

# **Návrh a realizace databázového systému ve formě webové hry**

Design and implementation of the database system in the browser  
game form

Bc. Zbyněk Marada

---

Diplomová práce  
2011



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2010/2011

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Zbyněk MARADA**  
Osobní číslo: **A09682**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Návrh a realizace databázového systému ve formě  
webové počítačové hry**

Zásady pro vypracování:

1. Analyzujte problematiku a vypracujte literární rešerši na dané téma.
2. Navrhněte strukturu databáze a webových stránek systému.
3. Vytvořte databázi a aplikaci, popište postup řešení.
4. Zaveďte základní prvky pro vyhledávání (SEO).
5. Zajistěte správu a zabezpečení systému.
6. Vytvořte podporu uživatelů (helpdesk).

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **CASTAGNETTO, Jesus, et al. Programujeme PHP profesionálně. Praha : Computer Press, a. s., 2002. 656 s.**
2. **ASLESON, Ryan; SCHUTTA, Nathaniel T. Ajax : Vytváříme vysoce interaktivní webové aplikace. Brno : Computer Press, a. s., 2007. 263 s.**
3. **SMIDEK, Vojtěch; BERIT, Jaroslav; MRÁZEK, Lubor. PHP v příkladech. České budějovice : Kopp, 2001. 117 s.**
4. **SATRAPA, Pavel. Web design. Havlíčkův Brod : Neokortex spol. s.r.o., 1997. 414 s.**
5. **JANOVSKÝ, Dušan. Jak psát web [online]. 2003 [cit. 2011-01-26]. Jak psát web. Dostupné z WWW: [http://www.jakpsatweb.cz].**
6. **ŽDÁREK, Roman. SEO Optimalizace pro vyhledávače [online]. c2004 [cit. 2011-01-26]. Jak dělat weby. Dostupné z WWW: [http://www.jakdelatweby.cz/seol].**

Vedoucí diplomové práce:

**doc. Ing. Zdenka Prokopová, CSc.**

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

**24. února 2011**

Termín odevzdání diplomové práce:

**18. května 2011**

Ve Zlíně dne 24. února 2011

prof. Ing. Vladimír Vašek, CSc.  
*děkan*

L.S.

doc. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Diplomová práce se věnuje vytváření databázového systému ve formě webové hry, za použití běžných technologií, kterými jsou HTML, PHP a MySQL. Práce obsahuje analýzu rozebírající aktuální situaci webových her a zabývá se vývojem nové hry jak z pohledu softwarového inženýrství tak z pohledu implementace jednotlivých částí hry. Samotná hra obsahuje jak část uživatelskou tak část administrátorskou, včetně helpdesku. Práce zároveň řeší i zabezpečení celého systému a optimalizaci pro internetové vyhledávače.

Klíčová slova: webová hra, databáze, automobilový závod, helpdesk, HTML, PHP, MySQL, bezpečnost, cross-site scripting, SEO

## **ABSTRACT**

Diploma thesis is dedicated to creating a database system in the browser game form, with used the classic technology, that are HTML, PHP and MySQL. The work includes analysis, described the current situation of browser game, and deals with the development of new games from the side of software engineering and from the side of implementation of all parts of the game. The game contains a user and admin section, including helpdesk. The work also solves the security of the system and search engine optimization.

Keywords: browser game, database, car race, helpdesk, HTML, PHP, MySQL, safety, cross-site scripting, SEO

## **Poděkování**

Tímto bych chtěl poděkovat doc. Ing. Zdence Prokopové CSc., za odbornou pomoc při tvorbě diplomové práce a za poskytování cenných rad, při psaní této zprávy.

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně  
16.5.2011

podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 WEBOVÉ HRY</b> .....	<b>11</b>
<b>2 ANALÝZA</b> .....	<b>13</b>
2.1 TVŮRCI WEBOVÝCH HER .....	13
2.1.1 Bigpoint .....	13
2.1.2 Game Forge .....	13
2.1.3 Travian Games .....	14
2.1.4 Další tvůrci .....	14
2.2 SROVNÁNÍ HER .....	14
2.2.1 Srovnání podle tématu.....	16
2.2.2 Srovnání podle stylu.....	18
2.3 ZHODNOCENÍ ANALÝZY .....	19
<b>3 POUŽITÉ TECHNOLOGIE</b> .....	<b>20</b>
3.1 HTML.....	20
3.2 CSS.....	20
3.3 JAVASCRIPT.....	21
3.3.1 JQuery .....	21
3.4 PHP.....	21
3.5 MYSQL.....	22
<b>4 PRÁCE S PHP A MYSQL</b> .....	<b>23</b>
4.1 MYSQL OBECNĚ .....	23
4.2 KOMUNIKACE PHP S MYSQL.....	24
<b>II PRAKTICKÁ ČÁST</b> .....	<b>26</b>
<b>5 POUŽITÝ SOFTWARE</b> .....	<b>27</b>
<b>6 POPIS SYSTÉMU</b> .....	<b>29</b>
6.1 NEFORMÁLNÍ POPIS .....	29
6.2 MODEL PŘÍPADŮ UŽITÍ.....	30
<b>7 NÁVRH SYSTÉMU</b> .....	<b>34</b>
7.1 NÁVRH DATABÁZOVÉHO MODELU .....	34
7.2 STRUKTURA WEBU .....	37
7.3 ROZVRŽENÍ STRÁNKY .....	39
7.4 UŽIVATELSKÉ ROZHRAŇÍ.....	40
7.4.1 Rychlé informace .....	41
<b>8 IMPLEMENTACE</b> .....	<b>42</b>

8.1	UŽIVATELSKÁ ČÁST .....	42
8.1.1	Registrace uživatele.....	42
8.1.2	Přihlášení uživatele .....	43
8.1.3	Automobil .....	43
8.1.4	Vykreslení automobilů .....	44
8.1.5	Nákup autodílů .....	46
8.1.6	Instalace autodílů.....	47
8.1.7	Plnění úkolů .....	48
8.1.8	Závody.....	49
8.1.9	Trénink .....	50
8.2	HELPDESK .....	50
8.2.1	FAQ.....	50
8.2.2	Vlastní dotaz.....	51
8.2.3	Dotazy ostatních uživatelů .....	51
8.3	ADMINISTRACE SYSTÉMU .....	51
8.3.1	Možnosti správce .....	52
8.3.2	Možnosti administrátora.....	52
8.3.3	Simulátor závodů .....	52
8.3.4	Helpdesk.....	53
<b>9</b>	<b>ZABEZPEČENÍ SYSTÉMU .....</b>	<b>54</b>
9.1	CROSS-SITE SCRIPTING .....	54
9.2	PHP A SQL INJEKCE .....	55
9.3	CROSS-SITE REQUEST FORGERY .....	57
9.4	DETEKCE ÚTOKŮ .....	58
9.5	UKLÁDÁNÍ HESEL .....	58
<b>10</b>	<b>TESTOVÁNÍ .....</b>	<b>60</b>
<b>11</b>	<b>SEO .....</b>	<b>61</b>
<b>12</b>	<b>INSTALACE SYSTÉMU .....</b>	<b>62</b>
	<b>ZÁVĚR .....</b>	<b>63</b>
	<b>ZÁVĚR V ANGLIČTINĚ .....</b>	<b>64</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>65</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>66</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>67</b>
	<b>SEZNAM TABULEK.....</b>	<b>68</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>69</b>

## ÚVOD

Cílem diplomové práce bylo navrhnout a následně realizovat webový databázový systém ve formě webové hry. Důvodem výběru takového tématu byl zájem o problematiku databázových webových aplikací, zvláště pak webových her, které se v posledních letech stávají stále více populárními. Celá práce je rozdělena na dvě části, a to část teoretickou a část praktickou.

V teoretické části se práce zabývá webovými hrami obecně. Je zde popsáno, co taková hra je, a proč jsou v dnešní době tak oblíbené. Jelikož nebylo zadáno konkrétní téma či styl hry, předchází samotnému návrhu celková analýza, ve které se zabývám dostupností webových her převážně na české scéně. Zde zmíním některé z největších výrobců těchto her a představím produkty, které u nás patří k nejúspěšnějším. V teoretické části ještě představím jednotlivé technologie, pomocí kterých byl systém vyvíjen, a obecně popíši i práci s některými z nich.

Praktická část se pak zaměří na návrh a implementaci systému. Nejprve zde bude celý systém popsán, a to jak neformálně, tak v podobě modelu případů užití. Ukáži zde databázový model systému, včetně popisu jednotlivých jeho částí. Dále bude popsáno uživatelské rozhraní i struktura celé aplikace. Praktická část dále bude obsahovat podrobný popis jednotlivých částí aplikace, a to jak z pohledu uživatele, tak z pohledu administrátora. Jelikož je u podobných aplikací kladen důraz na jejich bezpečnost, budu se závěrem praktické části zabývat zabezpečením celého systému, proti případným útokům. Také zde popíši, jakým způsobem byla aplikace optimalizována pro internetové vyhledávače. Praktickou část uzavře popis testování celé aplikace na různých operačních systémech, a jeho instalace na webový server.

## I. TEORETICKÁ ČÁST

## 1 WEBOVÉ HRY

V dnešní době je internet nezbytnou součástí života velké části populace. Mnozí jej potřebují na práci či ke studiu, jiní na komunikaci, a někteří jen pro zábavu. Poslední možnost se týká spíše mladší generace, která si na internetu krátí dlouhé chvíle hraním internetových her.

Výhodou takové hry je, že pro její hraní stačí i méně výkonný počítač s připojením k internetu, a libovolným webovým prohlížečem. A jelikož takovou kombinaci má dnes skoro každý doma, nebo alespoň ve škole při hodinách informatiky, staly se internetové hry velice populárními.

Jedním typem her, nazvěme je třeba jednorázové, jsou hry, kterou si uživatel spustí, zahraje si a opět hru vypíná bez toho, aniž by se hra nějak uložila. Takové hry jsou většinou vytvářeny v programu Adobe Flash. Takzvanou „fleškovku“, jak se hrám také říká, hraje hráč sám nebo jen s pár přáteli v reálném čase.

Pojďme se ale spíše podívat na druhý typ her, který se týká i této práce. Jsou to hry založeny na rozsáhlém databázovém systému, který uchovává hru v takovém stavu, v jakém ji hráč opustil. Díky tomu jsou tyto hry dlouhodobějšího rázu, a hráč tak stále může vylepšovat svoje postavení ve hře. Další faktor, který přidává takovým hrám na úspěšnosti, je počet hráčů. Na rozdíl od jednorázových her zde totiž můžete soupeřit řádově s tisíci, nebo dokonce i s desítkami tisíc spoluhráčů. Příkladem může být například hra Travian, kterou považuji za jednu z nejhranějších her nejen v České republice. Tato hra je rozšířená po celém světě, a jen u nás má přibližně 51 000 aktivních hráčů.

Takové hry je také možné vytvářet, alespoň z části, v programu Adobe Flash, a na internetu taky pěknou řádku těchto her nalezneme a jsem přesvědčen o tom, že budou oplývat skvělou dynamikou. Spousta her však stále vsází na klasický styl tvorby dynamických stránek za použití HTML, nějakého serverového programovacího jazyka a databázového systému. Takové hry jsou pak tzv. „klikačky“, kde si pro obsluhu takové hry vystačíme pouze s myší. Pro lepší představu uvedu příklad, kdy například pro splnění nějakého úkolu či mise, nemusí hráč fyzicky ovládat nějakou postavu, ale jen stiskne tlačítko pro splnění mise, ta se za určitý čas splní a hráč dostane určitou odměnu.

Mohlo by se zdát, že takový styl hraní nebude pro hráče příliš lákavý. Opak je však pravdou. Hra skýtá výhodu v tom, že hráč nemusí u takové hry trávit denně několik hodin,

a i přesto, že se přihlásí jen několikrát za den na krátkou chvíli, dokáže udržet krok s aktivnějšími hráči.

Otázkou pak zůstává, proč lidé vůbec hrají takové hry? Někdy to bývá lákavá výhra pro nejúspěšnější hráče. Mnohdy však stačí přirozená lidská soutěživost a touha být v něčem lepší než ostatní. Zároveň to může být jistý způsob relaxace, která člověka přivede na jiné myšlenky.

## 2 ANALÝZA

Cílem analýzy bylo zjistit, jaké webové hry se dnes na zábavním trhu objevují, a to především ty, které jsou dostupné z České republiky, a nejlépe i v českém jazyce. Jelikož neznám nikoho, kdo by se vývojem webových her zabýval, přistoupil jsem k analýze z pohledu hráče, který si chce vyhledat a zahrát hru. Proto jsem spustil vyhledávač Google a hledal.

Říká se, že se stránky, které nejsou při vyhledávání zobrazeny na první stránce, jakoby neexistovali. Tohoto hesla jsem se při analýze nedržel tak doslovně, ale pravdou je, že když by hráč při vyhledávání dostal na první straně deset různých her, jen málokterý by se podíval na stranu další.

### 2.1 Tvůrci webových her

Při hledání jsem narazil na několik firem, zabývajících se tvorbou webových her. Jedná se především o německé firmy, které však provozují i hry u nás, v českém jazyce. Jedna z největších, alespoň co se počtu her týče, je firma Bigpoint.

#### 2.1.1 Bigpoint

Bigpoint je německý výrobce her. Na kontě má zatím 14 prohlížečových her, které jsou dostupné i v českém jazyce. Hry jsou vyvíjeny především v PHP a Adobe Flash. Podle internetových stránek je po celém světě registrováno přibližně 190 milionů hráčů. Témata her jsou různorodé a vedou od starověkého Řecka, přes období mafie, až po futuristické sci-fi.

#### 2.1.2 Game Forge

Jedná se opět o německého výrobce, který se může pyšnit třinácti prohlížečovými hrami, které jsou rozšířené po celém světě, ve většině případů i včetně České republiky. Na kontě má hráči známé hry jako třeba Gladius, Bite fight nebo Ikariam.

### 2.1.3 Travian Games

Za zmínku určitě stojí i výrobce Travian Games, taktéž se sídlem v Německu. Tahle firma nemá na kontě tolik titulů jako předchozí dvě firmy, nicméně vše dohání kvalitou, o čemž svědčí již zmiňovaná hra Travian.

### 2.1.4 Další tvůrci

Mezi další německé vývojáře patří i firma Inno Games. Má na kontě čtyři hry, z nichž tři jsou dostupné i u nás. Jednou z nich je například Tribal wars, která je u nás známá pod názvem Divoké kmeny.

Výrobce Playa games je mladá firma, která vznikla roku 2009 opět v Německu. Pouze jedna hra, z celkových tří, je hratelná v češtině, nicméně v českém herním světě patří k velmi oblíbeným. Tou hrou je Shakes & Fidget a po celém světě má přes jeden milion hráčů.

Jako poslední zmíním i jednoho Ruského výrobce a to firmu AmberGames. Je tvůrcem tří prohlížečových her, z nichž však žádná není dostupná v českém jazyce.

## 2.2 Srovnání her

Abych mohl vybrat pro svoji hru vhodné téma, musel jsem provést srovnání her, které dnes dominují na české herní scéně. Do této analýzy jsem zapojil veškeré hry od výše zmiňovaných výrobců. V následující tabulce jsou uvedeny prohlížečové hry od jednotlivých výrobců. U každé z nich je uvedeno téma a styl hry. Co jednotlivé témata a styly znamenají bude popsáno dále v podkapitolách, kde bude i uvedeno přehledné grafické porovnání. Mimo to je v posledním sloupci tabulky uvedena informace o tom, zda je hra dostupná i v českém jazyce.

Tabulka 1: Seznam her podle jednotlivých tvůrců

	Název hry	Téma	Styl	Česky 
Bigpoint	Seafight	Piráti	Vývojová	ano
	Darkorbit	Sci-fi	Vývojová	ano
	Farmerama	Zvířecí	Budovatelská	ano
	Deepolis	Sci-fi	Budovatelská	ano
	Xblaster	Sci-fi	Vývojová	ano
	Damoria	Středověk	Budovatelská	ano
	War of Titans	Starověk	Vývojová	ano
	Gangs of crime	Mafie	Vývojová	ano
	Hell Blades	Vojenské	Vývojová	ano
	K.O. Champs	Sportovní	Vývojová	ano
	MG-Hockey	Sportovní	Budovatelská	ano
	MG-Soccer	Sportovní	Budovatelská	ano
	Space invasion	Sci-fi	Budovatelská	ano
	Zoomumba	Zvířecí	Budovatelská	ano
GameForge	Europa 1400	Středověk	Budovatelská	ano
	Ikariam	Starověk	Budovatelská	ano
	Bite Fight	Fantasy	Vývojová	ano
	Gladius	Starověk	Vývojová	ano
	Battle knight	Středověk	Vývojová	ano
	Ogame	Sci-fi	Budovatelská	ano
	Gun West	Divoký západ	Budovatelská	ano
	Tunoth	Fantasy	Vývojová	ano
	WarpFire	Sci-fi	Vývojová	ano
	RageClaw	Fantasy	Vývojová	ano
	Kings Age	Středověk	Budovatelská	ano
	Katsuro	Středověk	Vývojová	ano
	Drivals	Závodní	Vývojová	
Inno Games	Grepolis	Starověk	Budovatelská	ano
	The West	Divoký západ	Budovatelská	ano
	Tribal Wars	Středověk	Budovatelská	ano
	SevenLands	Fantasy	Vývojová	
Travian Games	Travian	Starověk	Budovatelská	ano
	Weewai	Sportovní	Budovatelská	
	Imperion	Sci-fi	Budovatelská	
	Goalunited	Sportovní	Budovatelská	ano
	Ballersunited	Sportovní	Budovatelská	
Playa Games	Shakes & fidget	Fantasy	Vývojová	ano
	Soccer Star	Sportovní	Budovatelská	
	Blood Game	Fantasy	Vývojová	
Amber Games	1100 AD	Středověk	Budovatelská	
	Shards of the dream	Fantasy	Vývojová	
	1 Pirat	Piráti	Vývojová	

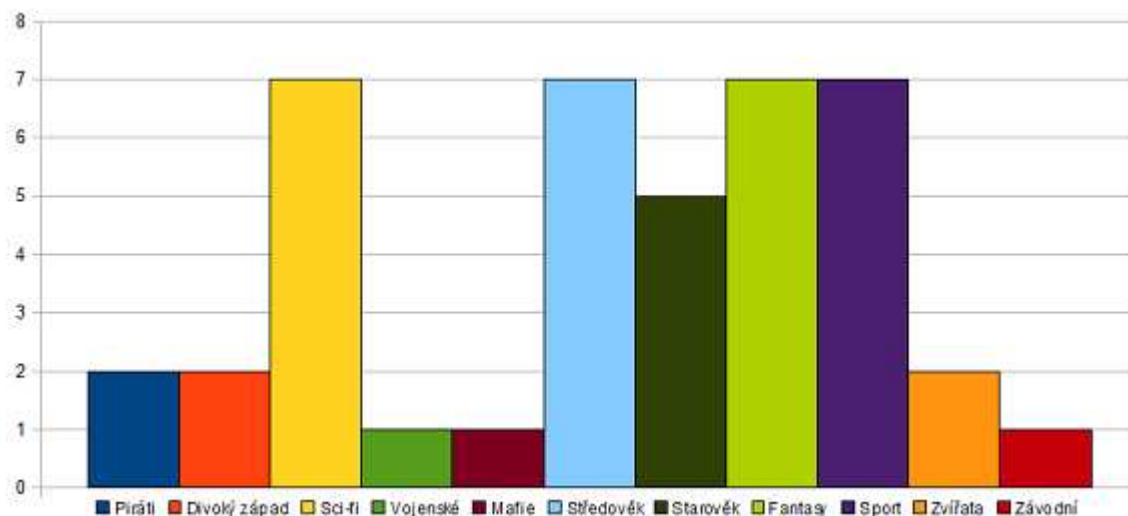
### 2.2.1 Srovnání podle tématu

Téma hry, nám zde říká, o čem samotná hra vlastně je. Když jsem procházel jednotlivé hry, přišel jsem na konci celkem k jedenácti různým tématům, do kterých lze hry zařadit. Podívejme se na výčet jednotlivých témat i se stručným popisem.

- **Pirátské** – Tematika je zřejmá na první pohled. Období pirátů, kteří na svých lodích brázdí karibským mořem.
- **Divoký západ** – Období kovbojů a indiánů stojící proti sobě na divoké prérii.
- **Středověk** – Období rytířů, kteří neznají jiné zbraně, než jejich meč a štít.
- **Starověk** – Jedná se období starověkého Říma, kde největší zábavou jsou především gladiátorské zápasy.
- **Mafie** – Období černých smokingů a klobouků, kde města jsou ovládána gangstery a mafii.
- **Fantasy** - Téma podobné většinou středověku, s tím rozdílem, že není dbán důraz na realističnost, a proto se zde objevují draci a jiné vymyšlené postavy. Proto jsem do této kategorie zařadil i hry, ve kterých jde o boj mezi upíry a vlkodlaky.
- **Sci-fi** – Většinou pohled do budoucnosti, kde se děj odehrává ve vesmíru, a objevují se mimozemské civilizace. Do této kategorie jsem však zařadil všechny futuristické příběhy a proto tu najdeme i téma z podmořského světa.
- **Zvířata** – zvláštní kategorie, kde se tematika točí především okolo zvířat. Jde například o budování vlastní zoo nebo farmy.
- **Vojenské** – válečná tematika ze současné či nedávné doby.
- **Sportovní** – Tematika se soustředí především na fotbal a hokej, kdy hráč je v pozici manažera svého týmu. Objeví se tu však i baseball či box.
- **Závodní** – Téma zabývající se auty a automobilovými závody.

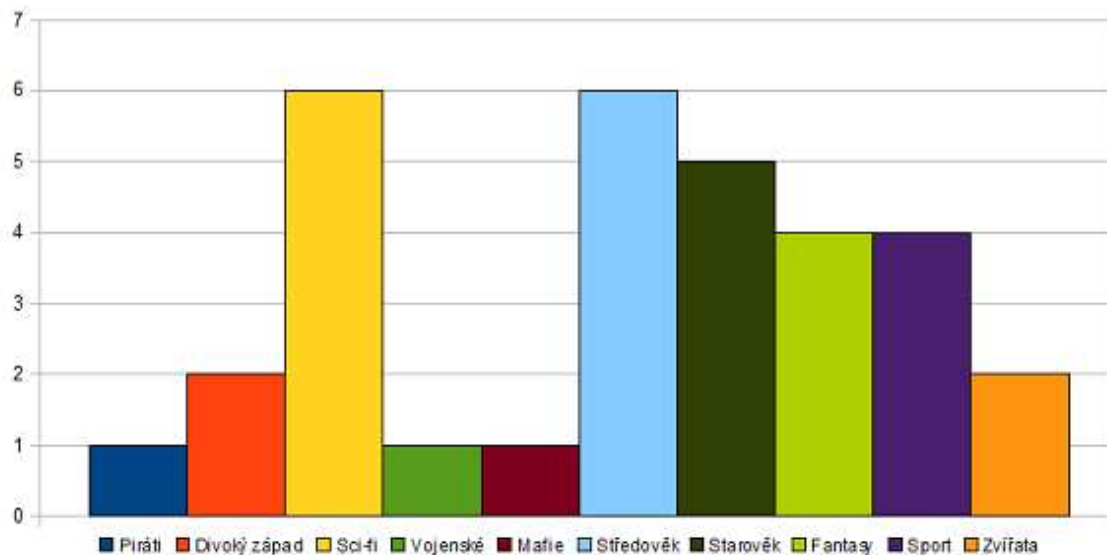
Takové rozdělení je u některých tématech poněkud hrubé, a téma by se tak třeba dalo dále rozdělit podle určité specifikace. Ve finále by jsme však mohli zjistit, že každá hra je něčím specifická, a proto toto rozdělení považuji za dostatečné.

Po zařazení jednotlivých her do těchto kategorií, získáme následující graf.



Obrázek 1: Rozdělení prohlížečových her podle tématu

V grafu můžeme vidět, kolik her spadá do jaké kategorie. Můžeme pak říci, že nejrozšířenějším tématem je sci-fi, fantasy, středověk a sport, těsně následovaný starověkem. Podívejme se ještě jak se graf změní, když se omezíme pouze na české tituly.



Obrázek 2: Rozdělení českých prohlížečových her podle tématu

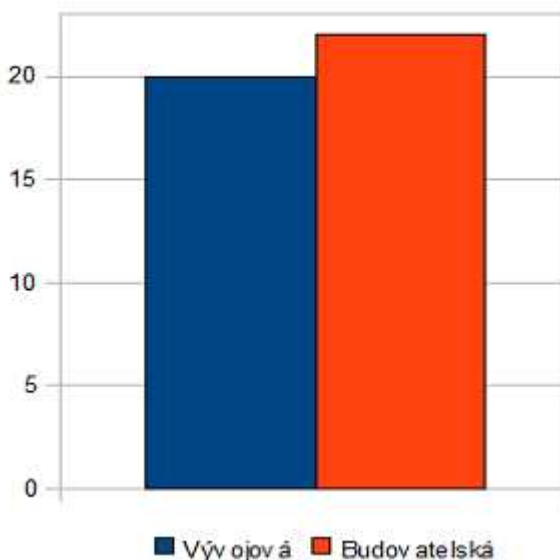
Zde si můžeme povšimnout, že pokles zaznamenaly sportovní a fantasy hry, a také nám z grafu zcela vypadla závodní kategorie.

### 2.2.2 Srovnání podle stylu

Druhým kritériem pro srovnání bude styl, jakým se hra hraje. Jinak řečeno, ve hře jde většinou o to, něco rozvíjet, v něčem se zlepšovat a být lepší než ostatní. Čili princip je vždy stejný. Proto rozdělení podle stylu je bráno spíše podle toho, jak hráč tohle rozvíjení vnímá. Hry jsem tak rozdělil buď na budovatelské nebo vývojové. Pro lepší představu popíši jednotlivé kategorie.

- Vývojové – při takových hrách, se sám hráč stává typicky jednou postavou ze hry. V průběhu hry pak plní úkoly, z jejichž odměn získává lepší výbavu, oděv či výzbroj, čímž se stává silnější oproti druhým.
- Budovatelské – zde se hráč stává majitelem typicky jedné vesnice, umístěné na jednom místě rozsáhlé mapy. Pro rozvoj pak těží suroviny, staví domy a vytváří armády. Hra se pak stává více strategičtější. K tomuto typu her jsem zařadil i sportovní manažerské hry, což by mohla být kategorie sama o sobě, nicméně princip zůstává stejný v tom, že se hráč stará o určitou skupinu a ne jen o jednu postavu.

Když si potom necháme takto rozřazené hry vykreslit v grafu, výsledek bude následující.



Obrázek 3: Rozdělení prohlížečových her podle stylu

Jak můžeme vidět, výsledky jsou téměř vyrovnané. Podobný graf bychom dostali i kdybychom vycházeli pouze z českých her.

### 2.3 Zhodnocení analýzy

Na základě analýzy jsem si pro svoji práci zvolil téma „automobilové závody“, a to z toho důvodu, že jsem mezi českými hrami na podobnou hru nenarazil, jak může být patrné z předchozích grafů. Navíc i v Evropě se podobných her nevyskytuje mnoho ve srovnání s ostatními tématy. Přitom si myslím, že klasické PC hry s automobilovou tematikou mají mezi hráči velkou skupinu příznivců, kteří by právě podobné téma ocenili i v podobě webových her.

### 3 POUŽITÉ TECHNOLOGIE

V této kapitole shrnu veškeré technologie, které byly použity při tvorbě aplikace, a jednotlivé technologie popíši.

#### 3.1 HTML

HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. [11]

Jazyk byl navržen v roce 1990 i s protokolem pro jeho přenos v počítačové síti (HTTP). HTML je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (*Standard Generalized Markup Language*). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka. [11]

Základními stavebními kameny jazyka HTML jsou jednotlivé elementy – tzv. tagy (značky). Zdrojový kód HTML není nijak překládán a dále šířen jako samostatný spustitelný soubor, po celou dobu návrhu, ladění a prohlížení zůstává v textové podobě. O jeho zobrazení se postará prohlížeč, který funguje jako interpret. [11]

Pro psaní HTML kódu není zapotřebí žádný speciální program. Vystačíme si s běžným textovým editorem, jako je například NotePad. Pro usnadnění práce však existují editory, které dokáží barevně zvýraznit syntaxi nebo například napovídat značky. Jeden z takových editorů je dosti rozšířený PSPad, ve kterém byl vytvářen i program k této práci. [13]

#### 3.2 CSS

CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. [13]

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Za pomocí CSS nastavujeme u tzv. selektorů jednotlivé atributy od velikosti a barvy písma až po rozmístění jednotlivých bloků na stránce. Selektorem v

tomhle případě může být tag HTML, určitý element se specifickým ID, nebo celá skupina elementů, která má nastaven společný atribut CLASS. [13]

Interpretace CSS stylů se může v jednotlivých prohlížečích lehce lišit. Proto je nutné při vytváření webové aplikace výsledný produkt otestovat na různých prohlížečích, a popřípadě za pomoci různých triků a „hacků“ aplikaci ošetřit, aby vypadala všude stejně. [13]

### 3.3 Javascript

JavaScript je programovací jazyk, který se používá v internetových stránkách. JavaScript je klientský skript. To znamená, že se program odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonávám.[13]

JavaScript lze využít pro tvorbu programů, nebo pro oživení stránek. Pomocí JavaScriptu a CSS (DHTML) můžeme dosáhnout velice zajímavých dynamických efektů. JavaScript se oproti PHP skriptům vykonává přímo v prohlížeči čtenáře, proto je rychlý, a k jeho funkčnosti je zapotřebí jen internetový prohlížeč, který podporuje JavaScript. JavaScript se může zapisovat přímo do webových stránek nebo do externích souborů s příponou .js.[13]

Nevýhodou javascriptu je fakt, že uživatel může mít v prohlížeči javascript zakázán, nebo může používat prohlížeč, který javascript vůbec nepodporuje. Tato skutečnost je však dnešní době spíše výjimečná a určitě se netýká hráčů webových her.[13]

#### 3.3.1 JQuery

JQuery je javascriptová knihovna, která klade důraz na interakci mezi JavaScriptem a HTML. Stejně jako CSS odděluje „zobrazovací“ charakteristiky od struktury HTML, jQuery odděluje „chování“ od struktury HTML. Cílem JQuery je také usnadnit programátorovi práci s javascriptem. Výsledné skripty v JQuery jsou mnohem kratší, než stejný skript napsaný čistě v javascript.

### 3.4 PHP

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 zadanych 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. [12]

Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez větších úprav na mnoha různých operačních systémech. Podporuje mnoho knihoven pro různé účely - např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP...). [12]

PHP je programovací jazyk umožňující procedurální i objektově orientované programování. PHP také patří mezi jazyky, kde například není nutné předem definovat typ proměnných, navíc jakákoliv proměnná, může kdykoli změnit svůj typ. To dává programátorovi určitou volnost při psaní kódu. [12]

### 3.5 MySQL

MySQL je multiplatformní databázový systém. Komunikace s databází probíhá pomocí jazyka SQL. Díky tomu, že se jedná o volně šiřitelný software je stále hojně využíván při vytváření webových databázových systémů.

## 4 PRÁCE S PHP A MYSQL

### 4.1 MySQL obecně

Jak již bylo zmíněno, celý systém je implementován za pomoci jazyku PHP a databázového systému MySQL. Tato kombinace je pro webové aplikace velice běžná, protože PHP obsahuje spoustu funkcí, kterými můžeme s MySQL databází komunikovat. Než si ukážeme funkce, které v aplikaci používám pro práci s databází, podívejme se jak MySQL pracuje s databází.

Jak bylo psáno dříve, komunikace probíhá za pomoci SQL dotazů. Nejčastější dotaz na tabulku začíná jedním z těchto 4 příkazů:

- SELECT – používáme, když chceme vybrat data z tabulky
- UPDATE – použijeme, pokud chceme upravit záznam v tabulce
- DELETE – použijeme pro smazání záznamu z tabulky
- INSERT – používáme při vkládání nového záznamu do tabulky

Důležitým slůvkem je pak WHERE, které se používá pro filtrování dat a objevuje se ve většině dotazů. Pro představu a lepší pochopení příkladů, které budou v práci později uvedeny, uvedu obecnou ukázkou jednoduchého SQL dotazu.

```
SELECT sloupec FROM tabulka WHERE podminka
```

MySQL od verze 5.0 podporuje procedury, triggerů či pohledy, čímž dříve oproti jiným databázovým systémům zaostávala. Dnes je však můžeme směle využívat a usnadnit si tak práci s databází. Jelikož se dále v práci budou zmíněné názvy objevovat, stručně je popíši.[1]

Uložená procedura je sada příkazů SQL, která je zkompileována a uložena na serveru. Klienti se mohou odkazovat na uloženou proceduru, místo provádění všech těchto příkazů SQL. Pomocí uložených procedur získáme výhodu výkonu, protože příkaz SQL jsou již analyzovány a zkompileována na serveru a mezi serverem a klientem je potřeba posílat méně dat. [1]

Trigger je uložená procedura, která je vyvolána, když nastane konkrétní událost. Trigger je nastaven na určitou tabulku i na konkrétní příkaz, který je nad tabulkou

proveden. Například při smazání řádku z tabulky, ke které se váže trigger na příkaz DELETE, se provede činnost, která je v triggeru nastavena. [1]

Pohled je přizpůsobená prezentace dat obsažených v jedné nebo více tabulkách. Pohled přebírá výstup dotazu a zachází s ním jako s tabulkou, proto o pohledu můžeme uvažovat jako o „uloženém dotazu“ nebo „virtuální tabulce“. Pohledu není přiřazeno žádné místo uložení. [1]

## 4.2 Komunikace PHP s MySQL

Nyní se podívejme, jak se na databázi dotazujeme pomocí PHP. Nejprve malá ukázka krátkého kódu, který bude záhy popsán.

```
(1) $pripojeni = mysql_connect($host, $uzivatel, $heslo);  
(2) mysql_select_db($databaze, $pripojeni);  
(3) $vysledek = mysql_query("select * from users",$pripojeni);  
(4) $radek=mysql_fetch_array($vysledek);  
(5) $id=$radek['id'];  
(6) mysql_close($pripojeni);
```

V bodě (1) vytvoříme připojení k databázovému systému. Příkazem v bodě (2) zvolíme požadovanou databázi. (3) Provedeme libovolný SQL dotaz. V bodě (4) vybereme z výsledku dotazu jeden řádek v podobě pole. Konkrétní hodnotu získáme v bodě (5). Poslední bod (6) nám ukončí spojení s databázovým systémem.

Tohle byl základní postup jednoduchého dotazu na databázi z PHP. Funkcí existuje samozřejmě celá řada. V následující tabulce je výčet funkcí použitých při implementaci aplikace.

Tabulka 2: Seznam PHP funkcí pro práci s databází MySQL [14]

<b>mysql_affected_rows</b>	Vrátí počet ovlivněných (změněných) záznamů v MySQL po posledním dotazu
<b>mysql_close</b>	Ukončí (uzavře) MySQL spojení
<b>mysql_connect</b>	Vytvoří spojení s MySQL Serverem
<b>mysql_errno</b>	Vrátí číselnou hodnotu chybové hlášky předchozího MySQL příkazu
<b>mysql_error</b>	Vrátí text chybové zprávy předchozího MySQL příkazu.
<b>mysql_fetch_array</b>	Načte výsledný řádek do asociativního, číselného pole nebo obojího
<b>mysql_fetch_row</b>	Načte výsledný záznam do pole
<b>mysql_insert_id</b>	Vrací generovanou hodnotu id posledního příkazu INSERT
<b>mysql_num_rows</b>	Vrací počet záznamů ve výsledku
<b>mysql_query</b>	Pošle MySQL dotaz
<b>mysql_real_escape_string</b>	Upraví řetězec pro bezpečné použití v mysql_query
<b>mysql_select_db</b>	Nastaví MySQL databázi

## **II. PRAKTICKÁ ČÁST**

## 5 POUŽITÝ SOFTWARE

V následující kapitole stručně popíši veškerý software, který byl při vytváření práce použit. Především, že jsem si vybral jen takový software, který je volně dostupný, a tudíž jeho používání bylo zcela legální.

Jistě nejdůležitějším programem je textový editor, ve kterém byli napsány veškeré zdrojové texty. Pro tento účel jsem zvolil program PSPad. Tento textový editor podporuje otevření mnoha formátů, nicméně jeho předností je jednoduché ovládání, přehledná práce s více soubory najednou, které jsou řazeny vedle sebe v podobě záložek, a snad nejvíce ocenitelná je barevně odlišená syntaxe. Vedle mnoha dalších formátů dokáže zvýraznit syntaxi pro HTML (\*.html) i PHP (\*.php) soubory, čímž se programování stává snadnějším.

Pro práci s obrázky a obecně s grafikou jsem kombinoval programy Gimp a Inkscape. První jmenovaný je bitmapový editor s částečnou podporou vektorové grafiky. V komerční oblasti by šel přirovnat k programu Adobe Photoshop. Je to silný nástroj zejména pro úpravu fotografií. Tento program jsem využil zejména pro výřezy z fotografií či změnu barevné hloubky.

Druhým grafickým programem je již zmíněný program Inkscape. Tento grafický software slouží především pro práci s vektorovou grafikou. Kdybych ho měl opět porovnat s nějakým známějším placeným programem, zvolil bych program CorelDraw. Program zvládá kreslení základních geometrických tvarů, besierových křivek a oplývá rozsáhlými možnostmi výplní, včetně různých barevných přechodů. Práce s ním se stává přehlednější díky možnosti používat vrstvy, které lze zamykat či skrývat. V práci byl tento program použit pro kreslení některých tlačítek, či pro další zpracování obrázku po úpravě v programu Gimp.

Pro návrh databáze byl použit program MySQL WorkBanch. Za pomoci toho softwaru jsme schopni navrhnout celý databázový systém. Přehledně nám umožňuje vytvořit novou tabulku, zvolit její název a jednotlivé sloupce, u kterých můžeme zvolit typ i vlastnosti, které nabízí databázový systém MySQL. Mezi tabulkami dále můžeme vytvářet základní vazby a podle toho i volit cizí klíče. Výsledné grafické zpracování je pak přehledné, protože jsou zobrazeny a vyznačeny všechny zmiňované údaje. Návrh tabulek

pro náš systém bude uveden a popsán dále, kde bude vidět právě i přehlednost databáze z programu MySQL WorkBanch.

Při vývoji byl použit i program VirtualBox, což je virtualizační program umožňující simulaci jiného operačního systému. K čemu byl program využit bude popsáno dále.

Pod operačním systémem Windows byl použit i software XAMPP. Nejedná se vyloženě o program, nýbrž o balíček, který zajistí v počítači fungování PHP a MySQL. Díky němu můžeme provádět i různé konfigurace či vypínat a spouštět jednotlivé části. Alternativou je možná více známý balíček EasyPHP či VentrigoServ, nicméně pod systémem Windows Vista se mi nejvíce osvědčil právě XAMPP.

Za zmínku stojí i administrační software Adminer. Opět nejde o program, ale pouze o PHP skript, který spustíme v běžném internetové prohlížeči. Adminer poskytuje grafické uživatelské rozhraní pro správu databázového systému. Umožňuje vytváření tabulek, triggerů či indexů, zkrátka veškeré operace, které můžeme s MySQL databází provádět. Podobný produkt a možná i více známý je PhpMyAdmin. Výhodou Admineru je však to, že jde pouze o jeden skript, který stačí načíst v prohlížeči.

## 6 POPIS SYSTÉMU

Než se pustím do popisu jednotlivých částí systému, začnu neformálním popisem celého systému, aby měl čtenář představu, o čem celá práce vlastně je. Doposud se totiž mohl akorát dočíst, že půjde o webovou hru na téma automobilových závodů, což je pořád dosti obecná představa.

V druhé podkapitole se potom podíváme na systém ve formálnějším popisu z pohledu softwarového inženýrství v podobě use-case modelu, neboli modelu případů užití.

### 6.1 Neformální popis

Jak už to běžně u webových her bývá, aby mohl uživatel hrát, musí si vytvořit svůj herní účet. To provede jednoduchou registrací. Po té se již kdykoliv může ke svému účtu přihlásit za pomoci přihlašovacího jména a hesla. Při prvním přihlášení si hráč vybere auto, se kterým již může začít svou kariéru závodníka.

Na začátku tedy hráč začíná s autem, nemá však žádný kredit ani zkušenosti. Kreditem se rozumí virtuální platidlo, za které si hráč například může kupovat lepší autodíly, se kterými auto vylepší. Zkušenosti pak udávají schopnost řidiče, které může využít při závodech s ostatními hráči.

Jak kredity, tak zkušenosti může hráč získávat několika způsoby. Prvním z nich je plnění úkolů. Splnění úkolu trvá určitý čas, a za jeho dokončení hráč získá určitou výši kreditu či zkušeností. Druhou možností jak získat pouze zkušenosti je trénink. Délku tréninku si hráč řídí sám. Kdykoliv může trénování zahájit i ukončit. Za dobu po kterou hráč trénoval pak dostane patřičné zkušenosti. Třetí možností, je účast na závodech. Za jakoukoliv účast na závodech získá hráč zkušenosti, nicméně umístí-li se na vrchních příčkách, obdrží i určité množství kreditu.

Aby hráč mohl ovlivňovat své počínání na závodech, musí se patřičně starat o své auto, které může různě vylepšovat. Uživatel může vylepšovat motor auta, kola, přední, střední i zadní část karoserie a v případě že má hodně našetřeno může vyměnit karoserii celou. Jednotlivé autodíly může za své kredity nakupovat v obchodě, a zakoupenými díly pak může ve své garáži auto vylepšit. Díky tomu se budou auto zlepšovat jeho vlastnosti.

Vlastnosti auta, které hráč ovlivňuje jsou rychlost, brzdy, smýkavost, akcelerace a aerodynamika. Každý z těchto parametrů je pak zohledněn při závodu v závislosti na trase, na které se závod koná. Například při klikatých trasách uživatel uplatní automobil s dobrými brzdami a velkou akcelerací, naopak při trasách s dlouhými úseky ocení spíše vysokou rychlost a dobrou aerodynamiku. Při závodech se zohlední samozřejmě i získané zkušenosti z předchozích závodů či z tréninku.

Samotné závody vypisuje správce hry či administrátor, který určí dobu závodu, trasu, na které se závod pojede, odměnu pro vítěze a limit, který udává kolik se může závodů nanejvýš zúčastnit hráčů. Hráč si pak může vypsaný závod vyhledat a přihlásit se na něj.

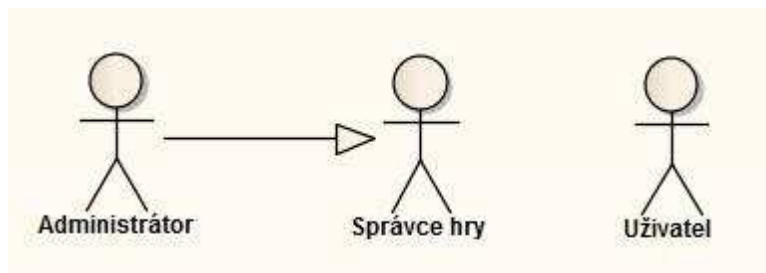
Součástí aplikace je i administrátorská část, pomocí které lze základní prvky hry snadno dotvářet. Administrátor nebo uživatel s jeho právy, nazývaný také jako správce hry, tak může přes připravené rozhraní do hry přidávat nové úkoly, trasy či vypisovat nové závody. Dále pak má na starost i správu uživatelů, kde může uživateli odstavit herní účet, například za špatné chování.

Aplikace obsahuje i helpdesk, což je rozhraní mezi uživateli a správcem hry, které slouží jako uživatelská podpora v případě problémů. Uživatel zde může najít otázku na svůj dotaz, případně oslovit přímo administrátora nebo správce hry.

## 6.2 Model případů užití

Podívejme se nyní na popis systému z formálního pohledu - v podobě modelu případů užití. Tento model byl vytvořen v rámci předmětu softwarové inženýrství.

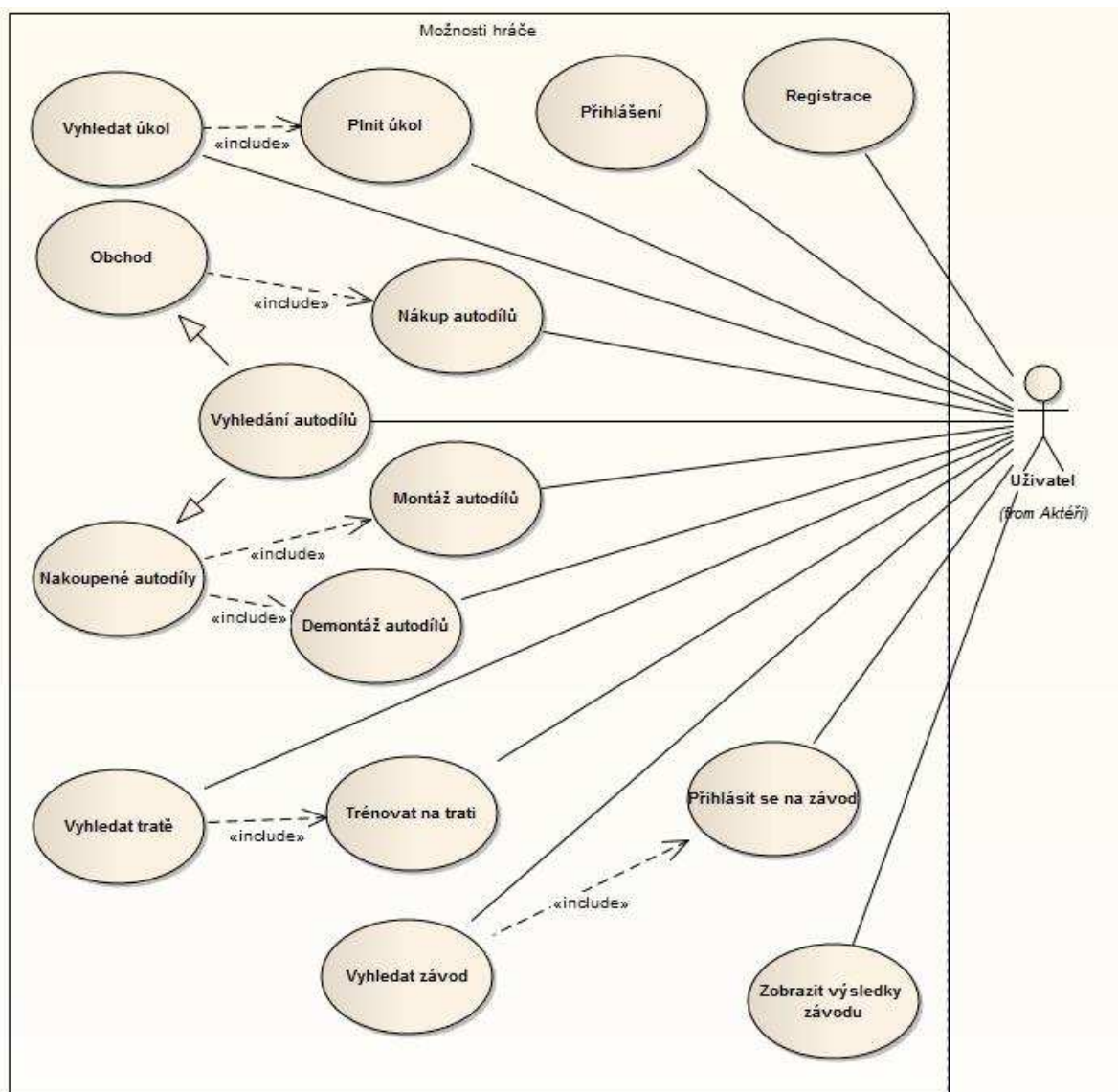
Nejprve představím jednotlivé aktéry a jejich role v systému.



Obrázek 4: Aktéři v modelu případů užití

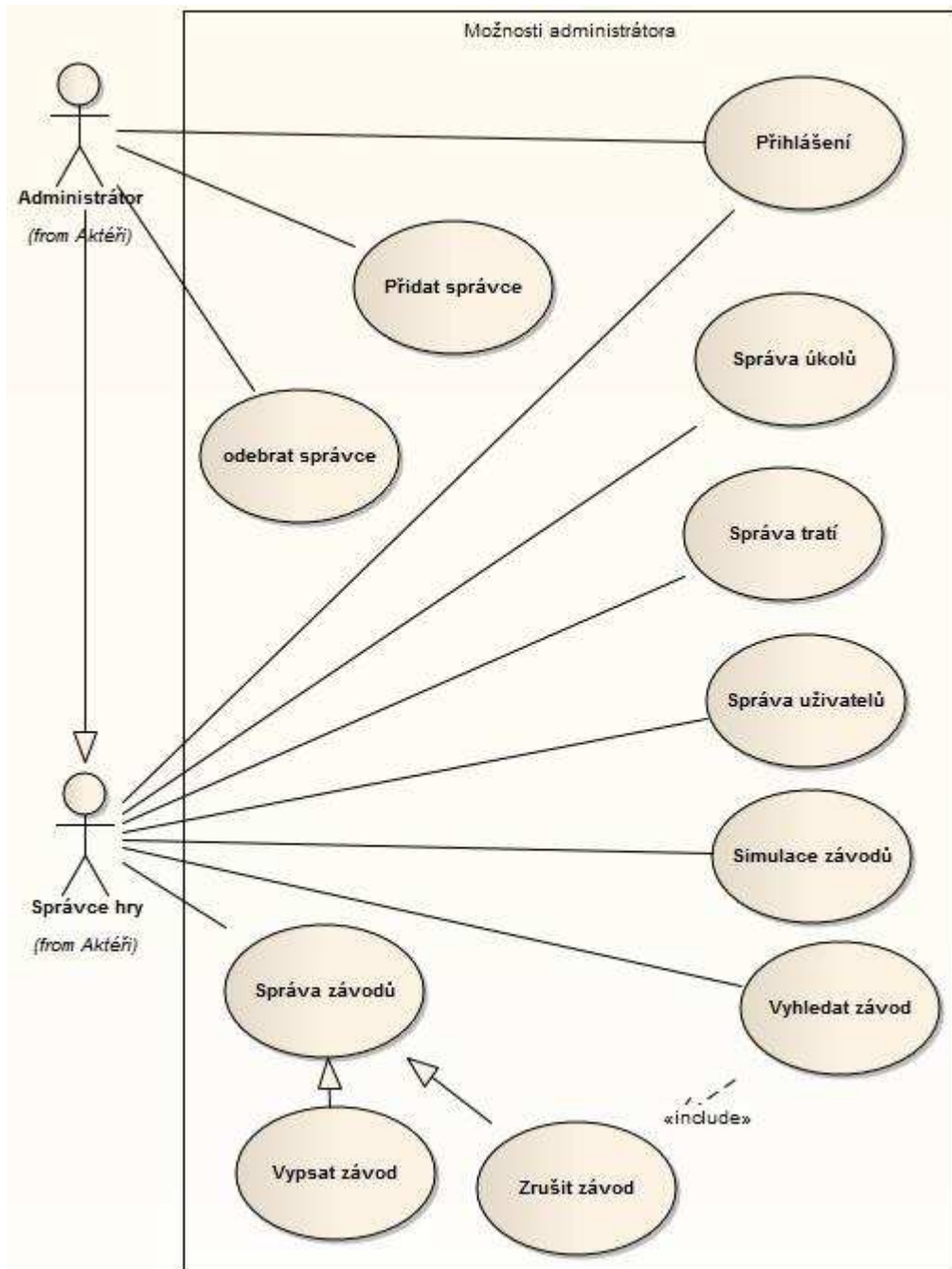
Na obrázku 4 vidíme, že v systému se vyskytují tři aktéři, kterými jsou administrátor, správce hry a uživatel.

- Uživatel je běžný hráč, který se do systému registruje, a poté může využívat jeho herní funkce.
- Administrátor je správce celého systému, který může herní systém jakkoliv měnit. Zároveň může udělovat práva hráčům, a tak z nich udělat správce hry.
- Správce hry je svým způsobem pomocník administrátora, který se přes určité rozhraní stará o průběh hry a poskytuje hráčům určitou podporu.

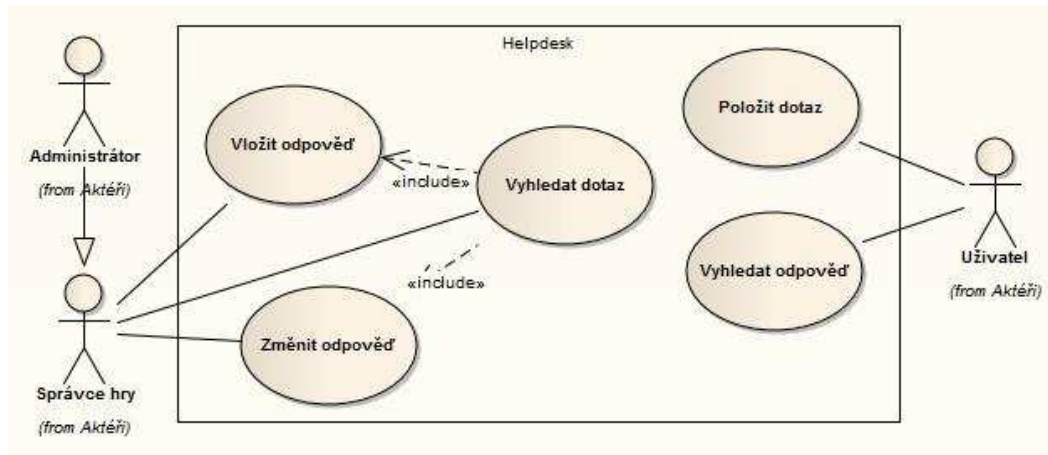


Obrázek 5: Model případů užití - hráč

Obrázek 5 představuje model případů užití z pohledu hráče. Ukazuje možnosti, které hráč smí v systému provádět. Oproti tomu obrázek 6 řeší celou situaci z pohledu administrátora a správce hry. Obrázek 7 pak zobrazuje pouze helpdesk. Ten jsem schválně oddělil, protože jde o část, kterou používají obě strany, každá však jiným způsobem.



Obrázek 6: Model případů užití - Administrátor



Obrázek 7: Model případů užití - helpdesk

## 7 NÁVRH SYSTÉMU

### 7.1 Návrh databázového modelu

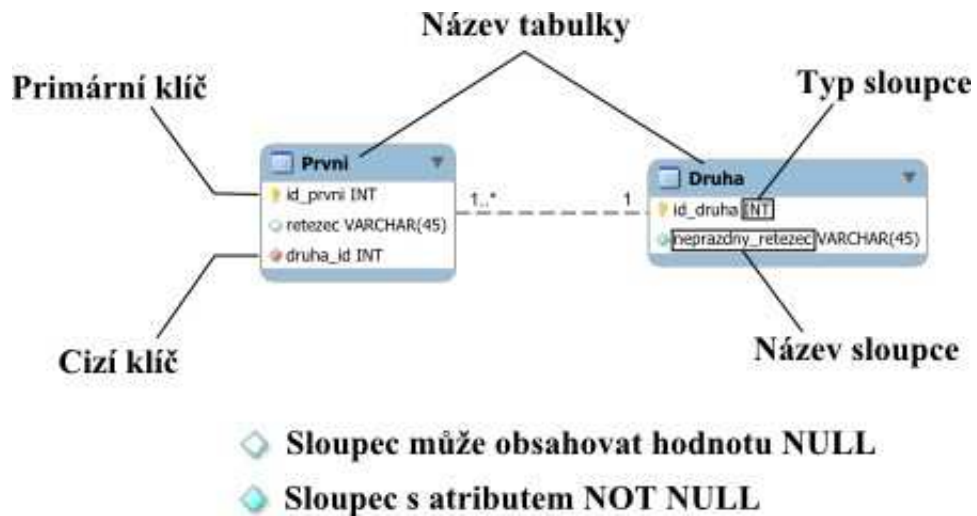
Při vývoji databázového systému, je návrh databázového modelu nedílnou součástí. Správný databázový model hned při návrhu nám v budoucnu ušetří dosti práce při implementaci. Nevyhnul jsem však tomu, že by prvotní návrh databáze zůstal tak jak je i během implementace. Někdy jsem právě až při psaní kódu narazil na problém, díky kterému jsem musel databázový model pozměnit. Během vývoje se tak původní model značně změnil.

Jak jsem zjistil, důležité je již od počátku dbát na názvech sloupců a vnést do nich určitý řád, protože upravovat několik skriptů, kvůli drobné změně v názvu sloupce tabulky, je značně pracné a neefektivní. Názvy tabulek i jejich sloupců jsem snažil volit výstižně, aby již z názvu bylo na první pohled zřejmé k čemu tabulka či sloupec slouží. Veškeré názvy jsou psány anglicky, abych omezil problémy s diakritikou v českém jazyce. Názvy jsou také psány pouze malými písmeny. Windows při psaní MySQL dotazů sice nerozlišuje velká či malá písmena v názvech tabulek a sloupců. A i když byl systém vyvíjen pod Windows, je nutné myslet na to že, většina webových serverů je založena na Linuxu. A právě Linux je k MySQL dotazům case-sensitive. Používáním pouze malých písmen se vyhneme nejasnostem při psaní kódu, ale právě i případným komplikacím při přechodu na jiný operační systém.

Jelikož většina tabulek obsahuje pro zajištění jedinečnosti sloupec obsahující pouze identifikátor daného záznamu (ID), přišlo mi vhodné zavést určitou sémantiku i pro názvy těchto sloupců. Jelikož se identifikátory používají i jako odkazy do jiných tabulek, bylo by při větších dotazech dosti nepřehledné, pokud by se sloupec s identifikátorem jmenoval ve všech tabulkách stejně, například typicky „id“. Zvolil jsem proto takové názvy, kdy název sloupce určující identifikátor začíná prefixem „id\_“, který následuje názvem tabulky. Pro název sloupce odkazující do jiné tabulky pak používám opačný název, který začíná názvem tabulky kam sloupec odkazuje a obsahuje koncovku „\_id“, aby bylo zřejmé že se jedná o identifikátor.

Jak již bylo řečeno v předchozí kapitole, k návrhu databáze jsem použil program MySQL WorkBanch. Dříve než se podíváme na databázový model celého systému, uvedu

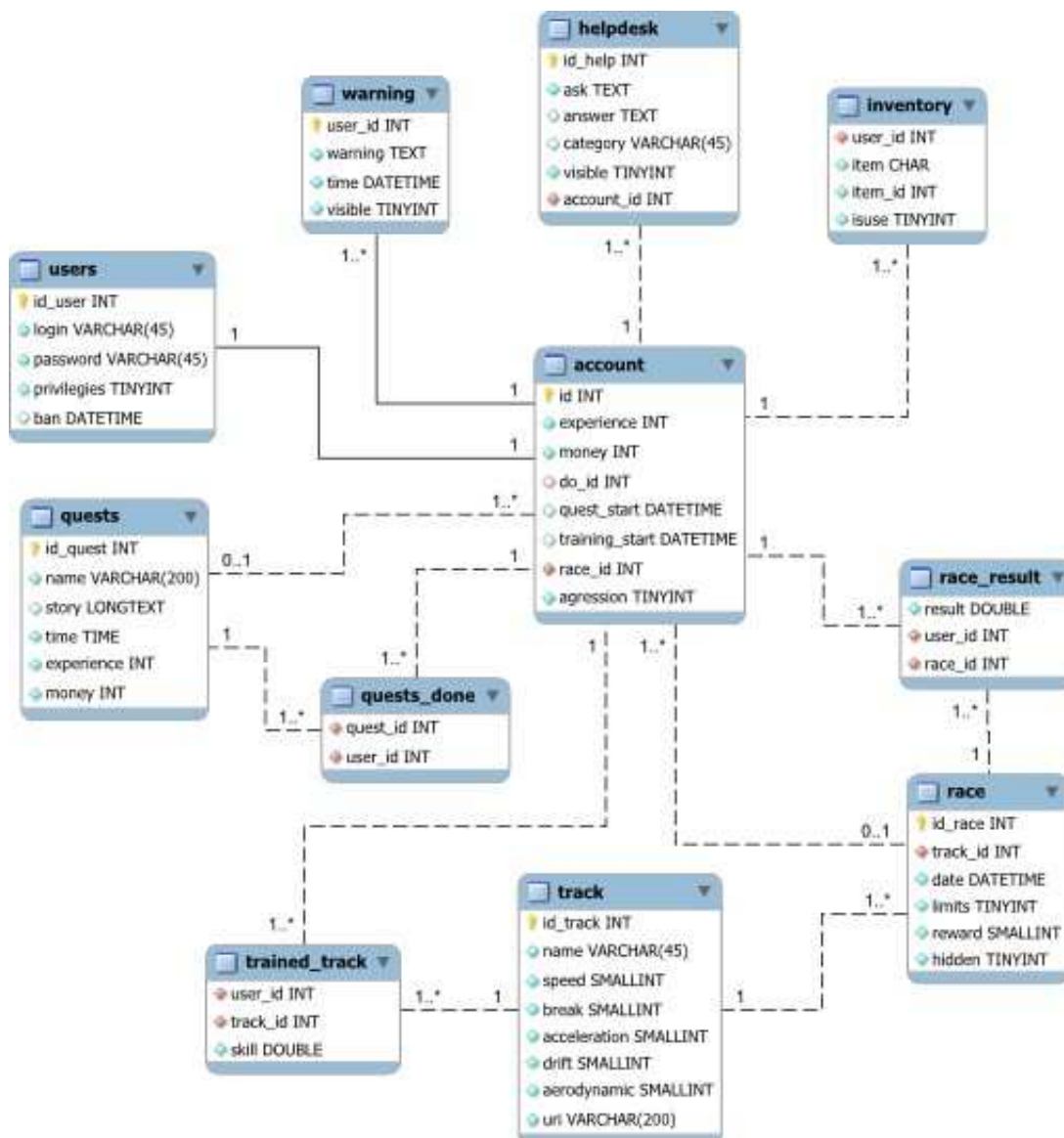
ukázkou, jakým způsobem MySQL WorkBanch zobrazuje jednotlivé tabulky. To nám pomůže více porozumět celému návrhu.



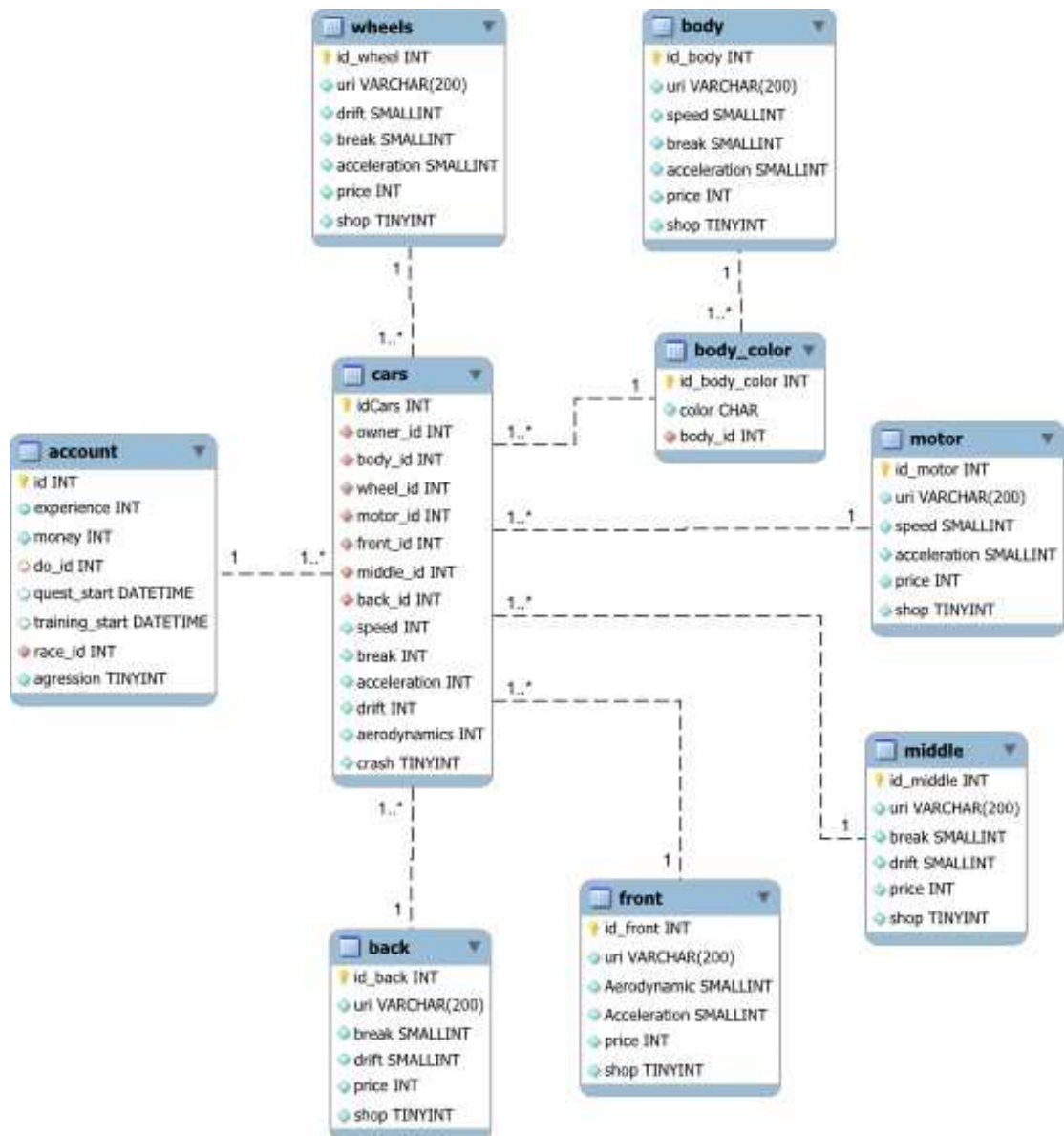
Obrázek 8: Ukázka návrhu databáze v programu MySQL WorkBanch

V příkladu vidíme, že tabulka je zobrazena se svým názvem, jednotlivými sloupci i jejich datovým typem. Podle ikony u názvu sloupce poznáme, zda se jedná o primární či cizí klíč. Ikona modrého čtverce značí obyčejný sloupec v tabulce. Červený čtverec znázorňuje cizí klíč. Tyto čtverce mohou být ještě vyplněny barvou pokud se jedná o sloupec s nastaveným atributem NOT NULL. Ikona primárního klíče má pak tvar klíče, jak je z obrázku patrné.

Na následující stránce se můžeme podívat na databázový model již našeho systému. Z důvodů rozsáhlosti je rozdělen do dvou částí. První část ukazuje tabulky, které mají spojitost přímo s uživatelským účtem (tabulka `account`, kterou můžeme vidět uprostřed). Druhá část pak také vychází z tabulky `account`, ale představuje jakoby samostatnou větev, týkající se automobilů.



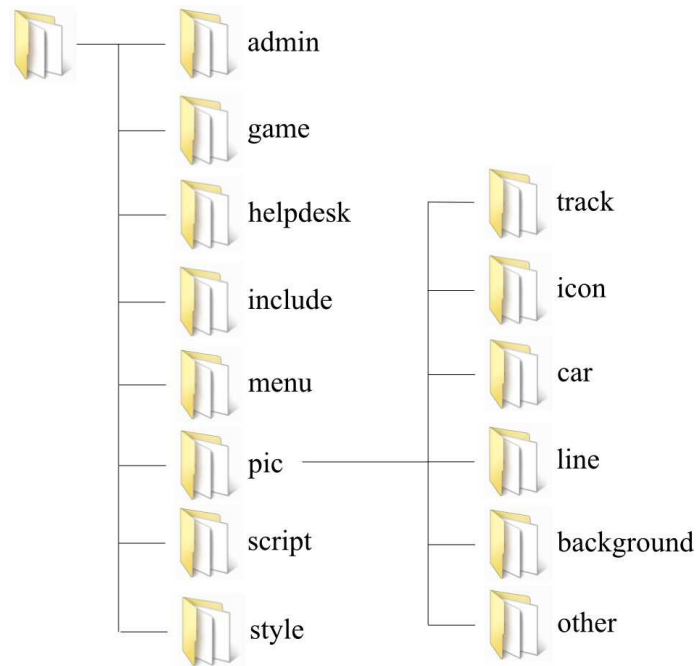
Obrázek 9: Návrh databázového systému - 1. část



Obrázek 10: Návrh databázového systému - 2. část

## 7.2 Struktura webu

Promyslet předem návrh databáze není to jediné nad čím se musíme před samotným programováním zamyslet. I přehledná struktura celého webu, kdy jsou kaskádové styly pro vzhled aplikace umístěny ve vlastní složce, skripty pro javascript ve složce druhé a v další složce například obrázky, je velice důležitá. Umožní to snadnou orientaci v celém projektu nejen pro samotného programátora, ale i pro někoho, kdo by se například později chtěl podílet na údržbě či rozvoji systému. Adresářovou strukturu si můžeme prohlédnout na obrázku číslo 11.



Obrázek 11: Adresářová struktura systému

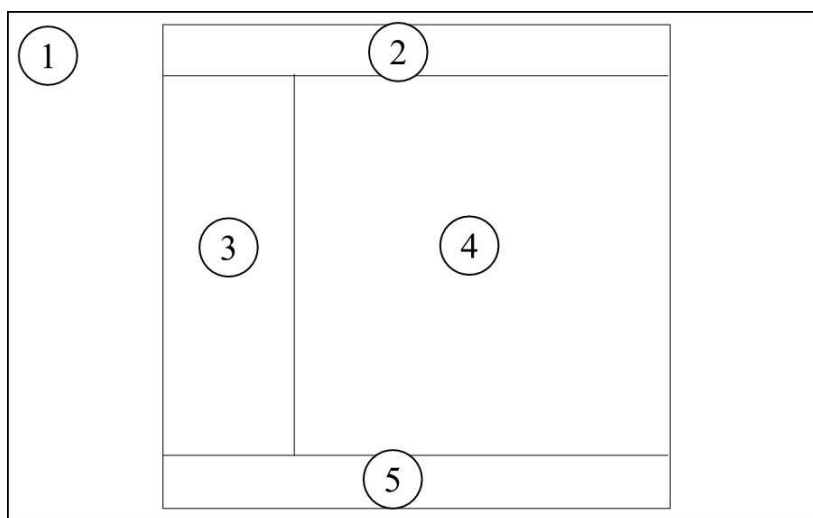
Přestože z názvů složek přímo vyplývá, co složka obsahuje, stručně každou z nich popíšeme. Kořenová složka mimo další složky obsahuje i soubor *index.php*, který slouží jako startovací stránka celého systému. Odtud se dále odkazujeme na stránky jiné. Každá další stránka je pak tvořena samostatným souborem, obvykle s příponou *php*. Proto vedle indexu nalezneme i soubory *prihlaseni.php*, *registrace.php*, *logout.php* a *install.php*. Dále už nebudu vyjmenovávat všechny soubory, ale jen popíši, co která složka obsahuje.

- admin - v této složce jsou obsaženy soubory, ke kterým se přistupuje z administrátorské části. Každý soubor je skriptem, který řeší správu jedné věci (například soubor pro správu závodů, tratí, atd.).
- game - obdobná jako složka admin, s tím rozdílem, že obsahuje soubory pro hráčskou část celé aplikace.
- helpdesk - soubory realizující část helpdesku ze strany hráče. Odděleny od složky game jsou z důvodů přehlednosti.
- include - soubory s definovanými funkcemi či konfiguračními soubory. Obsahuje například soubor *connect.php*, který obsahuje přihlašovací údaje do databáze. Při zavedení systému na jiný server pak stačí změnit údaje pro přihlášení do databáze pouze na jednom místě.

- menu - složka obsahuje jednotlivá menu, která jsou poté vkládány do stránek. Obsahuje menu pro nepřihlášeného uživatele, pro přihlášeného uživatele nebo administrátora. Takto oddělené menu nám opět poskytuje flexibilitu při případných změnách.
- pic - tato složka obsahuje veškeré obrázky, se kterými systém pracuje. Složka obsahuje další podadresáře v závislosti na povaze obrázku. Obrázky jsou pak přehledně rozděleny podle toho zda se jedná například o pozadí, automobil nebo ikonu.
- script - obsahuje javascriptové soubory (\*.js).
- style - obsahuje soubory s kaskádovými styly, které upravují vzhled aplikace (\*.css).

### 7.3 Rozvržení stránky

Podívejme se nyní na základní layout celé jedné stránky.

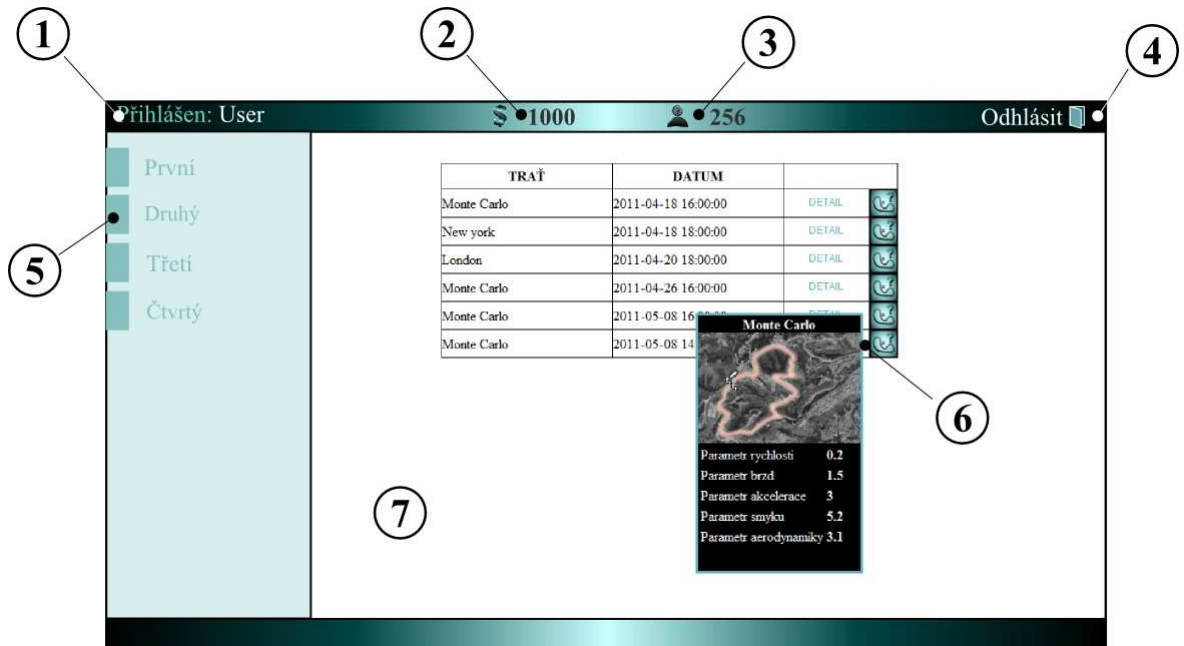


Obrázek 12: Rozvržení stránky

Obrázek 12 znázorňuje základní rozvržení stránky. Bod (1) znázorňuje celé okno obrazovky. Bod (2) je hlavičkou dokumentu, která obsahuje hlavně informace o uživateli. Body (3) znázorňuje prostor pro menu, kterým se aplikace ovládá. (4) Hlavní rámec dokumentu, kde se vykresluje obsah stránky podle toho, kde se uživatel právě nachází. Bod (5) vytyčuje prostor pro další informace, kterými může být například copyright.

## 7.4 Uživatelské rozhraní

Ukažme si nyní uživatelské rozhraní celé aplikace. Zvolil jsem klasické rozložení stránky, s ovládacím menu umístěným vlevo. Takový "layout" je obvyklý u mnoha webových stránek, proto uživatel nebude mít problém s celkovou orientací v aplikaci. Snažil jsem se, aby aplikace byla jednoduchá a přehledná. Na následujícím obrázku vidíme přibližný vzhled celé aplikace.



Obrázek 13: Uživatelské rozhraní systému

Stručně si popišme základní prvky, které uživatel uvidí na obrazovce.

- (1) Přezdívka aktuálně přihlášeného uživatele.
- (2) Výše hráčova herního kreditu.
- (3) Zkušenostní body hráče.
- (4) Odhlášení ze systému.
- (5) Jednotlivé odkazy menu (neměnné).
- (6) Rychlé informace (viz. kapitola 7.4.1)
- (7) Obsah stránky

### 7.4.1 Rychlé informace

Při práci se systémem, může uživatel narazit na jednu ze tří ikon (viz. *Obrázek 14*). Tyto ikony jsem si nazval jako "ikony rychlé informace".



*Obrázek 14: Ikony pro rychlou informaci*

Objevují se převážně u tabulek, kde slouží jako informace pro určitý řádek. Pro lepší pochopení bude lepší uvést příklad. Mějme například tabulku s vypsáním závodů. Každý závod je vypsán na konkrétní trati. U každého řádku tak může být ikona, pro rychlou informaci o trati. A proč říkám rychlou? Uživateli stačí na ikonu pouze najet ukazatelem myši, a vedle ikony se hned objeví patřičné údaje. Jak si můžeme všimnout, jsou tři různé ikony. Obrázek napovídá, čeho se bude informace týkat. První z nich se týká automobilu, druhá uživatele a třetí tratě.

Tyhle údaje, jsou ze serveru získávány pomocí techniky AJAX. Ta slouží k asynchronní komunikaci se serverem, což nám umožní získat data například z databáze, bez nutnosti obnovení stránky. Takovou komunikaci se serverem nám zajišťuje javascript.

Dříve nebyl AJAX příliš jednoduchou záležitostí. To se však změnilo s příchodem již zmiňované javascriptové knihovny jQuery. Díky ní je asynchronní komunikace se serverem otázkou pár řádků. Následující ukázka kódu provede odeslání dat na server, a přijmutí odpovědi.

```

$.info_track).bind('mouseover',function(){
    $.get("test.php",
        { trasa: "trasa" },function(data){
            $("#track_info").html(data);
        });
});

```

Jak vidíme, nejde o nic složitého, avšak celou aplikaci to obohatí o větší dynamičnost. Jelikož odeslaná data zpracovává klasicky PHP, je možné tak vracet aktuální data i z databáze.

## 8 IMPLEMENTACE

V této kapitole popíši implementaci jednotlivých částí aplikace, které jsou klíčové nebo stojí za zmínku. Nejdříve popíši uživatelskou část, kde bude uvedeno, jaké možnosti má hráč. Po té bude zmíněna část administrátorská.

### 8.1 Uživatelská část

#### 8.1.1 Registrace uživatele

Registrace je první věc, se kterou se hráč setká. Pro registraci uživatel vyplní jednoduchý formulář obsahující přihlašovací jméno a heslo. Při odeslání formuláře systém zkontroluje, zda již přihlašovací jméno není v databázi obsaženo. Pokud ano, sdělí to uživateli a vyzve ho ke změně přihlašovacího jména. Pokud je vše v pořádku, systém přidá nového uživatele do databáze.

K tomu slouží tabulka **users**, která obsahuje sloupec *id\_user*, *login*, *password* a *privileges*. První sloupec má nastavený atribut AUTO INCREMENT, což znamená, že si hodnotu doplní sám databázový systém – obvykle číslo o jedničku větší než číslo posledního záznamu. Do pole *login* se uloží přihlašovací jméno, a do pole *password* zadané heslo zakódované hašovací funkcí SHA1. O hašovacích funkcích bude řeč později v kapitole o zabezpečení systému. Pole *privileges* určuje práva daného uživatele a při registraci se automaticky nastaví na hodnotu 0 (nula), což udává že se jedná o běžného uživatele. Další možností jsou hodnoty 1 pro správce hry nebo 2 pro administrátora.

Tímto vznikl nový uživatel, nevznikl však uživatelský účet. V databázovém modelu můžeme vidět, že přihlašovací údaje jsou od herních údajů odděleny. Herní údaje jako například výše kreditu či zkušeností jsou v samostatné tabulce s názvem **accounts** a s tabulkou **users** jsou spojeny vazbou 1:1, kde společnou hodnotou je sloupec s identifikátorem. Takto rozdělené data jsou jednak z důvodu přehlednosti, ale také pro oddělení režie pro přihlašování a samotným hraním.

Nový záznam v tabulce **accounts** se vytvoří automaticky při vložení nového záznamu do tabulky **users**. Jak jsem již psal, aby bylo možné rozeznat, které údaje patří k sobě, mají obě tabulky stejný identifikátor. To vše nám zajistí trigger u tabulky **user**.

### 8.1.2 Přihlášení uživatele

Jakmile je uživatel zaregistrovaný, může se do hry přihlásit. K tomu opět slouží formulář, kam uživatel zadá své přihlašovací jméno a heslo. Zadané heslo je převedeno na SHA1 hash, aby ho bylo možné porovnat s hodnotou uloženou v databázi. Proveďte se kontrola, zda zadané přihlašovací jméno a heslo patří opravdu k sobě. Pokud tomu tak není, uživatel je informován a požádán o vyplnění přihlašovacích údajů znovu. Pokud však přihlášení proběhlo v pořádku, ocitne se uživatel na úvodní stránce svého herního účtu.

Nutné bylo ještě vyřešit předávání informací o přihlášeném uživateli mezi jednotlivými stránkami aplikace. To je v aplikaci řešeno pomocí session, což nám dokáže zajistit právě jazyk PHP. Session by se dala přirovnat ke globální proměnné, která platí pro všechny stránky aplikace. Práce s nimi je velice jednoduchá. Stačí pouze proměnnou vytvořit a o zbytek se již postará PHP. Session proměnná může vypadat například takto - `$_SESSION[ 'prom' ]`. Tím vytvoříme proměnnou s názvem „prom“, která je platná mezi jednotlivými stránkami, až do doby uzavření prohlížeče, nebo dokud ji sami ručně nesmažeme. Takových proměnných můžeme mít v aplikaci samozřejmě více.

Po úspěšném přihlášení ukládá aplikace do session informace, které jsou často používány. Takovými informacemi je identifikační číslo uživatele, jeho přihlašovací jméno, práva, množství zkušeností a výše jeho kreditu. Možností by bylo ukládat jen identifikační číslo, ze kterého ostatní údaje můžeme zjistit, pomocí dotazování na databázi. Jelikož to však jsou údaje, které jsou používány téměř na každé stránce, zvýšila by se tak mnohonásobně komunikace s databází, což by negativně ovlivnilo i rychlost celé aplikace.

Jakmile je uživatel přihlášen, může začít využívat možnosti, která mu hra nabízí. Ty budou popsány v následujících kapitolách.

### 8.1.3 Automobil

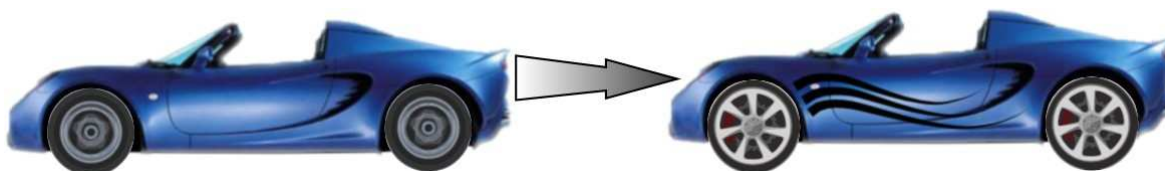
Automobil je podstatou celé hry. Hráči, který se poprvé přihlásí do systému, bude nabídnuto, aby si vybral svůj automobil. Ten si ze začátku bude moci poskládat pouze ze základních dílů, které bude v průběhu hry vylepšovat.

Tím se dostávám k implementaci automobilů, které jsou databázově řešeny tabulkou `cars`. Tabulka uchovává informace o jednotlivých automobilech. Důležité je zejména pole `owner`, které je cizím klíčem a obsahuje identifikační číslo hráče, který

automobil vlastní. Dalšími sloupci jsou právě jednotlivé součástky. To znamená *motor*, *body*, *wheel*, *front*, *middle*, *back*. Tyto sloupce obsahují odkazy do příslušných tabulek na konkrétní díl.

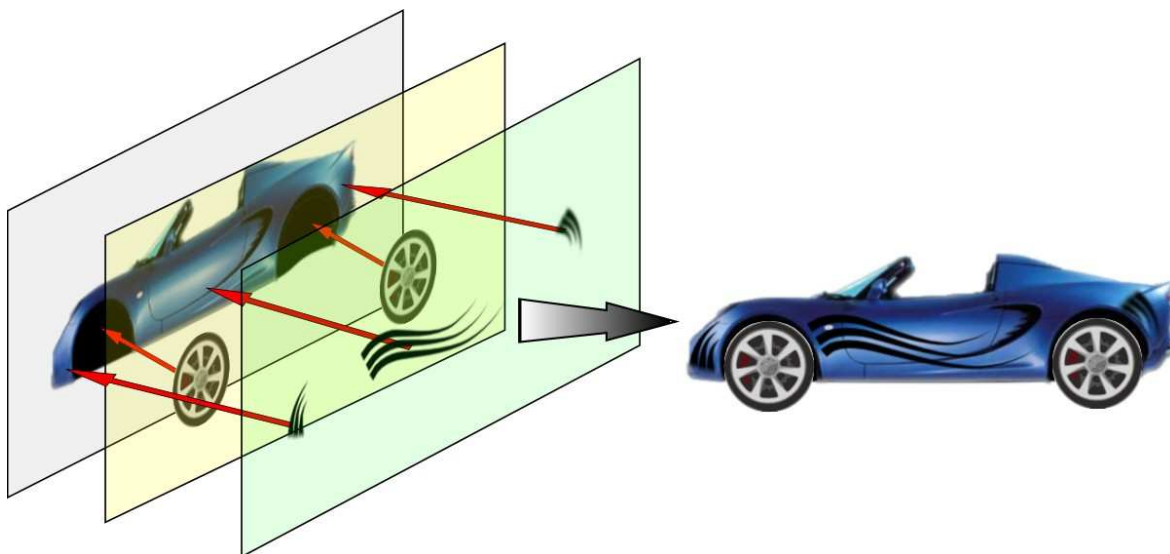
#### 8.1.4 Vykreslení automobilů

Aby měl hráč pocit, že automobil, o který pečuje opravdu prochází nějakou změnou, bylo nutno vyřešit grafické zobrazení automobilu. Jak bylo zmíněno dříve, na automobilu lze měnit motor, celá karoserie, kola a přední, střední i zadní díl karoserie. Kromě změny motoru se, všechny změny projeví na vzhledu auta. Změna celé karoserie provede změnu tvaru celého auta. Změna kol se projeví různým vzhledem disků na kole. Změny jednotlivých částí karoserie jsou pak řešeným různým grafickým motivem na určité části karoserie. Následující obrázek ukazuje změnu kol a středního dílu karoserie.



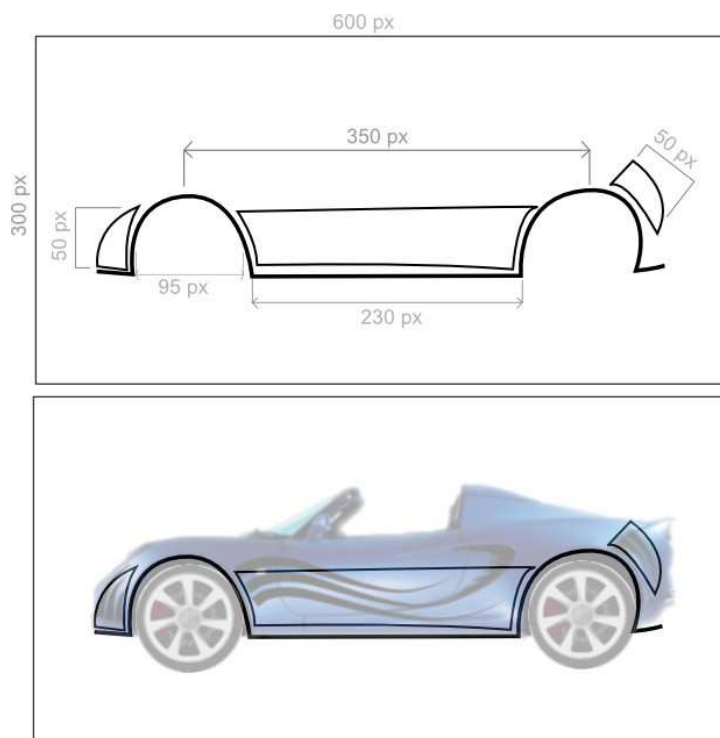
Obrázek 15: Ukázka změny autodílů na automobilu

Jelikož by mohlo být mnoho kombinací jak automobil může vypadat, není možné mít pro každou tuto kombinaci zvláštní obrázek. Proto je to řešeno vrstvením jednotlivých obrázků na sebe. Jednotlivé obrázky jsou ve formátu PNG nebo GIF. Oba tyto formáty podporují průhlednost, což nám právě dovolí skládat jednotlivé části na sebe. PNG je v tomto ohledu lepší, protože umožňuje i poloprůhlednost, což znamená, že výsledná barva je ovlivněna podkladem na kterém leží. Díky poloprůhlednosti lze pak vytvářet graficky zajímavější motivy. Nevýhodou však je velikost tohoto formátu. Formát GIF průhlednost podporuje jen formě ano nebo ne. Buď je pixel průhledný úplně, nebo vůbec. Využít by šel například na jednoduché linie s plnými barvami. Výhodou oproti formátu PNG je právě velikost obrázku, která může být několikanásobně menší. Princip vrstvení demonstruje následující obrázek.



Obrázek 16: Vykreslování automobilu v jednotlivých vrstvách

Aby výsledný obrázek tvořil podobu automobilu, musí však být splněny určité rozměry. Při jejich nedodržení by se mohlo stát, že kola budou zasahovat do karoserie nebo jakýkoliv jiný nežádoucí efekt. Proto jednotlivé části musí zapadat do následující šablony.



Obrázek 17: Šablona pro nové automobily

Ze šablony je patrné, kde musí mít celá karosérie prostor pro kola. Do toho musí pak i zapadat velikost a rozteč kol. Zároveň i místo pro grafické motivy by mělo být

dodrženo, i když například u prostřední části existuje určitá tolerance. Pokud budou jakékoliv nové části zapadat do šablony, tak jak můžeme pozorovat na dolním obrázku (viz. *Obrázek 17*), nemělo by se nám stát, že by některá kombinace částí neseděla.

Na závěr této podkapitoly se podívejme ještě na to, jak vypadá kód, který nám zajistí poskládání obrázků na sebe.

```
<div id="celek">
    <div class="vrstva"></div>
    <div class="vrstva"></div>
    <div class="vrstva"></div>
    <div class="vrstva"></div>
    <div class="vrstva"></div>
</div>
```

Pro názornost jsem v kódu použil názvy obrázků tak, aby bylo zřejmé o co se jedná. Jednotlivé vrstvy jsou uzavřeny do jednoho bloku, který má pomocí kaskádových stylů nastavenou šířku, výšku a absolutní či relativní pozicování. Originální rozměry celého bloku jsou 600px na šířku a 300px na výšku. Každý obrázek je pak ještě umístěn ve vlastním bloku. Každý tento blok má nastavenou šířku i výšku na 100%, která se váže k velikosti nadřazeného bloku. Dále má pak každý blok absolutní pozici, která se taktéž váže k rodičovskému bloku. Umístění od horní a levé hrany je nastaveno na 0, čímž docílíme, že se vrstvy naskládají na sebe. Samotné obrázky mají rozměry nastaveny také na 100%.

Takovou implementací nejen že dosáhneme požadovaného efektu, ale umožní nám jednoduchým způsobem pracovat s celkovou velikostí automobilu. Zmenšíme-li rozměr celého bloku, nejlépe aby byl zachován poměr stran 2:1, automaticky se zmenší všechny vrstvy a tím i auto, aniž by se rozházeli jednotlivé části.

### 8.1.5 Nákup autodílů

Základními věcmi, ze kterých se v našem případě automobil skládá, jsou kola, motor a celá karoserie. Základní modely těchto dílů, dostane hráč při vytvoření herního účtu, jak již bylo psáno dříve. Nákupem a montáží nových autodílů, může hráč zlepšit vlastnosti svého auta. Pro nákup nových dílů nám slouží obchod, kde hráč jednotlivé díly nakupuje za herní kredit. O tom jak tento kredit získat, se dočteme později.

Než se podíváme na implementaci samotného nakupování, ukažme si jakým způsobem jsou v databázi řešeny jednotlivé díly. Pro každý díl je vytvořena zvláštní tabulka. Pro motory je to tabulka **motors**, pro karoserie tabulka **body**, a tak dále. Celkem máme pro autodíly šest tabulek, kde kromě dvou již zmíněných nalezneme tabulky **wheel**, **front**, **middle** a **back**. Každá z nich obsahuje sloupce s jednoznačným identifikátorem, cenou a dále některé parametry, které ovlivňují právě vlastnosti auta. Každá tabulka dílu obsahuje ještě sloupec *shop*. Jde o atribut, který určuje zda je, nebo není daná součástka viditelná v obchodě.

Samotný obchod je pak tvořen tak, že systém projde všech šest tabulek a zobrazí uživateli všechny součástky s nastaveným atributem *shop* na hodnotu 1. Hráč si pak vybírá ze seznamu dostupných součástek. Pokud se rozhodne nějakou koupit, systém nejdříve ověří, zda má uživatel dostatek kreditu ke koupi. Pokud ano, provede se změna odečtením kreditu v tabulce herního účtu uživatele. Zároveň se příslušná věc vloží do uživatelova inventáře.

### 8.1.6 Instalace autodílů

Zde naváží na předchozí kapitolu, a to inventářem. Inventář je pro hráče seznam věcí, které vlastní a získal je buď zakoupením nebo odměnou. Z hlediska databáze je inventář tvořen jednou tabulkou, společnou pro všechny uživatele. Tabulka má název **inventory** a obsahuje následující sloupce. Identifikační číslo uživatele *user\_id*, které určuje, jakému hráči patří daná součástka. Na základě tohoto pole pak můžeme zjistit veškeré autodíly, které hráč vlastní. Dalšími sloupci jsou *item\_id* a *item*. Položka *item* určuje, o jaký typ součástky jde. To znamená, že jakoby ukazuje na jednu ze šesti tabulek, o kterých se psalo dříve. Sloupec *item\_id* pak obsahuje identifikátor konkrétního dílu v tabulce. Poslední sloupec se nazývá *isuse* a určuje pouze, zda je daný díl namontován nebo ne.

V aplikaci je možnost zobrazení inventáře a úpravy automobilu pod odkazem garáž. Systém z tabulky vybere pouze věci, patřící danému uživateli, které mu zobrazí. Aplikační logika zároveň přidá k jednotlivým dílům možnosti pro jejich montáž či demontáž. Při montáži se provede změna nejen u pole *isuse* v tabulce **inventory**, ale provede se také změna v tabulce **cars**.

### 8.1.7 Plnění úkolů

Podívejme se nyní na možnosti, jak si na vylepšení auta vydělat. První a nejjednodušší způsob jak si navýšit zkušenosti a kredit, je plnění úkolů. Uživatel si může vybírat z celé řady úkolů. U každého úkolu si může přečíst příběh, kde se dozví, o co v úkolu jde. Samozřejmě to je jen pro navození atmosféry při hraní, protože jak již bylo předesíláno dříve, nejedná se o hru, kde by hráč cokoliv ovládal v reálném čase. Mimo příběhu si hráč může zjistit i čas, jak dlouho splnění úkolu potrvá a také množství kreditu a zkušeností, které za splnění dostane.

Jakmile si hráč vybere vhodný úkol, zvolí možnost pro jeho plnění. V době, kdy hráč plní úkol má omezené některé možnosti hry. Průběh plnění může hráč sledovat. Zde uvidí, který úkol zrovna plní a další informace, včetně času, za jak dlouho bude úkol dokončen. Po splnění úkolu získá hráč patřičné zkušenosti a kredit. Jakmile hráč splní úkol, nemůže ho plnit znovu. Hráč může plnění úkolu kdykoliv zrušit. Nedostane však patřičnou odměnu, ale může úkol začít plnit od znovu.

Podívejme se na plnění úkolů z pohledu implementace. Veškeré úkoly jsou vedeny v tabulce s názvem „**quests**“. Pro každý údaj, který byl uveden dříve, je v tabulce zvláštní sloupec. Při zobrazování informací o úkolu se použijí informace právě z této tabulky. Při zahájení plnění se však využívá i tabulky „**accounts**“, která obsahuje i sloupce *do\_id* a *quest\_start*. Jakmile uživatel zahájí plnění úkolu, uloží se do těchto sloupců identifikační číslo daného úkolu a čas, ve který bylo plnění zahájeno. To nám umožní později vypočítat i dobu, ve kterou má zadaný úkol skončit. Jelikož se údaje o čase ukládají do databáze, může hráč při plnění úkolu aplikaci opustit, aniž by tím zadaný úkol ovlivnil.

Aby hráč nemohl plnit některý úkol vícekrát, ukládají se i údaje o tom, který hráč splnil jaký úkol. Jelikož jde o vazbu N:M mezi úkoly a hráči, musím si pomoci další tabulkou. Jde o tabulku **quest\_done** a obsahuje pouze sloupce *user\_id* a *quest\_id*, do kterých se zapisuje identifikační číslo uživatele a úkolu, který splnil. Na základě této tabulky již systém hráči nenabídne dané úkoly ke splnění.

Při dokončení úkolu se provede zápis do tabulky **quest\_done** a zároveň se provede změna v tabulce **accounts**, kde se uživateli na herní účet přičte patřičná odměna.

### 8.1.8 Závody

Další možností jak zbohatnout je účast na závodech. Jednotlivé závody vypisuje správce nebo administrátor hry. Z pohledu databáze nám pro závody slouží tabulka **race**, a pro ukládání výsledků závodů tabulka **race\_result**. Závody probíhají v určitý čas, který zvolí správce hry. K tomu nám v tabulce slouží sloupec *date*. Mimo to je nutné ještě nastavit maximální počet hráčů, kteří se mohou na závod přihlásit a trať, na které se závod pojede. K tomu slouží pole *limit* a *track\_id*. Jelikož jsem psal, že jde o další způsob jak získat kredit, nesmí chybět sloupec *reward*, který uchovává informaci o případné výhře. Samozřejmě nesmí chybět ani jednoznačné označení závodu, které je ve sloupci *race\_id*. Právě toto číslo se uloží do tabulky **account**, pokud se některý hráč na závod přihlásí. Z toho zároveň vyplývá, že hráč může být v jeden okamžiku přihlášen pouze na jeden závod.

Vyhodnocení závodu proběhne automaticky, ve stanovenou dobu závodu, na základě několika faktorů. Při vyhodnocování se projde pole uživatelů, kteří jsou na závod přihlášení, a každému se vypočítá určité hodnocení, které je poté zapsáno do tabulky **race\_result**. Hráč z nejvyšším hodnocením pak získá patřičnou odměnu.

Podívejme se nyní na to, co ovlivňuje výsledek závodu. Především jsou to parametry hráčova automobilu. Stejně parametry totiž obsahují i jednotlivé tratě, lépe řečeno jde o koeficienty k těmto parametrům, kterými se násobí jednotlivé vlastnosti hráčova auta. Na základě těchto koeficientů pak lze vytvářet tratě, pro různé druhy vozů. Některé tratě tak můžou zvýhodňovat hráče, kteří se zaměřili na vylepšování rychlosti, jiné zas takové hráče, kde dominuje akcelerace či aerodynamika.

Aby však nebylo vyhodnocení příliš statické, existují možnosti, kterými hráč může ovlivnit výsledek závodu. První z nich je styl jízdy.

Styl jízdy může hráč zvolit před přihlášením na závod. Styl může volit z hodnot od nuly do deseti, kde nula znamená opatrnou jízdu a desítka naopak jízdu agresivní. Zvolená hodnota se uloží do pole *agression*, tabulky **account**. Agresivní jízda pak dává hráči větší šanci na úspěch. Pokud ovšem závod vůbec dokončí, protože mimo šanci na úspěch zároveň zvyšuje i riziko havárie. V případě havárie pak hráč obdrží z celkového hodnocení nulu a k tomu i poškozený automobil, který je nucen za peníze opravit.

Druhou možností, jak ovlivnit výsledek závodu je trénink, o kterém bude psáno v následující, samostatné kapitole.

### 8.1.9 Trénink

Trénink je další věcí, kterou může uživatel při hraní provádět. Účelem trénování je zlepšit své schopnosti na určité trati. Pokud hráč zvolí možnost tréninku, uvidí jednotlivé tratě, a svoji znalost každé z nich. Znalost je uváděna procentuálně, v rozmezí od nuly do sta procent, a tato hodnota je uložena v tabulce **trained\_track**.

Pokud chce hráč zlepšit znalost některé z tratí, zvolí možnost tréninku u požadované trati. Do tabulky **accounts** je poté uložen čas, kdy trénink započal, a číslo trati, na které trénink probíhá. Během tréninku má však uživatel omezené herní možnosti, jako je například plnění úkolů.

Trénink může hráč kdykoliv zase přerušit a za dobu, po kterou trénoval, dostane patřičné procenta zkušeností.

Znalost jednotlivých tratí je pak další faktor, který ovlivňuje výsledek závodů. Jednak snižuje možnost havárie při zvolené rychlé jízdě, a za druhé znalost tratě přispívá i bonusovými body při vyhodnocení závodu.

Herní možnosti jsem tímto vyčerpali, a proto se podívejme, kde může hráč hledat podporu v případě potíží. Takovým místem je helpdesk.

## 8.2 Helpdesk

Účelem helpdesku je poskytnout podporu uživatelům. Uživatel by měl mít možnost, přes nějaké rozhraní, komunikovat v případě potíží se správcem hry. V našem případě je toto rozhraní samozřejmě také webové, jak tomu většinou bývá.

### 8.2.1 FAQ

Tato zkratka je z anglického *Frequently Asked Questions*, a v překladu znamená „často kladené dotazy“. Jde o seznam otázek a odpovědí, na které se hráči nejčastěji ptají. Pokud hráč vstoupí do helpdesku, první co uvidí budou právě tyto otázky a odpovědi. Však ne vždy v nich uživatel najde odpověď na to, co by potřeboval znát či vědět. Proto samozřejmě má možnost vložit svůj vlastní dotaz, a tím se opět dostáváme k databázím.

### 8.2.2 Vlastní dotaz

Helpdesk je v aplikaci řešen pouze jednou tabulkou. Důležitými poli jsou sloupce pro otázku a odpověď. Pro vložení nového dotazu slouží uživateli formulář, kam pouze zadany dotaz napíše a odešle. Dotaz se poté vloží do tabulky, společně s identifikačním číslem uživatele, pro které je v tabulce zvláštní sloupec. Sloupec pro odpověď zatím zůstává prázdný a má hodnotu NULL.

Na řadu se teď dostává správce hry, který má možnost nechat si zobrazit všechny otázky, na které ještě nebylo zodpovězeno. Pokud takové otázky jsou, jednu z nich si vybere a pomocí formuláře vloží odpověď. Zároveň má možnost poupravit samotnou otázku, pro případ upravení gramatiky, protože otázka s odpovědí se posléze stane veřejnou. Než potvrdí svou odpověď, zvolí ještě kategorie do které otázka spadá. Pro kategorii je v tabulce helpdesk připraven taky jeden sloupec.

Pokud správce hry na otázku odpověděl, provede se změna záznamu v tabulce. Hráč přihlášený pod svým účtem nyní tuto odpověď uvidí mezi svými dotazy, společně s jeho jinými dotazy, na které již bylo odpovězeno. Zároveň se však dotaz stává veřejným a mohou si ho zobrazit i jiní hráči.

### 8.2.3 Dotazy ostatních uživatelů

Jak bylo popsáno dříve, dotaz, který uživatel podá, je po zodpovězení veřejný i ostatním hráčům. Měl by proto hráč před vložení vlastního dotazu nejdříve zavítat do této sekce, kde jsou jednotlivé dotazy tříděny právě podle dříve zmíněných kategorií. Je totiž možné, že právě zde najde odpověď na svou otázku a ušetří si čas čekáním na odpověď od správce hry.

## 8.3 Administrace systému

Mimo hráčskou část bylo nutné ještě vytvořit jisté rozhraní pro administraci a správu celého systému. K tomu by samozřejmě mohl být použit i nástroj adminer, který byl použit pro práci s databázemi. Při rozsáhlém databázovém systému by však práce s ním, mohla být dosti složitá a nepřehledná, protože by jsme se museli prodírat i spoustou údajů, které nejsou potřeba spravovat. Zároveň nechceme, aby i případný správce hry, měl možnost měnit jakoukoliv část databáze.

Pro správce hry proto vytvořeno prostředí, pomocí kterého může omezeným způsobem udržovat hru v chodu. Podívejme se nyní, jaké má správce hry možnosti.

### 8.3.1 Možnosti správce

Jednou možností správce hry, je starat se o úkoly. Přes připravené formuláře je možné přidávat, mazat či upravovat jednotlivé úkoly, které jsou dále zobrazovány hráčům. Přidávat stále nové úkoly je ve hře nutností, aby se nenastala situace, že bude hráč mít již všechny úkoly splněné, což by pak hráče ochudilo o možnost jistého výdělku kreditů.

Další možností je správa tratí. Podobně jako v předchozím případě, umožňuje správci hry přidávat, mazat či upravovat jednotlivé tratě, na kterých je možné poté vypisovat různé závody. Stejným způsobem je pak možná i správa jednotlivých závodů.

Jedná se o takové části hry, které je nutné stále udržovat aktuální, aby se hráč najednou nedostal do bodu, že nemá co dělat.

### 8.3.2 Možnosti administrátora

Administrátorovi přibývají ještě některé možnosti, které správce hry nemá. Tou je například správa uživatelských účtů. Zde má administrátor možnost vyhledat jakéhokoliv hráče. První možnost je pak hráče smazat, například za opakované porušování pravidel. Jiná možnost, méně přísnější, je odstavení herního účtu na určitou dobu. Opět to může být jako trest pro hráče za porušení pravidel a varování, aby se to příště již neopakovalo. Poslední možností je, změnit uživateli práva, a tím z něj udělat správce hry.

### 8.3.3 Simulátor závodů

Simulátor závodů je pomocný prvek, který pomáhá při správě systému. Zejména při tvorbě tratí. Správce hry díky připravenému rozhraní, může simulovat závod určitých automobilu na kterékoliv trati. Správce hry vyplní jednotlivé parametry tratě a automobilů. Na základě zadaných hodnot pak systém vyhodnotí celkové skóre jednotlivých automobilů na dané trati.

Správce hry tak může se simulátorem experimentovat, aby při správě hry například nevytvořil trať, která by silně zvýhodňovala jen určité automobily.

### 8.3.4 Helpdesk

Helpdesk, jako celek, byl již popsán dříve. Zde chci jen připomenout, že právě správce hry je ta osoba, která u helpdesku stojí na druhé straně, než hráč. Správce hry si zde může vypsát dotazy od hráčů, a postupně na ně odpovědět.

## 9 ZABEZPEČENÍ SYSTÉMU

Při vytváření jakékoliv aplikace, je nutné myslet i na bezpečnost. U webových aplikací to platí dvojnásob, zejména pak u aplikací, které vyžadují registraci uživatelů, jako je ta naše.

Existuje mnoho způsobů, jak lze webovou aplikaci napadnout a každý způsob používá útočník k různým druhům útoku. A proč vlastně dochází k útoku na webové aplikace? Důvodů je hned několik. Snahou může být zjištění citlivých údajů, neoprávněný zásah do databáze, či změna skriptů. Některé takové útoky mohou být pro aplikaci ničivé. To může být motivací útočníka, který se snaží poškodit aplikaci ať už z důvodu konkurence, nebo jen tak pro zábavu.

Naštěstí existují způsoby jak aplikaci zabezpečit proti útokům, a ty nejběžnější, mezi které patří XSS útoky a PHP či SQL injekce, nyní popíši.

### 9.1 Cross-site scripting

Cross-site scripting, nebo také zkráceně XSS, je typ útoku, který se snaží aplikaci podstrčit skript napsaný pomocí javascriptu. K tomu využívá převážně formulářové vstupy nebo parametry v adresní řádce. Konkrétně v naší aplikaci je potřeba zabezpečit především uživatelské vstupy. Část naší aplikace pracující se vstupy od uživatele je například helpdesk, který pracuje tak, že uživatel pomocí textového vstupu vloží svůj dotaz, který si později administrátor zobrazí. Co by se však stalo, kdyby vstup nebyl ošetřený a uživatel zadal například následující skript.

```
<script>alert('\Utok XSS\')\;</script>
```

Javascriptová funkce alert zobrazí vyskakovací okno s daným nápisem. To znamená, že pokud by si následně správce hry zobrazil seznam dotazů, objevilo by se před ním okno s nápisem "Utok XSS". Samotná funkce alert nám nezpůsobí žádnou škodu, pokud není napsána 100 krát za sebou, což může být značně otravné. Je nutné si však uvědomit, že útočník může použít jiné funkce javascriptu, jako třeba přesměrování stránky nebo změnu DOM struktury stránky. Tak či i onak, je jakýkoliv javascriptový kód nežádoucí, a je nutné formulářový vstup ošetřit.

Ošetření není nijak složité. Jelikož je text z uživatelského vstupu poslán nejprve na server, kde je zpracován, využijeme funkci, kterou nám nabízí PHP. Ošetření spočívá v

odfiltrování nebezpečných znaků a jejich převedení na HTML entity. V PHP nám to řeší funkce *htmlspecialchars*. Tato funkce zapříčiní, že například znak "<" nahradí řetězcem "&lt;". Použití funkce je velice jednoduché. V následující příkladu opět uvažujeme odeslání nového dotazu, který je posílán pomocí metody POST.

```
$uzivatelsky_vstup = $_POST['dotaz'];  
$osetreny_vstup = htmlspecialchars($uzivatelsky_vstup);
```

Při odeslání stejného skriptu jako v předchozím případě, se do databáze zapíše následující tvar.

```
&lt;script&gt;alert('Utok XSS');&lt;script&gt;
```

Tento tvar již není nebezpečný. Při vypsání v HTML kódu, se zobrazí normálním způsobem s ostrými závorkami, avšak samotný skript se neprovede.

Jak již bylo zmíněno, pomocí javascriptu lze napadnout i parametry v adresní řádce. To ovšem v aplikaci neřeším, jelikož není místo, kde bych v adrese předával textové hodnoty, které by se přímo vypisovali. Ošetření by však bylo možné provést stejným způsobem.

Adresní řádek však lze zneužít i jiným způsobem, než pomocí javascriptu. Jedná se o PHP a SQL injekce, které budou popsány v následující kapitole.

## 9.2 PHP a SQL injekce

Typ útoku pomocí PHP či SQL injekce, opět může využívat formulářový vstup nebo adresní řádek. Cílem je nyní však podstrčit aplikaci škodlivý PHP skript nebo jen nesprávné hodnoty, které jsou pak zpracovány. Princip PHP i SQL injekci je v podstatě stejný. Liší se cílem útoku, kdy u PHP injekce je to například PHP skript, a u SQL injekce například databáze.

PHP injekce využívá zejména funkci *include*. Tato funkce umožní vložit do skriptu jiný kus kódu, který se běžným způsobem vykoná. Využít se dá například pro vložení externího PHP souboru, který obsahuje pouze deklarace funkcí, aby byli přehledně odděleny od ostatního kódu. Při takovém použití nelze *include* PHP injekcí zneužít. Někdy však je struktura webové prezentace taková, že stránky, které se mají zobrazit, se předávají

v adrese stránky jako parametr. Při neošetření může útočník vložit do stránky vlastní PHP skript, který se vykoná. V naší aplikaci však takové vkládání nepoužívám, proto ošetření proti PHP injekci není řešeno. Je však nutné tuto hrozbu brát v úvahu, při případné změně aplikace. Co však v aplikaci je řešeno, je obrana proti SQL injekci.

Princip tohoto útoku spočívá v tom, že se snaží pozměnit SQL dotaz na databázi. V aplikaci je často využíváno toho, že data upravující dotaz na databázi, jsou posílány viditelně v adrese stránky, pomocí parametrů. Pro pochopení uvedu příklad. Představme si špatně řešenou administrátorskou část a správu uživatelů. Správce hry si zobrazí seznam uživatelů, a chce některého smazat. Smazání hráče proběhne na základě jeho identifikátoru, který je předán v adrese stránky. Při mazání hráče s číslem 5, by mohla adresa vypadat následovně.

```
www.adresa.cz/admin/user.php?delete=5
```

Pokud by se dále pracovalo s přijatou hodnotou, bez jakéhokoliv ošetření, provedl by se následující příkaz do databáze.

```
delete from users where id_user=5
```

To by bylo samozřejmě v pořádku a smazal by se pouze hráč s ID pět. Aplikační logika tvoří adresy správně, čili by vše mohlo fungovat jak má. Útočník, který však bude znát strukturu předávaných parametrů, by napevno mohl do adresy zadat následující parametr.

```
www.adresa.cz/admin/user.php?delete='5 OR 1=1'
```

Příkaz, který by se poté provedl by vypadal:

```
delete from users where id_user=5 OR 1=1
```

Takový příkaz by pak smazal všechny řádky z dané tabulky, protože druhá podmínka by byla splněna vždy. Berme tento příklad jen jako ukázkou použití SQL injekce. Takový příkaz samozřejmě může provést pouze správce hry, a ten nemá zapotřebí aplikaci napadat. I přesto by však přijaté parametry měli být nejprve zkontrolovány, než budou použity v dotazu na databázi. Důvod uvedu v kapitole o Cross-site request forgery.

Opatření proti SQL injekci je několik. Prvním z nich může být použití metody POST, pokud jsou data posílány pomocí formuláře. To zapříčiní, že parametry nebudou viditelné v adresním řádku, jak tomu je u metody GET. To však není možné zajistit vždy, protože

někdy se parametry v adrese předávají na základě stisknutí tlačítka, která má adresu pevně nastavenou.

Druhým způsobem je ověření, zda získaná hodnota má takový formát, který očekáváme. V naší aplikaci často pracujeme s identifikátory, které jsou celočíselného typu. Zde můžeme využít PHP funkci `is_numeric`, která ověří, zda získaný parametr je číslo či nikoliv. Řetězec je pak možné ošetřit například pomocí regulárních výrazů, nebo porovnáváním řetězců, jedná-li se jen o určitou množinu slov, které v parametru čekáme.

Testování čísel pomocí funkce `is_numeric` však nemusí být vždy dostačující. Uvedu příklad, týkající se plnění úkolů. Jak bylo psáno dříve, hráč může každý úkol plnit pouze jednou. Aplikační logika nedovolí hráči plnit úkol znovu, jelikož mu již splněný úkol nenabídne ke splnění. Zadávání úkolů je řešeno parametrem v adrese, a proto nic nebrání hráči, zadat identifikátor úkolu ručně. Ošetření funkcí `is_numeric` nám nepomůže, protože číslo to bude. Uživatel by tak mohl plnit jeden úkol stále dokola, i když mu jej hra již nenabízí. V takovém případě v aplikaci ještě ověřuji, zda požadovaný úkol nemá hráč ještě splněný.

### 9.3 Cross-site request forgery

Tato metoda využívá k útoku jinou osobu. Rovnou uvedu příklad. Útočník nemá možnost přístupu do administrátorské sekce, nicméně může znát někoho kdo ano. Díky tomu může připravit útok, který provede prostřednictvím správce hry, který je přihlášený do administrátorské sekce. To může provést například pomocí podvrhnutého odkazu, který by správce hry otevřel. Při takovém útoku se využívá i technik sociálního inženýrství, aby například správce hry udělal to, co chce útočník, právě třeba kliknutí na určitý odkaz.

Jak jsem předesílal v předchozí kapitole, existovala by možnost, jak smazat všechny řádky z tabulky uživatelů. Útočník by pak vytvořil odkaz z patřičným parametrem, který by danou operaci provedl. Kdyby pak donutil osobu, přihlášenou do administrátorské sekce, na tento odkaz kliknout, měl by vyhráno.

To samé by mohl udělat jednodušeji, pokud by například náš helpdesk nebyl ošetřen proti zpracování javascriptu. Útočník by pak v dotazu mohl použít funkci pro přesměrování, a jakmile by si administrátor dotaz zobrazil, byl by přesměrován, a útočník

by také dosáhl svého. Předpokladem by však bylo, že útočník zná strukturu aplikace a předávaných parametrů.

Není však dobré spoléhat na nevědomost útočníka, a proto je nutné ošetřit i takové parametry, kde se nepředpokládá, že by byli jiné, než dovolí aplikace. Zároveň mnoha operacím v administrátorské části předchází hláška s dotazem, zda se má zadaná operace provést, kterou je potřeba potvrdit. Pokud by příkaz pro smazání přišel odjinud, má tak správce hry možnost smazání odmítnout.

## 9.4 Detekce útoků

Jak z předchozích kapitol mohlo vyplynout, ošetření útoků spočívá v tom, vytvořit určitým způsobem podmínku, přes kterou projdou jen korektní data. Jelikož aplikace, používá při běžném užívání pouze korektní parametry, lze jednoduše útok detekovat, pomocí klauzule `else` u dané podmínky.

V aplikaci se ukládají informace o případných pokusech napadnout systém. Slouží k tomu tabulka **warning**. Pokud některý parametr neprojde podmínkou, lze předpokládat, že se jedná o útok. A nejen že aplikace z pohledu uživatele neprovede nic, zapíše se pokus o útok to zmiňované tabulky.

Jelikož se na každé stránce aplikace nejdříve testuje, pomocí `session`, jaký je přihlášen uživatel a jaká má práva, nelze provádět útoky zvenčí. Uživatel proto musí být přihlášen, a tím pádem při detekci útoku můžeme zjistit konkrétního uživatele, který se o útok pokusil. Tento údaj se uloží do tabulky, společně s hláškou a datem, kdy byl pokus zaznamenán. Hláška určuje, o jaký typ útoku se hráč pokusil, což lze s podmínkou přibližně zjistit.

Tabulka pak může sloužit správcům hry, kteří na základě toho mohou odhalit hráče, kteří se snaží obcházet pravidla hry. Na základě toho pak mají možnost uživateli odstavit celý herní účet.

## 9.5 Ukládání hesel

Poslední kapitola z oblasti bezpečnosti se bude zabývat ukládáním hesel v databázi. I přesto, že by se k databázi neměl dostat nikdo neoprávněný, je vhodné citlivá data

nějakým způsobem šifrovat. V našem případě se bude jednat pouze o hesla k herním účtům.

Jak PHP tak MySQL mají zabudované hashovací funkce MD5 a SHA1. Princip spočívá v tom, že hashovací funkce vytvoří z nějakého řetězce posloupnost znaků. Správná hashovací funkce nesmí pro různé řetězce vytvořit stejný hash. Ze samotného hashe se pak již nedá jednoduše rozluštit vstupní slovo.

Princip přihlašování je pak takový, že se zadané heslo převede také na hash stejnou hashovací funkcí, a pro ověření totožnosti uživatele, se mezi sebou porovnají právě tyto zakódované řetězce.

V dnešní době se již funkce MD5 nepovažuje za bezpečnou, a proto je v naší aplikaci použita funkce SHA1.

Na závěr kapitoly si ukažme krátký kód, který demonstruje použití hashovacích funkcí v PHP, při registraci a přihlašování uživatele.

```
Registrace uživatele:
$databazove_heslo = sha1('$_POST['heslo']');
// poté uložení například do databáze

Přihlášení uživatele:
if(sha1('$_POST['vstup_od_uzivatele']')== $databazove_heslo){
    echo 'přihlášení OK';
}
```

## 10 TESTOVÁNÍ

Celý systém byl vytvářen pod operačním systémem Windows. Jelikož však mnoho webových serverů běží pod Linuxem, bylo nutné aplikaci průběžně testovat i na tomto operačním systému, protože některé věci řeší Linux jinak než Windows. Rozdíl, který při vývoji především zajímal mě, byl ten, že se Linux při práci s databází chová jako case-sensitive, což znamená, že rozlišuje velká a malá písmena.

Abych nemusel složitě přenášet či kopírovat zdrojové kódy na jiný stroj, kde běží Linux, využil jsem program VirtualBox. Díky tomuto virtualizačnímu programu, jsem získal možnost, spustit Linux přímo do systému Windows. Pomocí nastavení, které podporuje VirtualBox, jsem si zajistil síťovou komunikaci mezi systémy. Po té bylo možné, pomocí IP adresy, přistupovat z Windows k localhostu Linuxu. Zbývalo jen nastavit správné sdílení složky, se zdrojovými kódy, mezi jednotlivými systémy. V konečném výsledku jsem pak mohl z Windows, přistupovat k localhostu Linuxu, kde se mě zároveň překládali kódy, uložené pod Windows. Jednoduchá změna IP adresy, mě pak nahradila celý proces kopírování zdrojových dat mezi jednotlivými systémy.

Další věcí, která byla předmětem testování, byla nutnost běhu aplikace v jakémkoliv internetovém prohlížeči. Zde vznikali problémy, především s kaskádovými styly, protože každý prohlížeč si je přebere po svém. Ladění kaskádových stylů tak proběhlo ve třech nejvíce používaných prohlížečích, kterými je Internet Explorer, Firefox a Opera. Funkčnost v těchto prohlížečích byla prioritní, v konečném důsledku však byla potvrzena i v méně užívaných prohlížečích, jako je Netscape nebo Safari.

Samotné testování funkčnosti aplikace jsem prováděl sám a to i se simulací různých útoků, které byli popsány v předchozí kapitole.

## 11 SEO

SEO, neboli Search Engine Optimization, je metodika, která optimalizuje webovou prezentaci pro internetové vyhledávače. Cílem tvůrců prohlížečových her, je samozřejmě získat co nejvíce hráčů, a proto je taková optimalizace na místě.

Podívejme se nyní na pár způsobů, kterými byla optimalizována i naše aplikace. Prvním důležitým faktorem je správná volba klíčových slov. Jsou to slova, která nejlépe vystihují celou aplikaci. Musíme se zde zamyslet, jaká slova bude uživatel přes vyhledávač volit, při hledání prohlížečových her. Pro naši aplikaci tak byla volena například slova jako prohlížečová hra, auta, automobily, závodní, závody, tuning. Samozřejmě pak můžeme použít i anglické alternativy těchto slov, zejména pak například pojem "browser game", který je typický pro hledání her i v české republice. Jeden způsob jak pak klíčová slova využít, je jejich užití v hlavičce stránky. K tomu slouží v HTML meta tagy keywords a description. Zapisují se do hlavičky stránky v následující podobě.

```
<meta name="keyword" content="browser game prohlížečová hra auta ...">
```

Některé vyhledávač však meta tagy ignorují, proto se podívejme na další způsob. Tím je vhodné rozmístění klíčových v textu na stránce, pokud možno vložených mezi určité tagy. Největší váhu pak má tag <title> následovaný tagem <h1>. Určitou váhu slovu můžou přidávat i tagy <em> nebo <strong>. [6]

Při ostrém spuštění aplikace by pak bylo ještě vhodné, vytvoření zpětných odkazů. K tomu nám mohou posloužit různé katalogy stránek. Samozřejmě platí, že čím víc zpětných odkazů, tím lépe. Tento způsob optimalizace však zatím nebyl řešen, protože aplikace zatím neběží na internetu v ostrém provozu.

## 12 INSTALACE SYSTÉMU

Podívejme se na závěr, co je potřeba udělat pro to, aby jsme celou aplikaci zprovoznili. V první řadě k tomu budeme potřebovat nějaký server, na kterém běží PHP a databáze MySQL.

Mezi zdrojovými soubory je připraven instalační skript. Aby však vše proběhlo v pořádku, musíme ručně nastavit přihlašovací údaje k databázi. Ty nalezneme ve složce `include` v souboru `connect.php`. Obsah souboru má následující podobu.

```
<?php
    $host = "localhost";
    $uzivatel = "root";
    $heslo = "heslo";
    $databaze = "dp";
?>
```

Zde je potřeba vyplnit správné údaje pro přístup k databázi. Tento soubor je nejen při instalaci, vkládán všude tam, kde se pracuje s databází. V ukázce vidíme, jak může obsah souboru vypadat.

Po té již stačí spustit skript `install.php`, který najdeme v kořenovém adresáři. Ten neprovede nic jiného, než že v dané databázi vytvoří jednotlivé tabulky, potřebné pro běh aplikace. Zároveň vytvoří v tabulce **user** první záznam, s administrátorským účtem. Heslo k tomuto účtu si volíme sami, během instalace.

Pokud vše proběhlo v pořádku, můžeme přes soubor `index.php`, který je rovněž v kořenovém adresáři, načíst úvodní stránku a pustit se do hraní.

## ZÁVĚR

Aplikace byla vytvořena podle zadání, a byli splněny jednotlivé jeho body. Na základě analýzy jsem pro webovou hru zvolil téma automobilových závodů, a myslím si, že toto téma bylo zvoleno vhodně, s ohledem na aktuální situaci na české herní scéně.

I přesto, že je výsledná aplikace funkční, musela by před ostrým provozem podstoupit ještě mnoho změn. Takovou hlavní změnou by byl vzhled celé aplikace, o kterou by se musel postarat zkušený grafik, jelikož si myslím, že výborný vzhled je u podobných aplikací polovinou úspěchu. Celá aplikace tak byla řešena spíše po stránce logické, ve snaze získat rychlý a bezpečný systém.

Kouzlo webových her tkví v tom, že existují snad neomezené možnosti, jak hru stále vylepšovat. Proto bych celou aplikaci neoznačil za finální produkt, nýbrž za připravené prostředí, které lze různými způsoby dále rozvíjet.

Při vytváření aplikace jsem si vyzkoušel různé fáze vývoje softwaru, ať už jde o analýzu, specifikaci požadavků, tvorba databázového modelu či modelu případů užití, nebo o celkové zabezpečení celého systému, a myslím se, že získané zkušenosti budu moci využít i v budoucnu. Zejména pak zkušenosti s webovými databázovými systémy, kde jsem si díky práci objasnil mnoho pojmů, které jsou s takovými systémy spojovány.

Myslím si, že i psaní této zprávy mi přineslo cenné znalosti s vytvářením delších dokumentů, sloužící k popisu nějakého softwaru.

Na úplný závěr bych rád podotkl, že i když cílem práce bylo vytvoření hry, tak hra to rozhodně nebyla.

## ZÁVĚR V ANGLIČTINĚ

Application was created according to the instructions and have been met each of its points. Based on the analysis of browser game I've chosen the theme of car racing, and I think that this theme was chosen right, given the current situation in the Czech game scene.

Even though a functional of the resulting application, it would undergo a lot of changes before the real start. That change would be the graphics design of application, which would have to take care of a professional graphic designer, because I think, that the perfect vizual look is good for success of the application. Application was solved more in terms of logic in order to gain fast and secure system.

The advantage of browser game is existence probably infinite ways that constantly improve the game. That is why, I did not label the application as the final product, but a prepared environment, which can be further developed in various ways.

When creating applications I have tried different phases of software development, whether in terms of analysis, requirements specification, creating a database model or use-case model, or the overall security of the system, and I think that experience will be able to use in the future. Especially experience with web database systems, where I made it clear of many terms, that are associated with such systems.

I think the writing of this report was very beneficial for me whit the creation of long documents, used to describe any software.

At the end I would like to point out that although the goal was to create a game, it definitely didn't be a game.

**SEZNAM POUŽITÉ LITERATURY**

- [1] CASTAGNETTO, Jesus, et al. Programujeme PHP profesionálně. Praha: Computer Press, a. s., 2002. 656 s.
- [2] ASLESON, Ryan; SCHUTTA, Nathaniel T. Ajax : Vytváříme vysoce interaktivní webové aplikace. Brno: Computer Press, a. s., 2007. 263 s.
- [3] SMIDEK, Vojtěch; BERIT, Jaroslav; MRÁZEK, Lubor. PHP v příkladech. České budějovice: Kopp, 2001. 117 s.
- [4] SATRAPA, Pavel. Web design. Havlíčkův Brod: Neokortex spol. s.r.o., 1997. 414 s.
- [5] JANOVSÝ, Dušan. Jak psát web [online]. 2003 [cit. 2011-01-26]. Jak psát web. Dostupné z WWW: [<http://www.jakpsatweb.cz>].
- [6] ŽDÁREK, Roman. SEO Optimalizace pro vyhledávače [online]. c2004 [cit. 2011-01-26]. Jak dělat weby. Dostupné z WWW : [<http://www.jakdelatweby.cz/seo>].
- [7] BLAŽEK, L. Podvzení SQL několika způsoby [online]. 2007 [cit. 2007-06-11]. Dostupný z WWW: <http://owebu.blogger.cz/Databaze/SQL-injekce?km=d>
- [8] PEJŠA, J. Co je cross-site scripting a jak mu předcházet [online]. 2009 [cit. 2009-03-05]. Dostupný z WWW: <http://zdrojak.root.cz/clanky/co-je-xss-jak-mu-predchazet/>
- [9] ŠÁLEK, O. PHP Injekce [online]. 2006 [cit. 2006-11-18]. Dostupný z WWW: <http://mozektevidi.net/clanek/php-injekce>
- [10] RŮŽIČKA, P. Bezpečnost především – include v PHP [online]. 2003 [cit. 2003-5-10]. Dostupný z WWW: <http://interval.cz/clanky/bezpecnost-predevsim-include-v-php/>
- [11] HyperText Markup Language [online], poslední aktualizace 4. dubna 2011 20:26 [cit. 2011-5-12], Wikipedie. Dostupné z WWW: [<http://cs.wikipedia.org/wiki/HTML>].
- [12] PHP [online], poslední aktualizace 28. dubna 2011 20:26 [cit. 2011-5-12], Wikipedie. Dostupné z WWW: [<http://cs.wikipedia.org/wiki/PHP>].
- [13] MARADA, Zbyněk. *Webová aplikace pro tvorbu prezentací*. Brno, 2009. 29 s. Bakalářská práce. VUT v Brně, Fakulta informačních technologií.
- [14] MySQL funkce [online], poslední aktualizace 24. dubna 2007 [cit. 2011-5-12], PHP manual. Dostupné z WWW: [<https://students.kiv.zcu.cz/doc/php5/cs/ref.mysql.php.html>].

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

HTML	Hypertext Markup Language - jazyk pro tvorbu webových stránek
PHP	Skriptovací programovací jazyk
MySQL	Databázový systém
PNG	Formát obrázku
GIF	Formát obrázku
GET	Metoda posílání dat na sever
POST	Metoda posílání dat na sever
ID	Jednoznačný identifikátor
AJAX	Asynchronous JavaScript and XML - technika asynchronní komunikace se serverem.
SEO	Search Engine Optimization - optimalizace pro vyhledávače.

**SEZNAM OBRÁZKŮ**

<i>Obrázek 1: Rozdělení prohlížečových her podle tématu.....</i>	<i>17</i>
<i>Obrázek 2: Rozdělení českých prohlížečových her podle tématu .....</i>	<i>17</i>
<i>Obrázek 3: Rozdělení prohlížečových her podle stylu .....</i>	<i>18</i>
<i>Obrázek 4: Aktéři v modelu případů užití.....</i>	<i>30</i>
<i>Obrázek 5: Model případů užití - hráč .....</i>	<i>31</i>
<i>Obrázek 6: Model případů užití - Administrátor .....</i>	<i>32</i>
<i>Obrázek 7: Model případů užití - helpdesk.....</i>	<i>33</i>
<i>Obrázek 8: Ukázka návrhu databáze v programu MySQL WorkBanch .....</i>	<i>35</i>
<i>Obrázek 9: Návrh databázového systému - 1. část .....</i>	<i>36</i>
<i>Obrázek 10: Návrh databázového systému - 2. část .....</i>	<i>37</i>
<i>Obrázek 11: Adresářová struktura systému.....</i>	<i>38</i>
<i>Obrázek 12: Rozvržení stránky .....</i>	<i>39</i>
<i>Obrázek 13: Uživatelské rozhraní systému.....</i>	<i>40</i>
<i>Obrázek 14: Ikony pro rychlou informaci .....</i>	<i>41</i>
<i>Obrázek 15: Ukázka změny autodílů na automobilu .....</i>	<i>44</i>
<i>Obrázek 16: Vykreslování automobilu v jednotlivých vrstvách.....</i>	<i>45</i>
<i>Obrázek 17: Šablona pro nové automobily .....</i>	<i>45</i>

**SEZNAM TABULEK**

Tabulka 1: Seznam her podle jednotlivých tvůrců.....	15
Tabulka 2: Seznam PHP funkcí pro práci s databází MySQL [14] .....	25

## SEZNAM PŘÍLOH

- [1] CD se zdrojovými soubory