

Mendelova univerzita v Brně  
Provozně ekonomická fakulta

---

# **Webová aplikace pro správu sportovního oddílu**

**Bakalářská práce**

Vedoucí práce:  
Ing. Pavel Turčíněk, Ph.D.

Tereza Plíšková

Brno, 2017



Na tomto místě bych ráda poděkovala vedoucímu práce Ing. Pavlu Turčínkovi, Ph.D. především za jeho cenné rady a čas, který mi věnoval.

### **Čestné prohlášení**

Prohlašuji, že jsem tuto práci: **Webová aplikace pro správu sportovního oddílu**

vypracovala samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědoma, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

v Brně dne 15. května 2017

.....

**Abstract**

Plíšková, Web application for sport clubs administration. Brno, 2017

The bachelor thesis deals with the design and subsequent implementation and deployment of web applications for the management of the volleyball club, that makes internal management process easier. In the design phase is create Usecase diagram for a purpose unification and visualization requirements and on the basis of these requirements is built ERD diagram, which is used as a support for a real database. The bachelor thesis also deals with web design of web presentation. For this purpose are made wireframes for desktop version and also for a mobil version. The chapter Results summarizes the implementation process and testing of the application.

**Key words**

Zend Framework, SQL, PHP, web application, volleyball

**Abstrakt**

Plíšková, Webová aplikace pro správu sportovního oddílu. Brno, 2017

Bakalářská práce se zabývá návrhem a následnou implementací a nasazením webové aplikace pro správu volejbalového oddílu, která usnadňuje vnitřní procesy správy oddílu. Ve fázi návrhu aplikace je vytvořen Usecase diagram za účelem sjednocení a vizualizace požadavků oddílu a na základě těchto požadavků je sestaven ERD diagram sloužící jako podklad pro reálnou databázi. Práce se zabývá také návrhem webového designu webové prezentace, kdy jsou sestaveny drátové modely jak pro desktopovou tak mobilní verzi aplikace. V kapitole výsledky je shrnut postup implementace a následného testování aplikace.

**Klíčová slova**

Zend Framework, SQL, PHP, webová aplikace, volejbal

## Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
1.1	Charakteristika fungování volejbalového oddílu DDM Hippo Kuřim . . . . .	8
<b>2</b>	<b>Cíl práce</b>	<b>9</b>
<b>3</b>	<b>Přehled literatury a pramenů</b>	<b>10</b>
3.1	Webová aplikace . . . . .	10
3.2	MVC . . . . .	10
	Model . . . . .	11
	View . . . . .	11
	Controller . . . . .	11
3.3	ORM . . . . .	12
3.4	Stávající řešení řízení chodu sportovních oddílů . . . . .	12
	Sklub . . . . .	12
	Kao . . . . .	13
	Teamstuff . . . . .	13
	Sportvia.eu . . . . .	13
<b>4</b>	<b>Materiál a metody</b>	<b>14</b>
4.1	Funkční požadavky na aplikaci . . . . .	14
	Funkce pro vedení . . . . .	15
	Funkce pro členy . . . . .	15
	Funkce společné pro vedení i pro členy . . . . .	15
	Funkce pro veřejnost . . . . .	15
4.2	Nefunkční požadavky na aplikaci . . . . .	15
4.3	Role uživatelů . . . . .	16
4.4	Použité technologie a nástroje . . . . .	16
	Zend framework . . . . .	16
	MVC . . . . .	17
	ORM . . . . .	17
	Přístup a práce s databází . . . . .	18
	ACL a AccessFiltering . . . . .	19
	Mapy API . . . . .	19
	Google Charts . . . . .	19
	Visual Paradigm . . . . .	19
	Xampp . . . . .	19
<b>5</b>	<b>Výsledky</b>	<b>20</b>
5.1	Návrh aplikace . . . . .	20
	UseCase diagram . . . . .	20
	Entitně relační diagram . . . . .	22
5.2	Web design . . . . .	24

---

Předpokládání návštěvníci . . . . .	24
Organizační schéma . . . . .	24
Navigace . . . . .	27
Prostorové rozložení informací . . . . .	27
Barevné řešení . . . . .	32
Barevné schéma . . . . .	32
5.3 Použité principy Zend Frameworku . . . . .	33
Instalace . . . . .	33
Továrny . . . . .	33
Formuláře . . . . .	34
Přihlašování uživatelů . . . . .	37
5.4 Implementace . . . . .	37
Modul User . . . . .	37
Modul Application . . . . .	39
5.5 Testování aplikace . . . . .	41
Automatické testy pomocí phpUnit . . . . .	41
Testování uživateli . . . . .	42
Kufrový test navigace . . . . .	43
5.6 Implementace aplikace na webový server . . . . .	46
<b>6 Diskuze</b>	<b>47</b>
6.1 Zhodnocení výsledků, návrh možných zlepšení . . . . .	47
6.2 Srovnání výsledku s obdobnými řešeními . . . . .	47
<b>7 Závěr</b>	<b>49</b>
<b>8 Reference</b>	<b>50</b>
<b>9 Seznam použitých zkratek</b>	<b>52</b>
<b>10 Obsah příloženého CD</b>	<b>53</b>

# 1 Úvod

V dnešní době internetu a chytrých telefonů se čím dál tím víc rozšiřují webové aplikace, především kvůli jejich snadné dostupnosti z jakéhokoliv přístroje, stačí mít pouze přístup na internet. Každá organizace potřebuje uchovávat či získávat určitá data a díky webovým aplikacím je možné oslovovat a komunikovat s více uživateli oproti použití desktopové aplikace či pouze statických html stránek. Webové aplikace jsou oblíbené především pro jednoduchost správy a údržby aplikace, kdy oproti desktopovým aplikacím není třeba aktualizovat každou pracovní stanici zvlášť, či mít více variant aplikace dle operačního systému. Nevýhodou je samozřejmě to, že webové aplikace jsou závislé na internetovém připojení a tedy na poskytovateli toho připojení, kdy v případě výpadku nebude služba dostupná.

Aplikace bude defaultně vytvořena pro volejbalový oddíl DDM Hippo Kuřim. Momentálně má oddíl pouze statickou webovou prezentaci, která vzhledově příliš neodpovídá webovým standardům a ani barevné provedení není vhodně zvolené vzhledem k tomu, že oddíl pracuje s dětmi. Za tímto účelem byl navrhnout nový moderní design webové prezentace.

## 1.1 Charakteristika fungování volejbalového oddílu DDM Hippo Kuřim

Jak vyplývá již z názvu, volejbalový oddíl je veden pod záštitou Domova dětí a mládeže v Kuřimi a je tedy určen dětem do 18 let. Celý oddíl čítající asi kolem 50 členů je rozdělen na pět týmů, kdy jeden hráč může hrát za více týmů ačkoli primárně má přidělen jeden pro něho hlavní tým. Každý tým má tréninky dvakrát týdně. Účast na tréninku je evidována a z výsledků následně vyplývají odměny v podobě uspořádání společné akce pro členy oddílu, kteří mají více jak 70% účast na trénincích za danou sezónu. Účast na zápase je momentálně domlouvána ústně na trénincích a následně trenér ručně zadává na web informace o zápase, jako i jména zúčastněných a jejich komentáře k zápasu. Na stejný způsob funguje i domlouvání na společné akce a jejich následná prezentace. Zápasy ve volejbale jsou evidovány po jednotlivých kolech, kdy vítězem kola je ten, kdo vyhraje tři zápasy, které se hrají do 25 bodů s minimálním rozdílem dvou bodů, v případě stavu 2:2 na sety se hraje poslední zápas pouze do 15 bodů stále s minimálním rozdílem 2 bodů. Jako v každém zájmovém oddílu je i zde možnost získání sponzorů, avšak jelikož oddíl v současné době vedou tři trenéři, kteří jsou sami vytíženi prací, studiem či rodinou a nemají tak potřebný čas na shánění sponzorů, finance tekoucí do oddílu sestávají pouze z příspěvků od Domu dětí a mládeže.



## 2 Cíl práce

Cílem je vytvořit webovou aplikaci pro podporu organizace chodu volejbalového oddílu. Tato webová aplikace s sebou přinese nový moderní design webové prezentace oddílu a umožní zjednodušit procesy týkající se běžného chodu oddílu, tedy například správa financí a docházky, přihlašování na akce a zápasy, komunikace a výměna názorů mezi členy oddílu. Webová aplikace však musí i srozumitelně a přehledně prezentovat informace vnějšímu okolí, jedná se tedy především o aktuální informace týkající se dění v oddílu, ale i informace o vedení či o samotných hráčích a konaných akcích a zápasech. Podstatné jsou také ohlasy těchto návštěvníků, z tohoto důvodu bude vytvořena i chatovací místnost pro případné dotazy či připomínky. Součástí cíle je i vytvořit takovou webovou prezentaci, která by přilákala sponzory a oddíl se tak díky finanční podpoře mohl více vyvíjet a stát se tak atraktivnějším pro nové členy a to například pomocí zakoupení potřebného sportovního náčiní.

## 3 Přehled literatury a pramenů

### 3.1 Webová aplikace

Webová aplikace je aplikace umístěna na webovém serveru, ke které uživatelé přistupují přes internet či intranet (Wikipedia, [2016]). Webové aplikace umístěné v síti internet bývají nejčastěji využívány ke komerčním účelům a to například pro marketing či ke komunikaci s okolím organizace provozující danou aplikaci. Obsah webových stránek se zobrazuje dynamicky, kdy oproti statickým stránkám, které se uživateli zobrazí tak, jak jsou uloženy, je obsah dynamických stránek generován aplikací dle zadaných skutečností jako je například čas, přihlášený uživatel, nebo zadané parametry. Interaktivity s uživatelem je na stránkách docíleno především pomocí formulářů.

Při návrhu webové aplikace, která bude umístěna do sítě internet, je nezbytné počítat s velkou škálou uživatelů, kteří budou k aplikaci přistupovat přes různá zařízení a z různých míst. Prvky v aplikaci by tudíž měly být přizpůsobeny různým velikostem koncových zařízení.

Vývoj webové aplikace se dá přirovnat k vývoji informačního systému, který byl již dříve popisován mnoha autory, jako příklad může být uveden Vodopádový model W. W. Royce, Spirálový model B. Boehma, či Iterativní vývoj reprezentovaný například RAD přístupem, který navrhl J. Martin (Šilhavý et al., 2013, s. 5-8). V knize Vybrané aspekty návrhu webových informačních systémů uvádí R. Šilhavý, že v případě webových prostředí je potřeba tyto přístupy trochu pozměnit a rozdělují fáze vývoje do pěti částí a to:

1. koncepce a strategie,
2. struktura a design,
3. kódování a vývoj,
4. testování a spuštění,
5. optimalizace a údržba.

Tento vývoj tak spojuje základní kostru vodopádového modelu s následným evolučním vývojem. (Šilhavý et al., 2013, s. 9-12)

### 3.2 MVC

MVC (Model-View-Controller) je softwarový návrhový vzor, v PHP využívaný se zaměřením na programovací paradigma objektově orientované programování (Pitt, 2012, s. 1). Při použití tohoto přístupu je docíleno především oddělení business logiky, uživatelského rozhraní a řídicí logiky.

### Model

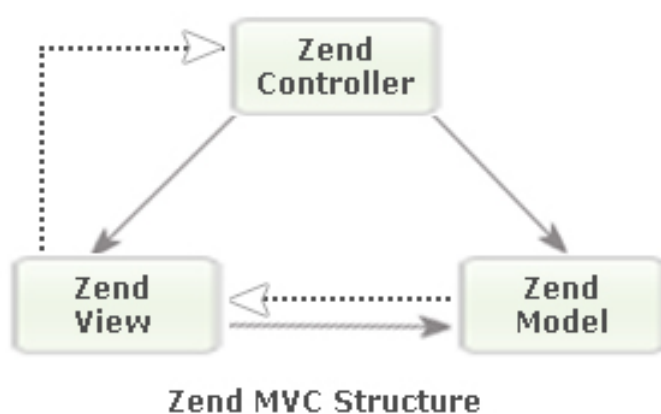
Modely jsou objekty, ve kterých jsou uchovávána data a business logika celé aplikace. Tato vrstva tedy zprostředkovává přístup k datům a manipuluje s nimi.

### View

Pohledy jsou komponenty, které uživatelům prezentují data uložená v modelech.

### Controller

Controllery jsou komponenty, které spojují pohledy a modely dohromady a zajišťují tedy předávání dat z modelu do pohledu a naopak. Controller je první s kým uživatel komunikuje a zpracovává tedy požadavky od uživatele. Příkladem může být zadání url adresy, kdy router podle zadané adresy zavolá daný Controller, který následně volá příslušný model pro práci s daty a nakonec vygeneruje příslušný pohled. (Čápka, 2013)



Obrázek 1: MVC struktura, zdroj: (Variya Soft, ©2009 - 2017)

### 3.3 ORM

ORM (Object Relational Mapping) je metoda mapování relační databáze na objekty (F.Tröster, ©2010). Pomocí této metody je nad tabulkou vytvořen objekt a díky tomu usnadňuje provádění operací jako je čtení, úprava a mazání dat z databáze bez nutnosti použití SQL dotazů. Implementaci ORM usnadňují volně dostupné frameworky jako je například Doctrine pro PHP, který pro psaní dotazů využívá DQL (Doctrine Query Language).

### 3.4 Stávající řešení řízení chodu sportovních oddílů

Existuje více možností správy vnitřního řízení chodu sportovních oddílů. Ve větších organizacích je tento problém řešen v podobě zakoupení licence informačního systému a následné vlastní upravení. Mezi nejznámější česká řešení patří například stránky sklub.cz a kao.cz. Častým řešením u velkých klubů bývá nechat si navrhnout aplikaci přímo na míru. To však neplatí u menších organizací či oddílů, ty většinou nemají finanční prostředky pro zakoupení takové aplikace. Zde přicházejí v úvahu open source systémy, které však ve většině případů nenabízí vše, co by daný oddíl potřeboval a nebo naopak obsahuje funkce, které vůbec nevyužije a jsou pro něj matoucí. Nevýhodou těchto systémů je, že jsou často dostupné pouze v anglickém jazyce. Z tohoto důvodu přichází na řadu možnost vytvořit si web pomocí CMS systémů. Nabídka těchto systémů je poměrně široká, jako příklad může být uveden sweb (Sweb, [2016]). Toto řešení však neřeší potřebu vnitřní správy, ale jen prezentace oddílu veřejnosti. Je zde možnost zkombinovat předešlé dva přístupy a webovou prezentaci široké veřejnosti vytvořit statický web například s pomocí CMS systému a následně vnitřní organizaci řešit pomocí zakoupení licence na systém poskytující tyto funkcionality, příkladem může být aplikace Sportvia.eu, avšak i zde oddíl musí počítat se zpoplatněním těchto systémů. Většinou se však jedná o systémy, které jsou navrženy pro velkou škálu sportů a některé funkcionality mohou být pro daný oddíl nepotřebné, například právě kvůli dětským členům, pro které by mohl být systém příliš složitý.

Dalším důležitým kritériem při výběru jsou i jazykové možnosti daného systému. V této práci bude řešen problém, kdy oddíl pracuje s dětmi, tudíž nemůže být použita aplikace dostupná pouze v anglickém jazyce.

#### Sklub

Stránky vytvořené pomocí redakčního systému nabízeného na stránkách společnosti SKLUB.cz s.r.o slouží především k prezentaci oddílu veřejnosti. Na stránkách návštěvník webu najde vše, co o týmu potřebuje vědět včetně aktuálního dění a výsledků zápasů (Sklub, [2017]). Tato aplikace se však zaměřuje pouze na webovou prezentaci nikoli na komunikaci mezi členy oddílu či na vnitřní fungování jako je například správa financí nebo přihlašování na akce. Poplatky za vedení stránek sice

nejsou tak vysoké a momentálně se pohybují kolem 290 Kč za měsíc, ale i taková částka může být pro oddíl bez sponzorů problém.

### **Kao**

Obdobného výsledku jako u stránek vytvořených pomocí systému Klub dosáhneme i za použití systému od Kao. Správa stránek probíhá také pomocí redakčního systému (Kao, ©2007 - 2017). Nevýhodou je, že většina stránek vytvořených pomocí tohoto systému zatím neposkytuje responzivní zobrazení a v případě mobilních zařízení se stránky stávají nepřehledné. Pro tento web není možné uvést cenové rozpětí, jelikož společnost provádí časovou a finanční analýzu až po obdržení požadavků.

### **Teamstuff**

Velmi dobře propracovaná aplikace je Teamstuff. Jedná se o poměrně mladou aplikaci, avšak momentálně nemá aplikace plnou podporu pro český jazyk a tak se stává, že polovina stránky je česky a druhá anglicky. Aplikace umožňuje komunikaci mezi členy, plánování akcí, přehled členů týmu, správu financí, přidávání fotografií do fotogalerie a obsahuje také spoustu dalších doplňkových funkcí sloužících pro komunikaci a řízení oddílu (Teamstuff, ©2017). Nevýhodou je ale nejen neúplnost překladu, ale i to, že aplikace neřeší prezentaci oddílu široké veřejnosti. Výhodou však je především to, že aplikace je poskytována zcela zdarma.

### **Sportvia.eu**

Řešení od firmy Sportvia.eu řeší vnitřní správu chodu oddílu a nezabývá se webovou prezentací. Aplikace se stará především správu členů, financí a dokumentů (Sportvia.eu, ©2008). Toto řešení navíc není příliš vhodné pro sportovní kluby pracující s dětmi, ale mohlo by být vhodné především pro sportovní kluby pracující s dospělými osobami, nabízející navíc například doplňky stravy, kdy sportvia.eu nabízí také skladovou evidenci zásob a pro sledování sportovních výsledků cvičební a výkonnostní informace. Základní cena pro pořízení aplikace je 450 Kč za měsíc.

Z výše uvedených důvodů je tedy pro malé oddíly, které žijí jen z členských příspěvků, velmi těžké vybrat optimální řešení, které by spojovalo jak správu oddílu tak i webovou prezentaci, bez nutnosti ukládání redundantních dat a mít tak vše na jednom místě a přitom neplatit velké měsíční poplatky.

Tato práce bude zaměřena na tvorbu aplikace pro specifické sportovní odvětví a to pro volejbal. Každý sport má jinak řešený systém turnajů a zápasů, proto byla vybrána jedna konkrétní oblast pro kterou bude aplikace vytvořena a pro vizualizaci a návrh webové prezentace budou použita data a obrázky konkrétního oddílu.

## 4 Materiál a metody

Jak již bylo nastíněno výše, vývoj webové aplikace probíhá v několika fázích. V průběhu vývoje je třeba průběžná konzultace se zadavatelem kvůli předejití následných nesrovnalostí mezi představami a konečným výsledkem. V první fázi koncepce a strategie je třeba sjednotit představy zadavatele a vývojáře. Po komunikaci jak s vedením oddílu tak i s jeho členy budou stanoveny funkce, které by měla aplikace poskytovat. Tyto požadavky budou dále popsány v podkapitole Požadavky na aplikaci.

Dále je třeba také zjistit, zda zadavatel již má konkrétní představu o designu webu, nebo zda bude navrženo více barevných návrhů, ze kterých bude následně zadavatel vybírat.

V druhé fázi struktura a design bude na základě požadavků vytvořen návrh aplikace. Tento návrh bude realizován pomocí UML diagramů. Z návrhu pak vyplynou potřeby pro ukládání dat. Na základě těchto potřeb bude vytvořen logický návrh datového modelu reprezentovaný entitně relačním diagramem. Z vytvořených diagramů bude následně odvozena struktura webu. Pro návrh designu stránek bude použito drátových modelů. Následně bude navrženo barevné schéma.

Ve fázi kódování a vývoj bude za použití předpřipravených modelů a diagramů vyvíjena webová aplikace a to pomocí programovacího jazyka PHP 7 a Zend frameworku.

Z velké škály dostupných PHP frameworků byl vybrán právě Zend Framework, ačkoli v České republice je tendence se více přiklánět k Nette, jelikož se jedná o český framework a velká část dostupných materiálů je tedy psána v češtině.

Pro práci s daty bude využívána databáze MySQL s vývojovým nástrojem PhpMyAdmin. Zend Framework nabízí spoustu komponent k odinstalaci avšak nenachází se zde komponenty pro práci s mapami a s grafy. Pro práci s mapami bude tedy použita externí knihovna od mapy.cz a vykreslení grafů bude zajišťovat nástroj od Googlu. Další použité technologie budou HTML, CSS a JavaScript.

Fáze testování a spuštění bude obsahovat především automatické testy pomocí PHPUnit frameworku a po nasazení aplikace na server bude provedeno testování samotnými uživateli. Pro otestování přístupnosti a pochopitelnosti webu bude použit kufrový test navigace pro vybranou podstránku. Na základě výsledků jednotlivých testů budou opraveny případné chyby a následně bude aplikace uvedena do ostrého provozu.

Fáze optimalizace a údržba pak probíhá po celou dobu existence webové aplikace. Jsou v ní především zpracovávány připomínky a požadavky uživatelů.

### 4.1 Funkční požadavky na aplikaci

Je potřeba rozlišit funkce navržené pro vedení oddílu, pro jeho členy a pro širokou veřejnost.

### **Funkce pro vedení**

- Zadávání docházky na tréninky
- Přidávání a editace zápasů a akcí, zahrnující nominaci členů
- Finanční deník - přehled pohybu svěřených financí
- Přidávání nových hráčů

### **Funkce pro členy**

- Přihlašování na akce a zápasy
- Přidávání komentářů k navštíveným zápasům a akcím

### **Funkce společné pro vedení i pro členy**

- Nahrávání souborů do fotogalerie
- Správa údajů o vlastní osobě (změna přihlašovacích údajů, změna fotografie, změna doplňujících informací)
- Chatovací místnost, sloužící především ke komunikaci o tématech týkajících se organizace a chodu oddílu či plánovaných událostech

### **Funkce pro veřejnost**

- Zobrazení základních údajů v podobě webové prezentace
- Přehledné zobrazení docházky do grafů s možností filtrace po jednotlivých měsících a týmech
- Chatovací místnost určená pro dotazy veřejnosti

## **4.2 Nefunkční požadavky na aplikaci**

Webová aplikace musí být umístěna na veřejný hostitelský server. Existuje velká škála takových serverů, kdy některé však neposkytují možnost práce s databází. Možnost práce s databází je tedy prvním kritériem při výběru serveru, kdy už není příliš potřeba dívat se na velikost prostoru pro uložení databáze, jelikož databáze není příliš rozsáhlá. Dále je také potřeba dívat se na verzi PHP. Pro aplikaci je třeba minimálně PHP 5.6 a vyšší. Jedním z funkčních požadavků byla možnost přidávat fotografie do fotogalerie, z tohoto požadavku vyplývá skutečnost, že hostitelský server musí poskytovat dostatečně velký prostor pro tyto soubory. Samozřejmostí je, že web musí být napsán podle pravidel přístupnosti a pro uživatele snadno ovladatelný, pochopitelný a bezpečný. Posledním nefunkčním požadavkem je požadavek konzistentního vzhledu. Vzhled vychází z loga oddílu viz. kapitola Analýza a návrh podkapitola Web design.

## 4.3 Role uživatelů

Jak již vyplývá z funkčních požadavků popsaných výše je potřeba implementovat dva typy rolí. Můžeme je popsat jako trenér a hráč. Trenér zde plní funkci administrátora a vytváří především zápasy a akce, na které se hráči přihlašují a dále spravuje informace o týmech a vytváří nové členy. Hráč může psát omluvenky k tréninkům a také se přihlašovat na zápasy a akce a následně k nim přidávat komentáře.

## 4.4 Použité technologie a nástroje

Na začátku vývoje každé webové aplikace stojí rozhodnutí jaké nástroje a technologie použít. V této fázi už víme, co by aplikace měla umět, tím pádem můžeme odhadnout složitost a vybrat programovací jazyk. V případě PHP nemá cenu snažit se psát aplikace pouze na čistém PHP. Existuje nesčetné množství frameworků, které usnadňují vývoj tím, že se například starají o navigaci, definují návrhové vzory a v případě MVC frameworků zajišťují oddělené logiky od prezentace výsledků.

V současné době se pro menší projekty rozmáhá trend vyvíjení middleware API. Jejich velkou výhodou je především jejich jednoduchost. Tento přístup se běžně používá i v jazycích jako je například Python nebo Node. Vybraný Zend Framework patří k jednomu z větších frameworků s velkou internetovou podporou (Zend, ©2006-2017-1).

### Zend framework

Zend Framework je PHP open source framework. Pro platformu Windows je k instalaci a práci s frameworkem používán composer, kdy po jeho stažení nainstalujeme framework. Můžeme nainstalovat pouze základní Zend\_MVC balík a nebo všech 61 balíků. To však ještě neznamená, že všechny nainstalované balíky musíme používat. Potřebné balíky, se kterými chceme pracovat následně uvedeme v projektu. Velkou výhodou Zend Frameworku je právě jeho flexibilita v užívání balíků. Jak je vidět už z postupu instalace, vybereme si pouze balíky, které chceme používat, případně si vytvoříme vlastní třídy (Zend, ©2006-2017-2). Jako každý framework má své výhody i nevýhody. Tato práce je psána v nejnovější verzi frameworku, tedy ve verzi 3. Oproti předešlým verzím se trochu zkomplikovala práce se Zend\_Form, který už neumožňuje práci s dekorátory a tak se ve view vyvolávají postupně všechny komponenty formuláře, se kterými je možné dál pracovat. Naproti tomu odpadla notace `array()`, díky nástupu php 7, která byla nahrazena pouze hranatými závorkami a jelikož v Zend Frameworku bylo `array()` velmi často využíváno, stával se kód často nepřehledným. Výhoda toho je vidět například při definování routovacích cest. Další výhodou jsou i vznik nového `FactoryInterface`, která zahrnuje metodu `__invoke`, které je předán jako argument `ContainerInterface`, který implementuje `ServiceManager` (Lonský, 2016).

Další výhodou Zend Frameworku je i rozsáhlá síť přispěvatelů a přehledná dokumentace, kdy ke každé komponentě existuje popis s možnostmi jak ji využít.



## MVC

**Model** — Jak již bylo nastíněno v Přehledu literatury, modely jsou třídy starající se o přístup k datům z databáze. Obsahují tedy funkce, které něco z databáze získávají a následně vrací získanou hodnotu, nebo do databáze vkládají či upravují data. Modely jsou umístěné ve složce ../Model a jsou vytvářeny pomocí továrny, které je prostřednictvím ServiceMeneger předán jako argumenty \$requestedName, což je název vytvářeného modelu a ContainerInterface, ze kterého si továrna vytáhne potřebné objekty jako ORM objekt User, třída pro práci s adapterem a třídu pro autentizaci a autorizaci. Pomocí této továrny jsou pak vytvářeny modely pro jednotlivé kontrolery. Názvy modelu a továrny, ze které jsou vytvářeny jsou uvedeny v module.config.phtml.

**View** — Views neboli šablony vykreslují podobu HTML stránky, v Zend frameworku se používají šablony s koncovkou .phtml, což umožňuje a usnadňuje případnou další práci s daty v pohledech. Šablony jsou uloženy ve složce dle názvu controlleru a pojmenovány dle dané akce ve formátu name-of-action.phtml. V případě, že v jedné šabloně potřebujeme zobrazit více akcí použijeme tzv. vhníždění akcí (dá se nahradit použitím, to však zobrazí akci jen na předem definovaném prostoru). Vytvoříme si proměnnou, do které nastavíme požadovanou šablonu.

```
$newView = new ViewModel();  
$newView->setTemplate('controllerName/name-of-' . 'action');  
$view->addChild($newView, 'new_view');  
return $view;
```

Následně v dané šabloně vhnížděnou šablonu vyvoláme. Lze také změnit layout, který je primárně definovaný v module.config.php a to příkazem \$this->layout('name-of-layout/layout');

**Controller** — Kontrolery řídí logiku aplikace a jsou stejně jako modely vytvářeny pomocí továren na stejném principu. Z ContainerInterface je tentokrát vytážen příslušný model a jelikož pro každý kontroler je v tomto případě vytvořena samostatná továrna, vrací továrna místo \$requestedName rovnou jméno kontroleru. Kontrolery jsou potomky AbstractActionController.

## ORM

Ačkoli objektově relační mapování ve velké řadě případů zjednoduší kód, v této aplikaci bude použit přístup ORM pouze na jednu tabulku ačkoli je to trochu nevšední způsob. Entita User zde představuje tabulku uživatel, se kterou se pracuje nejvíce a ORM technologie tuto práci více zpřehlední a to především při přihlašování uživatelů či při změně osobních údajů. Jedním z důvodů pro použití ORM právě u této tabulky je abstrakce této tabulky a používání metod findBy namísto psaní SQL dotazů, které jsou využívány u ostatních tabulek.



Obrázek 2: Class diagram tabulky "uzivatel"

### Přístup a práce s databází

Práce s databází je realizována pomocí phpMyAdmin. Jedná se o free softwarový nástroj pro správu databáze. Práce s databází při vývoji probíhá přes webové rozhraní localhost/phpmyadmin.

Nejdříve je potřeba v projektu nadefinovat přístupový adapter. V global.php pod klíčovým slovem "db" nadefinujeme driver a dsn.

```

'db' => [
    'driver' => 'Pdo',
    'dsn' => 'mysql:dbname=name-of-database;
            host=localhost;
            charset=utf8',
]
  
```

Dále je třeba definovat username a password.

Pro další předávání a práci s adapterem byla vytvořena třída Adapter.php, která zajišťuje kompletní připojení k databázi s přidanou funkcí `getArray($st)`, která vrátí pole vyselektovaných dat. Tato třída je předávána továrně pro vytváření modelů. V modelu následně vytvoříme dotaz:

```
$st = $this->adapter->query( $statement, Adapter::QUERY_MODE_EXECUTE),
```

pokud se jedná o select, použijeme metodu `getArray()`,  
`$data = $this->adapter->getArray($st);`.

### ACL a AccessFiltering

Zend ACL se stará o přístup uživatelů s přidělenou rolí k daným zdrojům. Pokud se uživatel snaží přistoupit ke stránce, ke které nemá oprávnění přistupovat, je přesměrován na domovskou stránku a pomocí pluginu `flashMessenger` je informován o tom, že nemá dostatečná práva pro přístup k dané stránce.

O přístup nepřihlášených uživatelů se stará `access_filter` definovaný v `module.config.php`, kdy pro každý kontroler je definovaný seznam akcí, ke kterým má nepřihlášený uživatel práva přistupovat, následovaný klíčovým slovem `'allow' => '*'`. Kdybychom nepotřebovali rozlišit role přihlášených uživatelů, mohla by být práva řešena pouze tímto způsobem, kdy pro přihlášené uživatele by byl výpis akcí následován `'allow' => '@'`.

Role, zdroje i `AccessFilter` jsou definované v `Module.php` ve funkci `onDispatch()`.

### Mapy API

Mapy API je Javascriptová knihovna, která je volně přístupná a slouží pro vkládání map z portálu `www.mapy.cz`. Knihovna má dobrou dokumentaci s názornými ukázkami. Je možné tedy vkládat nejen samotné mapy ale docílit i interaktivní práce s mapou (Seznam.cz, a.s., ©2017). Zde je tato knihovna použita na stránce s formulářem pro získání zeměpisné šířky a délky, které jsou uloženy do databáze pro další použití ve statických mapách se značkou ukazující vybrané místo konání akce.

### Google Charts

Google Charts je volně dostupné API od společnosti Google pro zobrazení dat pomocí grafů. API nabízí velkou škálu grafů s možnostmi interakce uživatele s grafem. Například při najetí myši na sloupec, či jinou část grafu představující data, se zobrazí údaje, které daná část grafu představuje (Google Developers, [2016]).

### Visual Paradigm

Visual Paradigm je pro účely tohoto projektu používán jako aplikace pro správu a vytváření diagramů UseCase, UML a ERD. Po vytvoření ERD diagramu aplikace umožňuje vygenerování SQL dotazů pro vytvoření struktury databáze. Diagramy jsou používány ve fázi návrhování a dokumentace projektu.

### Xampp

Xampp je nástroj pro instalaci webového serveru na lokálním počítači bez použití internetu a umožňuje tak vývoj webové aplikace. Xampp podporuje práci s MySQL a nástrojem `phpMyAdmin`, který usnadňuje práci s databází.

## 5 Výsledky

### 5.1 Návrh aplikace

#### UseCase diagram

UseCase diagram vychází z dohodnutých funkcí, které by měla aplikace poskytovat.

Jak je vidět z diagramu, kořenový aktér je zde webový návštěvník, který si může zobrazit informace o oddílu. Tím je myšleno, zobrazit si aktuality, informace o historii, oddílu, hráčích, konaných akcích a zápasech, dále také může přidat příspěvek do veřejného chatu a zobrazit si fotogalerii.

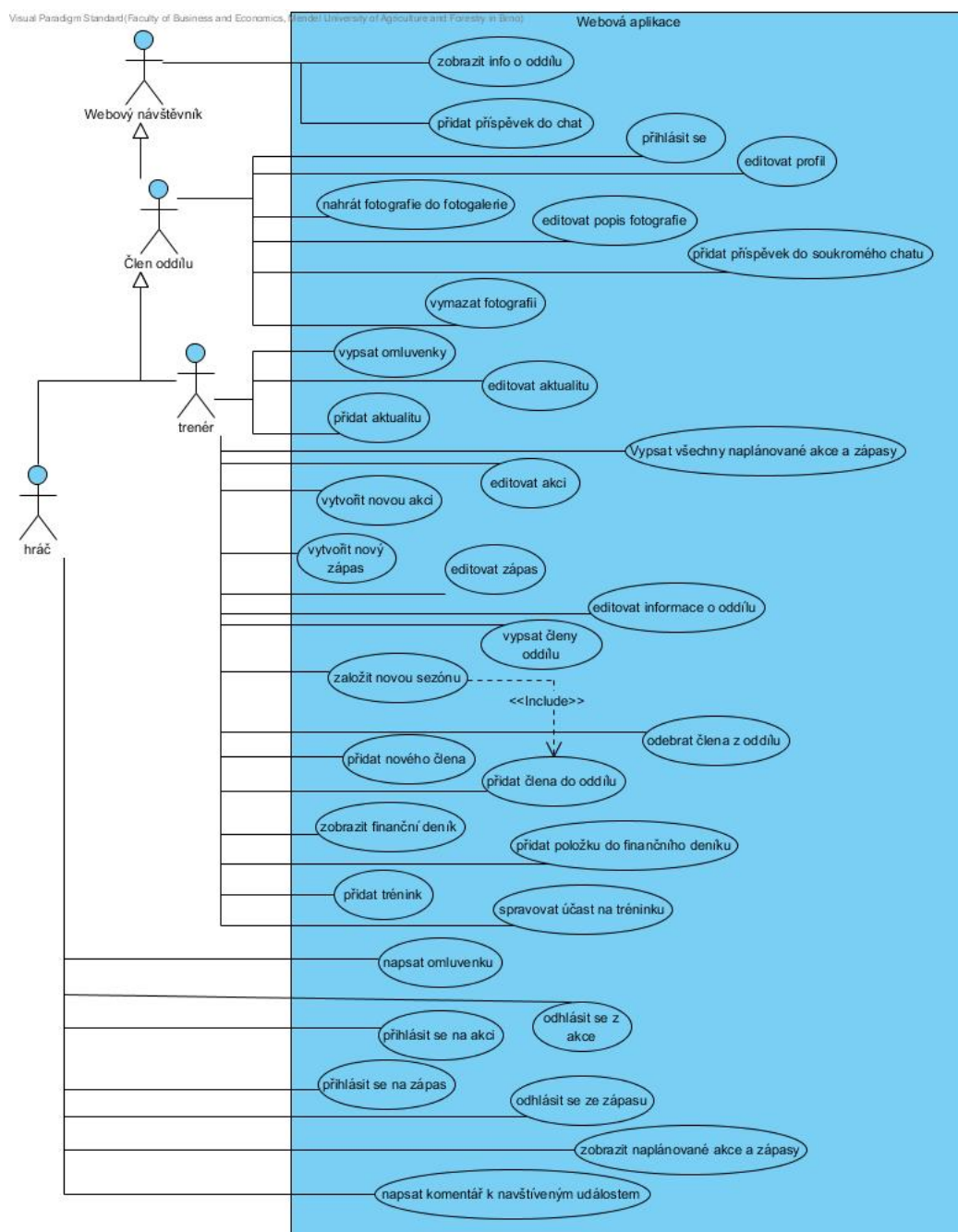
Další aktér je zde člen oddílu, který může provádět stejné funkce jako webový návštěvník, jelikož dědí všechny jeho funkce. Krom těchto funkcí se také může přihlásit do systému a po přihlášení editovat svůj profil, nahrát fotografie do fotogalerie a následně u jím přidávaných fotografií editovat popis či tyto fotografie mazat. Člen oddílu také může přidávat příspěvky do soukromého chatu přístupný pouze přihlášeným uživatelům.

Z člena oddílu dědí následně funkce ještě dva aktéři a to trenér a hráč.

Trenér může přidávat a editovat aktuality, aktualita má jako jeden z atributů datum expirace, kdy po dosažení tohoto data daná aktualita již nebude zobrazena a není zde tedy potřeba aktuality mazat. Další funkcionalitou je výpis omluvenek hráčů a dále funkcionality týkající se správy zápasů a akcí, mezi které patří především výpis všech plánovaných akcí a zápasů. Čili s datem konání menším než je dnešní datum, nehledě na to pro jaký tým byla událost vytvořena, a následně editace těchto událostí a vytváření nových. Trenér může také editovat informace o jednotlivých týmech a mazat či přidávat jejich členy. Při spuštění funkce přidání nové sezóny jsou automaticky hráči, kteří byli členy týmu v minulé sezóně, přidáni i do nově vytvořené sezóny do týmu, ve kterém působili minulou sezónu. Trenér pak dle potřeby toto členství upravuje. Kromě přidávání členů do týmu trenér také vytváří nové členy. Pro běžný chod oddílu je především potřeba editace docházky na trénincích. Trenér tedy může vytvářet jednotlivé tréninky a zadávat účast či neúčast.

Poslední funkcionality se týkají financí celého oddílu. Trenér si může zobrazit a filtrovat přehled financí a přidávat nové položky do tohoto finančního deníku.

Hráč může psát omluvenky, přihlašovat se na akce a zápasy, na které byl nominován trenérem (nominace probíhá v rámci vytváření nové události či její editaci) a následně se z událostí, na kterých je přihlášen může odhlásit. Po skončení konání události může k dané události napsat komentář viditelný i pro nepřihlášené návštěvníky.



Obrázek 3: UseCase diagram

## Entitně relační diagram

ERD diagram prezentuje základní strukturní předlohu pro vytvoření databáze. Daná databáze tedy obsahuje 16 tabulek. Jak již bylo zmíněno výše pro tabulku **uzivatel** je vytvořen ORM objekt usnadňující práci s danou tabulkou. Nad ostatními tabulkami tak budou vytvářeny klasické sql dotazy.

Tabulka **uzivatel** je středem samotné databáze, kdy oddělená bez jakékoliv vazby stojí jen tabulka **aktualita** a **finance** s následnou vazbou na **fkategorie**.

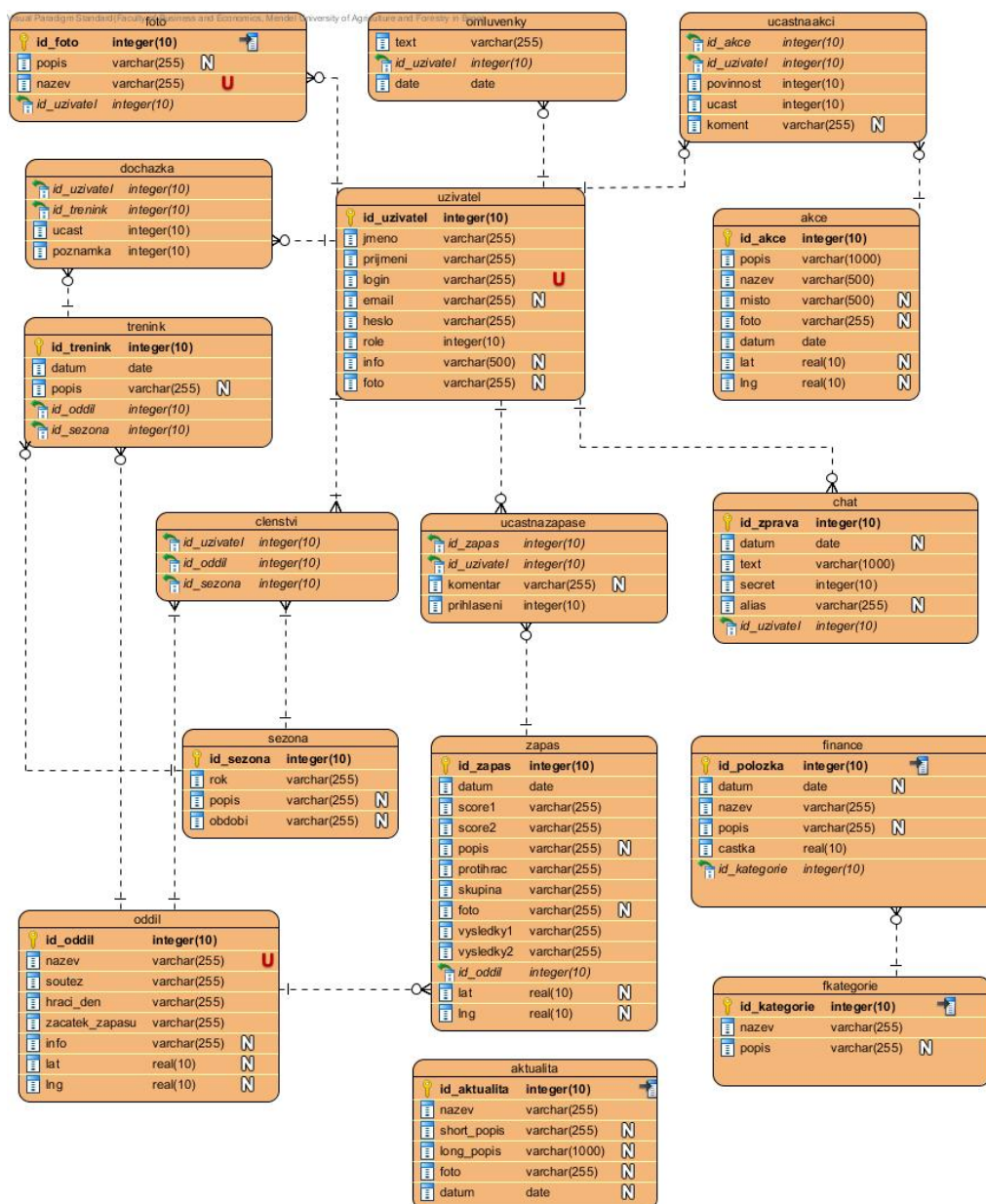
Tabulka **foto** v sobě uchovává informace o fotografiích zobrazovaných v sekci fotogalerie a má vazbu s tabulkou **uzivatel**, kdy v podobě cizího klíče uchovává informaci o tom, kdo danou fotku přidal, aby ji daný uživatel mohl následně editovat. Na stejném principu existuje i tabulka **omluvenky** a **chat**.

Tabulka **chat** navíc v atributu *secret* uchovává informaci o tom zda zpráva patří do chatu veřejného, tedy i pro nepřihlášené uživatele jejichž jméno, pod kterým odešlou danou zprávu, se ukládá pod atribut *alias*, nebo zda je to zpráva soukromá, tedy pouze pro přihlášené členy. V tomto případě nabude daný atribut hodnotu 1 v opačném případě hodnotu 0. Pro uchování informace o účasti na akci či na zápase jsou vytvořeny tabulky **zapas** a **akce**, jelikož jejich vztah s tabulkou **uzivatel** je M:N (uživatel se může zúčastnit více akcí a na dané akci může být více uživatelů), je potřeba také doplnit další informace týkající se dané účasti konkrétního uživatele, jako je například jeho komentář. Za tímto účelem byly vytvořeny vazební tabulky **ucastnazapase** a **ucastnaakci**, které spojují tyto dvě tabulky tím, že obsahují jako cizí klíče primární klíče daných tabulek, a ukládají doplňující informace. V tabulce **ucastnaakci** se dále nacházejí atributy *povinnost* a *ucast*. Atribut *povinnost* uchovává informaci o tom, zda je akce pro uživatele povinná a atribut *ucast* zaznamenává, zda se uživatel na danou akci přihlásil, což determinuje to, že se jí skutečně zúčastní a následně může do atributu *koment* zapsat svůj komentář k dané akci. To stejné je v tabulce **ucastnazapase** zaznamenáno v atributu *prihlaseni*. Na podobný způsob je vytvářena i tabulka **dochazka**, která spojuje tabulku **uzivatel** a **trenink**.

V tabulce **dochazka** se opět nachází atribut *ucast* a atribut *poznamka*. V tabulce **trenink** jsou uchovávány informace o tréninku jako je například datum či popis, které vkládá trenér při zadávání tréninku a dále také cizí klíče *id\_oddil* a *id\_sezona*, které odkazují na záznam do daných tabulek, které sdělují v jaké sezóně a pro jaký oddíl se trénink konal. Tabulka **oddil**, kromě atributů popisujících daný oddíl obsahuje atributy *lat* a *lng* sloužící pro zaznamenání zeměpisné šířky a zeměpisné délky místa, kde se odehrávají tréninky. Stejně atributy obsahují i tabulky **zapas** a **akce**, pro zaznamenání místa, kde se daná událost koná. Další vazební tabulkou je zde tabulka **clenstvi**, která zaznamenává pomocí cizích klíčů *id\_oddil*, *id\_sezona* a *id\_uzivatel*, informace o tom, který uživatel byl v které sezóně v kterém oddílu.

Tabulka **finance** uchovává informace jako je datum, název, popis a částka o jednotlivých záznamech ve finančním deníku a pomocí cizího klíče *id\_kategorie* odkazuje na záznam v tabulce **fkategorie**, kde se nachází příslušná kategorie. Ta-

bulka **aktuality** slouží pro uchování informací o aktuálním dění v kroužku, kdy aktualita je zobrazena pouze do dne zapsaného v atributu *datum*.



Obrázek 4: Entity relationship diagram

## 5.2 Web design

Pro návrh konzistentního a graficky správně navrženého webu je nejdříve podstatné vědět, pro koho web navrhujeme a z jakých zařízení budou k webu přistupovat. Podle toho pak může být správně navrhnutá informační architektura a zhotoven hrubý model stránek.

### Předpokládání návštěvníci

Návštěvníky webu daného volejbalového oddílu můžeme rozdělit do tří hlavních skupin.

První skupina se skládá z členů oddílu, tedy z hráčů a hráček. Těm web slouží k získání potřebných informací týkajících se především chystaných akcí, zápasů a jiných aktualit. Na web přistupují přes všechny druhy zařízení, přes mobilní telefony, tablety, notebooky a desktopové počítače.

Druhá skupina zahrnuje trenéry, kteří na web zadávají aktuální informace pro hráče a zároveň zde musí být prezentovány informace o nich jakožto vedoucích oddílu. Můžeme říct, že daná skupina bude na web přistupovat převážně přes desktopový počítač nebo notebook.

Třetí skupina jsou rodiče stávajících, ale i potenciálních hráčů a hráček, kteří hledají informace o tom, kde a v kolik hodin se odehrávají tréninky a zápasy, kdo jejich děti povede, jaká je historie oddílu a kolik členství v oddílu stojí. Stejně jako u druhé skupiny předpokládáme přístup přes desktopový počítač či notebook.

### Organizační schéma

Vzhledem k povaze daného webu použijeme tematické zařazení objektů. Jednotlivé stránky zařadíme dle tématu. Hlavní stránka bude obsahovat aktuální informace o dění v oddíle a základní informace včetně fotografií. V horním panelu v menu bude pět hlavních kategorií.

První kategorie s názvem **Historie** obsahuje stručný popis vývoje oddílu s odkazem na popis jednotlivých sezón.

Druhá a zároveň hlavní kategorie jsou **Týmy**, ta se dále rozděluje na subkategorie **Juniorky**, **Kadetky**, **Žákyně**, **Pohár 7. tříd** a **Přípravka**. Zde uživatel najde informace o tom, které hráčky patří do které skupiny (soupiska), výsledky zápasů daného týmu v dané sezóně a procentuální docházku hráček na tréninky.

Třetí kategorie s názvem **Trenéři** obsahuje informace o vedení oddílu.

Čtvrtá kategorie s názvem **Akce** zobrazuje seznam jednotlivých akcí a odkaz na podrobné informace o akci (popis, hráčky, které se zúčastnily a jejich komentáře a mapa zobrazující místo konání).

Pátá kategorie **Fotogalerie** zobrazuje fotografie nahrávané přihlášenými uživateli.

Šestá kategorie **Chat** slouží pro komunikaci s návštěvníky webu.

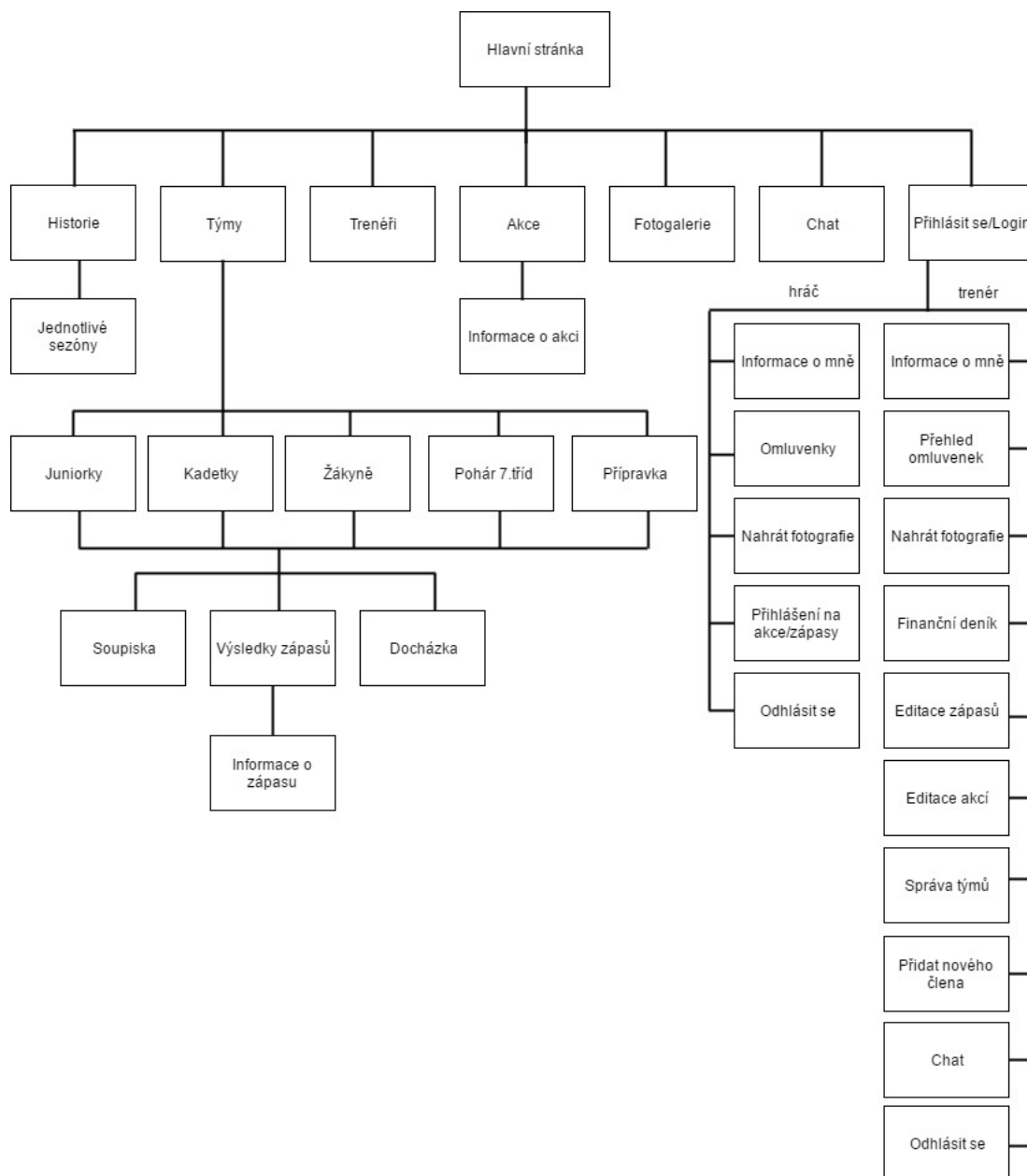


Na pravé straně navigačního menu se zobrazuje v případě nepřihlášeného uživatele možnost **Přihlásit se** a v případě přihlášeného uživatele jeho login v podobě rozevíracího menu s odlišnými subkategoriemi dle role přihlášeného uživatele.

Uživatel v roli trenéra může stejně jako hráč editovat informace o své osobě, jako je například změna jména, příjmení, přihlašovacího loginu a hesla, či přidání doplňujících informací. Dále si trenér může zobrazit omluvky od hráčů v sekci Přehled omluvenek, které se mu vypíší za poslední měsíc.

Další kategorie je společná i pro hráče a to je přidání fotografie. Tyto fotografie jsou následně zobrazeny ve fotogalerii a přihlášený uživatel, který danou fotografii přidal ji následně může mazat či měnit popis.

Následují kategorie **Finanční deník**, **Editace zápasů**, **Editace akcí**, **Správa týmu**, **Přidat nového člena**, **Chat** (myšleno soukromý chat pouze pro přihlášené uživatele) a poslední kategorie je společná a slouží pro odhlášení. Uživatel v roli hráče má kategorie **Omluvenky**, kde píše své omluvky k tréninkům a dále **Přihlášení na zápasy/akce**, kategorie **Informace o mně**, **Nahrát fotografie** a **Odhlásit se** jsou stejné jako u role trenér.



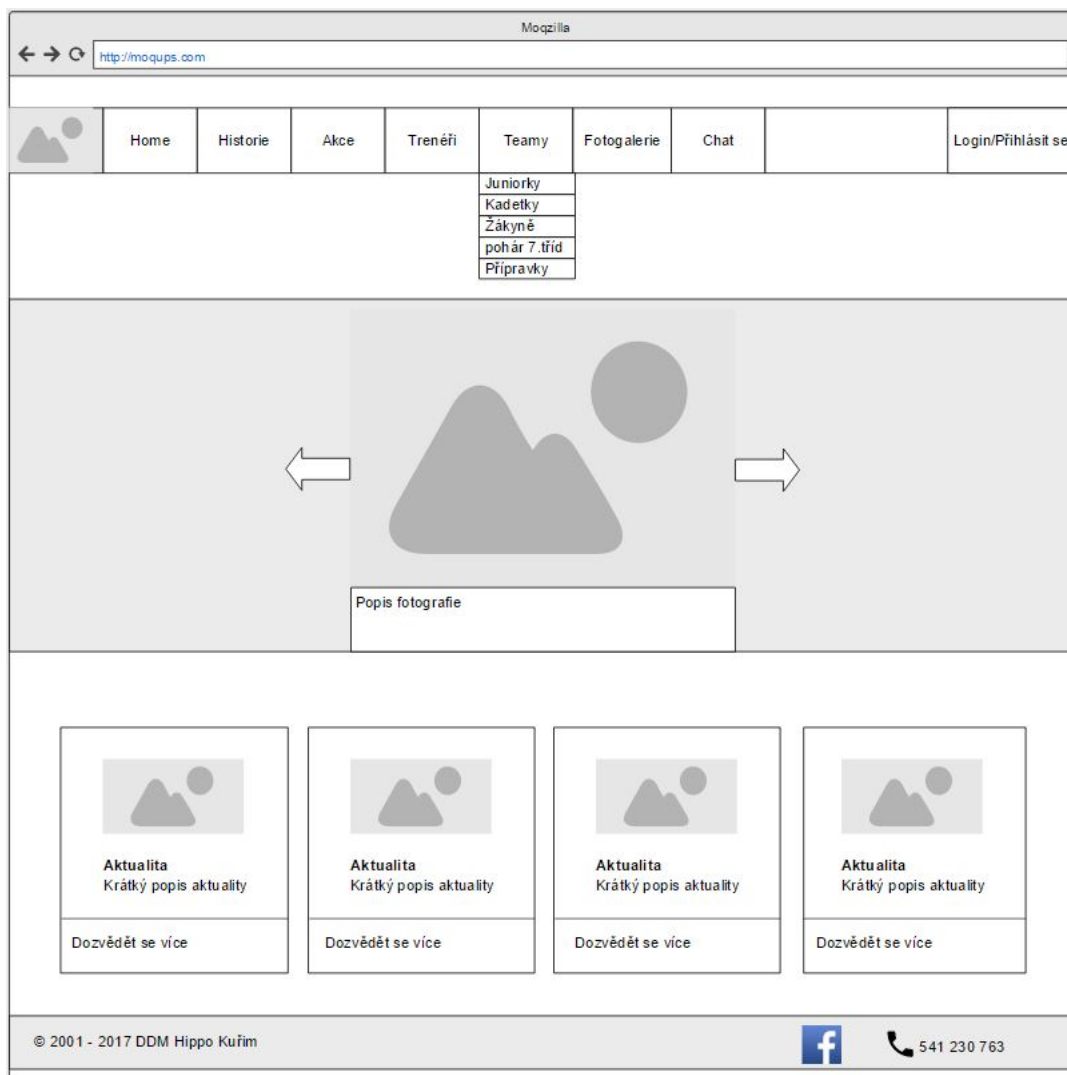
Obrázek 5: Diagram webového serveru

## Navigace

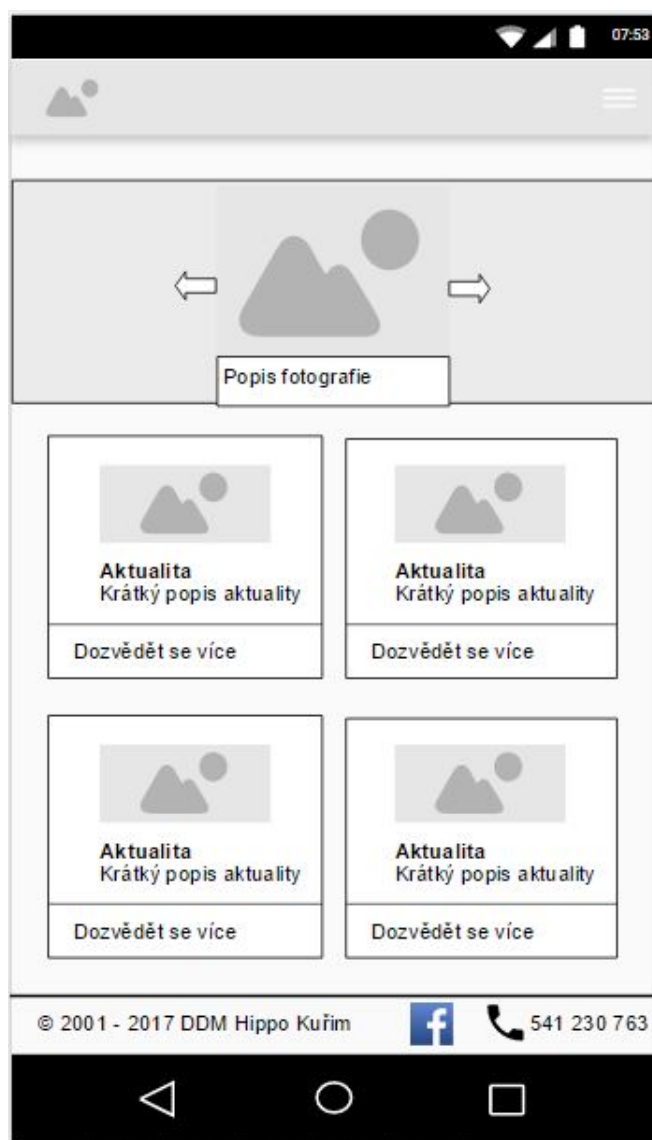
Primární navigace bude v podobě horizontálního menu umístěného u horního okraje stránky. Před tímto panelem je v případě desktopové verze umístěn panel s obrázkem, reprezentujícím maskota týmu a vedle obrázku je umístěn název oddílu. Panel bude třídit obsah do kategorií. Některé kategorie je potřeba už v hlavní navigaci rozdělit do podkategorií, za tímto účelem bude použita sekundární navigace v podobě tzv. "drop down" menu. V rámci přizpůsobení navigace pro mobilní zařízení, bude pro toto zařízení použito tzv. "hamburger" menu. Na pravé straně se nachází v případě nepřihlášeného uživatele slovní spojení "Přihlásit se". V případě přihlášeného uživatele jeho login, kdy po kliknutí se zobrazí dropdown menu s možnostmi pouze pro přihlášeného uživatele odvozené od jeho role.

## Prostorové rozložení informací

Hlavní stránka bude obsahovat prezentaci fotografií a pod ní se bude nacházet kontextová navigace v podobě dlaždic odkazujících na aktuální informace (například na informace o náboru do oddílu, či na chystanou akci). V případě, že se jedná o přihlášeného uživatele s rolí "trener", objeví se pod přehledem aktuálních informací tlačítko se zástupnou ikonou "plus" pro přidání nové aktuality.

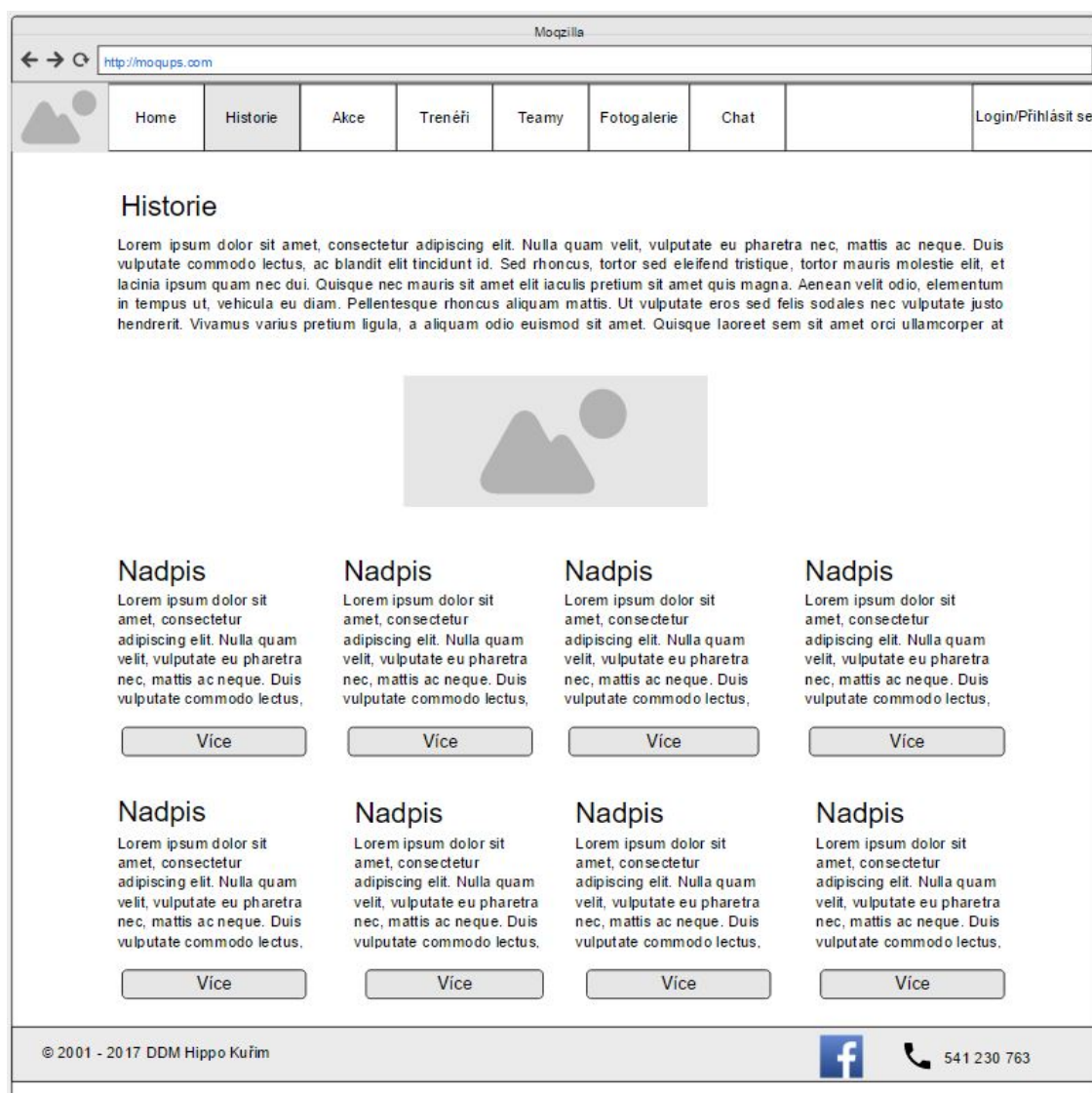


Obrázek 6: Hlavní stránka - desktopová verze



Obrázek 7: Hlavní stránka - mobilní verze

Vybraná podstránka historie bude obsahovat úvodní text a fotografii a dále krátké úryvky textů s následnými kontextovými odkazy, v podobě tlačítka, na daný text. Jednotlivé informace o sezónách se následně otevrou v tzv “tiny boxu”. Na stejný způsob je řešena většina odkazů, kdy cílený obsah není příliš objemný. Toto řešení se tedy týká například i formulářů či zobrazení fotek v plné velikosti. Za tímto účelem byla použita Javaskriptová knihovna TinyBox, která umožňuje zobrazení modálních oken dle potřeby pomocí nadefinovaných parametrů (Scriptiny, ©2009). Knihovna byla upravena tak, aby se modální okno zobrazovalo správně i na mobilních zařízeních.



Obrázek 8: Podstránka - desktopová verze



Obrázek 9: Podstránka - mobilní verze

### Barevné řešení

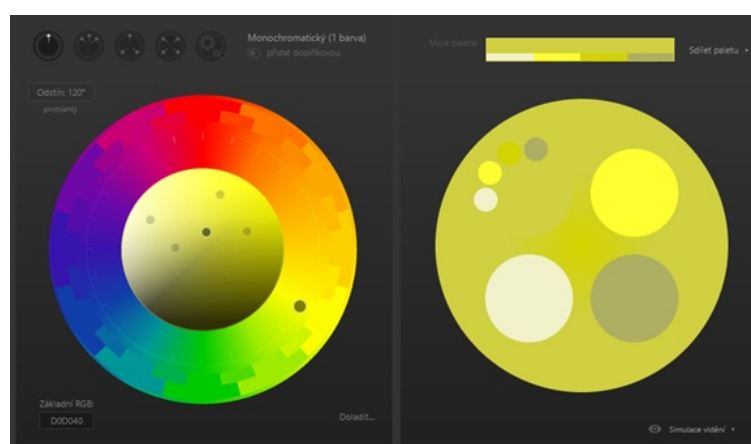
Barevný návrh vychází především z loga oddílu. Text bude použit černý na bílém podkladě. Další dvě barvy které budou použity jsou šedá a žlutá. Šedá barva bude použita převážně pro navigaci a hlavní ovládací prvky. Žlutá pak bude použita pro zvýraznění a tlačítka. Cílem je aby stránka na uživatele působila jednoduše, elegantně a poskytla správně prezentované informace.



Obrázek 10: Logo oddílu

### Barevné schéma

Jelikož budou použity barvy šedá a žlutá, jedná se o schéma monochromatické. Schéma tedy vychází z odstínu žluté a pomocí změny sytosti a jasu dostaneme šedou barvu.



Obrázek 11: Barevné schéma



## 5.3 Použité principy Zend Frameworku

V této části práce budou uvedeny principy Zend Frameworku, které jsou v práci použity. Na konkrétních příkladech z práce a ukázkách kódu bude nastíněna práce s danými komponentami.

### Instalace

Jak již bylo řečeno instalace projektu a především tedy Zend Frameworku a jeho knihoven probíhá přes composer. Po instalaci samotného composer následuje instalace Zend Frameworku přes příkazový řádek ve zvolené složce. Composer tedy nainstaluje zadané balíky a vytvoří kostru projektu.

Pro instalaci Zend Frameworku slouží příkaz `composer require zendframework/zendframework` a pro vytvoření projektu následně příkaz `composer create-project -n -sdev zendframework/skeleton-application path/to/install` (Zend, ©2006-2017-2). Po provedení těchto příkazů v příkazové řádce budou nainstalovány všechny potřebné komponenty a vytvořena kostra projektu ve složce, která byla zadána v části `path/to/install`.

### Továrny

Za pomoci Zend Service Manager jsou jednotlivé kontrolery vytvářené pomocí továren. Pokud nepotřebujeme vytvářet továrnu, tedy pokud kontroleru nepotřebujeme předat žádné parametry (například adapter či model), můžeme použít předdefinovanou továrnu `InvokableFactory`. Jako příklad bude uvedena továrna sloužící pro vytváření jednotlivých modelů.

```
class ModelControllerFactory implements FactoryInterface
{
    public function __invoke(ContainerInterface $container, \
        $requestedName, array $options = null)
    {
        $entityManager = $container->get('doctrine.entitymanager
            .orm_default');
        $adapter = $container->get(AdapterManager::class);
        $authService = $container->get(\Zend\Authentication\
            AuthenticationService::class);
        $userManager = $container->get(UserManager::class);
        return new $requestedName($entityManager, $adapter,
            $authService, $userManager);
    }
}
```

Pro vytvoření továren je použito objektu `FactoryInterface`, který se přes atribut `$container` dostane ke všem potřebným objektům a třídám v aplikaci, které jsou

uloženy do proměnných a následně předávány samotnému kontroleru. Uvedená továrna nevytváří žádný konkrétní model přímo, ale při použití `$requestedName` je možné pomocí jedné továrny vytvářet více tříd. Jaké třídy budou vytvořeny pomocí které továrny je definováno v `module.config.php` v rámci příslušného modulu, v poli `factories`.

```
'service_manager' => [
    'factories' => [
        Model\ChatModel::class => Model\Factory\
            ModelControllerFactory::class ,
        Model>MainModel::class => Model\Factory\
            ModelControllerFactory::class ,
    ],
],
```

Místo uvedeného `service_manager` však může být například v případě kontrolerů `controllers`.

## Formuláře

Formuláře jsou vytvářeny pomocí balíku Zend Form. Pro přehlednost jsou formuláře umístěny ve složce `Forms` v modulu `Application` zaobalené do tříd a následně v kontroleru vykresleny a v případě potřeby upraveny a předány do `views`, kde jsou vyvolány jednotlivé elementy. Každý element musí mít definované jméno a typ, dále je možné do sekce `options` definovat označení, či v případě typu `Select` prázdnou hodnotu a vybírané hodnoty. V sekci `attributes` je pak možné definovat hodnotu, třídu či `id`. Ve formulářích je také možné definovat validátory a filtry, kde je možné použít buď již předdefinované z knihoven a nebo si definovat své vlastní. Pro příklad bude uvedena část formuláře pro přidání příspěvku do chatu.

```
class chatForm extends \Zend\Form\Form {

    public function __construct($name = null) {
        parent::__construct('mesic-form');
        $this->setAttribute('method', 'post');

        $this->add([
            'name' => 'alias',
            'type' => Element\Text::class,
            'options' => [
                'label' => 'Zobrazené jméno',
                'label_attributes' => [
                    'class' => 'mdl-textfield__label'
                ]
            ],
            'attributes' => [
```

```

        'class' => 'mdl-textfield__input',
        'id' => 'jmeno'
    ],
    'filters' => [
        ['name' => 'StringTrim'],
    ],
    'validators' => [
        [
            'name' => 'StringLength',
            'options' => [
                'min' => 1,
                'max' => 255
            ],
        ],
    ],
]);
...
}
}

```

Třída `chatForm` vytváří formulář s prvky `alias`, představující jméno pod kterým píše nepřihlášený uživatel. V případě, že do chatu přidává příspěvek přihlášený uživatel toto textové pole se mu nezobrazí, protože již v kontroleru bude odebráno pomocí `$form->remove("alias")`. Dalšími prvky, které však nejsou v ukázce uvedeny, jsou `text` a `submit` pro odeslání formuláře. Jak je z příkladu vidět třída `chatForm` rozšiřuje třídu `Form` a v konstruktoru se nastaví aktuální název formuláře, který je v šabloně prezentováno jako selektor `id` a `name` v elementu. Na dalším řádku je pomocí `setAttribute` nastavená metoda pro odeslání formuláře na `POST`. Nový prvek do formuláře je definován pomocí `$this->add([...])` a následně musí být nastaveno unikátní jméno prvku a typ. Poté je do pole `options` možné doplnit označení prvku a hodnotu selektoru `class` pro element daného prvku. Dále v poli `attributes` je možné definovat selektor `class`, `id` a v případě textového pole i `rows` pro samotný element. Je možné samozřejmě definovat i filtry a validátory pro daný prvek a to za použití již předdefinovaných tříd v Zendu jako je například validátor `StringLength` pro omezení délky textu, či si definovat vlastní validátory. Pro vytvoření úhledného vzhledu formulářů jsou použity třídy `mdl`. `Material Design Lite` je balík stylů, který optimalizuje vzhled stránek napříč zařízeními (Google, [2015]). Většina formulářů se po jejich zavolání otevře v novém modálním okně. To má za účel uživatele zbytečně nerozptylovat a přimět ho soustředit se pouze na vyplnění formuláře.

The image shows a web form titled "Přidat akci" (Add event) with the following fields and elements:

- Datum** (Date): A text input field with the placeholder "dd.mm.mrr".
- Název** (Name): A text input field.
- Popis** (Description): A text input field.
- Akce pro všechny** (Action for everyone): A checkbox.
- Vyberat z oddílu juniorky** (Select from juniors): A dropdown menu with options: Kamila Smrčková, Eliška Smrčková, Bára Lukešová.
- Vyberat z oddílu kadetky** (Select from cadets): A dropdown menu with options: Zuzana Rubešová, Tereza Lipová, Anna Bulgurovská.
- Souřadnice (vygenerují se z mapy)** (Coordinates (generated from map)): A label above a yellow "ODESLAT" (SEND) button.
- Mapa** (Map): A map showing a location in Turbínov, with a red pin marker.

Obrázek 12: Formulář

## Přihlašování uživatelů

Pro přihlašování a zprávu uživatelů byl použit příklad “User Demo Sample”. Autor Oleg Krivtsov zde vytvořil entitu User s potřebnými atributy, které byly modifikovány na potřeby vytvářené aplikace, a následně implementoval autentizaci a autorizaci pomocí loginu a hesla (Krivtsov, ©2017). Z tohoto příkladu byl převzat i postup práce s Access Filter avšak implementován pouze pro nepřihlášené uživatele, pro přihlášené uživatele jsou definovaná práva pomocí Zend\_Acl.

## 5.4 Implementace

Celá aplikace se skládá ze dvou modulů a to modulu User pro správu a editaci uživatelských profilů a modulu Application, který obsahuje aplikační logiku celé aplikace. V každém modulu ve složce config je soubor module.config.php, který obsahuje mimo jiné i definice routovacích cest a definování kontrolerů.

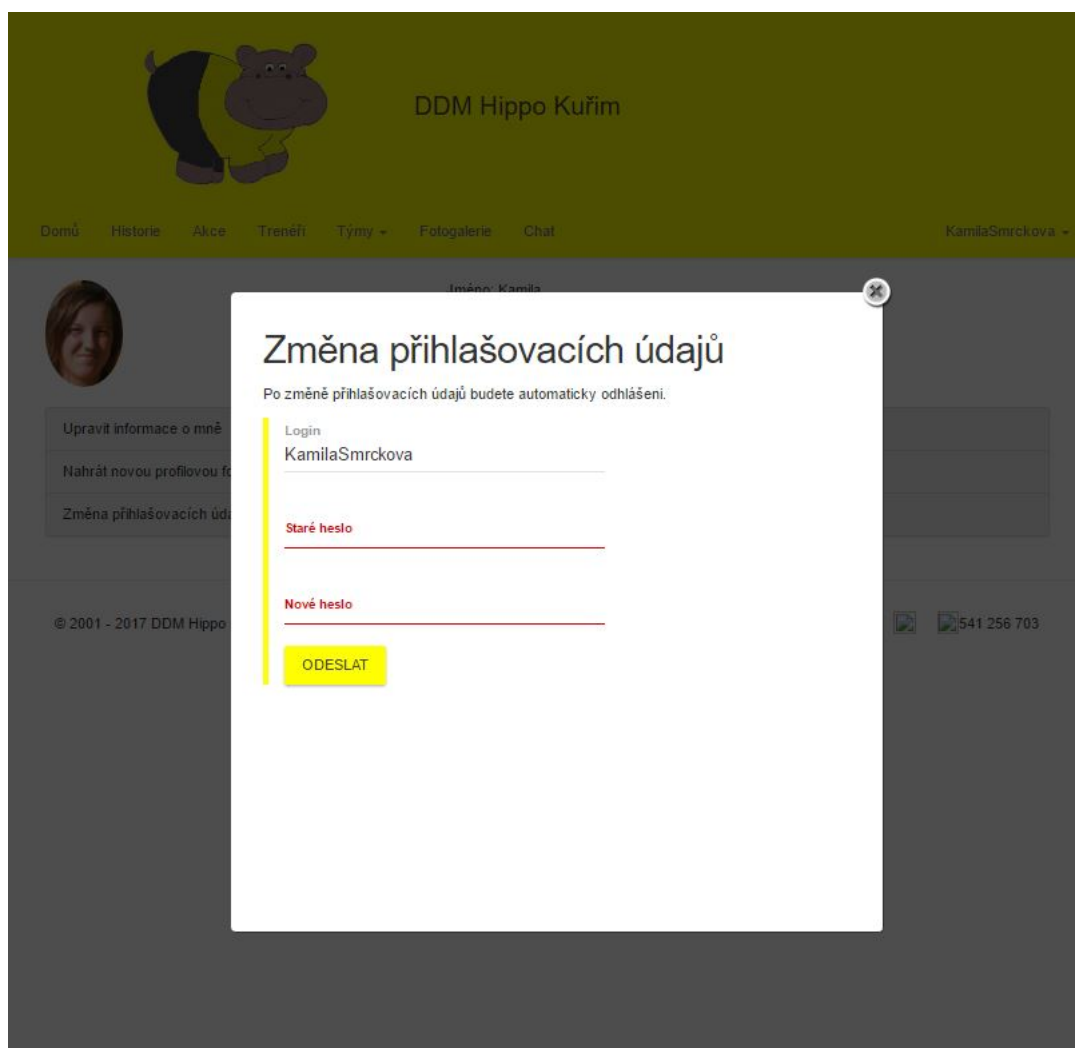
### Modul User

Jak již bylo nastíněno v podkapitole Přihlašování uživatelů, základní logika správy uživatelů byla převzata z příkladu “User Demo Sample”. Modul pro přístup k databázi používá objektově relační mapování pro práci s tabulkou *uzivatel*. Za tímto účelem byla ve složce src/Entity vytvořena třída User, která definuje danou tabulku a metody pro práci s atributy.

Ve složce src/Service se nachází třídy AuthAdapter, AuthManager a UserManager, které zajišťují autentizaci a autorizaci uživatele. Třída UserManager obsahuje funkce pro vytváření nových uživatelů, změnu přihlašovacích a osobních údajů uživatele.

Tento modul obsahuje ve složce src/Controller pouze dvě třídy a to třídu AuthController, která zajišťuje přihlašování a odhlašování uživatelů. Funkce loginAction používá formulář LoginForm uložený ve složce src/Form. V této třídě je vytvořena instance tohoto formuláře, která je následně předána do příslušného pohledu, který je ve funkci vytvořen. Třída UserController obsahuje již zmíněné funkce pro editaci profilu.

V modulu User je také ve třídě Module ve funkci onDispatch definován seznam přístupových práv platný pro celou aplikaci.



Obrázek 13: Změna přihlašovacích údajů

## Modul Application

Funkcionality aplikace jsou rozděleny do několika tříd. Jak již bylo uvedeno v modulu User jsou zakomponovány funkcionality týkající se správy uživatelů. V modulu Application se nachází všechny ostatní funkcionality, které jsou zařazeny do osmi kontrolerů. Každý z kontrolerů je vytvořen pomocí továrny, kde mu je předán jako parametr příslušný model. Tyto kontrolery obsahují akce, které vytváří příslušné pohledy. Pohledy jsou uloženy ve složce `./view/nazevKontroleru`.

Třída `IndexController` se stará o prezentaci úvodních informací o oddílu. Vytváří hlavní stránku a podstránky historie i jednotlivé sezóny, které jsou zobrazovány dle získaného paramteru z URL. Zpracovává také dotaz uživatele na aktuální informace, které jsou umístěny na hlavní stránce.

Třída `AkceController` spravuje informace o akcích, avšak stará se pouze o funkcionality, které se netýkají vytváření a editace akcí. Vytváření a editace akcí je možné pouze pod účtem s přidělenou rolí *trener* a všechny funkcionality určené pouze pro tuto roli jsou uloženy ve třídě `TrenerController`. Třída `AkceController` se tedy stará pouze o zpracování informací týkající se výpisu jednotlivých akcí a zobrazení detailních informací o vybrané akci.

Třída `AgendaController` zajišťuje funkcionality, které jsou společné jak pro roli *trener* tak pro roli *hrac*. Tyto funkcionality jsou například získání informací o zápase nebo editace profilové fotografie.

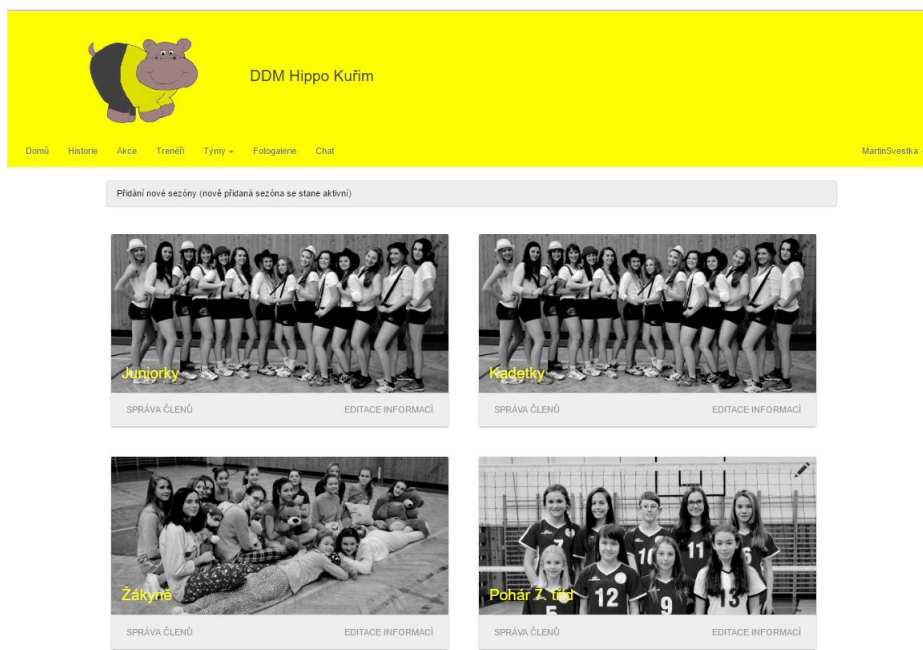
Třída `ChatController` se stará o ukládání a zobrazování zpráv a to jak veřejných tak mezioddílových.

Třída `FotoController` zajišťuje správné ukládání fotografií do fotogalerie a také jejich vykreslení do příslušného pohledu. Součástí této třídy jsou i funkce pro editaci a mazání fotografií.

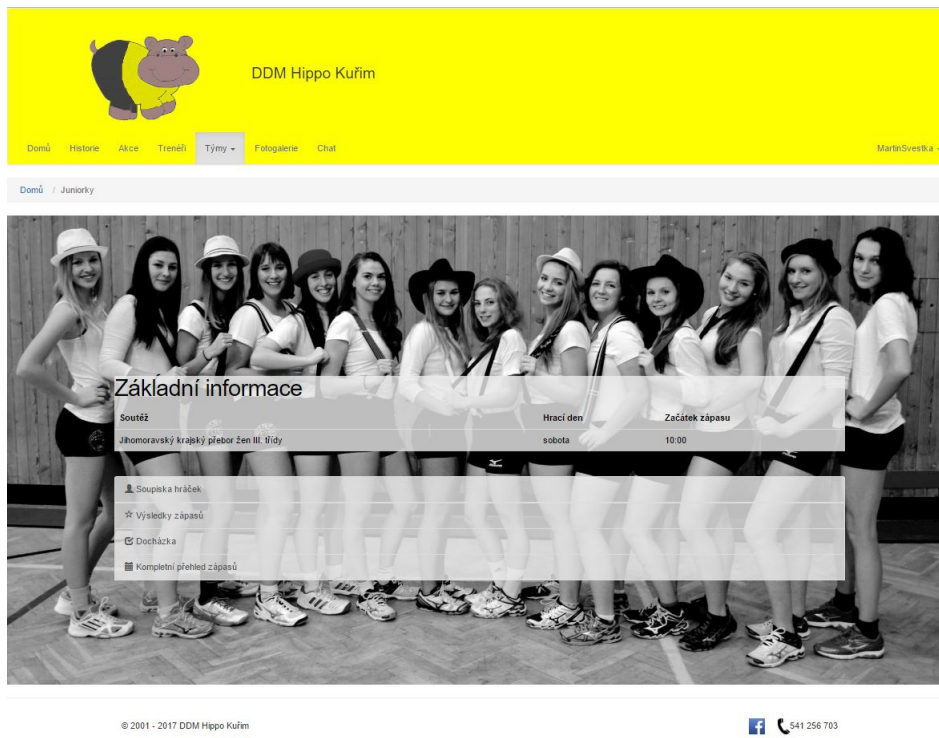
Třída `TeamyController` zpracovává požadavky uživatelů na informace o jednotlivých týmech. Dle parametru z URL zobrazí informace o oddílu a dále také o soupisce hráčů vybraného oddílu, jejich docházce na tréninky a výsledcích zápasů.

Třída `HracController` slouží k definování funkcionality dostupných pouze pro uživatele přihlášených pod účtem s přidělenou rolí *hrac*. Jedná se o psaní omluvek, přidávání komentářů k zápasům a akcím, přihlašování na akce a zápasy a také odhlašování ze zápasů a akcí.

Třída `TrenerController` je nejobsáhlejší třída. Definuje všechny funkcionality pro uživatele s rolí *trener*. Třída se stará o ukládání záznamů do finančního deníku a také o filtrování a výpis těchto záznamů, dále také o vytváření a editaci aktuálních informací zobrazených na hlavní stránce. Pro chod celé aplikace je hlavní správa týmů zahrnující nejen editaci informací o týmech, ale i přidávání a odebrání členů z týmů. Mezi hlavní funkcionality aplikace patří i správa zápasu a akcí. V této třídě jsou definovány funkce pro vytváření, editaci a mazání zápasů a akcí. Při vyvážení či editaci události trenér může pomocí výběracího pole ve formuláři vybrat členy, kteří se mají události zúčastnit, těm se pak událost zobrazí a mohou se na ni přihlásit. Jak již bylo řečeno, přihlašování a odhlašování z událostí řeší třída `HracController`.



Obrázek 14: Rozcestník správy oddílů



Obrázek 15: Prezentace týmu juniorky



## 5.5 Testování aplikace

### Automatické testy pomocí PHPUnit

Jednotkové testy testují funkcionalitu jednotlivých komponent. Při testování je tedy komponenta zavolána, je změněn její stav a testuje se, zda se získaný výsledek shoduje s očekávaným (Böhmer, 2010, s. 343). Ve funkci setUp() je nastavena konfigurace testovaného modulu. Hlavní metoda je zde metoda setApplicationConfig(), která nastaví odpovídající konfiguraci celé aplikace. Ve funkci testIndexActionCanBeAccessed() jsou testovány pomocí “assertů”, které reprezentují podmínky, jednotlivé argumenty a tyto podmínky musí být splněny. Po napsání testů je v příkazové řádce spuštěno testování pomocí příkazu “vendor/bin/phpunit” a PHPUnit vygeneruje přehled, ze kterého je možné zjistit, které testy prošly a které ne. Je zde také možné vidět proč neprošly.

Pomocí PHPUnit testů je například možné testovat, zda daná stránka obsahuje dotazovaný prvek, či zda se stránka načítá metodou GET nebo POST. Jedním z provedených testů je testAssertTemplateName(), který zjišťuje zda zavolaná stránka pomocí metody dispatch() renderuje do dotazované šablony.

```
public function testAssertTemplateName()
{
    $this->dispatch('/akce/detail');
    $this->assertTemplateName('layout/empty');
    $this->assertTemplateName("application/akce/detail");
}
```

Akce new-aktualita obsahuje formuláře, kdy při vyvolání akce je vrácen prázdný formulář pomocí metody GET a následně jsou data z formuláře odeslána pomocí metody POST. Pro otestování zda tomu tak skutečně je byly napsány dva testy, testDefaultDispatchMethod() pro ověření metody při načtení formuláře a testDispatchMethodSetOnRequest() po odeslání formuláře.

```
public function testDefaultDispatchMethod()
{
    $this->dispatch('/trener/new-aktualita');
    $this->assertEquals('GET', $this->getRequest()->
        getMethod());
}

public function testDispatchMethodSetOnRequest()
{
    $this->getRequest()->setMethod('POST');
    $this->dispatch('/trener/new-aktualita');
    $this->assertEquals('POST', $this->getRequest()->
        getMethod());
}
```

Je možné také testovat zda aplikace přesměruje nepřihlášeného uživatele na stránku s přihlašovacím formulářem. Pro zjištění zda se tak stane v případě chatu pro členy byla napsána testovací funkce `testSecretChatNotLoggedIn()`.

```
public function testSecretChatNotLoggedIn () {
    $this->dispatch ( '/chat/secret-chat ' );

    if ( $this->assertFalse ( $this->auth->hasIdentity ( ) ) ) {
        $this->assertControllerName ( \User\Controller\
            UserController :: class );
        $this->assertActionName ( 'login ' );
    }
}
```

### Testování uživateli

Dvěma uživatelům v roli hráče byly zadány následující úkoly:

1. Přihlaste se do aplikace pomocí loginu a hesla, které jste obdržel.
2. Napište omluvenku k tréninku, datum i text si zvolte.
3. Přihlaste se na zápas a akci, na které jste nominován.
4. Napište komentář k akci, na které jste byl. (pozn. akce je připravená ve výpisu akcí)
5. Změňte svoji fotografii.
6. Přidejte fotografii do fotogalerie.
7. Přidejte příspěvek do mezioddílového chatu.

Uživateli v roli trenéra pak byly zadány tyto úkoly:


1. Přihlaste se do aplikace pomocí loginu a hesla, které jste obdržel.
2. Vytvořte novou sezónu a do týmu juniorky přidejte vámi zvolené hráče.
3. Upravte informace o týmu juniorky.
4. Přidejte zápas pro tým juniorky a nominujte na něj tři hráče.
5. Přidejte akci a nastavte, že tato akce je pro všechny hráče.
6. Přidejte nového člena oddílu s rolí hráč a následně tohoto hráče přidejte do týmu juniorky a změňte příjmení tomuto uživateli.
7. Na hlavní stránce vytvořte novou aktualitu.

8. Vytvořte nový záznam ve finančním deníku, k tomuto záznamu vytvořte i příslušnou kategorii.
9. Změňte svoji fotografii.
10. Přidejte fotografii do fotogalerie.
11. Přidejte příspěvek do mezioddílového chatu.

Po seznámení s aplikací neměl žádný z vybraných uživatelů zásadní problém s vykonáním zadaných úkolů. Pouze u přidávání hráčů a to buď do sezóny nebo nominace na akce či zápasy byla vznesena námitka, že textové pole s uživateli je příliš malé a tím pádem je výběr méně přehledný. Velikost textových polí se však dá hůře odhadnout, protože není možné přesně určit kolik záznamů se v polích objeví.

### **Kufrový test navigace**

1. O jaký server se jedná? Návštěvník pozná, o jakou stránku se jedná podle loga a názvu klubu umístěného u navigace v záhlaví.
2. Na jaké jste stránce? V desktopové verzi je v navigaci daná stránka zobrazena tak, že vypadá jak tzv. „zmáčklé“ tlačítko a pokud jsou stránky více vnořeny, zobrazuje se tzv. drobečková navigace. U podstránek se zobrazená stránka pozná podle nadpisu, jediná stránka bez nadpisu je tedy hlavní stránka.
3. Jaké jsou hlavní sekce tohoto serveru? V desktopové verzi je klasické horizontální menu viditelné na každé stránce. V mobilní verzi je tzv. „hamburger menu“, které se při kliknutí rozevře.
4. Jaké máte na této úrovni možnosti? Návštěvník může přejít buď na nějakou z hlavní stránek v menu, anebo kliknutím na kontextový odkaz, v podobě tlačítka, modální okno s daným popisem.
5. Jakým způsobem můžete vyhledávat? Web není příliš obsahově náročný, a proto zde není potřeba vyhledávání.
6. Jak hluboko jsem v hierarchii? U stránek ke kterým se nepřístupuje přímo pomocí hlavní navigace je v horní části drobečková navigace. Na stránkách jsou často využívána tzv. modální okna a v těch navigace není uvedena.



**DDM Hippo Kuřim**

1

Domů
Historie
Aktce
Trenér
Týmy
Fotogalerie
Chat
Přihlásit se


2
3

### Informace o kroužku

Kroužek volejbalu DDM Kuřim byl založen v roce 2001 a je součástí organizace Dům dětí a mládeže Kuřim, okres Brno-venkov. Kroužek navštěvovalo asi 8 hrátek, které se zatím pouze připravovaly na soutěže, a tedy se žádného soutěžního cyklu neúčastnily. Hráčky se připravovaly pod vedením vyškoleného trenéra III. třídy Jiřího Smetány a asistenta Ondřeje Fikara.

V současné době má volejbalový kroužek přibližně 80 hráčiček čtenů a trojici trenérů III. třídy. Hráčky DDM HIPPO Kuřim jsou převážně z Kuřimi, ovšem hrají nebo hrály zde hráčky z Podolí, Brna, Ivanovic, Cebín, Malhostovic, Drásova, Třanova, Kału, Moravských Kňanic, Vevverské Bytčsky, Svinovic, České, Lelekovíc, Lipůvky, Nuzičova, Rozdolevic, Třanova, a Šebrova. Každý tým má dvě tréninkové jednotky trvající 2 hodiny. Venkovní tréninky probíhají na antukovém kurtu ZS Týřkova. Halové tréninky jsou v tělocvičnách ZS Týřkova a ZS Komenského. Týmy kadetek i žákyně hrají své domácí zápasy na ZS Týřkova. Soutěžní turnaje v minivolejbalu bohužel pořádát nemůžeme z důvodu kapacit tělocvičen. Kategorie jsou rozděleny převážně věkově, ale velkou roli zde hraje i herní úroveň každé hráčky či hráče. Proto i mládež hráčky nastupují ve vyšších věkových kategoriích, pokud jsou schopny zvládnout požadované herní dovednosti. Nově přichází hráčky a hráči jsou napádk podle své úrovně zařazeni do příslušné výkonnostní kategorie, přičemž se snažíme, aby co nejdříve zvládli dané herní prvky a mohli se zařadit do své věkové kategorie. Pokud vás tyto informace zaujaly nebo oslovily, rádi bychom přivítali každého, kdo by se chtěl zapojit, jak do hráčské, trenérské či sponzorské činnosti. Hráče a hráčky rádi uvítáme na náboru nebo na klenčínskému tréninku.

Pro každého, kdo by se chtěl zapojit do trenérské činnosti, vyřízení hráčů seženeme. Zkušenosti s volejbalem nebo trenérská opčnírní nepožadujeme. V neposlední řadě bychom chtěli oslovit všechny, kdo by měli zájem ná finančně podpořit formou sponzoringu, obilous nebo hmotného daru. Pokud by měl někdo zájem, je možný kontakt ná klenčínskou e-mail v kategorii O nás, nebo navštívte jakýkoliv ná trénink - příslušné časy a místa jsou v infu o jednotlivých kroužcích. Na vlastní oči se můžete přesvědčit o fungování volejbalového kroužku.



#### Historie kroužku

**Sezóna 2002/2003**

V sezóně 2002/2003 se kroužek volejbalu zúčastnil svého prvního soutěžního ročníku, Krajského přeboru kadetek, ve kterém převážně sbíral zkušenosti ze soutěžních zápasů.

DOZVĚDĚT SE VÍCE >

**Sezóna 2003/2004**

V sezóně 2003/2004 se mimo kadetek zapojil do soutěže i tým žákyně, tvořený hráčkami z loňského náboru. Zúčastnil se soutěže v Krajském přeboru žákyně a hned v první sezóně se umístil ve střední tabulce, což bylo pro všechny velice příjemné.

DOZVĚDĚT SE VÍCE >

**Sezóna 2004/2005**

Do sezóny 2004/2005 jsme nastoupili se třemi týmy: juniorkami, kadetkami a přípravkou hráčiči Krajský přebor v minivolejbalu. Jak se později ukázalo pečtemi! jsme svoje síly.

DOZVĚDĚT SE VÍCE >

**Sezóna 2005/2006**

Před novou sezónou 2005/2006 proběhlo sloučení týmu juniorek a kadetek a sloučený tým nastoupil opět do Krajské soutěže juniorek. Tým přípravy postoupil do věkové kategorie žákyně a nastupoval v tomto roce k zápasu Krajské soutěže žákyně.

DOZVĚDĚT SE VÍCE >

**Sezóna 2006/2007**

V sezóně 2006/2007 nastupovalo DDM HIPPO Kuřim ve třech soutěžích. Tým juniorek nastupoval v krajském přeboru, tým žákyně také v krajském přeboru a tým

DOZVĚDĚT SE VÍCE >

**Sezóna 2007/2008**

V sezóně 2007/2008 nastoupily týmy DDM HIPPO Kuřim ve čtyřech soutěžích. Tým kadetek v Krajském přeboru kadetek, tým žákyně v Krajském přeboru žákyně a čtyř

DOZVĚDĚT SE VÍCE >

**Sezóna 2008/2009**

V sezóně 2008/2009 nastoupily týmy DDM HIPPO Kuřim ve čtyřech soutěžích. Tým kadetek v Krajském přeboru kadetek, tým žákyně v Krajském přeboru žákyně a Pohár

DOZVĚDĚT SE VÍCE >

**Sezóna 2009/2010**

V sezóně 2009/2010 nastoupily týmy DDM HIPPO Kuřim ve čtyřech soutěžích. Tým kadetek v Krajském přeboru kadetek, tým žákyně v Krajském přeboru žákyně a Pohár

DOZVĚDĚT SE VÍCE >

4

Obrázek 16: Kufrový test navigace - dektopová verze



Obrázek 17: Kufrový test navigace - mobilní verze

## 5.6 Implementace aplikace na webový server

Jako vhodný webový hosting byl vybrán poskytovatel endora. Jelikož jedním z požadavků bylo aby webové služby byly poskytovány bezúplatně, byla vybrána doména 3.řádu jecool.net. Odkaz na webovou stránku je [www.volejbalkurim.jecool.net](http://www.volejbalkurim.jecool.net). Soubory je možné nahrávat přes webové rozhraní a nebo přes FTP klienta.

Po úspěšném nahrání všech souborů a po vytvoření databáze je webová aplikace plně funkční. Po dohodě se zástupcem oddílu byly vybrány fotografie pro webovou prezentaci a doladěn celkový vzhled.

## 6 Diskuze

### 6.1 Zhodnocení výsledků, návrh možných zlepšení

Dle zadaných funkčních a nefunkčních požadavků byla vytvořena webová aplikace, která byla následně zpřístupněna na webu. Po testování konkrétními uživateli byly opraveny drobné nedostatky a objasněna funkčnost některých prvků. Aplikace má stále spoustu míst, které by bylo možné rozšířit či pozměnit. Bylo by například možné správci, tedy v tomto případě trenérovi, umožnit aby mohl editovat statický text ve webové prezentaci, nebo při přihlašování či odhlašování z události hráčům umožnit připsat komentář, proč se nemůžou zúčastnit a následně pro tyto odpovědi udělat výstup. Momentálně je tento případ řešen pomocí soukromého chatu a nebo napsáním omluvenky. Dále by bylo možné vytvářet automaticky tréninky dle dne ve kterém obvykle probíhají. Od tohoto bylo upuštěno z důvodu možného přesouvání tréninků a jelikož při vytváření nového tréninku kromě data zadává trenér rovnou i hráče z daného oddílu, kteří se dostavili na trénink či chyběli nenastal zde žádný krok navíc, který by agendu zpomaloval. Je zde počítáno i s případnou chybou na straně zadavatele tréninku, a proto je zde dodatečná možnost zaevidování docházky na trénink pomocí tlačítek plus a minus. Návštěvníci webu se samozřejmě mohou dozvědět kdy a kde probíhají pravidelné tréninky. Tyto informace trenér edituje ve správě týmů a jsou tedy ukládány do databáze. V případě jednorázových změn pak může být vytvořena aktualita na hlavní stránce. Co se týče aktualit i ty by do budoucna mohly být vylepšeny a to například tím, že při vytváření by administrátor mohl zvolit i barvu pozadí aktuality a tím docílit větší důležitosti aktuality.

V případě, že by se oddíl rozhodl na stránkách prezentovat větší množství fotografií či souborů, bylo by nutné upgradovat dosavadní program na placený hosting s větším místem na webu. V tomto smyslu by se dalo uvažovat i o vytvoření rozšíření v podobě nahrávání nejen pdf souborů, ale i například xls či csv souborů obsahujících statistiky ze zápasů. Tato možnost byla v počátcích s oddílem konzultována, avšak nakonec zavrhnuta kvůli momentálnímu nezájmu oddílu tyto statistiky vytvářet. Je zde tedy do budoucna možnost toto rozšíření implementovat.

### 6.2 Srovnání výsledku s obdobnými řešeními

Jak již bylo řečeno v kapitole Přehled literatury v podkapitole Stávající řešení, každé sportovní odvětví je něčím specifické, i když základ zůstává stejný. Co se týče vnitřní administrativy je výsledek této práce porovnatelný například s aplikací Teamstuff avšak je zde zahrnuta i webová prezentace a není ji tedy nutné řešit externě. Tím se docílí především toho výsledku, že návštěvník webu může sledovat aktuální informace například přes grafické znázornění docházky či aktuální seznam členů jednotlivých týmů a správcům týmů tak nevzniká povinnost upgradovat data dvakrát. Pro větší oddíl by však tato aplikace nemusela být dostačující. V takových

případech by byla lepší aplikace Teamstuff, která má větší podporu komunikace mezi členy oddílu.

V aplikaci není přidaná možnost nahrávání videí do webové prezentace, což může být mínus oproti řešení od firmy Sklub, avšak na druhou stranu je zde vyřešeno přihlašování na zápasy, akce a správa financí.



## 7 Závěr

Výsledná aplikace obsahuje veškeré funkční i nefunkční požadavky, které byly stanoveny. Aplikace byla přizpůsobena konkrétnímu volejbalovému týmu. Za účelem prezentace oddílu pomocí webové prezentace byl navržen také webový design pro daný oddíl, který byl následně použit při implementaci samotné aplikace. Podle stanoveného nefunkčního požadavku byla aplikace zpřístupněna na internetu pod adresou <http://volejbalkurim.jecool.net>. Toto umístění je pouze dočasné a po zkušebním provozu bude aplikace přesunuta na web Domova dětí a mládeže v Kuřimi.

V případě zájmu jiného oddílu není problém databáze a obsah webové prezentace včetně obrázků přizpůsobit. Pro tento účel byla vytvořena i samostatná třída, naplňující položky do navigace pro následné snažší přizpůsobení. Aplikace není tak velkého rozsahu aby dostáčovala velkým oddílům zaměřeným na více sportovních odvětví, avšak po menších úpravách by byla aplikovatelná i na jiné sportovní odvětví než pouze na volejbal.

Jestli aplikace přinese i žádoucí ekonomický přínos, například v podobě sponzorů či úspory času trenérů při správě dat, se ukáže až po skončení sezóny 2017/2018, kdy bude aplikace plně využívána.

## 8 Reference

- BÖHMER, M. *Zend Framework: programujeme webové aplikace v PHP*. 1. vyd. Brno: Computer Press, 2010. 416 s.  
ISBN 978-80-251-2965-4.
- ČÁPKA, D. *MVC architektura* [online].[cit. 2017-04-27].  
Dostupné z: <https://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor>.
- GOOGLE *Material Design Lite* [online].[cit. 2017-04-27].  
Dostupné z: <https://getmdl.io>.
- GOOGLE DEVELOPERS *Google Charts* [online].[cit. 2017-04-27].  
Dostupné z: <https://developers.google.com/chart/>.
- KAO *Internetové systémy pro sportovní kluby* [online].[cit. 2017-04-27].  
Dostupné z: <http://kao.cz/sluzby/internetove-systemy/websystem-pro-sportovni-kluby/>.
- KRIVTISOV, O. *Get User Demo Sample from GitHub* [online].[cit. 2017-04-27].  
Dostupné z: <https://github.com/olegkrivtsov/using-zf3-book-samples/blob/master/userdemo/README.md>.
- LONSKÝ, M. *ZendFramework 3* [online].[cit. 2017-04-27].  
Dostupné z: <http://www.zendframework.cz/php/zendframework/2016/06/02/zend-framework-3.0.html>.
- LOPEZ, A. *Lerning PHP 7*. Birmingham: Packt Publishing Ltd., 2016.  
ISBN 978-1-78588-054-4.
- MOLINARO, A. *SQL Kuchařka programátora*. Brno: Computer Press, 2009.  
ISBN 978-80-251-2617-2.
- PITT, CH. *Pro PHP MVC* Apress, 2012. 500 s.  
ISBN 978-1-4302-4165-2.
- SCRIPTINY *TinyBox JavaScript Popup Box* [online].[cit. 2017-04-27].  
Dostupné z: <http://www.scriptiny.com/2009/05/javascript-popup-box/>.
- SEZNAM.CZ, A.S. [online].[cit. 2017-04-27].  
Dostupné z: <https://api.mapy.cz>.
- SKLUB *O projektu*. [online].[cit. 2017-04-27].  
Dostupné z: <http://www.sklub.cz/o-projektu>.
- SPORTVIA.EU *sportvia.eu* [online].[cit. 2017-04-27].  
Dostupné z: <http://sportvia.eu>.

SWEB [online].[cit. 2017-04-27].

Dostupné z: <https://www.sweb.cz/>.

ŠILHAVÝ R., P. ŠILHAVÝ, Z. PROKOPOVÁ, P. POKORNÝ, M. SYSEL, M. MATÝSEK, K. VLČEK A L. SVOBODOVÁ *Vybrané aspekty návrhu webových informačních systémů* Vsetín: Šilhavý s.r.o, 2013. Scientific Press, 2/2013. ISBN 978-80-251-2617-2.

TEAMSTUFF [online].[cit. 2017-04-27].

Dostupné z: <https://teamstuff.com/cs>.

TRÖSTER, F. *ORM frameworky pro PHP 5: Obecný úvod*. [online].[cit. 2017-04-27].

Dostupné z: [www.zdrojak.cz/clanky/orm-frameworky-pro-php5-obecny-uvod](http://www.zdrojak.cz/clanky/orm-frameworky-pro-php5-obecny-uvod).

VARIYA SOFT *Zend Framework* [online].[cit. 2017-05-07].

Dostupné z: <http://www.variyasoft.com/zend-framework-development.html>.

WIKIPEDIA: THE FREE ENCYCLOPEDIA *Webová aplikace*. [online].[cit. 2017-04-27].

Dostupné z: [https://cs.wikipedia.org/wiki/Webová\\_aplikace](https://cs.wikipedia.org/wiki/Webová_aplikace).

ZEND, A ROGUE WAVE COMPANY *About* [online].[cit. 2017-04-27].

Dostupné z: <https://framework.zend.com/about>.

ZEND, A ROGUE WAVE COMPANY *Getting started: A skeleton application* [online].[cit. 2017-04-27].

Dostupné z: <https://docs.zendframework.com/tutorials/getting-started/skeleton-application/>.

## 9 Seznam použitých zkratek

### **ACL (Access Control List)**

Seznam přístupových práv

### **API (Application Programming Interface)**

Aplikační rozhraní

### **CAPTCHA**

Turingův test pro rozeznání uživatele od robota

### **CSM (Content Management System)**

System pro správu obsahu webových stránek

### **CSS (Cascading Style Sheets)**

Kaskádové styly pro úpravu zobrazení elementů na webových stránkách

### **DDM**

Dům dětí a mládeže

### **ERD (Entity Relationship diagram)**

Diagram zobrazující entity a vztahy mezi nimi

### **FTP (File Transfer Protocol)**

Protokol pro přenos souboru mezi serverem a klientem

### **HTML (HyperText Markup Language)**

Značkový jazyk pro tvorbu webových stránek

### **MVC (Model-View-Controller)**

Softwarová architektura rozdělující aplikaci do tří hlavních komponent a to údaje, prezentace a řízení Böhmer(200.70, s.397)

### **ORM (Object-relational mapping)**

Metoda mapování databáze na objekty

### **PHP (Hypertext Preprocessor)**

Programovací jazyk pro tvorbu dynamických webových stránek

### **SQL (Structured Query Language)**

Dotazovací jazyk vyvinutý pro práci s daty v relačních databázích

### **UML (Unified Modeling Language)**

Vizuální jazyk pro tvorbu diagramů

### **URL (Uniform Resource Locator)**

Jednoznačné umístění zdroje v počítačové síti

## 10 Obsah příloženého CD

### **aplikace/**

Zdrojový kód aplikace včetně automatických testů a zkušební databáze.

### **prace/**

Elektronická verze bakalářské práce ve formátu PDF.

### **readme.txt**

Požadavky na aplikaci a návod k jejímu spuštění.

## Seznam obrázků

Obrázek 1: MVC struktura, zdroj: (Variya Soft, ©2009 - 2017)	11
Obrázek 2: Class diagram tabulky "uzivatel"	18
Obrázek 3: UseCase diagram	21
Obrázek 4: Entity relationship diagram	23
Obrázek 5: Diagram webového serveru	26
Obrázek 6: Hlavní stránka - desktopová verze	28
Obrázek 7: Hlavní stránka - mobilní verze	29
Obrázek 8: Podstránka - desktopová verze	30
Obrázek 9: Podstránka - mobilní verze	31
Obrázek 10: Logo oddílu	32
Obrázek 11: Barevné chéma	32
Obrázek 12: Formulář	36
Obrázek 13: Změna přihlašovacích údajů	38
Obrázek 14: Rozcestník správy oddílů	40
Obrázek 15: Prezentace týmu juniorky	40
Obrázek 16: Kufrový test navigace - dektopová verze	44
Obrázek 17: Kufrový test navigace - mobilní verze	45