



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

REZERVAČNÍ SYSTÉM VYUŽÍVAJÍCÍ GRAFICKÉ MOŽNOSTI HTML5

HTML5 GRAPHICS BASED TICKET RESERVATION SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETR KLIMEŠ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADEK POLIŠČUK, Ph.D.

BRNO 2011

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2010/2011

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Petr Klimeš

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Rezervační systém využívající grafické možnosti HTML5

v anglickém jazyce:

HTML5 Graphics based Ticket Reservation System

Stručná charakteristika problematiky úkolu:

Diplomová práce představuje ukázkou alternativního řešení dnes relativně běžných webových rezervačních systémů, založenou na grafických možnostech interaktivních technologií HTML5 a AJAX. Parametry rezervačních jednotek i klientská část aplikace budou založené na uživatelsky definovaném grafickém návrhu prostřednictvím komponenty canvas. Použitelnost řešení bude demonstrována ukázkou funkčního systému pro rezervaci a prodej místenek.

Cíle diplomové práce:

1. Seznámení s komponentou Canvas v HTML 5 a s možnostmi jejího využitím ve spojení s technologií AJAX.
2. Analýza požadavků na rezervační systém s možností tvorby uživatelského grafického návrhu.
3. Tvorba obecné webové aplikace s funkčností klasického rezervačního systému a s graficky založeným návrhem uživatelských parametrů.
4. Rozbor možných výhod představeného řešení a jeho potenciálu pro další využití.

Seznam odborné literatury:

<http://www.w3.org/TR/html5/>

<http://en.wikipedia.org/wiki/Xmlhttprequest>

Vedoucí diplomové práce: Ing. Radek Poliščuk, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2010/2011.

V Brně, dne 3.11.2010

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

ABSTRAKT

Cílem této diplomové práce je návrh a realizace on-line rezervačního systému, který bude graficky zobrazovat interaktivní plán rezervačního místa a bude umožňovat tato místa on-line vytvářet a upravovat. Systém má sloužit pro rezervaci a prodej vstupenek na kulturní, společenské a jiné události.

ABSTRACT

The purpose of the thesis is to design and implement on-line reservation system, which will graphically display an interactive map of the reservation place and will enable to create and redesign these places. The system is going to serve as reservation and sale tool for tickets on cultural, social and other events.

KLÍČOVÁ SLOVA

rezervační systém, PHP, MySQL, HTML5, Canvas, AJAX

KEYWORDS

ticket reservation system, PHP, MySQL, HTML5, Canvas, AJAX

BIBLIOGRAFICKÁ CITACE

KLIMEŠ, P. *Rezervační systém využívající grafické možnosti HTML5*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011. 51 s. Vedoucí diplomové práce Ing. Radek Poliščuk, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a na základě literatury a pramenů uvedených v seznamu literatury.

V Brně dne 26. května 2011

.....

PODĚKOVÁNÍ

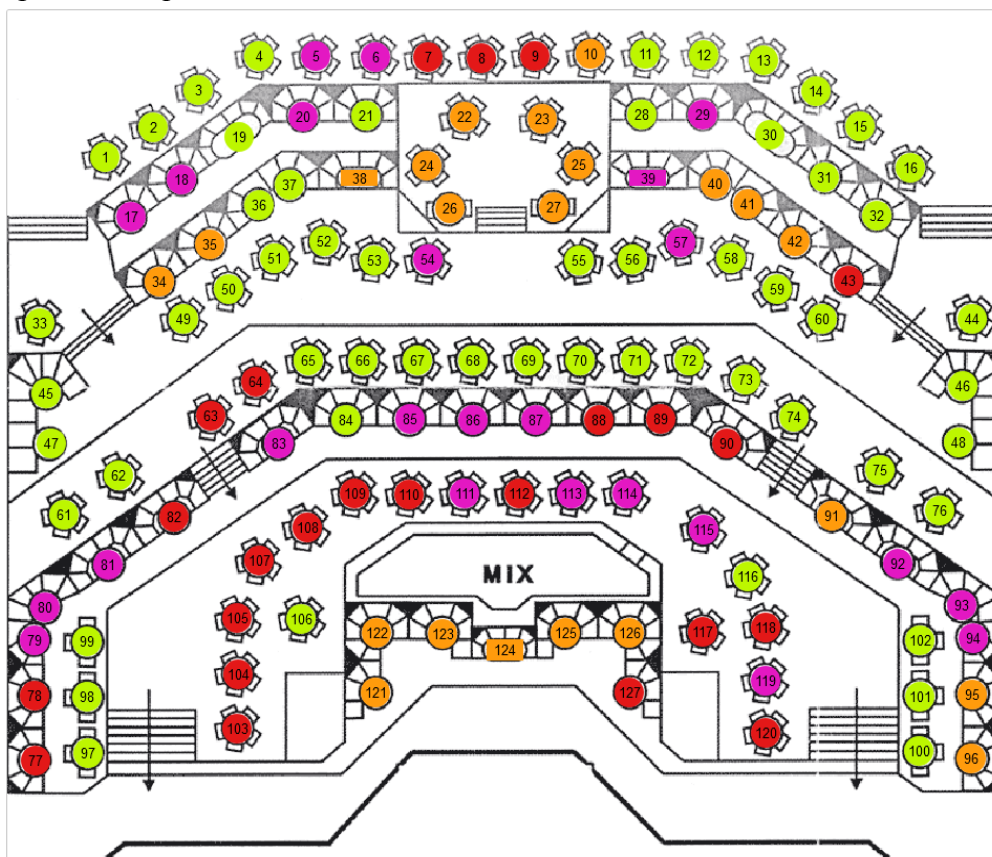
Děkuji vedoucímu diplomové práce panu Ing. Radkovi Poliščukovi, Ph.D, za jeho čas, podněty, připomínky a pomoc při návrhu a realizaci práce.

Obsah:

	Zadání závěrečné práce.....	3
	Abstrakt.....	5
	Bibliografická citace.....	7
	Prohlášení.....	9
	Poděkování.....	11
1	Úvod.....	15
2	Elektronické rezervační systémy.....	17
2.1	Speciální požadavky.....	17
2.1.1	Peněžní kontrola.....	17
2.2	Vhodné technologie.....	18
2.2.1	HTML 5 - Canvas.....	18
2.2.2	AJAX.....	20
2.2.3	PHP 5 a MySQL.....	22
3	Návrh projektu.....	25
3.1	Architektura rezervačního systému.....	25
3.1.1	Rezervace a prodej vstupenek.....	27
3.1.2	Databáze.....	27
3.2	Zpracování na straně serveru.....	29
3.2.1	Tvorba stránky.....	30
3.2.2	Šablonovací systém.....	31
3.3	Komunikace pomocí AJAXu.....	31
3.3.1	Přenášení polí.....	32
3.4	Ochrana před kolizemi.....	33
4	Produkční systém.....	35
4.1	Uživatelské rozhraní.....	35
4.1.1	Rezervace.....	36
4.1.2	Administrace.....	36
4.1.3	Statistiky.....	37
4.2	Grafická rezervace.....	38
4.2.1	Zobrazení místa.....	38
4.2.2	Rezervace místenky.....	40
4.3	Grafický návrhář.....	41
4.3.1	Kreslicí nástroje.....	41
4.3.2	Ukládání údajů.....	43
4.4	Optimalizace.....	44
4.4.1	HTML a grafika.....	44
4.4.2	JavaScript a jQuery.....	44
4.4.3	Dotazy SQL.....	44
5	Závěr.....	47
	Seznam použité literatury.....	49
	Seznam příloh.....	51

1 ÚVOD

Cílem této práce bylo vytvořit on-line rezervační systém, který bude vyhovovat požadavkům zadavatele. Podnět k jejímu vzniku dala taneční škola Dynamic, která díky vzrůstu objemu společenských událostí (jak z pohledu počtu účastníků, tak z pohledu frekvence událostí) potřebovala nahradit stávající rezervační systém na bázi tabulkového editoru elegantnějším a efektivnějším způsobem. Jako vhodné řešení byl zvolen on-line rezervační systém odstraňující hlavní nedostatky stávajícího řešení - nemožnost kontroly prodeje vstupenek a následné úniky finančních prostředků, složitost organizace prodeje ve větším objemu více zaměstnanci, a chybějící evidenci statistických údajů z uplynulých událostí, potřebné k plánování událostí budoucích.



Obr. 1 Zobrazení sálu v on-line rezervačním systému.

Jádro představované práce tvoří databáze a uživatelské rozhraní ve formě webové aplikace, které umožní administrátorům spravovat rezervační mapy zajišťovaných událostí bez nutnosti zásahu do zdrojových kódů. Důležitým přínosem je následná statistika výnosnosti použitelná pro plánování dalších akcí, uživatelské rozhraní, díky němuž si bude správce moci sám nadefinovat plán prostor a snížení finančních nákladů (současné využití systému více prodejci, nižší nároky na obsluhu) a zároveň zvýšení výnosů z prodeje.

Grafické prostředí rezervačního systému využívá nové možnosti jazyka HTML 5, zejména objekt Canvas, či prvky SVG (z angl. Scalable Vector Graphics – „škálovatelná vektorová grafika“). Projekt přitom musí respektovat dané výkonnostní limity počítačového vybavení, používaného na prodejních místech zadavatele.

Diplomová práce se skládá z pěti hlavních částí, přehledu použité literatury a příloh.

V úvodu jsou uvedeny cíle práce, požadavky na ni, návrh řešení, jeho přínosnost a zdůvodnění výběru vhodné technologie.

Druhá část popisuje problematiku rezervačních systémů obecně – jejich typy, výhody a nevýhody, a technologie, které jsou k tvorbě rezervačních systémů nejčastěji používány.

Třetí část se zabývá samotným návrhem projektu. V této kapitole je zobrazen a vysvětlen funkční mechanismus rezervačního systému včetně schémat upřesňujících architekturu, na které je rezervační systém vystavěn.

Čtvrtá část představuje rezervační systém v praxi, včetně snímků obrazovek z již nasazené varianty. Tato kapitola seznamuje s grafickým rozvržením aplikace, zejména s uživatelským rozhraním, grafickým průběhem rezervace a prací v grafickém návrháři. Jsou zde diskutovány také možnosti optimalizací a zabezpečení celého řešení.

Závěrečná kapitola pak sumarizuje splněné cíle práce a výhled pro budoucí využití vytvořeného systému.

2 ELEKTRONICKÉ REZERVAČNÍ SYSTÉMY

Rezervační systémy jsou používány nejčastěji při realizaci větších společenských akcí, kdy je výhodné rezervace a prodej vstupenek kontrolovat a spravovat v počítačovém systému. Umožňují dobrý přehled nad stavem události a zároveň udržují místenkovou i finanční kontrolu. Je-li třeba, aby mělo do systému přístup více uživatelů, musí být systém synchronizovaný, aby nedocházelo ke kolizím. Pro synchronizaci více-přístupového řešení je vhodné systém vytvořit na webové platformě a provozovat ho na síti. Webové rezervační systémy dělíme na 3 základní skupiny:

- ♦ *Místenkové* – Jde o systémy, kde je potřeba znát zasedací pořádek a podle něj rezervovat vstupenky na daná místa. Typickým příkladem použití jsou kina, plesy apod.
- ♦ *Časové* – Takové systémy zajímá pouze čas, ve kterém je či není rezervované místo obsazeno. Tyto systémy jsou typické pro rezervace sportovišť, solných jeskyní, zasedacích místností apod.
- ♦ *Kombinované* – Většinou velké systémy, které zahrnují obě předcházející možnosti rezervace a kombinují je mezi sebou.

2.1 Speciální požadavky

Online rezervační systém by měl splňovat požadavky a specifika každého zadavatele. Proto by měl být navržen tak, aby tato specifika mohla být snadno implementována – tohoto cíle lze dosáhnout modularností rezervačního systému a možností jednoduchých úprav pro programátora. Mezi nejčastější požadavky zadavatelů patří:

- ♦ Různé typy rezervací (místenkové, časové, kombinované) a z toho plynoucí nároky na rezervační systém
- ♦ Jednoduché a uživatelsky přátelské ovládací rozhraní
- ♦ Přehled o obsazenosti a vytíženosti zařízení/událostí
- ♦ Možnost správy více paralelních zařízení/událostí
- ♦ Statistika finančních dat a reporty
- ♦ Možnost tisku vstupenek přímo z rezervačního systému
- ♦ Možnost vstupu více osob do systému a zároveň ochrana proti rezervačním kolizím
- ♦ Jednoduché zařazení rezervačního systému do stávajících webových stránek společnosti.

Pokročilejší zadavatelé mají zájem rovněž o možnost samostatné správy zařízení/událostí, bez nutnosti zásahu programátora. Tyto rezervační systémy vyžadují tvorbu pokročilejšího uživatelského rozhraní, obohaceného o možnost samostatného vytváření událostí a z toho plynoucí variabilitu možných grafických návrhů.

2.1.1 Peněžní kontrola

Online rezervační systémy by kromě primární rezervační funkce měly plnit i funkci peněžní kontroly, zejména pro management společnosti. Cílem implementace rezervačního systému by měly být pro vedení společnosti:

- ♦ snadný přehled o množství prodaných a rezervovaných vstupenek,
- ♦ sledování peněžního toku,
- ♦ přehled o zodpovědnosti jednotlivých zaměstnanců za příjmy a výdaje.

Z výše uvedeného přehledu o množství prodaných a rezervovaných vstupenek plyne možnost reakce managementu na obsazenost jednotlivé události – v případě nízké účasti se může rozhodnout pro marketingovou podporu události, a naopak v případě plné obsazenosti iniciovat rozšíření události (např. otevření dalšího sálu) nebo marketingovou podporu snížit či úplně zastavit.

Sledování peněžního toku nabízí vedoucím přehled o výnosnosti všech událostí a umožňuje tak přizpůsobit rozhodovací proces o nadcházejících událostech výsledkům analýzy peněžního toku (úspěšná událost může být zopakována častěji, neúspěšná může být naopak vynechána apod.).

Zavedením individuální zodpovědnosti zaměstnanců dochází k zamezení ztrát vstupenek a omezení rizika černého obchodu. Každý zaměstnanec je v systému přihlášen pod svým jménem, a pokud prodá vstupenku, je zodpovědný za příslušný odvod peněz do firemní pokladny.

2.2 Vhodné technologie

Technologie použité v souvislosti s rezervačním systémem jsou zvoleny na základě zadání práce. Ve výběru byly upřednostněny technologie, které jsou distribuovány zdarma i pro komerční využití.

2.2.1 HTML 5 - Canvas

Web se neustále vyvíjí, a tempo tohoto vývoje se značně zrychluje. Každý den vznikají tisíce nových webových stránek, stávající stránky jsou vylepšovány a inovovány. Jedinými bariérami, které stojí v cestě vývojářům hledajícím pokročilejší techniky a lepší funkcionalitu, zůstávají omezení skriptovacích jazyků a webových prohlížečů.

Při realizaci diplomové práce byl použit jazyk HTML 5, který oproti předchozím verzím nabízí řadu nových a vylepšených funkcí, mezi které patří například pokročilé nástroje pro práci s formuláři, grafikou, různými API, multimédií, strukturami a novou sémantikou. Díky těmto inovacím nalézají autoři webu více flexibility a interoperability, což jim umožňuje vytvářet kvalitnější a uživatelsky přátelštější webové stránky. Jelikož se vývoj HTML 5 posouvá neustále dopředu, následující fakta je třeba uvažovat v kontextu ostatních dnešních technologií.[1]

Pro tvorbu grafického prostředí rezervačního systému jsem volil mezi grafickými možnostmi HTML 5. Jako vhodné techniky se jevily objekt Canvas, či prvky SVG. Technologie SVG umožňuje přístup ke každému elementu přes SVG DOM API, což by mohl být hlavní argument pro volbu této varianty. Při nárůstu složitosti dokumentu však dochází při použití této technologie k výraznému zpomalení vykreslování. Po přihlédnutí k faktu, že rezervační systém bude běžet na nepříliš výkonných počítačích a že systém by tak mohl být značně nepružný, neboť již v počáteční fázi projektu jde o stovky vykreslených prvků, jsem zvolil objekt Canvas, který je díky svému rastrovému vykreslování rychlým nástrojem pro tvorbu rozsáhlejších grafických projektů.

Prvek Canvas nabízí snadnou možnost kreslení grafiky pomocí kódu v jazyce JavaScript. V každém elementu Canvas je možné nadefinovat kontext, pomocí jehož JavaScriptové metody lze nakreslit téměř cokoliv. Podpora tohoto prvku (nebo alespoň jeho 2D kontextu) je již dnes ve všech novějších prohlížečích plně implementována, což dokazuje následující obrázek:

HTML5 Graphics & Embedded Content

	WIN										MAC				
															
	FIREFOX		SAFARI	IE						CHROME	OPERA	FIREFOX	SAFARI	OPERA	
	3.6	4	5	6	7	8	9	10	10	11.1	4	5	11.1		
Canvas	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	87%	
Canvas Text	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	85%	
SVG	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	86%	
SVG Clipping Paths	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	86%	
SVG Inline	✗	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✗	✗	23%	

Obr. 2 Podpora elementu Canvas a SVG prvků v dnešních prohlížečích.[2]

Samotný element Canvas je v HTML zapsán jednoduchým způsobem. Určuje pouze polohu a velikost „plátna“, na které budeme kreslit. Uvnitř prvku můžeme umístit alternativní text, který se zobrazí pouze v případě, že Canvas není v daném prohlížeči podporován. Následné operace provedeme pomocí JavaScriptu.

```
<canvas width='300' height='225'>
    Tento text bude v prvku zobrazen pouze tehdy,
    není-li Canvas v daném prohlížeči podporován.
</canvas>
```

Obr. 3 Implementace prvku Canvas v HTML.

Od teď se již Canvasu budeme věnovat pouze v JavaScriptové části programu. Každé plátno má svůj kontext, což je místo určující, kam budeme kreslit.

DOM, neboli „Document Object Model“, v překladu objektový model dokumentu, představuje aplikační programové rozhraní (z angl. Application Programming Interface - API), které umožňuje pracovat se samotnou webovou stránkou tak, že dává přístup k jednotlivým objektům XML dokumentu (elementy, atributy, texty, komentáře) a k práci s nimi. Díky DOM můžeme prostřednictvím JavaScriptu vložit zvolený obsah na přesně dané místo zobrazovaného HTML dokumentu.[16]

Nejdříve je třeba najít prvek Canvas v DOM a zavolat jeho metodu *getContext*. Po zavedení kontextu můžeme kreslit prostřednictvím Canvasu, který má pro vykreslování jednoduchých tvarů předdefinovány následující základní metody:

- ♦ *fillRect*(x, y, width, height)
- ♦ *strokeRect*(x, y, width, height)
- ♦ *clearRect*(x, y, width, height)
- ♦ *moveTo*(x, y)
- ♦ *lineTo*(x, y)
- ♦ *arc*(centerX, centerY, radius, startingAngle, endingAngle, antiClockwise(bool))
- ♦ *bezierCurveTo*(controlX1, controlY1, controlX2, controlY2, endX, endY)
- ♦ *quadraticCurveTo*(controlX, controlY, endX, endY)

Detailní popis všech metod je možné najít v mnoha publikacích i na internetu.[3][4]

2.2.2 AJAX

AJAX (Asynchronous JavaScript and XML) je zkratka webových technologií, které dohromady umožňují načítání dat do webové stránky bez nutnosti jejího celého znovunačítání. Díky tomu lze tvořit uživatelsky přátelštější internetové aplikace. Při vhodném použití je také možné AJAXovou komunikací snížit přenos dat a vytížení serveru. Podpora technologií potřebných pro tento přenos je dnes ve všech standardních prohlížečích plně zajištěna.[5]

Základními pilíři aplikací využívajících AJAXu jsou:

- ♦ Prezentační rozhraní založené na zavedených standardech využívajících *HTML (XHTML)* a *CSS*.
- ♦ Dynamické zobrazování údajů a interakce s uživatelem prostřednictvím *DOM* (Document Object Model).
- ♦ Asynchronní výměna údajů se serverem na pozadí prohlížeče s využitím *XMLHttpRequest*.
- ♦ Programovací jazyk JavaScript, pomocí kterého je uskutečněna součinnost vyjmenovaných technologií.

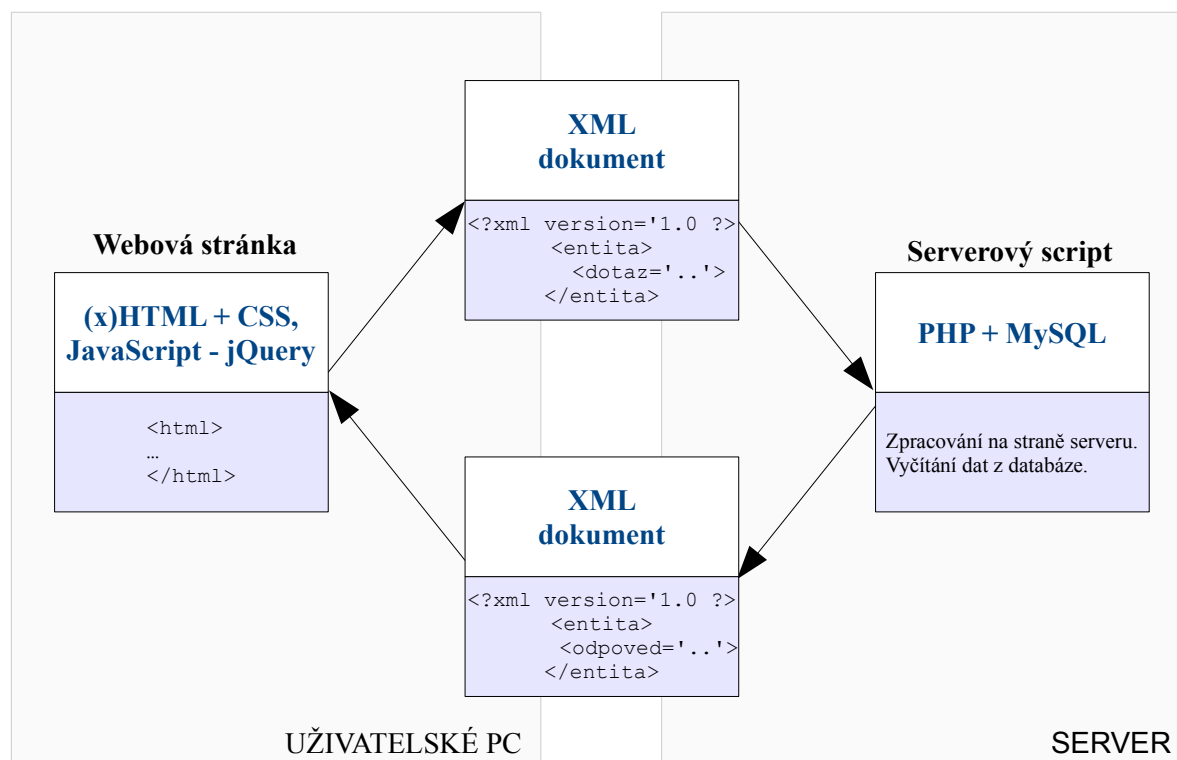
Mezi hlavní výhody technologie AJAX patří podstatné zrychlení běhu uživatelského rozhraní internetových a intranetových aplikací, neboť není třeba po každé operaci uživatele znovu načítat obsah stránky, jako je tomu u klasického modelu serverem generovaných webových stránek. Takto je možné zlepšit funkcionalitu webových aplikací z pohledu uživatele na míru srovnatelnou s klasickými desktopovými aplikacemi. Další výhodou AJAX řešení je podstatná úspora přenosové kapacity, plynoucí z faktu, že se již při každé změně dat nemusí generovat a posílat kód celé stránky. Uživatel tak kromě volnější přenosové linky navíc získá i rychleji a plynuleji fungující aplikaci.

Aplikace technologie AJAX má však i své nevýhody – a to zejména nemožnost užívání některých prvků, které si běžní uživatelé oblíbili. Jedná se především o tlačítka „Zpět“ a „Další“ webového prohlížeče, která mají uživatele vrátit na předchozí stránku či posunout na stránku následující. Tato tlačítka buď vůbec nefungují, a nebo se chovají nestandardně z pohledu uživatele – například tlačítko „Zpět“ uživatele vrátí na předchozí stranu, ale bez zaznamenání změn a operací, které uživatel provedl na původní stránce. Tímto způsobem může docházet ke ztrátě údajů. Dalšími možnými nevýhodami pro běžného uživatele jsou nemožnost uložit stránku mezi oblíbené, či poslat mailem adresu URL stránky v konkrétním požadovaném stavu.

Možným rizikem je také potenciální zvýšení komunikační zátěže mezi klientem a serverem: klient sice klade na server menší požadavky, ale zato mnohem častěji. Implementace technologie AJAX tedy může počet vyměňovaných http požadavků i zvýšit (záleží na vhodnosti implementace).

Nevýhody této technologie se dají zmírnit či zcela odstranit, avšak vyžadují použití dalších technik, čímž ve výsledku komplikují návrh stránek.

Použití technologie AJAX je pro webový rezervační systém velkým přínosem. Program se chová podobně jako běžné systémové aplikace, na které jsou uživatelé zvyklí, a přitom stále využívá výhod plynoucích z on-line řešení.



Obr. 4 Schéma AJAXové komunikace mezi uživatelem a serverem.

Technologie AJAX je využitelná na platformách PHP, ASP, Eclipse atd., přičemž princip fungování je na všech těchto platformách stejný: Nejdříve je na stránce spuštěn skript, umožňující komunikaci se serverem nezávisle na prohlížeči. Pokud klient prostřednictvím aktivního prvku vznesne požadavek na server, je JavaScriptovým kódem vygenerován XML dokument, který obsahuje údaje o operaci provedené klientem. Tyto informace jsou získány od ovládacího prvku. XML dokument je pomocí objektu *XmlHttpRequest* zaslán na server, který na tento požadavek reaguje vygenerováním dalšího XML dokumentu. Vygenerovaný dokument s požadovanými informacemi je zaslán klientovi. Na straně klienta jsou prostřednictvím technologií DOM a DHTML informace zobrazeny na stránce bez nutnosti jejího opětovného načtení. [7]

Jádro tohoto funkčního mechanismu tvoří objekt *XmlHttpRequest*, při jehož vytváření je třeba brát v potaz, že v internetovém prohlížeči Internet Explorer je tento objekt interpretován jako objekt ActiveX. Co se ostatních prohlížečů týče, byla již implementována třída *XMLHttpRequest* jako nativní objekt JavaScriptu. Aby přenos požadavku a údajů prostřednictvím objektu *XMLHttpRequest* proběhl v pořádku, je třeba při realizaci dbát na hodnotu metody *readyState* a na odpověď serveru *status*, neboť tento objekt je závislý na mnoha faktorech a okolnostech. [7]

Z důvodu ulehčení relativně složité implementace jsou na celý proces přenosu předprogramovány funkce v JavaScriptových frameworkích. Tyto funkce většinu výjimek a problémů vyřeší za nás. V nejznámějším JavaScriptovém frameworku jQuery je AJAXová komunikace věnována rozsáhlá sbírka funkcí. Nejpoužívanější z nich jsou následující:

- ♦ *get* - Načítá data ze serveru pomocí HTTP GET požadavku.
- ♦ *getJSON* - Načítá data ve formátu JSON ze serveru pomocí HTTP GET požadavku.
- ♦ *load* - Načítá data ze serveru a umístí vrácená HTML data do zadaného elementu na stránce.
- ♦ *post* - Načítá data ze serveru pomocí HTTP POST požadavku.

```

1 $.get("test.php", { name: "John", time: "2pm"},
2     function(data) {
3         alert("Data Loaded: " + data);
4     });

```

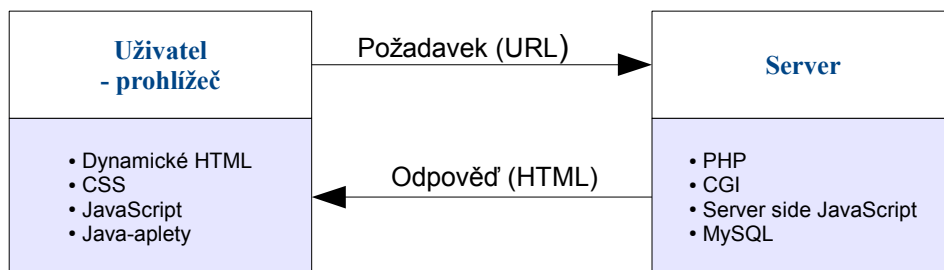
Obr. 5 Příklad použití funkce `get` v `jQuery`. [9]

Pro ilustraci jednoduchosti práce s AJAXem v `jQuery` uvádím příklad použité funkce `get`. V příkladu se odešle na server požadavek na zpracování skriptu `test.php` s proměnnými `name` a `time` za pomoci metody `GET`. Výstup ze souboru `test.php` se dostane ke klientovi v proměnné `data`. V příkladu je proměnná `data` vypísána ve vyskakovacím okně JavaScriptu a je sloučena s předurčeným řetězcem znaků.

2.2.3 PHP 5 a MySQL

V poslední době se pro tvorbu a správu dynamických webových stránek používají zejména programovací jazyk PHP a databázový systém MySQL. Jejich obliba mezi uživateli je založena především na jednoduchosti, podpořené nepřebírným množstvím funkcí pro správu i náročných či rozsáhlých projektů.

Mezi výhody PHP patří jeho specializace na webové stránky, multiplatformnost (jak Linux, tak Windows), vysoká podpora na hostingových službách a obsáhlá dokumentace. Databázový systém MySQL se řadí mezi nejrychlejší systémy, které mají současní weboví vývojáři k dispozici. Jedním z nejdůležitějších kritérií při výběru programovacího jazyka a databázového systému je cena, která rovněž hovoří ve prospěch obou – tedy PHP 5 i MySQL. Oba jsou totiž poskytovány jako Open Source, tedy zdarma, přičemž nabízejí možnosti srovnatelné s jejich placenými konkurenty, jakými je např. ASP.NET. [8]



Obr. 6 Průběh komunikace mezi prohlížečem a WWW-serverem. [6]

Mnohé z nedostatků PHP 4 byly odstraněny ve verzi PHP 5. Autoři šli vstříc požadavkům vývojářů a implementovali spoustu nových funkcí, které podporují moderní trendy programování. Došlo ke zdokonalení hlavně v oblasti objektově orientovaného programování, podpory databázového systému MySQL a strukturovaného jazyka XML. V oblasti objektově orientovaného programování byly přidány například funkce:

- ♦ konstruktory,
- ♦ destruktory,
- ♦ `public`, `protected` a `private` proměnné a metody,
- ♦ interfaci,
- ♦ abstraktní třídy,
- ♦ statické proměnné a metody.

Co se týče rozšíření podpory MySQL v PHP 5, byly rozšířeny zejména:

- ♦ připravené dotazy,
- ♦ vázané vstupní a výstupní parametry,
- ♦ SSL připojení,
- ♦ funkce pro multidotazování.

Inovace podpory jazyka XML přináší následující vylepšení:

- ♦ možnost jednoduše číst a modifikovat XML dokumenty (SipleXML),
- ♦ pracuje dohromady jako celek,
- ♦ standardizováno na jedné XML knihovně libxml2,
- ♦ plně vyhovuje specifikaci W3 konsorcia,
- ♦ efektivně zpracovává data.[13]

Technologie PHP a MySQL byly pro tvorbu rezervačního systému vybrány z důvodu jednoduchosti implementace a zároveň dostačující funkcionality, která má značnou podporu uživatelů a s tím související bohatou dokumentací. Neméně důležitým důvodem byl také fakt, že jsou obě dvě potřebné technologie distribuovány jako Open Source.

3 NÁVRH PROJEKTU

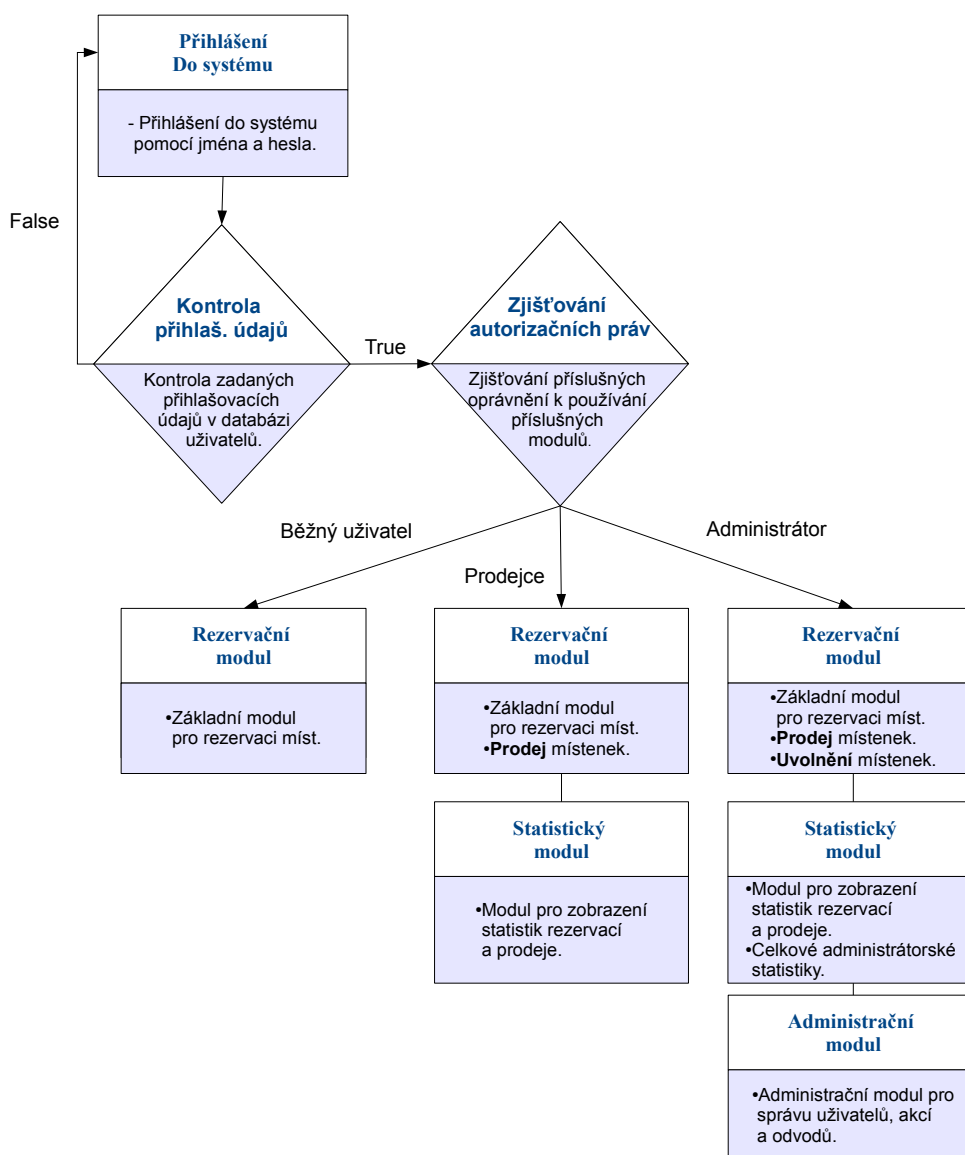
Projekt měl původně vycházet z představ společnosti a měl být jen „ušit na míru“, ovšem v průběhu návrhu a realizace projektu bylo přihlédnuto také k praktickému budoucímu využití v dalších aplikacích.

3.1 Architektura rezervačního systému

Pro vstup do rezervačního systému je nutná autorizace. Prakticky to znamená, že systém je zpřístupněn pouze po úspěšném přihlášení uživatele. Každý uživatel má nastavena autorizační práva (dle potřeb zaměstnance), podle kterých se mu zpřístupňují moduly systému. Tato autorizační práva jsou rozdělena do tří hlavních kategorií:

- ◆ Běžný uživatel
 - Tato práva má každý registrovaný uživatel. Je mu zpřístupněn základní rezervační modul s možností rezervace míst. Úspěšně přihlášený uživatel si může v grafickém prostředí vybrat místa a v systému si je zarezervovat. Rezervace vstupenek je potvrzená emailem.
- ◆ Prodejce
 - Uživatelský profil prodejce vytváří vždy administrátor dané akce. Uživateli s prodejními právy je zpřístupněn modul pro rezervaci s možností prodeje vstupenek. Pro každého prodejce je zavedena kasa, která ukazuje, jaké množství vstupenek a v jaké peněžní hodnotě již bylo prodejcem prodáno. Tyto peníze může na hlavní stránce odeslat do „odpisu“ a předat do firemní pokladny. Takto fungující odpisy byly základním požadavkem zadavatele při tvorbě systému. S prodejními právy je umožněno také využívat základní funkce ve statistickém modulu (počet celkově prodaných vstupenek na danou akci apod.).
- ◆ Administrátor
 - Uživatel s administračními právy má zpřístupněny veškeré moduly včetně administrátorského. V rezervačním modulu je kromě rezervace a prodeje umožněno místa uvolňovat pro případ omylu nebo stornování prodeje. Administrační modul umožňuje vytvářet nové a upravovat stávající události, profily prodejců, a sledovat odvody do firemní pokladny a detailnější statistiky.

Hlavním prvkem při ověřování přihlašovacích údajů je heslo. Zadané heslo je porovnáno s údajem v databázi. Tam se však neukládá přesné znění hesla, nýbrž jen jeho otisk, zašifrovaný pomocí algoritmu MD5. Z tohoto důvodu ani administrátor s přístupem do databáze nezná hesla uživatelů, pouze jejich otisk. Heslo není nikdy přímo zaznamenáno a nemůže být v případě ztráty znovu sděleno uživateli. V takovém případě je nutné vytvořit otisk nově vygenerovaného hesla. PHP má přímo vestavěnou funkci MD5, která daný otisk vytvoří. Struktura systému z hlediska uživatelských účtů je popsána na Obr. 7.

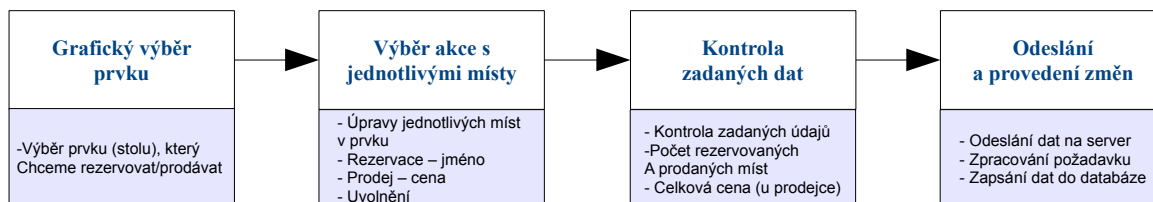


Obr. 7 Struktura systému z pohledu uživatelských účtů.

Funkční mechanismus systému je následující: Uživatel se přihlašuje pomocí uživatelského jména a hesla. Prvním krokem je kontrola přihlašovacích údajů. Pokud uživatelské jméno a heslo není v databázi, uživatel je tážán zadat jméno a heslo znovu. Pokud jsou údaje rozpoznány, následuje zjišťování autorizačních práv. V tomto kroku se rozhoduje o oprávnění uživatele, který tak získá přístup buď jako běžný uživatel do rezervačního modulu, či jako prodejce do rozšířeného rezervačního modulu (s možností vstupenky nejen rezervovat, ale i prodávat) a do základní verze statistického modulu (obsahujícího základní statistické funkce), nebo jako administrátor s nejvyššími právy, který má možnost v rezervačním modulu vstupenky také uvolňovat, dále rozšířené možnosti statistického modulu a mimo to přístup do administrativního modulu, kde může spravovat uživatelské účty a plány společenských událostí.

3.1.1 Rezervace a prodej vstupenek

Po autorizovaném vstupu do systému je hlavním cílem uživatele rezervace, respektive prodej vstupenek. Rezervační proces je zahájen výběrem akce a poté je prováděna procedura o 4 standardních krocích, zobrazených na Obr. 8. Podobnou strukturu kroků najdeme ve většině rezervačních systémů, tedy i konkurenčních.



Obr. 8 Postup při rezervaci a prodeji vstupenek.

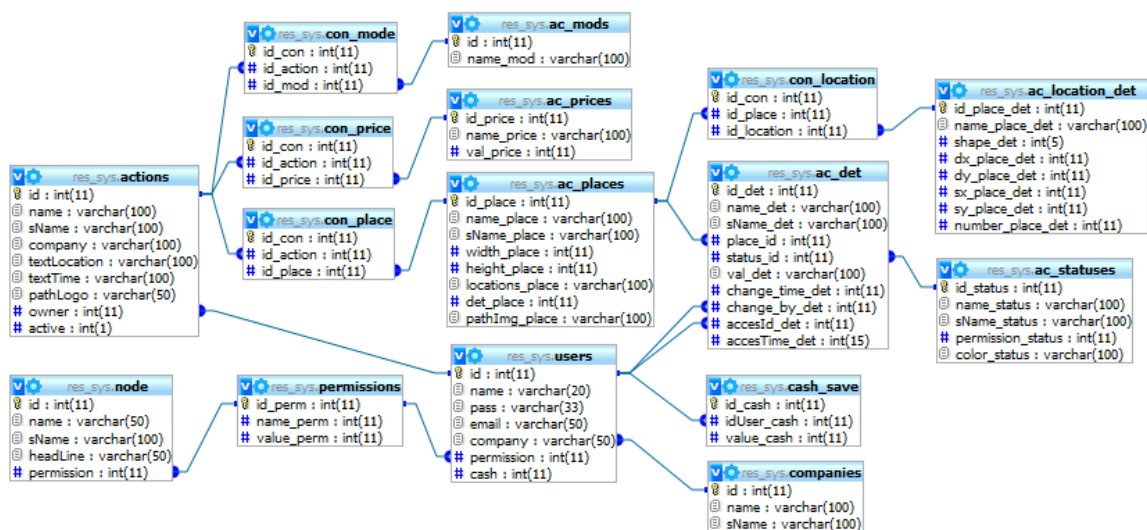
Jednotlivé kroky lze charakterizovat jako:

- ♦ **Výběr stolu**
 - Grafický výběr stolu, ve kterém chceme provádět změny. Tento výběr dává možnost zvolit si typ stolu a jeho umístění v sálu v dané události.
- ♦ **Výběr operace s jednotlivými místy**
 - Každý stůl má více jednotlivých míst k sezení. V nově otevřeném vyskakovacím okně stolu může uživatel, dle autorizačních práv, volit akce pro jednotlivá místa (rezervace, prodej, uvolnění).
- ♦ **Kontrola zadaných dat**
 - Operace provedené uživatelem na jednotlivých místech stolu se při kontrole zobrazí v přehledné tabulce. Pokud je uskutečněn prodej, je uvedena i celková cena, která má být prodejcem vybrána (dle typu vstupenky).
- ♦ **Dokončení a provedení změn**
 - Nakonec se provedené změny zapiší do databáze, a pokud došlo k rezervaci, je uživateli odeslán potvrzovací email. Při prodeji je částka, která má být vybrána, zapsána do pokladny prodejce. Prodejci je také zobrazena kalkulačka pro dopočet na vrácení peněz.

Takto dochází k rezervaci, prodeji a uvolňování vstupenek za současné finanční kontroly administrátora.

3.1.2 Databáze

Návrh databáze byl prováděn v programu phpMyAdmin v části návrhář. Celý návrh je možné vidět na Obr. 9. Hlavními tabulkami jsou tabulky „actions“ a „node“. Tabulka „actions“ obsahuje vytvořené rezervační akce. Na tuto tabulku navazují další tabulky, které ji rozšiřují a určují, jaké módy, ceny a místa se budou v dané akci nacházet. V tabulce „node“ nalezneme všechny stránky systémového webu.



Obr. 9 Relační schéma rezervačního systému.

Obsah tabulek návrhu je zobrazen v následujícím obrázku:

Popis
Tabulka obsahující základní informace o vytvořené akci.
Tabulka se všemi stránkami rezervačního systému.
Tabulka obsahující informace o uživatelích systému.
Tabulka obsahující přístupová práva.
Tabulka společností v systému.
Tabulka obsahující odvody peněz do firemní pokladny.
Tabulka obsahující možné doplňkové módy pro události.
Tabulka s cenami vstupenek.
Tabulka s rezervačními místy.
Tabulka obsahující detail rezervačního místa.
Tabulka s detailem veškerých jednotlivých míst v akci.
Tabulka s možnými statusy rezervačního místa.

Obr. 10 Stručný popis tabulek.

Protože každá akce může mít více módů (v podobě přídatných doplňků), cen i míst, je nutné, aby mezi těmito tabulkami byla relace M:N. V MySQL se tato relace realizuje pomocí spojovací tabulky tzv. „*Junction table*“. V relačním schématu jsou tyto tabulky označeny předponou „con-“. Jde o spojující tabulky, které umožňují, aby mohla mít každá tabulka více záznamů ve druhé a naopak. SQL dotaz využívající tuto relaci může vypadat podobně jako na Obr. 11.

```

1 SELECT ac_prices.name_price, ac_price.val_price
2 FROM ac_prices, con_price, actions
3 WHERE (action.id = 1)
4       AND (action.id = con_price.id_action)
5       AND (con_price.id_price = ac_prices.id_price)

```

Obr. 11 SQL dotaz na tabulky spojené relací M:N pomocí spojovací tabulky.

Stejné řešení bylo aplikováno u spojení detailů akce a jejich lokací. Detail lokace se určuje v návrhářské akci, který obsahuje předpis, jak se má dané místo zobrazovat. Detail akce obsahuje informace o stavu všech jednotlivých míst v akci, která se mohou nacházet na více lokacích. Z tohoto důvodu je třeba využít relaci M:N.

Při návrhu databáze byly brány na zřetel normalizační formy. Pro reálné nasazení je třeba zvolit střední cestu mezi normalizací a složitostí. Obecně se doporučuje navrhnout databázi na 3. normální formu. Při vyšší normalizaci dochází k velkému růstu složitosti SQL dotazů. Díky tomu je omezena rychlost vývoje systému a v důsledku složitých dotazů může docházet ke snížení výkonu celého systému. Pro splnění určité normální formy je nutné splnit i ty předchozí. Pro 3. normální formu je nutné splnit následující normální formy:

- ♦ 1NF: Každý atribut obsahuje pouze atomické hodnoty.
- ♦ 2NF: Každý neklíčový atribut je plně závislý na primárním klíči.
- ♦ 3NF: Všechny neklíčové atributy musí být vzájemně nezávislé.

3.2 Zpracování na straně serveru

Hlavní část systému je zpracovávána na serveru prostřednictvím PHP skriptů. Uživatel posílá na server data prostřednictvím URL požadavku, nejčastěji kliknutím na odkaz. Program na serveru po přijetí požadavku zajistí zpracování a zapsání potřebných dat do databáze a vrácení vyžádaných dat zpět k uživateli v HTML podobě.

Po vstupu do systému se uživatel musí přihlásit, a podle přihlašovacích práv, která mu jsou přiřazena, mu jsou nabídnuty systémové moduly s jeho právy. Po odeslání přihlašovacích údajů a jejich následné kontrole a zpracování jsou po ukončení běhu skriptu tato data smazána. Jelikož je třeba si některá data o úspěšném přihlášení uchovat, byl v programu použit mechanismus *session*.

Session představuje relaci umožňující přesnou identifikaci uživatele a jeho pohyb na serveru. Používá se zejména tam, kde je nutné monitorovat nebo ověřovat uživatele. Relace *session* byla implementována do PHP od verze 4.1.0 v prosinci roku 2001. *Session* pracuje na jednoduchém principu: Každý uživatel, který vstoupí na stránku, obdrží unikátní jednoznačný řetězec znaků SID (Session ID), který se skládá z číslic a znaků anglické abecedy. Tímto řetězcem je uživatel jednoznačně identifikován. Tento parametr je přenášen ve všech odkazech. PHP implicitně předává SID v proměnné *PHPSESSID*. *Session* tak umožňuje předávání de facto neomezeného počtu proměnných. Aby *session* v PHP fungoval, je třeba, aby start *session* byl volán před prvním výstupem. [10]

Obr. 13 znázorňuje třídy, které se starají o zpracování dat na serveru a odpovědi na dané požadavky. Pro práci s třídami je využita funkce *__autoload*, představená v PHP 5. Bez ní by musela být vždy třída přímo součástí skriptu, který ji chtěl využít, nebo by k němu musela být alespoň připojena pomocí funkce *include_once* (případně *require_once*).

```

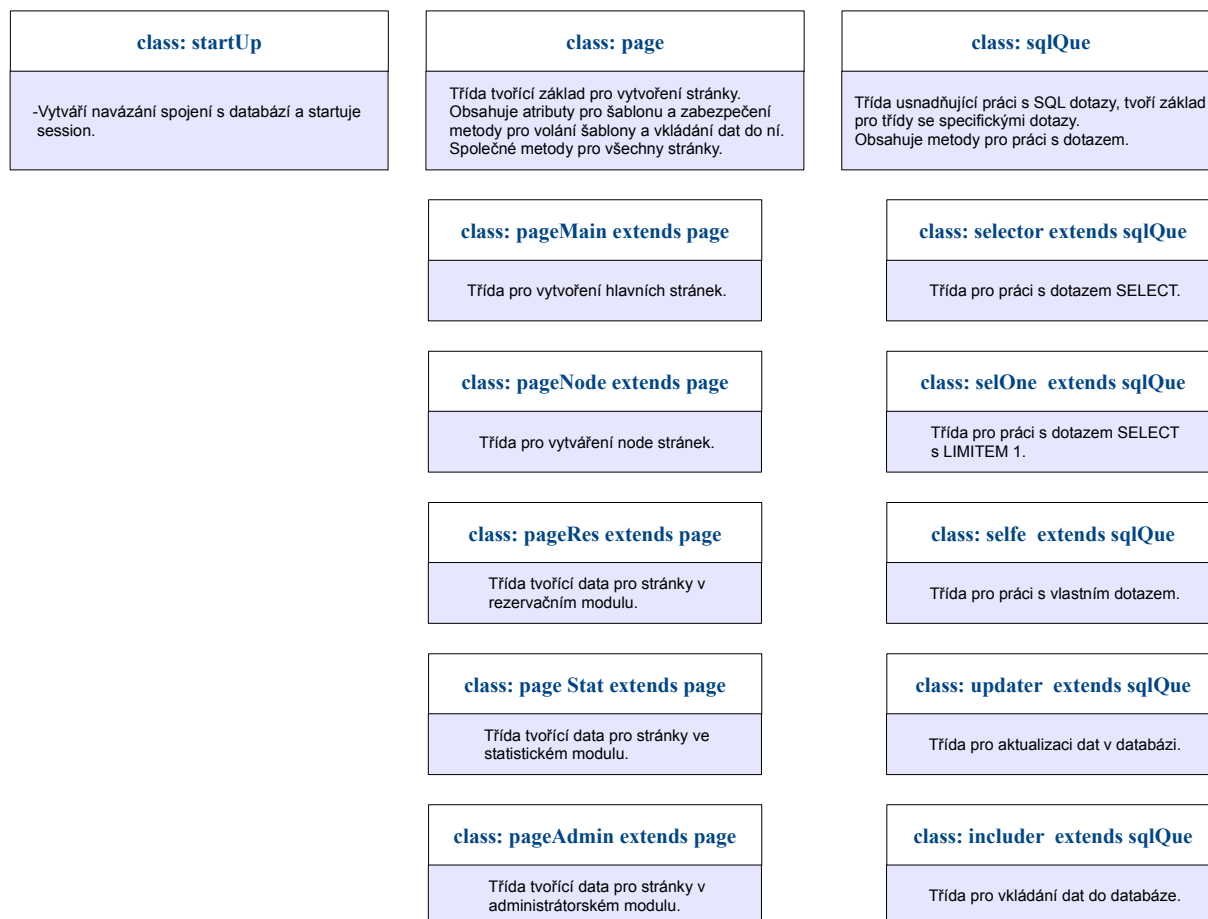
1 <?php
2 function __autoload($className)
3 {
4     require ("./scripts/PHP/$className.php");
5 }
6 ?>
```

Obr. 12 Příklad použití funkce *__autoload* v PHP.

Např. pokud je třeba vytvořit novou instanci třídy „*page*“ pomocí příkazu „*new page*“, připojí se k vykonávajícímu skriptu kód souboru *page.php*. Kód třídy se připojí až když je skutečně vyžadován. Tím se zefektivňuje chod aplikace a zpřehledňuje kód.

Základem systému jsou 3 základní třídy:

- ♦ Třída „*startUp*“, která zahajuje stránku startem *session* a navázáním spojení s databází systému.
- ♦ Třída „*page*“ tvoří základ pro vytváření stránek na serveru. Z této třídy dědí specifické třídy stránek metody pro práci s daty. Navíc tato třída obsahuje atributy pro zapsání dat do šablony stránky a přihlašovací práva daného uživatele.
- ♦ Třída „*sqlQue*“ pomáhá zjednodušit komunikaci s databází pomocí předdefinovaných SQL dotazů a jejich zpracování.



Obr. 13 Návrh tříd zajišťujících zpracování dat na straně serveru.

3.2.1 Tvorba stránky

Protože systém bude provozován na serveru, kde nebude moct být nastaveno *session.auto_start = 1*, je třeba nastartovat *session* manuálně, a to dříve, než dojde k jakémukoliv výstupu ze skriptu. Při tvorbě stránky je proto hned na začátku volána třída „*startUp*“, která nastartuje *session* a provede připojení k databázi. Před touto třídou je ještě volána funkce *_autoload* pro načtení veškerých tříd, které budou za běhu programu využívány.

Dále je při tvorbě stránky nutné vytvořit instanci příslušné třídy, která dědí z třídy „*page*“. Při každém vytváření takové třídy se kontroluje existence přihlašovacích práv v *session*. Pokud není *session* nalezeno, nebo je právo nedostatečné k zobrazení dané stránky, je uživatel přesměrován na úvodní stránku s upozorňujícím textem v informačním poli stránky. Tímto se zamezí přístupu na danou stránku při zadání přímého URL bez autorizace. Ve vytvořené třídě zavoláme metody, které zpracují URL požadavky a zapíší data do

šablonových atributů, které jsou zděděné z třídy „page“. Ukázka tvorby úvodní stránky je vidět na Obr. 14. Dle potřeby výstupu jsou nakonec zavolány metody, které vypisují data z šablonových atributů do šablony, a tu posílají na výstup ze skriptu.

```

1 <?php
2 include './scripts/PHP/autoload.php';
3 $start = new startUp(); //nastartovani session a pripojeni k dtb
4 $page = new pageMain("esh","tmp");
5 $page->pIndex("Domů"); // volani metody pro vytvoreni stranky
6 $page->createPage("resSys"); // vytvoreni stranky pomoci sablony
7 ?>

```

Obr. 14 Ukázka vytvoření úvodní stránky index.php.

3.2.2 Šablonovací systém

Pro jednoduchou a do budoucna modulativní tvorbu stránek je výhodné, aby byly stránky vytvářeny pomocí šablony, která má již nadefinované velikosti, barvy a styly stránky. Šablona je v podstatě HTML stránka s vloženými PHP prvky. Do takovéto šablony se při vytváření stránky nahrají výstupní data, a celá stránka i se šablonou se pošle na výstup. Hlavní výhoda tohoto řešení nastává při změně vzhledu stránky: Stačí pouze upravit šablonu, a změna se projeví na všech stránkách v systému. Toto řešení podporuje i mnohotvárnost celého systému: je možné vytvořit více šablon a podle potřeb volat různé vzhledy stránek.

Pro rezervační systém využívající HTML 5 prvky byla šablona vytvořena dle nové HTML 5 sémantiky. Nové prvky *header*, *article*, *footer* apod. zjednodušily a zpřehlednily vypsaný html kód. HTML 5 je stále ve stádiu vývoje, a proto je nutné novým sémantickým prvkům nadefinovat veškeré nové CSS styly. Použitá šablona je vidět v příloze č.1.

Nahrávání dat do šablony a posílání na výstup se provádí v systému metodou „createPage“. Tato metoda načte pomocí *include_once* šablonu. Název šablony je volán v parametru metody. Načtená šablona se pomocí metody „saveOnScreen“ vyplní daty z šablonových atributů třídy, které byly naplněny během vytváření stránky. Naplněná šablona je poslána na výstup k uživateli.

3.3 Komunikace pomocí AJAXu

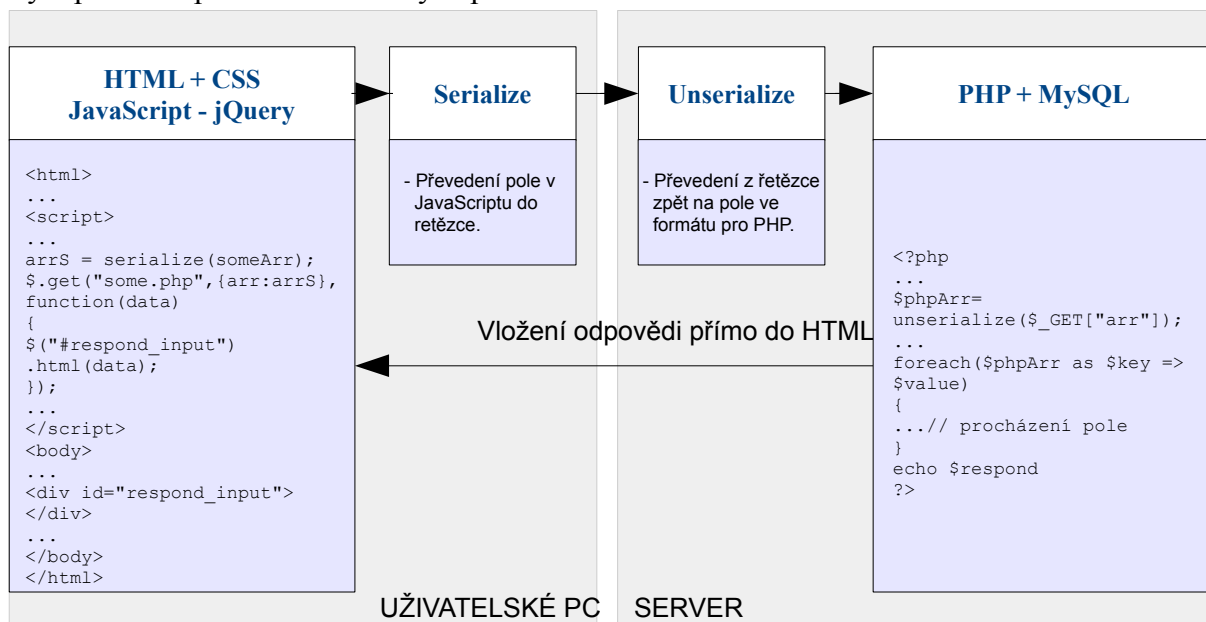
Pro zefektivnění chodu uživatelského prostředí a zmenšení objemu přenášených dat je vhodné v systému použít AJAXové komunikace, jak bylo popsáno v kapitole 2.2.2. Obzvláště u grafického výběru místa k rezervaci, respektive prodeji, kde se přenáší mnoho dat, která nemají velký dopad na vizuální změnu celé stránky. Pro potřebu vykreslení grafického místa rezervace v Canvasu jsou nezbytné informace o souřadnicích, velikostech a datech přiřazených ke každému prvku. Z důvodu rychlosti vykreslování a náročnosti JavaScriptového programu jsem zvolil uchování a přenášení dat v dvojrozměrném poli. Systém bude primárně spuštěn na notebookových platformách, a proto je důležité, aby nároky na výkon byly i u složitějších míst stále přijatelné i navzdory většímu množství dat.

Dvojrozměrné pole pro vykreslení musí obsahovat data pro zobrazení všech elementů rezervovaného místa i s informacemi, které se na dané prvky váží. Prvky v poli jsou využity při vykreslování, a také pro rozhodování při výběru prvku. Každý element v poli se skládá z 10 prvků:

- ◆ Cx - X souřadnice umístění levého horního rohu
- ◆ Cy - Y souřadnice umístění levého horního rohu
- ◆ Sx - X délka
- ◆ Sy - Y délka
- ◆ shape - Tvar elementu (1: obdélník, 2: kruh)
- ◆ status – Obsazenost místa
- ◆ active – Aktivita prvku
- ◆ idNum – Identifikační číslo
- ◆ mainName – Zobrazovaný název
- ◆ numSit – Počet míst k sezení

3.3.1 Přenášení polí

Pro možnost práce s výše popsáním polem je třeba přenést data z tabulky v databázi do pole v PHP, a poté k uživateli pro práci v Canvasu v JavaScriptu. Pro přenos pole na server byl aplikován převod na řádkový zápis.



Obr. 15 Přenos polí mezi JavaScriptem a PHP.

Přenos pole z PHP do JavaScriptu se dá realizovat vcelku snadnou implementací pole přímo do HTML výstupu ze skriptu. Z JavaScriptu do PHP se však žádná taková možnost nenabízí a při AJAXovém volání je potřeba předávat parametr v řetězci. Pro převod dvourozměrného pole do řádkového zápisu byla vybrána metoda využívající serializace. Serializační funkci JavaScript implicitně neobsahuje, proto byla využita funkce vytvořená autorem Arpad Ray [11]. Takto převedené pole je možné přenášet v URL parametru na server. V PHP skriptu je potřeba tento řádkový zápis převést zpět na asociativní pole v PHP. K tomuto účelu disponuje PHP funkcí *unserialize*, kterou však ve výchozím stavu není možné využít při kódování UTF-8. Nezbytná úprava pro multibytové kódování je na Obr. 16 :

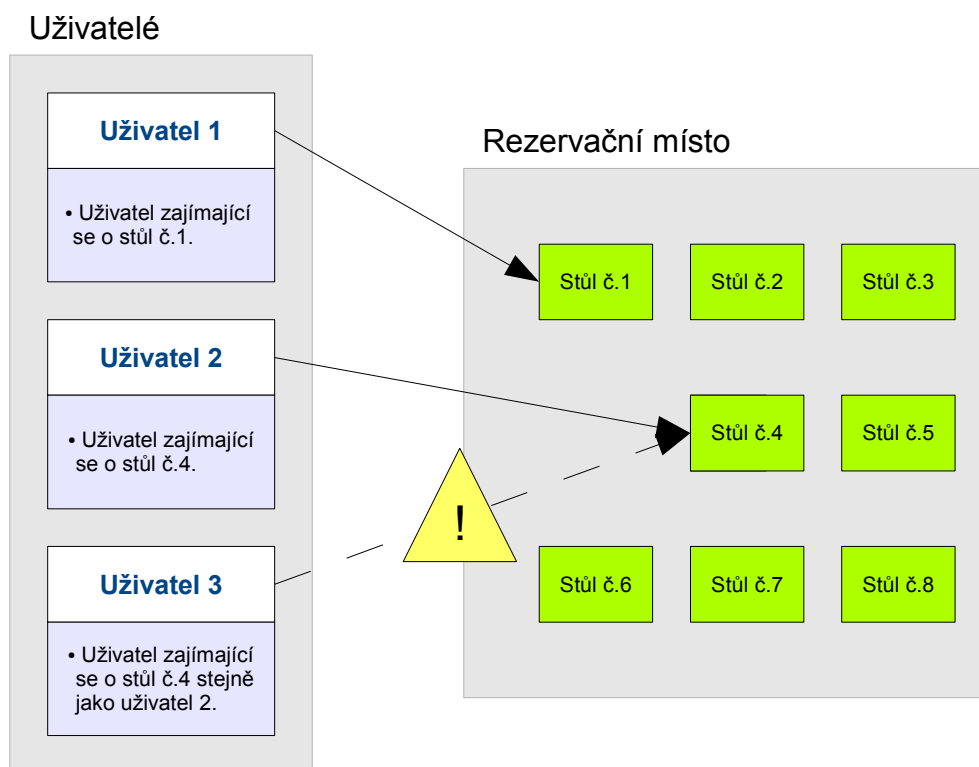
```

1 public function mb_unserialize($serial_str)
2 {
3     $out=preg_replace('!s:(\d+):"(.*)"!;!se',
4     "'s:'.strlen('$2').':\"$2\";'", $serial_str);
5     return unserialize($out);
6 }
  
```

Obr. 16 Úprava funkce unserialize pro kódování UTF-8.

3.4 Ochrana před kolizemi

Systém je budován tak, aby s ním mohlo pracovat současně a nezávisle na sobě více uživatelů z různých míst. Problém by mohl nastat v případě, kdy by si více uživatelů chtělo ve stejný čas rezervovat místo u jednoho stolu.



Obr. 17 Nákres možných kolizí při rezervaci.

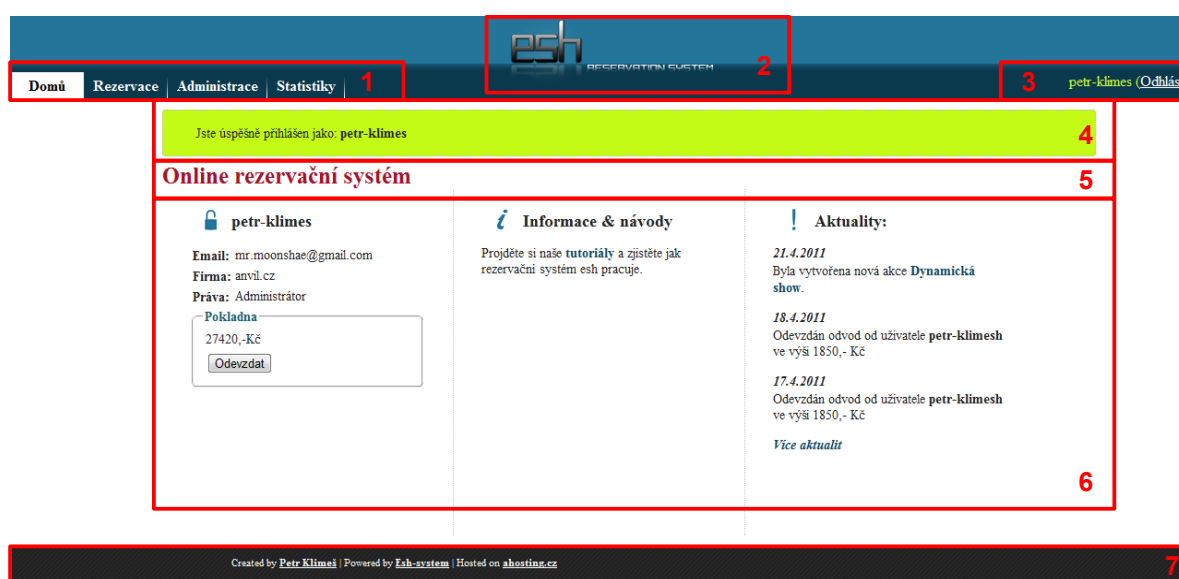
Z důvodů možné kolizní situace jsou při vstupu do rezervace jednotlivých stolů v databázi evidována časová razítka a data uživatele, který se stolem aktuálně pracuje. Pokud se jiný uživatel snaží vstoupit do rezervace stejného stolu, obdrží varování a informace o tom, kdo právě daný stůl rezervuje, a jaký je jeho časový limit pro tuto operaci. Po dokončení rezervace (respektive prodeje) jsou dočasné údaje z databáze vymazány, čímž je umožněno aby do rezervace stolu mohl vstoupit další uživatel.

4 PRODUKČNÍ SYSTÉM

Tvorba systému byla rozdělena do dvou fází, jelikož zadavatel chtěl systém co nejrychleji uvést do provozu. V první fázi byl vytvořen systém, který umožňoval pouze jednoduchou rezervaci s finanční kontrolou prodeje. Postupně byly na systém navázány zbývající funkce, a při pořádání dalších událostí byl systém doladěn dle požadavků zadavatele. Ve druhé fázi vývoje byl na nových technologiích vytvořen zcela nový rezervační systém, rozšířený o řadu dalších funkcí. Nyní je systém více modulativní a jeho transformace pro využití jinými společnostmi je snazší.

4.1 Uživatelské rozhraní

Uživatelské rozhraní bylo navrženo tak, aby bylo intuitivní a jednoduše ovladatelné. Při vstupu do rezervačního systému je zobrazena úvodní strana (viz Obr. 18).



Obr. 18 Uživatelské rozhraní rezervačního systému.

V úvodní straně může přihlášený uživatel vidět:

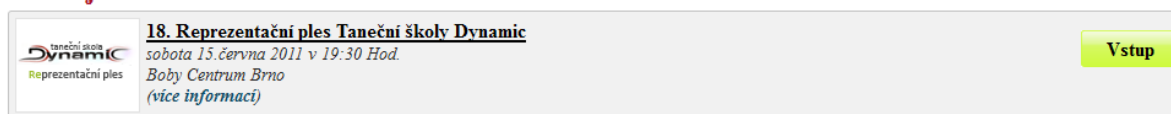
- Hlavní menu - dělí se na 4 základní části:
 - Domů - základní obrazovka systému,
 - Rezervace - zobrazení rezervačního modulu,
 - Administrace - zobrazení administrátorského modulu,
 - Statistiky – zavedou uživatele do statistického modulu.
- Logo systému – Esh reservation system – tvoří dominantu úvodní strany, proto je umístěno na střed horního elementu.
- Přihlášený uživatel – v pravém horním rohu je vidět přihlašovací jméno přihlášeného uživatele a možnost odhlásit se.
- Informační element – v tomto poli se zobrazují informační zprávy o právě provedených operacích uživatele.
- Nadpis stránky – informuje o názvu aktuální strany.
- Hlavní tělo stránky – jádro rezervačního systému – oblast, kde můžou přihlášení uživatelé pracovat se systémem a provádět operace.
- Pata stránky obsahuje odkaz na tvůrce webu a technologie, které jsou při provozu systému využívány.

Barevné schéma uživatelského rozhraní bylo vytvořeno s pomocí aplikace pro generování barevných schémat, která je dostupná na webových stránkách [12]. Tyto stránky pomáhají vytvářet vlastní schémata, která vizuálně odpovídají dnešním trendům a standardům, a jsou tak užitečným nástrojem současných vývojářů.

4.1.1 Rezervace

Vstup do rezervačního modulu je možný po úspěšném přihlášení uživatele, prostřednictvím odkazu Rezervace v hlavním menu. Do tohoto modulu mají přístup všichni registrovaní uživatelé. Na úvodní straně rezervací je výpis všech dostupných společenských událostí se základními informacemi (datum, čas a místo konání události), logem a tlačítkem pro vstup do rezervací míst dané události. Výpis jedné události je zobrazen na Obr. 19.

Prodej a rezervace



Obr. 19 Výběr akce v rezervačním modulu.

Po výběru události tlačítkem „Vstup“ se dostáváme do grafického rezervačního prostředí. Toto prostředí je jedním z hlavních přínosů systému, proto je mu věnována samostatná kapitola 4.2.

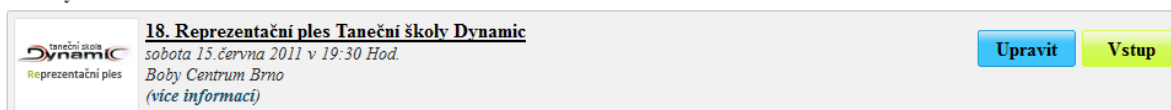
4.1.2 Administrace

Do administračního rozhraní se, stejně jako do rezervačního, dostává prostřednictvím hlavního menu. Tato nabídka se zobrazí pouze úspěšně přihlášeným uživatelům s administrátorskými právy. Podobně jako v rezervačním rozhraní se i zde zobrazuje výpis dostupných událostí, které daný administrátor spravuje. Kromě základních informací, loga a tlačítka pro vstup do události, je v administrátorském rozhraní navíc tlačítko „Upravit“, a na konci výběru možnost vytvořit vlastní novou událost. Toto administrátorské zobrazení je vidět na Obr. 20.

Administrace akcí

administrační rozhraní

Vámi vytvořené akce:



Vytvořit novou akci:

Vytvoř

Obr. 20 Ukázka možností administrace akcí.

Vytvoření nové události je jednou z činností administrátora. Po kliknutí na odkaz pro tvorbu nebo úpravu události je zobrazen formulář, který je vidět na Obr. 21. V tomto formuláři je možné vyplnit základní informace o události, nahrát logo, nastavit ceny vstupenek a rozvržení grafického uspořádání. V grafickém uspořádání je možno si vybrat z již vytvořených míst, popřípadě si tato místa upravit nebo vytvořit nová. Úprava a tvorba nových míst je zobrazena v grafickém návrháři, kterému se detailněji věnuji v kapitole 4.3.

Upravení akce

administráční rozhraní -> administrace akcí

Základní informace

Název akce: 18. Reprezentační ples Taneční školy Dynamic

Místo konání: Bobby Centrum Brno

Čas konání: sobota 15. června 2011 v 19:30 Hod.

Uložit

Logo akce:



- změnit

Rezervační místo

☒ Hlavní sál **upravit**

vytvořit nové místo

Ceny

Povolit prodej: ☒

Normální	350	,- Kč
Studentská	200	,- Kč
VIP	0	,- Kč

vytvořit novou cenu

Obr. 21 Úprava akce v administrátorském rozhraní

Systém má za úkol udržovat přehled o peněžních pohybech prodejců při prodeji vstupenek a následném odevzdávání peněz do firemní pokladny. Pro splnění tohoto úkolu je v administrátorském rozhraní přehled odvodů do firemní pokladny v podobě tabulky, která zobrazuje profily prodejců vytvořené administrátorem a jejich aktuální peněžní zůstatky doplněné součtem peněz odevzdaných do firemní pokladny. Všechny provedené odvody, seřazené podle data odevzdání, jsou vypsány pod přehledem. Při odeslání odvodu je administrátor upozorněn informativním emailem s údaji kdo, kdy a jak velký odvod provedl. Tato možnost kontroly byla jedním z hlavních požadavků firmy, aby zabránila ztrátě peněz u prodejců vstupenek.

4.1.3 Statistiky

V hlavním menu v záložce statistiky, která je zpřístupněná administrátorům a prodejcům, jsou zobrazeny důležité ukazatele dané události. Data pro ukázkou statistik na Obr. 22 byla nahrána z události pořádané taneční školou v minulosti.

Mezi základní statistické ukazatele patří:

- ◆ Počet volných/prodaných/rezervovaných míst.
- ◆ Celkový počet míst v události.
- ◆ Počet prodaných místenek dle ceny vstupenky.
- ◆ Celková cena, za kterou byly vstupenky prodány.

Statistiky pro akci: 18. Reprezentační ples Taneční školy Dynamic

statistiky

Místa

Stav	Počet
Volno:	210
Rezervováno:	25
Prodáno:	696
Celkově 931 míst.	

Prodej

Prodejní název	Cena	Počet
Normální:	270	488
Studentská:	180	185
VIP:	0	23
Celkově vybráno 165060,- Kč.		

Obr. 22 Ukázka ze statistického modulu.

Ve statistikách jsou podle aktivních módů zobrazeny další informace, jako např. prodej vstupenek dle skupin, generování .pdf souboru dle rezervací stolů pro VIP hosty apod.

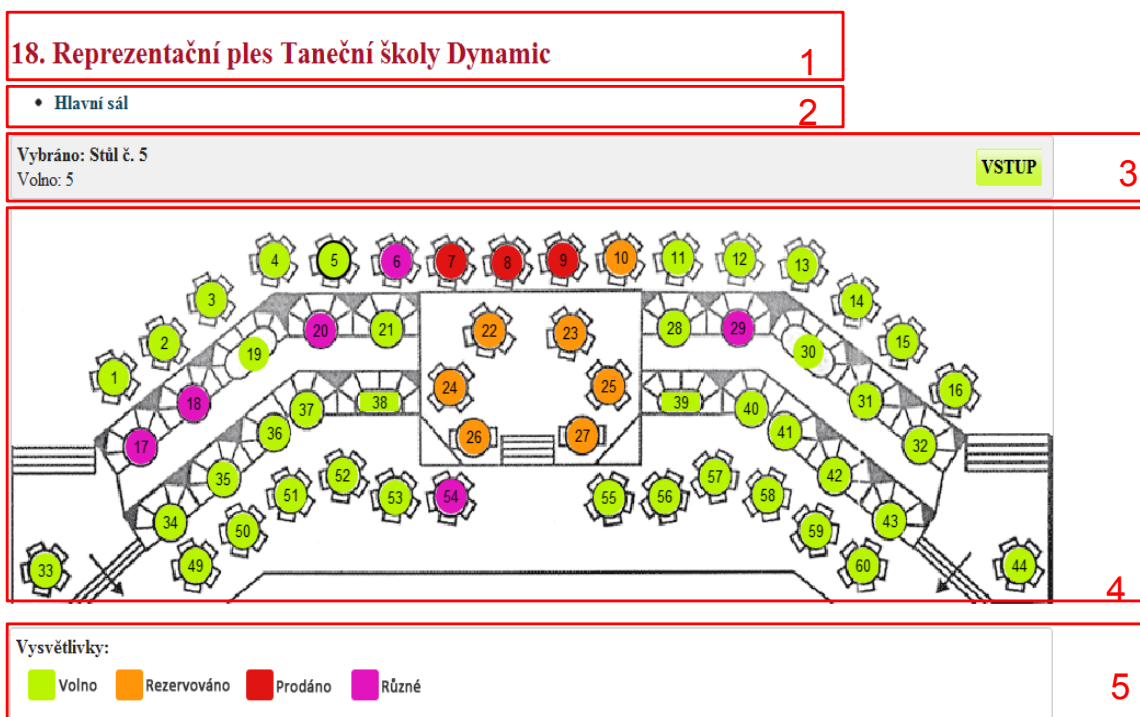
Statistiky jsou dobrým zdrojem informací pro plánování podpůrných reklamních činností události a plánování událostí budoucích. Je to jeden z doplňkových požadavků zadávající firmy.

4.2 Grafická rezervace

Zobrazení grafické verze rezervace umožňuje přesný výběr místa v sálu. Nabízí také globální pohled na obsazenost celého sálu.

4.2.1 Zobrazení místa

Při vstupu do rezervace události je zobrazeno místo konání (v tomto případě „hlavní sál“), vytvořené pořadatelem, jak je možné vidět na Obr. 23 .



Obr. 23 Zobrazení grafického výběru místa.

Grafický výběr místa obsahuje:

1. Název události
 - Oficiální název vybrané události.
2. Výběr zobrazeného sálu
 - Možný výběr pro více sálů v jedné události.
3. Informační panel
 - Panel podávající základní informace o vybraném stole.
 - Umožňuje vstup do rezervace stolu.
4. Grafické zobrazení sálu
 - Zobrazení sálu určeného k rezervaci.
5. Vysvětlivky
 - Informace o barevném rozložení stavů jednotlivých stolů. Status stolu „různé“ znamená, že jednotlivá místa stolu mají různé statusy – některá místa jsou volná, některá rezervovaná apod.

Administrátorem vytvořený plán místa konání události znázorňuje zasedací pořádek v jednotlivých sálech či místnostech. V horním menu rezervace je možné mezi sály přepínat. Každý stůl je vybarven barvou podle jeho stavu, popis těchto barev je vidět dole na Obr. 23 ve vysvětlivkách. Každý stůl je označen identifikačním číslem. Toto číslo je unikátní pro každý stůl v daném sálu.

Výběr stolu je proveden kliknutím myši na příslušný stůl. Poté se stůl označí 2px silným černým ohraničením, jak je možné vidět u stolu číslo 5 na obrázku výše. AJAXovou komunikací se posílá požadavek na zjištění základních informací, které jsou ve výsledku vráceny do Informačního panelu. Při čekání na odpověď serveru je v informačním panelu zobrazen animovaný obrázek načítání. Při rozhodnutí vstoupit do rezervace vybraného stolu je otevřeno vyskakovací okno, jehož parametry budou podrobněji rozebrány v následující kapitole, která je věnována rezervaci místenky.

Data potřebná pro vykreslení všech prvků rezervace jsou přenášena ze serveru k uživateli ve formě JavaScriptového pole, které je deklarováno globálně pod názvem „*arr*“. Toto dvourozměrné pole nesoucí informace o zobrazení všech prvků se skládá v první úrovni z identifikačního čísla stolu a v druhé úrovni, ve vnořeném poli, jsou zahrnuta příslušná data o daném prvku. Pole je vepsáno do HTML stránky při vstupu do rezervace. Dále je s ním pracováno přes AJAXovou komunikaci se serverem.

Pomocí funkce *draw*, kterou je možné vidět v příloze č.2, je pole nesoucí informace o daném zobrazení vykresleno do objektu *Canvas*. V této funkci, do které vstupuje jako argument vykreslovací pole, je proveden cyklus *for* přes všechny prvky pole. V cyklu se určuje:

- ♦ Barva prvku dle statusu stolu,
- ♦ Tvar prvku podle zadání,
- ♦ Souřadnice a velikost prvku,
- ♦ Vykreslení prvku s daným identifikačním číslem.

Při kliknutí myši do objektu *Canvas* se spouští pomocí JavaScriptové události *onclick* funkce „*decideElement*“, která je zobrazena na Obr. 24 . Do funkce vstupují dva parametry: Souřadnice *x* a *y*, na které bylo kliknuto myši. Vyhodnocuje se, zda bylo kliknuto do nějakého elementu, nebo do prázdného místa. V případě nalezení elementu na daných souřadnicích v globálním informačním poli je cyklus zastaven, a je vrácena hodnota identifikačního čísla stolu. V opačném případě vrací funkce hodnotu *false*.

```

1 function decideElement(ClickX, ClickY)
2 {
3     var newX = ClickX;
4     var newY = ClickY;
5
6     for (var i in arr)
7     {
8         var mapStartX = arr[i][1];
9         var mapEndX = Number(arr[i][1]) + Number(arr[i][3]);
10        var mapStartY = arr[i][2];
11        var mapEndY = Number(arr[i][2]) + Number(arr[i][4]);
12        if ( (mapStartX<=newX)
13            && (mapEndX>=newX)
14            && (mapStartY<=newY)
15            && (mapEndY>=newY) )
16        {
17            return i;
18        }
19    }
20    return false;
21 }
```

Obr. 24 Rozhodovací funkce pro výběr rezervačního prvku.

4.2.2 Rezervace místenky

Po vybrání stolu a vstupu do něj se zobrazí vyskakovací okno (Obr. 25), které ukazuje počet míst v daném stole a jejich momentální stav. Pokud byly již provedeny nějaké změny, je u nich uveden údaj o čase a uživateli, který je provedl. U každého jednotlivého místa se vpravo nachází akční menu, kterým vybíráme operaci s daným místem. Po výběru operace se prostřednictvím JavaScriptu zobrazí další nabídka. V případě prodeje je možno zvolit cenu vstupenky, v případě rezervace je zobrazen formulář pro vyplnění jména zákazníka.

Pozice	Stůl č.5	akce	jméno	cena
Pozice 1	Stůl č.5	Rezervovat	Petr Klimeš	
Pozice 2	Stůl č.5	Prodat		Normální 350 Kč
Pozice 3	Stůl č.5	Prodat		Studentská 200 Kč
Pozice 4	Stůl č.5	Prodat		VIP 0 Kč
Pozice 5	Stůl č.5	-		

Obr. 25 Vyskakovací okno pro výběr jednotlivých míst.

Po navolení operací s příslušnými místy pokračujeme tlačítkem v horním pravém rohu „Pokračovat →“. V okně je vykreslena přehledná tabulka pro kontrolu zadaných dat. V případě prodeje je zde uvedena celková cena.

Po potvrzení zkontrolovaných údajů je odeslán požadavek na server pro zpracování a provedení změn. V posledním kroku uživatel obdrží zprávu o úspěšném provedení operace. Při prodeji je navíc zobrazena kalkulačka pro dopočet peněz na vrácení při přijetí platby v hotovosti.

Rezervační okno bylo vytvořeno pouze ze základních funkcí jQuery společně s vlastnostmi CSS. Problémem jQuery pluginů zobrazujících takováto okna je, že se chovají nestandardně a při menších rozlišeních nezobrazují veškeré informace korektně - některé mohou „utíkat“ za obraz. Při návrhu tohoto vyskakovacího okna bylo bráno na zřetel, že systém bude často používán i na malých firemních netboocích, kde bude třeba počítat i s nutností rolování HTML dokumentů v okně prohlížeče.

4.3 Grafický návrhář

Grafický návrhář, který se v administrátorském rozhraní využívá při tvorbě nových míst nebo při úpravě stávajících, umožňuje kreslení vlastních rezervačních míst prostřednictvím webového rozhraní. Není proto nutný zásah grafika nebo programátora při úpravách sálů nebo rezervací nových místností. S tímto nástrojem jsou budoucí tvůrci nových událostí seznámeni a zaškoleni. Pro vykreslování zobrazeného místa se používá stejná funkce jako u rezervačního modulu, a kreslení nových prvků je stejným způsobem zapisováno do dvourozměrného pole v JavaScriptu.

4.3.1 Kreslicí nástroje

Zobrazení grafického návrháře je podobné zobrazení rezervace samotné (Obr. 23) s tím rozdílem, že místo informačního panelu se zobrazuje nástrojová lišta, přes kterou je tvůrce schopen dané místo upravit a vytvořit nové místo dle jeho požadavků. Nástrojovou lištu je možné vidět na Obr. 26 . V liště si nalevo vybíráme nástroj a vpravo se dynamicky zobrazují jeho možnosti.

Nástrojová lišta		Uložit
Popis místa	Kreslení jednotlivých prvků	
Vybrat a upravit	Tvar prvku: čtverec	Hlavní název: Stůl č.
Kreslit	Velikost x: 28	Identifikační číslo: 1
Pozadí	Velikost y: 28	Počet míst: 5

Obr. 26 Nástrojová lišta v grafickém návrháři.

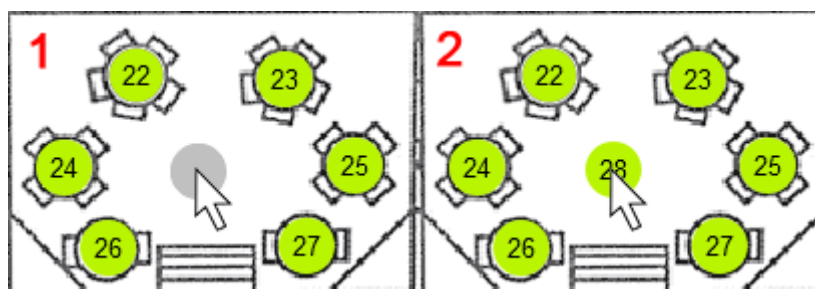
Možnosti jednotlivých nástrojů:

- ◆ Popis místa
 - Zadání názvu místa
 - Zadání velikosti místa
- ◆ Vybrat a upravit
 - Úpravy atributů vybraných prvků
 - Mazání prvků
- ◆ Kreslit
 - Zadání počátečních atributů prvků
 - Kreslení příslušných prvků na dané místo
- ◆ Pozadí
 - Nahrání obrázku na pozadí rezervačního místa.

Při vstupu do grafického návrháře je administrátorovi prvně nabídnut nástroj pro zadání názvu místa a jeho velikosti. Jméno je potřebné pro jeho další zobrazování v systému. Velikost rezervačního místa je zadávána v pixelech („px“), stejně jako ostatní rozměry.

Kliknutím na nástroj „Vybrat a upravit“ je zpřístupněn režim pro úpravu prvků. Při kliknutí do objektu Canvas je spuštěna funkce, která rozhoduje, zda a na který prvek bylo kliknuto, a podle toho se zobrazí informace o daném prvku v nástrojové liště, s možností jejich změny nebo smazání. Při změnách atributů prvků jsou nové informace přepisovány v globálním poli „arr“. Označený prvek je zvýrazněn černým ohraničením.

Nástroj kreslení umožňuje samotné kreslení prvků do objektu Canvas. Při aktivaci tohoto nástroje jsou v možnostech nástrojové lišty zobrazena formulářová pole pro vyplnění informací o novém prvku, jak je možné vidět na Obr. 26. Identifikační číslo stolu je zvětšováno automaticky, je však možné ho zadat i ručně (ovšem při zadání stejné hodnoty, jaká již v daném místě existuje, je zobrazena chybová hláška a je zadáno automatické číslo). Po vyplnění informací a njetí myši na plochu Canvas je pro přehlednost spuštěna funkce „mouseFollower“, která pod myši vytvoří šedý obraz prvku, který má být po kliknutí vykreslen. Jde o jednoduchý element v HTML, kterému jsou zadávány vlastnosti: tvar, velikost a umístění pomocí JavaScriptu. Použití této funkce lze vidět na obr. 29. Tato funkce velmi usnadňuje přehled o vykreslování prvků.



Obr. 27 Zobrazení funkce mouseFollower.

Při kliknutí myši se vytvoří daný prvek se středem v souřadnicích, kde bylo na plochu

Canvas kliknuto. Při vytvoření prvku je skryt „*mouseFollower*“, aby bylo možné nově vykreslený prvek vidět. Není však umožněno křížení prvků, proto při kliknutí myši je spuštěna funkce, která zjišťuje, zda v daném místě již neleží prvek. Pokud ano, je vykreslování zastaveno a je vrácena chybová hláška o křížení. Při úspěšném průchodu kontrolující funkcí jsou informace o prvku převedeny do formátu používaného dvourozměrného pole, kterému jsou následně přiděleny pod svým identifikačním číslem, které určuje hodnotu názvu prvního stupně v používaném poli. O vykreslení se postará vykreslovací funkce „*draw*“ (příloha č. 2), která je volána s parametrem pole čítajícím pouze daný nově nakreslený prvek.

Posledním nástrojem je tlačítko „Pozadí“. V tomto nástroji je možné na pozadí plátna Canvas nahrát obrázek. Tento obrázek slouží k zřehlednění rezervačního místa a snazšímu kreslení nových elementů. Je možné jej po nakreslení rezervačních prvků odstranit a do celého místa konání události neukládat. Pro práci s obrázky jsem využil třídu *simpleImages*[14].

4.3.2 Ukládání údajů

Pro uložení vytvořených, popřípadě upravených míst, je třeba přenést vytvořené dvourozměrné pole, nesoucí informace o zakreslených prvcích, na server, a tam data uložit do databáze. Mechanismus přenosu dvourozměrného pole na server pomocí AJAXu s využitím serializační funkce je detailněji popsán v kap. 3.3.1.

Ukládání je provedeno bez nutnosti znovunačítání celé stránky, proto lze nakreslený návrh dále upravovat a měnit. Tím je uživateli umožněno pohodlné meziukládání návrhů při kreslení rezervačního místa.

```

1 $('#saveLoc').click(function(e) {
2     e.preventDefault();
3     $('#statusField').html(loadingSave()).show();
4     var postArr = serialize(arr);
5     var nameAc = $("#iNamePlace").val();
6     var widthAc = $("#iWidth").val();
7     var heightAc = $("#iHeight").val();
8     $.post("ajax_saveLocation.php",
9     {
10         arr: postArr,
11         edit: edit_id,
12         name: nameAc,
13         width: widthAc,
14         height: heightAc,
15         imgPath: path
16     }, function(data) {
17         $('#statusField').html(data).show(500);
18         $('#statusField').delay(5000).hide(500);
19     });
20 });
21 });

```

Obr. 28 Ukládání vytvořeného místa.

Na Obr. 28 je uveden skript zajišťující ukládání rezervačního místa. Při kliknutí na tlačítko „Uložit“ je do elementu *statusField* vložen obrázek znázorňující načítání. Poté se dvourozměrné pole nesoucí data o zakreslených prvcích převede na řádkový zápis potřebný pro přenos, a vytvoří se nové proměnné, do kterých se ukládají data zadaná do formuláře v nástroji „Popis místa“. Následně je volána metoda *post* pro AJAXový přenos dat na server do PHP skriptu pro zpracování a uložení dat do databáze. Po úspěšném dokončení

AJAXového požadavku je v elementu „*statusField*“ nahrazen animovaný obrázek textem informujícím o úspěšnosti provedené akce. Tento element je po 5 sekundách zneviditelněn, aby zbytečně nemátl uživatele při další práci.

4.4 Optimalizace

Pro rychlý a plynulý chod stránek je potřeba provést optimalizační operace, které zmenší počet přenesených dat ze serveru a zkrátí dobu zpracování skriptu a výpisu dat z databáze.

4.4.1 HTML a grafika

Rychlost načítání stránek je ovlivněna z velké míry počtem přenesených dat ze serveru ke klientovi. Největší objem dat je při přenosu obsažen v grafice dané stránky. Proto je nutné volit obrázky ve vhodném formátu s vyváženou barevnou hloubkou.

Pro zobrazování přechodových ploch je v systému využíváno CSS, a to v podobě 1px širokých opakujících se obrázků.

Pro obrázky kreslené pomocí počítačového softwaru byl zvolen formát .PNG, a pro fotografie byl zvolen formát .JPG. Tímto způsobem bylo dosaženo dobrého vzhledu s přijatelnou velikostí potřebné grafiky.

Velikost HTML kódu je v porovnání s grafickými prvky zanedbatelná, avšak k pomalému zobrazování může docházet i z důvodů složitosti kódu – např. rozsáhlé vnořené tabulky. Pro rychlost zobrazování je doporučeno mít kód validní dle normy w3c, aby nedocházelo k chybám a nestandardním zobrazením stránky, které může systém rovněž zpomalit. Na základě těchto poznatků byl systém navržen tak, že využívá jen minimálně tabulkové zobrazení, a HTML kód je plně validní s nejnovější w3c normou. Celá šablona je tvořena novou sémantikou HTML 5, která napomáhá minimalizovat a zpřehlednit kód.

4.4.2 JavaScript a jQuery

Během celého procesu realizace projektů v JavaScriptu je třeba dbát na jeho optimalizaci. Ta je pro rozsáhlejší projekty, jakým popisovaný rezervační systém bezpochyby je, nutným požadavkem. Následující výčet shrnuje základní pravidla optimalizace JavaScriptu, kterých by se měli vývojáři aplikací a webových stránek držet:

- ◆ Umisťovat odkazy na JavaScripty až na konec elementu `<body>`.
- ◆ Zredukovat počet nahrávaných souborů.
- ◆ Neprovádět nadměrné kalkulace v cyklech *for* a *while*.
- ◆ Používat funkci *switch* namísto funkce *if-Then-Else*.
- ◆ Co nejméně modifikovat DOM struktury.
- ◆ Vyhledávat DOM elementy pomocí *id*.
- ◆ Vyhnout se duplicitnímu vytváření jQuery objektů.

Dodržování těchto základních pravidel nabízí efektivní způsob zrychlení běhu JavaScriptových aplikací. Detailní popis těchto pravidel je možné najít na webu.[18]

Při tvorbě systému byla tato pravidla dodržována. Pro celkové profilování rychlosti načítání webových stránek byl využíván open-source program Firebug [19]. Pomocí tohoto doplňku prohlížeče Firefox byl systém pravidelně během vývoje kontrolován a dle výsledků optimalizován.

4.4.3 Dotazy SQL

Jednu z největších časových prodlev na straně serveru tvoří čtení dat z databáze. Při

špatně formulovaných dotazech na databázi dochází k velké časové prodlevě při odpovědi z databáze, a tím je ovlivněn celý chod systému. Je nutné také brát na zřetel strojový čas serveru, který by mohl být mnoha neoptimalizovanými dotazy zahlcen. Optimalizace SQL dotazů musí být podpořena dobrým návrhem databáze samotné. Základní pravidla optimalizace SQL dotazů jsou následující:

- ◆ Kvalita dotazů nad kvantitou
 - Lepším řešením je spojit tabulky v dotazu a udělat jeden složitější dotaz než-li více jednodušších. Více malých dotazů zpomaluje čtení z databáze.
- ◆ Selekcce dat přímo v databázi
 - Potřebná data vybírat přímo při tvoření dotazů, nežli později přes příkazy ve zpracujících skriptech. Výběr dat v databázi je rychlejší a nemusí se vracet velký objem dat.
- ◆ Vyjmenování sloupců v tabulce
 - Nepoužívat v seznamu sloupců (*), ale vypsát jednotlivé sloupce, které budou používány. Databáze nemusí zjišťovat seznam sloupců tabulky.
- ◆ Nepoužívat klauzuli LIKE
 - Není doporučeno používat tuto klauzuli při vyhledávání ve velkých textových polích. Velká pole pro vyhledávání mohou obsahovat velké množství dat a jejich procházení je potom pomalé.
- ◆ Nepoužívat klauzuli IN, NOT IN
 - Vhodnějším řešením a náhradou je klauzule WHERE, která je rychlejší.
- ◆ Využívat klauzule typu LIMIT
 - Když víme, že máme vybírat pouze extrém nebo určitý počet záznamů, je vhodné použít do dotazu klauzuli LIMIT.
- ◆ Na začátek dávat obecnější podmínky
 - V klauzuli WHERE dávat na začátek podmínky, po kterých ubude ze seznamu nejvíce záznamů. Ušetříme tím strojový čas další podmínky, která bude prováděna na menším počtu záznamů.
- ◆ Vhodné pořadí spojení tabulek
 - Vybírat takové pořadí spojení ze všech možných, aby bylo spojeno co nejméně položek.
- ◆ Nastavení indexů
 - Při vhodném použití indexů na sloupce, podle kterých je často hledáno nebo tříděno, dochází k procházení tabulky v mnohem kratším časovém intervalu.[15]

Dodržování těchto pravidel vede ke značnému zrychlení chodu celého systému. Jedním z nejdůležitějších kroků při optimalizaci větších databázových projektů je vhodné nastavení indexů na nejpoužívanější sloupce. Index představuje určitou část paměťového prostoru, do kterého se uloží informace o rozložení hodnot indexovaných sloupců v tabulce. Pokud později dojde k dotazu na indexovaný sloupec, je vyhledáváno podle indexu, což je mnohem rychlejší než-li hledání dle toho, jak jsou řádky uloženy ve sloupci.

5 ZÁVĚR

Cílem diplomové práce bylo navrhnout a vytvořit rezervační systém vyhovující požadavkům zadavatele - brněnské taneční školy Dynamic.

Rezervační systém byl navržen tak, aby odstranil všechny nedostatky, které byly zmíněny v úvodu práce:

- ♦ umožňuje administrátorovi systému kontrolu nad stavem peněžních prostředků plynoucích z prodeje vstupenek,
- ♦ poskytuje možnost současného prodeje více prodejci a neomezuje administrátora počtem míst a kapacitou systému,
- ♦ nabízí statistické funkce, které pomáhají plánovat propagaci společenských událostí,
- ♦ navíc nabízí administrátorovi možnost samostatného vytváření prostor konání společenských akcí bez potřeby programátora.

Z těchto důvodů je možné cíl diplomové práce považovat za splněný. S využitím technologie AJAX je chod systému rychlejší a srovnatelný s klasickými desktopovými aplikacemi, není nutno čekat na znovunačítání celé stránky po každé provedené akci. Mění se pouze části stránky, které s danou akcí souvisejí.

Systém je v grafické rezervaci a grafickém návrháři díky vykreslování pomocí objektu HTML 5 Canvas nenáročný na strojový čas uživatelského počítače, a umožňuje tak bezproblémový chod systému i na pomalejších počítačích.

Společnost první verzi rezervačního systému již implementovala a na základě oboustranně úspěšné spolupráce se rozhodla tuto spolupráci prohloubit. Společnost již zaznamenala finanční úlevu zejména v oblasti úniku peněz při prodeji vstupenek (využití systému přineslo cca 5% z celkové vybrané částky) a v následné fázi projektu chce přejít k dalšímu kroku, kterým je vlastní správa systému včetně práce v grafickém návrháři.

Při návrhu systému bylo myšleno na možnost budoucího použití více společnostmi. Proto v případě úspěšného provozu bude užívání rezervačního systému nabídnuto dalším společnostem. Koncept tohoto návrhu spočívá ve vytvoření veřejně dostupného webového rozhraní spravovaného hlavním administrátorem. Ten bude přidělovat administrační práva jednotlivým administrátorům odlišných společností. Po vytvoření událostí jedním administrátorem a vytvoření uživatelských profilů prodejců vstupenek budou tito prodejci moci vidět pouze společenské události vytvořené jejich administrátorem. Běžný uživatel internetu se bude moci zaregistrovat pouze s možností rezervace, ale uvidí všechny společenské události (vytvořené různými administrátory), pokud společnost nebude preferovat pouze interní formu rezervačního systému.

Grafický návrhář by v budoucnu mohl být s drobnými úpravami dále využit v jiných praktických aplikacích. Jeho hlavní výhodou je možnost kreslení místa přímo z prostředí webového prohlížeče, přičemž každý nakreslený prvek může nést libovolně nastavitelné informace, které jsou při vstupu na daný prvek možné vyčíst. Díky těmto vlastnostem by se tento nástroj mohl dále využívat např. při uživatelském kreslení webového zobrazení inteligentního domu.

SEZNAM POUŽITÉ LITERATURY

- [1] HUNT, Lachlan. *Seznámení s HTML 5* | *Interval.cz* [online]. 24. 12. 2007 [cit. 2011-03-21]. Dostupné z WWW: <<http://interval.cz/clanky/seznameni-s-html-5/>>.
- [2] *HTML5 & CSS3 Support, Web Design Tools & Support ~ FindMeByIP* ~ [online]. 2011 [cit. 2011-03-21]. HTML5 & CSS3 Support. Dostupné z WWW: <<http://www.findmebyip.com/litmus/>>.
- [3] PILGRIM, Mark. *Dive Into HTML5* [online]. 2009, 2011 [cit. 2011-04-16]. Dostupné z WWW: <<http://diveintohtml5.org/>>.
- [4] *HTML5 Canvas Tutorials* [online]. 2011 [cit. 2011-04-16]. HTML5 Canvas Basic Tutorials Introduction. Dostupné z WWW: <<http://www.html5canvastutorials.com/tutorials/html5-canvas-tutorials-introduction/>>.
- [5] AJAX. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13. 7. 2007, last modified on 22. 3. 2011 [cit. 2011-04-02]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/AJAX>>.
- [6] KOSEK, Jiří. *PHP : Tvorba interaktivních internetových aplikací*. První vydání. Praha : Grada Publishing, spol. s.r.o., 1999. 489 s. ISBN 80-7169-373-1.
- [7] LACKO, Luboslav. *Ajax : Hotová řešení*. Praha : Computer Press, 2008. 272 s. ISBN 978-80-251-2108-5
- [8] KOFLER, Michael; ÖGGL, Bernd. *PHP 5 a MySQL 5 : Průvodce webového programátora*. Brno : Computer Press, a. s., 2007. 607 s. ISBN 978-80-251-1813-9.
- [9] *JQuery.get()* [online]. 2010 [cit. 2011-04-03]. JQuery API. Dostupné z WWW: <<http://api.jquery.com/jquery.ajax/>>.
- [10] LHOTÁK, Jaroslav. *Interval.cz* [online]. 01. 03. 2001 [cit. 2011-04-25]. Pracujeme se session v PHP. Dostupné z WWW: <<http://interval.cz/clanky/pracujeme-se-session-v-php/>>.
- [11] ARPAD, Ray. *Phpjs.org* [online]. 2008 [cit. 2011-04-28]. JavaScript serialize. Dostupné z WWW: <<http://phpjs.org/functions/serialize>>.
- [12] *Adobe Kuler* [online]. 2006, 10.15.2008 [cit. 2011-05-01]. Kuler. Dostupné z WWW: <<http://kuler.adobe.com/>>.
- [13] KRYL, Milan. *Milan Kryl web* [online]. 21.07.2004 [cit. 2011-05-15]. PHP 5 podrobně. Dostupné z WWW: <<http://kryl.info/clanek/134-php-5-podrobne>>.
- [14] JARVIS, Simon. *White Hat* [online]. 08/11/06 [cit. 2011-05-16]. Resizing images with PHP. Dostupné z WWW: <<http://www.white-hat-web-design.co.uk/articles/php-image-resizing.php>>.
- [15] DUŠEK, Roman. 2007 [cit. 2011-05-16]. Optimalizace SQL dotazů. Dostupné WWW: <http://www.fi.muni.cz/~kripac/PV136/dusek/opt_sql_dot.pdf>.
- [16] GRIMMICH, Šimon. *Tvorba webu* [online]. 2008 [cit. 2011-05-22]. DOM: Document Object Model. Dostupné z WWW: <<http://www.tvorba-webu.cz/dom/>>
- [17] Scalable Vector Graphics. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 3. 8. 2006, last modified on 25. 4. 2011 [cit. 2011-05-24]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Scalable_Vector_Graphics>.

- [18] AMBRUŠ, Martin. *Interval.cz* [online]. 17. 02. 2011 [cit. 2011-05-24]. Optimalizácia JavaScript a jQuery kódu. Dostupné z WWW: <<http://interval.cz/clanky/optimalizacia-javascript-a-jquery-kodu/>>.
- [19] *Firebug : Web Development Evolved*. [online]. 2005, 13.05.2011 [cit. 2011-05-24]. Dostupné z WWW: <<http://getfirebug.com/>>.

SEZNAM PŘÍLOH

Příloha č.1- Ukázka šablony v HTML 5

Příloha č.2 - Vykreslovací funkce draw

Příloha 1. Ukázka šablony v HTML 5

```
1 <!DOCTYPE html>
2 <html lang="cs">
3 <head>
4   <meta http-equiv='content-language' content='cs'>
5   <meta charset="UTF-8">
6   <meta name='robots' content='index, follow'>
7   <meta name='description' content="
8   <?php
9     this->saveOnScreen($this->eDescription);
10   ?>">
11   <meta name="author" content="Petr Klimes">
12   <meta name="keywords" content="
13   <?php
14     $this->saveOnScreen($this->eKeywords);
15   ?>">
16   <link rel="shortcut icon" href="./design/images/favicon.ico">
17   <title><?php $this->saveOnScreen($this->eTitle); ?></title>
18   <!-- CSS -->
19   <link rel="stylesheet" href="./design/styles/main.css">
20 </head>
21 <body>
22 <div id="black/one">
23 </div>
24   <header>
25     <?php $this->saveOnScreen($this->eHeader); ?>
26   </header>
27   <nav>
28     <?php $this->saveOnScreen($this->eMainNav); ?>
29   </nav>
30   <article>
31     <?php $this->saveOnScreen($this->eHeadLine); ?>
32     <?php $this->saveOnScreen($this->eSections); ?>
33   </article>
34   <footer>
35     <?php $this->saveOnScreen($this->eFooter); ?>
36   </footer>
37   <!-- JS -->
38   <?php $this->saveOnScreen($this->eScripts); ?>
39
40 </body>
41 </html>
```

Příloha 2. Vykreslovací funkce draw

```
1 function draw (drawArr)
2 {
3     for (var i in drawArr)
4     {
5         switch (drawArr[i][6])
6         { //status
7             case 1: // volno
8                 ctx.fillStyle = "rgb(185,245,0)";
9                 break;
10            case 2: // rezervovano
11                ctx.fillStyle = "rgb(255,150,10)";
12                break;
13            case 3: // prodano
14                ctx.fillStyle = "rgb(255,20,20)";
15                break;
16            case 4: // VIP
17                ctx.fillStyle = "rgb(20,165,255)";
18                break;
19            case 5: // ruzne
20                ctx.fillStyle = "rgb(225,20,190)";
21                break;
22        }
23        switch (drawArr[i][5]) // shape
24        {
25            case 1: // obdelnik
26                ctx.fillRect(drawArr[i][1],
27                    drawArr[i][2],
28                    drawArr[i][3],
29                    drawArr[i][4]);
30                break;
31            case 2: // kruh
32                ctx.beginPath();
33                ctx.arc(drawArr[i][1]+(drawArr[i][3]/2),
34                    drawArr[i][2]+(drawArr[i][3]/2),
35                    drawArr[i][3]/2,
36                    0,
37                    Math.PI*2,true);
38                ctx.closePath();
39                ctx.fill();
40                break;
41        }
42        ctx.font = "10pt Arial";
43        ctx.textAlign = "center";
44        ctx.fillStyle = "rgb(0,0,0)";
45
46        var centerX = (drawArr[i][3]/2) + drawArr[i][1];
47        var centerY = (drawArr[i][4]/2) + drawArr[i][2] + 5;
48        ctx.fillText(drawArr[i][8], centerX, centerY);
49    }
50 }
```