

UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

Databáze pro správu portrétů  
Jiří Vácha

Bakalářská práce  
2014

**Místo pro zadání strana 1**

**Místo pro zadání strana 2**

**Prohlášení autora**

Prohlašuji, že jsem tuto práci 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 4. 12. 2014.

Jiří Vácha

## **Poděkování**

Rád bych poděkoval panu Mgr. Tomáši Hudcovi za čas, který mi věnoval při vedení mé bakalářské práce a za cenné rady, které vedly k jejímu dokončení. Dále bych chtěl poděkovat rodičům, kamarádům a přítelkyni za trpělivost a podporu během studia.

## **Anotace**

Práce se zabývá implementací funkční aplikace pro správu portrétů, která využívá sdílené autentizace, pomocí serverů třetích stran. Pojednává o bezpečnostních rizikách a obranou proti nim. Aplikace je typu CMS, takže umožňuje spravovat portréty a postavy bez znalosti programování. Pro návrh systému jsou použity technologie HTML, CSS 3, PHP a MySQL.

## **KLÍČOVÁ SLOVA**

sdílená identita, MySQL, HTML, CSS 3, PHP, správa portrétů

## **TITLE**

Database for portrait management

## **ANNOTACION**

The work implements the application for management of portraits, that uses shared authentication covering third-party servers. It discusses the security risks and defenses against them. The application is a type of CMS, allowing the management of portraits and characters without any programming knowledge. Used technologies: HTML, CSS 3, PHP, and MySQL.

## **KEYWORDS**

shared identity, MySQL, HTML, CSS 3, PHP, management of portrait

## Seznam zkratek

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
CD	Compact Disc
SQL	Structured Query Language
IDP	Identity Provider
SP	Service Provider
UA	User Agent
MySQL	My Structured Query Language
CMS	Content managemet systém
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol – Secure
SSL	Secure Sockets Layer

# Obsah

0. Úvod.....	11
1. Teoretická část .....	12
1.1. CMS – Systém pro správu obsahu .....	12
1.2. Normalizace.....	14
1.3. Normální formy .....	14
1.4. MySQL .....	18
1.5. Autentizace ve webových aplikacích .....	19
1.6. Sdílená identita .....	20
1.7. Řešení sdílené identity.....	21
2. Praktická část .....	25
2.1. Návrh databáze pro aplikaci .....	25
2.2. Úvod aplikace .....	27
3. Závěr .....	33



## Seznam obrázků

Obrázek 1 – Příklad CMS – Wordpress, zdroj: [10] .....	13
Obrázek 2 – Obecná ukázka sdílené identity, zdroj: vlastní.....	20
Obrázek 3 – Možnost řešení sdílené identity, zdroj: [9].....	22
Obrázek 4 – Schéma databáze, zdroj: vlastní .....	26
Obrázek 5 – Úvodní stránka aplikace, zdroj: vlastní .....	27
Obrázek 6 – Sekce vyhledávání, zdroj: vlastní.....	28
Obrázek 7 – Stránka odkazy, zdroj: vlastní .....	28
Obrázek 8 – Přihlášení do aplikace, zdroj: vlastní.....	29
Obrázek 9 – Výpis postav uživatele, zdroj: vlastní.....	30
Obrázek 10 – Vzhled konkrétní postavy, zdroj: vlastní.....	30
Obrázek 11 – Přidání portréту do systému, zdroj: vlastní.....	31
Obrázek 12 – Ukázka dialogu, zdroj: vlastní.....	32
Obrázek 13 – Ukázka kódu – tvorba dotazu vyhledávání, zdroj: vlastní.....	37

## Seznam tabulek

Tabulka 1 – Špatně navržená tabulka 0. formy .....	14
Tabulka 2 – Základní rozdělení dat do 1.NF .....	15
Tabulka 3 – Špatné řešení - rozšíření tabulky do šířky.....	15
Tabulka 4 – Zkrácená tabulka po rozdělení .....	15
Tabulka 5 – Data po rozdělení .....	16
Tabulka 6 – Špatné rozvržení relace při tvorbě 2. NF .....	16
Tabulka 7 – Tabulka rozdělená do 2. NF .....	17
Tabulka 8 – Tabulka rozdělená do 2. NF.....	17
Tabulka 9 – Počáteční návrh tabulky pro 3. NF .....	17
Tabulka 10 – Rozdělení tabulek v 3. NF .....	18
Tabulka 11 – Rozdělení tabulek v 3. NF .....	18

## 0. Úvod

Cílem mé bakalářské práce je vytvoření systému pro správu portrétů ke hře Neverwinter Nights. Celá aplikace bude navržena tak, aby byla co nejvíce uživatelsky přátelská a mohli ji ovládat i uživatelé bez jakékoliv znalosti programovacího jazyka. Aplikace bude umožňovat správu veškerých portrétů a postav uživatelů.

Práce je rozdělena do dvou částí. V první části se budu zabývat teorií návrhu databázového modelu. V této části popíši normalizaci a normální formy relační databáze. Dále se budu věnovat autentizaci uživatelů na serveru a blíže sdílené identitě a řešení jejího využití na konkrétním modelu.

Druhá část je praktická, zde se budu zabývat vývojem webové aplikace tvořené v jazyce HTML za využití skriptů PHP a jazyka CSS. V obou částech budou popsány funkce a prostředí aplikace, zvláště autentizace na serverech a práci s portréty uživatelů.

# 1. Teoretická část

## 1.1. CMS – Systém pro správu obsahu

Jedná se o software obsahující skupinu nástrojů a služeb umožňující správu webové aplikace bez znalostí programování. Pomocí těchto nástrojů můžeme například:

- upravovat, editovat a publikovat text,
- zpracovávat fotografie a tvořit galerie,
- spravovat statistiky,
- přidělování a úprava práv uživatelů,
- spravovat diskuse.

CMS je nejvhodnější pro organizace, které si chtějí samy upravovat svůj obsah a ušetřit tím peníze za správu webových stránek.

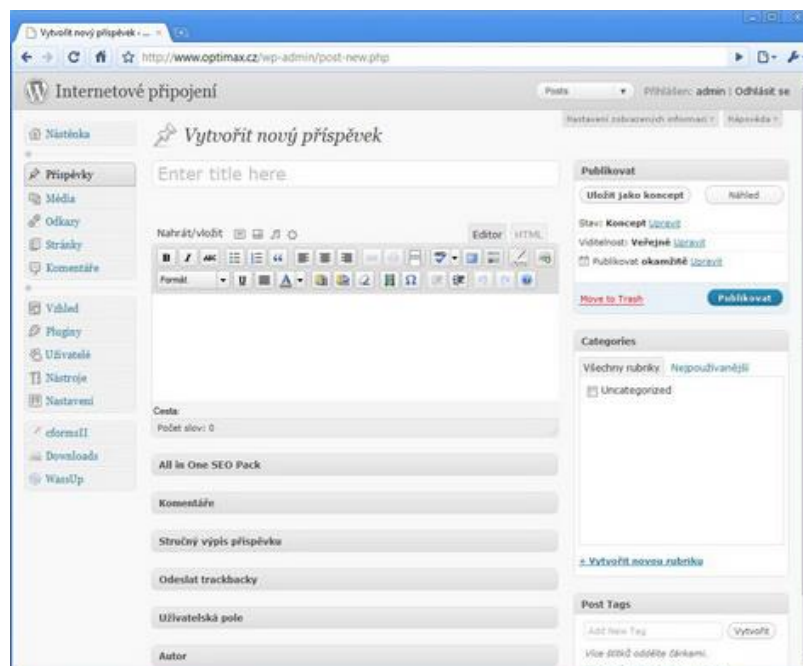
### Dostupné CMS

Na internetu lze sehnat již vytvořené CMS, které jsou bezplatně ke stažení, stačí je pouze nainstalovat na webový server a během chvíle tak lze zprovoznit kvalitní systém podle potřeby.

Volně ke stažení se dají najít systémy např. pro obchod, blog, fórum, kalendáře a lze je případně doplnit dalšími moduly.

### Příklad CMS - Wordpress

Wordpress je zástupce volně šiřitelných systémů CMS, uživatelé ho využívají nejčastěji pro publikování článků. Oblíbený je díky jednoduchosti instalace a používání. Snadno ho lze přizpůsobit do vzhledu, který si navrhne.



**Obrázek 1 – Příklad CMS – Wordpress, zdroj: [10]**

Aplikace popsaná v praktické části, umožňuje uživatelům vkládat a spravovat obsah webové prezentace. Můžeme ji tedy považovat za systém pro správu obsahu bez znalosti programování, tím pádem pro každého uživatele.

### Návrh databáze

Databáze je nezbytnou součástí složitějších aplikací, ve kterých potřebujeme uchovávat a později vyhledávat informace. Aby nám databáze byla prospěšná a s informacemi v ní uložené se nám pracovalo, je třeba myslet při jejím návrhu na množství a složitost uložených informací.

Nejčastějším typem databáze je relační databáze, která je založená na relačním modelu. Ten sdružuje data do relací, nebo-li tabulek, které obsahují řádky a sloupce s informacemi. Tabulky mají pevně daný formát sloupců. Každý sloupec může být jiný datový typ, to nám umožňuje ukládat např. text, čísla, datum atd., musí být však jednoznačně pojmenován a splňovat rozsah daného typu. Sloupce stejného typu z různých tabulek můžeme využít jako vazby mezi těmito tabulkami.

Navrhnout databázi do posledního sloupce je často nemožné a proto bychom měli myslet na možnost dalšího doplnění. Základními chybami při návrhu a pozdějším ukládání/vyhledávání jsou duplikátní data, která nám zbytečně zabírají místo, špatně se v nich vyhledává a těžko se mění. Setkáváme se proto s pojmem normalizace a normální formy.

## 1.2. Normalizace

Normalizace je proces, při kterém relace rozkládáme, abychom si usnadnili práci s daty, mohli s nimi lépe pracovat, zamezili jejich duplikaci (opakování) a udržovali konzistenci dat.

## 1.3. Normální formy

Normální formy jsou pravidla pro organizaci a uchovávání dat. Tato pravidla by data měla v relaci splňovat. Čím vyšší normální formu splňuje databáze, tím jednodušší je výběr a úprava dat v databázi. Normální formy jsou číslována od 0 do 5, kde každá vyšší normální forma v sobě obsahuje všechny nižší. Blíže se budeme věnovat pouze třem normám, jelikož jsou nejčastěji využívány.

### Normální forma 0

Relace je v nulté normální formě, pokud alespoň jeden jeho atribut obsahuje více než jednu hodnotu. Nesplňuje atomicitu – hodnota je dále nedělitelná.

**Tabulka 1 – Špatně navržená tabulka 0. formy**

Uživatel
Josef Novák Pardubice 777123456 777987654
Martin Novák Pardubice 605111111

Takto navržená tabulka je špatně z mnoha hledisek. Jen těžko budeme upravovat jednotlivé hodnoty, protože nevíme jak je přesně vybrat. Nelze použít výběr města nebo vyhledávat podle ulice, jelikož je hodnota dále dělitelná na jméno, příjmení, ulici, město, poštovní směrovací číslo, kraj a telefon. [8]

### Normální forma 1

V první normální formě se snažíme o rozdělení hodnot do více atributů, se kterými můžeme později lépe pracovat. Snažíme se tedy o atomicitu – hodnoty z pohledu databáze dále nedělitelné. [8]

**Tabulka 2 – Základní rozdělení dat do 1.NF**

Jméno	Příjmení	Město	Telefon
Josef	Novák	Pardubice	777123456, 777987654
Martin	Novák	Pardubice	605111111

Chceme-li u našich osob dále uchovávat více stejných informací v našem případě telefonních čísel, měli bychom přemýšlet o rozšíření naší relace. Špatnou variantou je rozšiřování tabulek do šířky.

**Tabulka 3 – Špatné řešení - rozšíření tabulky do šířky**

Jméno	Příjmení	Město	Telefon1	Telefon 2
Josef	Novák	Pardubice	777123456	777987654
Martin	Novák	Pardubice	605111111	

U řešení v tabulce 3 lze jen těžko odhadnout do kterého sloupce označeného telefon, jsme hodnotu opravdu uložili, navíc lze jen těžko uchovávat více jak dvě telefonní čísla. V tomto případě je třeba dále rozdělit relaci na dvě menší relace. Při rozkladu je třeba myslet na to, abychom se k datům dále dostali, zvolíme si proto primární klíč.

V příkladu je zvolený primární klíč jméno uživatele v praktickém použití je třeba myslet i na to, že databáze může obsahovat více uživatelů se stejným jménem. Pro náš příklad zatím stačí mít klíč jméno uživatele, který je v tabulce 5 cizím klíčem.

Po rozdělení relace na dvě menší dostáváme tento výsledek:

**Tabulka 4 – Zkrácená tabulka po rozdělení**

Jméno	Příjmení	Město
Josef	Novák	Pardubice
Martin	Novák	Pardubice

**Tabulka 5 – Data po rozdělení**

Jméno	Telefon
Josef	777123456
Josef	777987654
Martin	605111111

Nyní můžeme uchovávat více telefonních čísel u jednoho uživatele bez složitého vybírání z databáze a podmínek v našem kódu. [8]

### Normální forma 2

„Tabulka je v 2. NF, právě tehdy, když všechna data v tabulce závisí na celém primárním klíči“ [8]. Pro představu budeme pracovat s následujícím příkladem.

**Tabulka 6 – Špatné rozvržení relace při tvorbě 2. NF**

Název článku	Kategorie	Popis kategorie	Článek
Svět kolem nás	Cestování	Zážitky našich cestovatelů	...
Krkonoše a Sněžka	Cestování	Zážitky našich cestovatelů	...
Pes a kočka	Zvířata	Informace o zvířatech	...

Jako primární klíče byly zvoleny sloupce *Název článku* a *Kategorie*, aby byla možnost mít více článků se stejným názvem, ale v jiných kategoriích. Po vložení článků do relace je třeba myslet na to, že pokud bychom chtěli změnit popis kategorie, museli bychom projíždět všechny články v databázi, které jsou v kategorii Cestování a jednotlivě měnit popis kategorie. Tím zbytečně zatížíme server, na kterém běží databáze. Popis kategorie nám nezávisí na celém klíči ale pouze na jeho části – *Kategorii*.

Abychom tabulku dostali do druhé normální formy, rozdělíme ji na dvě menší relace. Z tabulky nám vypadne Popis kategorie a zůstane nám pouze primární klíč *Kategorie*.



Po úpravě dostáváme tyto tabulky:

**Tabulka 7 – Tabulka rozdělená do 2. NF**

Název článku	Kategorie	Článek
Svět kolem nás	Cestování	...
Krkonoše a Sněžka	Cestování	...
Pes a kočka	Zvířata	...

**Tabulka 8 – Tabulka rozdělená do 2. NF**

Kategorie	Popis kategorie
Cestování	Zážitky našich cestovatelů
Zvířata	Informace o zvířatech

Nyní nám již oprava popisku kategorie zabere pouze jeden dotaz provedený do databáze a při úpravě popisku kategorie ušetříme server o zbytečnou zátěž. Tímto rozdělením lze také předejít nesprávnosti (konzistenci) dat. V případě, že bychom upravovali popisky kategorií, kde může být i tisíce článků a došlo by k výpadku serveru, tabulka by mohla obsahovat články kde půlka má popisek starý a půlka nový. [8]

### Normální forma 3

K vysvětlení 3. normální formy využijeme rovnou příklad, abychom si to uměli představit.

Představme si, že chceme rozšířit příklad z 2. normální formy a u každého článku, chceme dovolit diskusi uživatelů k článku. Vytvoříme si proto další tabulku, kde budeme uchovávat komentář a jméno autora.

**Tabulka 9 – Počáteční návrh tabulky pro 3. NF**

Id komentáře	Název článku	Kategorie	Autor	Komentář
1	Svět kolem nás	Cestování	Filip	Krásný článek
2	Svět kolem nás	Cestování	Filip	Hrozný článek

Nyní si nemůžeme být jisti, že uživatel Filip je jeden a ten samý uživatel. K zamezení krádeže identity jiného uživatele, docílíme registrací uživatelů. Abychom mohli uživatele ověřit, přidáme mu heslo a získáme tím jasné ověření.

Vložením dalšího sloupce s heslem uživatele docílíme pouze duplikaci hesla v každém komentáři. Navíc sloupec Heslo by nebyl závislý na sloupci Autor. Problém vyřešíme rozkladem na více tabulek.

**Tabulka 10 – Rozdělení tabulek v 3. NF**

<b>Id komentáře</b>	<b>Autor</b>	<b>Komentář</b>
1	Filip	Krásný článek
2	Tomáš	Hrozně napsané

**Tabulka 11 – Rozdělení tabulek v 3. NF**

<b>Autor</b>	<b>Heslo</b>	<b>Email</b>
Filip	tajneheslo	filip@email.cz
Tomáš	mojeheslo	tomas@email.cz

Rozdělením tabulek na dvě menší dokážeme předejít zásadnímu problému: v případě, že bychom chtěli komentáře uživatele Filipa smazat, připravili bychom ho nejen o komentáře, ale i o jeho identitu na našem serveru. [12]

### **Databáze pro webovou aplikaci**

Při výběru databáze pro webovou aplikaci je třeba si uvědomit některé faktory. Jeden z hlavních faktorů je především možnost toho, aby náš projekt dokázal s databází pracovat. Následně je třeba zvážit i další kritéria jako cena, výkon, kompatibilita v případě více aplikací na serveru. V neposlední řadě také podpora české abecedy.

### **Příklady databází**

MySQL, PostgreSQL, MS Access, Oracle

## **1.4. MySQL**

MySQL databáze je celosvětově velmi populární. Využívají ji společnosti jako Facebook, Google, Adobe nebo Alcatel pro její rychlost cenu a rozšiřitelnost a využitelnost na hlavních platformách. Pro bez komerční použití ji můžeme využít bezplatně. Velice oblíbená je pro její jednoduchost, proto se ji uživatelé mohou rychle naučit.

Nevýhodou MySQL souvisí s její jednoduchostí, nelze zde vytvořit složitější programátorské konstrukce. Některé složitosti lze obejít skriptováním, při vyšší náročnosti aplikace, nemá dostatečný výkon.

### **Odlišnosti MySQL od ORACLE**

- Databáze MySQL nabízí funkci autoincrement, kterou například můžeme využít pro automatické vygenerování dalšího čísla. Nejčastěji se využívá při vkládání ID (identifikační číslo). Funkce nám zajistí, že se nám jako ID vloží následující hodnota po poslední přidané. Lze nastavit i jiný „skok“ čísla než pouze o jedničku. Tuto funkci nám Oracle nezajišťuje. V databázi Oracle musíme vytvořit sekvenci pro všechna ID, která chceme automaticky vygenerovat. [1]
- V MySQL (ver. 4) nelze použít složitější programování např. trigger, sekvence a další. [2]
- Oracle nabízí víc datových typů než MySQL.

## **1.5. Autentizace ve webových aplikacích**

Autentizace neboli ověření uživatele je na portálech, kde uživatel pracuje s daty nezbytností. V případě, že spravujeme obsah, který vkládáme, upravujeme či odstraňujeme, je třeba mít jistotu, že uživatel přistupuje pouze ke svým souborům. Autentizaci lze provést mnoha způsoby např. HTTP/BASIC, PHP skripty, sdílenou identitou.

### **HTTP/BASIC**

Autentizace jménem a heslem na úrovni HTTP protokolu. Chceme-li přistoupit na chráněnou stránku, pošle HTTP server odpověď a prohlížeč zobrazí okno pro zadání jména a hesla. Po ověření pošle hlavičku:

```
GET /chranene/ HTTP/1.1
```

```
Authorization: Basic bWFr dWl6bWFr dWJpaw==
```

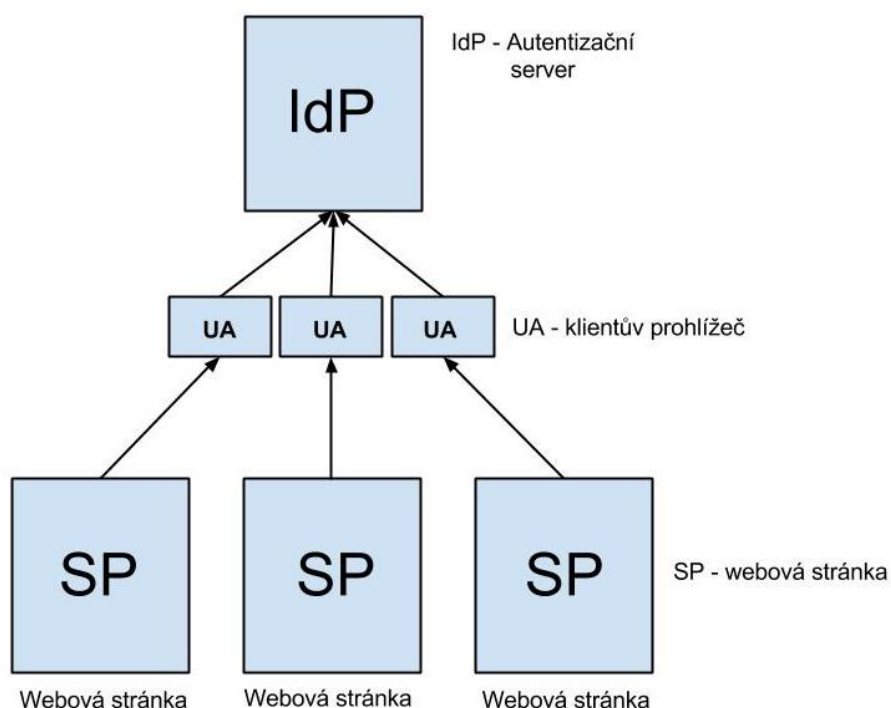
Kód bWFr dWl6bWFr dWJpaw== je zakódovaný funkcí base64, kterou lze rozkódovat. Pokud chceme využívat HTTP basic zabezpečení, musíme využívat protokolu SSL. Protokol SSL šifruje komunikaci mezi serverem a klientem, útočník má přístup k přenesenému řetězci, ale nedokáže jej dešifrovat.

## PHP kód

Pro autentizaci využijeme databázi, ve které uchováváme jméno a heslo. Pomocí formuláře údaje odešleme na server, který nám odpoví, a díky PHP skriptům dále zjistíme, zda uživatel existuje a pro uložení této informace využijeme např. session. „Session řeší problém bezstavovosti protokolu HTTP, udržíme pomocí ní informace o stavu aplikace a práci uživatele s ní.“ [13]

### 1.6. Sdílená identita

Využívaná metoda, ve které se snažíme uživatelům ušetřit práci. Uživatelé vlastní na každém serveru své uživatelské jméno a heslo a musí si jej pamatovat. Servery často mívají rozdílné požadavky na podobu hesla a přihlašovací jména, to uživatelům může způsobit komplikace. Sdílená identita pracuje na myšlence, že vytvoříme pouze jeden účet uživateli na autentizačním serveru a na každé webové stránce se ověříme právě proti tomu serveru. Uživatel si proto bude pamatovat pouze jedno uživatelské jméno a heslo.



**Obrázek 2 – Obecná ukázka sdílené identity, zdroj: vlastní**

Na obrázku lze vidět obecně nakreslenou sdílenou identitu. IDP – autentizační server obsahuje databázi se jménem a heslem, pomocí kterého se ověříme. Servery SP můžou

být nezávislé stránky, ze kterých dojde pouze k přesměrování, UA značí klientský prohlížeč.

### **Bezpečnost sdílené identity**

Při využití sdílené identity je třeba myslet i na bezpečnost. Autorizační server uchovává veškeré údaje o všech uživateli. Servery poskytující služby proto často pracují s velice citlivými údaji, jako jsou adresy, čísla účtů, telefonní čísla a další velice citlivé údaje uživatelů a jejich získání může být velice nebezpečné.

### **Klady sdílené identity**

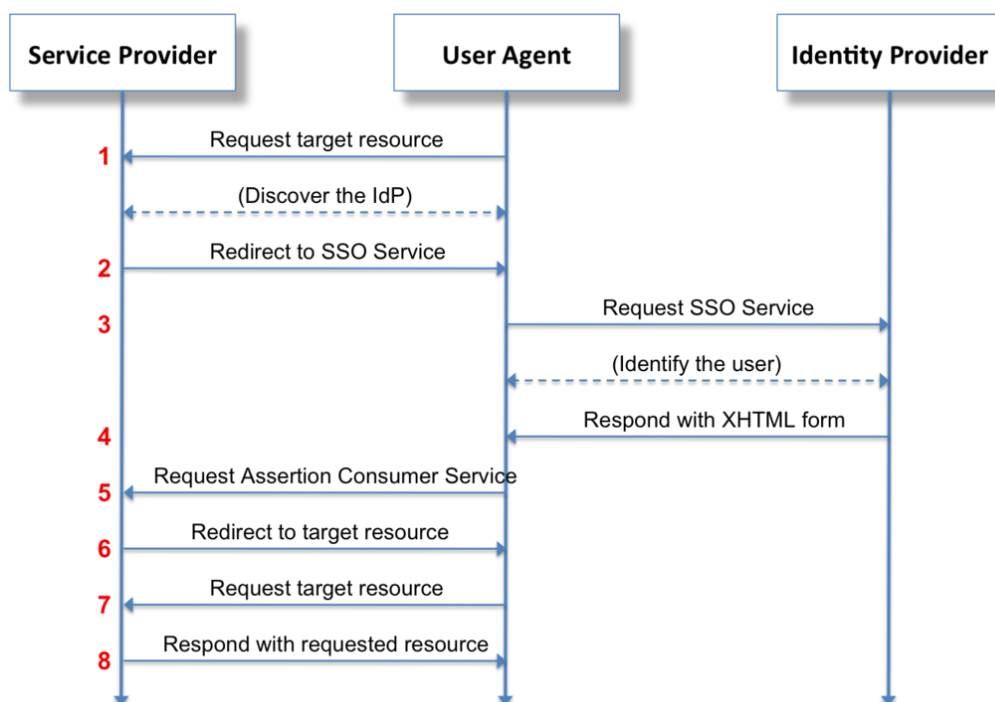
- Uživatelé si pamatují pouze jedno přihlašovací jméno a jedno heslo.
- Lze využít jeden AS pro více nezávislých webových serverů.
- Jednoduchost řešení.

### **Zápory sdílené identity**

- Ukradení údajů na straně AS je velice vážný problém bezpečnosti.
- Nutnost používat SSL pro komunikaci mezi klientem a WS.
- Nutnost přesměrování na AS

## **1.7. Řešení sdílené identity**

K řešení autentizace založené na sdílené identitě budeme v aplikaci využívat modelu na Obrázku 3. Jedná se o obecný model komunikace.



**Obrázek 3 – Možnost řešení sdílené identity, zdroj: [9]**

#### **Legenda obrázku**

- UA = User Agent = prohlížeč uživatele,
- SP = Service Provider = webová stránka, na které běží náš CMS systém,
- IDP = Identity Provider = server, který obsahuje databázi s uživatelskými jmény a hesly.

#### **Kroky v modelu:**

1. Uživatelův prohlížeč požádá server SP o jeho identifikaci přes navázaný SSL kanál.
2. Server SP odpovídá uživateli a odesílá mu jeho identifikaci, která je nyní uložena u uživatele.
3. Uživatel je přesměrován na server IDP, na kterém dochází k ověření uživatele.
4. Pokud ověření proběhne v pořádku, zasílá IDP server informaci o ověření uživatele.
5. Klientský prohlížeč zprostředkuje přenos klíče ze serveru IDP na server SP.
6. Server SP získává od UA identifikační číslo.
7. Probíhá ověření se získaným identifikačním číslem v bodě 6 a s klíčem zaslaným od IDP
8. Dojde-li k úspěšnému ověření, je klient UA považován za přihlášeného a je mu nastaven status uživatele a smí již pracovat s obsahem.

## Komunikace klientů

Pro dodržení bezpečnosti komunikace mezi servery je třeba dodržet následujících pravidel. Je třeba si uvědomit možné bezpečnostní mezery, kterých by mohl útočník využít a ukrást citlivé informace.

### Pravidla komunikace

- Komunikace mezi UA a SP musí probíhat šifrovaně. Nutno využít protokolu SSL, kterým zabezpečíme, aby útočník nemohl měnit údaje mezi nimi posílané. Na serveru SP, který vlastníme lze proto SSL protokol spustit.
- U komunikace mezi UA a IDP nelze vždy zajistit, že bude probíhat na šifrovaně, jelikož server IDP není v naší správě. Jediným řešením, které lze použít bez závislosti na funkčnosti SSL, je data posílat nešifrovaně.

Je třeba zajistit, aby útočník nemohl získat:

- Heslo uživatele, které se vkládá na serveru IDP. Bohužel nemáme přístup k serveru IDP, proto toto bezpečnostní riziko je v rukou serveru IDP.
- Session uživatele nesmí útočník podvrhnout, jelikož by se mohl útočníkův UA (prohlížeč) tvářit, že ověření uživatele proběhlo u něho.

### Možné útoky a obrana proti nim

1. Útočník zahájí komunikaci od bodu 5. Vyčká, až se uživatel ověří jménem a heslem proti databázi, odchytí informaci posílanou serverem IDP na server SP o úspěšném přihlášení a nahradí za svojí vlastní komunikaci serverem SP a bude se tvářit, že on je ten ověřený uživatel.

**Obrana:** V bodě 1 pošle server SP klientovi UA id, kterým se uživatel později prokáže po ověření na IDP. Komunikace musí probíhat šifrovaně, aby útočník nemohl odchytit id.

2. Útočník podvrhne v bodě 3 do požadavku vlastní identifikaci (prohlížeče), nahradí tím id klienta (oběti) a oběť se ověří se špatnou identifikací (útočníka), který pak pošle serveru SP správně vygenerovanou odpověď klienta (bod 5), čímž dojde k podvrženému ověření útočníka jako oběť (SP považuje útočníka za oběť).

**Obrana:** Komunikace mezi UA a IDP neběží šifrovaně, nýbrž server SP přes klienta UA posílá tajný řetězec, který je zašifrovaný klíčem, útočník tak nemůže zjistit, jakou hodnotu posílá IDP na SP a tím pádem, neví co nahradit do řetězce. V případě, že by se útočník pokusil nahradit řetězec za svůj, dochází po bodu 5

k ověření informací zaslaných na IDP oproti informacím uloženým v bodě 2. Nelze tak bez znalosti klíče podvrhnout komunikaci.

3. Útok na bod 4 – útočník si vymyslí fiktivní odpověď o ověření uživatele. Tj. útočník zahájí komunikaci bodem 4 a falšuje tak ověření na IDP.

**Obrana:** Útok lze zabránit zašifrováním tajného klíče, který útočník nezná, nelze tak vygenerovat řetězec, který by server SP schválil při ověření.

4. Útočník se pokusí vyměnit bajt v zašifrované zprávě, kterou posílá IDP na SP, čímž by teoreticky mohl s pravděpodobností zhruba (dle použitého algoritmu) 1:256 nastavit požadovanou hodnotu bajtu, aniž by znal klíč.

**Obrana:** IDP ke zprávě připojí její hash a vzniklý řetězec zašifruje. Po převzetí a dešifrování na straně serveru SP dojde k ověření hashe. (Hash zprávy se přepočítá a porovná se se zaslanou hodnotou.) Pokud útočník změní některý bajt zašifrované zprávy, nebude kontrola úspěšná. (Analogie s elektronickým podpisem.)



## **2. Praktická část**

### **Základní popis aplikace**

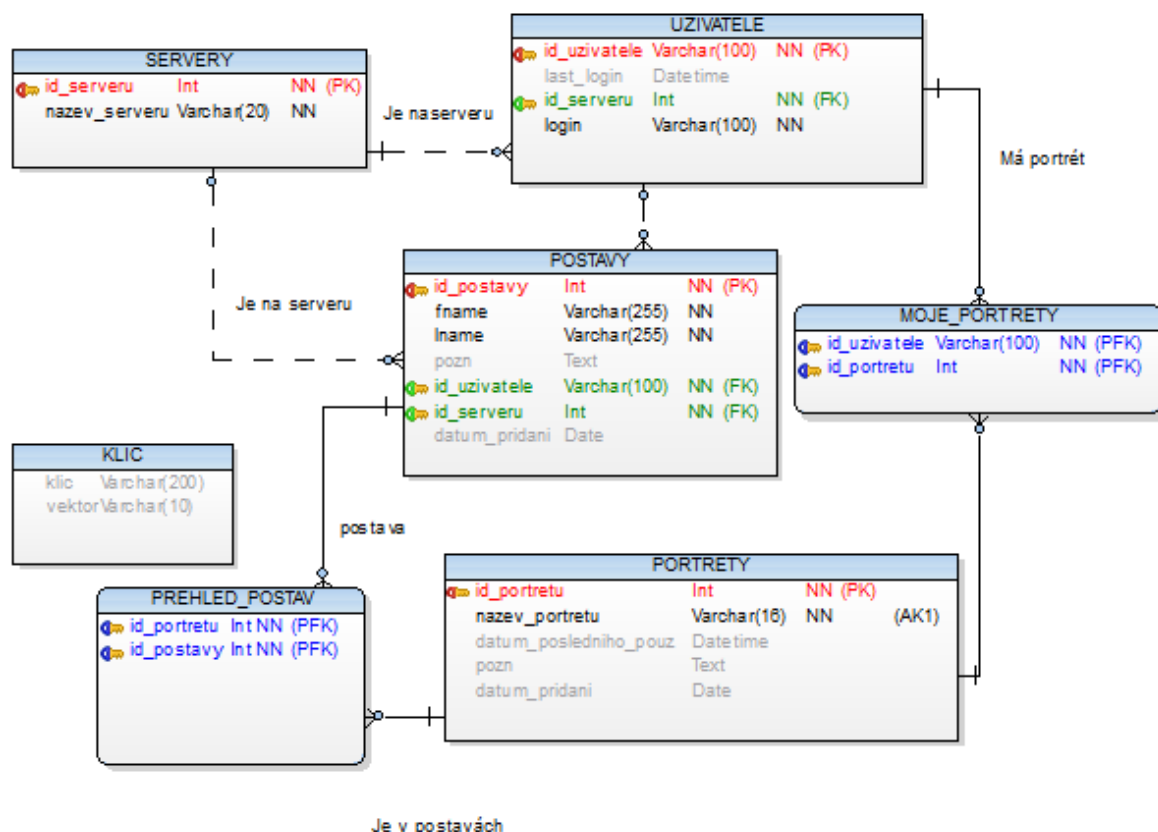
V praktické části byla vytvořena webová aplikace pro správu portrétů. Je napsána za použití programovacích jazyků HTML5, CSS3 a PHP. Řídí se vlastnostmi CMS, tudíž umožňuje spravovat obsah na stránkách bez znalosti programování. Umožňuje nahrávat portréty uživatelů a spojovat je s postavami. Jedna z hlavních funkcí je vyhledávání a stahování portrétů ve formátu zip. Soubor uživatel následně rozbalí do adresáře portraits, který je v instalačním adresáři hry Neverwinter Nights, a portréty může využívat.

### **Vzhled aplikace**

Aplikace byla navržena, aby spolupracovala se všemi typy rozlišení, minimální podporované rozlišení je 1024 bodů na šířku, poté se uživateli zobrazí rolovací tlačítko. Všechny prvky webových stránek se přizpůsobují nejen rozlišení, ale i šířce okna a snaží se o maximum zobrazených informací.

### **2.1. Návrh databáze pro aplikaci**

Databáze byla navržena pro uchovávání postav, portrétů a uživatelů. Ke správnému běhu aplikace bylo třeba vytvořit i relace, které nám spojují portréty, postavy a uživatele. Model databáze je vytvořen pro MySQL databázi a navržen v programu Toad Data Modeler.



Obrázek 4 – Schéma databáze, zdroj: vlastní

## Popis tabulek

- **Tabulka uživatelů**

O každém uživateli potřebujeme uchovávat jeho uživatelské jméno, které nám pošle autentizační server IDP, id serveru a pro naši informaci si zapisujeme, kdy se naposledy přihlásil do systému. K jasné identifikaci uživatele využíváme id uživatele.

- **Tabulka postav uživatele**

Každý uživatel si smí na serveru vytvořit svou vlastní postavu, postav může vytvořit kolik postav chce, nesmí však vytvořit postavu stejného jména a příjmení. Tabulka eviduje jméno postavy, příjmení postavy, poznámku o postavě, id uživatele, který jej vlastní, id serveru na kterém ji evidujeme a datum vytvoření.

- **Tabulka portrétů uživatele**

Tabulka portrétů eviduje všechny nahrané portréty na server. U každého portréту uložíme do databáze jeho jméno (maximální 16 znaků), id uživatele, který jej vytvořil, datum posledního použití, poznámku a datum přidání.

- **Tabulka přehledu postav**

Tabulka přehled postav nám upřesňuje, které portréty vlastní určitá postava, portrét se smí být k určité postavě přiřazen pouze jednou. Jiná postava ho může opět využít. U postavy může být více různých portrétů.

- **Tabulka vlastních postav**

Tabulka obsahující všechny portréty, které si uživatel na serveru vybral, jak ty které nahrál, tak i ostatních uživatelů.

- **Tabulka serverů**

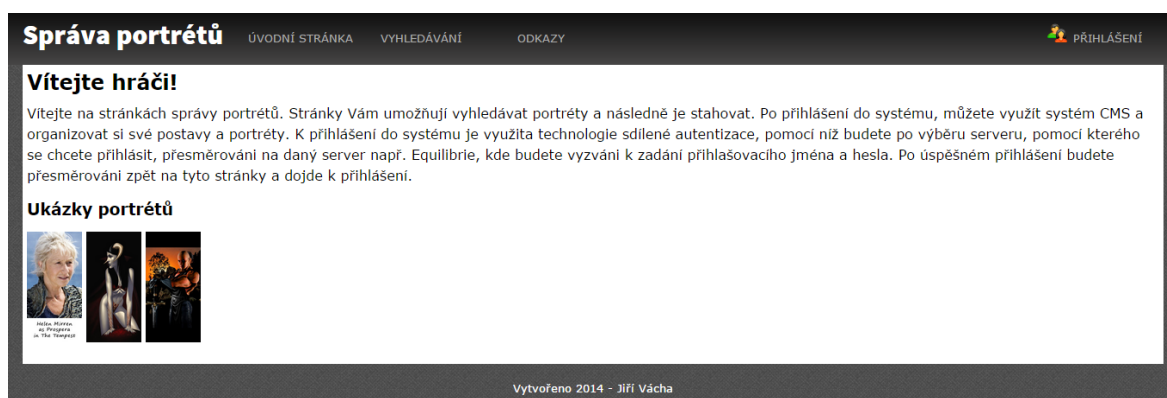
Udává pouze označení serveru, pro případné další použití. Pokud není zde server uveden nelze se přihlásit.

- **Tabulka administrátorských údajů**

V tabulce je uložen klíč a vektor, pomocí kterého se šifrují data posílané v url adrese.

## 2.2. Úvod aplikace

Aplikace je tvořena v jazyce HTML za použití skriptů PHP a její design je tvořen pomocí stylů CSS3. Vzhled aplikace je jednoduchý. Hlavička stránky po logu obsahuje navigaci. V navigaci stránek máme odkazy zpět na úvodní stránku, vyhledávání, odkazy a přihlášení do systému. V případě, že jsme již úspěšně přihlášení, zobrazí se nám odkaz na můj účet a odkaz přihlášení se změní na odhlášení. Obsah úvodní stránky a odkazů je plně v roli administrátora stránek. Sekce vyhledávání a můj účet je vytvořený podle požadavků vedoucího práce.

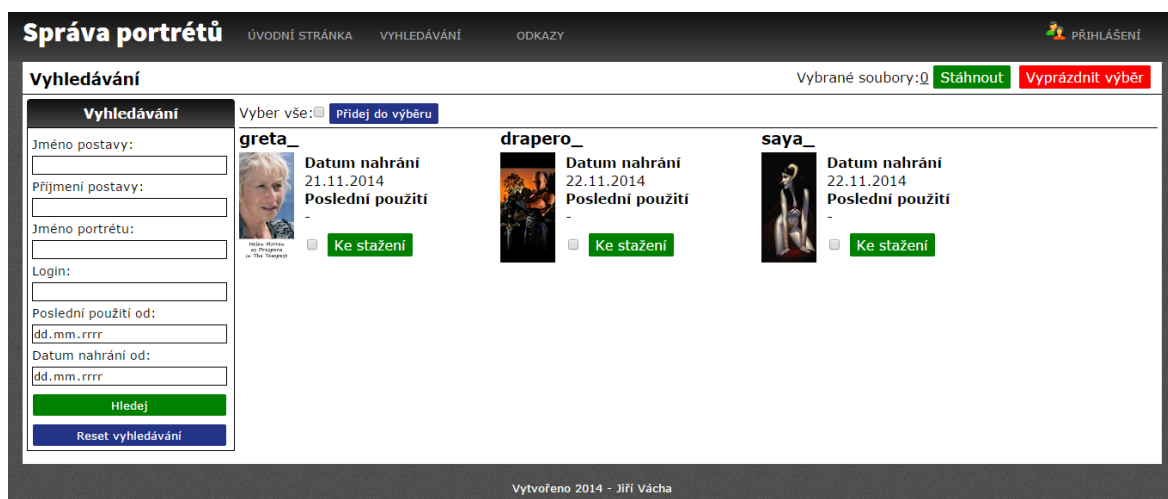


Obrázek 5 – Úvodní stránka aplikace, zdroj: vlastní

## Sekce vyhledávání

Slouží pro vyhledávání portrétů, v levé části lze v panelu vyhledávání určit parametry, podle kterých chceme vyhledávat v databázi. V hlavní části obrazovky se nám zobrazí již vyhledané portréty. U každého portréty je možnost soubor vložit do seznamu ke stažení, na tento účel slouží tlačítko Ke stažení. V případě vícenásobného výběru zaškrtneme všechny portréty, které chceme vložit do seznamu a po kliknutí na tlačítko Přidej do výběru se nám portréty do něj vloží.

V pravé části obrazovky můžeme vidět počet portrétů námi vybraných do seznamu a tlačítko Stáhnout. Po jeho kliknutí vytvoříme archiv a budeme následně vyzváni k jeho uložení. Pokud jsme si rozmysleli, a chceme seznam souborů vyprázdnit na to nám poslouží tlačítko Vyprázdnit výběr. K zobrazení vybraných portrétů stačí kliknout na počet souborů a budeme přesměrováni na další stránku.



Obrázek 6 – Sekce vyhledávání, zdroj: vlastní

## Sekce odkazy

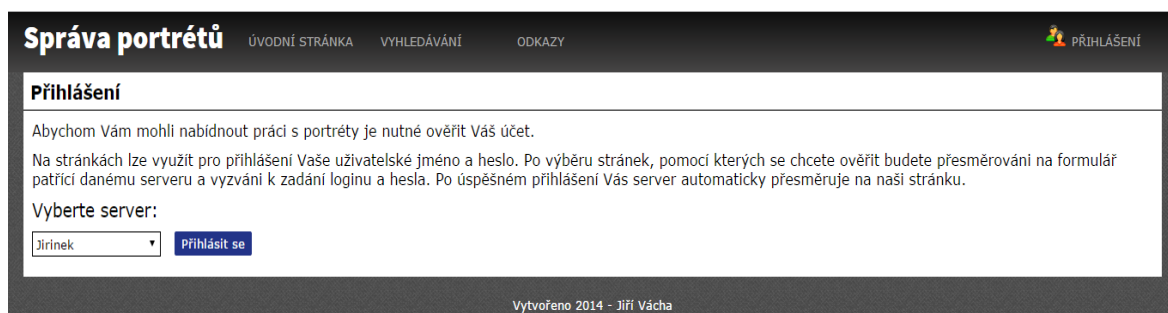
Zobrazuje odkazy, které administrátor určí. Jednoduchá podstránka s tabulkou.



Obrázek 7 – Stránka odkazy, zdroj: vlastní

## Přihlášení do aplikace

V případě, že uživatel chce na stránkách spravovat vlastní obsah, je nutné se přihlásit, po kliknutí na stránku přihlášení si vyberete, pomocí kterého serveru se chcete přihlásit. Komunikace začne pracovat na protokolu HTTPS proto, abychom šifrovali přenos mezi klientským prohlížečem a serverem.



Obrázek 8 – Přihlášení do aplikace, zdroj: vlastní

## Průběh přihlášení

- Přihlášení pracuje na principu sdílené identity: po výběru serveru se uloží do proměnné session vaše identifikační číslo pomocí kterého budete později ověřen a dojde k přesměrování na server IDP, kde budete vyzváni k zadání jména a hesla. Po úspěšném ověření bude vytvořen klíč a budete přesměrován zpět na stránky správy portrétů.
- Server IDP přijme zašifrovaný údaj od SP, uloží jej a čeká na úspěšné přihlášení. Pokud se uživateli povede ověřit proti databázi IDP, spustí speciální funkce na tvorbu nové šifry. Funkce vezme příchozí zašifrovaný údaj a pomocí klíče uloženého v databázi jej rozšifruje, přidá k němu údaje o uživateli, serveru a kontrolní hash a celý řetězec opět zašifruje klíčem. Pomocí funkce header dojde k přesměrování prohlížeče uživatele zpět na server SP.
- SP přijme zašifrované údaje a dojde k dešifrování klíčem uloženým v databázi (klíč je stejný jako na serveru IDP). Po dešifrování údajů jsou vyjmuty hodnoty o uživateli a serveru. Pomocí session ověříme identifikaci klienta, která mu byla na začátku přidělena, a vypočteme hash z přijaté zprávy, který porovnáme s příchozím hashem. V případě, že jsou hodnoty shodné, dojde k úspěšnému přihlášení uživatele. Na to nám slouží třída Authenticator: uložíme si hodnoty uživatelského jména a serveru, zjistíme, zda takový uživatel byl již na našem serveru přihlášen, pokud nebyl, do databáze ho vložíme. Nakonec vybereme všechny informace o uživateli z databáze a uložíme je do session.

## Sekce Můj účet

V této sekci má uživatel možnost spravovat své postavy a portréty. V levé části je menu, ve kterém je výběr všech jeho postav, v případě, že nemá ještě žádnou postavu přidanou tak pouze tlačítko přidat postavu. V hlavní části jsou všechny postavy vypsané. U každé postavy lze vybrat její úpravu či odebrání ze seznamu postav.

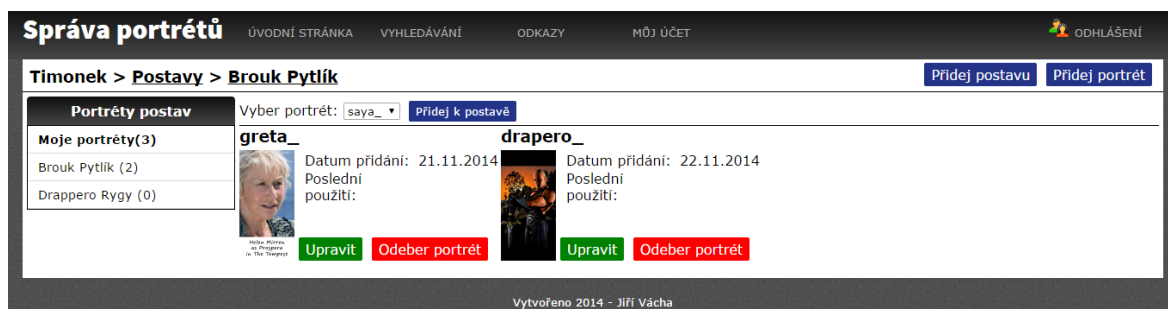


Obrázek 9 – Výpis postav uživatele, zdroj: vlastní

## Práce s portréty

V pravé horní části sekce můj účet je tlačítko Přidej postavu. Po jeho kliknutí se zobrazí formulář pro zadání jména, příjmení a poznámky k postavě. Každý uživatel nesmí mít dva stejně pojmenované postavy. Dvě stejné postavy u různých uživatelů je možný.

- Po kliknutí na úpravu postav se uživateli zobrazí formulář pro zadání údajů: jména příjmení a poznámky k postavě. Po úpravě uživatel potvrdí tlačítkem Potvrdit úpravu a zobrazí se informativní hláška o úspěšnosti/neúspěšnosti operace.
- Nechce-li uživatel již evidovat postavu, lze vymazat. Všechny portréty nebudou smazány, neustále jsou uchovávány v sekci Mé portréty.
- Po vybrání postavy se uživateli zobrazí všechny portréty, které k postavě přidal. V případě že chce přidat další, vybere ji v rozvíracím seznamu a tlačítkem přidej k postavě, ji přiřadí. Portrét musí být vložen v uživatelských portrétech, jinak se ve výběru nezobrazí. To lze provést nahráním portréту nebo v sekci vyhledávání přes tlačítko Moje +.



Obrázek 10 – Vzhled konkrétní postavy, zdroj: vlastní

## Přidání portrétu

Přidání portrétu je jedna z hlavních funkcí celé aplikace. Pro úspěšné přidání portrétu je třeba dodržet všechna zadaná pravidla.

**Správa portrétů** ÚVODNÍ STRÁNKA VYHLEDÁVÁNÍ ODKAZY MŮJ ÚČET ODHLÁŠENÍ

**Timonek > Postavy** Přidej postavu Přidej portrét

**Portréty postav**

Moje portréty(3)
Brouk Pytlík (3)
Drappero Rygy (0)

**Přidej portrét**

Vyber soubor:  Soubor nevybrán

Poznámka:

**Poznámky k souborům**

V prvním kroku vyberte, pro kterou postavu chcete nahrát portrét. V případě, že je již portrét nahraný na serveru, nelze znovu portrét nahrát. Portrét můžete přidat ke své postavě v kategorii ke stažení.

**Úprava portréту**

Chcete-li Vámi nahraný portrét přehrát, lze tak učinit pouze v případě, že daný portrét **žádný** z hráčů nepoužívá.

**Soubor musí splňovat tyto vlastnosti, jinak nebude přijat:**

- Archiv musí být typu **zip** nebo **rar**.
- Jméno portréту uloženého v archivu smí mít maximálně 16 znaků.
- V archivu smí být pouze soubory typu tga.
- Archiv musí obsahovat minimálně soubory **\_m**, **\_l**, **\_s**, **\_t**.
- Povolené velikosti souborů: **\_h**(256x512), **\_m**(128x256), **\_l**(64x128), **\_s**(32x64), **\_t**(16x32).

Vytvořeno 2014 - Jiří Vácha

Obrázek 11 – Přidání portrétu do systému, zdroj: vlastní

## Pravidla

- Nahrát lze pouze archiv typu zip nebo rar.
- Jména všech portrétů v archivu musí být stejná s maximální délkou 16 znaků.
- Všechny soubory musí být typu tga.
- Archiv musí obsahovat minimálně soubory končící písmeny m, l, s, t. Volitelně může obsahovat ještě soubor končící h (který klient NWN nepoužívá).
- Povolené rozlišení souborů: h – 256×512, m – 128×256, l – 64×128, s – 32×64, t – 16×32.
- Portrét stejného jména nesmí na serveru existovat.

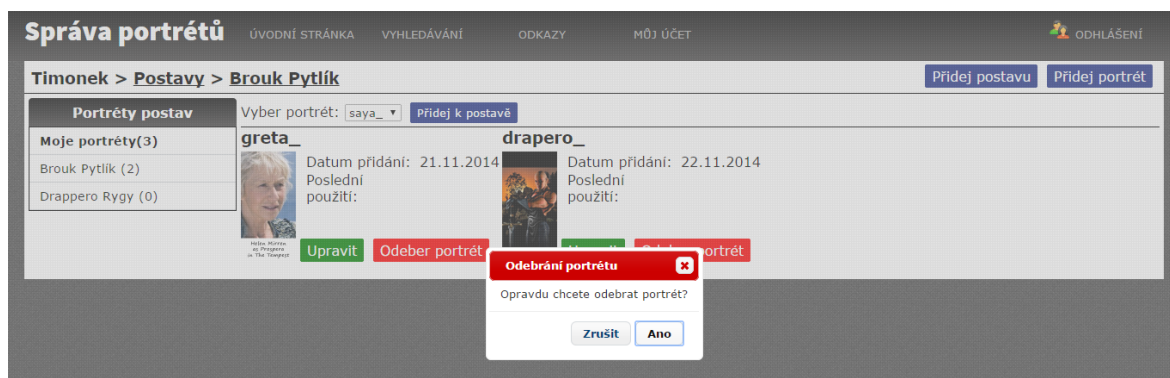
V případě, že jsou splněny všechny podmínky, dochází k nahrání celého archivu na server a převedení obrázků o velikosti m a l na formát png, aby byl dále použitelný pro zobrazení na webových stránkách.

## Úprava portréту

Úpravu portréту smí provést pouze uživatel, který portrét nahrál. Nelze to však učinit pokaždé, vlastní-li ho některá z dalších postav ať už uživatele, který ho přidal nebo ostatních uživatelů nelze portrét přehrát novou verzí. Uživateli bude umožněno pouze upravit poznámku u portréту. Uživateli zbývá portrét přejmenovat a nahrát ho pod jiným názvem.

## Další funkce aplikace

Aplikace byla vytvořena, aby byla co nejvíce uživatelsky přátelská. Při tvorbě bylo myšleno i na nechtěné smazání postav či portrétů uživatelů. V aplikaci byly využity dialogy vytvořené pomocí jQuery (javascriptová knihovna pro snadné programování interaktivních webových stránek). Uživatel je upozorněn, že se chystá provést změny a může je potvrdit či zrušit.



Obrázek 12 – Ukázka dialogu, zdroj: vlastní



### 3. Závěr

Cílem práce bylo vytvořit portál pro správu portrétů, který aplikuje vlastnosti CMS systému. S využitím programovacích jazyků HTML, PHP a CSS byla vytvořena aplikace s přátelským uživatelským rozhraním umožňujícím spravovat portréty a postavy hry Neverwinter Nights.

Na základě zkoumání technologie sdílené autentizace byla navržena metoda, využívající servery třetích stran, pomocí které se uživatel do webové aplikace přihlašuje. Hlavním hlediskem byla bezpečnost přenosu mezi servery a jednoduchost implementace pro více serverů třetích stran.

K aplikaci byla navržena databáze, splňující pravidla relačních databází pro správné uchovávání informací s minimálním zatížením serveru. Dále je možno v aplikaci nahrávat portréty, jejich přijetí závisí na kontrole souboru dle zadaných pravidel.

Do budoucna by se aplikace mohla rozšířit o více informací přímo ze hry Neverwinter Nights, které by uživatelům umožnily větší přehled o využívaných portrétech.

## Literatura

1. JANOVSKEÝ, Dušan. *Jak psát web: Úvod do CSS*. [online]. 2014-11-25 [cit. 2014-04-21]. Dostupné z: <http://www.jakpsatweb.cz/css/css-uvod.html/>
2. KOSEK, Jiří. *PHP - tvorba interaktivních internetových aplikací: podrobný průvodce*. Vyd. 1. Překlad Martin Domes. Praha: Grada, 1999, 490 s. Průvodce (Grada). ISBN 80-716-9373-1.
3. PEXA, Petr. *Tvorba WWW a WAP: vytváříme přizpůsobitelné a přístupné stránky pomocí XHTML a CSS*. 1. vyd. Překlad Martin Domes. České Budějovice: Kopp, 2001, 215 s. ISBN 80-723-2161-7
4. ULLMAN, Larry. *PHP a MySQL: názorný průvodce tvorbou dynamických WWW stránek*. Vyd. 1. Brno: Computer Press, 2004, 534 s. ISBN 80-251-0063-4.
5. CEDERHOLM, Dan. *Flexibilní webdesign: vytváříme přizpůsobitelné a přístupné stránky pomocí XHTML a CSS*. Vyd. 1. Překlad Martin Domes. Brno: Computer Press, 2006, 227 s. ISBN 80-251-1018-4.
6. KRČÁL, Martin. *Citace.com* [online]. 2004 [cit. 2014-11-25]. Dostupné z: <http://www.citace.com>
7. MySQL: The world's most popular open source database. *MySQL* [online]. 2014 [cit. 2014-11-25]. Dostupné z: <http://www.mysql.com/>
8. KULHAN, Jakub. Normalizace relačních databází. *Programujte.com: Databáze* [online]. 23. 7. 2008 [cit. 2014-11-25]. Dostupné z: <http://programujte.com/clanek/2008071900-normalizace-relacnich-databazi/>
9. Security Assertion Markup Language. *WIKIPEDIA: The free encyclopedia*. [online]. 2014 [cit. 2014-12-03]. Obrázek. Dostupné z: [http://en.wikipedia.org/wiki/Security\\_Assertion\\_Markup\\_Language](http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language)
10. *WORDPRESS – ČESKÁ PODPORA* [online]. [cit. 2014-12-03]. Obrázek. Dostupné z: <http://www.cwordpress.cz/>

11. ČÁPKA, David. 2. díl – Knihovna DateUtils pro český datum a čas v PHP. *ITnetwork.cz: Sociální síť pro IT profesionály* [online]. 2014 [cit. 2014-11-25]. Dostupné z: <http://www.itnetwork.cz/php-tutorial-knihovna-dateutils-pro-cesky-datum-a-cas>
12. DOSTAL, Martin. Normální formy. In: *DBM1 – cvičení* [online]. 2010 [cit. 2014-12-01]. Dostupné z: [http://home.zcu.cz/~madostal/index.php?page=vyuka&subpage=dbm1&lesson=normalni\\_formy](http://home.zcu.cz/~madostal/index.php?page=vyuka&subpage=dbm1&lesson=normalni_formy)
13. RŮŽIČKA, Pavel. Začínáme používat sessions v PHP. *Interval.cz* [online]. 21. 11. 2002 [cit. 2014-12-03]. ISSN 1212-8651. Dostupné z: <http://interval.cz/clanky/zaciname-pouzivat-sessions-v-php/>

## **Příloha A – Instalační CD**

Instalační CD obsahuje veškeré informace týkající se nastavení aplikace. Dále na CD jsou zdrojové kódy, SQL příkazy a konfigurační soubory.

## Příloha B – ukázky zdrojových kódů

Jedna z hlavních sekcí webové aplikace. Umožňuje uživatelům vyhledávat portréty podle parametrů. Dotaz směřující do databáze závisí na počtu a typu parametrů, které uživatel zadá. Na obrázku 13 lze vidět, jak se dotaz tvoří. Musíme rozhodnout, které parametry se do dotazu vloží.

```
if(isset($_POST['hledej']))
{
    $hledani_portretu="SELECT DISTINCT PORTRETY.id_portretu,PORTRETY.nazev_portretu,DATE_FORMAT(PORTRETY.datum_posledniho_pouz, '%d.%m.%Y')
    . "DATE_FORMAT(PORTRETY.datum_pridani, '%d.%m.%Y') as dat_prid FROM PORTRETY"
    . " JOIN PREHLED_POSTAV USING (id_portretu) JOIN POSTAVY USING(id_postavy)"
    . " JOIN UZIVATELE ON(POSTAVY.id_uzivatele=UZIVATELE.id_uzivatele)";
    if($_POST['jm_postavy']!=''){
        if($dalsiF->getWhere()){ $hledani_portretu.=" WHERE "; $dalsiF->setWhere(false); }else{ $hledani_portretu.=" AND "; }
        $jm_postavy = $conn->osetriParametr($_POST['jm_postavy']);
        $hledani_portretu.= " POSTAVY.fname LIKE '%$jm_postavy%'";
    } if ($_POST['pr_postavy'] != "") {
        if ($dalsiF->getWhere()) {
            $hledani_portretu.=" WHERE ";
            $dalsiF->setWhere(false);
        } else {
            $hledani_portretu.=" AND ";
        }
        $pr_postavy = $conn->osetriParametr($_POST['pr_postavy']);
        $hledani_portretu.= " POSTAVY.lname LIKE '%$pr_postavy%'";
    } if ($_POST['jm_portretu'] != "") {
        if ($dalsiF->getWhere()) {
            $hledani_portretu.=" WHERE ";
            $dalsiF->setWhere(false);
        } else {
            $hledani_portretu.=" AND ";
        }
        $jm_portretu = $conn->osetriParametr($_POST['jm_portretu']);
        $hledani_portretu.= " PORTRETY.nazev_portretu LIKE '%$jm_portretu%'";
    }
}
```

Obrázek 13 – Ukázka kódu – tvorba dotazu vyhledávání, zdroj: vlastní