.
- 12) RUMBAUGH, James, JACOBSON, Ivar, BOOCH, Grady. *The Unified Modeling Language Reference Manual*. 2nd edition. [s.l.] : Addison-Wesley, 2004. 721 s. Addison-Wesley Object Technology Series. ISBN 0321245628.
- 13) ŠIMONOVÁ, Stanislava, PANUŠ, Jan. *Databázové systémy I*. Pardubice : Univerzita Pardubice, 2007. 106 s. ISBN 80-7194-988-6.



## **Seznam zkratek**

BP – bakalářská práce

DP – diplomová práce

GPL/GNU – General Public License

LDAP - Lightweight Directory Access Protocol

MySQL -

PDF - Portable Document Format

PHP - Hypertext Preprocessor

RD – relační databáze

STAG – studijní agenda

UML – Unified Modeling Language

UP – Unified Process

UPCE – Univerzita Pardubice

## Seznam obrázků

Obrázek 1: Grafické značení v jazyku UML (Zdroj : vlastní).....	15
Obrázek 2: Přihlášení uživatele diagram případu užití (Zdroj:vlastní) .....	18
Obrázek 3: Přihlášení uživatele - sekvenční diagram (Zdroj:vlastní) .....	19
Obrázek 4: Požadavky uživatele Vedoucí práce na systém - diagram případu užití (Zdroj:vlastní).....	20
Obrázek 5: Vytvořit práci - sekvenční diagram (Zdroj:vlastní).....	21
Obrázek 6: Správce prací uživatele Vedoucí práce - sekvenční diagram (Zdroj:vlastní).....	23
Obrázek 7: Správce řešitelů uživatele Vedoucí práce - sekvenční diagram (Zdroj:vlastní).....	25
Obrázek 8: Požadavky uživatele Student - diagram případu užití (Zdroj:vlastní) .....	26
Obrázek 9: Výběr práce studentem – sekvenční diagram (Zdroj:vlastní) .....	27
Obrázek 10: Požadavky uživ. Vedoucí ústavu- diagram případu užití (Zdroj:vlastní).....	29
Obrázek 11: Uznání práce uživatelem Vedoucí ústavu (Zdroj:vlastní) .....	30
Obrázek 12: Rozhraní uživatele v roli Administrátor (Zdroj:vlastní) .....	31
Obrázek 13: Analytický model tříd (Zdroj:vlastní) .....	32
Obrázek 14: Class diagram – návrh (Zdroj:vlastní) .....	34
Obrázek 15: Entitně relační diagram (Zdroj:vlastní).....	38

## **Seznam tabulek**

Tabulka 1: Počty vypsaných prací podle stupně odbornosti (Zdroj : vlastní) .....	9
--	---

## **Seznam příloh**

Příloha A – vytvoření návrhu práce

Příloha B – přehled prací

Příloha C – uznání práce

Příloha D – informace o práci z pohledu studenta

Příloha E – potvrzení řešení, tisková sestava prací

Příloha F – instalace modulu Thesis\_bp

## Příloha A – vytvoření návrhu práce

Bakalářská práce    Diplomová práce    Přidej práci    Moje práce

### Formulář pro přidání práce

Typ práce

Obor

Název práce\*   
Anotace\*

Trebuchet    1 (8 pt)    Jazyk

Identifikace a formuace požadavku na systém.  
Návrh systému pro evidenci bakalářských a diplomových prací.  
Implementace navrženého systému pro evidenci bakalářských a diplomových prací.

Cesta: body

Uložit změny    Zrušit




Formulář obsahuje povinná pole


Příloha A – Vytvoření práce (Zdroj:vlastní)

## Příloha B – přehled prací

Bakalářská práce    Diplomová práce    Přidej práci    Moje práce

Vytiskni všechny práce


Typ práce	Obor	Název práce	Stav	Možnosti
Bakalářská	USII	System pro evidenci bakalářských a diplomových prací	Nepotvrzená práce	 X
Bakalářská	USII	<b>Obsazená práce</b> Budoucnost jazyka PHP	Potvrzená práce	 X
Bakalářská	USII	UML	Potvrzená práce	 X

 Dokumentace k této stránce

Jste přihlášení jako Ing. Miloslav Hub, Ph.D. (Odhlásit se)

[Titulní stránka](#)


## Příloha C – uznání práce

	Obor	USII
	Vedoucí práce	Ing. Miloslav Hub, Ph.D.
	Čas vytvoření	pondělí, 17. srpen 2009, 21.02
Název práce	Systém pro evidenci bakalářských a diplomových prací	
Anotace	Identifikace a formulace požadavku na systém. Návrh systému pro evidenci bakalářských a diplomových prací. Implementace navrženého systému pro evidenci bakalářských a diplomových prací.	

Potvrdit práci


Odpověď

Trebuchet 1 (8 pt) Jazyk **B** *I* U ~~S~~  $x_2$   $x^2$



Tato práce splňuje požadavky

Cesta: body




[Návrat zpět](#)

Příloha C – formulář pro uznání práce Vedoucím ústavu (Zdroj:vlastní)

## Příloha D – informace o práci z pohledu studenta

usii ► K001 ► Thesis

Vytiskni práci

	Obor	USII
	Vedoucí práce	Ing. Miloslav Hub, Ph.D.
	Stav	<input type="button" value="Odhlásit"/>
Název práce	Systém pro evidenci bakalářských a diplomových prací	
Anotace	Identifikace a formulace požadavku na systém. Návrh systému pro evidenci bakalářských a diplomových prací. Implementace navrženého systému pro evidenci bakalářských a diplomových prací.	


[Návrat zpět](#)

[Jste přihlášení jako Martin Tulačka \(Odhlásit se\)](#)



## Příloha E – potvrzení řešení, tisková sestava prací

K001 ► Thesis Vytiskni práci

	Obor	USII
	Jméno řešitele	Martin Tulačka
	Stav	Potvrzený student
Název práce	Systém pro evidenci bakalářských a diplomových prací	
Anotace	Identifikace a formulace požadavku na systém. Návrh systému pro evidenci bakalářských a diplomových prací. Implementace navrženého systému pro evidenci bakalářských a diplomových prací.	

Potvrdit  studenta

[Návrat zpět](#)

Příloha E – potvrzení studenta jako řešitele (Zdroj:vlastní)

## Thesis

Obor - USII Typ práce - Bakalářská Vedoucí práce - Ing. Miloslav Hub, Ph.D.	Budoucnost jazyka PHP historie jazyka PHP PHP verze 4.0	
Obor - USII Typ práce - Bakalářská Vedoucí práce - Ing. Miloslav Hub, Ph.D.	Systém pro evidenci bakalářských a diplomových prací Identifikace a formulace požadavku na systém. Návrh systému pro evidenci bakalářských a diplomových prací. Implementace navrženého systému pro evidenci bakalářských a diplomových prací.	Obsazená práce
Obor - USII Typ práce - Bakalářská Vedoucí práce - Ing. Miloslav Hub, Ph.D.	UML Historie Budoucnost	

Příloha E – sestava prací určená pro tisk (Zdroj:vlastní)

## Příloha F - instalace modulu

Pokud se budete při instalaci držet tohoto postupu, nemělo by dojít během instalace modulu k vážnějším problémům.

Modul byl vyvíjen a testován ve verzi Moodle 1.9.5+

Postup:

1. zkopírovat složku Thesis\_bp do „umístění instalace/moodle/mod“,
2. zkopírovat složku block\_thesis\_bp do „umístění instalace/moodle/blocks“,
3. zkopírovat obsah složky font do „umístění instalace/moodle/lib/fpdf/font“,
4. zkopírovat obsah souboru copy\_to\_adminlib.txt do souboru „umístění instalace/moodle/lib/adminlib.php“, (poznámka – v případě prohození kroků 4 a 5 dojde při instalaci modulu k chybě, protože modul vyžaduje ke své činnosti třídu kterou je nutné zkopírovat do souboru adminlib.php)
5. přihlásit se do systému Moodle jako administrátor a v navigačním bloku Správa stránek zvolit volbu informace, nyní by se měl modul úspěšně nainstalovat,
6. v záložce bloku Správa stránek/moduly/činnosti zvolit volbu nastavení, zde je potřeba nastavit následující volby a časové milníky:
  1. čas otevření registrace studentů k pracem
  2. čas uzavření registrace studentů k pracem – POZOR! Systém v tomto kroku nekontroluje zda milník uzavření registrací je větší než jejich otevření. Tato kontrola je provedena až při samotném běhu aplikace.
  3. počet dnů rezervace práce (přednastaveno 3 dny)
  4. interval mezi prohledáváním databáze kvůli uplynutí rezervace práce
7. vytvořit kurz kam se umístí navigační blok „Navigace thesis“
8. na navigačním bloku zvolit volbu Zobrazit práce (poznámka - další kroky může provést pouze uživatel s globální rolí Administrátor)
9. posledním krokem je vytvoření oborů. Obory je nutné vytvářet v oficiálních zkratkách