

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Webová aplikace využívající principy MVP
Bakalářská práce

Autor: Kristýna Rehovičová
Studijní obor: Aplikovaná informatika

Vedoucí práce: Malý Filip, doc. Ing. Ph.D.
Odborný konzultant: Jan Slavík
Jacon s.r.o.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 10.11.2017

Kristýna Rehovičová

Poděkování:

Děkuji vedoucímu bakalářské práce doc. Ing. Filipu Malému, Ph.D. za metodické vedení práce, cenné rady a připomínky v průběhu vytváření práce.

Anotace

Tato bakalářská práce se zabývá tvorbou webové aplikace využívající principy MVP za použití PHP frameworku. Projekt odpovídá reálnému zadání a bude nasazen v provozu pro myanmarského klienta firmy Jacon s.r.o.. Jejím cílem je poskytnout plně funkční a uživatelsky přívětivý rezervační systém pro prodej lístků a vstupenek do sedačkových sálů. Součástí aplikace je rovněž rozhraní pro kontrolu pravosti vstupenek při vstupu na danou akci, možnost tvorby nových sálů či vzhledu vstupenek. V začátku rozebereme problematiku MVP architektury a srovnání s jinými přístupy jako MVC a MVVM. Ve druhé části se zaměříme na výběr použitých technologií a zbytek práce bude věnován samotné tvorbě webové aplikace odpovídající požadavkům zákazníka.

Annotation

Title: Web application based on MVP architecture

In this thesis, we will discuss designing web applications which implement the MVP architecture with the assistance of the PHP framework. The project is based on an actual business application that was developed for a Myanmar based company that contracted Jacon s.r.o. to do the work. The goal is to provide a fully functioning and user-friendly ticket booking system for venues that will host events such as concerts, operas, etc. Parts of the app include an interface for validating tickets upon entry to the event, creating your own venue, and designing the appearance of your tickets. First, we will analyze the MVP architecture and compare it to other architectures such as MVC and MVVM. Next, we will explore the technologies chosen for the project. Lastly, attention will be given to the creation of the project and how it meets the customer's requirements.

Obsah

1	Úvod.....	1
2	Cíl práce.....	2
3	Prezentační vzory	3
3.1	MVC	3
3.2	MVP	4
3.3	MVVM	5
4	Výběr technologie.....	6
4.1	Webový framework.....	6
4.2	Client-side jazyk.....	8
5	Analýza a návrh systému.....	9
5.1	Požadavky na systém.....	9
5.2	Návrh systému.....	9
5.2.1	Model typových úloh.....	9
5.2.2	Analytický model tříd	11
5.2.3	Návrhový model tříd	12
5.2.4	Databázová struktura	13
5.3	Nette Framework.....	14
5.3.1	Adresářová struktura	14
5.3.2	Rozšíření	16
6	Implementace	19
6.1	Funkce systému	19
6.1.1	Přihlášení a odhlášení ze systému	19
6.1.2	Hlavní stránka.....	20
6.1.3	Opakující se prvky webu	21
6.1.4	Dealer a uživatel.....	22

6.1.5	Vytváření akcí	24
6.1.6	Prodej vstupenek.....	30
6.1.7	Kontrola vstupenek	34
6.1.8	Další funkce.....	35
6.2	Zabezpečení	37
7	Shrnutí výsledků	38
8	Závěry a doporučení.....	39
9	Seznam použité literatury.....	40
10	Přílohy.....	42

Seznam obrázků

Obr. 1 Schéma MVC.....	3
Obr. 2 Schéma MVP.....	5
Obr. 3 Schéma MVVM.....	5
Obr. 4 Srovnání frameworků z pohledu popularity mezi vývojáři.....	6
Obr. 5 Use case model.....	10
Obr. 6 Analytický model tříd.....	11
Obr. 7 Návrhový model tříd	12
Obr. 8 Databázová struktura.....	13
Obr. 9 Struktura projektu.....	14
Obr. 10 Použití sweet alertu.....	17
Obr. 11 Použití sweet alertu s callbacky na tlačítka	18
Obr. 12 Přihlašovací formulář.....	19
Obr. 13 Informace o přihlášeném uživateli.....	19
Obr. 14 Hlavička webu.....	20
Obr. 15 Detail menu.....	20
Obr. 16 Hlavní stránka.....	21
Obr. 17 Příklad rozložení stránky.....	22
Obr. 19 Formulář pro vytvoření dealera	23
Obr. 19 Formulář pro vytvoření uživatele.....	23
Obr. 20 Formulář pro vytvoření akce	24
Obr. 21 Formulář pro upload podkladu pro layout	25
Obr. 22 Formulář pro nadefinování sálu	25
Obr. 23 Vyplněný formulář pro nadefinování sálu	27
Obr. 24 Ace editor pro vytvoření šablony vstupenky	28
Obr. 25 Seznam Povolených zástupných klíčů.....	28
Obr. 26 Ukázka kódu pro tisk vstupenek	29
Obr. 26 Tlačítko pro prodej	30
Obr. 27 Ukázka formuláře pro prodej vstupenek.....	30
Obr. 28 Sweet Alert zobrazující kód rezervace.....	31
Obr. 29 Tabulka s výčtem označených sedadel	32

Obr. 30 Sweet Alert – vstupenky.....	32
Obr. 31 Ukázka tisku vstupenek.....	33
Obr. 32 Stavby sedadel.....	33
Obr. 33 Validace vstupenky.....	34
Obr. 34 Možné výsledky kontroly kódu vstupenky	34
Obr. 35 Formulář pro export reportu	35
Obr. 36 Formulář pro vytváření rolí.....	36

Seznam tabulek

Tab. 1 Ukázkový příklad reportu	35
---------------------------------------	----

1 Úvod

Tématem bakalářské práce je tvorba webu na zakázku. Důvodem výběru byla možnost pracovat na reálném zadání, které vytvořila firma Forever Group, jakožto přední mediální společnost ve státě Myanmar (dříve Barma). Jedná se o rezervační systém, umožňující přímý prodej nebo rezervaci a následný prodej vstupenek na nejrůznější akce – kino, divadlo, představení, opery a nespočet dalších. Aplikaci budou využívat v místech určených pro prodej vstupenek. Nebude tedy veřejnosti běžně dostupná. Jelikož není předem známo kde a na jakých místech se budou akce konat, je třeba začlenit do návrhu aplikace také tvorbu sálů a samozřejmě i vzhledu vstupenek. Vedle požadavku na samotný prodej, stojí také požadavek na řešení kontroly tištěných vstupenek při vstupu na danou akci.

2 Cíl práce

Cílem této práce je vytvořit stabilní rezervační systém, který bude odpovídat potřebám zákazníka. Zároveň musí být co nejjednodušší a přehledný. Kvůli omezené rychlosti internetu, kterému bude projekt vystaven, je také třeba optimalizovat všechny databázové dotazy. Důležité je zdůraznit, že web nebude veřejný, ale bude sloužit pouze osobám, pověřeným k prodeji vstupenek.

Každé akce bude obsahovat informaci o layoutu sálu a vzhledu vstupenek. To vše je potřeba vytvářet v aplikaci. Zákazník bude mít možnost si například přes telefonu zarezervovat vstupenky a následně do dvou hodin od rezervace zaplatit na pokladně. Kromě prodeje musí systém umět kontrolovat vstupenky na již probíhajících akcích pomocí čtečky qr/bar kódu nebo vložením desetimístného kontrolního kódu. Mezi další požadavky patří možnost spravovat role v systému podle potřeb nebo stahovat reporty z prodeje. Zákazník nezadal žádná kritéria k designu, tudíž je zde prostor pro vlastní kreativitu.

Projekt by mohl pomoci seskupit kulturní akce na jedno místo a zákazník by tak měl větší přehled o dění v dané lokalitě.

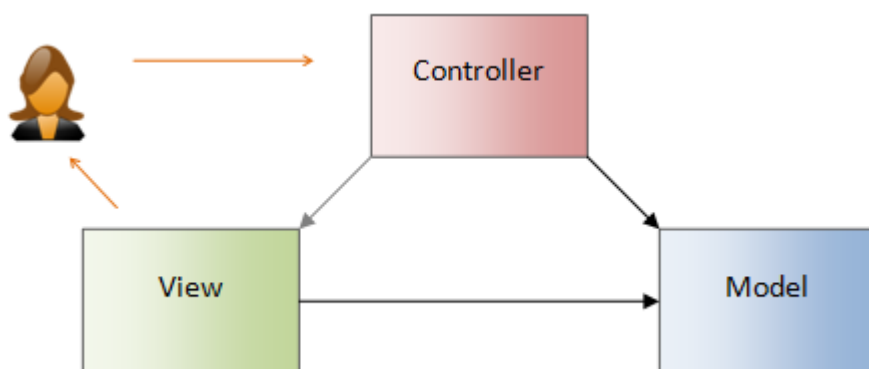
3 Prezentační vzory

Projekt bude postaven na architektuře MVP¹. Základní principy MVP lze vysvětlit a pochopit za poměrně krátký čas, nicméně pochopit celý koncept do detailu zabere spoustu času a úsilí. Vedle architektury MVP se můžeme setkat ještě s řadou jiných přístupů. Mezi ně patří například MVC² či MVVM³.

3.1 MVC

Poprvé tuto architekturu popsal norský profesor Trygve Reenskaug v 70. letech minulého století. Architektura MVC dělí aplikace na 3 základní části:

- Model – reprezentuje data
- View – neboli pohled, se stará o vykreslení dat z modelu
- Controller – má na starosti aplikační logiku a reaguje na události



Obr. 1 Schéma MVC

Zdroj: <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>

¹ Model View Presenter

² Model View Controller

³ Model View ViewModel

Model tedy drží, získává a přepočítává veškerá data na základě požadavku controlleru. Controller poté předá zprávu view, aby aktualizovala data z modelu. Z předchozího schématu (Obr. 1) lze tuto závislost odvodit.

Controller má přímou vazbu na model, aby mohl upravovat data, se kterými model pracuje. View má také vazbu na model, aby mohl aktualizovat data na obrazovce. Různé variace MVC se těmito vazbami mírně liší. Občas narazíme i na verzi, kde controller má vazbu na pohled nebo naopak. Co se ale nesmí v konceptu Model-View-Control vyskytovat, je přímá vazba modelu na view nebo controller.

V prostředí webu může tok událostí vypadat následovně:

1. Uživatel vykoná akci
2. Controller akci zachytí
3. Aktualizuje data v modelu
4. Informuje view o změnách
5. View se dotáže modelu na změny
6. View zobrazí změny

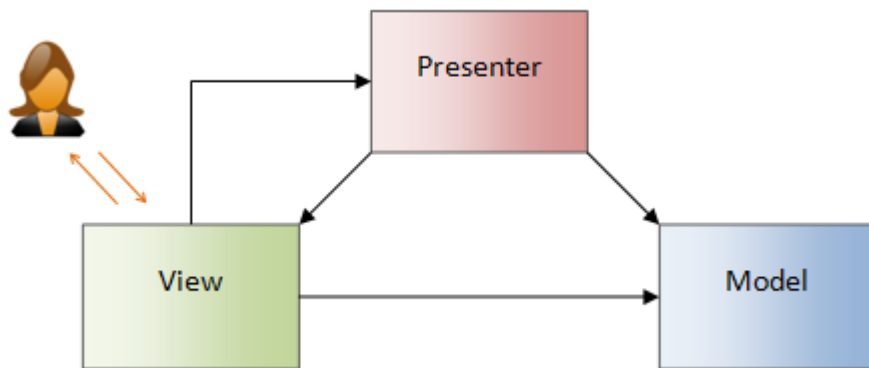
Ve většině frameworků založených na modelu MVC je přítomna ještě nepřímá vazba controlleru na view. V tomto případě se jedná pouze o to, že controller rozhoduje, který pohled zobrazit.

3.2 MVP

Návrhový vzor MVP vznikl v 90. letech a vychází z MVC. Rozdíl není pouze v posledním písmenku, které označuje presenter, ale také ve vazbách na jednotlivé komponenty. Stejně jako u MVC, zde máme tři komponenty.

- Model – reprezentuje data
- View – se stará o vykreslení dat z modelu a reaguje na události

- Presenter – má na starosti aplikační logiku a zajišťuje aktualizaci view



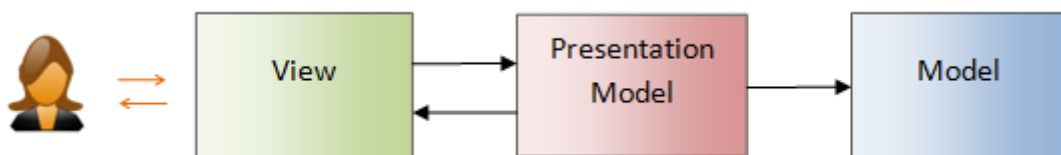
Obr. 2 Schéma MVP

Zdroj: <https://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>

Rozdílů oproti MVC je hned několik. Například view kontroluje i uživatelský vstup, což v běžné praxi znamená, že reaguje na uživatelské akce. Dalším rozdílem je, že view má přímou vazbu na presenter a zobrazuje pouze data, která mu presenter předá.

3.3 MVVM

MVVM se řídí podobnými principy jako MVC nebo MVP. Jeho hlavním rozdílem je chybějící vazba view na model. Toto chybějící spojení se nahrazuje vrstvou, která kombinuje presenter a model – View Model (také Presentation Model). Tato mezivrstva notifikuje view o změnách.



Obr. 3 Schéma MVVM

Zdroj: <https://www.zdrojak.cz/clanky/alternativy-k-mvc-a-zaverecne-poznamky/>

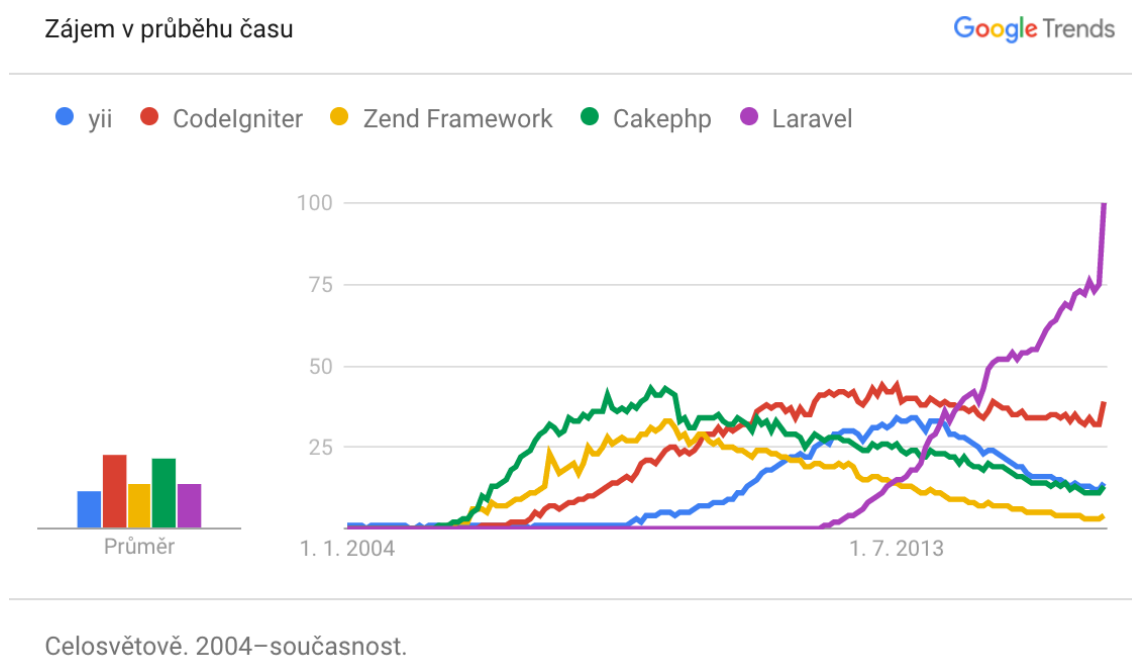
- Model – reprezentuje data
- View – se stará o vykreslení dat z view modelu
- View Model – má na starosti aplikační logiku a zajišťuje aktualizaci view

4 Výběr technologie

Výběr programovacích jazyků hraje velkou roli při vývoji aplikace. Je nutné si prvně rozmyslet, jaké funkce budou v projektu zahrnuty a jakým způsobem je možné je řešit. Správně zvolené technologie mohou také významně ulehčit práci.

4.1 Webový framework

Prvně bylo třeba vybrat základní stavební kámen webu, a to PHP framework založen na MVP. Při výběru frameworku hrál hlavní roli především výkon, dokumentace a aktivní podpora ze stran vývojářů. Tyto vlastnosti ale dle oficiálních stránek má úplně každý framework. Otázkou tedy je, který Framework vybrat.



Obr. 4 Srovnání frameworků z pohledu popularity mezi vývojáři

Zdroj:

<https://www.google.com/trends/explore?date=all&q=yii,CodeIgniter,Zend%20Framework,Cakephp,Laravel&hl=en-US>

Dle průzkumu firmy Google se raketovou rychlostí dostal do vedení framework Laravel. Je ale těžké odhadnout, jestli je to jen programátorský trend, nebo zda se

na první příčce udrží. Jeho hlavní nevýhodou je absence bulit-in funkcí, kterými disponují ostatní konkurenti. (1)

Na druhém místě se umístil Codeigniter, který v posledních letech spíše upadá. Codeigniter vznikl roku 2006, od té doby PHP prošlo velkými změnami, ale CI nestačilo plně zareagovat. (2) Je tedy zřejmé, že pro jistotu dlouhodobé podpory jsou lepší možnosti.

Podobnou popularitu mají Yii a Cakephp. Oba tyto frameworky by se daly nazvat zlatý středem, nemají žádný výrazný nedostatek, ale ani v ničem nevynikají.

Na posledním místě se umístil Zend Framework. Zend bývá často kritizován za výkon (nový Zend Framework 3 slibuje až čtyřikrát větší zrychlení) (3), ačkoliv jeho robustnost, komplexnost a délka působení na trhu ho dělají silným. V případě většího projektu a jistoty rychlého připojení k internetu by Zend framework nebyl špatnou volbou. Rychlost je však číslo 1 na seznamu požadavků.

Ted' na chvíli opustíme srovnávací graf (Obr. 4) a podíváme se na Nette Framework. Nette Framework je český open source projekt pro tvorbu webových aplikací. Jeho přední vlastností je rychlost. Podpora Nette je jak v češtině, tak angličtině. Rychle reaguje na změny a moderní přístupy v programování. Chlubí se dokonalým zabezpečením nebo také bezkonkurenčními ladícími nástroji. (4) Z průzkumu na stránce *www.sitepoint.com* vyšel Nette jako třetí nejoblíbenější framework (nejvíce hlasů samozřejmě dostal ze strany Čechů a Slováků) (5).

Z celého porovnávání frameworků máme dva vítěze: Laravel a Nette. Je těžké se rozhodnout a vybrat nejlepší, každý má svá pro a proti. O něco starší Nette je však přesvědčivější o trochu více a taky je třeba dát šanci českým projektům.

4.2 Client-side jazyk

Pro kostru celého webu byl zvolen jazyk PHP, který komunikuje se serverem. Nicméně žádný dynamický web se neobejde bez client-side programovacího jazyka. Na výběr je opět celá škála nejrozličnějších programovacích jazyků. Zde už sáhneme po jednom z nejrozšířenějších jazyků na internetu vůbec – Javascript. Přesněji řečeno, využijeme jeho knihovnu jQuery. Každá jeho verze je ke stažení v jednom jediném souboru s velikostí pouze do 40kB. Dělá Javascript jednodušším a jeho hlavní filosofií je oddělit chování od HTML.

5 Analýza a návrh systému

Nejprve budou prezentovány požadavky na systém, poté návrh systému a vybraný PHP framework.

5.1 Požadavky na systém

Aplikace musí být postavena na PHP, aby v případě potřeby mohla firma dělat drobné změny. Ze stejného důvodu je preferovaná i databáze MySQL. Všechny databázové dotazy musí být optimalizované z důvodu kolísavého připojení k internetu.

Důležitým požadavkem jsou pravidelné zálohy dat a také případná rychlá obnova ze záloh. Z důvodu bezpečnosti musí být web dostupný pouze z firemní sítě. Systém nebude mít pevně stanovené role uživatelů. Přes seznam všech možných akcí si firma bude moci spravovat a vytvářet role dle aktuálních potřeb.

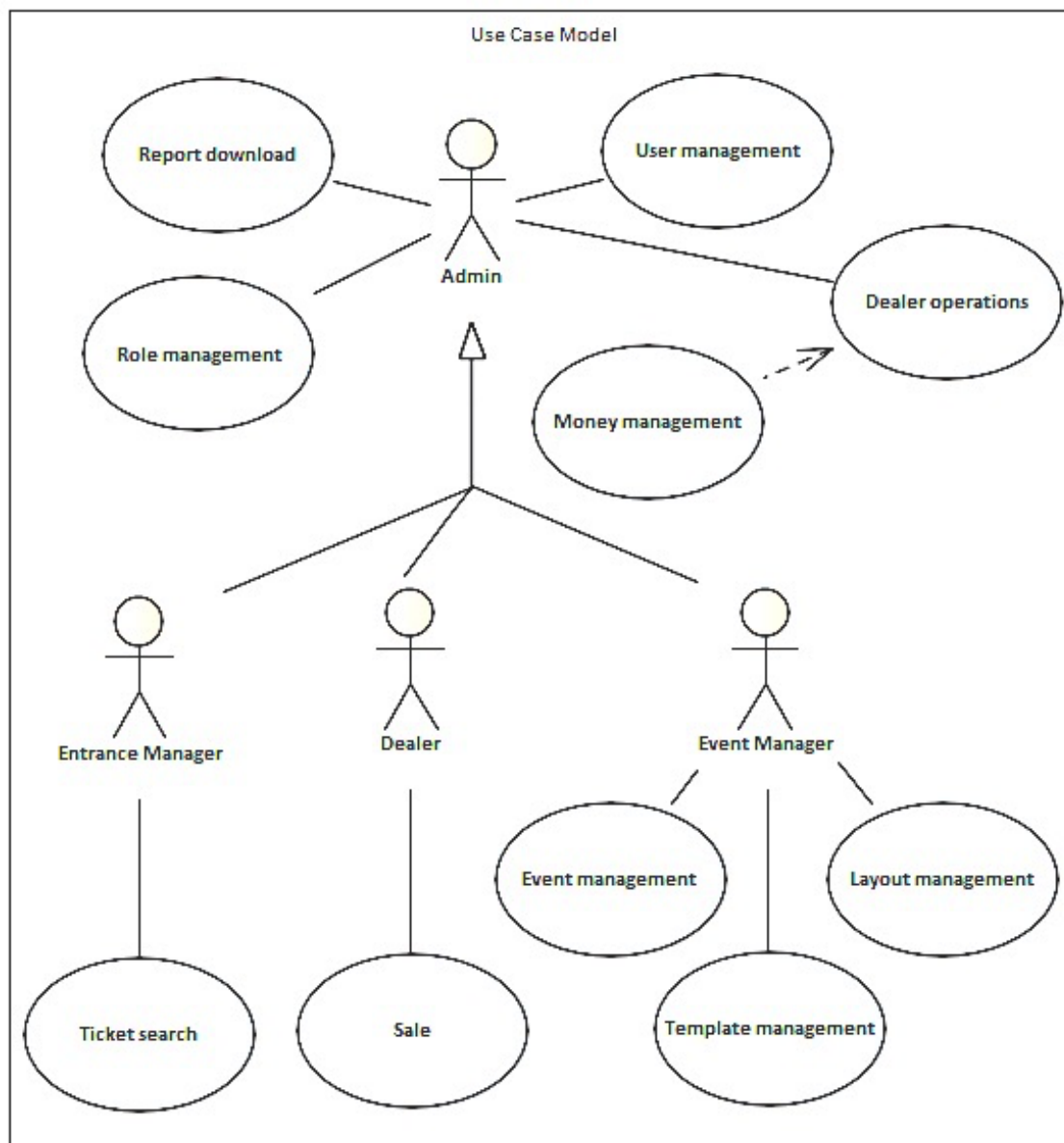
Webové stránky musí co nejintuitivnější a přehledné, aby člověk i bez návodu byl schopný systém ovládat. Zároveň musí být zajištěna kompatibilita se všemi prohlížeči.

5.2 Návrh systému

Návrh je realizován v jazyce UML za pomoci programu Enterprise Architect.

5.2.1 Model typových úloh

Model typových úloh je tvořen na základě testovacích rolí v systému. Zobrazení jsou čtyři aktéři.



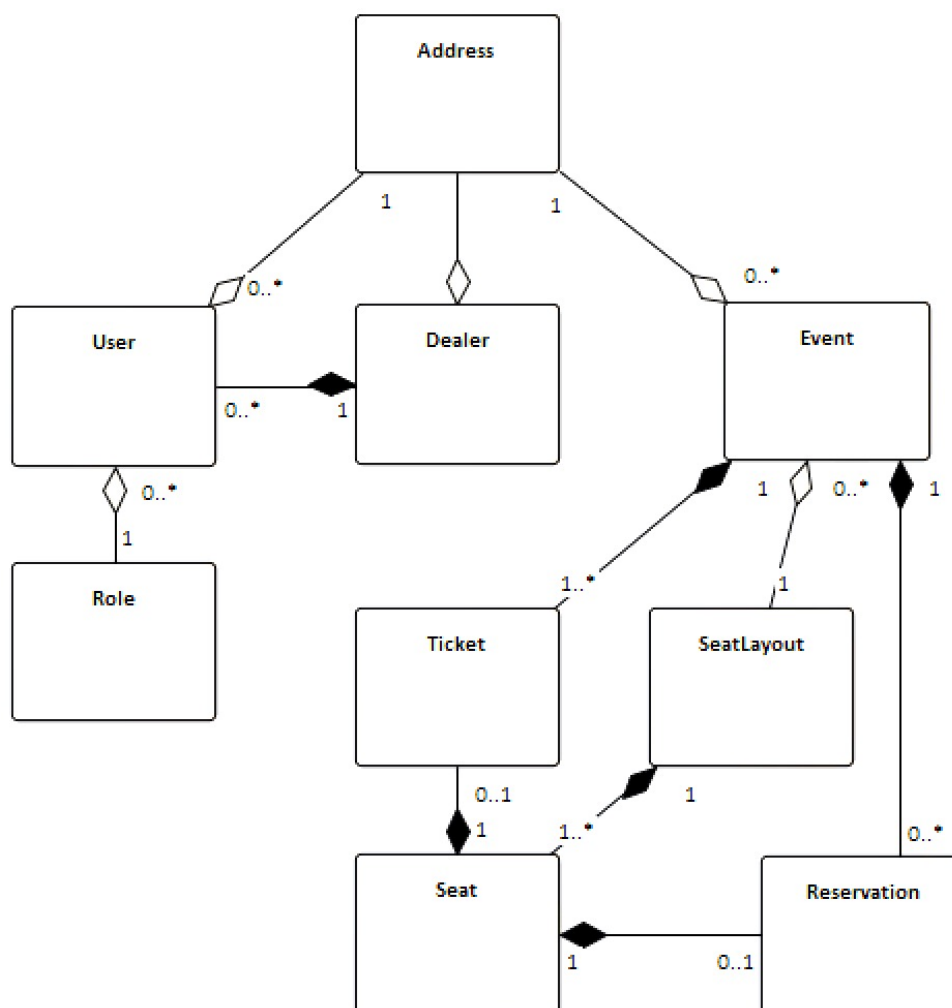
Obr. 5 Use case model

Zdroj: autor

Dealer – prodejce – má právo pouze na prodej vstupenek. *Entrance Manager*, jakožto člověk, který kontroluje platnost vstupenky při vstupu, může pouze vyhledat listky. Větší práva má *Event Manager*, jenž je zodpovědný za vše, co se kulturních akcí týče – tvorba sálů, design vstupenek a správa veškerých akcí. *Admin* má práva na veškeré zmíněné aktivity, a navíc může spravovat uživatele, dealery a role nebo stahovat reporty.

5.2.2 Analytický model tříd

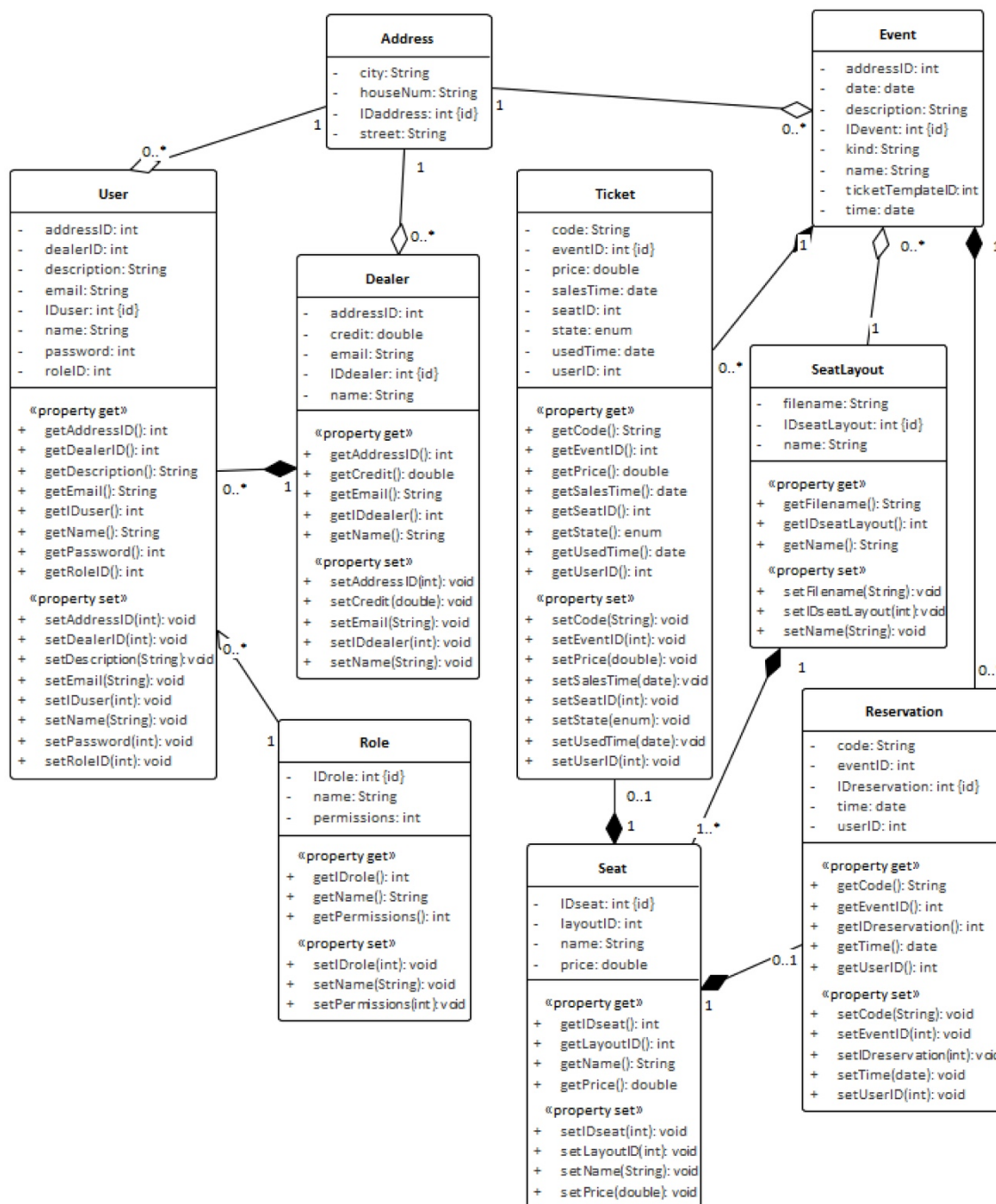
Zobrazuje základní strukturu systému a vazby mezi jednotlivými prvky.



Obr. 6 Analytický model tříd
Zdroj: autor

5.2.3 Návrhový model tříd

Na obrázku níže je zobrazen návrhový model tříd. Vychází z analytického modelu a je obohacen o potřebné atributy a metody ze scénářů.



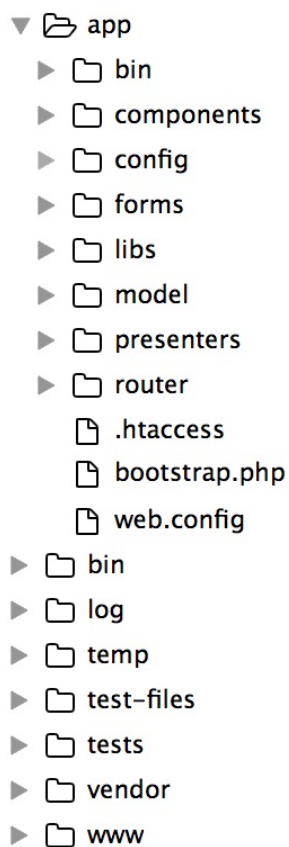
Obr. 7 Návrhový model tříd
Zdroj: autor

5.3 Nette Framework

Nette Framework byl již představen v kapitole 4.1. V následujících řádcích bude nastíněna především struktura a rozšíření projektu.

5.3.1 Adresářová struktura

Následující obrázek (Obr. 9) ukazuje adresářovou strukturu projektu v Nette Frameworku. Zajímavostí je, že tato adresářová struktura není povinná a projekt od projektu se může lišit. (6)



Obr. 9 Struktura projektu
Zdroj: autor

- **app**
Adresář webové aplikace. Zde se nachází aplikační logika projektu. Obsahuje další složky, které rozdělují model, view a presenter.
 - config
Obsahuje konfigurační soubory a nastavení. Např.: připojení k databázi, registrace services, nastavení session a další.
 - forms
V této složce nalezneme všechny třídy obsluhující formuláře.
 - libs
Slouží k umístění knihoven a jiných frameworků, které nestáhneme přes composer.
 - model
Uchovává jednotlivé modely.
 - presenters
Tento adresář obsahuje veškeré presentery a rovněž složku templates, ve které se nachází views jednotlivých presenterů.
 - router
Obsahuje informace o routování.
- **log**
Uchovává veškerá chybová hlášení. U této složky je nutné nastavit práva k zápisu.
- **temp**
Zde se nachází různá dočasná data – například cache. Stejně jako u adresáře log i zde musíme nastavit práva k zápisu.
- **vendor**
Obsahuje samotné Nette a další podpůrné knihovny stažené přes composer
- **www**
Je jediným adresářem přístupným z internetu. Obsahuje obrázky, JavaScript soubory nebo také kaskádové styly.

5.3.2 Rozšíření

Všechna použitá rozšíření výrazně usnadňují práci při psaní kódu. Z přechozí kapitoly je zřejmé, že pro umístění rozšíření se mohou použít dva adresáře – `libs` a `vendor`.

5.3.2.1 Rozšíření Nette

Ve složce `libs` se nachází pouze jedna pomocná třída, a to *CsvResponse*, který má na starosti parsování CSV.

Oproti tomu adresář `vendor` obsahuje rozšíření více. Jedním z nich je *Adminer*, šikovný nástroj pro správu databáze. V mnoha ohledech převyšuje známý *phpMyAdmin* a to především výkoností, možnostmi nastavení, uživatelským rozhraním i jednoduchou instalací. Design Admineru se dá také velice jednoduše měnit, neboť jde pouze o jeden CSS soubor.

Mezi další použitá rozšíření se řadí knihovna pro čtení QR a čárového kódu. Práce s nimi je velice jednoduchá nevyžaduje žádná nastavení pro jejich použití.

Posledním rozšířením je balíček *Nextras*. Za názvem *Nextras* se skrývají dva čeští vývojáři, kteří vyvíjí různé komponenty především pro *Nette Framework*. V aplikaci jsou použity tři komponenty:

- **DataGrid**
Umožňuje vytvořit přehlednou tabulku s možností řazení nebo filtrování.
- **LatteMacros**
Doplněk k *DataGridu*. Pomáhá utvářet vzhled tabulky.
- **Forms**
Přidává k běžným formulářovým prvkům `date picker` a `datetime picker`.

5.3.2.2 Front-end rozšíření

Mezi častá rozšíření mnoha webových vývojářů patří framework *Bootstrap*. Jedná se o nejrozšířenější HTML, CSS a JS Framework pro vývoje webových aplikací. (7)

Jednotlivé komponenty jsou dostupné také samostatně. Projekt využívá dvě bootstrapové komponenty: `glyphicons` a `datetime picker`.

`Glyphicons` je sada ikon dostupných na <http://glyphicons.com/>. Za normálních okolností nejsou ikony zdarma dostupné, nicméně jejich tvůrce poskytl Bootstrapu výjimku. (8)

Na Bootstrap navazuje i další front-end rozšíření, kterým je `Sweet Alert`. Je to balíček CSS a JS souborů, který řeší zobrazování user-friendly alertů, neboli dialogů. Je kompatibilní s Bootstrapem a jeho použití je velice jednoduché a intuitivní.

```
swal({
  title: "Are you sure?",
  text: "You won't be able to recover the delaeer again!",
  type: "warning",
  showCancelButton: true,
  confirmButtonColor: "#DD6B55",
  confirmButtonText: "Yes, delete!"
});
```

Obr. 10 Použití `sweet alertu`
Zdroj: autor

```

swal({
  title: "Are you sure?",
  text: "You won't be able to recover the delaeer again!",
  type: "warning",
  showCancelButton: true,
  confirmButtonColor: "#DD6B55",
  confirmButtonText: "Yes, delete!"
},
function (isConfirm) {
  if (isConfirm) {
    $.ajax({
      url:link.attr('href'),
      success:function(data){
        swal("Success!", "Task completed!", "success");
      },
      error:function(data){
        swal("Oops", "Something went wrong.", "error");
      }
    });
  }
});

```

Obr. 11 Použití sweet alertu s callbacky na tlačítka
Zdroj: autor

Tento kód varuje uživatele před smazáním jiného uživatelského účtu. Uživatel má na výběr buď to potvrdit smazání, nebo zrušit akci. Po potvrzení smazání se zobrazí další dialog, informující o úspěchu nebo neúspěchu akce.

Všechna tyto rozšíření utváření kompletní sadu nástrojů potřebnou pro vývoj zadané aplikace.

6 Implementace

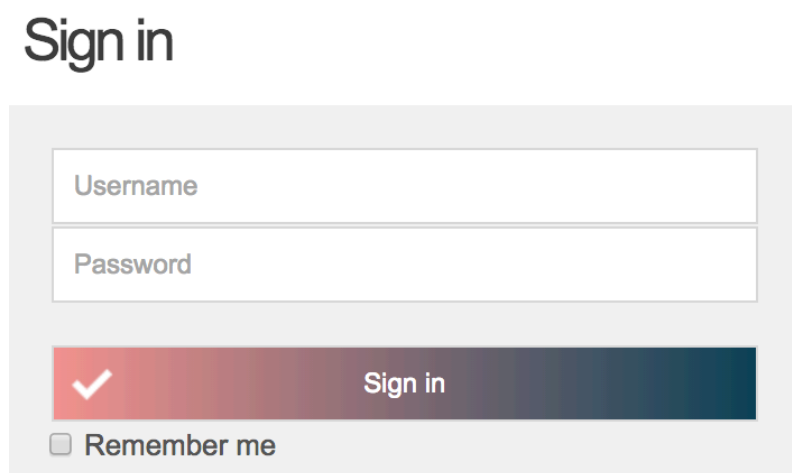
Na základě předchozí kapitoly je možné přejít k samotné implementaci.

6.1 Funkce systému

Nyní budou představeny hlavní funkce systému. Od přihlášení až po samotný prodej vstupenek nebo nastavení uživatelských práv.

6.1.1 Přihlášení a odhlášení ze systému

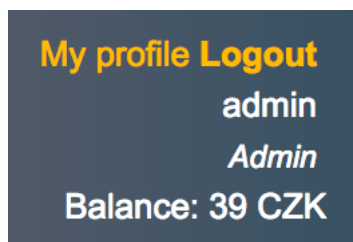
Při každém požadavku na stránku se kontroluje, zda je uživatel přihlášený a systém poté vyhodnotí, kam uživatele přesměruje. Pokud uživatel není přihlášený, pak se automaticky zobrazí přihlašovací formulář.



The image shows a 'Sign in' form. At the top, the text 'Sign in' is displayed in a large, dark font. Below this, there are two input fields: 'Username' and 'Password'. The 'Username' field is on top, and the 'Password' field is below it. Both fields have a light gray border and a white background. Below the input fields is a large, dark blue button with a white checkmark icon on the left and the text 'Sign in' in white. Below the button is a checkbox labeled 'Remember me'.

Obr. 12 Přihlašovací formulář
Zdroj: autor

Po přihlášení se uživatel dostane na hlavní stránku, kde kromě jiného uvidí v hlavičce webu informace o přihlášeném uživateli.

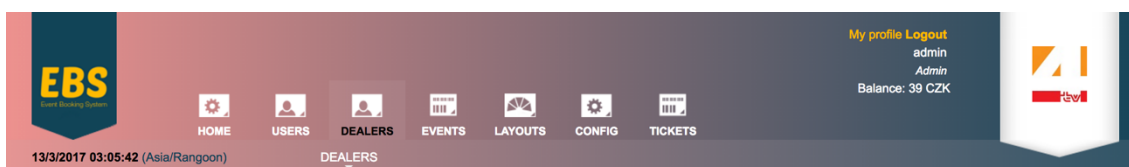


Obr. 13 Informace o přihlášeném uživateli
Zdroj: autor

V první řádce se vyskytují dva odkazy. My profile zobrazí detailnější informace o uživateli a logout uživatele odhlásí ze systému. Po odhlášení je přesměrován na přihlašovací stránku. Na druhé řádce je pak zobrazeno jméno uživatele, v tomto případě admin. Níže je možno vidět uživatelskou roli – Admin. A jako poslední je zobrazen číselný údaj, který udává kolik kreditu má dealer, pod kterým je uživatelský účet vytvořen.

6.1.2 Hlavní stránka

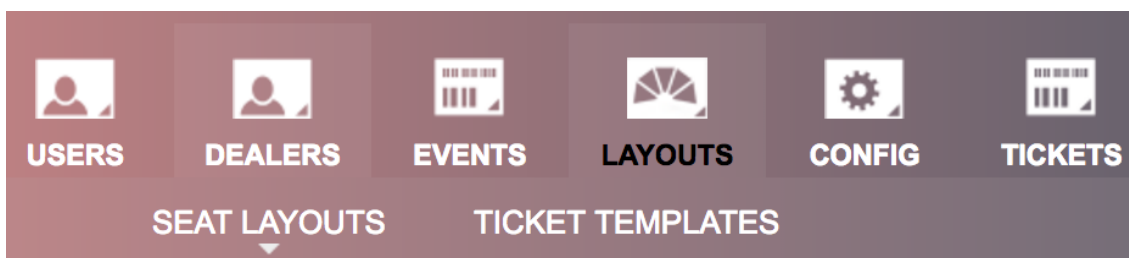
Jako první je možno si všimnout barevné hlavičky, která obsahuje logo systému, čas a časovou zónu, menu, submenu, informace o uživateli (Obr. 14) a také logo firmy (zprava).



Obr. 14 Hlavička webu

Zdroj: autor

Množství odkazů v menu se může měnit v závislosti na roli uživatele. Na obrázku (Obr. 15) jsou vidět všechny možnosti, které mohou být zobrazeny. Každá položka může rovněž obsahovat další menu neboli submenu, které se zobrazuje pod hlavním menu.



Obr. 15 Detail menu

Zdroj: autor

Pod hlavičkou webu se nachází informace o akcích, rozdělené do tří skupin:

1. Actual events
2. Last chance
3. Sold out

Actual Events

Event	Date & Time ▲	Available	VIP	Sold
Example	Mar 12, 2017 17:45	1823	0	777

Last Chance!

Event	Date & Time ▲	Available	VIP
Example One	Mar 15, 2017 10:20	3	0
Example Two	Mar 14, 2017 12:12	3	0

Sold Out

Event	Date & Time ▲
Example Three	Mar 13, 2017 20:20
Example Four	Mar 14, 2017 16:30

Obr. 16 Hlavní stránka
Zdroj: autor

V první skupině je možno vidět všechny akce, které proběhnou v následujících pěti dnech od dnešního dne včetně. Druhá skupina – poslední šance – zobrazuje rovněž všechny akce v následujících pěti dnech, navíc ale vybírá pouze ty, které mají počet volných sedadel rovno deset nebo menší. Na konci jsou vidět již vyprodané akce.

6.1.3 Opakující se prvky webu

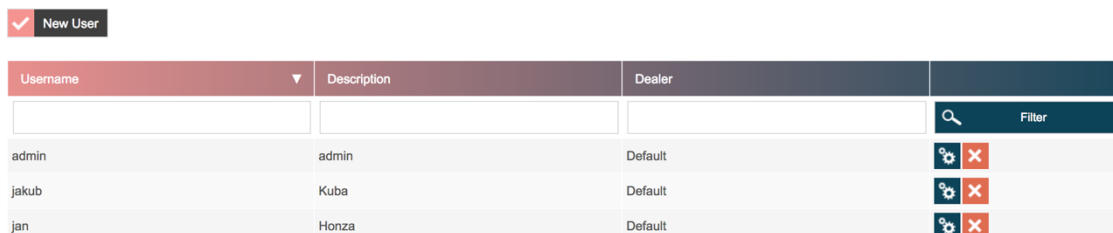
Než bude možno přejít dál, je třeba vysvětlit funkcionalitu opakujících se prvků.







Na každé zbylé stránce je rozložení následující:

1. Nadpis
2. Tlačítko pro vytvoření určité entity
3. Datagrid

Pod nadpisem je tedy vždy tlačítko, umožňující přidat konkrétní entitu do databáze. Pod ním jsou v datagridu zobrazeny veškeré entity konkrétní tabulky. Na obrázku níže (Obr. 17) je zobrazena stránka *USERS*.

Users



Username	Description	Dealer	
admin	admin	Default	 
jakub	Kuba	Default	 
jan	Honza	Default	 

Obr. 17 Příklad rozložení stránky
Zdroj: autor

Na webu jsou také tři tlačítka, která jsou součástí každého datagridu. Jejich vzhled i funkce jsou vysvětleny níže. Umístění jednotlivých ikon je vidět na příkladu rozložení stránky *USERS* (Obr. 17).



Slouží pro editaci. Po jeho stisknutí se objeví formulář s daty.



Vymaže danou entitu.



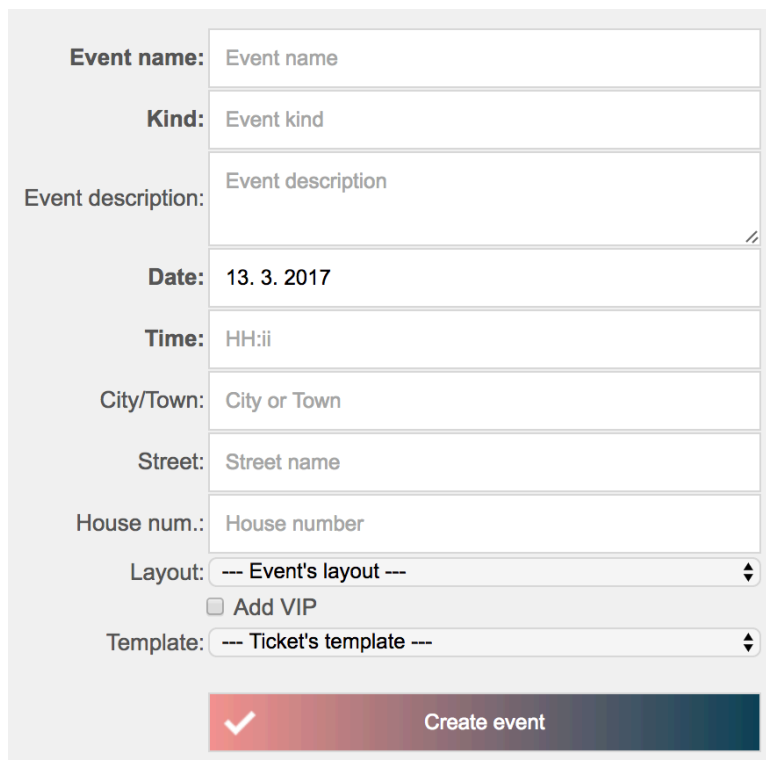
Zahájí filtrování uvnitř datagridu.

6.1.4 Dealer a uživatel

Pro vytvoření dealera a usera slouží dva jednoduché formuláře. Dealer není brán jako uživatel, nemůže se do systému přihlásit, a tudíž nemá ani žádná práva. Jeho formulář je tedy kratší než uživatelův.

6.1.5 Vytváření akcí

Core funkcí celého systému je vytváření akcí a prodej vstupenek. Akce se vytváří za pomoci formuláře, který má kromě obyčejných textových polí také dva selectboxy – *Event's layout* a *Ticket's template*. Jedná se o výběr sálu a šablony pro vzhled vstupenek, která je požadovaná při tisku. Tyto údaje nejsou povinné, avšak bez vyplnění není možné provádět prodej.



Obr. 20 Formulář pro vytvoření akce

Zdroj: autor

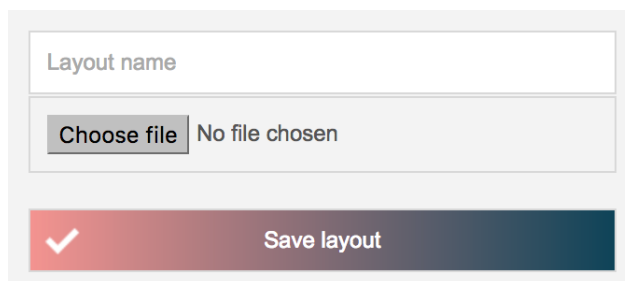
Dalším zajímavým údajem je checkbox *Add VIP*. Při vybrané možnosti *Add VIP* se po uložení akce, a vytvoření sedačkového layoutu, zobrazí uživateli layout, ve kterém je možné vybírat VIP sedadla.

6.1.5.1 Tvorba sálů

Nedílnou součástí vytvoření akce je rovněž vytvoření sálu. Tento krok je složitější. Celý proces je rozdělen do dvou fází:

1. Upload podkladu
2. Vložení rozložení sedaček

V prvním kroku stojí pouze jednoduchý formulář pro nahrání podkladu a pojmenování pro layout.



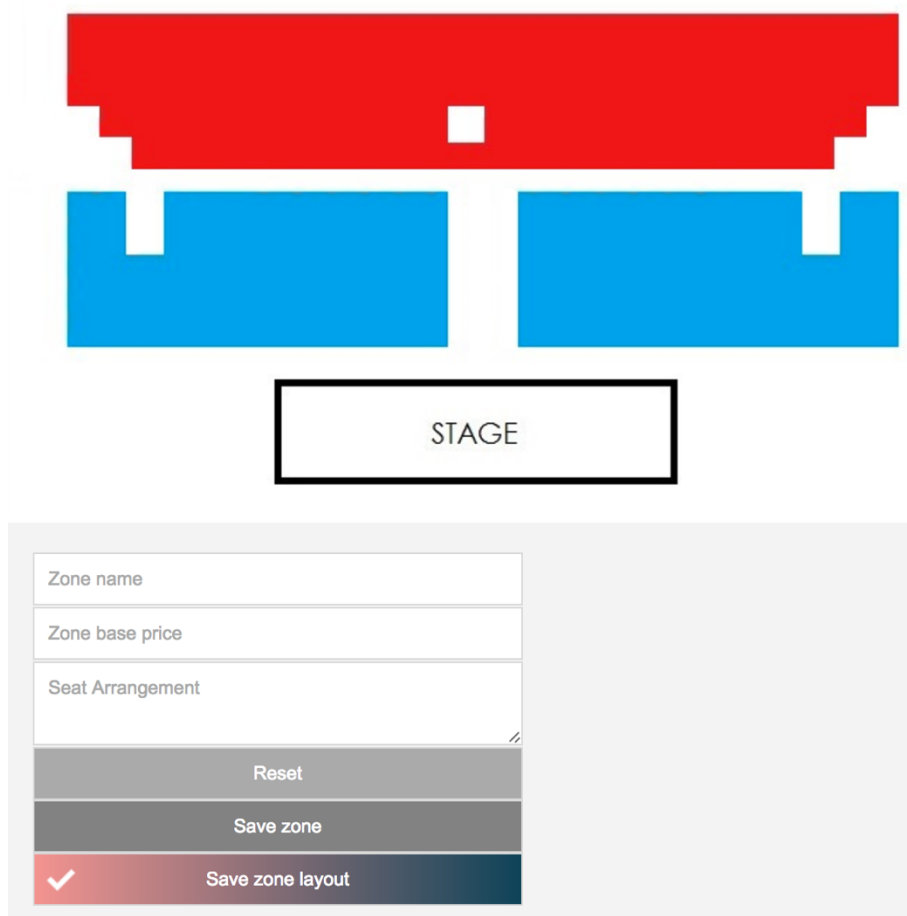
Layout name

Choose file No file chosen

Save layout

Obr. 21 Formulář pro upload podkladu pro layout
Zdroj: autor

Ve druhém kroku se definuje samotné rozložení sedaček, případně rozdělení zón (například pro zvýraznění cenových rozdílů, nebo odlišných částí sálu). Na obrázku (Obr. 22) je možno vidět, jak vypadá druhý krok.



STAGE

Zone name

Zone base price

Seat Arrangement

Reset

Save zone

Save zone layout

Obr. 22 Formulář pro nadefinování sálu
Zdroj: autor

Testovací sál je zde (barevně) rozdělen do dvou zón. *Zone name* by v tomto případě bylo například *červená* nebo *modrá*. *Zone price* je základní cena zóny a *Seat Arrangement* je místo pro vložení definice rozložení sedaček. Jeho struktura je následovná:

A:(B:C,B:C,B:C,...);A:(B:C,B:C,B:C,...);

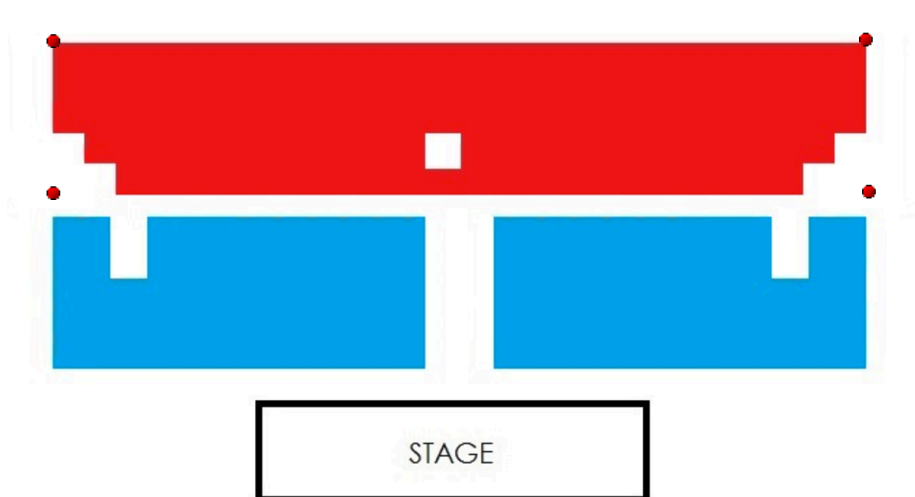
Kde:

- A = název řady
- B = název (číslo) sedadla
- C = cena sedadla vyjádřená procentem ze základní ceny zóny

Tato struktura má přísně stanovená pravidla. Ve struktuře nesmí být zalomený řádek, vše musí být inline. Každá nová řada musí končit středníkem a musí mít stejný počet sedaček. Je pochopitelné, že každý sál nemá stejný počet sedadel v řadě. Chceme-li sedadlo vynechat, stačí na místo znaku *B* vložit jako cenu *0* (nula).

Dalším povinným údajem, který je třeba zadat, je vymezení zón. To se provede kliknutím na vrcholy zón na podkladě layoutu.

Tlačítkem *Save zone* je možné uložit rozpracovanou zónu a přidat další. Pokud jsou již zóny zadány všechny, celý layout se uloží kliknutím na tlačítko *Save zone layout*. Celý vyplněný formulář vypadá následovně.



red
300
<p>A: (1:90,2:90,3:90,4:90,5:90,6:90,7:90,8:90,9:90,9:90,11:90,12:90,13:90,14:90,15:90,16:90,17:90,18:90,19:90, 20:90,21:90,22:90,23:90,24:90,25:90,26:90);B: (1:100,2:100,3:100,4:100,5:100,6:100,7:100,8:100,9:100,10:100,11:100,12:100,13:100,14:100,15:100,16:10 0,17:100,18:100,19:100,20:100,21:100,22:100,23:100,24:100,25:100,26:100);C: (1:100,2:100,3:100,4:100,5:100,6:100,7:100,8:100,9:100,10:100,11:100,12:100,13:100,14:100,15:100,16:10 0,17:100,18:100,19:100,20:100,21:100,22:100,23:100,24:100,25:100,26:100);</p>
Reset
Save zone
✓ Save zone layout

Obr. 23 Vyplněný formulář pro nadeřinování sálu
Zdroj: autor

6.1.5.2 Tvorba šablon vstupenek

Vytvoření šablony vstupenek spočívá pouze ve vytvoření HTML kódu. Vkládá se do Ace editoru, což je speciální komponenta formuláře, která vypadá jako editor kódu.

Ticket template



```
1 <html>
2 <head>
3 <style>
4   body {
5     font-family: "Trebuchet MS", Helvetica, sans-serif;
6     width: 300px;
7   }
8   h1 {
9     font-weight: bold;
10    font-size: 30px;
11    text-transform: uppercase;
12  }
13  h1,h2,h3{
14    padding: 0px;
15    margin: 0px;
```

Obr. 24 Ace editor pro vytvoření šablony vstupenky

Zdroj: autor

V této HTML šabloně je možné využívat zástupné klíče, do kterých se při tisku doplní informace o akci. Píší se vždy do složených závorek jako *{klíč}*.

Allowed variables:

Event's name **{event}**

Event's kind **{kind}**

Event's date **{date}**

Event's time **{time}**

Ticket's price **{price}**

QR code **{qr_code}**

Barcode **{bar_code}**

Unique code **{unique_code}**

City **{address-city}**

Street **{address-street}**

House number **{address-house-number}**

Obr. 25 Seznam Povolných zástupných klíčů

Zdroj: autor

Před tiskem je tedy nutné vytvořit HTML pro tisk. V šabloně se vyhledávají všechny klíče, které jsou v případě nalezení nahrazeny reálnými hodnotami.

```
728 $search = array('{kind}', '{event}', '{date}', '{time}', '{seat}',
729               '{bar_code}', '{qr_code}', '{unique_code}', '{price}', '{address-city}',
730               '{address-street}', '{address-house-number}');
731 foreach($seats as $seat) {
732     //Tickets code
733     $ticket = $this->database->table(self::TICKETS_TABLE_NAME)->where(array(
734         self::TICKETS_EVENTS_ID => $event_id,
735         self::TICKETS_SEATS_ID => $seat['seats_id']
736     ))->select(self::TICKETS_CODE)->fetch();
737
738     $qr_style = "#qrcode".$ticket->tickets_code.
739     "{background: url(\".$qrCode->setText($ticket->tickets_code)->
740     getDataUri().\") no-repeat; width:148px; height:148px; margin:0px auto;}";
741     $h = substr_replace($head, $qr_style, strpos($head, '</style>'), 0);
742     $head = $h;
743
744     $replace = array($event->kinds_name, $event->events_name,
745                   date('M j, Y', strtotime($event->events_date)),
746                   $event->events_time->format('%H:%i'), $seat['seats_name'],
747                   $barcode->code39($ticket->tickets_code),
748                   '<div class="qrcode"><div id="qrcode'.$ticket->tickets_code.'"></div></div>',
749                   $ticket->tickets_code,
750                   $seat['seats_price'], $event->address_city, $event->address_street,
751                   $event->address_house_num);
752
753     array_push($filled, str_replace($search, $replace, $body));
```

Obr. 26 Ukázka kódu pro tisk vstupenek

Zdroj: autor

Soubor: EventManager.php

Pole search drží možné klíče. V cyklu níže se prochází všechna sedadla, pro něž je nutné vytvořit vstupenku. Z databáze, kromě jiného, je potřeba získat kód vstupenky, díky kterému je možné vytvořit QR kód a čárový kód. Oba kódy jsou vytvářeny pomocí knihoven. QR kódu se musí nejdříve definovat CSS styl, následně ho vložit do hlavičky HTML a obalit kód vstupenky do tagu s id qrcode. Oproti tomu čárový kód vyžaduje pouze kód vstupenky bez předchozího stylování.

Pole replace drží všechny hodnoty klíčů, ve stejném pořadí, jak je definováno v poli search. Na řádce 753 probíhá samotná výměna klíčů za hodnoty.

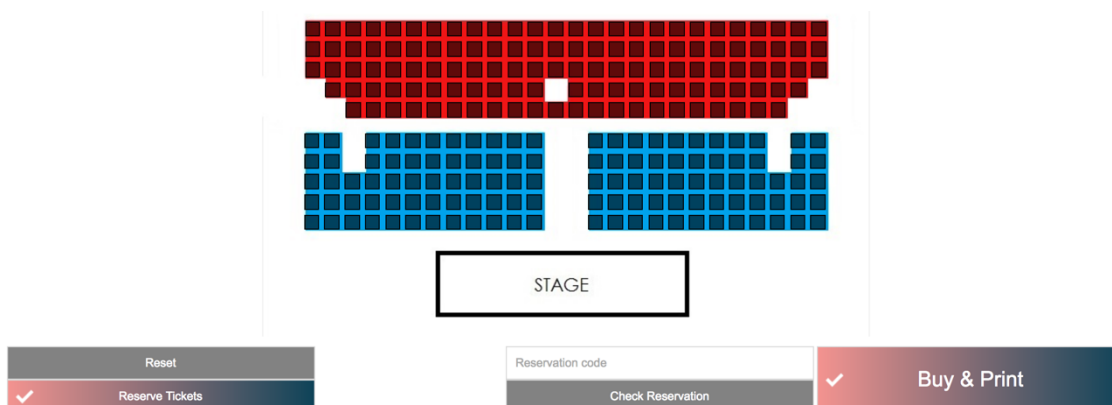
6.1.6 Prodej vstupenek

Po tom, co se vytvořili všechny potřebné layouts a šablony k založení nové akce, je možné prodávat vstupenky. V tabulce akcí k tomu slouží tlačítko s ikonou čárového kódu.



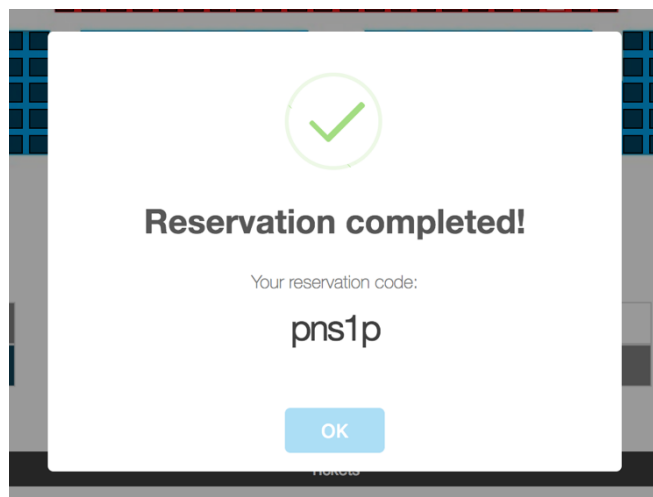
Obr. 27 Tlačítko pro prodej
Zdroj: autor

Po kliknutí můžeme jako první vidět rozložení sedaček v sále, které bylo nadefinováno dříve (Obr. 22), pod ním několik tlačítek a pole pro vložení rezervačního kódu.



Obr. 28 Ukázka formuláře pro prodej vstupenek
Zdroj: autor

Prodej má dvě možnosti – rezervace a přímý prodej. Po vybrání sedaček je možno kliknout na *Buy & Print*, zaplatit a vytisknout vstupenky. Druhý scénář dovoluje vstupenky pouze rezervovat tlačítkem *Reserve Tickets*. Po stisknutí tlačítka vyběhne alert s rezervačním kódem.



Obr. 29 Sweet Alert zobrazující kód rezervace
Zdroj: autor

Rezervační kód je možné vyplnit do formuláře a tlačítkem *Check Reservation* označit všechna sedadla vybrané při rezervaci a poté pokračovat jako při přímém prodeji. Rezervace je platná dvě hodiny, poté se sedadla opět stanou dostupnými.

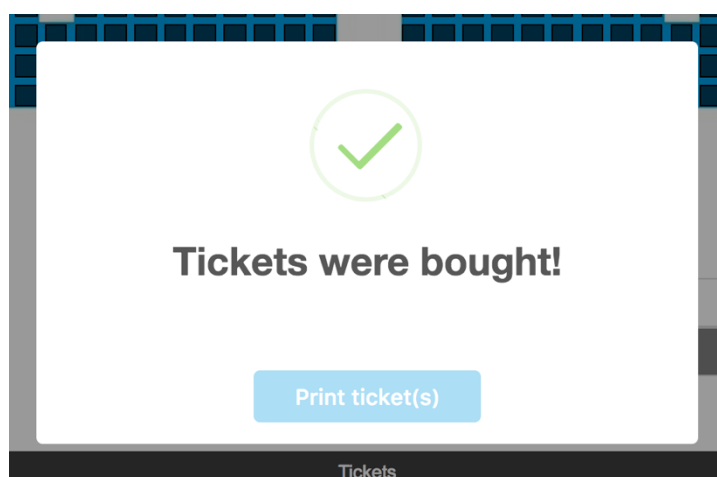
V momentě kdy se vybere alespoň jedno sedadlo, se pod formulářem objeví tabulka, která zobrazuje jméno a cenu sedadla. Při každém dalším vybraném sedadle se tabulka aktualizuje.

Reset	pns1p	Buy & Print
✓ Reserve Tickets	Check Reservation	✓

Tickets	
Name	Price
C22	300
D21	300
E20	300
Total price: 900	

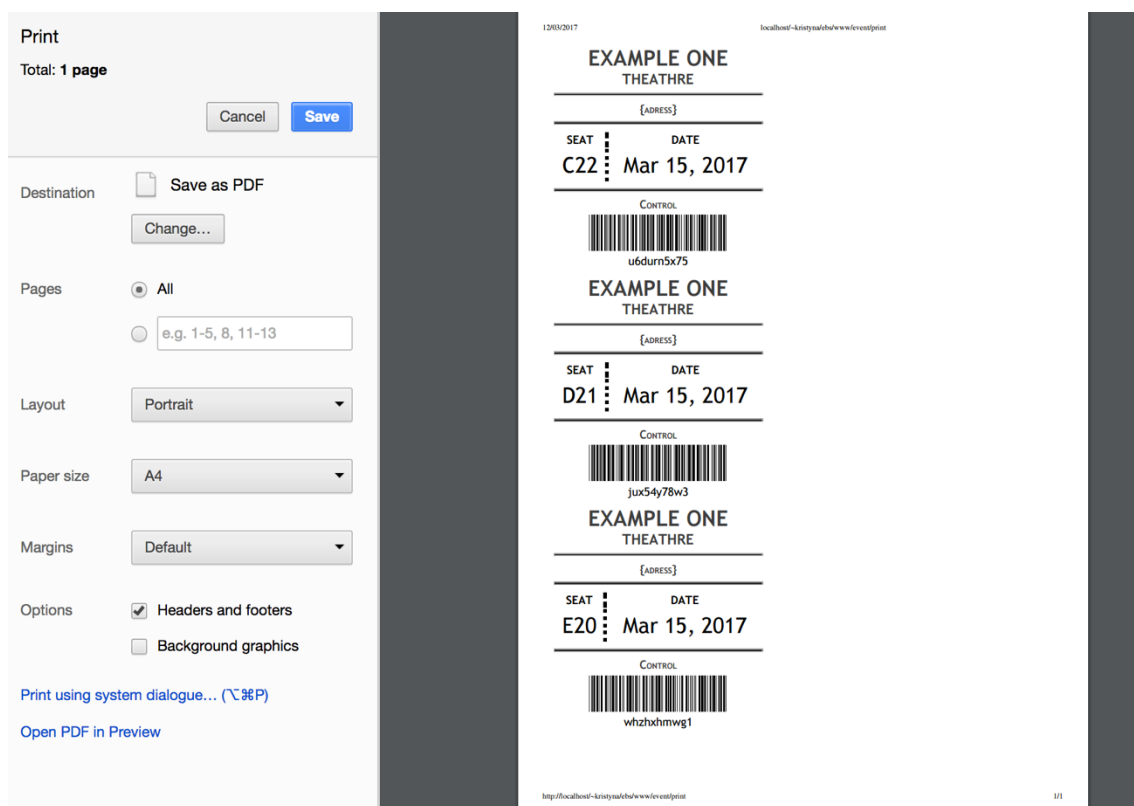
Obr. 30 Tabulka s výčtem označených sedadel
Zdroj: autor

Po zakoupení na obrazovce vyběhne další Sweet Alert s odkazem pro tisk vstupenek.



Obr. 31 Sweet Alert – vstupenky
Zdroj: autor

Stisknutím tlačítka se otevře nová záložka s předpřipravenou volbou pro tisk. S takto vytištěnou vstupenkou končí prodej.



Obr. 32 Ukázka tisku vstupenek
Zdroj: autor

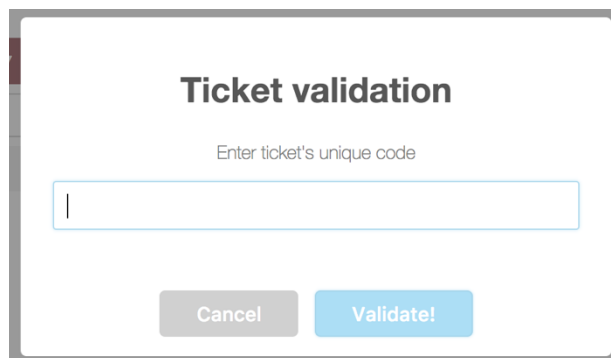
Každý stav sedadla je jinak barevně rozlišen. Na obrázku níže je možno vidět všechny stavy na různých podkladech v tomto pořadí (zleva): neobsazené, aktuálně označené, rezervované, VIP, aktuálně označené VIP a již prodané sedadlo.



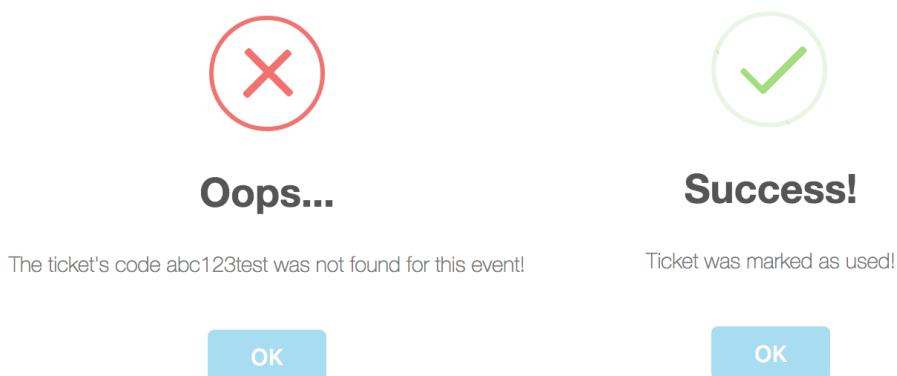
Obr. 33 Stavy sedadel
Zdroj: autor

6.1.7 Kontrola vstupenek

Posledním krokem je kontrola vstupenek při vstupu na akci. Probíhá přes webové rozhraní a rovněž je připravena i Android mobilní aplikace. Validace probíhá jednoduše. Do připraveného pole se vloží desetimístný kód pod čárovým kódem, aplikace ověří, zda kód existuje v databázi a zda již lístek nebyl použit, a následně ho označí jako použitý.



Obr. 34 Validace vstupenky
Zdroj: autor



Obr. 35 Možné výsledky kontroly kódu vstupenky
Zdroj: autor

Vložit desetimístný kód je také možné pomocí čtečky čárových kódů připojenou pomocí USB do počítače.

Reports

Form for report export with options: Last Month, Current Month, Date Range. Fields: From: dd/mm/yyyy, To: dd/mm/yyyy.

Obr. 36 Formulář pro export reportu
Zdroj: autor

6.1.8 Další funkce

Mezi poslední funkce systémů patří například exportování prodaných vstupenek do CSV. Uživatel může zadat počáteční a koncové datum nebo kliknout na některé z předpřipravených možností – minulý nebo tento měsíc.

Níže je zobrazeno, jak může vypadat exportované CSV.

<i>Tickets sales time</i>	<i>Events name</i>	<i>Events date</i>	<i>Events time</i>	<i>Seats name</i>	<i>Tickets code</i>	<i>Tickets price</i>	<i>Users name</i>	<i>Deales name</i>
13.03.17	One	15.03.17	10:20	C22	u6durn5x75	300	admin	Def.
13.03.17	One	15.03.17	10:20	D21	jux54y78w3	300	admin	Def.
13.03.17	Two	16.03.17	17:20	E20	ahzhxhmgw1	200	admin	Def.

Tab. 1 Ukázkový příklad reportu

Nedílnou součástí celé aplikace je rozdělení práv. Počet rolí ani výskyt jednotlivých rolí v systému není omezen. Vytváření rolí vyžaduje pouze název role a dále se pomocí checkboxů vyberou úkony, které bude moct uživatel v dané roli vykonávat.

Role's name:

- New User
- Edit User
- Delete User
- New Dealer
- Edit Dealer
- Delete Dealer
- New Event
- Edit Event
- Delete Event
- New Layout
- Edit Layout
- Delete Layout
- New Template
- Edit Template
- Delete Template
- Sale
- Search
- Roles
- Credit
- Sale VIP
- Reports

Update role

Obr. 37 Formulář pro vytváření rolí
Zdroj: autor

6.2 Zabezpečení

Často používanou metodou pro útok na webovou stránku bývá podstrčení javasriptového, nebo jiného škodlivého kódu, pomocí neošetřených vstupů. Jednou z nich je například Cross-Site Scripting, neboli XSS. Nette nabízí funkci zvanou Context-Aware Escaping, která všechny vstupy ošetřuje automaticky. Není tak třeba kontrolovat URL adresu nebo každý formulářový vstup.

Nette také dbá na správu session, která bývá prostředkem pro získání přístupů do webových aplikací. Útočník tak může bez znalosti hesla nebo uživatelského jména získat přístup kamkoliv. S tímto typem útoku se spojují konkrétní metody, jako například session hijacking, session stealing nebo session fixation. Nette Framework nabízí ochranu pomocí automatické konfigurace PHP. Nutné je však povolit funkci `ini_set()`.

Framework nabízí i funkci formuláře, která dokáže předejít Cross-Site Request Forgery. Tento typ útoku využívá formulář k podstrčení jiné stránky, která vykoná útok na webovou aplikaci, v jež jsme přihlášení. Funkce není vykonávána automaticky a pro její použití je nutné u každého formuláře funkci zapnout jedním příkazem `addProtection()`. (9)

Díky Nette se tudíž při programování není třeba soustředit na ochranu vstupů, url adres, podstrčení škodlivého apod.

7 Shrnutí výsledků

Stěžejním bodem celého systému bylo navrhnout a domluvit se na vhodném postupu vytváření sálů a jeho následné zobrazení. Všechny tyto úkony jsou primárně řešeny za pomoci JavaScriptu. V průběhu implementace se objevovaly nové problémy a s tím i nové nápady na celkové řešení. Datum předání hotového projektu však nebylo přizpůsobeno takovému scénářům, a tak se vždy musela volit nejrychlejší cesta.

Zvolené technologie byly naprosto dostačující a nebylo potřeba dělat žádné ústupky. Práce s Nette Frameworkem byla jednoduchá, dokumentace srozumitelná a díky velké základně českých fanoušků byla většina dotazů na oficiálním fóru zodpovězena rychle.

Díky optimalizovaným databázovým dotazům se povedlo splnit cíl a aplikace je tak připravena i na omezené připojení k internetu. Aplikace rovněž obsahuje správu rolí, vytváření layoutů jak sálů, tak vstupenek. Navigace na webu je přehledná a veškeré funkce jsou dostupné na pár kliknutí. Byly tak splněny veškeré požadavky klienta a stanovené cíle.

Výsledkem je funkční rezervační systém, připravený k ostrému testování.

8 Závěry a doporučení

Cílem práce bylo vytvořit webovou aplikaci na zakázku. V zadání práce je vytvořit uživatelsky přívětivý rezervační systém, s možností vytváření vlastních sálů a designu vstupenek. Aplikace primárně slouží pro centra, která zaštiťují kulturní akce. Neslouží jako portál volně dostupný veřejnosti.

Každý uživatel spadá pod jednoho dealera, jehož účelem je spravovat finance. Z kreditu, který dealer vlastní je odečtena celková částka z každého prodeje, který uživatel učiní. Z čehož vyplývá, že pokud dealer nemá dostatek kreditu, není možné provádět prodej. Tento systém zabraňuje případnému vymáhání peněz z prodeje.

Z pohledu autorky je v sekci vytváření layoutů sálů velký prostor pro zlepšení UX⁴. Návrh layoutu rovněž není přizpůsoben pro nepravidelné rozložení sedaček. Momentální řešení nepočítá s možností jiného uspořádání, než jsou řady se sedačkami přesně za sebou. Tyto postřehy by mohly být zahrnuty v případném updatu aplikace.

V závěru se aplikace podrobila testování, díky kterému se odladily některé chyby. Dalším krokem bude test na reálné akci.

⁴ User Experience

9 Seznam použité literatury

1. **Kumar, Harish.** What are the Pros and Cons of Laravel? *Quora*. [Online] 5. únor 2016. [Citace: 27. září 2016.] www.quora.com/What-are-the-Pros-and-Cons-of-Laravel.
2. **Sturgeon, Phil.** 5 Things CodeIgniter Cannot Do (without a rewrite). *Phil Sturgeon*. [Online] 5. prosinec 2012. [Citace: 27. září 2016.] <https://philsturgeon.uk/codeigniter/2012/12/05/5-things-codeigniter-cannot-do-without-a-rewrite/>.
3. **Weier O'Phinney, Matthew.** Zend Framework 3 Released! *Zend Framework*. [Online] 28. červen 2016. [Citace: 27. září 2016.] <https://framework.zend.com/blog/2016-06-28-zend-framework-3.html>.
4. **Nette Foundation.** Nette Framework. *Nette*. [Online] [Citace: 27. září 2016.] <https://nette.org/cs/>.
5. **Skvorc, Bruno.** The Most Popular Framework of 2015. *Site Point*. [Online] 30. březen 2015. [Citace: 27. září 2016.] www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results.
6. **Nette Foundation.** Adresářová struktura. *Nette*. [Online] [Citace: 12. 12 2016.] <https://doc.nette.org/cs/0.9/quickstart/adresarova-struktura>.
7. **Bootstrap core team.** The world's most popular mobile-first and responsive front-end framework. *Bootstrap*. [Online] [Citace: 12. 12 2016.] <http://getbootstrap.com/>.
8. —. Components. *Bootstrap*. [Online] [Citace: 12. 12 2016.] <http://getbootstrap.com/components/#glyphicons>.
9. **Nette Foundation.** Zabezpečení před zranitelnostmi. *Nette Framework*. [Online] [Citace: 5. březen 2017.] <https://doc.nette.org/cs/2.4/vulnerability-protection>.
10. **Borek, Bernard.** Prezentační vzory z rodiny MVC. *Zdroják*. [Online] březen 2009. [Citace: 27. září 2016.] www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc.
11. **tým, jQuery.** jQuery. *jQuery*. [Online] [Citace: 27. září 2016.] www.jquery.com.

12. **Borek, Bernard.** Alternativy k MVC a závěrečné poznámky. *Zdroják*. [Online] 15. květen 2009. [Citace: 27. září 2016.] <https://www.zdrojak.cz/clanky/alternativy-k-mvc-a-zaverecne-poznamky/> .
13. **Nette Foundation.** Začínáme. *Nette*. [Online] 2016. [Citace: 1. březen 2017.] <https://doc.nette.org/cs/2.4/quickstart/getting-started>.
14. —. Zabezpečení před zranitelnostmi. *Nette*. [Online] 2016. [Citace: 1. březen 2017.] <https://doc.nette.org/cs/2.4/vulnerability-protection> .
15. **Vrána, Jakub.** Adminer. *Adminer*. [Online] [Citace: 1. březen 2017.] <https://www.adminer.org/cs/> .

10 Přílohy

1. CD se zdrojovými kódy

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2016/2017

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Rehovičová Kristýna	Dvouletky 1210/56, Ostrava - Hrabůvka	I14135

TÉMA ČESKY:

Webová aplikace využívající principy MVP

TÉMA ANGLICKY:

Web application based on MVP architecture

VEDOUcí PRÁCE:

doc. Ing. Filip Malý, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl práce:

Cílem je vytvořit stabilní rezervační systém dle potřeb vybraného zákazníka. Systém by měl obsáhnout prodej a rezervaci lístků, definici sálů, tvorbu designu vstupenky, dále poskytovat správu přístupových práv (implementovat mechanismy autentizace, autorizace, logování) a splňovat požadavky tvorby moderních webových aplikací.

Osnova

1. Úvod
2. Cíl práce
3. Úvod do problematiky
4. Návrhové vzory
5. Výběr technologie
 - 5.1. PHP Framework
 - 5.2. Skriptovací jazyk
6. Analýza a návrh
7. Závěr

SEZNAM DOPORUČENÉ LITERATURY:

Google Scholar

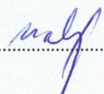
Podpis studenta:



Datum:

30.9.2016

Podpis vedoucího práce:



Datum:

30.9.2016