

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Návrh a implementace BitTorrent trackeru a systému  
pro správu a sdílení torrent souborů

Pavel Jandejsek

Bakalářská práce

2010

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2009/2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Pavel JANDEJSEK**  
Osobní číslo: **I07640**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Návrh a implementace BitTorrent trackeru a systému pro správu a sdílení torrent souborů**  
Zadávající katedra: **Katedra informačních technologií**

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

V teoretické části práce bude provedena rešerše peer-to-peer sítí a jejich dnešní pozice ve světě Internetu. Dále bude potřebné zaměřit se na popis protokolu BitTorrent a uplatnění v peer-to-peer sítích.

Praktická část bude primárně zaměřena na návrh a implementaci aplikace BitTorrent trackeru a systému pro správu a sdílení torrent souborů.

Výsledná aplikace bude otestována v reálném provozu.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

1. HBELER, J. **Peer-to-Peer: Building Secure, Scalable, and Manageable Networks.** 2001, McGraw-Hill Companies
2. BUFORD, J., HEATHER, Y. **P2P Networking and Applications.** 2008, Morgan Kaufmann

Vedoucí bakalářské práce:

**Ing. Jan Fikejz**

Katedra softwarových technologií

Datum zadání bakalářské práce:

**15. ledna 2010**

Termín odevzdání bakalářské práce:

**14. května 2010**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2010

## **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 Kolíně dne 24. 04. 2010

Pavel Jandejsek

## **Poděkování**

Na tomto místě bych chtěl poděkovat Ing. Janu Fikejzovi za vedení této práce. Také bych velice rád poděkoval Ing. Filipu Janděskovi za cenné rady při tvorbě této práce. Dále bych chtěl poděkovat svým přátelům, zejména Vladimíru Kaděrovi, Romanu Svobodovi a Jindřichu Syrovému za pomoc při testování aplikace.

## **Anotace**

Tato práce se zabývá návrhem a implementací systému umožňujícího správu a sdílení torrent souborů a obsahujícího funkce BitTorrent trackeru. Nejprve se práce zaměřuje na peer-to-peer síť a popis protokolu BitTorrent. Dále je provedeno stanovení cílů, analýza aplikace a popis implementace. Závěr se zabývá zhodnocením vytvořené aplikace.

## **Klíčová slova**

P2P, peer-to-peer, BitTorrent, torrent, tracker, sdílení souborů, PHP

## **Title**

Design and implementation of a BitTorrent tracker with a system for sharing and management of torrent files.

## **Annotation**

This thesis deals with design and implementation of a system capable of sharing and managing torrent files and containing functions of BitTorrent tracker. The first part of the thesis focuses on peer-to-peer networks and description of BitTorrent protocol. Objective setting along with application analysis and description of implementation is done next. Conclusion evaluates the created application.

## **Keywords**

P2P, peer-to-peer, BitTorrent, torrent, tracker, file sharing, PHP

## Obsah

Seznam zkratk	9
Seznam obrázků	10
Seznam tabulek	11
1 Úvod	12
2 Peer-to-peer síť	13
2.1 Porovnání peer-to-peer a klient-server architektur	13
2.1.1 Kapacita	13
2.1.2 Spolehlivost	13
2.1.3 Bezpečnost	13
2.1.4 Aktualizace	14
2.2 P2P v Internetu	14
3 BitTorrent	15
3.1 Základní termíny	16
3.1.1 Bencoding	16
3.2 Struktura souboru s meta informacemi (torrentu)	17
3.2.1 Klíče torrentu	17
3.3 Druhy komunikace	19
3.3.1 Komunikace klient-tracker	19
4 Analýza aplikace	23
4.1 O aplikaci	23
4.2 Cíle aplikace	23
4.3 Uživatelské role v systému	23
4.3.1 Návštěvník	23
4.3.2 Registrovaný uživatel	23
4.3.3 Administrátor	24
4.4 Dostupné alternativy	24
4.4.1 TBDev.net	24
4.4.2 xbtit	24
4.4.3 phpMyBitTorrent	25
4.5 UML	25
4.5.1 Use case diagram	25
4.5.2 Activity diagram	28

4.5.3	Class diagram .....	31
5	Zvolené technologie.....	32
5.1	PHP .....	32
5.2	MySQL.....	32
5.3	Návrh databázové struktury .....	33
5.3.1	ER diagram databázové struktury .....	34
5.3.2	Popis databázových tabulek .....	35
5.4	Popis implementace návrhového vzoru MVC.....	39
5.4.1	Model (model) .....	39
5.4.2	View (pohled).....	39
5.4.3	Controller (kontrolér).....	39
5.4.4	Class diagram použitých tříd .....	40
6	Testování .....	46
6.1	Testování kódu systému.....	46
7	Problémy při vytváření aplikace.....	47
7.1	Foreach a DOMNodelist .....	47
7.2	DOMNode a metoda appendChild .....	47
7.3	MySQLi_STMT, metoda bind_param a velké číslo.....	47
7.4	MySQL a spojování řetězců v dotazu.....	47
8	Závěr .....	48
9	Zdroje .....	49
	Příloha .....	51
A	Instalační příručka .....	51
B	Uživatelská příručka .....	53
C	Dokumentace kódu aplikace .....	61
D	Obsah CD .....	62



## Seznam zkratek

P2P	peer-to-peer
eD2k	eDonkey2000
BBC	British Broadcasting Corporation
UTF	Unicode Transformation Format
URL	Uniform Resource Locator
POSIX	Portable Operating System Interface for Unix
SHA	Secure Hash Algorithm
DHT	Distributed Hash Table
MD5	Message-Digest Algorithm 5
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
RFC	Request for Comments
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
UML	Unified Modeling Language
GNU	GNU's Not Unix
GPL	General Public License
PHP	Hypertext Preprocessor
OOP	Object-oriented Programming
IIS	Internet Information Services
CGI	Common Gateway Interface
ODBC	Open Database Connectivity
IDE	Integrated Development Environment
ER	Entity-Relationship
MVC	Model-View-Controller
DOM	Document Object Model
XML	Extensible Markup Language
SQL	Structured Query Language
XHTML	Extensible Hypertext Markup Language
XPath	XML Path Language

## Seznam obrázků

Obrázek 1 – Porovnání klient-server a klient-klient architektur .....	13
Obrázek 2 – Příklad přenosu souboru pomocí protokolu BitTorrent .....	15
Obrázek 3 – Příklad diagramu užití .....	26
Obrázek 4 – Diagram užití s nepřihlášeným uživatelem .....	26
Obrázek 5 – Diagram užití s přihlášeným uživatelem .....	27
Obrázek 6 – Diagram užití s administrátorem.....	27
Obrázek 7 – Příklad diagramu aktivit .....	28
Obrázek 8 – Diagram aktivit nahrání torrentu.....	29
Obrázek 9 – Diagram aktivit odpovědi trackeru .....	30
Obrázek 10 – Příklad diagramu tříd.....	31
Obrázek 11 – ER diagram databázové struktury .....	34
Obrázek 12 – Znázornění návrhového vzoru MVC .....	39
Obrázek 13 - Class Diagram pomocné třídy Connector .....	40
Obrázek 14 – Class diagram pomocné třídy Template.....	40
Obrázek 15 – Class diagram tříd modelu 1 .....	41
Obrázek 16 – Class diagram tříd modelu 2 .....	42
Obrázek 17 – Class diagram tříd modelu 3 .....	43
Obrázek 18 – Class diagram tříd modelu 4 .....	44
Obrázek 19 – Class diagram tříd modelu 5 .....	45
Obrázek 20 – Vzhled systému .....	53
Obrázek 21 – Registrační formulář.....	54
Obrázek 22 – Přihlašovací formulář .....	54
Obrázek 23 – Uživatelské detaily .....	55
Obrázek 24 – Formulář pro varování uživatele .....	55
Obrázek 25 – Formulář pro ban uživatele .....	56
Obrázek 26 – Formulář pro nahrání torrentu.....	57
Obrázek 27 – Výpis seznamu torrentů.....	58
Obrázek 28 – Detaily torrentu .....	59
Obrázek 29 – Formulář pro ban torrentu.....	60

## Seznam tabulek

Tabulka 1 – Příklad popisu databázových tabulek .....	35
Tabulka 2 – Popis databázové tabulky users .....	35
Tabulka 3 – Popis databázové tabulky torrents .....	35
Tabulka 4 – Popis databázové tabulky peers .....	36
Tabulka 5 – Popis databázové tabulky categories .....	36
Tabulka 6 – Popis databázové tabulky completed .....	37
Tabulka 7 – Popis databázové tabulky downloaded .....	37
Tabulka 8 – Popis databázové tabulky levels .....	37
Tabulka 9 – Popis databázové tabulky torrents_categories .....	38
Tabulka 10 – Popis databázové tabulky torrent_comments .....	38
Tabulka 11 – Popis databázové tabulky torrent_bans .....	38
Tabulka 12 – Popis databázové tabulky user_bans .....	38
Tabulka 13 – Popis databázové tabulky user_warning .....	39

# 1 Úvod

Cílem této práce je navrhnout a implementovat systém umožňující správu a sdílení torrent souborů a současně disponujícím funkcemi BitTorrent trackeru. Výsledná aplikace bude obsahovat evidenci torrentů a uživatelů, spolu s nástroji pro jejich správu. Navrhovaný systém je koncipován jako webová aplikace.

V první části se práce zabývá peer-to-peer sítěmi a popisem protokolu BitTorrent.

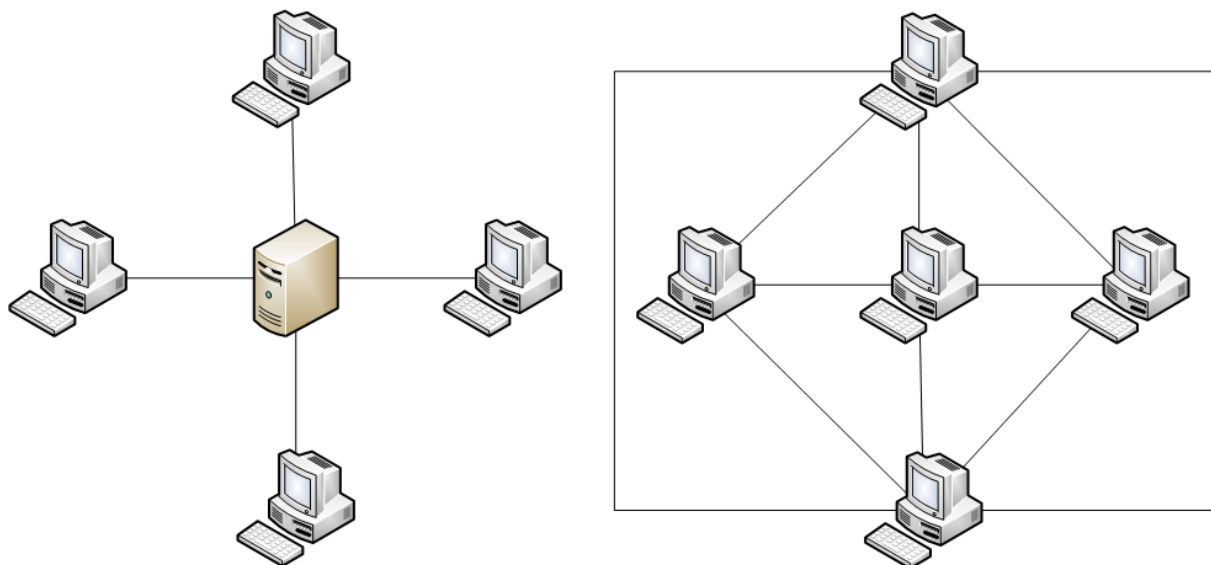
Druhá část je věnována stanovení cílů, porovnání s ostatními dostupnými systémy a provedením analýzy aplikace.

Poslední část je věnována popisu použitých technologií, testování a problémům při vytváření aplikace.

## 2 Peer-to-peer síť

Peer-to-peer, někdy označováno jako P2P, nebo také klient-klient, je síťová architektura, ve které všichni účastníci poskytují své zdroje (např. výpočetní výkon, šířku pásma síťového připojení) ostatním účastníkům sítě (bez nutnosti použití centralizovaných uzlů) [1]. Tato architektura je opakem tzv. klient-server modelu, kde komunikace probíhá s využitím centrálního uzlu (popř. uzlů).

### 2.1 Porovnání peer-to-peer a klient-server architektury



Obrázek 1 – Porovnání klient-server a klient-klient architektury, zdroj: vlastní

#### 2.1.1 Kapacita

Obecně se nechá říci, že s větším počtem uzlů v p2p síti se také zvětšuje její kapacita, naopak v případě klient-server architektury dochází k dělení zdrojů mezi uzly a snižování výkonu celého systému (čím více klientů, tím méně zdrojů na jednoho klienta) [1].

#### 2.1.2 Spolehlivost

V případě výpadku centrálního uzlu u architektury klient-server dochází k výpadku a ztrátě funkčnosti celého systému [1]. V případě výpadku jednoho uzlu u p2p s dostatečným počtem uzlů prakticky nedochází k omezení funkčnosti systému (může dojít pouze ke zmenšení kapacity systému). V případě sdílení dat nemusí ke ztrátě funkčnosti dojít ani v případě výpadku všech uzlů s kompletními daty, a to v případě, že v systému existují uzly, které složením a výměnou svých dostupných dat jsou schopny tyto kompletní data znovu vytvořit.

#### 2.1.3 Bezpečnost

V případě klient-server architektury je relativně snadné zabezpečit centrální uzel a povolit přístup a změnu dat pouze oprávněným klientům [1]. Naopak u p2p architektury je

velké nebezpečí výskytu falešných uzlů (nebo napadených uzlů), šířících špatná a potenciálně nebezpečná data.

#### **2.1.4 Aktualizace**

Aktualizace dat u architektury klient-server probíhá pouze na jednom místě a z časového hlediska zanedbatelnou dobu, oproti tomu aktualizace u architektury klient-klient musí probíhat u každého uzlu a tento proces může být velmi zdlouhavý.

Ze zmíněného porovnání je poměrně dobře vidět, že obě architektury mají své výhody i nevýhody. Také proto většinou obě bývají využívány pro odlišné typy systémů.

## **2.2 P2P v Internetu**

P2P architektura nachází v dnešní době největšího využití ve formě systémů pro distribuci a sdílení dat mezi uživateli [1]. Asi nejčastějším a nejznámějším využitím p2p ve světě internetu jsou výměnné sítě, které jsou často nesprávně označovány za ilegální. Mezi nejznámější výměnné sítě patří například Direct Connect, BitTorrent, eD2k a nyní již neexistující Napster. Výměnné sítě ale také využívají i velké firmy ke snadné a hlavně levné distribuci dat, např. většina Linuxových distribucí je k dispozici na těchto sítích, společnost Blizzard do své známé hry World of Warcraft implementovala BitTorrent klienta, který je schopen automaticky stahovat aktualizace do této aplikace, nebo například britská BBC experimentuje s používáním výměnné sítě jako s jednou z možností pro streamování videa.



### 3.1 Základní termíny<sup>2</sup>

Torrent - Soubor s meta daty popisující sdílené soubor, popřípadě soubory.

Peer – Klient v BitTorrent síti.

Swarm – Skupina peerů, kteří sdílejí stejný torrent.

Seed/Seeder – Peer, který vlastní kompletní kopii sdílených dat.

Leech/Leecher – Peer, který nevlastní kompletní kopii sdílených dat.

Tracker – Server uchovávající informace o peerech.

#### 3.1.1 Bencoding

Bencoding je způsob kódování dat, který je používán protokolem BitTorrent. Podporuje 4 datové typy: `string`, `integer`, `list`, `dictionary`.

##### String

Datový typ `string` je určen ke kódování řetězců. Řetězec je zakódován následujícím způsobem: Nejdříve je uvedeno číslo v desítkové soustavě, které odpovídá délce kódovaného řetězce, za ním následuje dvojtečka a za ní samotný řetězec. Například `4 : spam` tak odpovídá řetězci `spam`.

##### Integer

Datový typ `integer` je určen ke kódování čísel. Číslo je zakódováno následujícím způsobem: Nejdříve je uvedeno písmeno `i`, po něm kódované číslo v desítkové soustavě, za které je vloženo písmeno `e`. Například `i-10e` tak odpovídá číslu `-10`. Všechny zakódované čísla, kde hodnota čísla začíná nulou, jsou neplatné. S výjimkou `i0e`, které odpovídá číslu `0`. `i-0e` také není platnou reprezentací čísla.

##### List

Datový typ `list` je určen k reprezentaci výčtových typů. Výčtový typ je kódován následujícím způsobem: Nejdříve je uvedeno písmeno `l`, za které jsou vloženy bencode hodnoty výčtového typu, které jsou následovány písmenem `e`. Například `l3 : cow3 : mooe` odpovídá `[cow, moo]`.

##### Dictionary

Datový typ `dictionary` je určen ke kódování typů klíč, hodnota. Kódování probíhá následujícím způsobem: Nejdříve je uvedeno písmeno `d`, za které jsou vloženy dvojicemi bencode klíč, bencode hodnota, které jsou následovány písmenem `e`. Například `d3 : cow3 : moo3 : dog4 : barfe` odpovídá `{cow => moo, dog => barf}`. Klíče

---

<sup>2</sup> Zdroje tohoto oddílu: [3][4].



musí být řetězce a musí být seřazeny podle binárního porovnání řetězců. Hodnoty mohou být jakékoli typy, včetně typu list a dictionary.

## **3.2 Struktura souboru s meta informacemi (torrentu)**

Veškerá data v souboru jsou zakódována použitím bencode pravidel a všechny řetězce používají kódování UTF-8. Tento soubor má koncovku `.torrent` a jeho vnitřní struktura obsahuje dictionary s konkrétními klíči. Všechny řetězce uvnitř tohoto souboru musí používat kódování UTF-8.

### **3.2.1 Klíče torrentu<sup>3</sup>**

#### **announce**

Povinný údaj. String obsahující URL announce skriptu trackeru.

#### **announce-list**

Volitelný údaj. List obsahující další listy, které obsahují string announce URL trackerů.

#### **creation date**

Volitelný údaj. Integer obsahující POSIX čas vytvoření torrentu.

#### **comment**

Volitelný údaj. String obsahující komentář autora.

#### **created by**

Volitelný údaj. String obsahující jméno a verzi programu použitého pro vytvoření torrentu.

#### **encoding**

Volitelný údaj. String obsahující znakovou sadu použitou u klíče `pieces` v dictionary `info`.

#### **info**

Povinný údaj. Dictionary obsahující informace o souborech, které torrent popisuje.

#### **Klíče dictionary info**

##### **piece length**

Povinný údaj. Integer obsahující velikost části. Obvykle mocnina 2.

---

<sup>3</sup> Zdroje tohoto pododdílu: [3][4].

## **pieces**

Povinný údaj. String, jehož délka je násobkem 20. Skládá se z podřetězců o délce 20 znaků, které odpovídají SHA1 otisku části na odpovídajícím indexu.

## **private**

Volitelný údaj. Integer, který označuje torrent jako soukromý. Pokud se vyskytuje a má hodnotu 1, klient musí používat pouze peery dostupné od trackeru a nesmí využívat žádné externí zdroje (např. DHT). I přes to, že specifikace neuvádí nic o jeho pozici, by tento klíč měl být pro správnou funkčnost uveden jako poslední, jinak u některých klientů dochází ke špatnému zpracování info sekce torrentu (např. Azureus, BitTornado).

## **Další klíče dictionary info**

Další klíče se liší podle toho, zda byl torrent vytvořen z jednoho nebo více souborů.

### **Klíče info dictionary specifické pro torrent z jednoho souboru**

#### **name**

Povinný údaj. String obsahující název souboru.

#### **length**

Povinný údaj. Integer obsahující velikost souboru v bytech.

#### **md5sum**

Volitelný údaj. String obsahující 32 znaků dlouhý řetězec odpovídající md5 otisku souboru.

### **Klíče info dictionary specifické pro torrent z více souborů**

#### **name**

Povinný údaj. String obsahující název adresáře kam se mají uložit stahované soubory.

#### **files**

Povinný údaj. List dictionary údajů, obsahujících klíče `length`, `md5sum` a `path`. Každý dictionary odpovídá jednomu souboru. Klíče `length` a `md5sum` mají stejný význam jako u údajů pro jeden soubor. Klíč `path` je list údajů, které reprezentují cestu a název souboru. Např. `17:slozkaA7:slozkaB10:soubor.txt` odpovídá

slozkaA/slozkaB/soubor.txt. Klíč md5sum je volitelný, zbylé dva jsou povinné.

### 3.3 Druhy komunikace

V protokolu BitTorrent jsou definovány 2 způsoby komunikace [3][4]. Komunikace mezi klienty, tzv. peery, při které dochází k aktuální výměně souborů a komunikace klient-tracker, která je nutná jako prostředek pro řízení sdílení. Tato práce se zabývá pouze implementací trackeru, proto se popisem komunikace klient-klient (tzv. Peer wire protocol) podrobněji zabývat nebudu.

#### 3.3.1 Komunikace klient-tracker<sup>4</sup>

##### Co to je tracker

Tracker je HTTP nebo HTTPS služba, která odpovídá na konkrétní žádosti zaslané metodou GET. Žádosti zasílané klienty obsahují údaje, které pomáhají trackeru udržovat statistiky o torrentech. URL žádosti se skládá z URL uvedené v torrentu (announce) a parametrů žádosti, které posílá klient. Všechny binární data v URL musí být správně ošetřeny dle RFC1738. Tyto žádosti bývají někdy také nazývány jako announce.

##### Parametry žádosti klienta na tracker.

###### **info\_hash**

Povinný údaj. SHA1 otisk info části torrentu.

###### **peer\_id**

Povinný údaj. 20 bytový unikátní řetězec identifikující klienta. Tuto hodnotu si generuje klient při každém startu.

###### **port**

Povinný údaj. Číslo portu, na kterém klient naslouchá.

###### **uploaded**

Povinný údaj. Celkový upload klienta od startu v desítkové soustavě. Specifikace neudává žádný údaj, o jaké jednotky se jedná, ale obecným pravidlem je používání bytů.

###### **downloaded**

Povinný údaj. Celkový download klienta od startu v desítkové soustavě. Specifikace neudává žádný údaj, o jaké jednotky se jedná, ale obecným pravidlem je používání bytů.

---

<sup>4</sup> Zdroje tohoto pododdílu: [3][4].

## **left**

Povinný údaj. Počet bytů, které klientovi zbývají k dokončení souboru, v desítkové soustavě.

## **compact**

Volitelný údaj. Nastavení toho údaje na 1 označuje, že klient je schopen přijmout list peerů v kompaktním tvaru.

## **no\_peer\_id**

Volitelný údaj. Pokud je nastaven tracker může při odpovědi v dictionary peers vynechat klíč peer id. Při nastavení compact na 1 je tento parametr ignorován.

## **event**

Volitelný údaj. Pokud je specifikován musí obsahovat buď prázdný řetězec (stejně jako když není specifikován), nebo jednu z hodnot `started`, `completed`, `stopped`. `Started` je posíláno při startu torrentu, `completed` při dokončení stahování a `stopped` při zastavení torrentu. Pokud tento parametr není uveden, jedná se o běžnou periodickou žádost klienta na tracker.

## **ip**

Volitelný údaj. Obsahuje IPv4 adresu klienta v tečkové notaci, nebo IPv6 adresu ve formátu definovaným RFC3513.

## **numwant**

Volitelný údaj. Číslo obsahující počet peerů, které klient žádá od trackeru. Povolena je i nulová hodnota.

## **key**

Volitelná hodnota. Klíč sloužící k identifikaci klienta v případě změny jeho IP adresy.

## **Odpověď trackeru**

Tracker na HTTP žádosti klientů odpovídá dokumentem s nastaveným MIME typem `text/plain`. Tento dokument obsahuje dictionary. Odpověď trackeru může buď obsahovat pouze klíč `failure reason`, jehož hodnotou je bencoded řetězec obsahující chybovou zprávu vysvětlující proč chybě došlo, nebo obsahuje standardní klíče odpovědi trackeru.

## Klíče odpovědi trackeru

### **interval**

Povinný údaj. Integer obsahující počet sekund, které má klient čekat před zasláním další žádosti trackeru.

### **min interval**

Volitelný údaj. Integer obsahující počet sekund kdy klient nesmí znovu zaslat požadavek na tracker.

### **complete**

Volitelný údaj. Integer obsahující počet seederů ve swarmu torrentu.

### **incomplete**

Volitelný údaj. Integer obsahující počet leecherů ve swarmu torrentu.

### **peers**

Povinný údaj. V případě odpovědi v kompaktním tvaru se jedná o string složený z řetězců o velikosti 6 bytů. První 4 byty obsahují IP adresu a další 2 port, oboje v big endian notaci. V opačném případě se jedná o list, který obsahuje pro každého peera dictionary s klíči `peer_id`, `ip`, `port`. `Peer_id` obsahuje hodnotu řetězce `peer_id` z žádostí klientů, `ip` obsahuje IPv4 adresu v tečkové notaci nebo IPv6 adresu dle RFC3513, `port` obsahuje číslo portu peera.

## **Scrape**

Většina trackerů také podporuje další formu žádosti, která bývá nazývána jako scrape. Tato žádost zjišťuje statistiky o konkrétních (nebo všech) torrentech, které tracker sleduje. Scrape žádosti také probíhají pomocí protokolu HTTP metodou GET. Scrape URL a podpora scrape žádostí je zjištěna z announce URL. Pokud se za posledním znakem / v announce URL nachází řetězec announce, potom tracker scrape podporuje. Scrape URL je pak získána nahrazením tohoto řetězce announce za řetězec scrape. Např. pokud je announce URL `http://tracker.com/aaaa/announce.asp`, potom je scrape URL `http://tracker.com/aaaa/scrape.asp`, pokud by announce URL byla `http://tracker.com/aaaa/an`, popř. `http://tracker.com/xxannounce.php`, pak scrape není podporován.

### **Scrape žádost**

Scrape žádost může buď probíhat bez jakýchkoli parametrů, a v tom případě tracker vrací statistiky o všech sledovaných torrentech, nebo také s parametrem `info_hash`, který určuje torrent, jehož statistiky klient požaduje. Tento parametr

bývá mnohdy v žádosti několikrát opakován, pro zjištění informací o více specifických torrentech naráz, a to i přes to, že opakování tohoto parametru není v oficiální specifikaci uvedeno.

### **Odpověď na scrape žádost**

Tracker na scrape žádost odpovídá dokumentem s nastaveným MIME typem `text/plain`. Tento dokument obsahuje dictionary s klíčem `files`, jehož hodnotou jsou další dictionary (pro každý torrent o kterém jsou zasílány statistiky jeden). Klíčem těchto dictionary je `info_hash` torrentu a jejich hodnotou je další dictionary, který obsahuje klíče `complete`, `downloaded`, `incomplete`, `name`. Tyto klíče jsou až na `name` povinnými údaji. `Complete` je integer, který udává počet seederů ve swarmu torrentu, obdobně pak `incomplete` udává počet leecherů. `Downloaded` je integer, který udává kolikrát tracker na daném torrentu zaznamenal `event completed`, tzv. kolik klientů dokončilo stahování daného torrentu. `Name` je string, který udává interní jméno torrentu, tj. řetězec uvedený v klíči `name`, v `info` sekci torrentu.

## 4 Analýza aplikace

### 4.1 O aplikaci

Primárním cílem této práce bylo vytvoření systému pro správu a sdílení souborů s využitím protokolu BitTorrent. Tento systém by měl svým uživatelům přinášet jednoduchou možnost jak sdílet uživatelský obsah navzájem mezi sebou (s využitím torrent souborů), komentovat tento obsah a zároveň sloužit jako BitTorrent tracker pro tyto soubory.

### 4.2 Cíle aplikace

- Přístup pouze pro registrované uživatele.
- Možnost vytváření a odebrání uživatelských rolí v systému.
- Private BitTorrent tracker, tj. tracker, který umožňuje přístup pouze registrovaným uživatelům.
- Možnost stahování pouze pro uživatele, kteří jsou aktivně zapojeni do komunity (vynucení poměru mezi celkovým objemem stažených a sdílených dat uživatele).
- Možnost vyhledávání v nahraných souborech.
- Možnost komentářů k nahraným souborům.
- Možnost relativně snadné úpravy vzhledu a rozložení stránek.
- Relativně snadná možnost budoucího rozšíření.

### 4.3 Uživatelské role v systému

I přes to že uživatelské role v systému nejsou pevně dané a správce dané aplikace má možnost je měnit dle svého uvážení, se základní role nechají vymezit do 3 skupin – administrátor, registrovaný uživatel a návštěvník (neregistrovaný uživatel).

#### 4.3.1 Návštěvník

Role návštěvníka by v systému měla být schopna pouze 3 základních úkonů:

- prohlížení úvodní stránky,
- přihlášení,
- registrace.

#### 4.3.2 Registrovaný uživatel

Role registrovaného uživatele je základní uživatelskou rolí v systému. Registrovaný uživatel by měl být schopen těchto úkonů:

- prohlížení úvodní stránky,
- nahrání nového .torrent souboru,
- prohlížení seznamu torrentů,
- vyhledávání v seznamu torrentů,
- stažení torrentu,

- přidávání komentářů k torrentům,
- zobrazení uživatelských detailů,
- odhlášení.

### 4.3.3 Administrátor

Role administrátor je rolí odvozenou od registrovaného uživatele. Tato role by měla být schopna všech úkonů uživatele, ale navíc také:

- zakázání („ban“) účtu uživatele,
- zrušení banu účtu uživatele,
- varování uživatele,
- smazání uživatele,
- zakázání („ban“) torrentu,
- zrušení banu torrentu.

## 4.4 Dostupné alternativy

Vzhledem k poměrně velké popularitě protokolu BitTorrent existuje značné množství více či méně zdařilých implementací systémů, které by mohly splňovat výše zmíněné požadavky na systém. Z tohoto velkého množství bych se zaměřil na 3 z nejpoužívanějších a to: TBDev.net<sup>5</sup>, xbtit<sup>6</sup> a phpMyBitTorrent<sup>7</sup>.

### 4.4.1 TBDev.net

#### Výhody:

- velká uživatelská komunita
- velké množství dostupných modifikací

#### Nevýhody:

- chyby zabezpečení
- nemožnost vytváření nových uživatelských rolí
- prolínání PHP a HTML kódu (těžkopádné úpravy rozložení stránek)
- příliš velká robustnost (v případě nasazení pro účely cílové aplikace by většina funkcí zůstala nevyužita, popřípadě by zpomalovala systém)

### 4.4.2 xbtit

#### Výhody:

- velká uživatelská komunita
- velké množství dostupných modifikací

<sup>5</sup> Více o aplikaci na adrese: <http://www.tbdev.net>.

<sup>6</sup> Více o aplikaci na adrese: <http://www.btiteam.org>.

<sup>7</sup> Více o aplikaci na adrese: <http://phpmybittorrent.com>.



- možnost náhrady implementace trackeru v PHP za C++ implementaci (rychlejší, s menšími nároky)
- šablonovací systém s podporou cache
- přehledná struktura kódu

#### **Nevýhody:**

- příliš velká robustnost (v případě nasazení pro účely cílové aplikace by většina funkcí zůstala nevyužita, popřípadě by zpomalovala systém)
- placený přístup k některým modifikacím

#### **4.4.3 phpMyBitTorrent**

##### **Výhody:**

- dobrá podpora na oficiálním fóru
- podpora eD2k
- možnost nastavení komu uživatel bude a nebude sdílet torrent

##### **Nevýhody:**

- chyby zabezpečení
- nemožnost vytváření nových uživatelských rolí
- prolínání PHP a HTML kódu (těžkopádné úpravy rozložení stránek)
- chyby v PHP kódu (např. porušování zapouzdření)
- příliš velká robustnost (v případě nasazení pro účely cílové aplikace by většina funkcí zůstala nevyužita, popřípadě by zpomalovala systém)

#### **4.5 UML**

K vizuálnímu vyjádření analýzy a pro podporu návrhu aplikace jsem využil Unified Modeling Language, zkráceně také UML. Tento jazyk je dnes již standardem v oblasti softwarového inženýrství a jeho použití jeho podporováno širokým spektrem programů. Jeho využití je velmi široké, např. vizualizace návrhu, určování specifikací, popis struktury dat. UML zahrnuje širokou řadu dostupných diagramů, z nichž jsem ve své části použil pouze některé, a to Class, Use case a Activity diagram. K vytvoření těchto diagramů bylo využito programu Visual Paradigm for UML 6.0 Community Edition<sup>8</sup>.

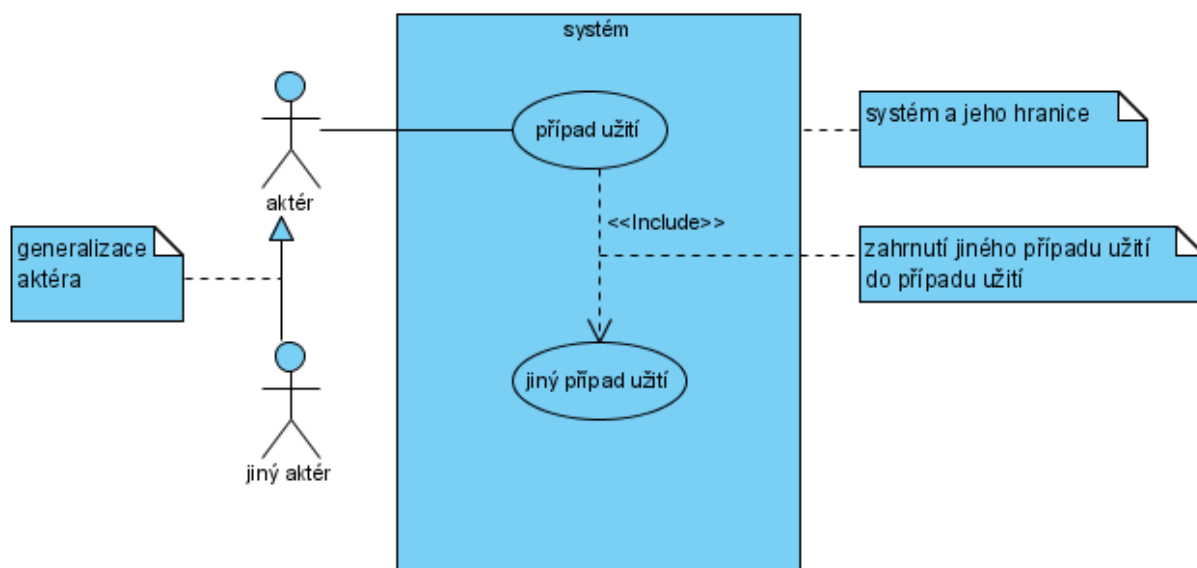
##### **4.5.1 Use case diagram**

Use case diagram neboli diagram užití (případy užití) patří do skupiny UML diagramů chování. Tento diagram zachycuje interakce mezi systémem a aktéry (nejčastěji uživateli), kteří od systému vyžadují nějakou akci. Případy užití mohou mít formu diagramů, ale i textů.

---

<sup>8</sup> Visual Paradigm for UML, nástroj pro vytváření grafů v jazyce UML, více o produktu na adrese: <http://www.visual-paradigm.com/product/vpum/>.

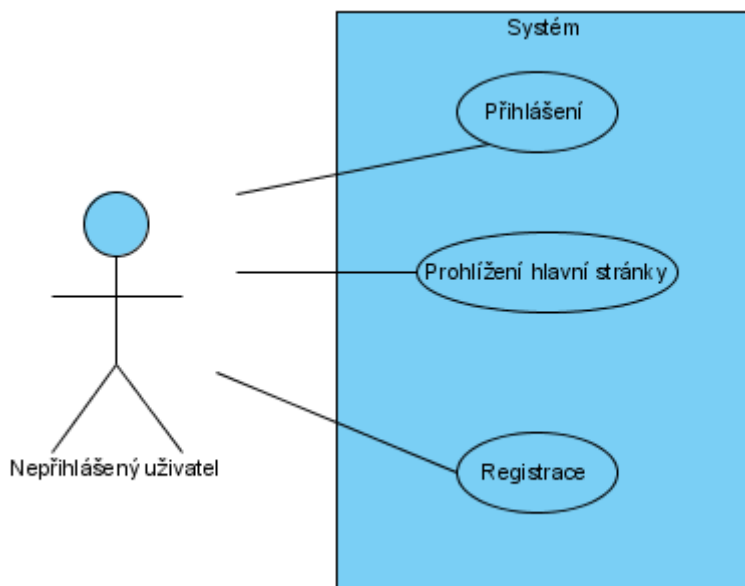
## Příklad diagramu užití



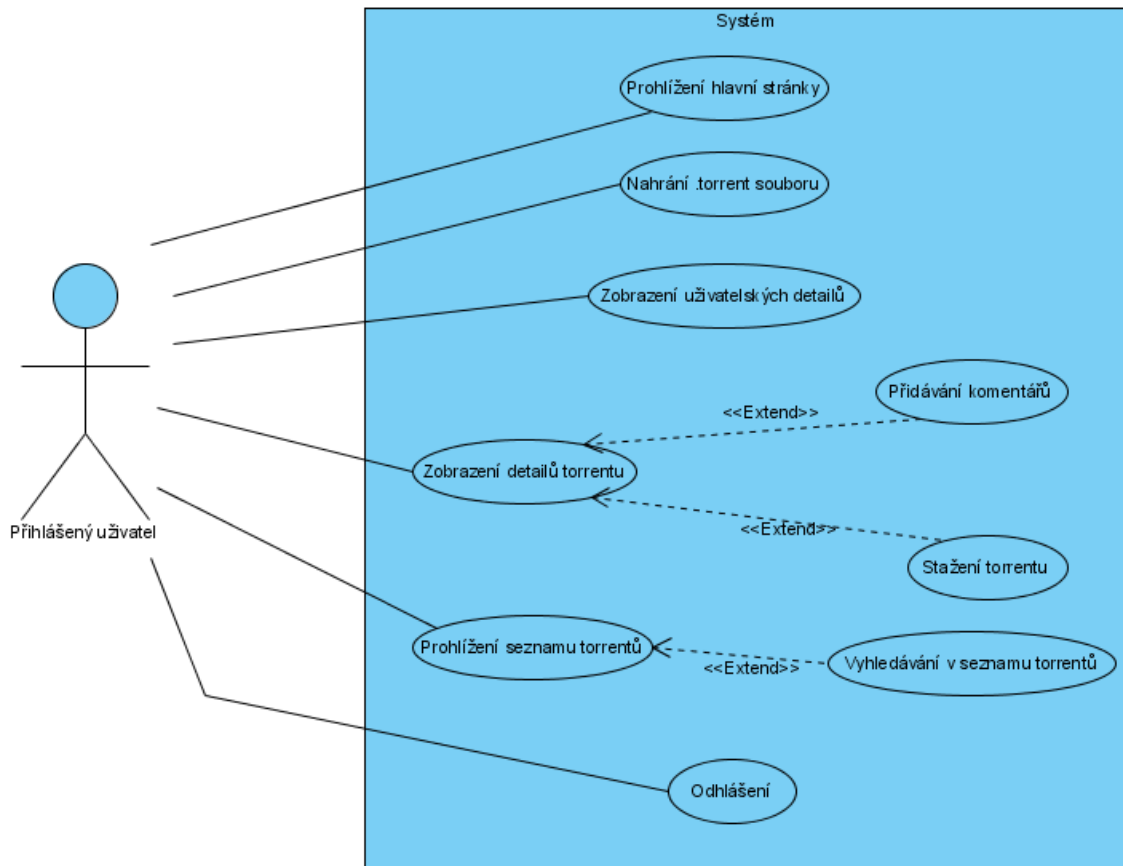
Obrázek 3 – Příklad diagramu užití, zdroj: vlastní

## Diagramy užití modelovaného systému

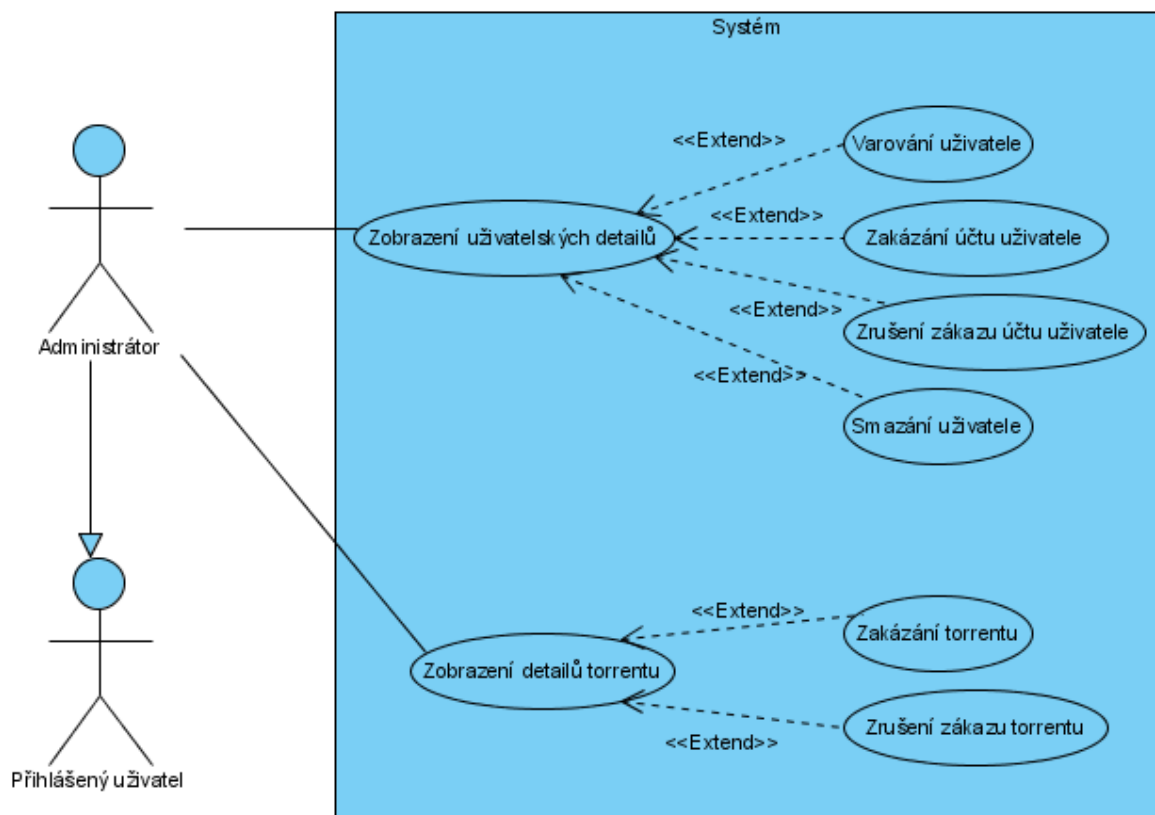
Následující diagramy užití popisují možnosti již zmíněných základních uživatelských rolí v systému.



Obrázek 4 – Diagram užití s nepřihlášeným uživatelem, zdroj: vlastní



Obrázek 5 – Diagram užití s přihlášeným uživatelem, zdroj: vlastní

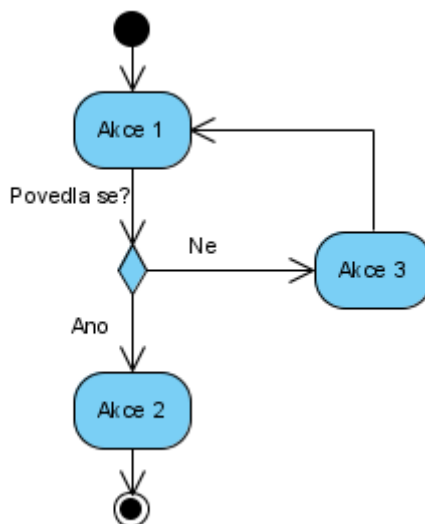


Obrázek 6 – Diagram užití s administrátorem, zdroj: vlastní

## 4.5.2 Activity diagram

Activity diagram neboli diagram aktivit patří, stejně jako Use case, do skupiny UML diagramů chování. Tento diagram reprezentuje toky aktivit business procesů a operací krok po kroku. Skládá se z operací a přechody mezi nimi. Tento diagram má také počáteční a koncový stav.

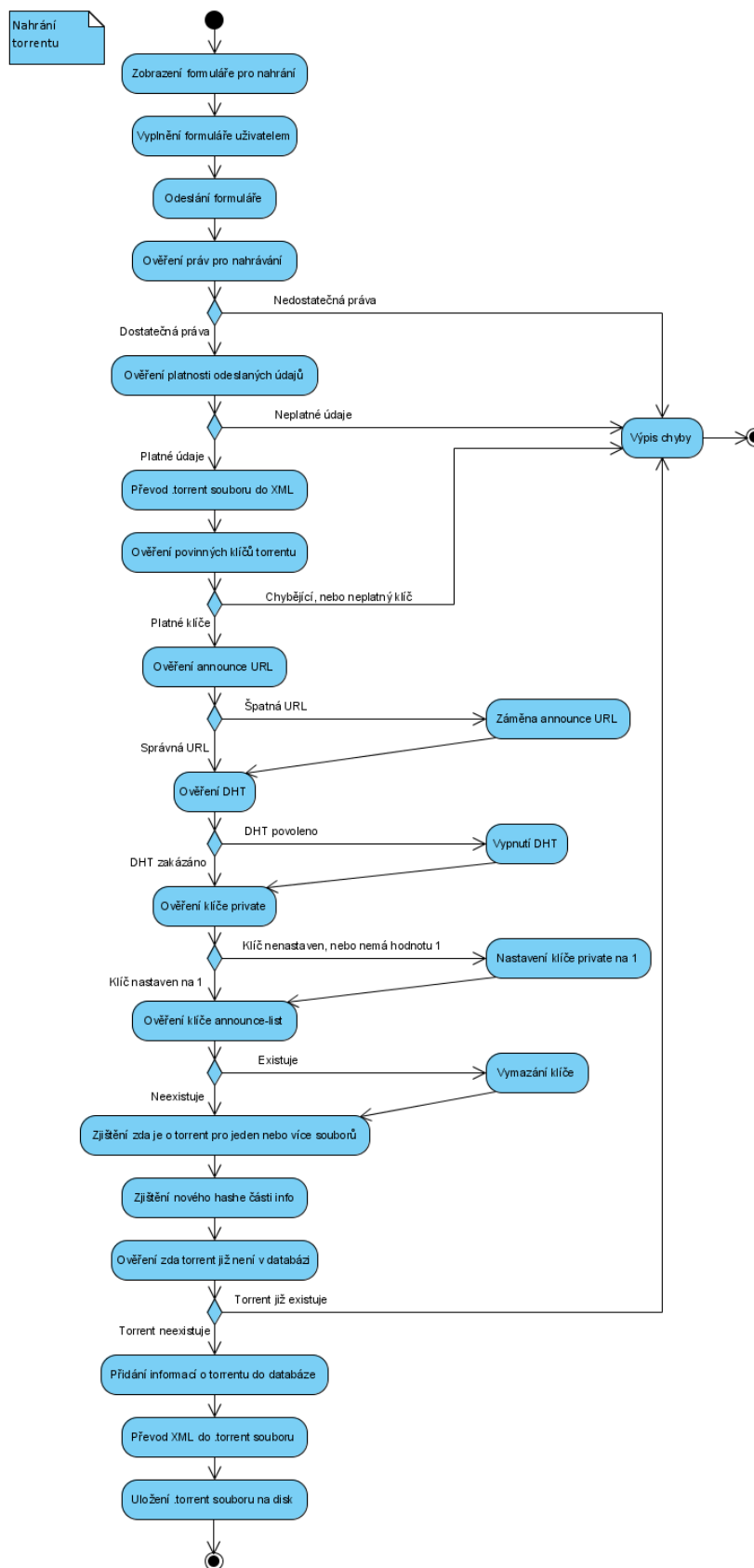
### Příklad diagramu aktivit



Obrázek 7 – Příklad diagramu aktivit, zdroj: vlastní

Následující diagramy aktivit popisují dvě nejdůležitější části takzvaného soukromého (private) trackeru a to nahrání torrentu a zpracování požadavku klienta na tracker. První diagram detailně popisuje způsob ověření formulářových dat a změnu informací v torrentu s využitím převodu do XML. Na druhém je popsáno ověření uživatele, torrentu, zaslaných údajů a rozhodování o odpovědi na požadavek na základě takzvané události (parametr `event`).

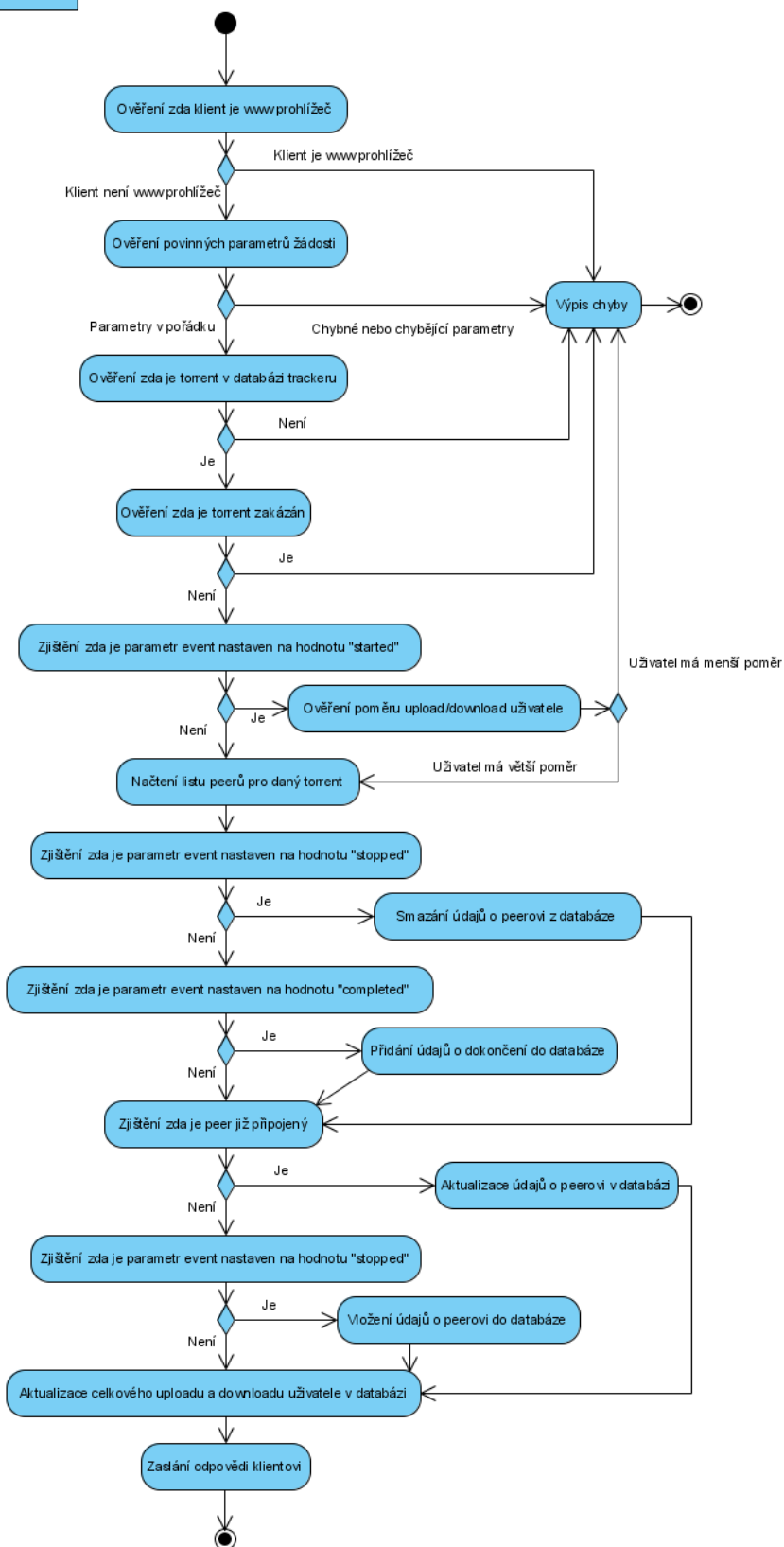
## Diagram aktivit nahrání torrentu



Obrázek 8 – Diagram aktivit nahrání torrentu, zdroj: vlastní

## Diagram aktivit odpovědi na požadavek na tracker

Odpověď na požadavek na tracker



Obrázek 9 – Diagram aktivit odpovědi trackeru, zdroj: vlastní

### 4.5.3 Class diagram

Class diagram neboli diagram tříd patří do skupiny UML diagramů popisujících strukturu systému. Používá se k popisu objektově orientovaných systémů a umožňuje zobrazit třídy, jejich atributy a metody a zobrazovat vztahy mezi nimi.

#### Příklad diagramu tříd



Obrázek 10 – Příklad diagramu tříd, zdroj: vlastní

## 5 Zvolené technologie

Základní funkcí aplikace je evidence velkého množství dat o BitTorrent klientech, souborech s meta informacemi a uživateli. Pro ukládání těchto údajů jsem se rozhodl použít databázový systém kvůli jeho rychlosti, jednoduchosti práce a podpoře současného přístupu více uživatelů. Jako systém řízení báze dat jsem zvolil MySQL. K tomuto rozhodnutí mě vedla jeho velká rozšířenost, kvalitní dokumentace a dostupnost Community Edition pod GNU GPL.

Pro vytvoření aplikace jsem jako programovací jazyk zvolil PHP. Pro tento jazyk jsem se rozhodl díky jeho velké rozšířenosti, kvalitní dokumentaci, dostupnosti pod Open Source licencí PHP License, ale také díky mým předchozím zkušenostem s tímto jazykem.

### 5.1 PHP

PHP neboli Hypertext Preprocessor je skriptovací jazyk vhodný pro vyvíjení webových aplikací [5][6][7]. Tento jazyk je interpretovaný, to znamená, že kód před nasazením neprochází překladem do výkonného kódu, ale při každém použití se kód znovu interpretuje. Syntaxí vychází z C, Javy a Perlu a díky těmto vlastnostem je poměrně lehký na naučení. Hlavním cílem jazyka je umožnit vývojářům snadný a rychlý vývoj aplikací. PHP je šířeno pod Open Source licencí ještě volněji než GNU GPL, a tudíž je pro běžné použití zdarma. PHP podporuje jak procedurální způsob programování, tak i objektově orientovaný způsob, popřípadě kombinaci obou zmíněných, avšak dobrá podpora pro OOP je až od verze 5. PHP lze nasadit na velké řadě platform od Unixových systémů, přes Microsoft Windows, RISC OS a další. Také podporuje velkou řadu webových serverů, např. Apache, IIS, Netscape, Personal Web Server. Na většině z nich může fungovat buď jako modul, nebo jako CGI procesor. PHP má také podporu práce s širokým spektrem databázových systémů, např. Oracle, MySQL, PostgreSQL, ODBC, Informix.

Pro tvorbu aplikace jsem se rozhodl použít kombinaci procedurálního a objektově orientovaného způsobu programování. Pro lepší přehlednost a zjednodušení budoucích rozšíření aplikace jsem se rozhodl využít návrhový vzor MVC – Model-View-Controller.

Pro psaní kódu aplikace jsem využil NetBeans IDE 6.8<sup>9</sup>.

### 5.2 MySQL

MySQL je v současné době světově nejpopulárnější open source databázový systém [8][9]. Mezi nejznámější organizace používající MySQL patří například Google, Nokia, Wikipedia nebo Facebook. MySQL je dostupné jak pod GNU GPL, tak pod několika proprietárními licencemi. Vyvíjeno je firmou MySQL AB, kterou později koupil Sun Microsystems, jenž je nyní součástí Oracle Corporation. MySQL má podporu širokého spektra platform, například Microsoft Windows, Unixové systémy, IBM i5/OS. Také je podporováno velkou řadou programovacích jazyků, např. C, C++, C#, Java, PHP, Perl, Python, Ruby. MySQL podporuje jak transakční tak netransakční zpracování požadavků.

---

<sup>9</sup> NetBeans IDE, vývojové prostředí, více o aplikaci na adrese: <http://netbeans.org>.



MySQL podporuje několik takzvaných tabulkových enginů, které se liší svými možnostmi, použitím a způsobem fyzického ukládání dat.

Při tvorbě návrhu databázové struktury aplikace jsem se rozhodl využít úložiště MyISAM [10][11]. Mezi typické vlastnosti tohoto úložiště patří nízké nároky na místo na disku a v paměti, velká rychlost, netransakční zpracování a absence podpory cizích klíčů.

### **5.3 Návrh databázové struktury**

Pro grafické znázornění návrhu jsem použil entity-relationship diagram. Pro lepší přehlednost uvádím v tabulkách pouze klíče a vynechávám ostatní atributy, ty jsou dále uvedeny pro každou tabulku v textové podobě. Pro vytvoření diagramu jsem použil Toad Data Modeler 3<sup>10</sup>.

---

<sup>10</sup> Toad Data Modeler, nástroj pro návrh databáze, více o aplikaci na adrese: [http://www.toadsoft.com/toaddm/toad\\_data\\_modeler.htm](http://www.toadsoft.com/toaddm/toad_data_modeler.htm).



### 5.3.2 Popis databázových tabulek

Databázové tabulky budou popsány následujícím způsobem:

Tabulka 1 – Příklad popisu databázových tabulek, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
Primární klíč tabulky	int	PRIMARY KEY
Atribut1	varchar(30)	NOT NULL
Atribut2	int	NOT NULL

Datový typ uvedený v tabulce odpovídá datovému typu MySQL, stejné platí i pro sloupec omezení, upřesnění.

Databázová tabulka `users` slouží k evidenci registrovaných uživatelů systému.

Tabulka 2 – Popis databázové tabulky `users`, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
id	int	PRIMARY KEY auto_increment
username	varchar(40)	NOT NULL UNIQUE
password	varchar(40)	NOT NULL
reg_email	varchar(320)	NOT NULL UNIQUE
email	varchar(320)	NOT NULL UNIQUE
reg_date	datetime	NOT NULL
avatar	varchar(5)	
level	int	NOT NULL
uploaded	bigint	NOT NULL default 0
downloaded	bigint	NOT NULL default 0
passkey	varchar(40)	NOT NULL UNIQUE
activation_key	varchar(40)	NOT NULL
active	tinyint(1)	NOT NULL default 0
deleted	tinyint(1)	NOT NULL default 0
reg_ip	int(10)	NOT NULL
last_ip	int(10)	NOT NULL
lastlogin	datetime	
user_agent	text	NOT NULL

Databázová tabulka `torrents` slouží k evidenci nahraných torrentů.

Tabulka 3 – Popis databázové tabulky `torrents`, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
id	int	PRIMARY KEY auto_increment
info_hash	varchar(40)	NOT NULL
md5sum	varchar(32)	NOT NULL
name	varchar(255)	NOT NULL
filename	varchar(255)	NOT NULL
size	bigint	NOT NULL

Název atributu	Datový typ	Omezení, upřesnění
show_date	date	
description	text	NOT NULL
type	enum('single','multi')	NOT NULL
numfiles	smallint(6)	NOT NULL
added	datetime	NOT NULL
uploader	int	NOT NULL
last_seeder	datetime	

Databázová tabulka `peers` slouží ke evidenci klientů aktuálně využívajících tracker.

Tabulka 4 – Popis databázové tabulky `peers`, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
id	int	PRIMARY KEY auto_increment
user_id	int	NOT NULL
torrent_id	int	NOT NULL
peer_id	varchar(20)	NOT NULL
unique_id	varchar(20)	NOT NULL
ip	int(10)	NOT NULL
port	smallint(5) unsigned	NOT NULL
uploaded	bigint	NOT NULL default 0
downloaded	bigint	NOT NULL default 0
up_speed	int	NOT NULL default 0
down_speed	int	NOT NULL default 0
to_go	bigint	NOT NULL default 0
seeder	tinyint(1)	NOT NULL default 0
started	datetime	
last_action	datetime	
connectable	tinyint(1)	NOT NULL default 0
client	varchar(30)	
version	varchar(10)	
user_agent	varchar(100)	NOT NULL

Databázová tabulka `categories` slouží k uchování údajů o kategoriích torrentů.

Tabulka 5 – Popis databázové tabulky `categories`, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
id	int	PRIMARY KEY auto_increment
name	varchar(20)	NOT NULL
picture	varchar(255)	NOT NULL

Databázová tabulka `completed` slouží k uchování údajů o dokončení stahování torrentů uživateli.

**Tabulka 6 – Popis databázové tabulky `completed`, zdroj: vlastní**

Název atributu	Datový typ	Omezení, upřesnění
<code>torrent_id</code>	<code>int</code>	PRIMARY KEY
<code>user_id</code>	<code>int</code>	PRIMARY KEY
<code>when</code>	<code>datetime</code>	PRIMARY KEY

Databázová tabulka `downloaded` slouží k uchování údajů o stažení torrent souborů uživateli.

**Tabulka 7 – Popis databázové tabulky `downloaded`, zdroj: vlastní**

Název atributu	Datový typ	Omezení, upřesnění
<code>torrent_id</code>	<code>int</code>	PRIMARY KEY
<code>user_id</code>	<code>int</code>	PRIMARY KEY
<code>when</code>	<code>datetime</code>	PRIMARY KEY

Databázová tabulka `levels` slouží k evidenci údajů o skupinách uživatelů a jejich právech.

**Tabulka 8 – Popis databázové tabulky `levels`, zdroj: vlastní**

Název atributu	Datový typ	Omezení, upřesnění
<code>id</code>	<code>int</code>	PRIMARY KEY
<code>group_name</code>	<code>varchar(20)</code>	NOT NULL
<code>can_download</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_upload</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_edit_users</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_edit_torrents</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_edit_news</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_delete_users</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_delete_torrents</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_access_admin_panel</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_access_mod_panel</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_ban_users</code>	<code>tinyint(1)</code>	NOT NULL default 0
<code>can_ban_torrents</code>	<code>tinyint(1)</code>	NOT NULL default 0

Databázová tabulka `torrents_categories` slouží jako vazební tabulka mezi tabulkami `torrents` a `categories`.

Tabulka 9 – Popis databázové tabulky torrents\_categories, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
torrent_id	int	NOT NULL
category_id	int	NOT NULL

Databázová tabulka torrent\_comments slouží k uchovávání informací o komentářích k torrentům.

Tabulka 10 – Popis databázové tabulky torrent\_comments, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
id	int	PRIMARY KEY auto_increment
torrent_id	int	NOT NULL
user_id	int	NOT NULL
comment_text	text	NOT NULL
added	datetime	NOT NULL

Databázová tabulka torrent\_bans slouží k uchování záznamů o zakázaných torrentech.

Tabulka 11 – Popis databázové tabulky torrent\_bans, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
torrent_id	int	PRIMARY KEY
when	datetime	NOT NULL
reason_displayed	text	NOT NULL
reason_full	text	NOT NULL
banned_by	int	NOT NULL

Databázová tabulka user\_bans slouží k uchování záznamů o uživateli se zakázaným účtem.

Tabulka 12 – Popis databázové tabulky user\_bans, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
user_id	int	PRIMARY KEY
when	datetime	NOT NULL
reason_displayed	text	NOT NULL
reason_full	text	NOT NULL
banned_by	int	NOT NULL

Databázová tabulka `user_warning` slouží k uchování záznamů o varováních udělených uživatelům.

Tabulka 13 – Popis databázové tabulky `user_warning`, zdroj: vlastní

Název atributu	Datový typ	Omezení, upřesnění
<code>user_id</code>	int	PRIMARY KEY
<code>when</code>	datetime	PRIMARY KEY
<code>reason_displayed</code>	text	NOT NULL
<code>reason_full</code>	text	NOT NULL
<code>warned_by</code>	int	PRIMARY KEY

## 5.4 Popis implementace návrhového vzoru MVC

MVC, neboli Model-View-Controller, je softwarová architektura, často také chápána jako návrhový vzor, popřípadě architektonický vzor, která odděluje prezentační, aplikační a datovou vrstvu aplikace [12]. Tyto tři vrstvy jsou rozděleny do částí model, view a controller.

### 5.4.1 Model (model)

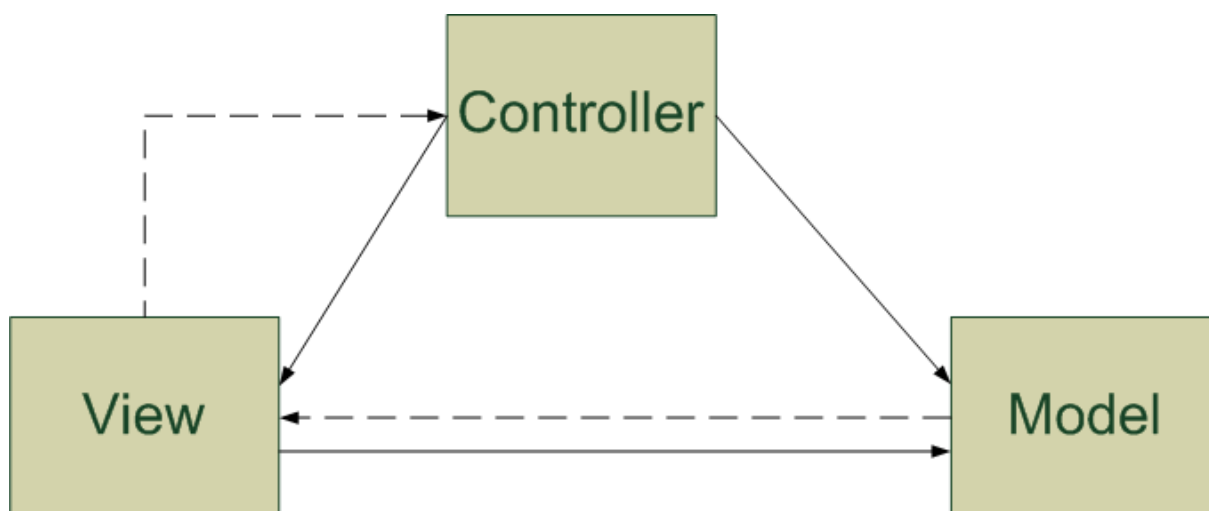
Model je specifická reprezentace dat, s kterými pracuje aplikace.

### 5.4.2 View (pohled)

View převádí data z modelu do formy výstupu, typicky například do formy uživatelského rozhraní.

### 5.4.3 Controller (kontrolér)

Controller reaguje na události a zajišťuje zpracování odezvy. Komunikuje s modelem a pohledem.



Obrázek 12 – Znázornění návrhového vzoru MVC, zdroj: vlastní

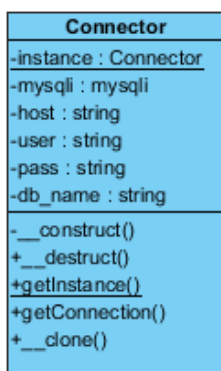
V mém případě jsou kontroléry php soubory umístěné v kořenovém adresáři aplikace. V případě, že ke kontroléru existuje model, pak je umístěn v adresáři `include/data/` se stejným názvem jako kontrolér. V případě, že ke kontroléru existuje pohled, pak je umístěn v adresáři `include/view/`, taktéž se stejným názvem jako kontrolér. Soubory ve složce `include/data` vždy obsahují třídu se stejným názvem jako má kontrolér, ale s velkým prvním počátečním písmenem a příponou `Data` (například pro kontrolér `login.php` existuje třída `LoginData`), tyto třídy se starají o operace modelu.

Jako přístup k datovému úložišti je využívána třída `Connector` umístěná v `include/classes/Connector.php`. Pro eliminaci vícenásobných připojení do databáze v rámci jednoho připojeného uživatele tato třída implementuje návrhový vzor `Singleton`.

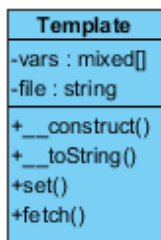
Pro načtení a práci s pohledy kontroléry využívají třídu `Template`, umístěnou v `include/classes/Template.php`. Tato třída je upravenou a rozšířenou třídou `Template` od Briana Loziera<sup>11</sup>.

#### 5.4.4 Class diagram použitých tříd

Následující dva diagramy zobrazují již zmíněné pomocné třídy `Connector` a `Template`; `Connector` je používána jako prostředek pro přístup k databázi a `Template` slouží k načítání souborů se šablonami.



Obrázek 13 - Class Diagram pomocné třídy `Connector`, zdroj: vlastní

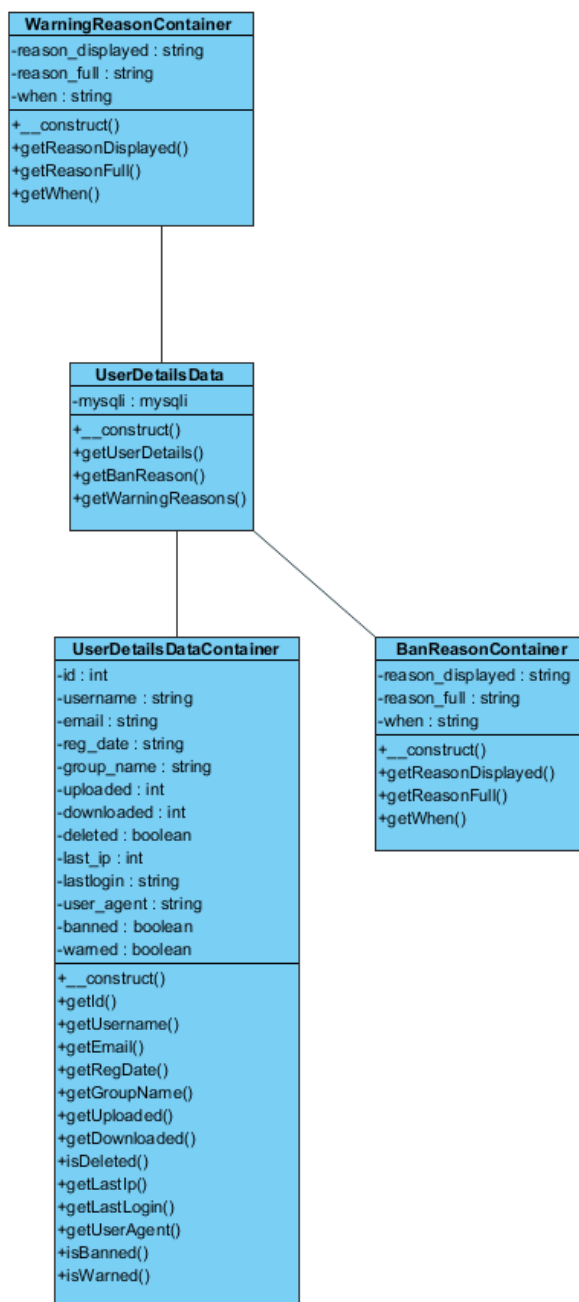


Obrázek 14 – Class diagram pomocné třídy `Template`, zdroj: vlastní

<sup>11</sup> Více informací na adrese: [http://www.massassi.com/php/articles/template\\_engines/](http://www.massassi.com/php/articles/template_engines/).

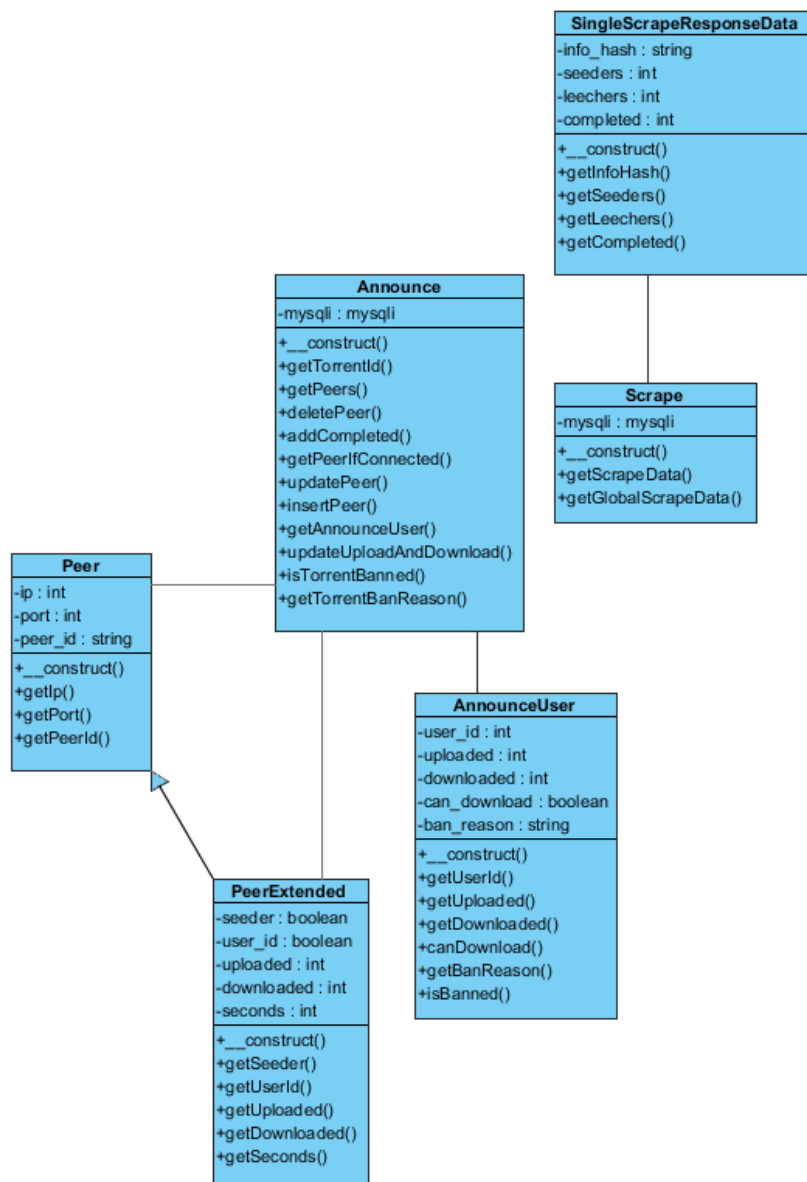


Následující diagramy popisují třídy modelu, pro přehlednost jsou rozděleny do pěti obrázků. První diagram popisuje třídy pro výpis detailů uživatele.



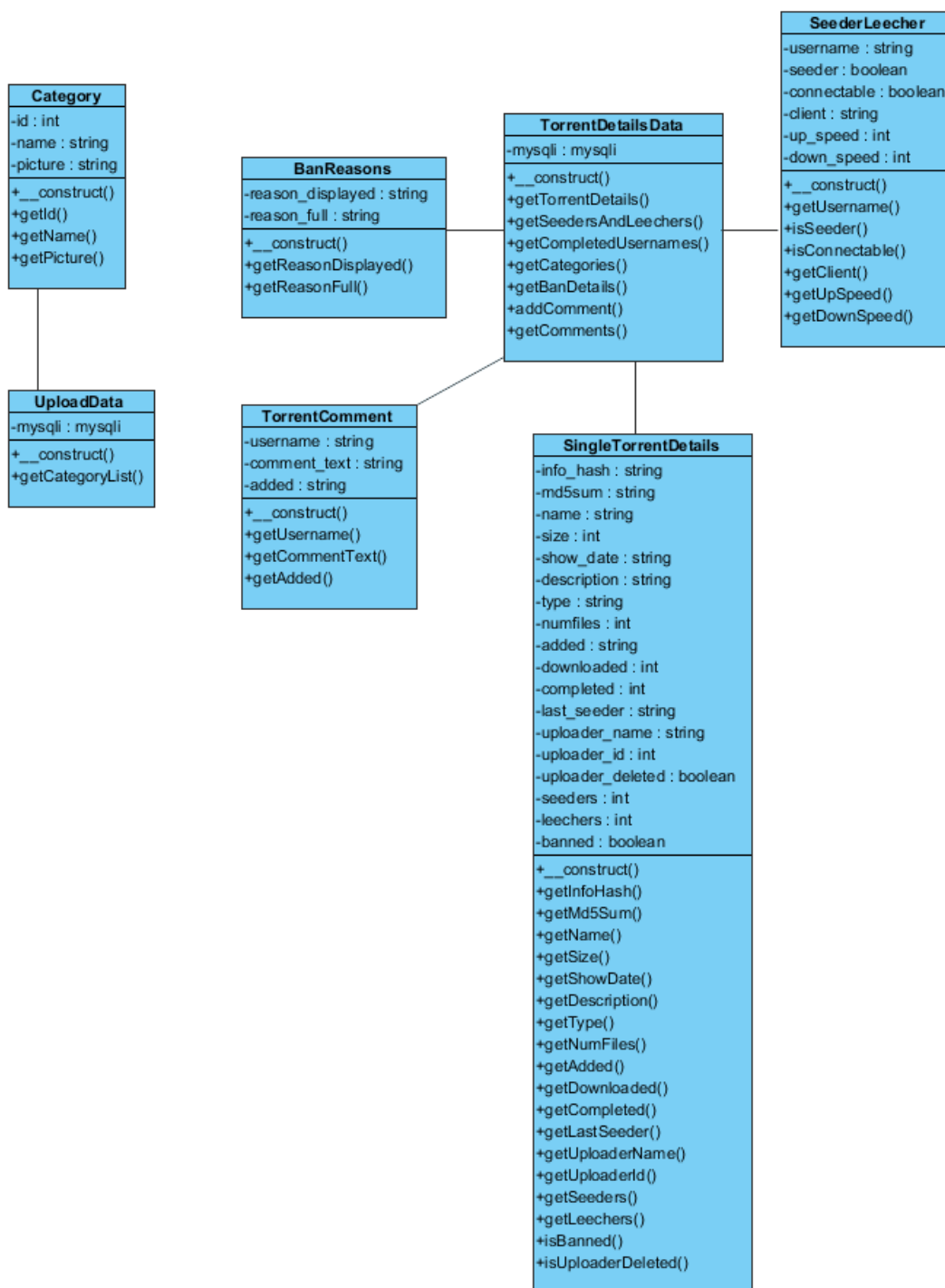
Obrázek 15 – Class diagram tříd modelu 1, zdroj: vlastní

Následující diagram popisuje třídy týkající se funkcí trackeru.



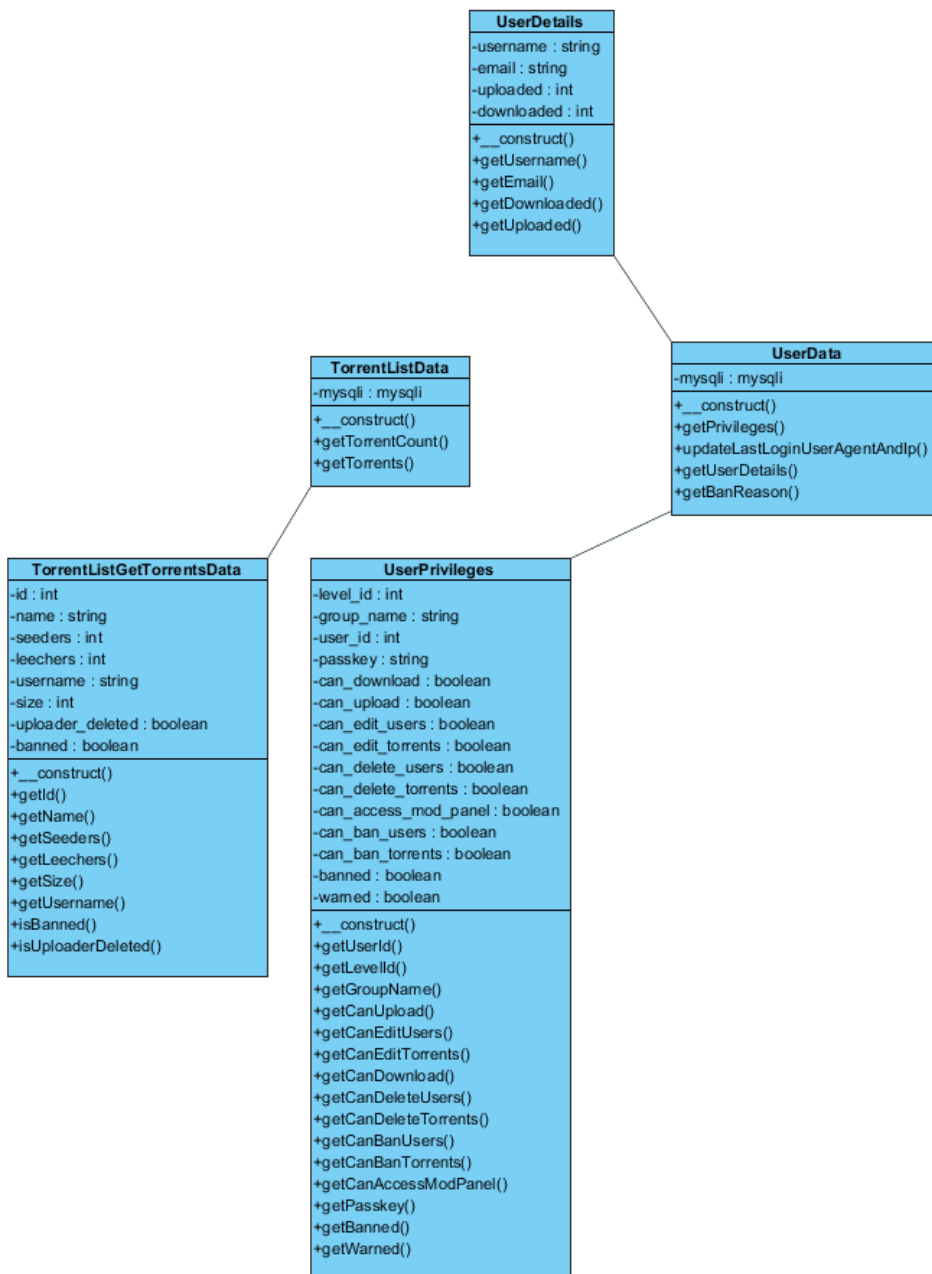
Obrázek 16 – Class diagram tříd modelu 2, zdroj: vlastní

Následující diagram popisuje třídy modelu detailů torrentu a formuláře pro nahrání nového torrentu.



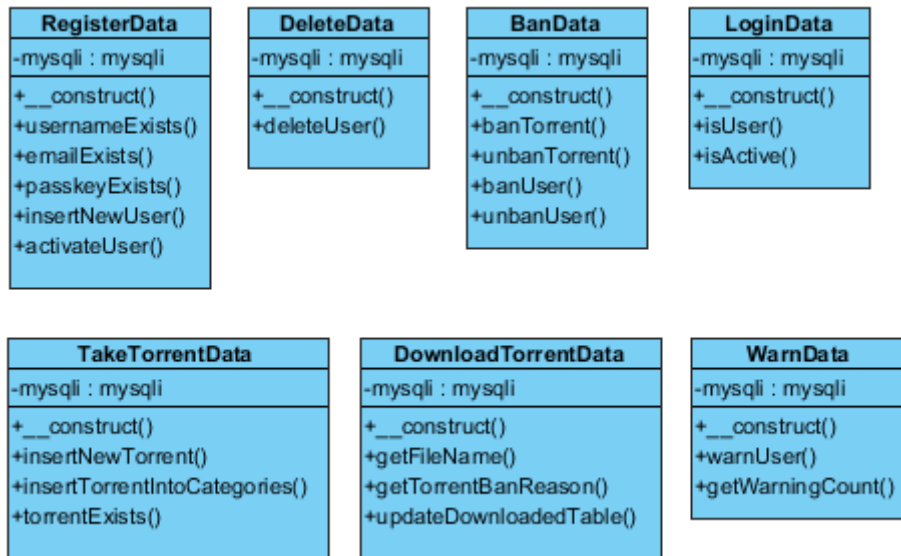
Obrázek 17 – Class diagram tříd modelu 3, zdroj: vlastní

Následující diagram popisuje třídy modelu seznamu torrentů a systému uživatelských práv.



Obrázek 18 – Class diagram tříd modelu 4, zdroj: vlastní

Následující diagram popisuje zbývající třídy modelu týkající se registrace, mazání, zakazování, přihlášení, nahrání a stažení torrentu a varování.



Obrázek 19 – Class diagram tříd modelu 5, zdroj: vlastní

## 6 Testování

Testování je důležitou součástí vývoje každého systému. Zde uvedu pouze některé ze způsobů testování, které jsem při vývoji aplikace použil.

### 6.1 Testování kódu systému

Při tvorbě kódu části aplikace jsem prováděl testování správnosti kódu [13][14]. Toto testování se nechá rozdělit na takzvané black-box testování (funkcionální) a white-box testování (strukturní).

V rámci strukturního testování jsem testoval funkce a metody tříd za účelem odhalení programátorských chyb v kódu. V této části mi velmi pomohla schopnost logování chyb PHP do externího souboru. V rámci funkcionálního testování jsem pak testoval, zda dané části aplikace reagují na požadované vstupy správně.

Asi nejdůležitější částí testování se ukázalo testování systému jako celku uživateli. Při tomto typu testování bylo odhaleno množství chyb, mezi nejdůležitějšími například dvojnásobné počítání velikosti odeslaných a přijatých dat uživatele nebo v některých případech přiřazení uživatele ke špatnému torrentu.

## 7 Problémy při vytváření aplikace

V této části bych se chtěl zmínit o několika problémech, se kterými jsem se při vytváření aplikace setkal.

### 7.1 Foreach a DOMNodelist

V některých případech procházení objektem třídy `DOMNodelist` pomocí konstrukturu `foreach` neprobíhalo korektně a přeskakovalo zhruba polovinu potomků. Odhalení této chyby mi trvalo několik dní a dle mých pozorování k ní docházelo víceméně náhodně. Zajímavostí je, že při použití `foreach` se starším rozšířením DOM XML z PHP verze 4, `foreach` fungoval bez jakýchkoli problémů. Po odhalení tohoto problému bylo řešení jednoduché a to nahrazení `foreach` cyklem `for`.

### 7.2 DOMNode a metoda appendChild

V některých případech metoda `appendChild` nepřipojila určeného potomka konkrétní `DOMNode`. Tento případ nastával víceméně náhodně, ale pouze u připojovaných potomků, kteří již měli nějakého rodiče. K odstranění této chyby stačilo jako parametr metody `appendChild` vkládat klonovaného potomka (výstup metody `cloneNode`).

### 7.3 MySQLi\_STMT, metoda bind\_param a velké číslo

Metoda `bind_param` třídy `MySQLi_STMT` umožňuje svázat parametr dotazu nad databází s jeho hodnotou. V případě nastavení parametru jako číslo je ale maximální povolená velikost tohoto čísla omezena na  $2^{32}-1$ . V případě použití datového typu `UNSIGNED INT` nebo `BIGINT` už ale tato hranice nemusí být dostačující a proto nemusí docházet k vložení správného čísla do databáze. Jednoduchým, avšak ne příliš optimálním řešením tohoto problému bylo svázání aktuálního číselného parametru jako řetězce, v tomto případě vše již proběhne v pořádku.

### 7.4 MySQL a spojování řetězců v dotazu

V případě použití podmínky `IF` v části SQL dotazu pro výběr sloupců a současném použití funkce `CONCAT` pro spojení řetězců v této podmínce docházelo k návratu špatně interpretovaných hodnot z databáze (nejspíše ve špatné znakové sadě). Řešením tohoto problému bylo přenesení podmínky a spojení řetězců do PHP části aplikace.

## 8 Závěr

Tato práce měla za cíl návrh a vytvoření systému umožňujícího správu a sdílení torrent souborů a současně disponujícím funkcí BitTorrent trackeru. Pro vývoj aplikace jsem využil kombinaci technologií Apache HTTP Server, PHP, MySQL, XHTML. Návrh vnitřní struktury probíhal v souladu s návrhovým vzorem model-view-controller. Po vytvoření systému došlo k testování, ve kterém se podařilo úspěšně odhalit a odstranit některé drobné chyby, které vznikly při implementaci aplikace.

V současné době je aplikace stále ještě v testovacím provozu, který slouží nejen k testování systému a odhalení případných nedostatků, ale také jako možnost seznámení uživatelů s aplikací, získání zpětné vazby a návrhů pro nové funkce. Výsledná aplikace je plně funkční a splňuje cíle předsevzaté v úvodu práce. Před nasazením do normálního provozu by do aplikace bylo vhodné doplnit možnost komunikace mezi uživateli prostřednictvím zpráv, umožnit personalizaci profilů uživatelů a přidat možnost komunikace více uživatelů, například zavedením diskusního fóra.

Jako přínos této práce bych označil relativně jednoduchou možnost zřízení soukromého trackeru pro člověka se základními znalostmi PHP a MySQL. Za hlavní přínos ale osobně považuji budoucí nasazení aplikace jako prostředku ke sdílení dat pro komunitu čítající zhruba 2000 uživatelů.

Osobním přínosem této práce pro mě byla možnost projít vývojovým cyklem většího projektu od jeho počátku až po nasazení do testovacího provozu. Dále jsem se seznámil s velkým množstvím nových technologií, např. DOM, XPath, MySQLi, phpDocumentor a návrhových vzorů jako MVC nebo Singleton.



## 9 Zdroje

- [1] *Peer-to-peer* In Wikipedia: the free encyclopedia [online]. Poslední revize: 2010-04-23 [cit. 2010-04-23]. Anglická verze. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Peer-to-peer>>.
- [2] **ERNESTO**, *BitTorrent Still King of P2P Traffic* In TorrentFreak [online]. 2009-02-18 [cit. 2010-04-23]. Dostupné z WWW: <<http://torrentfreak.com/bittorrent-still-king-of-p2p-traffic-090218/>>. Ověřeno z externích zdrojů.
- [3] *BitTorrentSpecification* In TheoryOrg [online]. Poslední revize: 2010-01-01 [cit. 2010-04-23]. Dostupné z WWW: <<http://wiki.theory.org/BitTorrentSpecification>>.
- [4] **COHEN Bram**, *The BitTorrent Protocol Specification* [online]. Poslední revize: 2008-02-28 [cit. 2010-04-23]. Dostupné z WWW: <[http://bittorrent.org/beps/bep\\_0003.html](http://bittorrent.org/beps/bep_0003.html)>.
- [5] *PHP* [online]. Preface – Manual: The PHP Group. Datum vydání neuvedeno [cit. 2010-04-23]. Dostupné z WWW: <<http://cz.php.net/manual/en/preface.php>>.
- [6] *PHP* [online]. License Information: The PHP Group. Datum vydání neuvedeno [cit. 2010-04-23]. Dostupné z WWW: <<http://php.net/license/>>.
- [7] *PHP* [online]. What can PHP do? – Manual: The PHP Group. Datum vydání neuvedeno [cit. 2010-04-23]. Dostupné z WWW: <<http://cz.php.net/manual/en/intro-whatcando.php>>.
- [8] *MySQL :: About MySQL* [online]. Oracle Corporation. Datum vydání neuvedeno [cit. 2010-04-23]. Dostupné z WWW: <<http://www.mysql.com/about/>>.
- [9] *MySQL :: Download MySQL Community Server* [online]. Oracle Corporation. Datum vydání neuvedeno [cit. 2010-04-23]. Dostupné z WWW: <<http://www.mysql.com/downloads/mysql/>>.
- [10] *MySQL :: The MyISAM Storage Engine* [online]. Oracle Corporation. Datum vydání neuvedeno [cit. 2010-04-23]. Dostupné z WWW: <<http://dev.mysql.com/doc/refman/5.1/en/myisam-storage-engine.html>>.
- [11] *MySQL :: Comparing Transaction and Nontransaction Engines* [online]. Oracle Corporation. Datum vydání neuvedeno [cit. 2010-04-23]. Dostupné z WWW: <<http://dev.mysql.com/doc/refman/5.1/en/storage-engine-compare-transactions.html>>.
- [12] *Model-view-controller* In Wikipedia: the free encyclopedia [online]. Poslední revize: 2010-04-23 [cit. 2010-04-23]. Anglická verze. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Model-view-controller>>. Ověřeno z externích zdrojů.

- [13] *Black-box testing* In Wikipedia: the free encyclopedia [online]. Poslední revize: 2010-04-23 [cit. 2010-04-23]. Anglická verze. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Black\\_box\\_testing](http://en.wikipedia.org/wiki/Black_box_testing)>. Ověřeno z externích zdrojů.
- [14] *White-box testing* In Wikipedia: the free encyclopedia [online]. Poslední revize: 2010-04-15 [cit. 2010-04-23]. Anglická verze. Dostupné z WWW: <[http://en.wikipedia.org/wiki/White\\_box\\_testing](http://en.wikipedia.org/wiki/White_box_testing)>. Ověřeno z externích zdrojů.

## Příloha

### A Instalační příručka

Instalační soubory aplikace se nacházejí na přiloženém CD v adresáři `install`. Pro korektní instalaci a provoz aplikace je nutné, aby na cílovém serveru byly k dispozici následující komponenty:

- Databázový systém MySQL ve verzi 5.0 a novější (testováno na verzi 5.0.51a)<sup>12</sup>
- Webový server podporující `.htaccess` a `mod_rewrite` (testováno na Apache HTTP Server 2.2.8)<sup>13</sup>
- PHP ve verzi 5.2 a novější (testováno na verzi 5.2.4)<sup>14</sup>
- PECL rozšíření BBCode (testováno s verzí 1.0.2)<sup>15</sup>
- Unixový systém obsahující Cron a Bourne Shell (testováno na Ubuntu 5.10<sup>16</sup>)

V případě nedostupnosti některých komponent je nutná jejich instalace, to je však nad rámec této příručky.

Instalační soubory a pokyny pro instalaci všech zmíněných aplikací lze nalézt na domovských stránkách těchto produktů.

#### A.1 Postup instalace aplikace

1. Kopie aplikace z adresáře `install/src` z disku CD, nastavení přístupových práv souborů a adresářů.
2. Vytvoření databáze a import souboru `install/sql/db.sql`
3. Konfigurace aplikace

Po provedení předchozích kroků by systém měl být již správně nakonfigurován a připraven k použití.

Zálohování systému probíhá formou standardní zálohy obsahu databáze (například exportem údajů do formátu SQL nebo kopií souborů databáze) a zálohou adresáře `torrents`.

#### A.2 Kopie aplikace z CD

Do adresáře, který byl určen pro umístění souborů informačního systému, nakopírujeme z adresáře `install/src`, nalézajícím se na přiloženém CD, adresáře `backup`, `error_log`, `images`, `include`, `torrents` a dále všechny soubory. Pro adresář `torrents` je nutné zajistit webovému serveru práva čtení a zápisu (například příkaz `chmod o+rw torrents`). Změnu přístupových práv je také nutné provést pro

---

<sup>12</sup> <http://www.mysql.com/>

<sup>13</sup> <http://httpd.apache.org/>

<sup>14</sup> <http://www.php.net/>

<sup>15</sup> <http://pecl.php.net/package/bbcode>

<sup>16</sup> <http://www.ubuntu.com/>

soubor `hack_log.txt` (`chmod o+w-r hack_log.txt`), soubor `error_log` v adresáři `error_log` (`chmod o+w-r error_log/error_log`) a pro adresář `backup` (`chmod +w backup`).

### A.3 Vytvoření databáze

V prostředí MySQL je nutné vytvořit databázi a do ní importovat soubor `db.sql` nacházející se v adresáři `install/sql` na CD.

### A.4 Konfigurace systému

Konfigurace připojení k databázi se provádí v souboru `Connector.php` v adresáři `include/classes`. V tomto souboru je nutno zadat adresu hosta, jméno a heslo uživatele databáze a jméno databáze. Nastavení vlastností systému se nachází v souboru `constants.php` v adresáři `include`. Tento soubor obsahuje následující konstanty:

- `_ANNOUNCE` udává URL adresu `announce` skriptu aplikace; chybné nastavení této hodnoty způsobí nefunkčnost torrent trackeru aplikace.
- `_ANNOUNCE_INTERVAL` ovlivňuje délku intervalu v sekundách, při kterém má docházet k periodickému hlášení klientů na tracker.
- `_DOWNLOAD_RATIO` udává hodnotu poměru celkového „uploadu“ k celkovému „downloadu“, pod kterou je uživateli znemožněno další stahování „torrentů“. Hodnota 0 vypíná tuto funkci systému.
- `_DOWNLOAD_ALLOWANCE` udává počet B, které uživatel může stáhnout bez ohledu na `_DOWNLOAD_RATIO`.
- `_WARNINGS_FOR_BAN` udává, při jakém počtu varování má být uživateli udělen automatický „ban“. Hodnota `false` vypíná tuto funkci systému.

V adresáři `install/cron` na disku CD se nachází 3 skripty určené pro pravidelné spuštění cronem. Skripty `torrentBackupDaily` a `torrentCleanupDaily` jsou určeny pro spuštění každých 24 hodin a skript `torrentCleanupHourly` pro spuštění každou hodinu. V souborech `torrentCleanupDaily` a `torrentCleanupHourly` je nutné nahradit hodnoty `DatabaseUser`, `DatabasePassword`, `NameOfDatabase` za jméno, heslo uživatele databáze a jméno databáze. V souboru `torrentBackupDaily` je nutné nastavit tyto proměnné:

- `MYPATH` obsahuje absolutní cestu k programu `mysqldump`
- `BCKPATH` obsahuje absolutní cestu do adresáře aplikace `backup`
- `DBNAME` obsahuje jméno databáze
- `APPATH` obsahuje absolutní cestu k aplikaci
- `DBUSR` obsahuje jméno uživatele databáze
- `DBPASS` obsahuje heslo uživatele databáze

## B Uživatelská příručka

### B.1 Úvod

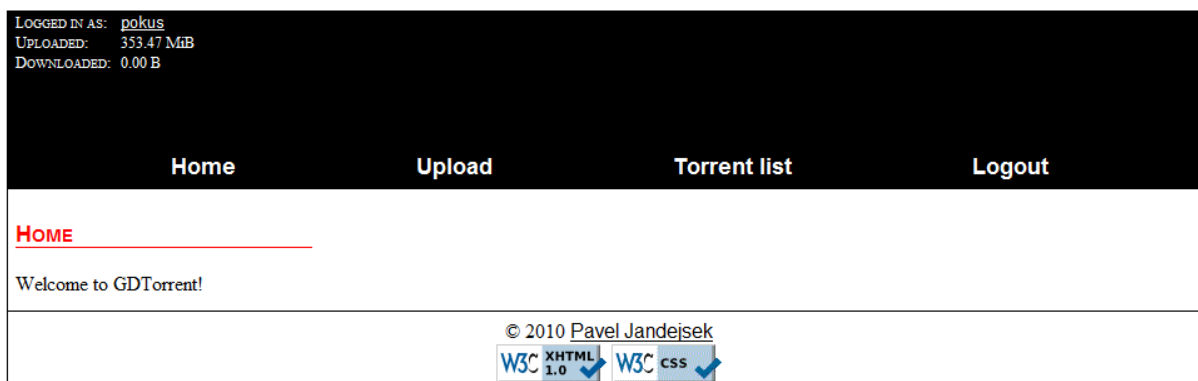
Tento text je koncipován jako základní uživatelská příručka pro aplikaci BitTorrent trackeru. V jednotlivých částech příručky je popsáno použití základních funkcí systému.

Navržená aplikace umožňuje nahrávání, správu a stahování torrentů, které uživatelé nahrají. Systém také obsahuje data o uživateli, kteří s ním pracují, společně s jejich základní správou.

Systém je navržen jako webová aplikace, ke které se uživatelé připojují z osobního počítače prostřednictvím webového prohlížeče a torrentového klienta. Doporučenými webovými prohlížeči jsou Microsoft Internet Explorer verze 8 a vyšší nebo Mozilla Firefox 3.6 a vyšší, pro něž byla aplikace optimalizována. Také je doporučeno mít zapnutou podporu JavaScriptu a cookies. Některé funkce aplikace nemusí být bez této podpory dostupné.

### B.2 Vzhled systému

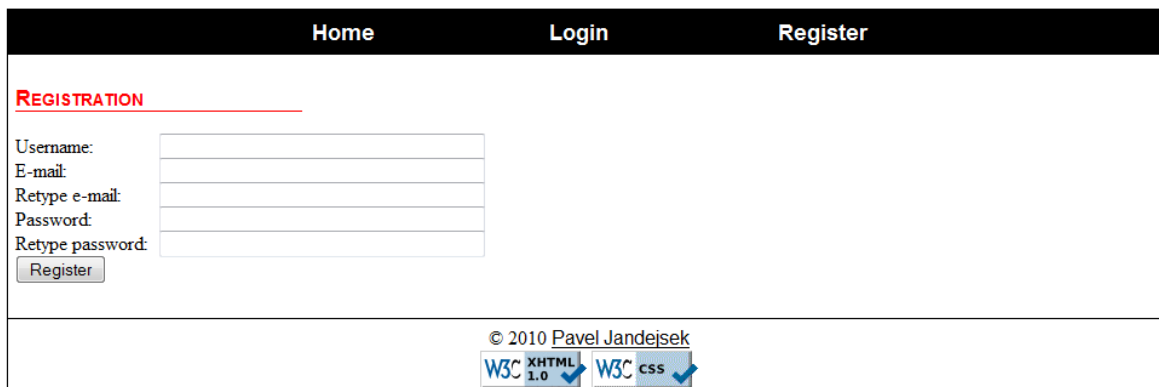
Každá stránka přístupná z webového prohlížeče má hlavičku, v její levé části se zobrazují základní informace o přihlášeném uživateli. Pod hlavičkou je umístěno hlavní menu aplikace, jehož položky vedou do jednotlivých sekcí systému. Poslední položkou hlavního menu je odkaz pro odhlášení ze systému. Pod hlavním menu se pak zobrazuje konkrétní obsah jednotlivých stránek.



Obrázek 20 – Vzhled systému, zdroj: vlastní

### B.3 Registrace uživatele

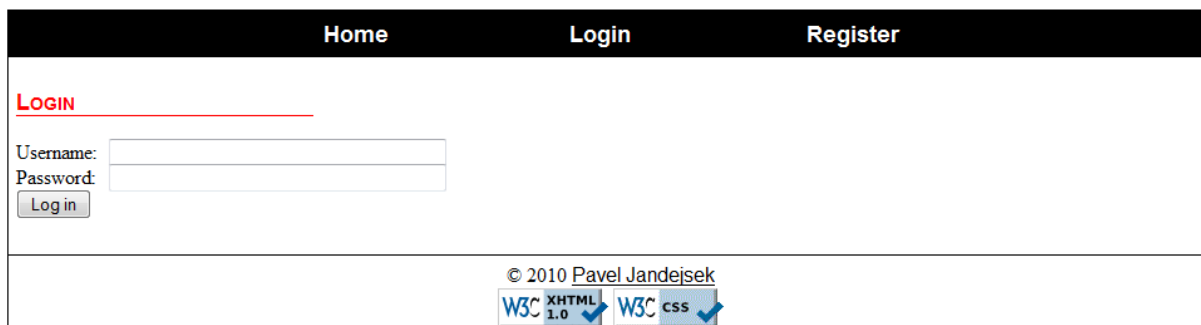
Po kliknutí na odkaz „Register“ v hlavním menu dojde k zobrazení stránky s registračním formulářem. Pro úspěšné odeslání registračních údajů je nutné vyplnit všechny políčka a nesmí existovat uživatel se stejným uživatelským jménem nebo e-mailem. Po úspěšném vyplnění registračního formuláře dojde k zaslání aktivační zprávy na adresu uvedenou při registraci. Po navštívení aktivační adresy uvedené v e-mailu dojde k aktivaci účtu a dokončení procesu registrace.



Obrázek 21 – Registrační formulář, zdroj: vlastní

### B.4 Přihlášení uživatele

Po kliknutí na odkaz „Login“ v hlavním menu dojde k zobrazení stránky s přihlašovacím dialogem. Pro přihlášení do systému je nutné zadat platné uživatelské jméno a přihlašovací heslo a potvrdit zadání stiskem tlačítka pro přihlášení. Pokud jsou přihlašovací údaje platné, zobrazí se uživateli stránka s potvrzením o platném přihlášení, v opačném případě je uživatel upozorněn na zadání neplatných údajů.



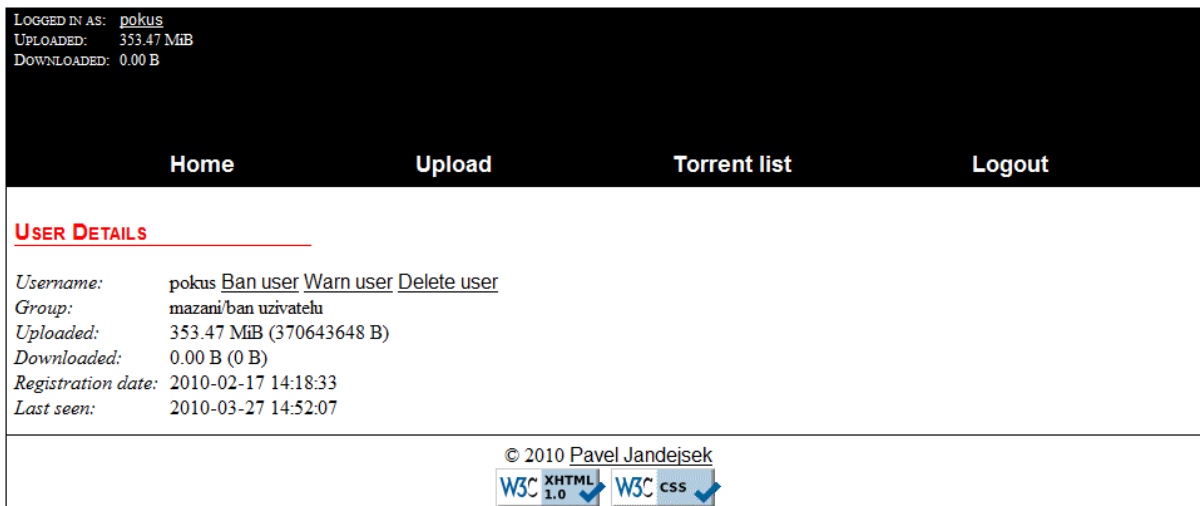
Obrázek 22 – Přihlašovací formulář, zdroj: vlastní

### B.5 Odhlášení uživatele

K odhlášení od aplikace slouží volba „Logout“ z hlavního menu aplikace.

## B.6 Uživatelské detaily

Po kliknutí na uživatelské jméno kdekoli v aplikaci (například v hlavičce) dojde k zobrazení stránky s detaily konkrétního uživatele. Tato stránka obsahuje uživatelské jméno uživatele, skupinu do které uživatel patří, datum registrace a posledního přihlášení, celkovou velikost dat které uživatel nasdílel a stáhl od ostatních uživatelů. Pokud byl uživatel vypovězen ze stránky, je zde také zobrazen důvod tohoto činu. V případě dostatečných práv přihlášeného uživatele zde mohou být i odkazy nabízející varování uživatele, „ban“ uživatele a také smazání uživatele.

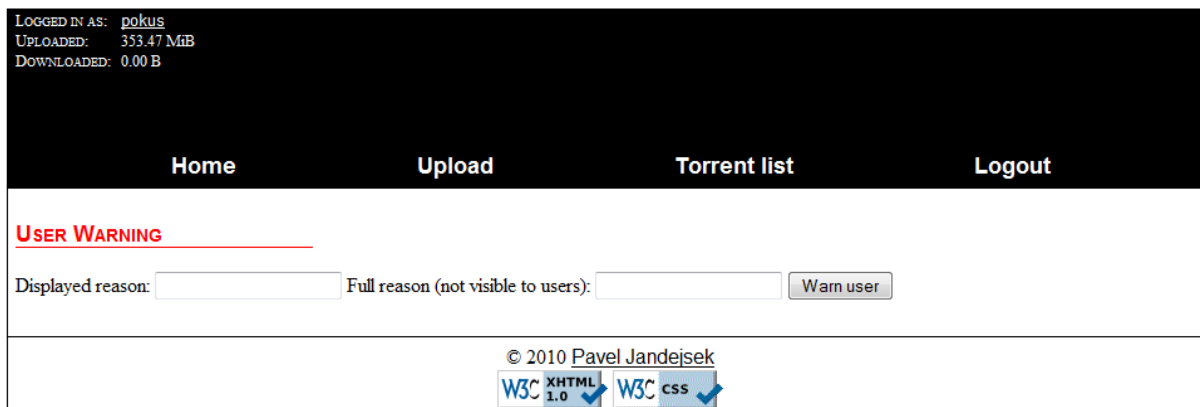


The screenshot shows the user details page for a user named 'pokus'. At the top, it displays the user's status: 'LOGGED IN AS: pokus', 'UPLOADED: 353.47 MiB', and 'DOWNLOADED: 0.00 B'. Below this is a navigation bar with links for 'Home', 'Upload', 'Torrent list', and 'Logout'. The main content area is titled 'USER DETAILS' and lists the following information: Username: pokus (with links for Ban user, Warn user, and Delete user); Group: mazani/ban uzivatelu; Uploaded: 353.47 MiB (370643648 B); Downloaded: 0.00 B (0 B); Registration date: 2010-02-17 14:18:33; Last seen: 2010-03-27 14:52:07. At the bottom, there is a copyright notice for Pavel Jandajsek and two W3C validation icons for XHTML 1.0 and CSS.

Obrázek 23 – Uživatelské detaily, zdroj: vlastní

## B.7 Varování uživatele

V případě dostatečných práv přihlášeného uživatele se tato stránka zobrazí po kliknutí na „Warn User“. Zde je nutno vyplnit políčko s veřejným důvodem varování a důvodem viditelným pouze pro ostatní uživatele s dostatečnými právy a odeslat formulář.

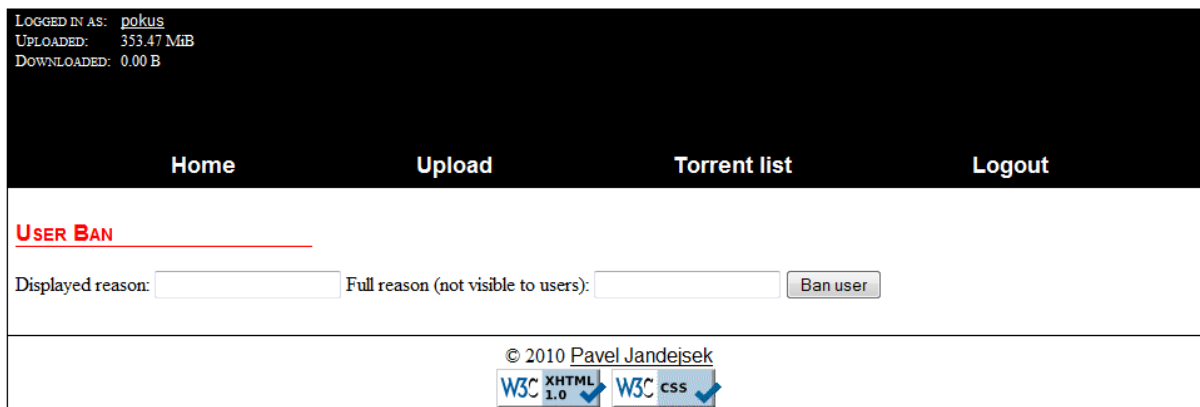


The screenshot shows the user warning form. At the top, it displays the user's status: 'LOGGED IN AS: pokus', 'UPLOADED: 353.47 MiB', and 'DOWNLOADED: 0.00 B'. Below this is a navigation bar with links for 'Home', 'Upload', 'Torrent list', and 'Logout'. The main content area is titled 'USER WARNING' and contains two input fields: 'Displayed reason:' and 'Full reason (not visible to users):'. A 'Warn user' button is located to the right of the second input field. At the bottom, there is a copyright notice for Pavel Jandajsek and two W3C validation icons for XHTML 1.0 and CSS.

Obrázek 24 – Formulář pro varování uživatele, zdroj: vlastní

## B.8 Ban uživatele

V případě dostatečných práv přihlášeného uživatele se tato stránka zobrazí po kliknutí na „Ban User“. Zde je nutno vyplnit políčko s veřejným důvodem banu a důvodem viditelným pouze pro ostatní uživatele s dostatečnými právy a odeslat formulář.



LOGGED IN AS: pokus  
UPLOADED: 353.47 MiB  
DOWNLOADED: 0.00 B

Home Upload Torrent list Logout

**USER BAN**

Displayed reason:  Full reason (not visible to users):

© 2010 Pavel Jandejsek  
W3C XHTML 1.0 W3C CSS

Obrázek 25 – Formulář pro ban uživatele, zdroj: vlastní

## B.9 Smazání uživatele

V případě dostatečných práv přihlášeného uživatele je po kliknutí na odkaz „Delete User“ daný uživatel smazán.

## B.10 Nahrání torrentu

Po kliknutí na odkaz „Upload“ v hlavním menu dojde k zobrazení formuláře pro nahrání torrentu. Pro úspěšné nahrání torrentu musí být vyplněna všechna políčka s výjimkou data. Pole „Torrent file:“ musí obsahovat validní torrent soubor. Popis torrentu může obsahovat tzv. BBCode pro formátování textu. V případě chybně vyplněného formuláře nebo chybě při zpracování souboru torrentu je uživatel o chybě informován a nahrání bude neúspěšné.



**UPLOAD**

Torrent info:

Torrent file:

Torrent name:

Description (BB Code allowed):

Date:

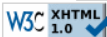

Year:

Month:

Day:

Categories:

© 2010 Pavel Jandejsek

Obrázek 26 – Formulář pro nahrání torrentu, zdroj: vlastní

## B.11 Výpis torrentů

Po kliknutí na odkaz „Torrent list“ v hlavním menu dojde k zobrazení stránky s výpisem nahraných torrentů. Výpis obsahuje název torrentu, jméno uživatele který ho nahrál, velikost souborů v torrentu, počet „seederů“ a počet „leecherů“, kteří jsou aktivní ve „swarmu“ daného torrentu. Výpis je dále možné seřadit podle názvu, velikosti, počtu „seederů“ a „leecherů“. Pomocí vyhledávacího pole lze zároveň v názvech nahraných torrentů vyhledávat; v případě zaškrtnutí políčka „Search in description:“ probíhá vyhledávání i v popisu torrentů.

## TORRENT LIST

Search text (use . as wildcard for single character and \* for more characters):

Search in description:

Torrent Name	Size	Seeders	Leechers
<u>DAPPER DRAKE</u> Uploaded by: <a href="#">Jarm</a>	502.75 MiB	0	0
<u>EDGY EFT</u> Uploaded by: <a href="#">Jarm</a>	555.20 MiB	0	0
<u>FEISTY FAWN</u> Uploaded by: <a href="#">pokus</a>	507.49 MiB	0	0
<u>GUTSY GIBBON</u> Uploaded by: <a href="#">pokus</a>	564.71 MiB	0	0
<u>HARDY HERON</u> Uploaded by: <a href="#">pokus</a>	526.57 MiB	0	0
<u>BREEZY BADGER</u> Uploaded by: <a href="#">machettaman</a>	526.57 MiB	0	0
<u>UBUNTU INTREPID IBEX</u> Uploaded by: <a href="#">machettaman</a>	621.93 MiB	0	0
<u>UBUNTU JAUNTY JACKALOPE</u> Uploaded by: <a href="#">machettaman</a>	572.34 MiB	0	0
<u>UBUNTU LUCID LYNX</u> Uploaded by: <a href="#">machettaman</a>	657.22 MiB	0	0
<u>UBUNTU KARMIC KOALA</u> Uploaded by: <a href="#">machettaman</a>	600.00 MiB	0	0

[1](#) [2](#) [3](#)

© 2010 Pavel Jandejsek



Obrázek 27 – Vypis seznamu torrentů, zdroj: vlastní

## B.12 Detaily torrentu

Po kliknutí na název torrentu na stránce s výpisem torrentů dojde k zobrazení jeho detailů. Kromě detailů o konkrétním torrentu, tato stránka obsahuje také seznam „peerů“ a komentáře torrentu. Po kliknutí na odkaz „Download Torrent“ je možné stáhnout konkrétní torrent soubor. Pokud torrent obdržel „ban“, je zde také zobrazen důvod tohoto činu, možnost stáhnutí není dostupná. V případě dostatečných práv přihlášeného uživatele je k dispozici také možnost „banu“ nebo zrušení „banu“ torrentu.

**TORRENT DETAILS**  
**WEEZER - BLUE ALBUM (1994)**

[Download Torrent](#)

*Uploader:* pokus

*Description:*

1. My Name Is Jonas
2. No One Else
3. The World Has Turned and Left Me Here
4. Buddy Holly
5. Undone - The Sweater
6. Surf Wax America
7. Say It Ain't So
8. In the Garage
9. Holiday
10. Only in Dreams

*Size:* 94.46 MiB (99051364 B)

*Number of files:* 14

*Added:* 2010-03-07 22:52:35

*Last Seeder:* 2010-03-10 14:25:46

*Downloaded:* 5x times

*Completed:* 1x times

*Seeders:* 0

*Leechers:* 0

*Info hash:* 5e8128e1f96476ce50d7f05c483f5275bd162391

*MD5 Sum:* 40f2c88c31d7a1b3ddcf1499085afdb2

*Categories:* prvni

*Completed users:*

[Jarm](#)

Peers

No one is seeding or leeching the torrent.

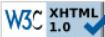

Comments

[pokus](#) 2010-03-23 14:45:53

Torrent does not contain the mentioned copyrighted files, only some random testing ones.

Add New Comment:

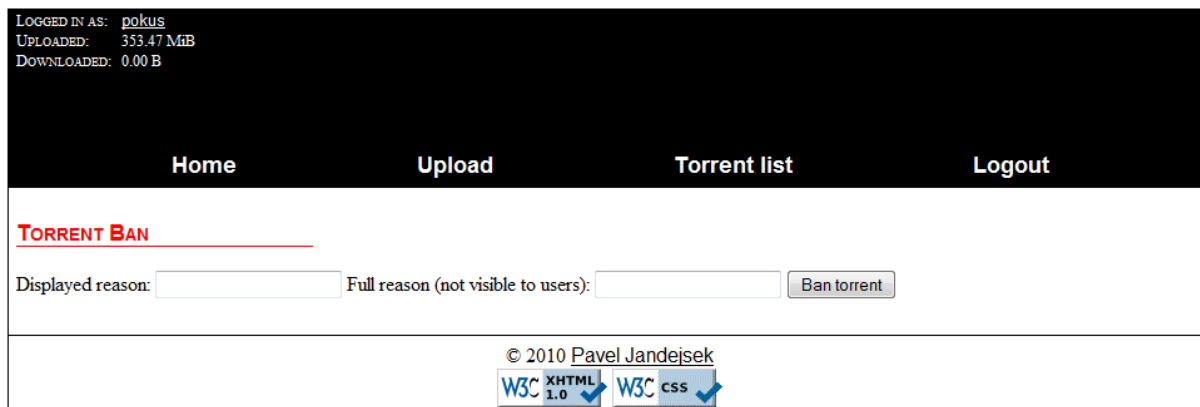
© 2010 Pavel Jandejsek

Obrázek 28 – Detaily torrentu, zdroj: vlastní

## B.13 Ban torrentu

V případě dostatečných práv přihlášeného uživatele se tato stránka zobrazí po kliknutí na „Ban Torrent“. Zde je nutno vyplnit políčko s veřejným důvodem banu a důvodem viditelným pouze pro ostatní uživatele s dostatečnými právy a odeslat formulář.



LOGGED IN AS: pokus  
UPLOADED: 353.47 MiB  
DOWNLOADED: 0.00 B

Home Upload Torrent list Logout

**TORRENT BAN**

Displayed reason:  Full reason (not visible to users):

© 2010 Pavel Jandejsek  
W3C XHTML 1.0 W3C CSS

Obrázek 29 – Formulář pro ban torrentu, zdroj: vlastní

## B.14 Zrušení banu torrentu

V případě dostatečných práv přihlášeného uživatele je po kliknutí na odkaz „Unban Torrent“ je u daného torrentu „ban“ zrušen.

## C Dokumentace kódu aplikace

Dokumentace kódu aplikace se nachází v adresáři `docs/doxy`. Pro její vygenerování bylo použito aplikace Doxygen verze 1.6.2.<sup>17</sup>

---

<sup>17</sup> Více informací na <http://www.doxygen.org/>

## D Obsah CD

[docs]	adresář s dokumentací aplikace
_ [doxy]	adresář s programátorskou dokumentací
_ index.html	hlavní stránka programátorské dokumentace
_ install_manual.pdf	instalační příručka
_ user_manual.pdf	uživatelská příručka
_ JandejsekP_NavrhImplementace_JF_2010.pdf	soubor s bakalářskou prací
[install]	adresář pro instalaci aplikace
_ [cron]	adresář se skripty pro cron
_ torrentBackupDaily	zálohovací skript
_ torrentCleanupDaily	skript na promazání databáze jednou za den
_ torrentCleanupHourly	skript na promazání databáze jednou za hodinu
_ [sql]	adresář obsahující skript pro vytvoření databáze
_ db.sql	soubor s SQL příkazy pro vytvoření databáze
_ [src]	adresář obsahující kód aplikace
_ [backup]	adresář pro zálohy aplikace
_ [error_log]	adresář pro log chyb
_ error_log	soubor s logem chyb
_ [images]	adresář s obrázky
_ [site]	adresář s obrázky aplikace
_ *.png	soubory s obrázky aplikace
_ [include]	adresář se zdrojovými kódy
_ [classes]	adresář s pomocnými třídami
_ *.php	soubory se zdrojovými kódy
_ [data]	adresář se zdrojovými kódy modelu
_ *.php	soubory se zdrojovými kódy modelu
_ [view]	adresář se zdrojovými kódy pohledu
_ *.php	soubory se zdrojovými kódy pohledu
_ *.php	soubory se zdrojovými kódy
_ [torrents]	adresář s .torrent soubory
_ .htaccess	soubor s dodatečnou konfigurací web serveru
_ favicon.ico	soubor s ikonou aplikace
_ hacklog.txt	soubor s logem podezřelých operací
_ style.css	soubor s kaskádními styly
_ *.php	soubory se zdrojovými kódy
cd_contents.txt	soubor s obsahem CD