

Univerzita Pardubice

Fakulta ekonomicko-správní

## **Centrální správa dokumentů**

**Martina Bendová**

Bakalářská práce

2010

Univerzita Pardubice  
Fakulta ekonomicko-správní  
Ústav systémového inženýrství a informatiky  
Akademický rok: 2009/2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martina BENDO VÁ**  
Studijní program: **B6209 Systémové inženýrství a informatika**  
Studijní obor: **Informatika ve veřejné správě**  
  
Název tématu: **Centrální správa dokumentů**

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

Rozbor typů používaných dokumentů v daném subjektu.  
Určení atributů vhodných pro kategorizaci a vyhledávání.  
Vypracování návrhu databáze (ER-model), Use-case rozbor.  
Realizace a implementace.

Rozsah grafických prací:

Rozsah pracovní zprávy:

30 - 40 stran

Forma zpracování bakalářské práce:

tištěná/elektronická

Seznam odborné literatury:

WELLING, Luke, THOMSON, Laura. PHP a MySQL : rozvoj webových aplikací. Praha : SoftPress, c2004. 910 s.

STRAKA , Miroslav. Vývoj databázových aplikací. Praha : Grada, 1993. 129 s. ISBN 80-85424-43-6.

ŠKULTÉTY, Rastislav. Java Script : Programujeme internetové aplikace. 2. aktualiz. vyd. Brno : Computer Press, 2004. 224 s. ISBN 80-251-0144-4.

Vedoucí bakalářské práce:

**Ing. Oldřich Horák**

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce:

**5. října 2009**

Termín odevzdání bakalářské práce:

**30. dubna 2010**

doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.

doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 5. října 2009

## **Prohlášení autora**

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 30. 6. 2010

Martina Bendová

## **Poděkování**

Na tomto místě bych ráda poděkovala Ing. Oldřichu Horákovi za čas, který mi při torbě práce věnoval. Za cenné rady a připomínky, kterými přispěl k vypracování této bakalářské práce, a za přístup ke školnímu serveru [cipisek.upce.cz](http://cipisek.upce.cz). Dále bych ráda poděkovala ZŠ Sion Hradec Králové, za poskytnuté materiály a informace o chodu subjektu školství.

## **Souhrn**

Bakalářská práce s názvem „Centrální správa dokumentů“ je věnována návrhu a realizaci webové aplikace. V první teoretické části práce je popsán systém centrální správy dokumentů, skriptovací jazyk PHP a databázový systém MySQL. Druhá část se zabývá samotnou realizací aplikace.

## **Klíčová slova**

Relační databáze, E-R diagram, Use-Case diagram, Správa dokumentů, MySQL, PHP

## **Title**

Central Administration of Documents

## **Summary**

Bachelor thesis entitled "Central management of documents" presents how to design and implement web applications. The first theoretical part describes a system of central document management, scripting language PHP and MySQL database system. Second part deals with the implementation of applications itself.

## **Keywords**

Relational database, E-R diagram, Use-Case Diagram, Document Management, MySQL, PHP

# Obsah

Úvod a cíl práce .....	9
1 Centrální správa dokumentů.....	10
1.1 Životní cyklus dokumentu .....	10
1.2 Výhody centrální správy dokumentů.....	10
1.3 Nevýhody centrální správy dokumentů.....	11
2 PHP a MySQL.....	12
2.1 PHP.....	12
2.1.1 Proměnné v PHP .....	12
2.1.2 Funkce v PHP .....	12
2.2 MySQL.....	14
2.2.1 Použité funkce MySQL.....	14
2.2.2 phpMyAdmin .....	15
2.2.3 Jazyk SQL.....	16
3 Realizace.....	18
3.1 Požadavky na aplikaci .....	18
3.2 Rozbor typů dokumentů .....	18
3.3 Definice atributů pro kategorizaci a vyhledávání .....	19
3.4 Návrh databázového modelu .....	19
3.4.1 E-R Diagram .....	20
3.4.2 Relační databázový model.....	22
3.4.3 Výhody relační databáze .....	24
3.4.4 Use Case model .....	25
3.5 Praktická fáze tvorby systému .....	27
3.5.1 Vytvoření databáze.....	28

3.5.2	Připojení k databázi .....	30
3.5.3	Kontrola polí formuláře .....	31
3.5.4	Rozložení aplikace.....	31
3.5.5	Registrace uživatele .....	32
3.5.6	Přihlášení uživatele.....	33
3.5.7	Přidělení uživatelských práv .....	34
3.5.8	Přidání dokumentu .....	36
3.5.9	Vypsání dokumentů.....	36
3.5.10	Vyhledávání dokumentů.....	37
3.5.11	Úprava dokumentu.....	37
3.5.12	Smazání dokumentu .....	38
3.5.13	Zálohování .....	39
3.5.14	Shrnutí praktické části .....	39
	Závěr .....	40
	Použitá literatura .....	41
	Seznam tabulek .....	43
	Seznam obrázků.....	43
	Seznam příloh .....	44

## Úvod a cíl práce

V dnešní době narůstá množství informací, které prochází ekonomickým subjektem, nekontrolovatelným tempem a s tímto informačním růstem souvisí i zvyšování objemu příslušné dokumentace. Nezáleží na tom, zda má podobu klasického papírového dokumentu, nebo zda je v moderní digitální formě. Z těchto příčin je tudíž nutné zavádět takové systémy řízení informací a dokumentů, jejichž technické vlastnosti zajistí, že potřebné dokumenty budou přístupné vždy, kdy je potřeba. A to nejen od pracovního stolu v kanceláři, ale i na cestách prostřednictvím internetu.

Většina podniků je nucena archivovat veškeré dokumenty, které podnikem projdou. Dříve k tomuto účelu s jistotou postačily nejrůznější kartotéky a archívy. Ty však, zvláště ve větších podnicích s velkým množstvím dokumentů, zabíraly velké prostory. Byly tedy náročné jak na skladnost, tak na čas zaměstnanců. S nástupem informačních technologií se začaly vyvíjet programy, které ulehčí práci s dokumenty a které zefektivní práci zaměstnanců. Jedná se o systémy na centrální správu dokumentů. Tyto systémy archivují veškeré dokumenty, které daným ekonomickým subjektem projdou, ať už jsou to dokumenty v elektronické podobě, nebo dokumenty, které musí být naskenováním do elektronické podoby upraveny. Papírové dokumenty musí být stále někde archivovány, ale s využitím systémů na centrální správu dokumentů již odpadá odbíhání zaměstnanců do archivů, nehledě na to, že tímto způsobem může jeden dokument prohlížet více uživatelů najednou. Náklady na pořízení takového systému však bývají často vysoké, a to bývá důvodem, proč tyto programy používají jen velké společnosti.

Cílem této práce je navrhnout systém na centrální správu dokumentů, který by mohly používat malé společnosti, podniky i jednotlivci. Zaměřila jsem se na subjekt z oboru školství, na základní školu. Výsledkem této práce je tedy úže specifikovaný systém na centrální správu dokumentů.

# 1 Centrální správa dokumentů

Centrální správa dokumentů je počítačový systém určený ke správě elektronických či digitalizovaných papírových dokumentů. Tento systém napomáhá jednotlivcům, skupinám pracovníků i velkým podnikům maximálně zefektivnit organizace dokumentů a zajistit jejich maximální dostupnost. [1]

## 1.1 Životní cyklus dokumentu

Každý dokument v administrativním prostředí prochází určitým životním cyklem, který zahrnuje následující fáze [1]:

- Vytvoření dokumentu
- Posouzení dokumentu
- Úpravy dokumentu
- Schválení dokumentu
- Publikování dokumentu
- Distribuce dokumentu
- Dlouhodobé uložení dokumentu

Systém pro centrální správu by měl sledovat celý tento životní cyklus dokumentu. Měl by usnadnit uživatelům vytvoření a uložení dokumentu do systému s následným přidělením práv pro administraci. [1]

Systémy správy dokumentů obvykle nabízejí funkce vyhledávání dokumentů, a to jak podle externích popisných dat, tak i podle obsahu. Mohou být přímo napojeny nebo integrovány do procesu vytváření dokumentů. Externí data, podle kterých je možno v systému vyhledávat, jsou například jména uživatelů, kteří dokumenty uložili. Pokud jsou data vyhledávány podle obsahu, jedná se přímo o data uložená uvnitř dokumentu. [1]

## 1.2 Výhody centrální správy dokumentů

Výhodou těchto systémů je fakt, že každý oprávněný uživatel počítačové sítě, ve které je systém spuštěn, si může snadno a rychle vyhledat potřebný dokument ze svého počítače. Další výhodou je jednodušší a méně časově náročné zavádění dokumentů do systému a nenáročnost na skladové prostory, jaké mají velké archívy.

### **1.3 Nevýhody centrální správy dokumentů**

Nevýhodou těchto systémů je především jejich cena, proto si je pořizují pouze podniky s velkým kapitálem. Další nevýhodou může být přetíženost systému, kdy se k systému připojí příliš mnoho uživatelů a systém zkolabuje, Dále pak nespolehlivost vnitřní sítě podniku, kdy se důležité dokumenty mohou dostat do rukou nepovolané osoby.

## 2 PHP a MySQL

Celá praktická část bakalářské práce je zpracována pomocí skriptovacího jazyka PHP a databázového systému MySQL. V následujících kapitolách je tato problematika blíže popsána.

### 2.1 PHP

PHP (Hypertext Preprocessor) je široce využitelný skriptovací jazyk, který je obzvláště vhodný pro vývoj webových aplikací a lze jej zapouzdřit do HTML. PHP se liší JavaScriptu tím, že kód je vykonáván na straně serveru, generováním HTML, který je pak poslán ke klientovi. Klient obdrží výsledky ze skriptu, ale neví, jak zdrojový kód vypadal. Neví to ani uživatel, který by si chtěl v internetovém prohlížeči kód zobrazit, PHP se mu jednoduše nezobrazí. [2] To je také výhodou, proč vytvářet webové aplikace v tomto jazyce. V následujících podkapitolách jsou uvedené některé základní operace v PHP.

#### 2.1.1 Proměnné v PHP

Každá proměnná v PHP musí mít jednoznačně definovaný název. Tento název začíná vždy znakem \$ a následuje pojmenováním proměnné. První znak toho pojmenování musí být buď písmeno a-z nebo podtržítka. Jiné znaky nejsou povoleny. Proměnné v PHP jsou citlivé na velikost písmen, to znamená že \$id není to samé co \$ID. V praxi se většinou názvy proměnných píše malými písmeny. Desetinná čísla se zadávají s tečkou, ne s čárkou. Řetězce se uzavírají do uvozovek nebo do apostrofů. [3] Ve výchozím nastavení je dané proměnné vždy přiřazena určitá hodnota. To znamená, že když je přiřazen výraz do proměnné, celá hodnota původního výrazu se zkopíruje do cílové proměnné. [4]

#### 2.1.2 Funkce v PHP

Funkce PHP obsahují větší počet procesů. Při programování v PHP často dochází k procesům, které se opakují, proto je lepší pro ně vytvořit funkci. To přináší lepší orientaci v psaném skriptu. Některé funkce jsou v PHP již předdefinované, je však možné si vytvořit funkce vlastní. Hlavní slovem, určeným k definování funkce je slovo `function`. [3]

```
<?
    function napis()
    {
        echo ("hello world");
    }
?>
```

Funkce `napis` může být vyvolána ve skriptu na jakémkoliv místě. Je-li tato funkce vyvolána, vypíše `hello word`. Funkce v PHP se mohou dělit na funkce [5]:

- bez argumentu
- s argumentem

### ***Funkce bez argumentu***

Příkladem funkce bez argumentu je výše zmíněná funkce `napis`, v závorkách u funkce není uveden žádný argument. [5]

### ***Funkce s argumentem***

Deklarace funkce je stejná jako v předchozím případě, kromě uvedení atributu v závorkách u názvu funkce. Ten zavádí proměnnou `$jmeno`.

```
<?
    function napis($jmeno)
    {
        echo($jmeno);
    }
?>
```

Takto popsaná funkce přiřadí výraz zadaný do závorek do proměnné `$jmeno` a při zavolání funkce jej vypíše.

```
<?
    napis("ahoj"); // vypíše ahoj
?>
```

## 2.2 MySQL

MySQL je nejpoužívanějším databázovým systémem. Je dobře známý pro svou rychlost, robustnost a malými režijními náklady na připojení. To je důležité pro obsluhu webových stránek, které musí být obslouženy tak rychle jak je to jen možné. [6]

### 2.2.1 Použité funkce MySQL

V této podkapitole jsou uvedeny nejpoužívanější funkce, které lze v souvislosti s MySQL a PHP používat. Jedná se o propojení skriptovacího jazyka PHP s databází MySQL. Tyto funkce byly použity v praktické části bakalářské práce.

**mysql\_connect** otevírá spojení s databází. Tato funkce nabývá hodnot TRUE, je-li spojení vytvořeno, nebo FALSE, pokud spojení selhalo. Parametry funkce mysql\_connect jsou server, uživatelské jméno a heslo pro připojení k databázi. [7]

**mysql\_select\_db** vybere databázi MySQL. Parametry této funkce jsou jméno databáze a spojení. Spojení je definované pomocí předchozí funkce mysql\_connect. Funkce mysql\_select\_db nabývá hodnot TRUE, je-li databáze vybrána, nebo FALSE pokud se příkaz nezdařil. [8]

**mysql\_query** odešle dotaz v jazyce SQL do databáze. Parametrem této funkce je tedy SQL dotaz. Dalším parametrem může být libovolné spojení. [9]

**mysql\_fetch\_row()** načte jeden záznam výsledku do pole s číselnými klíči. Každá hodnota sloupce je samostatná hodnota pole, klíče jsou číslovány od 0. Další volání mysql\_fetch\_row() vrátí následující záznam výsledku, nebo FALSE není-li žádný další záznam. [10]

**mysql\_fetch\_array** je rozšířenou verzí mysql\_fetch\_row. Data jsou zde uložena v poli nejen pod číselnými klíči, ale také pod asociativními textovými klíči, které se jmenují podle názvů sloupců SQL tabulky. Funkce vrací pole hodnot načteného záznamu nebo FALSE, není-li žádný další řádek. Pokud dva nebo více sloupců mají stejný název, bude dostupná hodnota pouze toho posledního. [11]

**mysql\_close** uzavře spojení s databází. Použití této funkce není povinné, spojení s databází se samo uzavře se zavřením aplikace. [12]

Účelem této podkapitoly nebylo vypsat veškeré funkce, které MySQL nabízí, ale ukázka funkcí, které jsou následně použité v praktické části této bakalářské práce.

## 2.2.2 phpMyAdmin

Nástroj phpMyAdmin je webová aplikace napsaná za pomoci skriptovacího jazyka PHP. Tato aplikace obsahuje klientský kód XHTML, CSS a Javascriptu. Protože poskytuje kompletní webové rozhraní pro administraci databází MySQL, je nejvhodnější a nejrozšířenější aplikací. Protože se informační technologie stále vyvíjí, rostou rychle i nároky na spravování databází. Z toho důvodu jsou vyvíjeny nové standardy MySQL a tak s nimi musí i tato aplikace držet krok. Nejčastějšími operacemi, které se v aplikaci phpMyAdmin uskutečňují, je vytvoření a odstranění databáze, a vytvoření, přejmenování, kopírování a odstranění databázové tabulky. [6] To vše může být aplikováno za pomoci jazyka SQL, proto o něm uvádím pár informací v následující kapitole.



Obrázek 1 – Náhled aplikace phpMyAdmin Zdroj: vlastní

### 2.2.3 Jazyk SQL

SQL (Structured Query Language) je nejpoužívanějším databázovým jazykem. Stal se standardem v oblasti zpracovávání dat. SQL náleží do třídy neprocedurálních jazyků. Popisuje tedy, co požadujeme od databáze, ale ne jakým způsobem to vykonáváme. [13]

Jazyk SQL se dělí na následující části [14]:

- Příkazy pro definici dat (Data Definition Language – DDL)
- Příkazy pro manipulaci s daty (Data Manipulation Language – DML)

#### ***Příkazy pro definici dat***

Tato část jazyka umožní uživateli definovat, modifikovat a rušit objekty, které se vyskytují v databázi. [13]

**CREATE** – umožní vytvořit novou databázi, nebo novou tabulku. Nová tabulka se vytvoří pomocí SQL takto [14]:

```
CREATE TABLE Pracovníci
    (id_prac CHAR(2) NOT NULL,
    Jmeno VARCHAR (12),
    Prijmeni VARCHAR (15),
    Dat_nastupu DATE );
```

**ALTER TABLE** – umožní upravit tabulku. Stávající tabulku Pracovníci lze upravit třeba takto:

```
ALTER TABLE Pracovníci MODIFY (Jmeno VARCHAR (15))
```

**DROP** – umožní zrušit databázi, tabulku, nebo indexy. Tabulku Pracovníci lze zrušit jednoduchým příkazem:

```
DROP TABLE Pracovníci
```

#### ***Příkazy pro manipulaci s daty***

Tyto příkazy umožní uživateli upravovat databázi, případně vybrat a odvodit data potřebná v aplikaci.

**SELECT** – je nejvýznamnějším a nejčastějším příkazem SQL, pomocí kterého je realizován výběr požadovaných dat z databáze.

```
SELECT * FROM nazev_tabulky
```

**INSERT** – pomocí tohoto příkazu je možné vložit jeden nebo několik řádků do tabulky. Při vkládání se kontrolují integritní omezení definované pro danou tabulku.

```
INSERT INTO nazev_tabulky [(seznam sloupců)]  
{VALUES (seznam hodnot)}
```

**DELETE** – umožní smazat data z tabulky, nebo podmnožinu řádků tabulky, na základě definované podmínky z výběru.

```
DELETE FROM nazev_tabulky [WHERE podmínka]
```

**UPDATE** – umožní změnit hodnoty uložené v databázi.

```
UPDATE nazev_tabulky SET nazev_sloupce = vyraz[...]
```

**WHERE** – podmínka, která může být definována jako součást příkazů DELETE, UPDATE, SELECT. Představuje logický výraz, který nabývá hodnot `true` nebo `false`, podle toho, zda je výraz pravdivý či nikoliv. [13]

## 3 Realizace

Tato kapitola se věnuje samotné praktické části této bakalářské práce. Nejprve prochází analýzu dokumentů, které jsou v subjektu školství používány. Z nich pak vyjímá atributy pro kategorizaci a vyhledávání. Dále je zde navržen E-R Diagram, Use-Case diagram a relační model databáze, který je následně implementován do databázového prostředí. V poslední části jsou ukázky ze samotné aplikace. V této kapitole se vyskytují diagramy, tabulky, části kódu a obrázky související s webovou aplikací.

### 3.1 Požadavky na aplikaci

V reálném světě jsou aplikace na centrální správu dokumentů firmám zadávány s přesně vymezenými kritérii, která daná aplikace musí splňovat. Proto i při tvorbě této práce byl použit stejný postup. Organizace má na aplikaci následující požadavky:

- Snadný uživatelský přístup
- Pevně dané rozdělení kategorií
- Design zapadající do vnitřního informačního systému organizace

Jedná se tedy o aplikaci tzv. na klíč. Úpravy aplikace jsou možné, ovšem provádět je nebude administrátor aplikace, ale programátor v příslušné zadavatelské společnosti.

### 3.2 Rozbor typů dokumentů

Dříve než začne samotná realizace webové aplikace, je třeba stanovit, jaké dokumenty se v daném subjektu vyskytují. Jak bylo výše zmíněno, tato práce je úžeji specifikovaná pro subjekt školství, pro základní školu. Proto i typy dokumentů tomu odpovídají. Jedná se zejména o dokumenty určené jako podklady pro výuku, družinu a dům dětí a mládeže. Dalšími typy dokumentů jsou dokumenty, které se vztahují k chodu daného subjektu. Jsou jimi vyhlášky a směrnice školského subjektu, ředitelská nařízení a doporučení, dokumenty vztahující se k programům, projektům a prezentacím. Dokumenty v subjektu se vyskytují ve všech typech přípon (například .doc, .jpg, .ppt). V následující tabulce jsou znárodněny jednotlivé oblasti, do kterých dokumenty spadají, spolu s osobami daného subjektu, které k dokumentům mají přístup.

**Tabulka 1 - Rozbor typů dokumentů Zdroj: vlastní**

Název oblasti	Přístup osob
Výuka	Vyučující, Studenti, Administrátor
Management	Vyučující, Administrátor
Provoz	Vyučující, Administrátor
Projekty	Vyučující, Administrátor
Prezentace	Vyučující, Administrátor
Družina	Vyučující, Studenti, Administrátor
Dům dětí a mládeže	Vyučující, Studenti, Administrátor

### 3.3 Definice atributů pro kategorizaci a vyhledávání

V každé dobře fungující aplikaci by měly být dokumenty nějakým způsobem rozděleny do kategorií. V tomto případě jsou kategoriemi oblasti, do kterých dané dokumenty zapadají. Jedná se tedy o stejné rozdělení jako v předchozí kapitole.

Vyhledávání je v aplikaci prováděno podle atributů, které mají dokumenty společné. Jedná se o atributy dokumentu, které uživatel zadá do databáze. Zadávanými atributy je název dokumentu, popis dokumentu, klíčová slova, datum přidání, stupeň, předmět a soubor. Z těchto atributů jsou pro všechny dokumenty stejné pouze atributy uvedené v následující tabulce.

**Tabulka 2 - Atributy na vyhledávání Zdroj: vlastní**

Název atributu	Popis
Název dokumentu	Atribut, udávající jednoduchý název dokumentu.
Popis dokumentu	Atribut, který udává bližší popis dokumentu.
Klíčová slova	Atribut, kde jsou heslovitě popsána klíčová slova, která dokument obsahuje.

### 3.4 Návrh databázového modelu

Každý návrh modelu prochází třemi úrovněmi návrhu. Jedná se o konceptuální, logickou a implementační úroveň. V tomto projektu byly splněny všechny úrovně.

### 3.4.1 E-R Diagram

Modelování E-R diagramu se řeší na konceptuální úrovni modelování. Na této úrovni se neřeší, jakým softwarem a programovacím jazykem bude aplikace vytvářena. Programátor se zde pouze seznamuje s objekty reálného světa. E-R konceptuální model je množina pojmů, které nám pomáhají na konceptuální úrovni abstrakce popsat uživatelskou aplikaci s cílem následné specifikace struktury databáze [13].

#### ***Definice entit a jejich atributů***

V systému centrální správy dokumentů mohou vystupovat různí uživatelé i dokumenty. Z každého je třeba definovat entity a jejich atributy. V systému centrální správy dokumentů, které je úžeji definováno pro subjekt školství, se vyskytují následující entity a jejich atributy.

- ***Dokumenty***

Na základě předchozího rozdělení typů dokumentů vyplynulo, že entitami, které odlišují dokumenty od sebe, je jejich zařazení do kapitol. Atribut, který je zvýrazněn, je primárním klíčem.

Tabulka 3 - Entity a jejich atributy 1 Zdroj: vlastní

Název entity	Atributy
Výuka	<b>ID_v</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Předmět, Stupeň, Soubor
Management	<b>ID_m</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Soubor
Provoz	<b>ID_voz</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Soubor
Projekty	<b>ID_pjkt</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Soubor
Prezentace	<b>ID_pre</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Soubor
Družina	<b>ID_d</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Soubor
Dům dětí a mládeže	<b>ID_ddm</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Soubor
Dokument	<b>ID_doc</b> , Název dokumentu, Popis dokumentu, Klíčová slova, Datum přidání, Předmět, Stupeň, Soubor

- **Uživatelé**

Uživatelé, kteří v systému vystupují, můžeme rozdělit na tři podskupiny.

Tabulka 4 - Entity a jejich atributy 2 Zdroj: vlastní

Název entity	Atributy
Vyučující	<b>ID_uc</b> , Jméno, Příjmení
Student	<b>ID_st</b> , Jméno, Příjmení
Administrátor	<b>ID_adm</b> , Jméno, Příjmení

### **Definice vztahů mezi entitami**

Vztah je definován jako vazba mezi jednotlivými entitami. Následující vztahy figurují v modelu centrální správy dokumentů v oboru školství.

- **Přiřazen**

Tento vztah je vazbou mezi kapitolou a dokumentem. Jedná se o nepovinný vztah s vazbou 1:N. K jedné kapitole může být přiřazeno více dokumentů, ale jen jeden dokument může být přiřazen právě jedné kapitole.

- **Uložen**

Tento vztah je vazbou mezi dokumentem a vyučujícím. Jedná se, stejně jako u předchozího, o nepovinný vztah s vazbou 1:N. Jeden vyučující může uložit více dokumentů, ale naopak jeden dokument může být uložen jen jedním vyučujícím.

- **Vyhledáván**

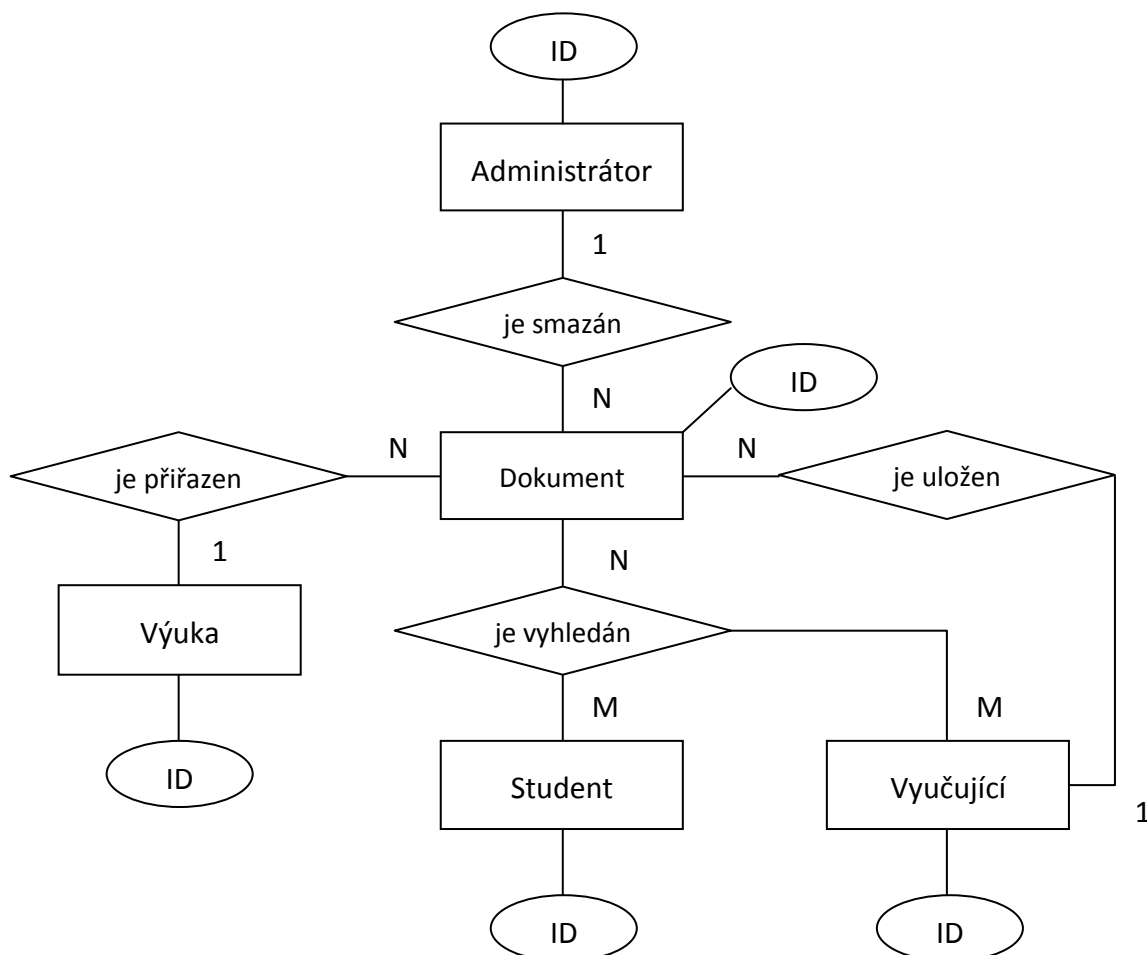
Tento vztah je vazbou mezi dokumentem a studentem, a dokumentem a vyučujícím. Jedná se o nepovinný vztah s vazbou N:M. Jeden student může v jeden čas vyhledat více dokumentů, a zároveň jeden dokument může být vyhledáván více studenty.

- **Smazán**

Tento vztah je vazbou mezi dokumentem a administrátorem. Jedná se o vztah s vazbou 1:N. Jeden administrátor může smazat více dokumentů, ale naopak jeden dokument může být smazán právě jedním administrátorem.

## E-R Diagram

Entity a jejich vztahy jsou nadefinované, je tedy možné zakreslit tyto náležitosti do schématu, kterému se říká E-R diagram. Pro větší přehlednost jsem použila pouze jednu z kapitol a to výuka. Modely s ostatními kapitolami jsou tvořeny analogicky.



Obrázek 2 - E-R diagram Zdroj: vlastní

### 3.4.2 Relační databázový model

V relační databázi definujeme vztahy (relace) mezi tabulkami, řádky (záznamy), sloupce (atributy), primárními a cizími klíči. Pravidla pro definici základních prvků relační databáze jsou o něco přísnější než u databází, u kterých se vztahy mezi objekty nedefinují. [15]

Skutečné uspořádání záznamů, nebo polí, v databázi je zcela nepodstatné a každý záznam v tabulce je definován polem, které obsahuje unikátní hodnotu. [16]

#### **Databázová tabulka**

Podle relačního modelu jsou data v relační databázi uložena ve vztazích, které uživatel vidí jako tabulky. [16] V databázových tabulkách jsou všechna data uložena v jednotlivých polích uspořádaných do řádků, které nazýváme záznamy, a sloupců, které nazýváme

atributy. [15] Tabulky jsou hlavními strukturami v databázi a každá vždy reprezentuje jednu specifickou entitu. [16]

### **Řádek**

Řádek je kombinace sloupcových hodnot v tabulce, která je identifikovatelná pomocí primárního klíče. [15]

Tabulka 5 - Řádek databázové tabulky Zdroj: vlastní – upraveno na základě [15]

ID	Uživatelské jméno	Heslo	Jméno	Příjmení	Typ uživatele
1	jan.novak	*****	Jan	Novák	Student
2	ticha.martina	*****	Martina	Tichá	Vyučující
3	vopicak	*****	Alois	Vopička	administrátor
4	Ignác.urban	*****	Ignác	Urban	vyučující

### **Sloupec**

Sloupec je množina dat jednoho typu. Sloupce představují jednotlivé atributy, každý sloupec má svůj název a obsahuje hodnoty stejného datového typu. Můžeme se tedy také setkat s označením, které přiřazuje sloupci název atribut. [17]

Tabulka 6 - Sloupec databázové tabulky Zdroj: vlastní – upraveno na základě [15]

ID	Uživatelské jméno	Heslo	Jméno	Příjmení	Typ uživatele
1	jan.novak	*****	Jan	Novák	Student
2	ticha.martina	*****	Martina	Tichá	Vyučující
3	vopicak	*****	Alois	Vopička	administrátor
4	Ignác.urban	*****	Ignác	Urban	vyučující

### **Klíče**

Klíče jsou specifickými poli, která hrají v tabulce velmi významnou roli. Typ klíče určuje jeho význam v tabulce. Existuje několik typů klíčů, ale nejvýznamnější jsou primární klíč a cizí klíč. [16]

- **Primární klíč**

Řádky v relaci nemají specifické pořadí, tudíž nejsou ani dosažitelné číslem řádku, musí existovat nějaká konstrukce, která nám umožní adresovat jednotlivé řádky. Tato konstrukce se nazývá primární klíč. [18] Primární klíč je sloupec, případně kombinace sloupců, který slouží k jednoznačné identifikaci každého řádku tabulky. Hodnota pole primárního klíče musí být v rámci tabulky jedinečná. Pole primárního klíče musí obsahovat konkrétní hodnotu, tedy nesmí být prázdná. [15]

Tabulka 7 - Primární klíč Zdroj: vlastní – upraveno na základě [15]

ID	Název	Popis	Klíč	Datum	Soubor
1	Bulharsko	Mapa země	Mapa Bulharsko	2010-04-06	Bulharsko.jpg
2	ČR	Státní znak	Státní znak ČR	2009-10-23	Vlajka.jpg
3	Německo	hranice	hranice Německo	2010-02-10	Hran_nem.doc
4	Rakousko	Tradice země	Tradice Rakousko	2010-03-15	Tradice.doc

V tomto případě je primárním klíčem pole ID, protože v ostatních sloupcích může dojít k duplicitě zadaných hodnot.

### 3.4.3 Výhody relační databáze

Relační databáze má několik výhod [16]:

- Zabudovaná víceúrovňová integrita

Integrita dat je zabudována přímo do modelu na úrovni položek, aby se zajistila přesnost dat. Na úrovni tabulek zajišťuje, že záznamy nejsou duplicitní a detekuje chybějící hodnoty primárních klíčů. Na úrovni vztahů zajišťuje, že vztah mezi dvojicí tabulek je platný.

- Logická a fyzická nezávislost dat na databázové aplikaci

Ani změny uživatele v logickém návrhu, ani změny poskytovatele databázového softwaru ve fyzické implementaci neovlivní aplikaci postavenou pro původní logický návrh a databázový software.

- Garantovaná konzistence a přenosnost dat

Data jsou konzistentní a přesná díky různým úrovním integrity, které se dají v databázi vytvořit.

- Snadné získávání dat

Na základě uživatelského příkazu mohou být data získána buď z určité tabulky, nebo z libovolného počtu tabulek, které jsou ve vztahu. To uživateli umožňuje zobrazovat informace téměř neomezeným počtem způsobů.

### 3.4.4 Use Case model

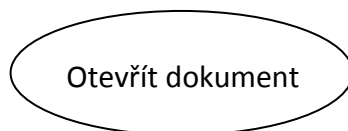
Use Case model je model případů užití. Jedná se o dynamické zobrazení struktury systému z pohledu uživatele. V tomto modelu se neodhaduje, jakou má systém vnitřní strukturu, řeší se pouze jeho chování. Chování systému se definuje pomocí scénářů. Každý scénář obsahuje posloupnost událostí, které zde probíhají a také popis komunikace mezi uživatelem a systémem. Modelu případů užití řeší odpovědi na otázky, které si klade každý programátor při tvorbě systému, např. jaké jsou úlohy aplikace, co bude uživatel v aplikaci vyhledávat atd. [19] Vše je popsáno pojmy běžného jazyka, proto není potřeba znát jazyky programovací. [20]

#### *Use Case diagram*

Diagramy případů užití poskytují rychlou představu o jednotlivých funkcích systému. Přesné postupy a rozšiřující scénáře musí být zachyceny v textové podobě. Při tvorbě diagramu definujeme případy užití, aktéry, a jejich vazby. [20]

- ***Případ užití***

Případ užití je popsán oválem a popisem případu. [20]



Obrázek 3 - Případ užití Zdroj: vlastní - upraveno na základě [19]

- ***Aktér***

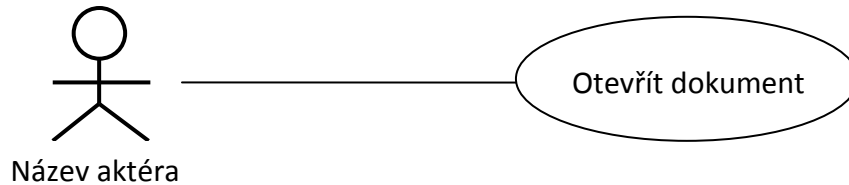
Pro aktéra je nejčastěji používán symbol postavy a jeho název. [20]



Obrázek 4 - Symbol aktéra Zdroj:vlastní - upraveno na základě [19]

- **Vazby**

Nejdůležitější vazbou v diagramu užití je vazba asociace. Jedná se o vazbu mezi aktérem a případem užití. Tato vazba se znázorňuje plnou nepřerušovanou čarou. U vztahu asociace se předpokládá stereotyp, i když se do diagramu neznázorňuje. [20]



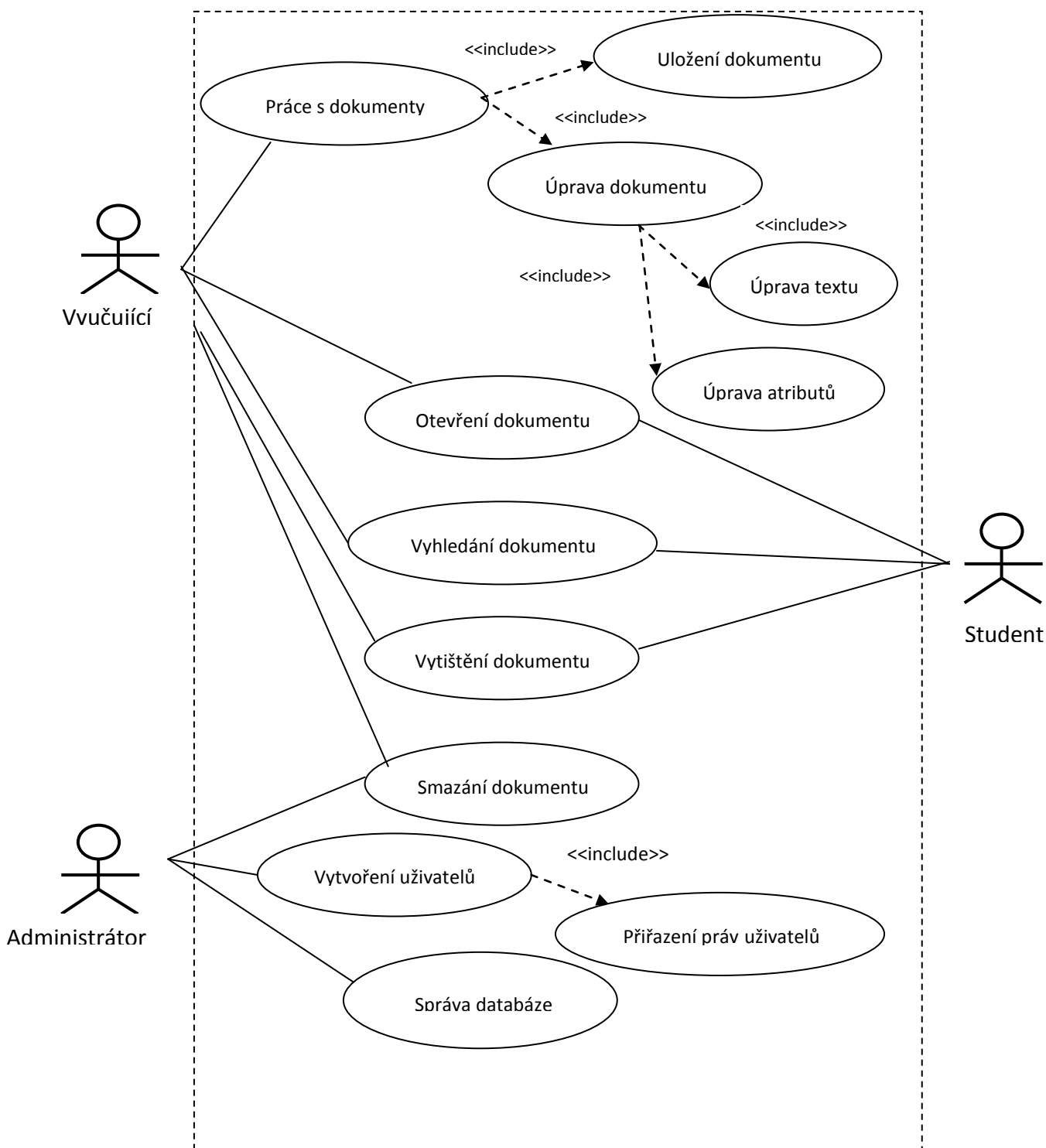
Obrázek 5 - Vazba asociace Zdroj: vlastní - upraveno na základě [19]

- **Hranice systému**

Hranice systému je v případě use case diagramu nepovinná a pouze ilustrativní. Znáznorňuje s obdélníkem kolem jednotlivých případů užití. [19]

### ***Use Case diagram pro centrální správu dokumentů***

V systému pro centrální správu dokumentů, definovaném pro subjekt oboru školství, jsou tři skupiny aktérů. Jedná se o vyučující, studenty a administrátora. Vyučující si mohou v systému vyhledávat podklady pro vyučování, ale i organizační dokumenty. Mají od administrátora přidělená práva ke změně dokumentů. Studenti si v systému mohou vyhledávat dokumenty související s výukou, otevírat jednotlivé dokumenty a zároveň si je i vytisknout. Administrátor spravuje databázi jako celek, vytváří účty uživatelům a přiděluje jim práva na úpravu, případné smazání dokumentu.



Obrázek 6 - Use Case diagram Zdroj: vlastní

### 3.5 Praktická fáze tvorby systému

Praktická část této bakalářské práce je tvořena pomocí skriptovacího jazyka PHP, značkovacího jazyka HTML, CSS kaskádových stylů a databázového systému MySQL. Ošetření formulářů proti odeslání prázdných údajů je vytvořeno pomocí skriptovacího jazyka Javascript. Podmínkou pro provoz této aplikace je webový server s podporou PHP

a databázový systém MySQL a webový prohlížeč. Vytvořená aplikace je optimalizovaná pro prohlížeč Mozilla Firefox.

### 3.5.1 Vytvoření databáze

K vlastnímu zpracování je, jak již bylo zmíněno, využíván databázový systém MySQL. V tomto případě byla použita verze MySQL 5.0.90. Pro vytvoření jednotlivých tabulek v databázi bylo využito prostředí phpMyAdmin. Při návrhu tabulek lze využít ruční zadání, nebo zadání příkazu SQL. Při využití jazyka SQL může vypadat příkaz následovně:

```
CREATE TABLE `e06810`.`nauka` (  
  `ID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `nazev` VARCHAR( 255 ) CHARACTER SET cp1250 COLLATE  
  cp1250_czech_cs NOT NULL ,  
  `popis` VARCHAR( 255 ) CHARACTER SET cp1250 COLLATE  
  cp1250_czech_cs NOT NULL ,  
  `klic` VARCHAR( 255 ) CHARACTER SET cp1250 COLLATE  
  cp1250_czech_cs NOT NULL ,  
  `datum` DATE NOT NULL DEFAULT '0000-00-00',  
  `predmet` VARCHAR( 255 ) CHARACTER SET cp1250 COLLATE  
  cp1250_czech_cs NOT NULL ,  
  `stupen` VARCHAR( 10 ) CHARACTER SET cp1250 COLLATE  
  cp1250_czech_cs NOT NULL ,  
  `pristup` VARCHAR( 20 ) CHARACTER SET cp1250 COLLATE  
  cp1250_czech_cs NOT NULL ,  
  `soubor` VARCHAR( 255 ) CHARACTER SET cp1250 COLLATE  
  cp1250_czech_cs NOT NULL  
  ) ENGINE = MYISAM
```

V celé databázi, která se vztahuje k této webové aplikaci, figuruje celkem devět tabulek, osm tabulek, které se vztahují ke kapitolám, a jedna tabulka, do které se ukládají registrovaní uživatelé.

#### ***Databázové tabulky***

V této kapitole jsou popsány databázové tabulky a jejich atributy. Jako příklad jsou uvedené pouze dvě tabulky, tabulka výuka, uživatelé a management. Zbýlých šest tabulek je vytvořeno analogicky k tabulce management.

Tabulka 8 - Popis tabulky výuka Zdroj: vlastní

Tabulka Vyuka		
Název atributu	Datový typ	Popis
ID	INT	Identifikační číslo záznamu, primární klíč
nazev	VARCHAR	Název dokumentu, atribut k vyhledávání
popis	VARCHAR	Popis dokumentu, slouží k přiblížení obsahu dokumentu, atribut k vyhledávání
klic	VARCHAR	Klíčová slova dokumentu, slouží jako atribut k vyhledávání
datum	DATE	Datum vložení dokumentu
predmet	VARCHAR	Předmět, ke kterému se dokument vztahuje, podle tohoto atributu jsou dokumenty v tabulce seřazeny
stufen	VARCHAR	Stupeň, ke kterému se dokument vztahuje. Zda se jedná o první nebo druhý stupeň.
pristup	VARCHAR	Pole, podle kterého se dokument zpřístupní studentům nebo ne.
soubor	VARCHAR	Název souboru, podle něj se vyhledává ve složce na serveru

Tabulka 9 - Popis tabulky uživatelé Zdroj: vlastní

Tabulka Uzivatele		
Název atributu	Datový typ	Popis
ID	INT	Identifikační číslo záznamu, primární klíč
uziv_jmeno	VARCHAR	Uživatelské jméno, slouží k přihlášení uživatele, unikátní klíč
heslo	VARCHAR	Heslo, slouží k přihlášení uživatele, uloženo pomocí md5 kódování
jmeno	VARCHAR	Jméno uživatele, nepovinné
prijmeni	VARCHAR	Příjmení uživatele, nepovinné
typ_uziv	INT	Typ uživatele, přidělují se podle něho uživatelská práva

Tabulka 10 - Popis tabulky Management Zdroj: vlastní

Tabulka Management		
Název atributu	Datový typ	Popis
ID	INT	Identifikační číslo záznamu, primární klíč
nazev	VARCHAR	Název dokumentu, atribut k vyhledávání
popis	VARCHAR	Popis dokumentu, slouží k přiblížení obsahu dokumentu, atribut k vyhledávání
klic	VARCHAR	Klíčová slova dokumentu, slouží jako atribut k vyhledávání
datum	DATE	Datum vložení dokumentu
pristup	VARCHAR	Pole, podle kterého se dokument zpřístupní studentům nebo ne.
soubor	VARCHAR	Název souboru, podle něj se vyhledává ve složce na serveru

### 3.5.2 Připojení k databázi

Kdykoli je potřeba vložit nějaká data do databáze, nebo je z ní vypsat, musí se k ní nejprve aplikace připojit. To se děje následující syntaxí:

```
<?
    $hostname= "jmeno_serveru";
    $databasename= "jmeno_databaze";
    $username= "jmeno_uzivatele";
    $password= "heslo";
    $spojeni = MySQL_Connect ($hostname, $username, $password);
    MySQL_Select_DB ($databasename);
    //práce s databází
    MySQL_Close ();
?>
```

Hodnoty jmeno\_serveru, jmeno\_databaze, jmeno\_uzivatele a heslo je vždy nutné přepsat na skutečné přihlašovací údaje do databáze. Tento skript je nutný pro jakékoliv operace s databází. Funkcí MySQL\_Connect se připojí aplikace k databázi. Funkce MySQL\_Select\_DB vybere tu správnou databázi, ke které se má aplikace připojit. Poté

následuje samotná práce s databází. V této fázi jsou obvykle prováděny SQL dotazy na databázi. Poté funkce `MySQL_Close` databázi opět uzavře.

### 3.5.3 Kontrola polí formuláře

Některá formulářová pole, pomocí kterých uživatelé zadávají data do databáze, nesmějí zůstat prázdná, jinak by se data neuložila správně. Ke kontrole slouží kód psaný ve skriptovacím jazyce Javascript. [21]

```
<script>
    function kontrolaDat(vstup)
    {
        if( vstup.nazev.value.length == 0){
            alert("Nevyplnili jste název, nelze odeslat.");
            return false; }
    }
</script>
```

Tato funkce se spustí při odesílání dat z formuláře. Pokud uživatel nevyplnil vstupní pole název, zobrazí se okno, které uživateli vypíše, kterou povinnou položku nevyplnil. Pokud jsou všechny povinné položky vyplněné, formulář se odešle ke zpracování. Vše zajistí tento kus kódu:

```
<form action = "zpracuj.php" method = "post" onsubmit = "return
    kontrolaDat(this)">
```

### 3.5.4 Rozložení aplikace

Aplikace je členěna do tří hlavních částí. Jedná se o hlavičku, kde je přihlášení registrovaných uživatelů, dále pak o levé menu, kde jsou dokumenty rozdělené podle jednotlivých kategorií a hlavní část, kde se hledané dokumenty zobrazují.



Obrázek 7 - Rozložení aplikace Zdroj: vlastní

### 3.5.5 Registrace uživatele

Každý systém má registrované uživatele. V tomto případě může uživatele zaregistrovat pouze administrátor, aby se předešlo nechtěnému vniknutí do systému. Ostatním uživatelům se nabídka Registrovat nezobrazí. Registrace je v této aplikaci pouze ilustrativní, předpokládá se propojení s již běžícím systémem generování uživatelů.

Uživatelé se registrují pomocí formuláře, kde administrátor vyplní uživatelské jméno, heslo, jméno a příjmení uživatele a zadá, zda je uživatel student, vyučující, či administrátor. To pak následně určí, jaká práva bude mít uživatel při nahlížení do aplikace. Registrace uživatele je ošetřena proti přidání duplicitních uživatelských jmen podmínkou, která vypadá následovně:

```
$r = mysql_query("SELECT uziv_jmeno FROM uzivatele  
WHERE uziv_jmeno='". $uziv_jmeno .' ' LIMIT 1");  
if(mysql_num_rows($r))  
echo 'Zadané uživatelské jméno již existuje, zadejte jiné.';
```

## Registrace uživatele

\*Uživatelské jméno:

\*Heslo:

Jméno:

Příjmení:

\*Typ uživatele:

- vyberte--
- Student
- Vyučující
- Administrátor

Obrázek 8 - Registrace uživatelů - formulář Zdroj: vlastní

### 3.5.6 Přihlášení uživatele

Zaregistrovaní uživatelé se mohou při vstupu do aplikace přihlásit. To je zajištěno přes funkci, která vypadá následovně:

```
function UzivConnect($login,$pass)
{
    $spojeni = MySQL_Connect($hostname, $username,$password);
    MySQL_Select_DB($databasename);
    MySQL_query("SET CHARACTER SET cp1250");

    $res = mysql_query('SELECT ID FROM uzivatele
    WHERE uziv_jmeno="'.htmlspecialchars($login) .'" AND
    heslo="'. md5($pass) .'" LIMIT 1');
    $r = mysql_fetch_array($res);
    if($r['ID'])
    {
        $_SESSION['uid']=$r['ID'];
        echo'

        <script>window.location.replace("http://cipisek.upce.cz/e06810
        /");</script>';

        return true;
    }
}
```

Přihlášení:

Obrázek 9 - Přihlášení - formulář Zdroj: vlastní

### 3.5.7 Přidělení uživatelských práv

Aby se nestalo, že například studenti budou měnit dokumenty, musela být aplikace ošetřena uživatelskými právy. Ty jsou různá podle přihlášeného uživatele.

#### ***Nepřihlášený uživatel***

Nepřihlášený uživatel má v aplikaci velmi omezená práva. Může si pouze prohlížet zadané záznamy, nemůže si však soubory zobrazovat. Ukázka je uvedena příloze B.

#### ***Uživatel Student***

Uživatel, který je přihlášený jako student, si může listovat záznamy, které jsou pro studenty zpřístupněny, může si u těchto záznamů zobrazit daný soubor a vyhledávat v záznamech. Nemá však přidělená práva na přidání, úpravu nebo smazání dokumentu. Ukázka je uvedena příloze C.

#### ***Uživatel Vyučující***

Uživatel, který je přihlášen jako vyučující, má veškerá práva jako administrátor, pouze nemůže registrovat uživatele. Má tedy práva k zobrazení, vyhledávání, přidání, úpravě a smazání dokumentů. Vyučující také rozhodují, zda uložený dokument bude pro studenty viditelný či nikoliv. Ukázka je uvedena příloze D.

#### ***Uživatel Administrátor***

Uživatel, přihlášený jako administrátor, má nad aplikací veškerá práva, která jsou zde nastavena. Jedná se o všechna práva výše uvedená, včetně registrace uživatelů. Ukázka je uvedena příloze E.

Přidělování práv v aplikaci je zajištěno pomocí funkce, která vypadá následovně:

```
function Prava($pravo)
{
    $spojeni = MySQL_Connect($hostname, $username, $password);
    MySQL_Select_DB($databasename);
    MySQL_query("SET CHARACTER SET cp1250");
    global $uid;
    if(!$uid)
        return false;
    else
    {
        $res = mysql_query("SELECT typ_uziv FROM uzivatele WHERE
            ID='". (int)$uid ." LIMIT 1");
        $r = mysql_fetch_row($res);
        if($r[0]>= $pravo)
            return true;
        else
            return false;
    }
}
```

Funkce načte typ uživatele z databáze a poté jej přiřadí do proměnné \$pravo. Funkce je poté v dalších skriptech vyvolávána vždy tam, kde je třeba některému z uživatelů zamezit přístup. Parametrem, který se zapisuje při vyvolání funkce, je hodnota typu uživatele. Tato hodnota se ukládá do databáze již při registraci uživatele, pokud se jedná o studenta, uloží se 1, pokud o vyučujícího, uloží se hodnota 2. V případě administrátora se uloží hodnota 3. Vyvolání funkce může být zapsáno i v podmínce, třeba takto:

```
<?
    if(Prava('2'))
        echo '<a href="?"stranka=vlozit_vyuka">Přidat dokument</a>';
?>
```

V tomto případě se odkaz „Přidat dokument“ ukáže uživatelům s hodnotou typu uživatele 2 a všem vyšším. Tedy vyučujícím a administrátorovi. Pro studenty zůstane tento odkaz skrytý.

### 3.5.8 Přidání dokumentu

Aby mohl systém na centrální správu dokumentů správně fungovat, musí být databáze naplněna potřebnými dokumenty. Tyto dokumenty jsou do databáze přidávány pomocí formuláře. Zde přihlášený uživatel vyplní název dokumenty, jeho popis a klíčová slova, podle kterých se následně budou dokumenty vyhledávat. Uživatel dále určí, zda chce dokument zpřístupnit ostatním uživatelům a v případě oblasti výuka zadá, o jaký předmět se jedná a zda se dokument vztahuje k prvnímu, či druhému stupni. Posledním a nejdůležitějším krokem, je vyhledat soubor v adresáři a odeslat formulář ke zpracování. Tím se zadané údaje uloží do databázové tabulky a soubor do adresáře na webovém serveru.

## Přidání dokumentu

Název dokumentu:	<input type="text"/>	*
Popis dokumentu:	<input type="text"/>	*
Klíčová slova:	<input type="text"/>	*
Předmět:	--vyberte--	*
Stupeň:	<input checked="" type="radio"/> První <input type="radio"/> Druhý	
Zpřístupnit studentům:	Ano	
Vložit soubor	<input type="text"/>	<input type="button" value="Procházet.."/>
<input type="button" value="Přidat dokument"/>		

Obrázek 10 - Přidání dokumentu - formulář Zdroj: vlastní

### 3.5.9 Vypsání dokumentů

K vypsání údajů o dokumentech z databáze slouží příkazy jazyka PHP a dotazy na databázi provedené pomocí SQL. Příkazy PHP nastaví řádek v databázové tabulce na první a dále pomocí dotazu SQL zapíše do tabulky připravené na údaje. Ta se pak zobrazí uživateli na monitoru. Tabulka má tolik řádků, kolik je záznamů v databázové tabulce. Uživatel si pak přímo z tabulky může zobrazit požadovaný soubor. Je zde také možnost úpravy nebo smazání daného záznamu, to však smějí udělat jen uživatelé s oprávněním. Uživatelům, kteří oprávnění nemají, se možnosti nezobrazí.

Název dokumentu	Popis	Klíčová slova	Datum uložení	Předmět	Stupeň	Soubor		
Vyjmenovaná slova	slova na zapamatování tvrdého y	vyjmenovaná slova	2010-06-24	Český jazyk	První		<input type="button" value="upravit"/>	<input type="button" value="smazat"/>
Stínování	práce tužkou	stínování tužka	2010-06-24	Výtvarná výchova	Druhý		<input type="button" value="upravit"/>	<input type="button" value="smazat"/>
Bulharsko	mapa	mapa Bulharsko	2010-06-24	Zeměpis	Druhý		<input type="button" value="upravit"/>	<input type="button" value="smazat"/>

Obrázek 11 - Výpis dokumentů do tabulky Zdroj: vlastní

### 3.5.10 Vyhledávání dokumentů

V subjektu, kde běží systém na centrální správu dokumentů, bývá velké množství uložených dokumentů. Proto je vhodné umístit do systému formulářový prvek navyhledání zadaného dokumentu. V tomto případě je vyhledávání umístěno v každé sekci nad tabulkou s vypsanými záznamy.

Vyhledávání dokumentů funguje na základě dotazování databáze pomocí SQL dotazu. Tento dotaz vybírá z databáze každý atribut, který odpovídá hledanému výrazu, i kdyby to bylo jen jedno písmeno. Jako atributy pro vyhledávání jsem zvolila název dokumentu, popis dokumentu a klíčová slova. Takto vypadá SQL dotaz u tabulky Výuka.

```
$sql = "SELECT * FROM vyuka
      WHERE nazev LIKE '%$_POST[vyraz]%'
      OR popis LIKE '%$_POST[vyraz]%'
      OR klic LIKE '%$_POST[vyraz]%' ";
```

Název dokumentu	Popis	Klíčová slova	Datum uložení	Předmět	Stupeň	Soubor		
Vyjmenovaná slova	slova na zapamatování tvrdého y	vyjmenovaná slova	2010-06-24	Český jazyk	První		<input type="button" value="upravit"/>	<input type="button" value="smazat"/>
Stínování	práce tužkou	stínování tužka	2010-06-24	Výtvarná výchova	Druhý		<input type="button" value="upravit"/>	<input type="button" value="smazat"/>

Obrázek 12 - Vyhledané dokumenty Zdroj: vlastní

### 3.5.11 Úprava dokumentu

Každému se může stát, že se zmýlí, proto tato aplikace počítá s případnou úpravou uložených dat. K tomu mají přístup pouze oprávnění uživatelé. Po zadání příkazu „Upravit“ se uživateli otevře formulář s daty, která jsou uložena v databázi. Uživatel může

opravit veškeré atributy a znovu data uložit. V databázi se data přepíše díky skrytému prvku formuláře, udávající identifikační číslo dokumentu.

## Úprava dokumentu

Název dokumentu:	<input type="text" value="Stínování"/> *
Popis dokumentu:	<input type="text" value="práce tužkou"/> *
Klíčová slova:	<input type="text" value="stínování tužka"/> *
Předmět:	<input type="text" value="Výtvarná výchova"/> *
Stupeň:	<input type="radio"/> První <input checked="" type="radio"/> Druhý
Zpřístupnit studentům:	<input type="text" value="Ne"/> ▾
<input type="button" value="Uložit upravené hodnoty"/>	

Obrázek 13 - Úprava dokumentu - formulář Zdroj: vlastní

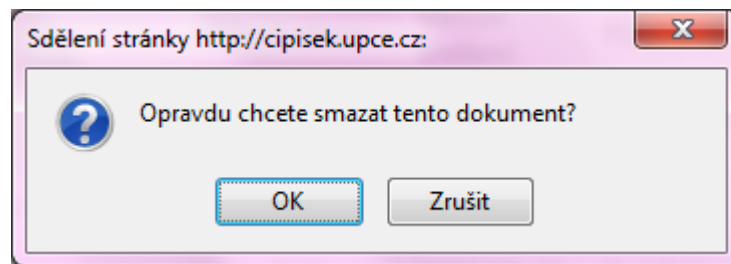
### 3.5.12 Smazání dokumentu

V každém systému se může stát, že je potřeba dokument smazat. Jedná se o případ, kdy uživatel omylem vloží soubor, který do dané sekce nepatří. Dále je potřeba ze systému smazat dokumenty neaktuální, zastaralé a nepotřebné. V subjektech, které využívají systémy na centrální správu dokumentů, jsou uplatňovány směrnice, jak dlouho dané dokumenty archivovat. Poté jsou ze systému smazány.

Smazání je z databáze zajištěno pomocí následujícího SQL dotazu, který podle identifikačního čísla smaže celý řádek v tabulce.

```
$sql = "DELETE FROM nazev_tabulky  
WHERE id = ". $_GET['id'] ."
```

Celý proces je ošetřen proti nechtěnému smazání vyskakovacím oknem, které se uživatele optá, zda chce dokument opravdu smazat. Po kladné odpovědi je dokument smazán.



Obrázek 14 - Dotaz před smazáním dokumentu Zdroj: vlastní

### 3.5.13 Zálohování

V aplikaci není zálohování řešeno samostatně do hloubky. Pokud by bylo potřeba zálohovat, databáze MySQL nabízí zálohování na úrovni databáze pomocí nástroje mysqldump.

### 3.5.14 Shrnutí praktické části

Výsledná aplikace je připravena podle konkrétních požadavků organizace, které jsou uvedeny v kapitole 3.1. V aplikaci nejsou řešeny některé administrační možnosti, například změnění práv uživatelů nebo smazání uživatele. Drobnými zásahy do zdrojového kódu lze například pevně dané kategorie upravit. Právo k zásahům má pouze programátor. Jedná se o plně funkční aplikaci splňující požadavky organizace na systém centrální správy dokumentů.

## **Závěr**

Hlavním cílem bakalářské práce bylo navrhnout funkční aplikaci pro systém centrální správy dokumentů. Hlavní důraz je zde kladen na praktickou část. Kromě té je část práce věnována také teoretickým základům skriptovacího jazyka PHP a databázovému systému MySQL.

V teoretické části je rozebráno, co je to centrální správa dokumentů a dále pak teoretické základy PHP a MySQL. Jelikož je tato problematika velmi rozsáhlá, byla vybrána jen tematika, která se blíže týkala zpracovávaného tématu, hlavně praktické části. V té je podrobněji popsána vlastní tvorba aplikace, od návrhu E-R modelu až po vlastní realizaci aplikace.

V textu jsou přidány i doporučené diagramy (E-R diagram, Use Case diagram), obrázky přímo z vytvořené aplikace a úryvky zdrojových kódů a funkcí, kterými je tato aplikace tvořena. Výsledná aplikace je k této práci přiložena na CD.

## Použitá literatura

- [1] *Printsoft* [online]. 2009 [cit. 2010-06-11]. Řešení správy dokumentů.  
Dostupné z WWW:  
<[http://www.printsoft.cz/reseni/document\\_management.htm](http://www.printsoft.cz/reseni/document_management.htm)>.
- [2] PHP Manual : What is PHP [online]. 2010 [cit. 2010-06-28]. Dostupné z WWW:  
<<http://www.php.net/manual/en/intro-what-is.php>>.
- [3] ZAJÍC, Petr PHP : Typy proměnných [online] 2004 [cit. 2010-06-27].  
Dostupné z WWW: <[http://www.linuxsoft.cz/article.php?id\\_article=178](http://www.linuxsoft.cz/article.php?id_article=178)>.
- [4] PHP Manual : Proměnné [online]. 2010 [cit. 2010-06-28]. Dostupné z WWW:  
<<http://www.php.net/manual/en/language.variables.basics.php>>.
- [5] PHP : Funkce. In *Tvorba webu* [online]. 2006 [cit. 2010-06-28]. Dostupné z WWW:  
<<http://www.tvorba-webu.cz/php/funkce.php>>.
- [6] DELISLE, Marc. *PhpMyAdmin : efektní správa MySQL*. první. Brno : Zoner Press, 2004. 264 s. ISBN 80-86815-09-9.
- [7] MySQL Manual : `mysql_connect` [online]. 2010 [cit. 2010-06-28].  
Dostupné z WWW:  
<<http://www.php.net/manual/en/function.mysql-connect.php>>.
- [8] MySQL Manual : `mysql_select_db` [online]. 2010 [cit. 2010-06-28].  
Dostupné z WWW:  
< <http://www.php.net/manual/en/function.mysql-select-db.php>>.
- [9] MySQL Manual : `mysql_query` [online]. 2010 [cit. 2010-06-28]. Dostupné z WWW:  
<<http://www.php.net/manual/en/function.mysql-query.php>>.
- [10] MySQL Manual : `mysql_fetch_row` [online]. 2010 [cit. 2010-06-28].  
Dostupné z WWW:  
< <http://www.php.net/manual/en/function.mysql-fetch-row.php>>.

- [11] MySQL Manual : `mysql_fetch_array` [online]. 2010 [cit. 2010-06-28]. Dostupné z WWW: <<http://www.php.net/manual/en/function.mysql-fetch-array.php>>.
- [12] MySQL Manual : `mysql_close` [online]. 2010 [cit. 2010-06-28]. Dostupné z WWW: <<http://www.php.net/manual/en/function.mysql-close.php>>.
- [13] MATIAŠKO, Karol. *Databázové systémy*. Žilina : EDIS, 2002. 367 s. ISBN 80-7100-968-7.
- [14] STRAKA , Miroslav. *Vývoj databázových aplikací*. Praha : Grada, 1993. 129 s. ISBN 80-85424-43-6.
- [15] LACKO, Luboslav. *Web a databáze: programujeme internetové aplikace*. Praha: Computer Press, 2001. 250 s. ISBN 80-7226-555-5.
- [16] HERNANDEZ, Michael J. *Návrh databází*. Praha : Grada Publishing a.s., 2006. 408 s. ISBN 80-247-0900-7.
- [17] WELLING, Luke, THOMSON, Laura. *PHP a MySQL : rozvoj webových aplikací*. Praha : SoftPress, c2004. 910 s.
- [18] Teorie relačních databází : Relační model dat. In *Databaze* [online]. [s.l.] : [s.n.], 2006 [cit. 2010-06-27]. Dostupné z WWW: <<http://www.manualy.net/article.php?articleID=9>>.
- [19] KUČEROVÁ, Helena. *Use Case model* [online]. [s.l.], 2010. Vyšší odborná škola informačních služeb, Praha 4, Pacovská 350/4. Dostupné z WWW: <<http://web.sks.cz/users/ku/pri/usecase.htm>>.
- [20] STEIN, René. *Návrh aplikací v jazyce UML : začínáme s případy užití*. *Interval.cz* [online]. 2004, [cit. 2010-06-19]. Dostupný z WWW: <<http://interval.cz/clanky/navrh-aplikaci-v-jazyce-uml-zaciname-s-pripady-uziti/>>.
- [21] ŠKULTÉTY, Rastislav. *Java Script : Programujeme internetové aplikace*. 2. aktualiz. vyd. Brno : Computer Press, 2004. 224 s. ISBN 80-251-0144-4.

## Seznam tabulek

Tabulka 1 - Rozbor typů dokumentů .....	19
Tabulka 2 - Atributy na vyhledávání .....	19
Tabulka 3 - Entity a jejich atributy 1 .....	20
Tabulka 4 - Entity a jejich atributy 2 .....	21
Tabulka 5 - Řádek databázové tabulky .....	23
Tabulka 6 - Sloupec databázové tabulky .....	23
Tabulka 7 - Primární klíč .....	24
Tabulka 8 - Popis tabulky výuka .....	29
Tabulka 9 - Popis tabulky uživatelé .....	29
Tabulka 10 - Popis tabulky Management .....	30

## Seznam obrázků

Obrázek 1 – Náhled aplikace phpMyAdmin .....	15
Obrázek 2 - E-R diagram .....	22
Obrázek 3 - Příklad užití .....	25
Obrázek 4 - Symbol aktéra .....	25
Obrázek 5 - Vazba asociace .....	26
Obrázek 6 - Use Case diagram .....	27
Obrázek 8 - Rozložení aplikace .....	32
Obrázek 9 - Registrace uživatelů - formulář .....	33
Obrázek 10 - Přihlášení - formulář.....	34
Obrázek 11 - Přidání dokumentu - formulář .....	36
Obrázek 12 - Výpis dokumentů do tabulky .....	37
Obrázek 13 - Vyhledané dokumenty .....	37
Obrázek 14 - Úprava dokumentu - formulář .....	38
Obrázek 15 - Dotaz před smazáním dokumentu .....	39

## **Seznam příloh:**

Příloha A	Výsledná aplikace – úvodní stránka .....	A
Příloha B	Výsledná aplikace – nepřihlášený uživatel.....	B
Příloha C	Výsledná aplikace – uživatel Student .....	C
Příloha D	Výsledná aplikace – uživatel Vyučující .....	D
Příloha E	Výsledná aplikace – uživatel Administrátor .....	E
Příloha F	Datový nosič s aplikací	

## Příloha A Výsledná aplikace – úvodní stránka



# CENTRÁLNÍ SPRÁVA DOKUMENTŮ

Přihlášení:

- VÝUKA
- MANAGEMENT
- PROVOZ
- PROJEKTY
- PREZENTACE
- PROGRAM
- DDM
- DRUŽINA

### Vítejte ve webové aplikaci na centrální správu dokumentů.

V této aplikaci je možné archivovat vešketé typy dokumentů spojené s chodem školského subjektu. Je možné dokumenty přidávat, upravovat, vyhledávat a mazat. Pro plnohodnotnou funkčnost aplikace se, prosím, přihlašte. Pokud nejste zaregistrovaným uživatelem, obraťte se na Vašeho administrátora, který vás ochotně zaregistruje.

Dokumenty jsou rozděleny do kategorií, podle jejich obsahu. Tyto kategorie jsou umístěny v menu nalevo. Zde si můžete tyto dokumenty zobrazit.

## Příloha B Výsledná aplikace – nepřihlášený uživatel



**CENTRÁLNÍ SPRÁVA DOKUMENTŮ**

Přihlášení:

Název dokumentu	Popis	Klíčová slova	Datum uložení	Předmět	Stupeň
Vyjmenovaná slova	slova na zapamatování tvrdého y	vyjmenovaná slova	2010-06-25	Český jazyk	Druhý
Bulharsko	mapa	mapa Bulharsko	2010-06-24	Zeměpis	Druhý

**VÝUKA**

**MANAGEMENT**

**PROVOZ**

**PROJEKTY**

**PREZENTACE**

**PROGRAM**

**DDM**

**DRUŽINA**

## Příloha C Výsledná aplikace - uživatel Student



# CENTRÁLNÍ SPRÁVA DOKUMENTŮ

Byl jste úspěšně přihlášen student [Odhlásit](#)

VÝUKA

MANAGEMENT

PROVOZ

PROJEKTY

PREZENTACE

PROGRAM

DDM

DRUŽINA

Název dokumentu	Popis	Klíčová slova	Datum uložení	Předmět	Stupeň	Soubor
Vyjmenovaná slova	slova na zapamatování tvrdého y	vyjmenovaná slova	2010-06-25	Český jazyk	Druhý	
Bulharsko	mapa	mapa Bulharsko	2010-06-24	Zeměpis	Druhý	

## Příloha D Výsledná aplikace – uživatel Vyučující



# CENTRÁLNÍ SPRÁVA DOKUMENTŮ

Byl jste úspěšně přihlášen vyučující [Odhlásit](#)

VÝUKA

MANAGEMENT

PROVOZ

PROJEKTY

PREZENTACE

PROGRAM

DDM

DRUŽINA

[Vyhledat](#)

[Přidat dokument](#)

Název dokumentu	Popis	Klíčová slova	Datum uložení	Předmět	Stupeň	Soubor		
Vyjmenovaná slova	slova na zapamatování tvrdého y	vyjmenovaná slova	2010-06-25	Český jazyk	Druhý		<a href="#">upravit</a>	<a href="#">smazat</a>
Stínování	práce tužkou	stínování tužka	2010-06-24	Výtvarná výchova	Druhý		<a href="#">upravit</a>	<a href="#">smazat</a>
Bulharsko	mapa	mapa Bulharsko	2010-06-24	Zeměpis	Druhý		<a href="#">upravit</a>	<a href="#">smazat</a>

## Příloha E Výsledná aplikace - uživatel Administrátor



# CENTRÁLNÍ SPRÁVA DOKUMENTŮ

Byl jste úspěšně přihlášen admin [Odhlásit](#) [Registrovat](#)

VÝUKA

MANAGEMENT

PROVOZ

PROJEKTY

PREZENTACE

PROGRAM

DDM

DRUŽINA

[Vyhledat](#) [Přidat dokument](#)

Název dokumentu	Popis	Klíčová slova	Datum uložení	Předmět	Stupeň	Soubor		
Vyjmenovaná slova	slova na zapamatování tvrdého y	vyjmenovaná slova	2010-06-25	Český jazyk	Druhý		<a href="#">upravit</a>	<a href="#">smazat</a>
Stínování	práce tužkou	stínování tužka	2010-06-24	Výtvarná výchova	Druhý		<a href="#">upravit</a>	<a href="#">smazat</a>
Bulharsko	mapa	mapa Bulharsko	2010-06-24	Zeměpis	Druhý		<a href="#">upravit</a>	<a href="#">smazat</a>