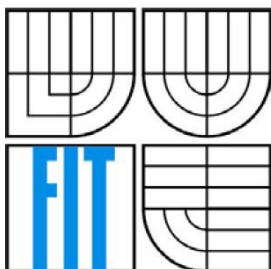


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM PRO HRU WORLD OF WARCRAFT

INFORMATION SYSTEM FOR THE GAME
WORLD OF WARCRAFT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTINA KLEMENTOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ROMAN LUKÁŠ , Ph.D.

BRNO 2009

Zadání bakalářské práce

Řešitel: **Klementová Martina**

Obor: Informační technologie

Téma: **Informační systém pro hru World of Warcraft**

Kategorie: Web

Pokyny:

1. Seznamte se s jazyky a prostředky pro tvorbu webových informačních systémů (HTML, CSS, PHP, Javascript, MySQL).
2. Proveďte detailní návrh systému, který bude umožňovat správu databáze on-line hry World of Warcraft a modifikaci jednotlivých objektů. Systém dále bude umožňovat správu jednotlivých postav.
3. Navrhněte způsob a možnosti automatizovaného omezení přístupu uživatelů za používání vulgarismů v rámci chatu.
4. Proveďte implementaci navrženého systému včetně skriptu pro zamezování přístupu jistým uživatelům. Dále prostudujte příbuzné webové stránky hry World of Warcraft a implementaci vašeho systému rozšiřte o načítání různých informací z ostatních systémů a ukládání těchto aktuálních informací do vlastní databáze. Ověřte funkčnost vaší implementace.
5. Zhodnoťte dosažené výsledky a navrhněte další možné rozšíření do budoucna.

Literatura:

- Kosek, J.: PHP - Tvorba interaktivních internetových aplikací, Grada Publishing, 1999.

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

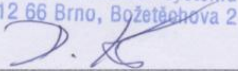
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Lukáš Roman, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2008

Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2


doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Bakalářská práce se zabývá vývojem informačního systému pro hru World of Warcraft. Tento systém je určen pro lidi, zabývající se správou a vývojem serveru, umožňujícího bezplatné hraní této hry. Práce je zaměřena především na rozšíření samotného systému o možnost získání dat z příbuzných webových stránek a jejich zpracování a uložení do vlastní databáze. Rovněž umožňuje použití nástroje pro vyhledání vulgarismů, který prohledává různé konverzace mezi hráči a při výskytu nevhodného výrazu dočasně omezí hráči přístup do hry. K implementaci systému byly použity jazyky HTML, CSS, PHP, Javascript, AJAX a MySQL.

Klíčová slova

informační systém, World of Warcraft, server, HTML, CSS, PHP, Javascript, AJAX a MySQL

Abstract

This Bachelor work deals with the development of information system for the game World of Warcraft. This system is designed for the people dealing with administration and development of the server that allows the player to play this game free of charge. The work is mostly designed as an extension of the existing system, with the possibility, to extract the required data, from other similar web sites and their processing and saving into our game server database. It also allows usage of a tool analysing presence of a vulgar expressions in communication channels, used by the players and bans the violator temporarily from playing the game upon finding a word contained in a dictionary of vulgarisms. The following programming languages were used for the system implementation: HTML, CSS, PHP, Javascript, AJAX and MySQL.

Keywords

information system, World of Warcraft, server, HTML, CSS, PHP, Javascript, AJAX a MySQL

Citace

Martina Klementová: Informační systém pro hru World of Warcraft, bakalářská práce, Brno, FIT VUT , v Brně, 2009

Informační systém pro hru World of Warcraft

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Romana Lukáše, Ph.D.

.....
Martina Klementová

20. května 2009

Poděkování

Chtěla bych poděkovat Ing. Romanovi Lukášovi Ph.D. za ochotu a podporu při tvorbě této práce a lidem kteří mi pomohli při vývoji a testování této práce.

© Martina Klementová, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Úvod	3
1 Technologie pro vývoj informačních systémů.....	4
1.1 Jazyk PHP	4
1.2 Značkovací jazyky HTML, XHTML a kaskádové styly	5
1.3 Značkovací jazyk XML	6
1.4 JavaScript	6
1.5 MySQL	6
2 World of Warcraft	8
2.1 Princip hry	8
2.2 Rasy a povolání	9
3 Quice	10
3.1 Možnosti práce s Quice	10
4 Návrh systému.....	11
4.1 Požadavky na systém.....	11
4.2 Rozdělení systému na jednotlivé části.....	11
4.2.1 Sekce server	12
4.2.2 Sekce eventy	12
4.2.3 Sekce tickety.....	12
4.2.4 Sekce administrator	12
4.2.5 Sekce hledat.....	13
4.2.6 Sekce creatury	13
4.2.7 Sekce gameobjecty	14
4.2.8 Sekce itemy	14
4.2.9 Sekce npc	14
4.2.10 Sekce questy	15
4.2.11 Sekce scripty.....	15
4.2.12 Postavy	15
5 Konceptuální datový model	17
5.1 Jazyk UML	17
5.1.1 Use-case diagram.....	17
5.2 Návrh Use-case diagramu.....	18
6 Implementace systému	20
6.1 Použité technologie	20
6.2 Grafické rozhraní systému.....	20
6.3 Implementace jednotlivých sekcí v systému	21
6.4 Správa serveru prostřednictvím SQL dotazů.....	21
6.4.1 Generování SQL dotazů	21
6.4.2 Automatické generování updatů.....	21
6.4.3 Automatické generování insertů a deletů	22
6.5 Implementace načítání informací z jiných webových stránek a ukládání těchto informací do databáze.....	23

6.5.1	Získávání kořisti z webové stránky	23
6.5.2	Získávání prodávaných předmětů z webové stránky	23
6.6	Používání vulgarismů ve hře	24
6.6.1	Možné způsoby obcházení cenzury	24
6.6.2	Implementace algoritmu pro vyhledání vulgarismů	25
6.6.3	Tvorba slovníku sprostých výrazů	25
6.6.4	Zpracování nalezených vulgarismů	25
7	Testování a provoz systému	26
7.1	Omezení posílaných informací	26
7.2	Optimalizace JavaScriptu při získávání dat z wowheadu	26
7.3	Instalace systému	26
8	Porovnání s existujícími systémy a možnosti rozvoje	27
8.1	Porovnání s Quicem	27
8.2	Možnosti rozvoje	27
9	Závěr	29

Úvod

Lidé jsou od narození hravé bytosti a proto většina z nás má zřejmě potřebu rozvíjet svůj intelekt, vnímání a schopnost soustředění prostřednictvím her. V dřívějších dobách, kdy ještě neexistovaly počítače, lidé hledali zábavu v různých deskových či karetních hrách, kde se kromě samotné zábavy rozvíjeli i po intelektuální stránce.

V dnešní době většinu karetních a deskových her nahradily hry počítačové, které nabízí široké možnosti. Kromě klasických akčních počítačových her jsou samozřejmě k dispozici různé adventury, arkády, strategické hry, různé simulátory a hry na hrdiny. Jednou z nejznámějších her na hrdiny je hra World of Warcraft. Jedná se o onlinovou hru, kterou je možné hrát pouze prostřednictvím placeného serveru. Z tohoto důvodu se začaly skupiny lidí zabývat vývojem serveru vlastního, který bude k dispozici pro všechny zájemce. Většina serverů je vyvíjena v jazyce C++ a databáze k serveru je uložena pomocí služby MySQL.

Nedokonalost serverů a hlavně chybějící nebo špatné záznamy v databázi mě přivedly na myšlenku vytvořit systém, který bude umožňovat správu databáze všem zájemcům bez ohledu na jejich znalosti.

Použité technologie při tvorbě systému jsou popsány v **kapitole první**, kde jsou postupně charakterizovány jednotlivé použité technologie PHP, HTML/XHTML, kaskádové styly, XML, JavaScript a MySQL.

Popis hry World of Warcraft je uveden v **kapitole druhé**. Popisuje základní principy hry, jednotlivé rasy a povolání, které jsou dostupné.

Ve třetí kapitole je popsán software Quice, který mě inspiroval při návrhu systému. Kapitola se zabývá popisem tohoto softwaru, zvýrazněním nedostatků a nastíněním jeho možností.

Čtvrtá kapitola se zabývá popisem návrhu systému. V rámci popisu návrhu jsou uvedeny požadavky na systém, které převážně vyplývají z popisu aplikace Quice. Na základě analýzy je dále popsáno rozdělení systému na jednotlivé sekce a podsekce.

Pomocí nástrojů pro konceptuální modelování UML je v **kapitole páté** navrhována a namodelována struktura pro systém. Z důvodů již navržené databáze je zde uveden pouze use-case diagram, který modeluje vztahy mezi systémem a jeho účastníky.

Šestá kapitola se podrobně zabývá implementací systému. Podrobně jsou zde popsány nestandardní funkce, jako je automatické generování SQL dotazů, načítání informací z jiných webových stránek a algoritmus, který automaticky vyhledává vulgarismy v diskuzích mezi hráči.

Testování a provoz systému je diskutován v **kapitole sedmé**. Jsou zde nastíněny problémy, které nastaly při uvedení systému do reálného provozu, včetně realizace jejich řešení. Dále se zde můžeme seznámit s postupem instalace systému na server a jeho uvedením do provozu.

V **osmé kapitole** nalezneme srovnání s již existujícím systémem Quice a možnosti dalšího rozvoje systému.

Závěrečná kapitola obsahuje zhodnocení dosažených výsledků a přínos pro správu serverů pro hru World of Warcraft.

Kapitola 1

Technologie pro vývoj informačních systémů

Jako webová aplikace se obvykle označuje skript (kód zajišťující funkci programu) běžící na straně serveru. Často bývá propojen s některou z databází, nejčastěji MySQL. Výstupem skriptu je potom samotná webová stránka, která je předána k zobrazení prohlížeči.

Na obrázku 1.1 je znázorněno schéma webové aplikace, kde skript je program nebo jeho fragment zapsaný v podobě kódu, např. skript psaný v jazyce PHP, který se nepřenáší do prohlížeče uživatele. Základem každé WWW stránky je jazyk HTML (nebo jeho modernější verze XHTML), používaný k tvorbě kostry stránky. Následně se přidávají kaskádové styly, které umožňují ovlivnit celkový vzhled internetové stránky. Posledním důležitým jazykem je JavaScript, který se používá k složitějším dynamickým efektům, které nevyžadují znovunačtení stránky.



Obrázek 1.1: schéma webové aplikace

1.1 Jazyk PHP

PHP je nástupcem staršího produktu, nazvaného PHP/FI. PHP/FI vytvořil Rasmus Lerdorf v roce 1995, na počátku jako jednoduchou sadu skriptů v jazyce Perl pro zpracování záznamů o přístupech k jeho webu. Tuto sadu nazval 'Personal Home Page Tools'. Protože byla třeba větší funkčnost, napsal Rasmus mnohem rozsáhlejší implementaci v jazyce C, která byla schopna komunikovat s databázemi a umožňovala uživatelům vyvíjet jednoduché dynamické aplikace pro Web. PHP/FI 2.0 bylo oficiálně uvolněno až v listopadu 1997, poté co strávilo většinu svého života v betaverzích. Krátce na to bylo následováno první alfav verzí PHP 3.0.

PHP 3.0 byla první verze, která se velmi blížila takovému PHP, jak ho známe dnes. Vytvořili ho Andi Gutmans a Zeev Suraski jako kompletně přepsaný celek. Jednou z nejsilnějších stránek PHP 3.0 byly jeho obrovské možnosti rozšíření, které umožňovaly spolupráci s mnoha databázovými

systemy. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka. PHP 3.0 bylo oficiálně uvolněno v červnu 1998.

Nejnovější verzí je PHP5, které bylo vydáno v červenci 2004. Je řízeno svým jádrem, Zend Engine 2.0, s novým objektovým modelem a je doplněno o mnoho nových vlastností.

PHP je v současnosti velmi rozšířená technologie umožňující snadné programování na straně serveru (server-side programming). Toho lze využít k tvorbě různých interaktivních webových stránek. Stručně lze říci, že skript napsaný v PHP je proveden na serveru podle zadaných kritérií a výsledek je odeslán volajícímu počítači stejným způsobem, jakým se odesílají běžné statické (XHTML) stránky. Jakmile je však stránka načtena klientem, pomocí PHP ji již není možné dále měnit.

1.2 Značkovací jazyky HTML, XHTML a kaskádové styly

World Wide Web se zrodil v CERNu, Evropské laboratoři pro fyziku částic ve švýcarské Ženevě a jeho autorem je Tim Berners-Lee. Ještě před příchodem do CERNu už Tim pracoval na tvorbě dokumentů a zpracování textů a v roce 1980 rozvinul svůj první hypertextový systém "Enquire", který sloužil pro jeho osobní potřebu.

Tim předvedl jednoduchou, ovšem atraktivní cestu publikování textu vyvinutím svého vlastního softwaru a zároveň vlastním protokolem HTTP pro získávání ostatních textů z dokumentů přes hypertextové odkazy. Timův vlastní protokol, HTTP, zaujal místo pro hypertextový přenosový protokol (HyperText Transfer Protokol). Textový formát pro http byl pojmenován HTML (HyperText Mark-up Language). Tím, že zachoval věci velmi jednoduché, povzbudil Tim ostatní, aby na jeho myšlenkách postavili a navrhli budoucí software pro zobrazování HTML a zřídili své vlastní HTML dokumenty připravené pro přístup.

HTML, které Tim vytvořil, bylo přísně založeno na SGML (Standard Generalized Mark-up Language), mezinárodně schválené metodě pro formování textu do strukturálních jednotek jako jsou odstavce, nadpisy, položky seznamu a tak dále. SGML může být implementováno na jakémkoliv počítači. Hlavní myšlenkou bylo, že jazyk bude nezávislý na formátoru (prohlížeč, nebo jiný zobrazovací systém), který ve skutečnosti zobrazuje text na obrazovce. Používání párů značek (tagů) je vzato přímo z SGML, které se chová úplně stejně.

Během roku 1993 a začátkem roku 1994 si mnoho prohlížečů přidávalo do HTML své vlastní prvky, čímž se tento jazyk stal špatně definovaným. V úsilí vytvořit pořádek v nastalém chaosu Dan Connolly spolu s kolegy soustředil všechny značky HTML, které byly široce užívány a uspořádali je do pracovního dokumentu, který definoval šířku jazyka, který Tim Berners-Lee nazval HTML2.

V listopadu 1995 se uskutečnila schůzka ve Versailles nedaleko Paříže kde se diskutoval rozvoj kaskádových stylů CSS (*Cascading Style Sheets*). CSS slouží pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Jazyk byl navržen standardizační organizací W3C a autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány zatím dvě úrovně specifikace CSS1 a CSS2, dokončuje se revize CSS 2.1 a pracuje se na verzi CSS3. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu.

V lednu 1997, W3C (*World Wide Web consortium*) formálně podpořilo HTML 3.2. Bylo přezkoumáno všemi členskými organizacemi včetně hlavních prodejců prohlížečů, jako Netscape a Microsoft. To znamenalo, že specifikace byla teď stálá a odsouhlasena. HTML 3.2 vzalo existující IETF standard HTML 2 a začlenilo rysy HTML+ a HTML 3. HTML 3.2 zahrnuje tabulky, aplety, obtékání textu kolem obrázků, skripty a superskripty.

Na jaře 1998 bylo vytvořen HTML 4. V současné době je standardizována verze HTML 4.01, která je také poslední vývojovou verzí. Předpokládaným nástupcem HTML měl být jazyk XHTML založený na XML, ale s dalším vývojem HTML 5.0 je zatím XHTML paralelně vyvíjeno a stále se pracuje na verzi 2.0.

1.3 Značkovací jazyk XML

XML (eXtensible Markup Language – „rozšiřitelný značkovací jazyk“) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat.

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezence dokumentu (vzhled) se potom definuje připojeným stylem. Další možností je pomocí různých stylů provést transformaci do jiného typu dokumentu, nebo do jiné struktury XML.

1.4 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

JavaScript je syntakticky podobný jazyku C, C++ a Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe.

JavaScript byl původně obchodní název implementace společnosti Netscape, kde byl vyvíjen nejprve pod názvem Mocha, později LiveScript. Ohlášen byl společně se společností Sun Microsystems v prosinci 1995 jako doplněk k jazykům HTML a Java. Pro verzi firmy Microsoft je použit název JScript. JScript je podporován platformou .NET.

Na rozdíl od jiných programovacích jazyků, jako je PHP, JavaScript funguje na straně klienta, což znamená, že jeho interpret je zabudován v prohlížeči a nevyžaduje tudíž žádnou kompilaci. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

JavaScript je možné použít i na straně serveru. První implementací JavaScriptu na straně serveru byl LiveWire firmy Netscape vypuštěný roku 1996. Dnes existuje několik možností, např. opensource implementace Rhinola, založená na Rhino, gcj a Apache.

1.5 MySQL

MySQL (My Structured Query Language) je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark.

Komunikace s ním probíhá, jak už název napovídá, pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. Do MySQL lze ukládat různá data (texty, obrázky atd.), s nimiž lze dále jednoduše pracovat (třídít, řadit, filtrovat apod.). Nejčastěji se MySQL používá ve spojení s jazykem PHP, který umožňuje přístup k uloženým datům.

Každá databáze v MySQL obsahuje tabulky, každá tabulka má sloupce a řádky a v každém řádku jsou záznamy předem určeného typu.

Pro svou snadnou implementovatelnost (lze je instalovat na celou řadu operačních systémů), výkon a především díky tomu, že se jedná o volně šiřitelný software, je MySQL v poslední době téměř nejoblíbenějším systémem.

Kapitola 2

World of Warcraft

World of Warcraft (zkráceně WoW) je počítačová onlinová hra na hrdiny. Byla vyvinuta společností Blizzard Entertainment, což je společnost, která naprogramovala také předchozí díly celé série Warcraft. WoW byl vydán k desátému výročí série Warcraft v roce 2004. Hra drží rekord v Guinessově knize rekordů za nejpobulárnější MMORPG (Massive-Multiplayer Online Role-Playing Game) hru. Z celkových 16 milionů hráčů pravidelně hrajících nejpobulárnější MMORPG hry, jich asi 11 milionů hraje World of Warcraft.

2.1 Princip hry

Ve hře se postava hráče pohybuje v rozlehlém virtuálním světě, setkává se, baví se a bojuje s jinými hráči, ale také s tzv. nehráčskými postavami (NPC: non-player character – postava, která je ovládána umělou inteligencí počítače, nikoli hráčem). Některé z nich jsou určeny k zabíjení, jiné zadávají úkoly (quests), které jsou povětšinou likvidačního charakteru „zabij 10 těch“, ale i typu „přines toto“, „doruč zprávu tam a tam“, „doprovod' toho a toho“ a další. Pomocí úkolů je hráč postupně prováděn po celém světě, protože úkoly jsou nabízeny vždy dle aktuální úrovně (levelu) hráče a postupně ho zavádějí do dalších míst v dosud neznámých oblastech a kontinentech.

Ve hře je spousta úkolů, za něž hráč dostává zkušenostní body (tzv. XP, zkratka z angl. Experience Points), které se dají získat také zabíjením nepřátelských nestvůr. Za určitý počet zkušeností (XP) hráč získá úroveň (postupuje na vyšší level). Maximální úroveň byla ze začátku hry 60, ale s přibývajícimi datadisky se postupně zvýšila na úroveň 80. Odměnou za splnění úkolů je obvykle nejen zkušenost (XP), umožňující postup na další úroveň, ale často také peníze, hodnotné věci (zbraně, brnění apod.) nebo například získávání reputace u daného města nebo frakce.

Reputace pak umožňuje další výhody - od nižších poplatků za služby a věci u nehráčských postav až po unikátní předměty. Reputace může být jak kladná (odstupňováno: Friendly, Honored, Revered, Exalted), tak neutrální (Neutral), u nepřátelských frakcí pak v různých stupních od nepřátelský (Unfriendly), kdy není možné komunikovat s nehráčskými postavami, až po nenáviděný (Hostile, Hated), kdy je hráč okamžitě napadán.

Po získání maximální úrovně hra nekončí, ale právě začíná. Na maximální úrovni se hráči otvírají místa, kde sídlí zlo samotného světa a kde může hráč získat nejlepší unikátní předměty, které hra nabízí. To ovšem není tak jednoduché a je zapotřebí složit skupinu 25 hráčů (v jednodušších modech 5 nebo 10 hráčů) a společně tak čelit zlu, které na ně čeká. To většinou zahrnuje několika hodinové trénování a vymýšlení strategií, jak porazit bosse (non-player postava s důkladně propracovanou umělou inteligencí). Když se hráčům po úmorných hodinách konečně podaří porazit bosse s kterým bojovali, čeká je odměna v podobě unikátních předmětů a peněz.

2.2 Rasy a povolání

Ve WoW je v současné době 10 hratelných ras. Jsou rozlišeny na dvě skupiny podle strany, za niž bojují, a to na Alianci (Alliance) a Hordu (Horde). Rasy Aliance jsou si celkem podobné a mají stejný ideologický základ - víru v dobro a světlo, odmítající demony a temnotu. Frakce Horda je složena z ras které hledají své místo ve světě a ctí tradice. Rozhodně se tyto dvě frakce nedají rozdělit na dobro a zlo, obě strany mají v sobě od každého trochu. Mezi hlavní rasy Aliance patří Lidé (Humans), Trpaslíci (Dwarves), Skřítkové (Gnomes), Noční Elfové (Night Elves) a Draenei. K hordě patří Taureni (Tauren), Orkové (Orcs), Nemrtví (Undead), Trollové (Trolls) a Krvaví Elfové (Blood Elves).

Každá rasa (kromě Trollů a Gnómů) má své hlavní město, jemuž dominuje (Gnómové jsou v azylu u trpaslíků, protože jejich město je plné nepřátel, a Trollové pobývají s Orky, protože hlavní město Trollů má v držení jiný kmen z rozpadlé říše).

Kromě rasy si každý hráč zvolí také povolání, kterých je celkem deset. Válečník (Warrior), paladin (svatý bojovník s omezenými magickými schopnostmi), šaman (Shaman), lovec (Hunter), druid (ochránce přírody se schopností měnit se v různá zvířata), zloděj (Rogue), kněz (Priest), černokněžník (Warlock), mág (Mage) a Rytíř Smrti (Death Knight). Povolání je hlavním faktorem ovlivňujícím styl hraní a roli ve skupině více hráčů. Různá povolání mají svá jedinečná kouzla a mohou používat různé zbraně a nosit různou zbroj. Ne každá rasa si může zvolit každé povolání, Rytíř Smrti je však přístupný pro všechny bez rozdílu.

Každá rasa má své unikátní schopnosti, které ovlivňují schopnosti dané postavy, nejsou však nijak výrazné. Většina hráčů se shoduje na tom, že nejdůležitější je to, jak se vám která rasa líbí a jak se vám za ni hraje, schopnosti ras (takzvané "racialky" (z angl. racial traits)) jsou až na druhém místě.

Kapitola 3

Quice

Jedná se o software určený přímo ke správě databáze k hernímu serveru mangos. Quice slouží převážně pro uživatele, kteří nemají příliš rozšířené znalosti v okruhu databází a umožňuje jednodušší práci s jednotlivými herními částmi.

Vzhledem k tomu, že má tento software celou řadu nevýhod pro bezpečnou práci s databází a že generování změn není úplně použitelné, rozhodla jsem se vytvořit systém, umožňující správu jednotlivých částí ve hře se zachováním jednoduchosti a zároveň s omezeními, které nezpůsobí narušení samotného serveru neoprávněným přístupem nebo vložением špatných dat.

Quice je aplikace, kterou si uživatel musí stáhnout k sobě do počítače a nastavit připojení k databázovému serveru. Pokud chceme, aby měl uživatel možnost měnit data přímo na serveru, musí mít účet s právy, na který se může quice připojit. Tohle sebou nese značná rizika, protože kromě aplikace quice bude mít uživatel možnost editace i pomocí jiných softwarů, jako například Navicat.

Z tohoto důvodu jsem se rozhodla řešit systém jako webovou aplikaci, přístupnou všem uživatelům z jakéhokoli webového prohlížeče. Kromě lepší dostupnosti systému není nutné vytvářet účty k databázovému serveru pro jednotlivé uživatele, ale stačí jeden účet s právy pouze pro localhost, což značně snižuje možnosti nechtěných přístupů a modifikací v databázi.

3.1 Možnosti práce s Quice

Quice umožňuje spravovat jednotlivé herní části úkoly (quest), nestvůry (creature), objekty (gameobject), předměty (item) a postavy (charakter). Pokud není verze quice kompatibilní se strukturou databáze, dochází k chybám a tím i k omezené funkčnosti některých částí.

Pokud chceme vyhledat informace například o úkolu, musíme zadat jeho id, což může být pro některé uživatele problém, protože ne všichni ví, kde se dá potřebné id najít. Po vyhledání id se všechny informace, které náleží k hledané části, stáhnou z databáze a zobrazí se v předem připravených tabulkách.

Při změně některého pole z tabulek se nám vygenerují SQL dotazy, které zahrnují změny všech náležitostí k dané sekci, což považuji za velice závažný nedostatek tohoto systému. Kromě toho, že je naprosto zbytečné kvůli jedné změně měnit další jiné záznamy, které se změnou přímo nesouvisí, je samotná oprava v podobě SQL dotazu nepřehledná, protože zahrnuje více automaticky generovaných změn, než samotnou opravu uživatele.

Vygenerované SQL dotazy pak může uživatel odeslat do databáze, ke které je připojen a nebo zkopírovat k dalšímu použití.

Kapitola 4

Návrh systému

Vzhledem k tomu, že je systém určen pro specifický druh uživatelů, nebyla možnost získat dostatečné množství požadavků na systém, proto jsem vycházela při návrhu systému především z nedostatků aplikace quice a ze svých zkušeností získaných při správě databáze.

4.1 Požadavky na systém

Při správě serveru jsem častokrát narážela na problém, že z bezpečnostních důvodů nemá většina ostatních uživatelů, kteří se o server starají, přístup k serverové databázi, což sebou nese značné množství komplikací. Proto jsem se snažila systém vytvořit tak, aby zahrnoval všechny potřebné části a uživatelé si tak mohli sami vyhledat potřebné informace a opravit jednoduché chyby, které se na serveru vyskytnou.

Zde bych chtěla uvést několik důležitých bodů, kterými jsem se řídila při návrhu systému:

- jednoduché a intuitivní ovládání systému
- rozdělené menu pro různé skupiny uživatelů
- zahrnutí pouze podstatných částí pro zpracování
- možnost zobrazení a zpracování jen částí, které jsou problematické a mohou se v nich vyskytovat chyby
- dostupnost orientačních informací pro hráče, které se týkají jednotlivých herních částí
- generování SQL dotazu pro každou změnu, který bude zahrnovat pouze informace o provedené změně
- uložení provedených změn do vhodného formátu
- možnost potvrzovat nebo rušit změny provedené v systému
- importování dat z jiných webových stránek
- implementování kalendáře pravidelných eventů
- vyhledávání libovolných dat i mimo rozsah samotného systému

4.2 Rozdělení systému na jednotlivé části

Po analýze požadavků na systém se jeví jako vhodné rozdělit jednotlivé herní části do samostatných sekcí hlavní nabídky. Jednotlivé sekce a podsekce nabídky jsou rozepsané v následujících podkapitolách.

4.2.1 Sekce server

Tato sekce umožňuje vyhledávání informací o jednotlivých herních částech a je přístupná pouze hráčům. To je z důvodu, že sekce je pouze informativní a menu pro správce serveru obsahuje rozšířenější možnosti jednotlivých částí.

Sekce zahrnuje informace o nestvůrách (creature), objektech (gameobject), úkolech (quest) a kořisti (loot). Vzhledem k častým problémům s chybějícími nebo špatnými výskyty (spawn), jsem umožnila hráčům zjistit přesný počet výskytu nestvůr nebo objektů, které nemohou ve hře najít nebo mají pocit, že se vyskytují v menších počtech, než by ve skutečnosti měli. Tato možnost by měla primárně zajistit zredukování chybně hlášených problémů o chybějících nestvůrách nebo objektech.

Kromě nepřesných výskytů nestvůr nebo objektů bývají velmi časté chyby v úkolech, které hráči plní. Proto jsem do sekce úkol (quest) zahrnuje stručný výpis informací o úkolech. Pokud jsou všechny návaznosti na úkol v pořádku, vypíše se informace o tom, jak se úkol jmenuje, kdo ho zadává, komu se dá po splnění odevzdat, kolik hráč dostane zkušeností a jakou musí mít úroveň, aby mohl úkol přijmout. V případě, že úkol nikdo nezadává, nebo se nedá po splnění nikomu odevzdat, vypíše se na obrazovku text s chybou.

V sekci kořist (loot) se hráči mohou ujistit, že předmět, který potřebují je možné opravdu získat. Drop kořisti se vyhledává pro 3 možné zdroje: nestvůra, objekt a předmět. Pokud je záznam o dropu kořisti nalezen, spočítá se jeho průměrná drop šance a výsledek se hráči vypíše na obrazovku. Pokud je drop šance předmětu nastavena na 0, jedná se o předmět, který padá ve skupině jiných předmětů a jeho šance je odvozena od počtu předmětů ve skupině, proto se hráči místo průměrné drop šance objeví text "random drop ze seznamu itemů z <zdroj>".

4.2.2 Sekce eventy

V této sekci je kalendář událostí (event - speciální události, které se ve hře objevují jen na určitou dobu, například cirkus, dětský den, velikonoce, aj.). Události se zobrazují pro aktuální rok od aktuálního měsíce, protože je zbytečné zobrazovat událost, která se již stala v minulém měsíci.

4.2.3 Sekce tickety

Každý hráč si může ve hře založit ticket (žádost o řešení nějakého problému, který má). Pokud je ve hře nějaký správce serveru, může si vyhledat nezpracované tickety a daný problém s hráčem vyřešit. Dost často se však stává, že problémy, které hráči mají nejsou řešitelné přímo ze hry. Proto jsem do svého systému zahrnuje možnost zobrazení založených ticketů, aby se správci serveru nemuseli přihlašovat do hry jen kvůli přečtení ticketů, který stejně nemohou vyřešit.

Do této sekce mají přístup i hráči, aby se mohli podívat, jaké problémy se řeší, případně zjistit, jestli se už někdo zabýval jejich ticketem.

4.2.4 Sekce administrátor

Jak už název napovídá, jedná se o sekci, do které mají přístup pouze administrátoři serveru. Všechny změny, které uživatel v systému provede a uloží, se neodešlou přímo do databáze, ale uloží se do xml souboru. Je tomu tak z důvodu, aby se minimalizoval počet špatných změn, které nejsou korektní s myšlenkou serveru nebo nevyřeší daný problém správně. Samozřejmě tento

přístup značně redukuje i možnost hackerských útoků na serverovou databázi nebo dokonce samotný server.

Administrátor má v této sekci možnost vyhledat všechny nezpracované změny, které může po zvážení odeslat do serverové databáze nebo smazat. Pokud dojde při zpracování SQL dotazů k chybě, nastaví se parametr změny na chybný a zakomentuje se SQL dotaz u kterého došlo k chybě. Chybné záznamy lze vyhledat a pokračovat dále v jejich zpracování.

Zpracované a smazané záznamy lze dohledat podle data jejich zpracování. V případě, že by některý z administrátorů smazal SQL dotaz, který by chtěl později odeslat do databáze, může si jej v sekci smazané najít a znovu jej zpracovat.

Na serveru jsou dva xml soubory: sql.xml a wowhead.xml. Do souboru sql.xml se ukládají všechny změny, které vygeneroval uživatel pomocí systému a do souboru wowhead.xml se ukládají všechny změny, které uživatel získal z webové stránky <http://www.wowhead.com>. Rozdělení těchto změn na dva soubory je nezbytné, protože data z wowheadu bývají často velmi rozsáhlá a docházelo by tak ke zbytečnému zvětšování jednoho souboru a později i ke zpomalení vyhledávání změn ze systému.

4.2.5 Sekce hledat

Tato sekce umožňuje správci serveru vyhledat informace z databáze pomocí SQL dotazu, které samotný systém neumožňuje vyhledat, ani zpracovat. SQL dotaz je z bezpečnostních důvodů rozdělen na 3 části SELECT, FROM a WHERE. I přesto, že uživatelský účet přes který se systém přihlašuje k databázi nemá práva na zápis ani mazání, je bezpečnější rozdělení dotazu. Uživatel je tak nucen zachovávat předem stanovený formát SQL dotazu, který není nijak zvlášť omezující.

Veškeré vyhledávání je omezené pouze na databázi, která uchovává informace o světě hry a limit zobrazených záznamů je omezen na 30. Limitní omezení jsem nastavila z důvodu vyhledávání většího počtu záznamů, aby nedocházelo ke zbytečnému vytížení serveru a též z důvodu, aby uživatelé nemohli získat větší množství dat, které by mohli použít pro jiné projekty.

4.2.6 Sekce creaturey

Nestvůra (creature, mob, npc) je označení pro postavu, která je ovládaná umělou inteligencí počítače a má ve hře spoustu různých rolí. Podsekce spawny je zaměřena na vyhledání informací o výskytu (spawnu) nestvůry ve hře a umožňuje upravit jeho základní vlastnosti, mezi které patří doba znovuoživení, vybavení (equip), který má mít nestvůra na sobě, způsob pohybu a typ výskytu.

V sekci addon můžeme nestvůře upravit některé doplňující vlastnosti, přiřadit mu zvíře, které je osedláno jako kůň (mount), přiřadit cestu, po které má chodit a přidat projev emocí (emote) nebo auru. Toto nastavení ovlivní pouze jeden konkrétní výskyt nestvůry. Pokud chceme nastavit globálně vlastnosti každému výskytu jednoho typu nestvůry, můžeme nastavení provést v sekci template_addon. Pokud mají nestvůry již nastavení v sekci addon, musíme je prvně smazat, jinak se budou vlastnosti řídit nastavením pro jeden konkrétní výskyt.

Pokud má nehráčská postava zadávat úkol nebo přijímat od hráče již splněný úkol, musí mít nastaven správně flag (ten nastavíme v sekci template) a přiřazený seznam úkolů, které má zadávat nebo přijímat. Seznam přiřazených úkolů můžeme vyhledat a upravit v sekcích questrelation a involvedrelation.

Kořisti (looty) všech předmětů jsou v databázi rozděleny do samostatných tabulek podle zdroje, ze kterého mají být získatelné. Speciální část tvoří tabulka, která obsahuje referenční záznamy. Při vyhledávání kořisti jsem musela zahrnout možnost prohledávání i v referenčních odkazech, protože by mohlo často dojít k tomu, že by předmět nebyl nalezen i přesto, že existuje v referenční tabulce. Vzhledem k tomu, že jsou seznamy kořistí často rozsáhlé a obsahují velké množství dat, rozhodla jsem se rozdělit vyhledávání v seznamech kořistí pomocí drop šance. Každé vyhledání kořisti je omezeno na 30 záznamů a sortováno sestupně podle drop šance. Pokud je počet vyhledaných záznamů větší, může si uživatel nastavit drop šanci menší než je 100% a vyhledat tak záznamy od uvedené drop šance. Potřebuje-li uživatel změnit konkrétní záznam kořisti, může si jej samostatně vyhledat zadáním obou primárních klíčů a to id zdroje a id předmětu. I přesto, že je způsob hledání seznamů kořistí komplikovanější a nese s sebou množstevní omezení, je možné vyhledat a editovat jakýkoli záznam v kořisti, který uživatel potřebuje.

Sekce reputation umožňuje nastavení hodnoty reputace, kterou hráč získá s danou frakcí po zabítí nestvůry. Reputace může být kladná i záporná nebo může být přidělena jen jedné hráčské straně (aliance nebo horda).

Všechny základní parametry a vlastnosti nestvůr můžeme upravit v sekci template. Vzhledem k tomu, že informací popisujících charakter nestvůry je hodně, snažila jsem se v systému zobrazit jen ty, které bývají často problematické, nebo ty, které jsou potřeba pro jiné změny.

4.2.7 Sekce gameobjecty

Pod pojmem gameobject rozumíme nějaký objekt, který se ve hře vyskytuje. Můžeme si pod tím představit například dveře, otevírací mechanismy, květiny, rudu, různé druhy truhel aj.

Tato sekce je velmi podobná sekci creature, proto ji nebudu podrobněji rozepisovat. Narozdíl od sekce creature má však méně navazujících možností, které s objekty přímo souvisejí.

4.2.8 Sekce itemy

Předmět může mít hráč na sobě nebo v batohu. Sekce loot slouží k vyhledání, podobně jako v sekci creature a gameobject, seznamů kořistí z předmětu, představujícího krabičku, která se dá otevřít. Všechny vlastnosti a parametry předmětů můžeme opět upravit v sekci template.

4.2.9 Sekce npc

I když je nehráčská postava v podstatě nestvůra a spadá do sekce creature, oddělila jsem tuhle sekci zvlášť, protože nepopisuje obecně nestvůry, ale umožňuje nastavení speciálních rolí, které mohou nehráčské postavy ve hře mít.

Sekce gossip sice nedává nehráčské postavě nějakou významnější roli ve hře, ale umožňuje přiřazení textu, který nehráčská postava sděluje hráči, který s ní promluvil. Seznam a číslo textu, který nehráčské postavě chceme přiřadit, nalezneme v sekci text.

Trainer je označení pro nehráčskou postavu, která má schopnost učit hráče novým věcem. Trenéři jsou ve hře rozděleni na 2 hlavní části. Trenéři, kteří mají schopnost učit hráče novým kouzlům, příslušící jejich povolání a trenéři, kteří učí hráče některé z možných profesí. Správci serveru je zde umožněn náhled na kompletní nastavení požadavků pro naučení kouzla (spellu).

Kromě trenérů se ve hře vyskytují obchodníci (vendoři), kteří mohou hráči prodat různé předměty. Stejně jako v sekci trainer je zde správci umožněno nastavit všechny náležitosti k prodáváním předmětům.

4.2.10 Sekce questy

Úkol (quest) je hráči zadán prostřednictvím nehráčské postavy, objektu nebo předmětu. Tato sekce slouží k nastavení požadavků, které hráč musí splňovat, aby mu mohl být úkol zadán, definuje, co je úkolem hráče a jaká je odměna za splnění úkolu.

4.2.11 Sekce scripty

Pokud je potřeba ve hře nascriptovat nějakou jednoduchou akci, lze ji přiřadit evetu, objektu, kouzlu nebo úkolu. Vzhledem k tomu, že je scriptování poměrně jednoduché, zahrnula jsem do této sekce všechny možné části, které lze scriptovat.

Scriptování se provádí tak, že prvně musíme určit, k čemu chceme přiřadit script. Po té už jen přiřazujeme časy akcí, které chceme, aby se provedly. K dispozici je celkem sedmnáct akcí (nejčastěji používané jsou: mluvení, projev emocí, pohyb na souřadnice, přemístění, splnění úkolu, znovuobjevení objektu, přivolání nestvůry, otevření a zavření dveří). Každá akce musí mít definované doplňující nastavení, které se liší podle typu akce. Podrobný popis akcí a jejich parametrů nalezneme v dokumentaci k databázi.

Kromě jednoduchého scriptování umožňuje server pomocí databáze vytvářet i složitější scripty pro umělou inteligenci. Tuhle možnost jsem však do systému nezahrnula, protože scriptování vyžaduje důkladné nastudování a zkoušení možností na vlastním test serveru, než uživatel plně pochopí, jak to funguje a jaké jsou možnosti.

4.2.12 Postavy

Oproti ostatním sekcím, které se týkají světa samotného a řeší hru ze strany serveru, je tahle sekce zaměřena čistě na hráče a jejich postavy.

Tato sekce umožňuje vyhledat uložené instance hráčů, které navštívili a určuje, kdy se instance resetuje. Sekce je pouze informativní a neumožňuje mazání záznamů, ani jejich změnu. Je to z důvodu, že server si při spuštění načítá všechny informace z databáze do paměti a s nimi dále pracuje. Při odstranění nebo změně záznamu by tak mohlo dojít k tomu, že nám server záznam znovu uloží do databáze nebo v horším případě uloží jen jednu část, čímž by docházelo ke špatné funkčnosti a hráčům by se buď uložené instance smazaly a nebo naopak neresetovaly, když mají.

Tuto sekci jsem do systému zahrnula hlavně z důvodu, že jsou častokrát problémy s resetováním instancí. Proto má správce serveru možnost nahlédnout do savů a řešit případné problémy s hráči ve hře.

V sekci inventory můžeme smazat hráči předmět, který získal neoprávněně. Vzhledem k tomu, že seznam předmětů, co hráči u sebe mají je velmi rozsáhlý, zvolila jsem omezení vyhledávání na předmět a číslo (guid) postavy, čímž se značně sníží množství hledaných a zobrazovaných dat.

Pokud má hráč problém s nějakým úkolem, který v minulosti nabral, ale zmizel mu z jeho nabídky úkolů nebo splnil úkol, který správně nenavazoval, může mu v této sekci správce serveru úkol odebrat, čímž umožní hráči úkol znovu splnit. Pokud byly za úkol nějaké unikátní odměny, je potřeba v sekci inventory odměny smazat, aby hráč nebyl zvýhodněn.

Každá postava má při založení defaultně stanovený seznam všech frakcí s kterými může během hry získat reputaci. Některé frakce se hráči zobrazí s prvním přihlášením do hry, jiné se zobrazí, až s nimi hráč začne získávat reputaci. Tato sekce umožňuje nastavení výše reputace, kterou má hráč s danou frakcí. Podobně jako v sekci inventory jsem zvolila omezení ve formě čísla (guidu) postavy a id frakce, aby nedocházelo k získávání zbytečného množství informací.

V případě špatně fungujícího kouzla (spellu) nebo zneužívání kouzla ve prospěch hráče, můžeme v sekci spell, hráči kouzlo zrušit nebo zneaktivnit. Smazání kouzla by bylo neefektivní, protože by se jej hráč mohl jít znovu naučit od trenéra.

V sekci bany si můžeme vyhledat aktivní nebo neaktivní zamezení přístupu (ban), které hráč na serveru dostal.

Kapitola 5

Konceptuální datový model

Konceptuální či sémantické modely jsou pokusem umožnit vytvoření popisu dat v databázi nezávisle na jejich uložení. Konceptuální model představuje formální popis modelované reality.

5.1 Jazyk UML

V roce 1996 pánové Jacobson, Booch a Rumbaugh, kteří již dříve významně přispěli k rozvoji OOP metodik, přešli k firmě Rational Corp., pro kterou vytvořili jazyk UML (Unified Modeling Language). V roce 1997 byl jazyk UML akceptován jako průmyslový standard a postupně začal vytlačovat ostatní analytické jazyky. Autoři UML podle vlastních slov postupovali tak, že z existujících konkurenčních jazyků vybrali nejlepší myšlenky a implementovali je do jazyka UML, který podtrhl jejich význam integrací do nově promyšleného celku.

Model v UML se skládá z různých diagramů, jež představují průhledy na různé části sémantického základu navrhované aplikace. Sémantickým základem je souhrn specifikací aplikace, který vymezuje teritorium, v němž se může návrh pohybovat.

Jazyk UML rozeznává pět základních pohledů na systém:

- **Pohled případů užití.** V případech užití jsou vyjádřeny základní požadavky kladené na systém
- **Logický pohled** se zabývá zejména pojmy z problémové domény zadavatele a jejich vzájemnými statickými vztahy
- **Procesní pohled** se soustřeďuje na chování systému, které musí splňovat požadavky a omezení z případů užití, jež jsou kladeny na průběh procesů
- **Implementační pohled** zachycuje fyzické rozdělení aplikace na samostatné komponenty a jejich závislosti
- **Pohled nasazení** mapuje komponenty na množinu fyzických výpočetních uzlů v cílovém prostředí

Pohledy jsou konkretizovány v různých typech diagramů. Pro namodelování informačního systému jsem použila Use-case diagram ze sady UML. Použití jiných diagramů, např. ER diagramu postrádá smysl, protože je databáze k serveru již navržena a jakékoli změny ve struktuře by způsobily špatnou funkčnost emulátoru.

5.1.1 Use-case diagram

Use Case diagramy znázorňují chování systému z pohledu uživatele. Umožňují definovat ohraničený systém a jeho vztahy s vnějším okolím. Podávají v podstatě obraz funkčnosti systému, která je vyvolávána podněty z vnějšku.

Use Case diagramy zahrnují aktéry (actors), systém a případy použití (use case). Aktér reprezentuje činnost osob nebo jiný prvek ovlivňující chování systému. V žádném případě neplatí, že každá fyzická osoba je reprezentována právě jedním aktérem. Ve skutečnosti totiž může každá osoba vystupovat v různých rolích a tudíž může být reprezentována více aktéry a naopak.

5.2 Návrh Use-case diagramu

Na obrázku 5.1 je znázorněn use-case diagram systému. Účastníci systému jsou hráči a správci serveru. Systém je vytvořen pouze pro uživatele, kteří na serveru hrají, nebo se o něj starají, proto je zbytečné do návrhu systému zahrnovat i neregistrované uživatele.

Hráč

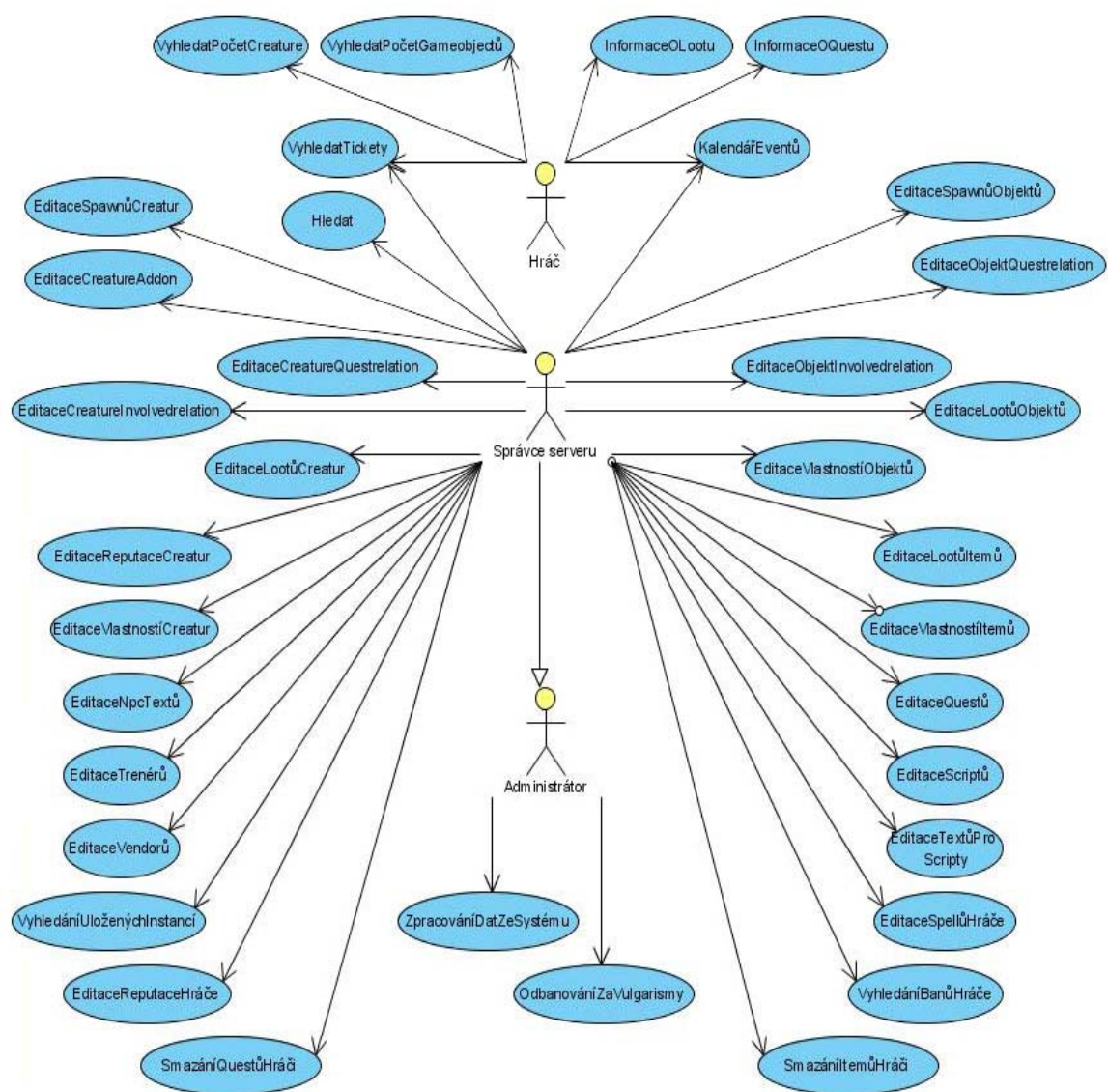
má v systému omezenou pravomoc pouze na vyhledávání informací o některých částech na serveru.

Správce serveru

je účastník, který zahrnuje jak správce serveru, tak administrátora. Má možnost generovat a ukládat SQL dotazy zahrnující různé změny na serveru.

Administrátor

kromě možností, co mají správci serveru, mohou navíc zpracovávat vygenerované SQL dotazy a vyhledávat, případně rušit bany za vulgarismy.



Obrázek 5.1: use-case diagram systému

Kapitola 6

Implementace systému

Při implementaci systému jsem se snažila dbát na to, aby byl systém jednoduchý a snadno pochopitelný i pro uživatele, kteří nemají doposud žádné zkušenosti se samotnou správou serveru.

6.1 Použité technologie

Při implementaci systému jsem použila následující technologie:

- PHP4 (pro správnou funkčnost ukládání SQL dotazů je nezbytná verze PHP5)
- MySQL5
- HTML 4.01
- CSS
- JavaScript
- AJAX

Pro funkci samotného systému postačuje verze PHP4. Nicméně funkce, zajišťující práci s xml soubory, již vyžaduje použití verze PHP5. V následujících kapitolách bych se chtěla zaměřit především na implementaci nestandardních vlastností systému.

6.2 Grafické rozhraní systému

Základem pro každou webovou stránku je zajímavý, pěkný a přehledný design. Design zapůsobí na návštěvníka prvním dojmem již při vstupu na stránku.

Samozřejmě poutavý vzhled není vše a je také nutné navrhnout web jednoduchý a přehledný, aby se nově přichozí uživatel rychle zorientoval. Kromě samotného obsahu je navigace to nejdůležitější na každé webové stránce.

Navigace by měla být zřetelně odlišená od obsahu stránky, proto se typicky umísťuje horizontálně nad obsah nebo vertikálně vedle obsahu. Při návrhu systému jsem zvolila umístění navigace doleva, vertikálně vedle obsahu. Umístění navigace horizontálně by bylo nepraktické, protože obsahuje podúrovně, které by zbytečně zakrývaly obsahovou část. Ve všech úrovních systému je k dispozici stejná navigace, která umožňuje uživateli rychlý přístup k jakékoli části systému.

Názvy navigačních odkazů jsou z části anglické a z části české. Uvědomuji si, že to nepůsobí na první pohled dobře, ale je to nezbytné z důvodu lepší orientace v systému. Anglické názvy popisují jednotlivé herní části, na které jsou jak hráči, tak správci servu zvyklí, takže jejich počeštěním by mohlo dojít k záměně významu nebo k nepochopení, o kterou část se jedná. Na druhou stranu jsem nechtěla zvolit pro systém pouze anglický jazyk, protože velká část uživatelů, kteří se systémem budou pracovat, anglicky neumí.

Celý systém je sladěný do modrozelené barvy s šedým pozadím a jako úvodní přihlašovací stránku jsem použila motivy ze samotné hry. Použitá grafika je optimalizovaná tak, aby načítání bylo co možná nejrychlejší.

Jedna z možností navigační nabídky je nápověda, aby se uživatel začal v systému orientovat co nejrychleji.

6.3 Implementace jednotlivých sekcí v systému

Nejdříve jsem implementovala kostru systému spolu s přihlašováním a odhlašováním uživatelů a poté jsem postupně implementovala jednotlivé sekce.

6.4 Správa serveru prostřednictvím SQL dotazů

Vhodný způsob jak získat data, které uživatel v systému změnil nebo vytvořil, je napsat nebo vygenerovat SQL dotaz a ten poslat do databáze nebo ho poslat administrátorovi k dalšímu zpracování. SQL dotaz by měl obsahovat změny, které uživatel provedl a zároveň změny, které jsou závislé na jeho provedené změně, aby případně nedocházelo k zbytečnému vytváření chyb.

Častou chybou při vytváření SQL dotazů bývá to, že uživatel používá pro jejich generování software, který generuje pro každou změnu delete a insert místo updatu. Což může způsobit na jedné straně opravu, ale na druhou stranu se může insertem do databáze vložit chyba jiná. Kromě možnosti vzniku chyby je tento způsob nevhodný i z toho důvodu, že z SQL dotazu není zjevné, co konkrétně autor měnil. Administrátor je tak odkázán pouze na jeho komentář, který nemusí být jednoznačný nebo může chybět.

Většinou však správci serveru neznají jazyk SQL vůbec a tak je problematické jim umožnit opravování jednoduchých chyb, které jsou intuitivní a spočívají pouze ve změně jedné hodnoty v tabulce. Na druhou stranu není bezpečné jim dát přímý přístup k databázi, aby mohli využívat některé z vizuálních nástrojů pro správu MySQL.

V závislosti na možnostech a získaných poznatcích jsem implementovala do systému automatické generování SQL dotazů jako prostředek přenášení změn vytvořené správcem serveru.

6.4.1 Generování SQL dotazů

Na základě vstupních informací, které uživatel zadal do formulářového pole, se mu vygenerují z databáze data, která se zobrazí formou tabulky, jejíž pole jdou editovat.

Při editaci jakékoli buňky v tabulce se do textového pole vygeneruje SQL dotaz. Generování SQL dotazů je zajištěno pomocí JavaScriptového frameworku jQuery. JQuery umožňuje snadno vyhledávat elementy DOMu pomocí selektorů CSS. Tento přístup má obrovskou výhodu v tom, že nevyžaduje znovunačtení stránky, a tak umožňuje generovat dotazy okamžitě, když uživatel stiskne klávesu v buňce tabulky.

6.4.2 Automatické generování updatů

Vzhledem k tomu, že je dost často potřeba změnit pouze pár hodnot v tabulce, postačí nám vygenerovat update, který popíše naši změnu.

Po načtení stránky se pomocí funkce *napln_tabulku* uloží všechny potřebné hodnoty z tabulky do paměti. Každý editovatelný sloupec je pak dále zpracováván samostatnou funkcí. U každého

pole tabulky je funkce, která zajišťuje validitu vkládaných hodnot. Ověřování validity je prováděné JavaScriptem, aby uživatel okamžitě věděl, že zadal špatnou hodnotu a nemusel se po odeslání formuláře vracet zpět a znovu vytvářet změny. Pokud uživatel zadá špatnou hodnotu, zobrazí se mu okno s popisem chyby a do pole se vrátí zpět hodnota z databáze.

V některých sekcích je možnost editovat celý sloupec. Pokud je zatržítko aktivní, nakopíruje se hodnota zadaná uživatelem do všech polí v daném sloupci a vytvoří se update, editující všechny hodnoty ve sloupci. Tato možnost má správci serveru zajistit komfortnější a rychlejší změny v systému při větším počtu záznamů.

Pokud je textové pole pro generování dotazů prázdné a uživatel změní některou z hodnot v tabulce, vypíše se nový SQL dotaz. Potřebuje-li však hodnotu znovu změnit, je potřeba zajistit, aby se negeneroval nový update, ale aby se pouze upravil již existující. Stejně tak je tomu v případě, když chce uživatel změnit hodnotu pro stejný řádek, ale jiný sloupec. Abychom zamezili vytváření nových SQL dotazů pro každé stisknutí klávesy, musíme nejprve zjistit, jestli jsme již v textovém poli vygenerovali dotaz, který můžeme použít nebo nikoli.

Vyhledávání v textovém poli je zajištěno regulárními výrazy. Nejprve zjistíme pomocí názvu sloupce, jestli je nutné generovat nový SQL dotaz, nebo můžeme použít již vytvořený. Pokud najdeme název sloupce v textovém poli, musíme ještě určit, jestli se jedná o update řádku nebo sloupce. V případě, že generujeme update pro sloupec je nutné dbát na to, že update zahrnující celý sloupec zneplatňuje updaty, editující pouze jednu hodnotu ve sloupci. Proto je potřeba updaty editující jen jednu hodnotu v daném sloupci zrušit.

To může být problematické, protože mohou být v textovém poli složené updaty zahrnující editaci několika sloupců. V tomto případě je potřeba zrušit pouze editaci pro daný sloupec, nikoli celý SQL dotaz. Nahrazování existujících updatů při vložení updatu pro sloupec, zajišťuje funkce *edit*, která postupně vyhledává všechny možné kombinace updatů a postupně je přepisuje.

6.4.3 Automatické generování insertů a deletů

U některých sekcí je možnost smazat nebo vytvořit nový záznam. Uživateli je nabídnuta tabulka, do které může vyplnit data, která chce vložit do databáze.

Vzhledem k tomu, že insertovaná a deletovaná data nemají žádnou hodnotu, která se nedá uživatelem změnit, je problém při vkládání více záznamů, zpětně dohledat správný SQL dotaz.

Zvažovala jsem řešení tohoto problému 2 způsoby:

1. Vygenerovat SQL dotaz až po vyplnění všech položek
2. Umožnit generovat pouze jeden SQL dotaz pro insert a jeden pro delete

První možnost by sice umožňovala vytvářet v rámci jedné stránky více insertů a deletů, ale nebylo by již možné vygenerovaný dotaz zpětně editovat. Tudíž v případě, že by uživatel vytvořil některý z insertů nebo deletů chybně, musel by si znovu načíst stránku a vytvořit SQL dotazy znovu. Po zvážení omezení a výhod vyplývajících z první možnosti jsem se rozhodla použít možnost druhou.

Tato možnost sice neumožňuje uživateli vytvořit v rámci jedné stránky více insertů a deletů, ale zachovává způsob generování SQL dotazů v rámci celého systému. Při editaci buňky v tabulce pro vložení nového záznamu se vygeneruje insert, který má předem stanovený počet sloupců, odpovídající počtu sloupců v databázi. Hodnota každého sloupce nabývá defaultní hodnoty, která je definovaná v databázi.

6.5 Implementace načítání informací z jiných webových stránek a ukládání těchto informací do databáze

Nejlepší a nejpoužívanější webovou stránkou zabývající se hrou World of Warcraft je <http://www.wowhead.com>. Samozřejmě existují i jiné webové stránky, které umožňují vyhledat různé informace ke hře, ale většinou informace obsažené na stránce webu nejsou přesné nebo neobsahují spolu s jednotlivými herními částmi i jejich id. Vzhledem k tomu, že v databázi se pracuje se všemi daty pomocí jejich id, je výskyt id na webové stránce pro získání dat vhodnější.

Abychom mohli získat data z jiné webové stránky, musíme si nejdříve stránku stáhnout a uložit. To uděláme pomocí funkce `get_web_page` jejíž parametr je adresa webové stránky. Funkce nám stáhne kompletně celou stránku od hlavičky, až po konec stránky.

Stránka obsahuje spoustu informací, které pro naše účely nepotřebujeme, proto je vhodné je před dalším zpracováním odstranit. V našem případě nám ke zpracování stačí pouze ta část, která vypisuje do webové stránky dané informace. Získáme ji tak, že zdrojový kód webové stránky rozdělíme na tři části, z čehož první a třetí můžeme zahodit. První rozdělení je potřeba provést pomocí části textu, který se na stránce vyskytuje pouze jedenkrát a je co nejbližší části, kterou potřebujeme získat. Druhé rozdělení je jednodušší, protože část s kterou chceme pracovat je ukončena závorkami a středníkem.

Nejdůležitější a často problematickou částí bývají drop tabulky nestvůr a seznamy předmětů, které nehrácké postavy prodávají. V systému jsem zahrnula možnost vyhledávat dropy a prodávané předměty jak podle id nehrácké postavy, tak podle id předmětu.

6.5.1 Získávání kořisti z webové stránky

Pokud chceme vyhledat seznam předmětů podle id nestvůry, musíme nejprve získat celkový počet pokusů, ze kterého můžeme později vypočítat drop šanci předmětů. Podle počtu pokusů můžeme též rozdělit stránku na dvě části a získat tak pouze potřebný odstavec, který dále budeme zpracovávat.

Pro samotné vyhledávání seznamu předmětů nebo nestvůr jsem použila funkci `preg_match_all`, která pomocí regulárního výrazu vyhledá v textu všechny odpovídající záznamy a postupně je uloží do pole.

Aby byly získané informace použitelné pro databázi k serveru, je nutné odlišit, jestli id předmětu odpovídá předmětu pro úkol (quest itemu). Jedná se o takový předmět, který má drop podmíněný úkolem, tudíž padá jen v případě, že má hráč nesplněný úkol ve svém seznamu úkolů. V databázi se drop šance předmětu pro úkol odlišuje zápornou hodnotou. Abychom zjistili z webové stránky, jestli se jedná o takový předmět, musíme kromě id předmětu získat i hodnotu `class`, která určuje o jaký typ předmětu se jedná.

Vzhledem k tomu, že se data na wowheadu stále mění získáváním nových informací, musíme drop šanci vypočítat procentuelně z počtu zabití nestvůry a z počtu, kolikrát předmět padl.

6.5.2 Získávání prodávaných předmětů z webové stránky

Kromě id předmětů, které obchodník (vendor) prodává, potřebujeme zjistit v jakém množství je prodává, popřípadě jaká je cena, pokud je cena předmětu jiná než v penězích. Počet prodávaných předmětů získáme z hodnoty `avail`.

Pokud je cena předmětu v penězích, nemusíme ji řešit vůbec, protože je tahle informace uložena v templatu předmětu. Pokud je cena předmětu uvedená v podobě jiného předmětu, honoru, aréna bodů aj., je potřeba ji získat z databáze. Může se však stát, že předmět v databázi nemá žádný záznam a proto nemůžeme speciální cenu předmětu zjistit. V takovém případě se místo čísla, určující speciální cenu vypíše "chybí" a uživatel je nucen hodnotu přepsat ručně na validní číslo.

6.6 Používání vulgarismů ve hře

Častokrát si hráči na českých serverech ani neuvědomují, že jejich způsoby vyjadřování jsou otřesné a že tím dokonce mohou kazit i požitky ze hry jiným hráčům. Samotná hra umožňuje nahrazování anglických vulgarismů za určitou sekvenci nealfabetických znaků, čímž se vulgarismus znečitelní a zbytek věty zůstane nezměněný.

Vzhledem k tomu, že se do hry nedá zasahovat, musíme cenzurování českých vulgarismů provádět na straně serveru. Některé servery mají podobný způsob autocenzury implementovaný, ale způsoby vyhledávání vulgarismů nejsou většinou moc dobré. Často dochází k záměně vulgarismu za normální slovo a některé vulgarismy jsou dokonce ignorovány.

V následujících kapitolách bych se chtěla zaměřit na možné způsoby obcházení algoritmů pro vyhledávání vulgarismů a na řešení, které jsem pro tuhle problematiku zvolila.

6.6.1 Možné způsoby obcházení cenzury

Jako vhodný materiál pro získání různých způsobů, jakými se hráči snaží obejít cenzurování, mi postačily logy konverzací získané z našeho serveru, kde se hráči snažili obejít nově implementovaný algoritmus pro cenzurování vulgarismů.

Mezi možné způsoby obcházení cenzury patří:

- proložení slova nealfabetickými znaky
- proložení slova mezery
- opakování písmen ve slově
- přesmyčka písmen nebo vynechání některých písmen

Proložení slova nealfabetickými znaky je způsob, který většinou napadne uživatele jako jeden z prvních. Uživatelé často používají k proložení znak * (např. c*e*n*z*u*r*a), tečku (c.e.n.z.u.r.a) nebo pomlčky (c-e-n-z-u-r-a). Samozřejmě musíme vycházet i z předpokladu, že ne každý uživatel prokládá jednotlivá písmena nealfabetickými znaky, ale že jej může vložit mezi kterékoli písmena ve slově, stejně tak nemusí použít jen jeden, ale i více (cen*zura nebo cen.....zura).

Proložení slova mezery je podobné, jako v případě prokládání nealfabetickými znaky, akorát se k oddělení písmen používá mezera (např. cen zura, c e n z u r a)

Opakování písmen ve slově byl způsob, který hráči příliš nepoužívali, ale přesto je to jeden z možných způsobů obcházení. Hráči opakují převážně samohlásky, protože český jazyk zdvojená písmena ve slovech téměř nemá a tak je pro většinu lidí zvláštní napsat např. cccccenzura. Většina lidí by spíše volila možnost cenzuraaaaa.

Přesmyčka písmen nebo vynechání některých písmen je sice jeden z možných způsobů obcházení cenzury, ale otázka je, jestli vynecháním písmen, či přesmyčkou písmen můžeme ještě hovořit o vulgarismu. Slovo je sice čitelné a většina lidí ho pochopí jako vulgarismus, ale může se stát, že překlepem v nevulgárním slově získáme výraz, který náš autocenzor označí za vulgarismus i přesto, že slovo vulgární není.

6.6.2 Implementace algoritmu pro vyhledání vulgarismů

Před samotným vyhledáváním jsem nejdříve provedla úpravu textu. Abych nemusela později hledat nealfabetické znaky a řešit jejich význam v textu, rozhodla jsem se je odstranit. Kromě toho jsem nahradila písmena, obsahující českou diakritiku za písmena bez diakritiky, aby nedocházelo k obcházení algoritmu používáním háčeků a čárek ve slovech, kde být nemají.

6.6.3 Tvorba slovníku sprostých výrazů

Kromě možných způsobů obcházení je důležité věnovat pozornost i možnosti záměny nevulgárního slova za vulgarismus. Český jazyk je velmi dynamický a členitý, proto jsem se při tvorbě slovníku snažila zaměřit na různé tvary slov a možnosti záměny.

Slovník sprostých výrazů je popsán regulárními výrazy a vychází z předpokládaného tvaru slova. Regulární výraz popisuje každý vulgarismus tak, aby co nejlépe zahrnul všechny tvary vulgarismu, zahrnul možnosti obcházení vložением mezer nebo opakováním písmen a zároveň zajistil, že nedojde ke shodě se slovem, které je nevulgární.

Ke shodě často dochází, pokud je slovo ve slovníku příliš krátké nebo je sprostý výraz tvořen koncem a začátkem jiného slova (např. na serveru, strašná sranda, chlapi co děláte, to je bug). Jako velmi problematické jsou třípísmenné výrazy, ale ke shodě může dojít i u výrazů, které jsou na čtyři a víc písmen (např. opice, sekunda, knihovna). Abych předešla možné záměně nevulgárního slova za vulgarismus, uvedla jsem u problematických výrazů kromě písmen, které slovo může nebo musí obsahovat i mezery na začátku nebo na konci výrazu.

6.6.4 Zpracování nalezených vulgarismů

Pokud najde algoritmus v prohledávaném textu slovo, které označil za vulgarismus, nalezne zbytek textu spolu s autorem vulgarismu a uloží jej do xml souboru. Abychom mohli zpětně dohledat, kdo je autorem, uložíme k textu i jméno postavy. Vzhledem k tomu, že hráč může mít více postav, je lepší kromě jména postavy uvést k vulgarismu i číslo účtu vlastníka postav, které získáme z databáze.

Po uložení všech potřebných informací do xml souboru, se vytvoří pro každého hráče, který byl skriptem nalezen, záznam do databáze, který jim zamezí přístup na server na jeden den.

Data z xml souboru lze zobrazit v sekci vulgarismy zadáním data od kdy do kdy nás nalezené vulgarismy zajímají. Uvedením čísla účtu (accountu), se zobrazí vulgarismy pouze pro daný účet. Pokud má uživatel platný záznam za vulgarismy, který mu zamezuje přístup na server, může mu administrátor omezení přístupu zrušit.

Kapitola 7

Testování a provoz systému

V průběhu vývoje jsem umožnila některým správcům serveru přístup k mému systému, aby mi pomohli s odlaďováním chyb a přispěli tak svými nápady ke zlepšení funkčnosti i práce se systémem. V současné době je systém na serveru přístupný všem hráčům i správcům serveru.

7.1 Omezení posílaných informací

I když je server v danou chvíli určen jen pro provoz hry, snažila jsem se omezit množství informací, které jsou uživateli posílány a které by mohly zbytečně zatěžovat server. Z toho důvodu je některé vyhledávání informací omezené. Pokud potřebuje uživatel nalézt informace, které se mu nezobrazili, musí zadat pro vyhledání více vstupních dat.

Později jsem zjistila, že by větší obsah informací zobrazený na stránce způsoboval i problémy s JavaScriptem, který rozeptá v následující kapitole.

7.2 Optimalizace JavaScriptu při získávání dat z wowheadu

Při testování systému v provozu jsem objevila problém s JavaScriptem. Při zpracování menšího obsahu dat JavaScript fungoval bez problému. Problém nastal ve chvíli, kdy jsem potřebovala zpracovat data získaná z wowheadu, která v některých případech dosahovala až 200 záznamů na stránce.

Původně data získaná z wowheadu byla zpracovávána stejným scriptem, jako data vyhledaná z databáze, což způsobilo značné zpomalení při práci na stránce a některé prohlížeče dokonce vypisovaly chybu. JavaScript se snažil uložit obsah tabulky do paměti i pro data vyhledaná z wowheadu, což byl klíčový problém.

Výsledná tabulka pro seznamy kořistí (looty) z databáze obsahuje více sloupců, než tabulka pro data získaná z wowheadu a JavaScript se snažil ukládat data pro sloupce, které neexistují. Snaha uložit neexistující data způsobila, že při načtení stránky JavaScript nefungoval správně.

Řešení problému bylo jednoduché. Stačilo vytvořit nový soubor, který se bude zabývat pouze zpracováním dat z wowheadu, čímž se značně urychlil JavaScript zpracovávající stránku.

7.3 Instalace systému

Pro zprovoznění systému je potřeba zkopírovat zdrojové soubory na server. Aby správně fungovalo ukládání do xml souborů, musíme povolit práva zápisu všem souborům ve složce system/data. Následně musíme vytvořit databázi k serveru, kterou nalezneme ve složce server/databáze a nastavit správně konfigurační soubory config.php, config_ban.php a config_modify.php ve složce system/php.

Kapitola 8

Porovnání s existujícími systémy a možnosti rozvoje

Při realizaci systému jsem vycházela z některých vlastností již existujících systémů pro správu databází. Velká část uživatelů používá Quice, proto jsem se snažila z této aplikace vycházet a zanechat některé charakteristické prvky, aby se uživatelé nemuseli učit pracovat s novým systémem.

8.1 Porovnání s Quicem

Princip hledání a zobrazování dat je velmi podobný. Drobný rozdíl nalezneme však v možnosti vyhledání, kde Quice neumožňuje vyhledat některou z herních částí podle jména. Vyhledávání podle jména bylo přání jednoho ze správců serveru, který mi pomáhal s testováním. Důvodem je, že jsou uživatelé, kteří neví, kde sehnat id a pro některé může být zadání jména i pohodlnější.

Podstatný rozdíl nalezneme v možnosti generování SQL dotazů a jejich zpracování. Quice neumožňuje zpracovat SQL dotaz jinak, než jej zkopírovat nebo vložit do databáze. Tento princip zpracování se mi nelíbil, protože není možnost zpětně dohledat, jaké změny byly provedeny. V případě, že by uživatel nemohl vložit data do databáze, musel by zkopírovaný SQL dotaz poslat administrátorovi nebo využít jiných možností, např. subversion. Tento způsob je jak pro uživatele, tak pro administrátora nepraktický, protože vyžaduje použití jiných aplikací, než je samotný systém. Z tohoto důvodu jsem umožnila zpracování vytvořených SQL dotazů v rámci systému.

Jako novou možnost jsem do svého systému implementovala vyhledávání informací z wowheadu. Častokrát je potřeba změnit nebo vytvořit nové záznamy drop tabulek, které obsahují velké množství informací a ruční vypisování by bylo pro většinu uživatelů nepříjemné.

8.2 Možnosti rozvoje

Armory - název pochází z oficiálních serverů a jedná se o webovou aplikaci, která umožňuje zobrazit jednotlivé postavy hráčů, předměty, které mají na sobě, počet životů, jejich dovednosti (skilly), talenty, dosažené reputace, aj. Armory je velice žádané, ale zatím ještě nikdo nevyvinul plně funkční systém, který by umožňoval zobrazování postav i na soukromých serverech. Proto bych se chtěla na tuhle problematiku v budoucnosti zaměřit a implementovat ji do systému.

Rozšíření možností systému - prozatím je systém navržen tak, aby zahrnoval základní možnosti správy jednotlivých herních částí. V budoucnosti však zvažuji, že systém rozšířím i možnosti spravování umělé inteligence, tvořené prostřednictvím databáze.

Získávání dat z jiných webových stránek - získávání dat by se v budoucnosti dalo rozšířit i na jiné webové stránky, než je wowhead, i když jej považuji za nejlepší zdroj informací. Jako další rozšíření by mohlo být získávání jiných informací, než jsou drop tabulky nestvůr nebo seznamy prodáváných předmětů.

Algoritmus pro vyhledávání vulgarismů - tato problematika by si rozhodně zasloužila další vývoj. I když jsem se snažila zahrnout nejčastější způsoby obcházení, věřím, že si uživatelé v budoucnu vymyslí nové, kterými se budou snažit mnou navržený algoritmus obcházet. Dále jsem zvažovala možnost implementace vkládání nových slov do slovníku prostřednictvím systému, aby nemusel administrátor editovat samotný kód scriptu.

Kapitola 9

Závěr

Během práce na serveru jsem se častokrát setkávala s problémem, že jiný správce serveru potřeboval vědět, jaká je nastavená hodnota v databázi, popřípadě ji změnit. To byl ovšem problém, protože i když se v databázi orientoval, neměl k ní přístup.

Díky tomuto problému jsem vyvinula systém umožňující správci, který se v databázi vyzná, měnit a vyhledávat potřebné informace a zároveň umožňující i správcům, kteří tyto znalosti nemají, poměrně rychlou orientaci v systému a rychlé pochopení jednotlivých problematik.

Za velký přínos pro server považuji možnost získání dat z wowheadu. Takle možnost ušetří spoustu práce s ručním vypisováním a vkládáním do databáze a umožní tak optimalizovat databázi a rychle řešit špatné záznamy v tabulkách.

Jako přínos považuji i algoritmus pro vyhledávání vulgarismů, i když z hráčské strany moc pozitivních ohlasů nezískal. Oproti jiným algoritmům, které porovnávají celé nebo části slov ze slovníku, jsem vytvořila slovník, který je popsán regulárními výrazy, což by mělo značně omezit záměnu s nevulgárními výrazy a zároveň zamezit možnostem obcházení algoritmu.

Vytvoření tohoto systému je velkým přínosem pro správu serveru. Díky rychlému pochopení základních věcí mohou i lidé, bez znalosti SQL přispět k úrovni serveru opravou minoritních chyb, čímž se sníží vytížení na programátory, kteří se tak mohou věnovat složitějším problémům.

V budoucnosti plánuji další vývoj systému, nejen v závislosti na uvedených bodech, ale hlavně v závislosti na přáních uživatelů, kteří systém využívají. Doufám, že v budoucnosti budu moci nabídnout svůj systém i jiným skupinám, zabývajícím se provozováním vlastních severů pro hru World of Warcraft.

Literatura

- [1] Kolektiv autorů: *PHP Manual*. 2009, <http://www.php.net/manual/en/index.php>
- [2] Kosek, J.: *PHP - tvorba interaktivních internetových aplikací*. Praha: Grada Publishing, 1999, ISBN 80-7169-373-1.
- [3] Janovský, D.: *Jak psát web*. 2009, <http://www.jakpsatweb.cz>, ISSN 1801-0458
- [4] Resig, J.: *jQuery*. 2009, <http://jquery.com>
- [5] Cyroň, M.: *CSS kaskádové styly praktický manuál*. Praha: Grada Publishing, 2005, ISBN 80-247-1420-5.
- [6] Flanagan, D.: *JavaScript Kompletní průvodce*. Computer Press, 2002, ISBN 80-7226-626-8

Významový slovník herních pojmů

event - za event se považuje nějaká událost, která ve hře není stále, ale objeví se jen na určitý časový okamžik.

ticket - jedná se o formu žádosti o pomoc. Pokud má hráč ve hře nějaký problém, může založit ticket, kterým se bude někdo ze správců serveru zabývat

creatura, mob, npc - (non-player character) postava, která je ovládána umělou inteligencí počítače

gameobject - objekt, který ve hře představuje dveře, otvírací mechanismy, květiny, rudu, truhly aj.

item, equip - jedná se o předmět, který hráč může během hry získat.

quest - úkol, prostřednictvím kterého získávají hráči zkušenosti a jednotlivé úrovně

level - úroveň hráče

xp - (Experience Points) zkušenosti

spawn - místo ve hře, kde se vyskytuje nějaký objekt nebo nehráčská postava

addon - soubor vlastností, kterých může nehráčská postava nabývat

questrelation - definuje zahájení questů

involvedrelation - definuje ukončení úkolů

loot, drop - forma kořisti, kterou hráč získá po zabití moba

reputation - reputace

template - šablona, definující vlastnosti a parametry objektů, nestvůr a předmětů

gossip - přiřazený text k nehráčské postavě, který nahrazuje defaultní text "Greetings \$N". \$N určuje jméno postavy s kterou se nehráčská postava baví

trainer - trenér, který může učit hráče novým kouzlům

vendor - prodejce, který nabízí hráči předurčený seznam předmětů

spell - kouzlo

instance - za instanci je označován prostor, který se vytváří pro každou skupinu hráčů. V instanci, narozdíl od okolního světa, se mohou potkat pouze hráči, kteří jsou spolu ve skupině

inventory - inventář, do kterého si hráči ukládají předměty, co ve hře našli

character - postava hráče

wowhead - webová stránka <http://www.wowhead.com>, zaměřená na hru world of Warcraft

mangos - Massive Network Game Object Server je objektově orientovaný serverový projekt, který podporuje síťový protokol hry World of Warcraft. Dalšími podobnými projekty jsou Trinity, Ascent (dříve známý jako Antrix), aj.

Seznam příloh

Příloha 1. Ukázky uživatelského rozhraní

Příloha 2. Zdrojové texty k informačnímu systému

Příloha 3. CD obsahující kompletní elektronickou podobu

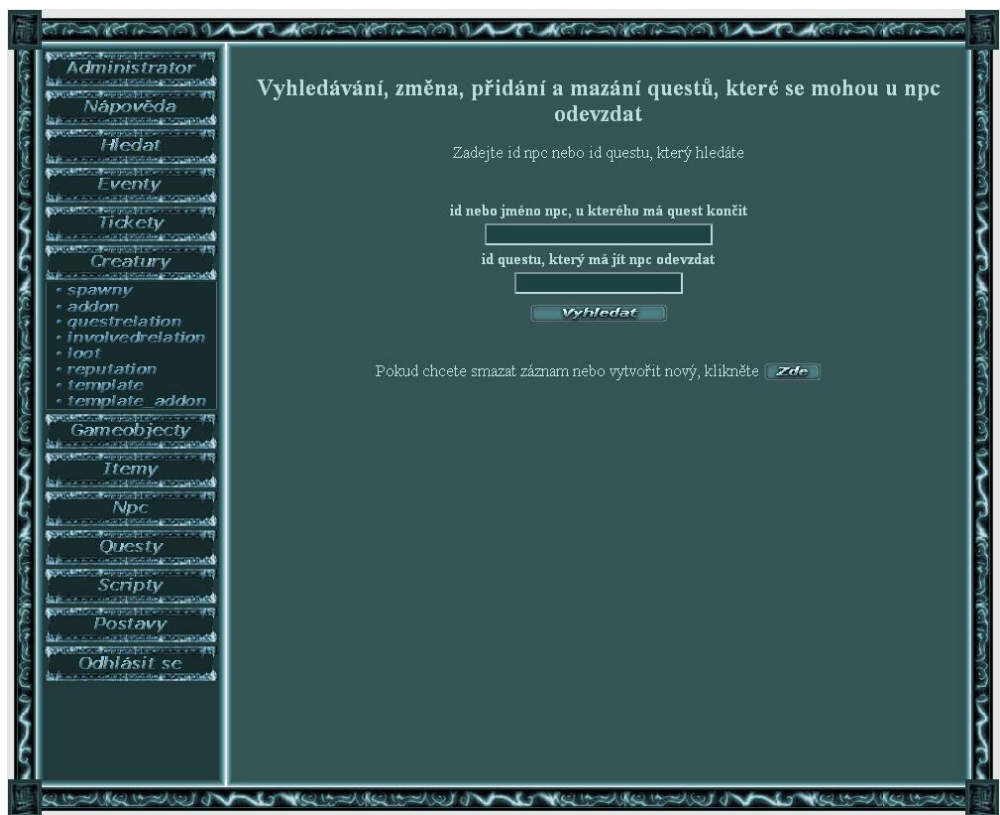
Příloha 1

Ukázky uživatelského rozhraní

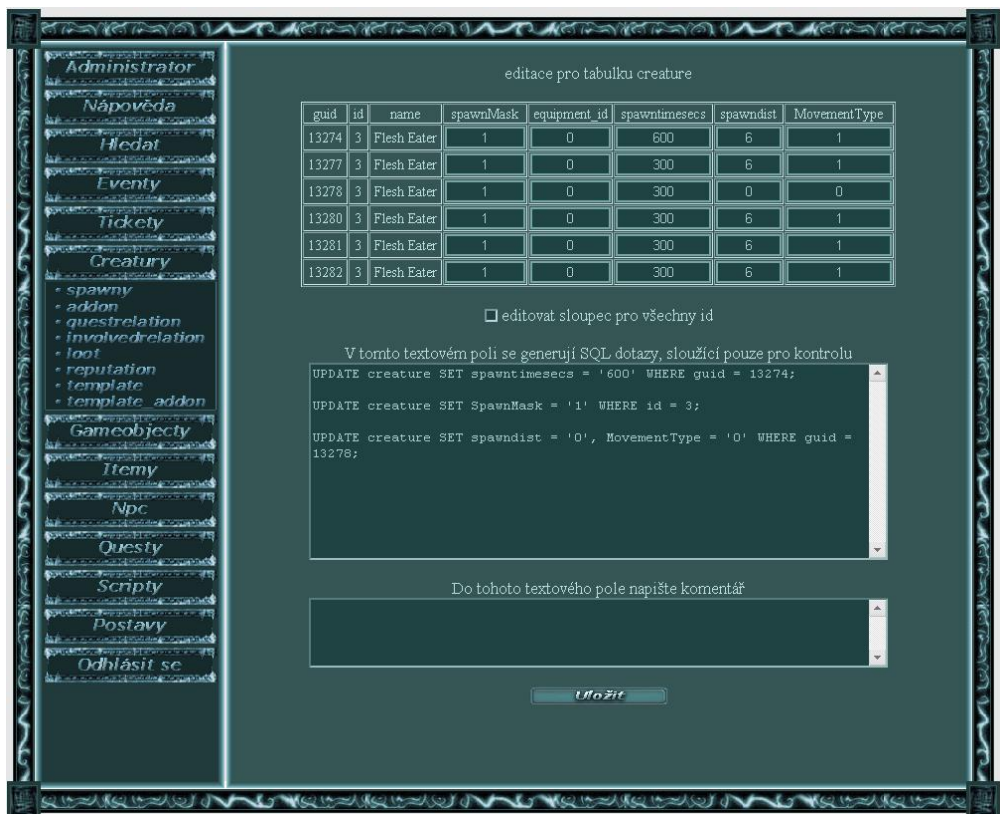
V této části jsou uvedeny ukázky uživatelského rozhraní pro informační systém. Na **obrázku 1** je zobrazen náhled pro přihlášení. Možnost vyhledávání je na **obrázku 2**, kde je zobrazeno vyhledávání pro sekci involvedrelation. **Obrázek 3** představuje možnosti generování SQL dotazů a **obrázek 4** jejich zpracování v sekci administrátor. Na **obrázku 5** je ukázka možnosti stažení informací z webové stránky <http://www.wowhead.com>.



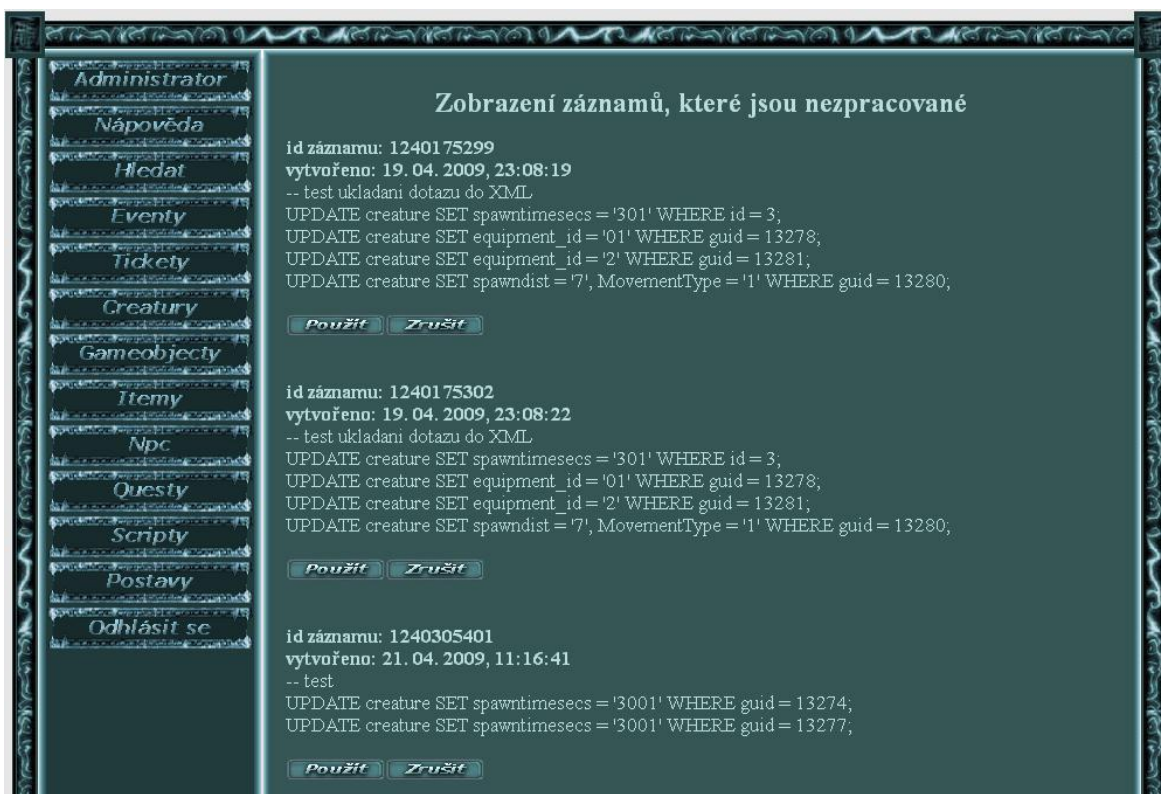
Obrázek 1: Přihlašovací obrazovka



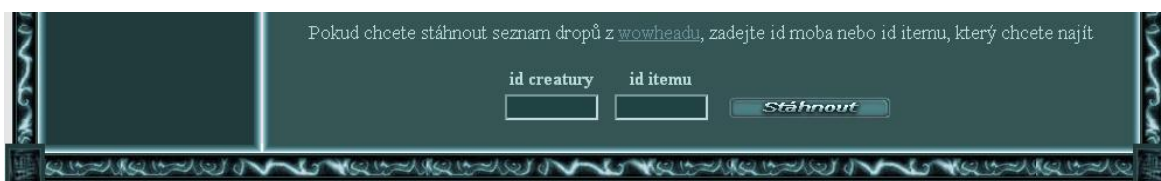
Obrázek 2: Ukázka vyhledávání



Obrázek 3: Ukázka generování SQL dotazů



Obrázek 4: Ukázka zpracování vygenerovaných SQL dotazů



Obrázek 5: Možnost stažení informací z jiné webové stránky