

Univerzita Pardubice – Fakulta elektrotechniky a informatiky

Semestrální práce z DAS2 a WWW

Aplikační část

Matěj Trakal
25.12.2009

Kapitola 1: Obsah

KAPITOLA 1: OBSAH	2
KAPITOLA 2: ZÁKLADNÍ CHARAKTERISTIKA	3
PODKAPITOLA 1: DOSTUPNOST SEMESTRÁLNÍ PRÁCE	3
PODKAPITOLA 2: ZVOLENÉ TÉMA	3
PODKAPITOLA 3: PRO KOHO	3
PODKAPITOLA 4: UML USE CASE DIAGRAM	3
PODKAPITOLA 5: UML RICH PICTURE DIAGRAM	4
PODKAPITOLA 6: UML ACTIVITY DIAGRAM	4
KAPITOLA 3: IMPLEMENTOVANÉ ČÁSTI	5
PODKAPITOLA 1: FUNKČNOST	5
PODKAPITOLA 2: VALIDITA (X)HTML	5
PODKAPITOLA 3: VALIDITA CSS	5
PODKAPITOLA 4: LAYOUT STRÁNEK	5
PODKAPITOLA 5: ZOBRAZENÍ V PROHLÍŽEČÍCH	5
PODKAPITOLA 6: WYSIWYG EDITOR	6
PODKAPITOLA 7: OPTIMALIZACE PRO TISK	6
PODKAPITOLA 8: ZABEZPEČENÝ PŘÍSTUP	6
PODKAPITOLA 9: EXPORT DO XML	6
KAPITOLA 4: ADRESÁŘOVÁ STRUKTURA	7
KAPITOLA 5: UKÁZKA ZDROJOVÝCH KÓDŮ	8
PODKAPITOLA 1: PRÁCE S DATABÁZÍ	8
PŘIPOJENÍ	8
UKÁZKA VÝBĚRU DAT Z DATABÁZE	8
PODKAPITOLA 2: PŘÍSTUP K ADMINISTRACI	8
PŘIHLÁŠENÍ UŽIVATELE	8
OVĚŘENÍ PŘIHLÁŠENÉHO UŽIVATELE	9
PODKAPITOLA 3: PARSOVÁNÍ XML SOUBORU	9
KAPITOLA 6: SEZNAM OBRÁZKŮ	11

Kapitola 2: Základní charakteristika

Podkapitola 1: Dostupnost semestrální práce

Díky využití SubVersion je semestrální práce dostupná v podobě open-source zdrojových kódů umístěných na službě Google code.

Zdrojové kódy: <http://code.google.com/p/wwwidas2-semestralka/source/browse/>

Postupné změny v jednotlivých revizích: <http://code.google.com/p/wwwidas2-semestralka/source/list>

Databázová část: http://wwwidas2-semestralka.googlecode.com/svn/trunk/_/Trakal_Matej_SP_IWWW_IDAS2.pdf

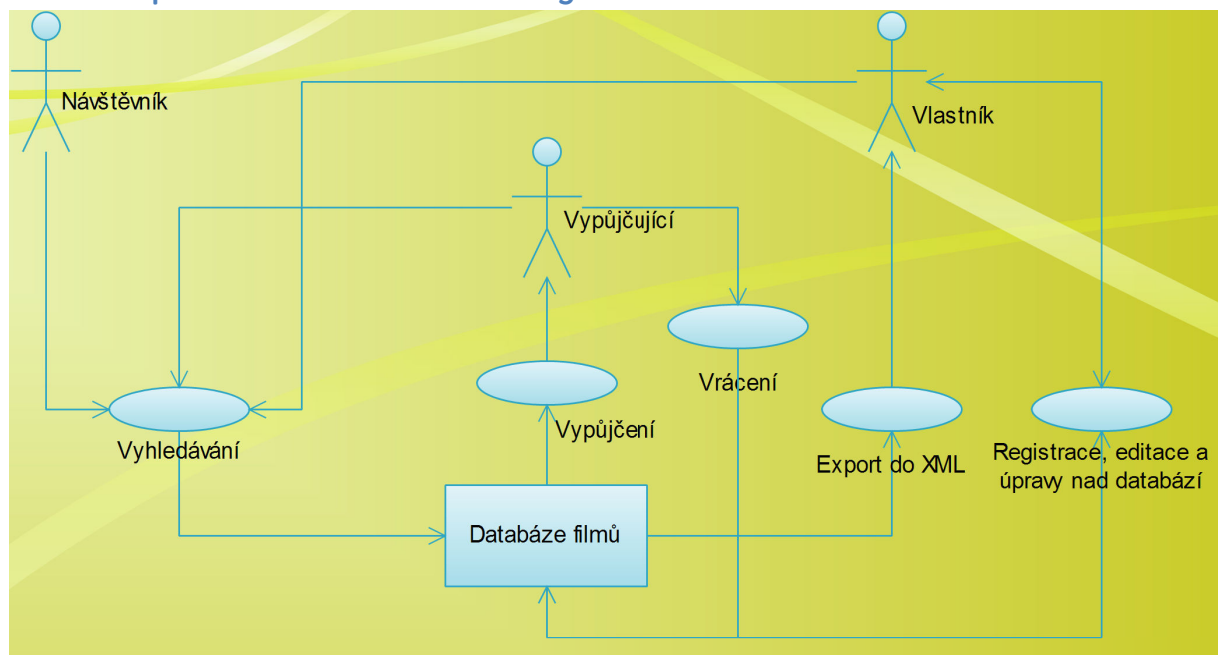
Podkapitola 2: Zvolené téma

Pro semestrální práci jsem si zvolil tvorbu filmové databáze v plánovaném rozsahu asi 5 tabulek a jen nepatrném množství webových stránek. Bohužel rozsah se výrazně zvětšil postupným zjišťováním návazností a potřeb dělení a zachování základních databázových norem. Tím se zvětšoval i rozsah potřebné funkcionality webové části, až nad rámec mého možného volného času a znalostí.

Podkapitola 3: Pro koho

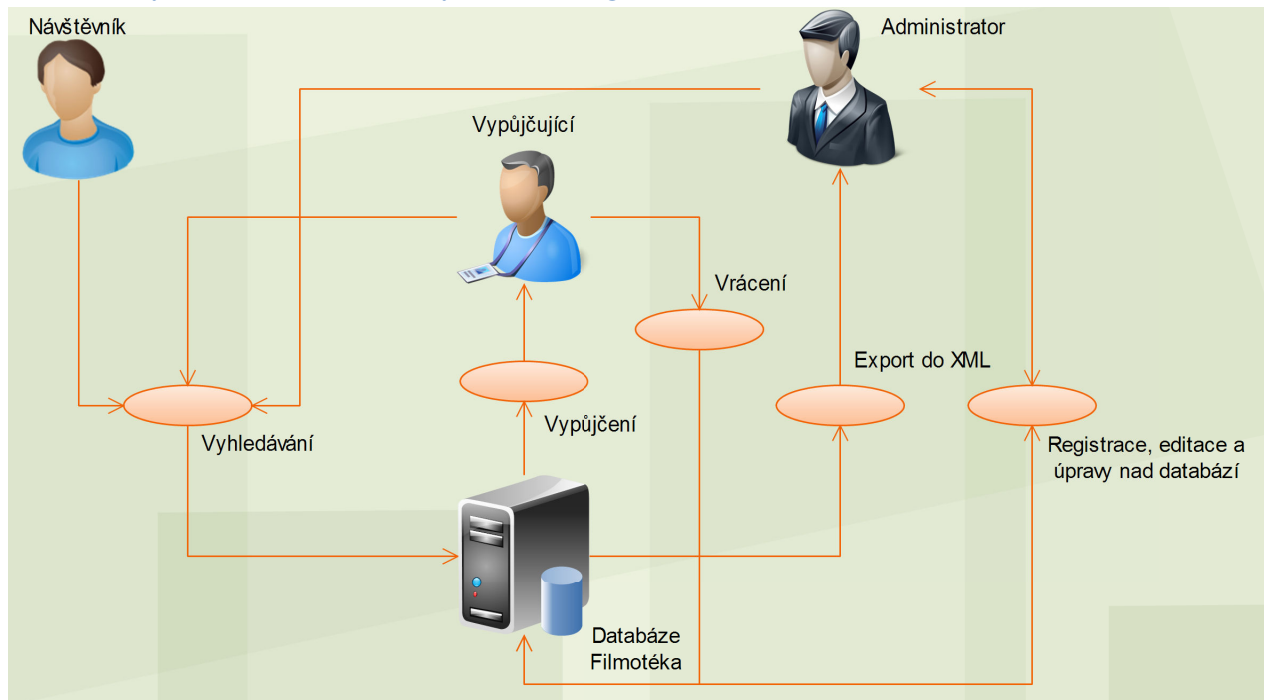
Filmotéka má sloužit pro jednoho uživatele vlastního domácí archiv filmů. Vlastník je hodný a tedy umožňuje svým známým výpůjčku jeho filmů. Měla být původně do jisté míry implementována i půjčovna.

Podkapitola 4: UML Use case diagram



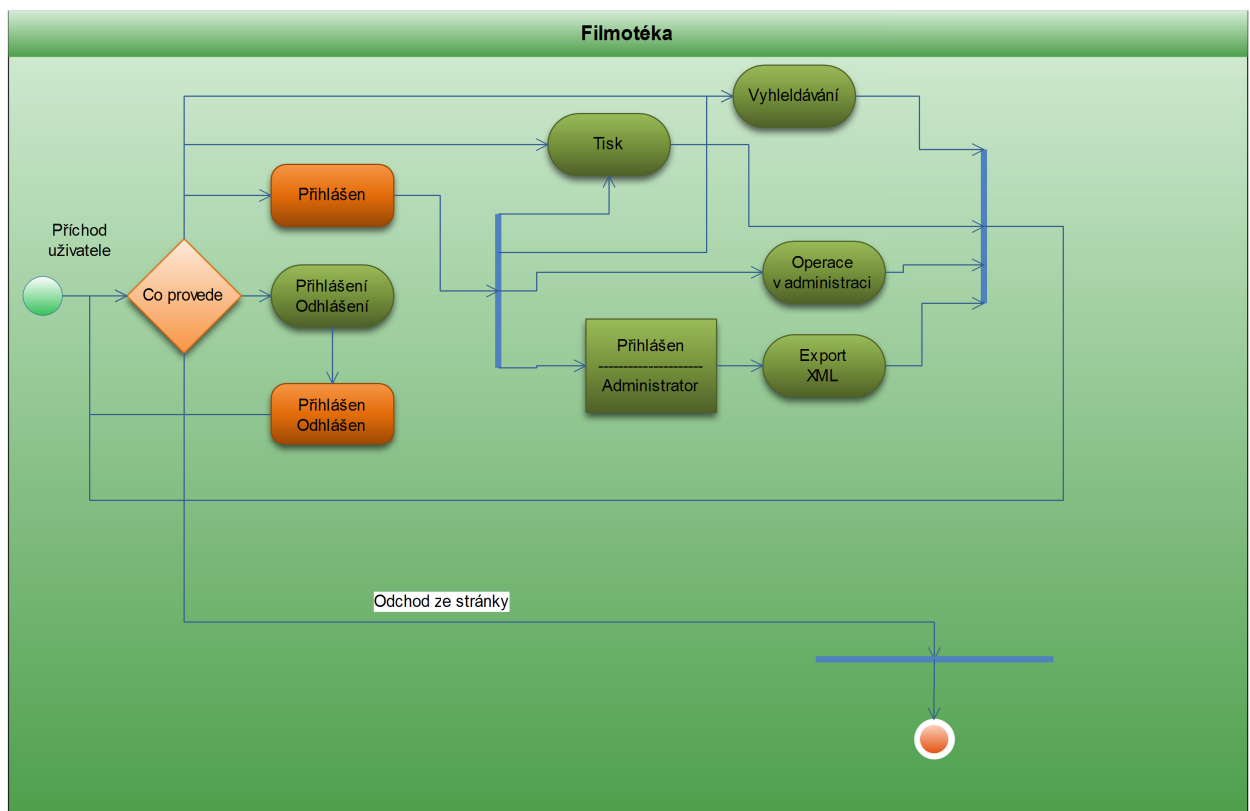
Obrázek 1: UML Use case diagram

Podkapitola 5: UML Rich picture diagram



Obrázek 2: UML Rich picture diagram

Podkapitola 6: UML Activity diagram



Obrázek 3: UML Activity diagram

Kapitola 3: Implementované části

Podkapitola 1: Funkčnost

V nynější rozpracované podobě lze z databáze vypisovat pouze tituly, a ne filmy jako takové. Pro ukázkou návrhu WWW stránek nynější podoba bohatě stačí a část databázová je splněna celkově, pouze není implementována do aplikace.

Je dostupné vyhledávání v databázi (obsahuje pouze 4 filmy, tedy real-timové vyhledávání funguje na slova: ame..., ces..., res... a evi...). Vyhledávání funguje tak, že se pomocí ajaxu posílají dotazy na databázi (po zadání minimálně 3 znaků) a vyhledává se v originálním, českém a anglickém názvu filmu. Jako výsledek se zobrazuje opět originální, CZ a EN název filmu + část popisu. Z výsledků je možné kliknout na název a zobrazit si o filmu podrobnosti.

Po přihlášení se administrátorovi zpřístupní na stránce s vyhledáváním export do XML souboru, který je generován pomocí databáze a dále zpracován pomocí PHP, díky kterému je ho možné uložit (zaslání speciálního headeru).

Dále je možné v administrační části přidávat filmy, uživatele, ... ovšem už ne je upravovat nebo mazat (opět jde spíše již o databázovou část).

Za vypůjčujícího uživatele je implementováno vše co za administrátora, jen s tím rozdílem, že v aktuální podobě nemá možnost exportovat filmovou databázi do XML.

Podkapitola 2: Validita (x)HTML

Stránky jsou validní dle konsorcia W3C a jejich doporučení (stále je pouze doporučením a ne standardem) pro návrh a tvorbu WWW stránek v XHTML 1.0 Strict s tím, že je zasílán i správný mine-type a kódování v hlavičce (vyjma Internet Exploreru, který application/xhtml+xml není schopen zpracovat, jako WWW stránku a obsah dává ke stažení, tedy je zaslán záměrně chybný mine-type).

Podkapitola 3: Validita CSS

Jelikož využívám prvků průhlednosti v CSS souborech (dostupných od CSS 3.0), nejsou tyto soubory validní. Pokud bych nechtěl využívat proprietárních technologií (-moz, filter a opacity) z prohlížečů, validní by samozřejmě byly.

Podkapitola 4: Layout stránek

Samozřejmostí při návrhu je netabulkový layout využívající výhod kontejneru DIV. Díky tomuto kontejneru je stránka schopna se vykreslovat i v době, kdy ještě nejsou načteny všechny dostupné obrázky a je tedy skládána postupně, což tabulkový layout neumožňoval (muselo se čekat na rozměry všech vnořovaných prvků a až následně bylo možné tabulky vykreslit).

Podkapitola 5: Zobrazení v prohlížečích

Zobrazení obsahu stránek je shodné v testovaných prohlížečích, tedy v Internet Explorer 8 (Trident), Opera 10.10 (Presto 2.2), Firefox portable 3.5.5 (Gecko 20091102) a Google Chrome 1.0.154.48 (WebKit).

Podkapitola 6: WYSIWYG editor

Je implementován v administrační části dle zadání. Přesněji při přidávání titulu filmu. Použitý editor je stylovací šablona *Texy* (<http://texy.info/cs/>) a její nastavba v podobě *Texyla* (<http://texyla.janmarek.net/>) který funguje jako grafická nastavba *Texy*.

Podkapitola 7: Optimalizace pro tisk

Je zajištěna pomocí druhotného kaskádového stylu a odkazu na něj v hlavičce s parametrem *media="screen"*.

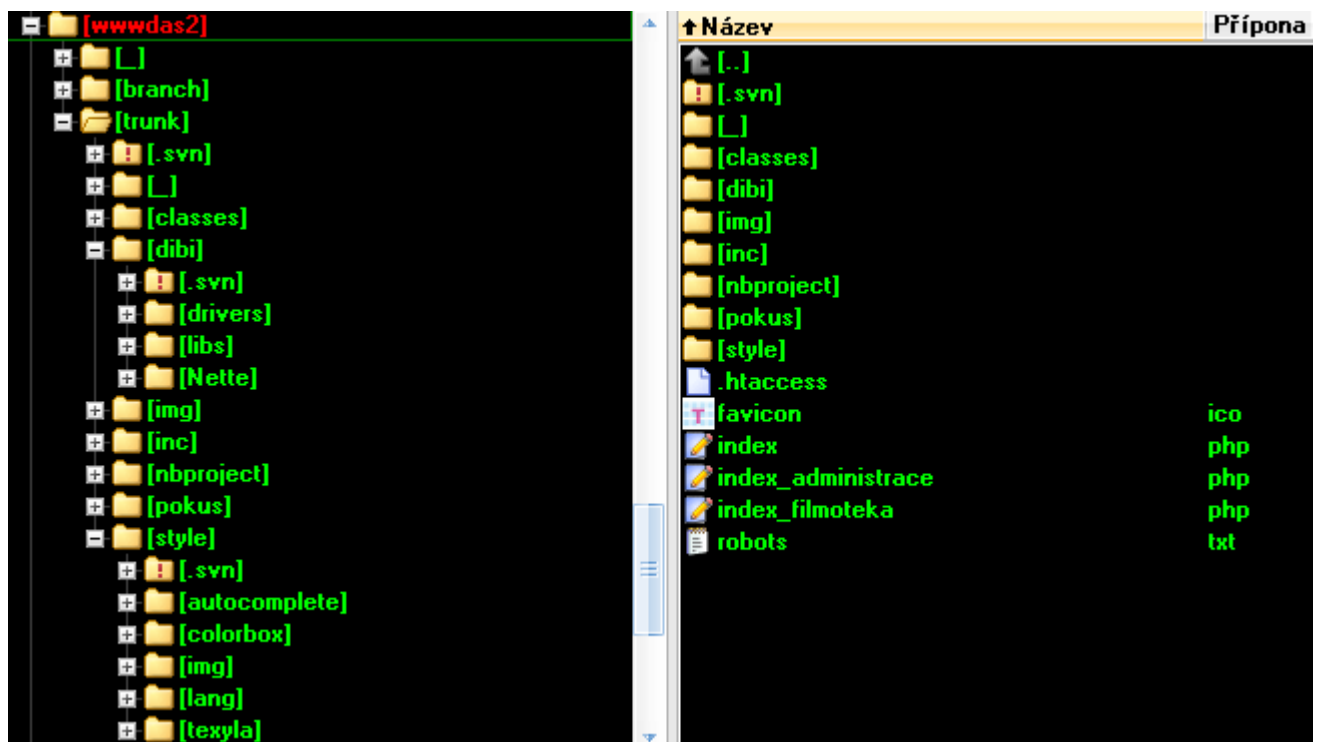
Podkapitola 8: Zabezpečený přístup

Je implementován do administrace, hlavní využití a ukázka je při využití exportu do souboru, který bez přihlášení za administrátora nelze provést. Soubory administrace jsou po zadání odkazu do prohlížeče normálně dostupné, jelikož jsou ve fázi rozpracování a finální podoba měla využívat výhod OOP, kde by nemuselo k ošetření dojít, jelikož by uživatel z „venku“ neměl možnost zavolat zapouzdřené funkce.

Podkapitola 9: Export do XML

Export je využit pro vybrání všech titulů z databáze. Ukázka je dostupná po přihlášení na stránce *Vyhledávání*.

Kapitola 4: Adresářová struktura



Obrázek 4: Adresářová struktura

Kapitola 5: Ukázka zdrojových kódů

Podkapitola 1: Práce s databází

Připojení

```
public function __construct() {
    try {
        dibi::connect(array(
            'driver' => 'oracle',
            'database' => 'sql101.upceucebny.cz:1521/oracle10',
            'username' => 'st22312',
            'password' => HESLO,
            'charset' => 'UTF8'
        ));
    } catch (DibiException $e) {
        echo get_class($e), ': ', $e->getMessage(), "\n";
    }
}

public function __destruct() {
    try {
        dibi::disconnect();
    } catch (DibiException $e) {
        echo get_class($e), ': ', $e->getMessage(), "\n";
    }
}
```

Ukázka výběru dat z databáze

```
/**
 * Informace o filmu
 * @param <integer> $id
 * @return <Array[][]> informace o vybraném filmu
 */
public function MovieGet($id) {
    $result = null;
    $row = array();
    if(dibi::isConnected()) {
        $result= dibi::query("select film_id as IDFILMU, cz as CZ, en as EN, original as ORIGINAL, csfd as
CSFD, imdb as IMDB, rok_vydani as ROK, delka as DELKA, popis as POPIS from ttitul where
film_id=".$id);
        $row = $result->fetchAll();
        return $row;
    }
}
```

Podkapitola 2: Přístup k administraci

Přihlášení uživatele

```
/**
 * Přihlášení uživatele a ověření skrz databázi
 * @param <string> $user
 * @param <string> $password
```



```
* @return <boolean> přihlášen úspěšně?  
*/  
public function Authorize($username, $password) {  
  
    $this->username = $username;  
    $this->password = sha1($password) . md5(strlen($password));  
    try {  
        $result = null;  
        $db = new Oci8();  
        $result = $db->UserAuth($this->username, $this->password);  
        unset($db);  
  
        if(($result != null) && ($this->username = $result[0]['USERNAME'] && ($this->password =  
$result[0]['PASSWORD']))) {  
            $_SESSION['login']=true;  
            $_SESSION['username']=$result[0]['USERNAME'];  
            $_SESSION['role'] = $result[0]['ROLE'];  
            return true;  
        } else {  
            return false;  
        }  
  
    } catch (DibiException $e) {  
        echo get_class($e), ': ', $e->getMessage(), "\n";  
        return false;  
    }  
}
```

Ověření přihlášeného uživatele

```
if($login->IsAuthorized() && isset($_SESSION['role'])) {  
    if($_SESSION['role'] == 'Administrator') {  
        // vlastní kód  
    }  
}
```

Podkapitola 3: Parsování XML souboru

```
class XMLParse {  
    public function MovieGet() {  
        header('Content-Disposition: attachment; filename="seznam-titulu.xml"');  
        header("Pragma: public"); // požadováno  
        header("Expires: 0");  
        header("Cache-Control: must-revalidate, post-check=0, pre-check=0");  
        header("Cache-Control: private", false); // požadováno u některých prohlížečů  
        header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");  
        header("Content-type: text/xml");  
        header("Content-Transfer-Encoding: binary");  
  
        echo '<?xml version="1.0" encoding="UTF-8"?>'. "\n";  
  
        $result = array();  
        $db = new Oci8();  
        $result = $db->MovieParseXML();  
        foreach ($result as $variable) {
```

```
    echo $variable['XML']."\n";  
};  
unset($result, $db);  
}  
}
```

Kapitola 6: Seznam obrázků

OBRÁZEK 1: UML USE CASE DIAGRAM	3
OBRÁZEK 2: UML RICH PICTURE DIAGRAM	4
OBRÁZEK 3: UML ACTIVITY DIAGRAM	4
OBRÁZEK 4: ADRESÁŘOVÁ STRUKTURA	7