



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**INFORMAČNÍ SYSTÉM FLORBALOVÉHO TÝMU**

INFORMATION SYSTEM OF A FLOORBALL TEAM

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JAKUB ZMEK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Doc. Ing. JAROSLAV ZENDULKA, CSc.**

BRNO 2019

## Zadání bakalářské práce



22115

Student: **Zmek Jakub**  
Program: Informační technologie  
Název: **Informační systém florbalového týmu**  
**Information System of a Floorball Team**  
Kategorie: Informační systémy

Zadání:

1. Seznamte se s problematikou vývoje webových aplikací, dostupnými prostředími a rámci pro jejich tvorbu.
2. Seznamte se s problematikou extrakce strukturovaných dat z webu a dostupnými prostředky na její podporu.
3. Seznamte se s požadavky kladenými na informační systém florbalového týmu a s veřejně dostupnými informacemi na webu, které se týkají zápasů týmu.
4. Požadavky analyzujte a navrhnete odpovídající informační systém, který bude využívat informace dostupné na webu. Při návrhu použijte vhodné modelovací prostředky jazyka UML.
5. Navržený informační systém implementujte a ověřte jeho funkčnost na vhodném vzorku dat.
6. Zhodnoťte dosažené výsledky a navrhnete případná další vylepšení a rozšíření.

Literatura:

- Welling, L., Thomsonová, L.: PHP a MySQL: rozvoj webových aplikací. Vyd. 1. Praha: SoftPress, 2003, 910 s. ISBN 80-86497-60-7.
- Žára, O.: JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015. ISBN: 978-80-251-4573-9
- What is web scraping - Part 1 - Beginner's guide. Dostupné na <https://www.scrapehero.com/a-beginners-guide-to-web-scraping-part-1-the-basics/>.
- Beginner's guide to Web Scraping with PHP. Dostupné na <http://www.prowebscraper.com/blog/web-scraping-using-php/>.
- Grässle, P., Baumann, H., Baumann, P.: UML 2.0 in Action: A Project Based Tutorial. Packt Publishing. 2005. 229 s. ISBN 1-904811-55-8.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Zendulka Jaroslav, doc. Ing., CSc.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 29. října 2018

## Abstrakt

Tato bakalářská práce se zabývá studiem webových technologií a následným vytvořením informačního systému pro florbalový tým FbC ReActive Aligators E kvůli přehledným statistikám a správě docházky na události. Informační systém je naprogramován formou webové aplikace s podporou frameworku Nette. Systém umožňuje přihlašování na události (tréninky, zápasy) a zobrazování dané docházky. Systém automaticky extrahuje data (zápasy, tabulku soutěže, statistiky hráčů) ze stránek Českého florbalu a ukládá je do databáze. Stažené statistiky jsou veřejně přístupné v informačním systému.

## Abstract

This bachelor thesis is focused on study of web technologies and the subsequent creation of an information system for floorball team FbC Reative Aligators E for well-aranged statistics and event attendance management. The information system is programmed as a web application with Nette framework support. The system allows logging to events (trainings, matches) and displaying off attendance. The system automatically extracts data (matches, competition table, player statistics) from Czech floorbal websites and stores them in the database. The downloaded statistics are the publicly available in the information system.

## Klíčová slova

PHP, Nette, informační systém, web, florbal

## Keywords

PHP, Nette, information system, web, floorball

## Citace

ZMEK, Jakub. *INFORMAČNÍ SYSTÉM FLORBALOVÉHO TÝMU*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Ing. Jaroslav Zendulka, CSc.

# INFORMAČNÍ SYSTÉM FLORBALOVÉHO TÝMU

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. Ing. Jaroslava Zendulky, CSc. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jakub Zmek  
15. května 2019

## Poděkování

Rád bych poděkoval panu Doc. Ing. Jaroslavu Zendulkovi, CSc. za cenné rady při konzultacích a vedení mé bakalářské práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Vývoj webových aplikací</b>	<b>4</b>
2.1	Webová aplikace . . . . .	4
2.2	Návrhové vzory MVC a MVP . . . . .	6
2.3	HTML . . . . .	8
2.4	CSS . . . . .	8
2.5	MySQL . . . . .	8
2.6	Programovací jazyky a frameworky . . . . .	9
2.7	Web scraping . . . . .	11
<b>3</b>	<b>Použité technologie</b>	<b>13</b>
3.1	Klient (front-end) . . . . .	13
3.2	Server (back-end) . . . . .	13
3.3	Hosting . . . . .	14
3.4	Databáze . . . . .	14
<b>4</b>	<b>Analýza požadavků</b>	<b>15</b>
4.1	Neformální specifikace . . . . .	15
4.2	Diagram případů užití . . . . .	15
4.3	Popis případů použití . . . . .	16
4.4	ER Diagram . . . . .	19
<b>5</b>	<b>Návrh a implementace informačního systému</b>	<b>23</b>
5.1	Návrh relační databáze . . . . .	23
5.2	Návrh uživatelského rozhraní . . . . .	25
5.3	Návrh podproblémů . . . . .	26
5.4	Implementace . . . . .	27
5.4.1	Databáze . . . . .	27
5.4.2	Extrakce dat . . . . .	27
5.4.3	Statistiky . . . . .	31
5.4.4	Uživatelé . . . . .	32
5.4.5	Aktuality . . . . .	33
5.4.6	Docházka . . . . .	34
5.4.7	Správa . . . . .	35
<b>6</b>	<b>Testování</b>	<b>37</b>
6.1	Testování programátorem . . . . .	37

6.2 Testování uživatelů . . . . .	39
<b>7 Závěr</b>	<b>40</b>
<b>Literatura</b>	<b>42</b>
<b>A DVD</b>	<b>44</b>
<b>B Návod k instalaci</b>	<b>45</b>
<b>C Uživatelská příručka</b>	<b>46</b>
C.1 Přihlášení . . . . .	46
C.2 Aktuality . . . . .	47
C.3 Statistiky . . . . .	49
C.4 Docházka . . . . .	49
C.5 Správa . . . . .	51
C.6 Odhlášení . . . . .	55

# Kapitola 1

## Úvod

V dnešní době se v praxi setkáváme s informačními systémy, které spravují údaje v nejrůznějších oblastech. S informačními systémy se setkáváme téměř všude – např. zaměstnání, škola, styk s úřady, internetové obchody apod.

Velice mě zajímá problematika vývoje informačních systémů. Řešení této problematiky jsem využil při tvorbě informačního systému pro florbalový tým. Florbalové prostřední je mi blízké. Proto jsem se rozhodl vytvořit informační systém pro tým s názvem „FbC ReActive Aligators E“, ve kterém hraji.

Abych mohl naimplementovat webovou aplikaci, musel jsem si nastudovat webové technologie. Konkrétně jsem si nastudoval jazyky HTML, CSS, Javascript a PHP. Pro PHP jsem si nastudoval framework Nette, ve kterém je aplikace napsána. Dále jsem si musel nastudovat, jak funguje extrakce dat z jiné webové stránky v PHP.

Společně se spoluhráči jsme zformulovali požadavky na informační systém. Tím je myšleno, jaké funkce by měl umožňovat. Funkcí, které by se mohly florbalovému týmu hodit, je samozřejmě mnoho.

Kromě základních očekávaných funkcí, jako je vedení statistik týmů v lize, našich hráčů a zápasů, které se samy pomocí skriptu aktualizují v databázi, byla zavedena správa aktualit. Jejich prostřednictvím může trenér informovat hráče a fanoušky o nejnovějších událostech v týmu.

Další užitečnou funkcí je správa docházky. Hráči se mohou přihlašovat na následujících pět událostí (tréninky, zápasy) seřazených podle data. U události je vidět celkový počet hráčů, kteří jsou přihlášení na událost, včetně jejich jmen.

Automatickou funkcí, která umožňuje spravovat databázi pomocí aplikace, je možnost přidávat, upravovat a mazat data. Každý hráč si sám může spravovat informace o své osobě.

Aby byl informační systém použitelný i na další sezóny, byly zavedeny další dvě funkce – úprava extrakčních odkazů a ukončení sezóny. V případě postoupení či sestoupení do jiné ligy se změní odkazy, na kterých budou dostupná data pro extrakci. Ukončení sezóny vymaže statistiky z databáze. Aby se však vedly i statistiky z předchozích sezón, je možnost je uložit ve formě textu do databáze a zobrazit je uživateli.

Obsah práce je rozdělen do sedmi kapitol dle postupného vývoje aplikace. Po úvodní kapitole následuje kapitola popisující podstatu webových aplikací. Ve třetí kapitole jsou popsány podrobněji webové technologie, které jsou použity v tomto informačním systému. V následující kapitole popisují, jaké požadavky na informační systém jsem získal. V páté kapitole je popsán návrh a implementace informačního systému. Po implementaci následuje kapitola zabývající se testováním. Poslední kapitolou je závěr, který také obsahuje návrhy na vylepšení informačního systému.

## Kapitola 2

# Vývoj webových aplikací

Tato kapitola popisuje základní webové technologie používané při vývoji webových aplikací, jak fungují a jaké jsou jejich principy. Použité technologie jsou pak důkladněji popsány v následující kapitole.

### 2.1 Webová aplikace

Webová aplikace je aplikace, která je uživatelům přístupná prostřednictvím internetu (nebo intranetu). Jsou velice oblíbené, jelikož používají internetový prohlížeč jako klienta. Jelikož má nyní již každý operační systém internetový prohlížeč, je webová aplikace je přístupná odkudkoliv z jakéhokoliv zařízení připojeného k internetu.

Webová aplikace je tudíž dostupná bez složitých instalací každému. Navíc je velice jednoduchá aktualizace programu. Zatímco v normální desktopové aplikaci bychom museli aktualizace instalovat na každém počítači zvlášť, webovou aplikaci stačí pouze nahrát na server a při dalším načtení klientem je aktualizace programu hotová.

Při vytváření této podkapitoly jsem vycházel ze zdroje o webových aplikacích. [7]

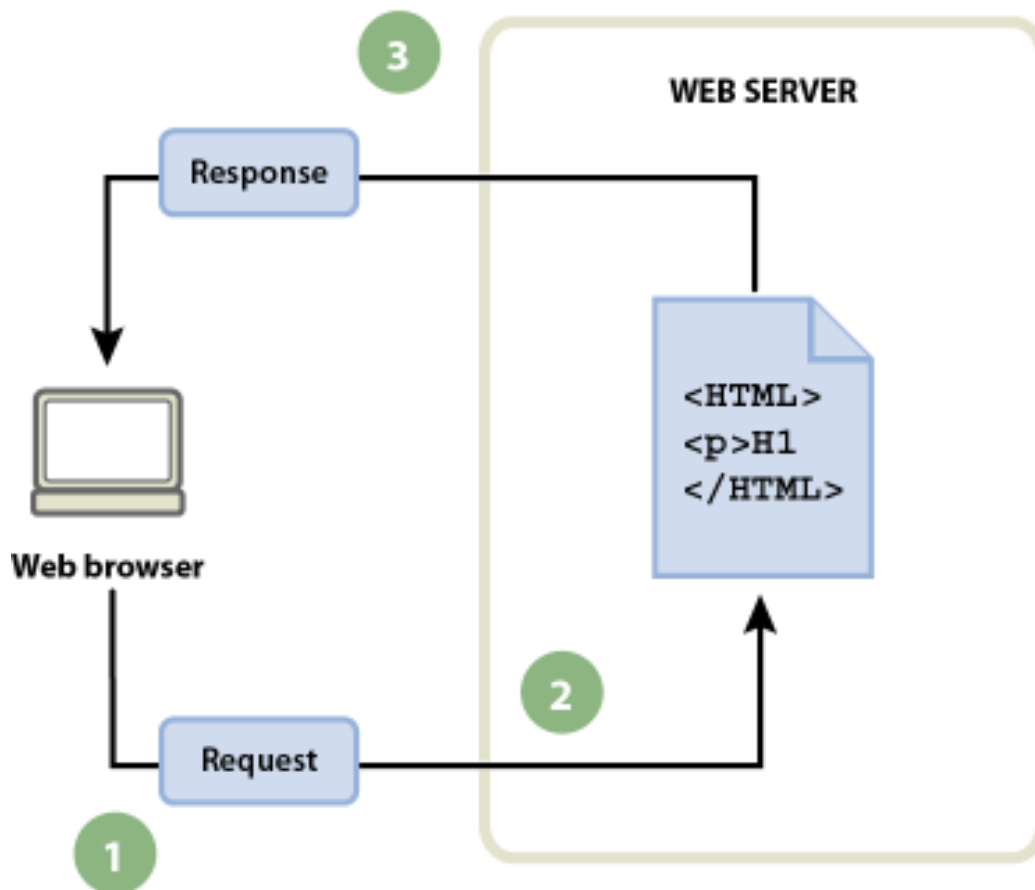
#### Jak funguje webová aplikace

Webová aplikace je kolekce statických a dynamických webových stránek. Statická webová stránka se nemění, když o ni návštěvník webové stránky požádá. Typicky se jedná pouze o stránku s HTML obsahem. Naproti tomu dynamickou webovou stránku server modifikuje před odesláním prohlížeči, který o ni požádal. Modifikací se myslí například vykonání PHP kódu či vyhodnocení dotazu z databáze. Obecně to jsou příkazy, které mohou být v různých situacích vyhodnoceny jinak.

#### Zpracování statistických webových stránek

Statické webové stránky jsou uloženy na serveru v souborovém úložišti. Typicky to jsou HTML stránky provázané odkazy. V případě žádosti klienta server odešle požadovaný soubor. Prohlížeč klienta pak tento kód zobrazí uživateli. Nákres průběhu zpracování dynamických stránek je vidět na obrázku 2.1.





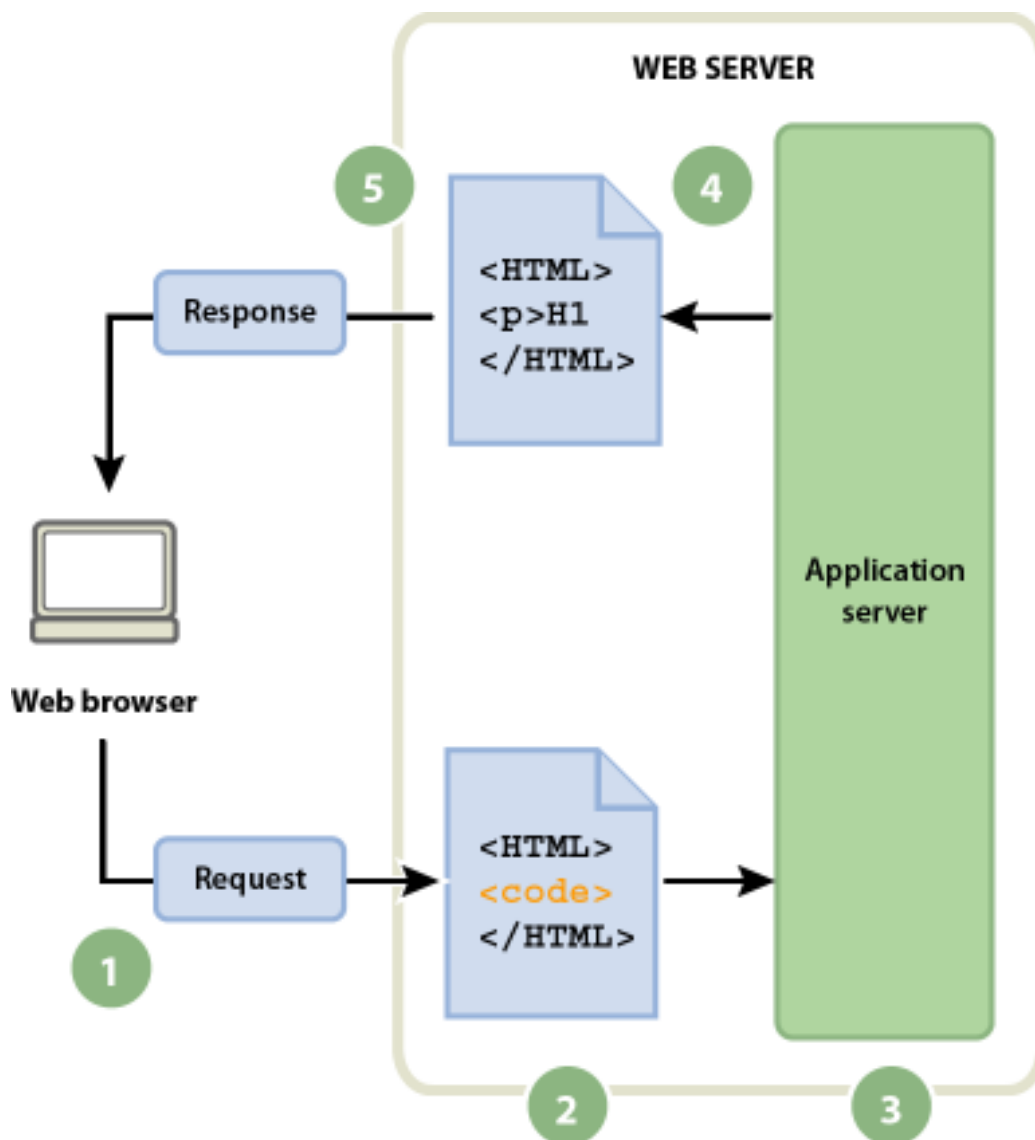
Obrázek 2.1: Zpracování statických webových stránek [7]

1. Webový prohlížeč požádá o statickou stránku.
2. Webový server vyhledá stránku.
3. Webový server pošle stránku prohlížeči, který o ni požádal.

### Zpracování dynamických webových stránek

Když ale webový server přijme požadavek na dynamickou stránku, reaguje jinak. Předá stránku zvláštnímu softwaru, který odpovídá za dokončení stránky. Tento speciální software se nazývá aplikační server.

Aplikační server si přečte kód na stránce, dokončí stránku podle instrukcí v kódu a pak odstraní kód ze stránky. Výsledkem je statická stránka, kterou aplikační server předá zpátky webovému serveru, který ji pak pošle prohlížeči, který o ni požádal. Všechno, co prohlížeč dostane, když stránka dorazí, je čisté HTML. Nákres průběhu zpracování dynamických stránek je vidět na obrázku 2.2.



Obrázek 2.2: Zpracování dynamických webových stránek [7]

1. Webový prohlížeč požádá o dynamickou stránku.
2. Webový server stránku vyhledá a předá ji aplikačnímu serveru
3. Aplikační server najde na stránce případně instrukce a dokončí ji.
4. Aplikační server předá dokončenou stránku zpátky webovému serveru.
5. Webový server pošle dokončenou stránku prohlížeči, který o ni požádal.

## 2.2 Návrhové vzory MVC a MVP

Většina programů také využívá nějaký návrhový vzor. Mezi základní modely patří MVC A MVP. Při zpracování této podkapitoly byl využit zdroj. [8]

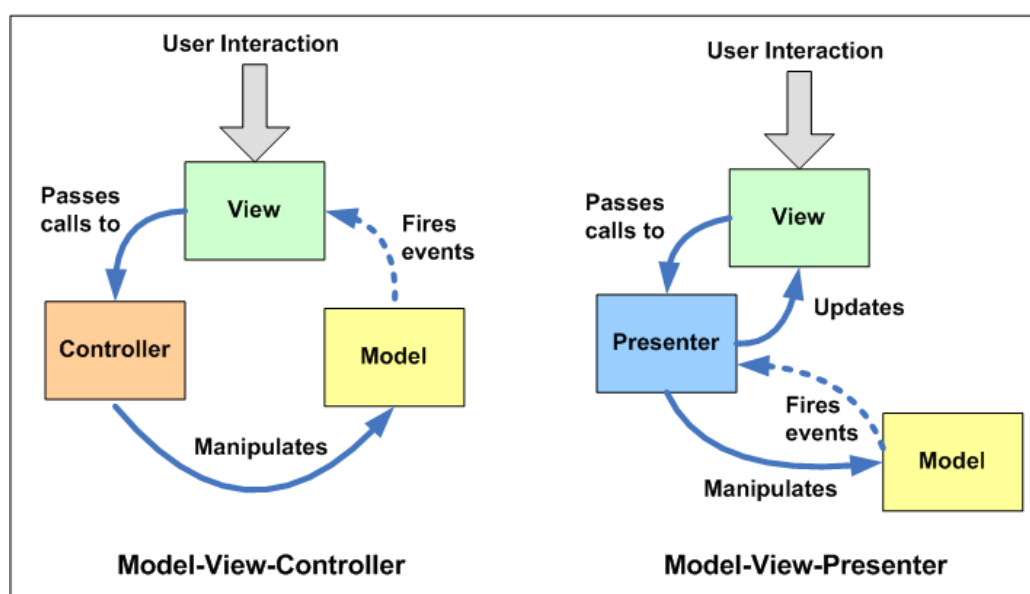
Konkrétně u návrhového vzoru MVC se jedná o rozdělení aplikace na následující části.

- Model – Data a jejich logika.
- View – Zobrazení dat z modelu, požaduje data z controlleru.
- Controller – Manipuluje s modelem a výsledky posílá do view.

Návrhový vzor MVP se dělí na následující části.

- Model – Data a jejich logika.
- View – Zobrazení dat z presenteru. Deleguje na presenter uživatelské akce.
- Presenter – Manipuluje s modelem a naplňuje view daty.

Jednoduchý diagram MVC a MVP lze vidět na obrázku 2.3.



Obrázek 2.3: Návrhové vzory MVC a MVP [10]

Jelikož je v aplikaci využíván model MVC, bude zde popsán podrobněji. Pro popis MVC byl použit zdroj. [5]

## MVC

Model-View-Controller je softwarová architektura, která odděluje aplikační logiku (model), obsluhu (controller) a zobrazení (view).

Model je datový základ aplikace. Typicky se jedná o data uložená v databázi. Jakákoliv akce uživatele (např. přihlášení, zobrazení dat, vložení položky do košíku) představuje určitou manipulaci s těmito daty. Model si spravuje vnitřní stav a controlleru poskytuje pevně dané rozhraní. Voláním těchto funkcí lze měnit či zobrazovat jeho obsah. Model se o existenci controlleru ani view nezajímá.

Controller je spojovací vrstva mezi modelem a view. Jedná se o řadič, který zpracovává požadavky uživatele. Na požadavky uživatele pak reaguje voláním modelu. Controller následně odpověď od modelu předává view.

View je vrstva, která se stará o interakci s uživatelem. Zobrazuje uživateli výsledky požadavků, stará se o vzhled aplikace. Obvykle používá šablonovací systém a ví, jak se která komponenta má zobrazit.

## 2.3 HTML

HTML (Hypertext Markup Language) je nejpoužívanější jazyk při zavádění webových stránek. Pomocí HTML se definuje základní rozložení stránky. Umožňuje tedy např. vymezit prostor pro menu či postranní panel. Jednotlivé stránky jsou propojeny hypertextovými odkazy, jejichž pomocí lze přepínat různé obsahy.

Základním prvkem HTML souboru jsou párové značky, které vymezují platnost obsahu. Pomocí toho lze například určit, který text má být podtržený pomocí párové značky `<u>` označující začátek a `</u>` označující konec. Párových značek je mnoho a umožňují definovat např. tělo souboru, tabulku, odstavec textu atd.

Pomocí HTML lze tedy vytvořit základní web s možností přepínání stránek, rozložením, obrázky, textem apod. V praxi je toto ale zcela nedostačující. K rozšíření webové aplikace slouží následující jazyky.

Při zpracování kapitoly o HTML jsem vycházel ze zdroje. [7]

## 2.4 CSS

CSS (Kaskádové styly) je jazyk pro definování vzhledu webové stránky. Pomocí CSS lze nadefinovat např. barvu, rozměry, zarovnání vpravo nebo vlevo apod. I když toto vše lze nadefinovat již v HTML souboru, když je do tagu přidán atribut `style`, definovat vzhled v CSS souborech je mnohem efektivnější.

Takzvané selektory umožňují vybrat skupinu elementů HTML, na které se má daná skupina pravidel uplatnit. To znamená, že lze např. nastavit velikost, barvu a zarovnání pro nadpis tag `<h1>`. Pravidlo pak bude aplikováno na všechny nadpisy `h1` v souborech, které mají připojený daný soubor CSS.

Pokud programátor nechce mít všechny html tagy v jednotném stylu, jsou potřeba méně obecné selektory. Pro tento případ je využíváno rozdělení programu do různých částí (tag `<div>`). Každý takový `div` má své jméno, které se může opakovat, a proto lze pro různé části programů definovat stejná pravidla. Pro tyto části pak pod jejich jménem definujeme pravidla, která se mají aplikovat opět např. na nadpis `h1`, ale v ostatních částech již takový nadpis může být jiný.

CSS jsem nastudoval ze zdroje. [9]

## 2.5 MySQL

MySQL sice není jazyk přímo určený pro web, ale úzce s ním souvisí. Zvláště v případě, kdy je stránkou informační systém, nedá se mu vyhnout. MySQL je způsob, jak uložit data.

MySQL je jazyk pro databázový systém. Jelikož jeho příkazy jsou téměř snadno pochopitelné v angličtině, je dost intuitivní a stačí se naučit pouze syntaxe. MySQL je tedy jazyk dostatečně jednoduchý, přesto však zvládá zpracovat všechny požadavky na databázi.

Pomocí MySQL lze spravovat úložiště dat. Slouží k definování tabulek (jak data vypadají a jaké atributy mají obsahovat) včetně jejich vazeb a následnému přidávání, upravování a mazání dat. Kromě toho umožňuje pravděpodobně nejpoužívanější funkci – `SELECT`.

Pomocí funkce výběru lze přehledně zobrazovat (typicky v tabulkách) všechny informace, které chceme. Výběr lze také nějakým způsobem omezit, např. zobrazit pouze objednávky pana Nováka. Výběr také umožňuje spojování tabulek, seskupování a řazení výsledků atd.

MySQL běží na serveru a existuje dobrá podpora pro spojení této databáze s PHP. Pomocí PHP lze zadávat dotazy do databáze, které jsou vyhodnoceny a navráceny do určené proměnné.

Při zpracování této kapitoly jsem vycházel ze zdroje. [12]

## 2.6 Programovací jazyky a frameworky

Pro kvalitní web nestačí pouhé definování rozložení a vzhledu. Pro zkvalitnění webu slouží programovací jazyky. Jazyky, které pomohou definovat, jak má stránka fungovat. Programovací jazyky tvoří základ dynamické stránky, jelikož v různých situacích mohou ukazovat rozdílné výsledky podle vyhodnocení kódu. Mezi takové jazyky patří např. PHP, Javascript, Python, C. [2]

### PHP

PHP (Hypertext Preprocessor) je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací. PHP skript běží na serveru, kde se vykoná a návštěvníkovi stránek posílá pouze výsledek.

Pomocí PHP tak lze měnit obsah stránek – např. ukázat obsah pro přihlášené uživatele a ostatní informovat o nemožnosti přístupu, nebo pro děti ukázat reklamu na hračky, naopak pro dospělé reklamy na cokoliv jiného. Z toho vyplývá, že PHP může určit, kdo co bude moci zobrazovat pomocí různých pravomocí účtů. Například v internetovém obchodě se zákazníkovi pouze otevírá možnost nakupování, zaměstnanec už může objednávky zpracovávat.

Ačkoliv je v PHP napsána velká většina stránek, samotné PHP se dnes bohužel považuje za nedostačující. Proto bylo vytvořeno mnoho frameworků, které usnadňují práci v PHP a zlepšují bezpečnost aplikace. Mezi nejoblíbenější frameworky patří:

- Laravel,
- Symfony,
- CodeIgnoter,
- Nette (v České republice).

Čerpal jsem z knihy o PHP a ze stránky o nejpoužívanějších frameworkcích pro PHP. [11][13]

### Nette

Nette je framework pro programovací jazyk PHP. Zatímco v PHP lze vše napsat do jednoho souboru, Nette striktně rozděluje kód podle architektury MVC.

Samotný View řeší pouze vzhled stránky, popřípadě nějaké základní konstrukce PHP jako je zobrazení stránky pro oprávněné uživatele apod. View kódy lze nalézt ve složkové struktuře /app/presenters/templates. Zde je hlavní šablona layout, kterou dědí všechny další šablony. V různých složkách jsou pak různé kódy pro zobrazovací účely.

Veškeré databázové operace řeší vrstva Controller. Objekt, který má funkce ovladače na starosti, se v Nette nazývá Presenter. Pro každou stránku je vytvořen objekt ovladače, ve kterém lze definovat funkce pro vybrání dat z databáze, vykreslení formuláře, zpracování formuláře atd. Tím lze velmi zpřehlednit kód ve vrstvě zobrazení a oddělit databázové operace od vzhledu stránky. V této vrstvě jsou velké výhody, které se týkají zjednodušení psaní kódu. Namísto psaní složitých SQL dotazů lze pomocí selektorů postupně vybrat data ve formě, jakou programátor požaduje. Vytvoření formuláře je také mnohem jednodušší, stačí vytvořit potomka třídy Form a k ní pomocí již implementovaných metod přidávat prvky formuláře, které se samy přeloží do HTML. Kódy controllerů se nachází ve složce app/presenters.

Příklad vytvoření jednoduchého formuláře z mé stránky, konkrétně funkce Konec sezóny:

```
$form = new Form;
$form->addCheckbox('ulozit', 'Chci statistiky ukoncene sezony ulozit do databaze ve
forme textu.')->setDefaultValue(true);
$form->addSubmit('send', 'Smazat');
$form->onSuccess[] = [$this, 'KonecsezonyFormSucceeded'];
```

Příklad výběru dat z tabulky TYM seřazen podle atributu BODY:

```
$tymy = $this->database->table('TYM')->order('BODY DESC');
```

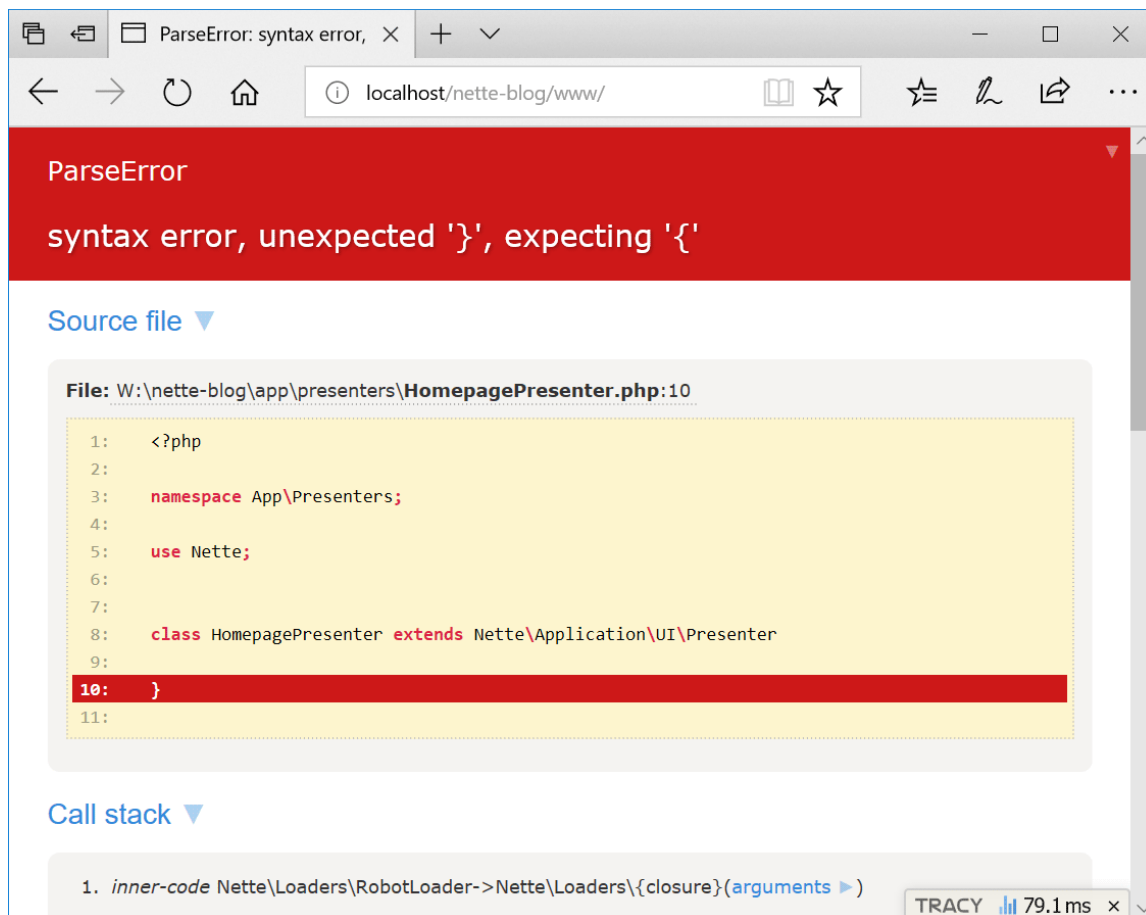
Model představuje databázový systém. S modelem spolupracuje controller. Vybírá data z databáze (typicky pro zobrazení) nebo případně ukládá data získaná uživatelem (typicky formulářem).

Další velikou výhodou je ladicí nástroj Tracy, který při problému se spuštěním programu srozumitelněji vypíše problém v kódu včetně informací, jako jsou hodnoty proměnných, HTTP request, HTTP response, `$_GET`, `$_POST`, ale i `$_COOKIE`. Pro zajímavost je zde také lišta, která ukazuje např. čas stažení stránky a vykonané SQL dotazy včetně doby trvání. Velikou výhodou je, že zobrazení tohoto ladicího nástroje se dá nastavit pouze pro určité IP adresy, takže návštěvníkům stránky se v případě problému zobrazí error 500 a chyby v kódu neuvidí. Některé chyby dokonce dokáže Tracy po odsouhlasení opravit sama. Jedná se však o jednodušší chyby, např. překlep v názvu proměnné v jednom znaku.

Syntaxi Nette Frameworku jsem nastudoval v článku "Píšeme první aplikaci" o několika kapitolách na české stránce o tomto frameworku. Stránka poskytuje značné množství informací o Nette Frameworku včetně fóra<sup>1</sup>, kde jsou ostatní programátoři ochotni pomoci s řešením netradičních problémů. Z článku „Píšeme první aplikaci“ také byl získán obrázek 2.4, konkrétně z kapitoly „Začínáme“. [6]

---

<sup>1</sup>Dostupné z: <https://forum.nette.org/cs/>



Obrázek 2.4: Ladicí nástroj Tracy [6]

## JavaScript

JavaScript je další dynamický jazyk, který na web přidá základní interakci s uživatelem. Javascript umí vytvořit efekty, jako je např. animace, změnit barvu tlačítka při najetí na něj nebo zobrazit dialogové okno s dotazem.

Na rozdíl od PHP Javascript je ze serveru posílán uživateli bez vyhodnocení formou kódu. Až dle reakcí uživatele je kód spuštěn a vrátí nějaký výsledek.

Nejčastější použití Javascriptu je pro již zmíněná dialogová okna, kterým se např. uživatelé stránka zeptá, zda chce opravdu smazat danou věc z databáze anebo jestli chce opravdu opustit formulář bez uložení hodnot.

Při zpracování této kapitoly jsem čerpal ze zdroje o programovacích jazycích. [3]

## 2.7 Web scraping

Web scraping je technika, která slouží k získávání dat z jiných webových stránek. Jedná se o data, která chceme využít pro své účely. Tato data jsou sice veřejně přístupná, ale ne ve formátu pro stažení.

Téměř všechna data, která lze vidět na webové stránce, jsou k dispozici v HTML formátu. Tento HTML formát webu lze zobrazit pomocí pravého tlačítka myši a výběrem možnosti „Zobrazit zdrojový kód stránky“.

Tato data je samozřejmě možné ručně přepisovat a využívat ke svým účelům. Toto je však velmi složité a je zde příliš velké riziko chyby. Jediným praktickým řešením je automatizovat tento proces a data z webu získávat pohodlně spuštěním daného kódu.

Poslední dobou je velká poptávka po datech pro průzkum trhu, cenovou inteligenci nebo analýzu konkurentů. To je přesně proces, který umožňuje web scraping.

Při zpracování této kapitoly byl použit zdroj, který tuto problematiku popisuje.[1]

## Techniky

Mezi první techniku patří **manuální** získávání dat. Patří to sem, i když je žádoucí se této technice vyhnout. Takto data získává velká většina průměrných uživatelů internetu. Těmto uživatelům se může zalíbit článek, a tak si ho jednoduše zkopírují do textového souboru a mají získaná data u sebe. Manuální získávání dat je ruční, pomalé a méně efektivní. Tento druh získávání dat funguje jen pro malé části dat, které zahrnují jednoduchý text.

Technika získávání dat, která byla využita v této bakalářské práci, se nazývá **DOM parsing**. Pomocí webových prohlížečů lze přistupovat k dynamickému obsahu, které vytvořily skripty na straně klienta. DOM parsing analyzuje webové stránky ve formě stromu DOM. HTML nebo XML kód je převeden na DOM strom. DOM formuluje strukturu dokumentu a způsob, jak dokument zpřístupnit. Z tohoto důvodu lze z DOM stromu jednoduše získat např. tabulky či jiný prvek. Této vlastnosti je v této bakalářské práci silně využíváno při extrakci dat.

Poslední technikou je získávání dat pomocí **regulárních výrazů**. Pomocí regulárního výrazu je definován vzor, který se hledá v textovém řetězci stránky. Typicky se používá při vyhledávání pomocí kláves CTRL + F. Pro složitější získávání dat je však nepoužitelný.



## Kapitola 3

# Použité technologie

Pro tvorbu jakéhokoliv programu je potřeba si stanovit, jaké technologie a jazyky použít. Je potřeba zvážit všechny výhody a nevýhody těchto technologií. Jelikož jsem byl rozhodnut, že chci informační systém vytvořit formou webové aplikace, byla volba technologií celkem jasná. Tato kapitola se zabývá popisem použitých technologií v informačním systému pro florbalový tým.

### 3.1 Klient (front-end)

Pro vytváření vzhledu stránek jsem zvolil technologie HTML a CSS. Tyto technologie jsou stále používány na velké většině webů a pro definování vzhledu naprosto stačí.

### 3.2 Server (back-end)

Pro programování serveru jsem zvolil jedno z nejpoužívanějších řešení – PHP. PHP poskytuje dostačující možnosti pro naprogramování konkurenceschopné stránky. Podpora pro tento jazyk je obrovská. Ať už se jedná o knihy, které seznamují čtenáře se základní syntaxí jazyka, až po diskuzní fóra<sup>1</sup>. Na diskuzních fórech se řeší téměř každý problém, který může nastat. [2]

PHP je velmi jednoduché propojit s HTML kódem pouhým označením tagy „<?php“ a „?>“. Tyto tagy označují začátek a konec PHP kódu. PHP zároveň umožňuje jednoduchou manipulaci s databází včetně přihlášení. Kvůli těmto výhodám jsem tedy zvolil PHP.

Bohužel žádný jazyk nemá jen samé výhody. To se týká i jazyku PHP. Psaní kódu je zdlouhavé, nepřehledné a téměř vůbec neusnadňuje práci programátorovi. Samotné PHP navíc nedbá na bezpečnost, čímž je myšlena velká zranitelnost informačních systémů využitím útoku SQL Injection. Z těchto důvodů jsem se rozhodl použít framework Nette, který usnadňuje zápis konstrukcí v PHP<sup>2</sup> a sám eliminuje útoky SQL Injection. [4]

#### Nette

Pro řešení nedostatků PHP jsem si zvolil framework Nette. Vybral jsem si ho kvůli velké podpoře v České republice. Existuje již zmíněná česká webová stránka se skvělou podporou k tomuto frameworku. [4]

---

<sup>1</sup>z velké většiny na [www.stackoverflow.com](http://www.stackoverflow.com)

<sup>2</sup>hlavně konstrukci ovladače a ovládání databáze

### 3.3 Hosting

Pro běh aplikace měl být původně využit školní server Eva, který umožňuje nahrávání a zobrazování webových stránek. S touto funkcionalitou serveru mám bohaté zkušenosti z předmětu Informační systémy. Po rozhodnutí, že bude použit Framework Nette, byla na serveru vytvořena základní struktura projektu v Nette. Bohužel byl zjištěn problém s Nette na Evě, a tak byl zvolen jiný hosting.

Pro hosting byla zvolena doména `php5.cz`<sup>3</sup>. Tato doména je v základním nastavení zdarma a umožňuje uložit až 50MB souborů. Spolu s vytvořením stránky je možné si vytvořit i databázi s uživatelským prostředím phpMyAdmin. Doména tak splňovala naprosto vše, co bylo pro aplikaci požadováno. Proto byla zvolena jako doména pro tuto bakalářskou práci. Informační systém byl umístěn na URL <http://www.aligators-e.php5.cz/www/>.

### 3.4 Databáze

U databáze byl výběr také celkem jednoznačný. Na hostingu, na kterém běží informační systém, je možnost si zdarma nechat vytvořit databázi s příjemným uživatelským prostředím phpMyAdmin. Této možnosti jsem využil. Tuto databázi je možné ovládat pomocí jazyku MySQL 5.7. Jelikož jsem s uživatelským prostředím phpMyAdmin pracoval již v předmětu Informační systémy, nebylo pro mě žádnou novinkou. Z tohoto předmětu také znám jazyk MySQL.

Prostředí phpMyAdmin umožňuje přehlednou správu databáze. Toto prostředí také umožňuje přehledné zobrazení obsahu databáze včetně úprav, přidávání a mazání záznamů. Je zde umožněno spustit skript s příkazy a vykonat je. Toho je možné využít pro vytvoření databáze a naplnění vzorkem hodnot.

---

<sup>3</sup>Stránky hostingu: <https://www.php5.cz/>

## Kapitola 4

# Analýza požadavků

V této kapitole popíšu požadavky na informační systém florbalového týmu, které jsme společně se spoluhráči připravili.

Všechny požadavky a připomínky spoluhráčů a trenéra byly průběžně zpracovávány a při každé ucelené změně jsme s týmem probrali, co by mohlo být jinak.

Kvůli snadné dostupnosti informačního systému musí být implementován formou webové stránky.

### 4.1 Neformální specifikace

Informační systém musí ukládat statistiky týmu, které bude automaticky stahovat. Tyto statistiky budou zobrazovány uživateli.

V informačním systému je potřeba uchovávat osoby z několika důvodů. Hlavním důvodem je možnost se přihlašovat do informačního systému. To bude umožněno přihlašovacím jménem a heslem. Dalším neméně důležitým důvodem je vedení určitých informací o osobě, které trenérovi zjednoduší komunikaci s osobami či administraci týmu. Mezi takové informace patří rodné číslo, telefonní číslo, adresa, e-mail. Další zajímavé informace jsou číslo dresu a post hráče.

Je třeba vést organizaci docházky na jednotlivé zápasy a tréninky. Tím by se florbalový tým měl vyvarovat nedostatečnému počtu hráčů na tréninku či zápase a včas doplnit sestavu z ostatních mužských či juniorských týmů.

Informační systém bude umožňovat informování fanoušků formou textových aktualit.

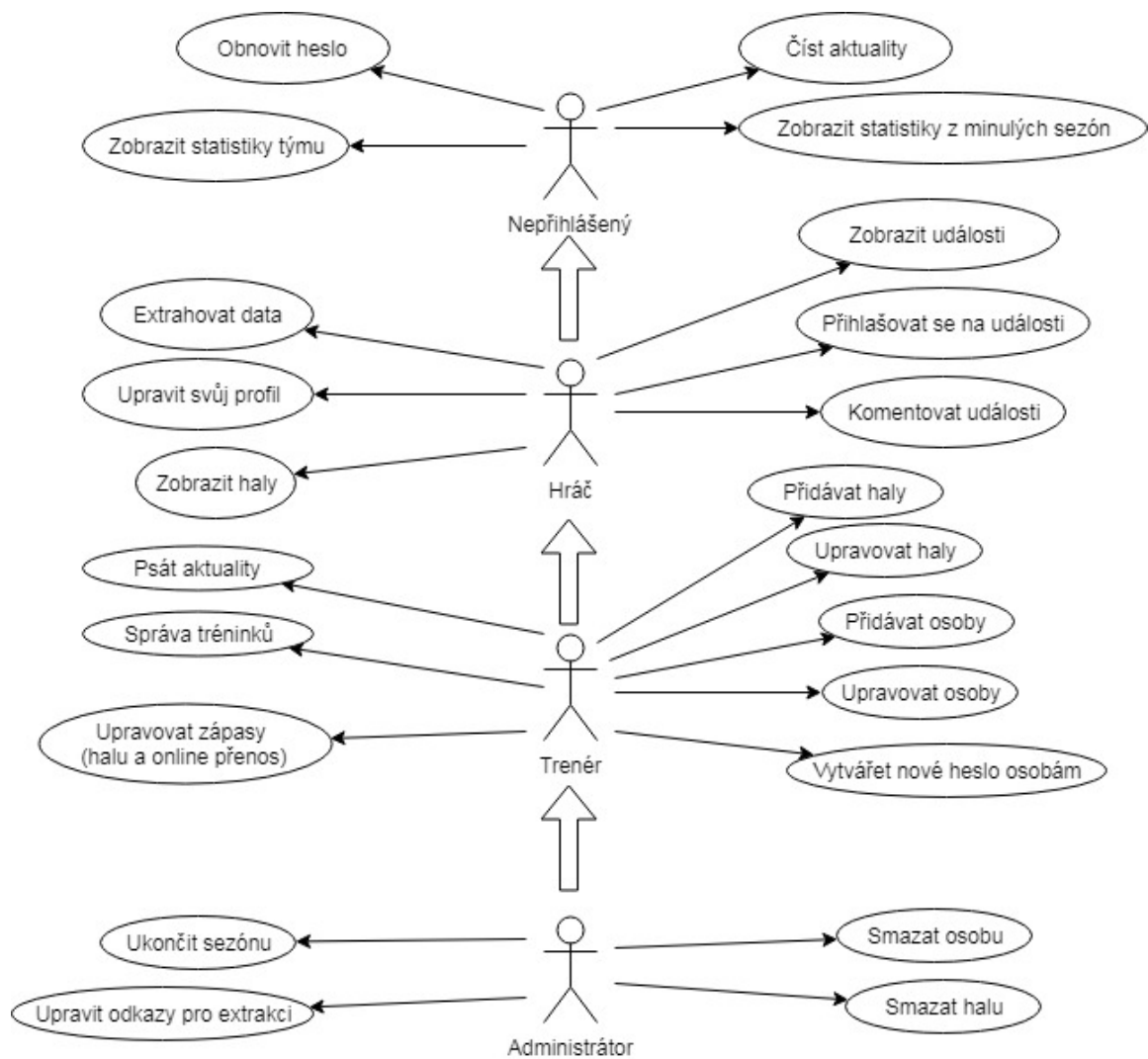
Ne každá osoba by v každém informačním systému měla mít možnost dělat vše. K tomu slouží zavedení role uživatele. Pro informační systém florbalového týmu jsou dostačující role:

- hráč,
- trenér,
- administrátor.

Jednotlivé pravomoci skupin uživatelů budou ukázány v následující části pomocí diagramu případů užití.

### 4.2 Diagram případů užití

Na obrázku 4.1 lze vidět navržený diagram případů užití pro informační systém pro florbalový tým na základě požadavků spoluhráčů a trenéra.



Obrázek 4.1: Digram případů užití

### 4.3 Popis případů použití

Zde budou podrobněji popsány všechny případy použití.

#### Číst a psát aktuality

Kromě informování prostřednictvím statistik by měla být možnost informovat o důležitých tématech formou textu. Tento problém řeší aktuality.

Aktuality slouží k informování fanoušků, hráčů, či celého týmu o důležitých tématech. Základní motivací byla možnost popisování průběhu zápasů, turnajů nebo informování o novinkách z týmu. Prostřednictvím aktualit je však možnost sdílet téměř vše, např. doplnění týmu o nové posily, či např. i narození dítěte jednoho z členů týmu.

## Obnovit heslo

Uživateli informačního systému se může stát, že zapomene heslo. Pro tento případ musí být možnost obnovy. Nové heslo bude zasíláno hráči na e-mail.

## Extrahovat data

Základem informačního systému florbalového týmu by mělo být určité vedení statistik. Kvůli automatizaci a velké pravděpodobnosti chyb při manuálním vyplňování formulářů, budou automaticky extrahovány ze stránek českého florbalu.

## Zobrazovat statistiky

Každý uživatel bude moct zobrazit statistiky z aktuální sezóny. Různé statistiky budou na různých stránkách a budou zobrazeny pomocí tabulek.

Každý hráč či fanoušek chce vědět, jak si jeho **tým** vede v porovnání s ostatními týmy v lize. Proto budou uchovávána data o jednotlivých týmech ligy. Konkrétně se jedná o název týmu, zápasy, výhry, remízy, prohry, body, branky vstřelené a obdržené, úspěšnost přesilovek a oslabení, trestné minuty. Týmy se budou řadit podle bodů, z čehož plyne i aktuální pořadí týmů v lize.

Mezi základní statistiky také určitě patří výsledky **zápasů**. Kromě samotného výsledku budou o zápase uchovávány údaje: datum, čas, kolo, domácí tým, hostující tým, hala a odkaz na online přenos.

Datum a čas bude sloužit hlavně při přihlašování na události. Podle data a času by měl vědět daný hráč, zda se v danou dobu může zápasu zúčastnit.

Nebudou se uchovávat údaje o statistikách hráčů a brankářů v konkrétních zápasech. Náměnou tohoto je již dříve zmíněná generovaná aktualita, která vypíše změny před extrakcí a po extrakci. Jak již bylo zmíněno, tyto aktuality budou pravděpodobně (podle času spuštění) obsahovat informace o celém hracím dni. Pokud by někdo chtěl statistiky pouze za určitý zápas, je plnou náhradou odkaz na online přenos, kde jsou uvedené branky i asistence v zápasech jednotlivých hráčů. Stejně tak se bude dát online přenos využít pro sledování aktuálního stavu v průběhu zápasu.

Uchovávat se budou pouze zápasy našeho týmu, stejně tak to dělají i informační systémy různých jiných sportovních klubů.

U každého **hráče** budou vedeny aktuální statistiky. U každého hráče bude veden počet zápasů, góly, góly v přesilovce, góly v oslabení, asistence, kanadské body a trestné minuty. Hráči při zobrazení statistik budou seřazeni podle celkového počtu kanadských bodů. Ke každému hráči bude pro zajímavost dopočítán průměrný počet kanadských bodů za zápas.

U každého **brankáře** budou taktéž vedeny aktuální statistiky, které se od statistik hráčů liší. Každý brankář je taktéž veden na stránkách českého florbalu jako hráč, tudíž bude veden i v tabulce statistik hráčů. Konkrétně statistiky brankáře budou obsahovat počet zápasů (na soupisce), odchytné zápasy (ve kterých brankář zasáhl přímo do hry), odehraný čas, výhry, obdržené góly, čistá konta, remízy, prohry, asistence, trestné minuty.

Ke statistikám bude dopočítán počet obdržených branek na zápas. I když často uvádí tato statistika jako obdržené branky za zápas, ve skutečnosti tomu tak není. Jedná se o počet obdržených branek na čas utkání. S délkou utkání 39 minut je správný vzorec 4.1.

$$\frac{\text{Obdržené branky}}{\text{Čas zápasu}} = \frac{\text{Obdržené branky}}{\text{Odehraný čas} / 39.00} = \left( \frac{\text{Obdržené branky}}{\text{Zápas}} \right) \quad (4.1)$$

Vzorec v závorce se bude s ostatními rovnat v případě, že brankář odehraje vždy celé utkání a nebude vystřídán. Nicméně na toto nelze spoléhat, a proto se to bude vždy počítat podle druhého vzorce.

**Statistiky z minulých sezón** bude možno vidět na stránce, kde budou seřazeny od nejnovějších statistik po nejstarší. V případě, že by se do předchozích sezón dostala nešikovným zacházením nějaká nechtěná data (např. data pouze z poloviny sezóny), bude možnost je smazat. Takovou pravomoc bude mít pouze administrátor. Tyto statistiky budou uloženy ve formě HTML textu, aby zde bylo možno definovat tabulky. Statistiky budou lépe vypadat.

## Správa hal

Každý trénink a zápas se odehrává v nějaké hale. Pro zjednodušení práce budou vedeny haly v informačním systému. Ke každé hale kromě názvu bude uvedena také adresa pro snadnější nalezení pomocí GPS či mapy.

Hráč bude moci zobrazovat přehled hal. Trenér bude moci přidávat a upravovat haly. Mazat haly bude moci pouze administrátor.

## Události

Nutným požadavkem je správa docházky. Přihlášení uživatelé budou mít možnost zobrazit pět následujících událostí. Ke každé události bude vytvořena anketa s možnostmi „Přijdu“, „Přijdu chytat“ a „Nepřijdu“. Každý se bude moci vyjádřit a jeho odpověď bude uložena. Pod touto anketou bude zobrazen celkový počet hráčů přihlášených k jednotlivým možnostem včetně jmen hráčů.

V případě, že by někomu vyjádření pomocí ankety nestačilo, bude zde možnost přidávat komentáře. Komentáře budou sloužit např. pro domlouvání dopravy, upřesnění omluvenky apod. Komentáře budou seřazeny od nejnovějšího po nejstarší.

Stejná forma správy docházky bude dostupná jak pro tréninky, tak pro zápasy. Zápasy budou vkládány automaticky pomocí extrakce. Tréninky bude moci vytvářet trenér či administrátor. U tréninků bude uvedeno datum, čas a hala. Bude možnost vytvořit jak jednorázový trénink, tak několik tréninků po týdnů až do zvoleného data uživatelem.

## Správa tréninků a zápasů

Plná správa tréninků bude umožněna trenérovi i administrátorovi. Bude možnost přidat více tréninků zároveň po týdnů v předem určeném rozmezí dat.

Jelikož zápasy jsou automaticky aktualizované, není třeba je upravovat. To se netýká haly a odkazu na online přenos. Hala a odkaz na online přenos jsou bohužel hůře dohledatelné a někdy jsou tyto informace i neznámé (například pro zápas, který se odehrává až za 3 měsíce). Tyto informace bude muset ručně doplnit trenér či administrátor.

## Správa osob a vytváření nového hesla

Trenér i administrátor budou moci spravovat osoby. Budou moci i vytvořit nové heslo osobě, aby jí vytvořili přístup (případně se osobě ztratí heslo a nepovede se jí ho automaticky obnovit).

## Ukončit sezónu

Administrátor bude mít možnost vymazat data statistik pomocí funkce ukončení sezóny. Tyto smazané statistiky bude moct uložit do statistik z minulých sezón.

## Upravit odkazy pro extrakci

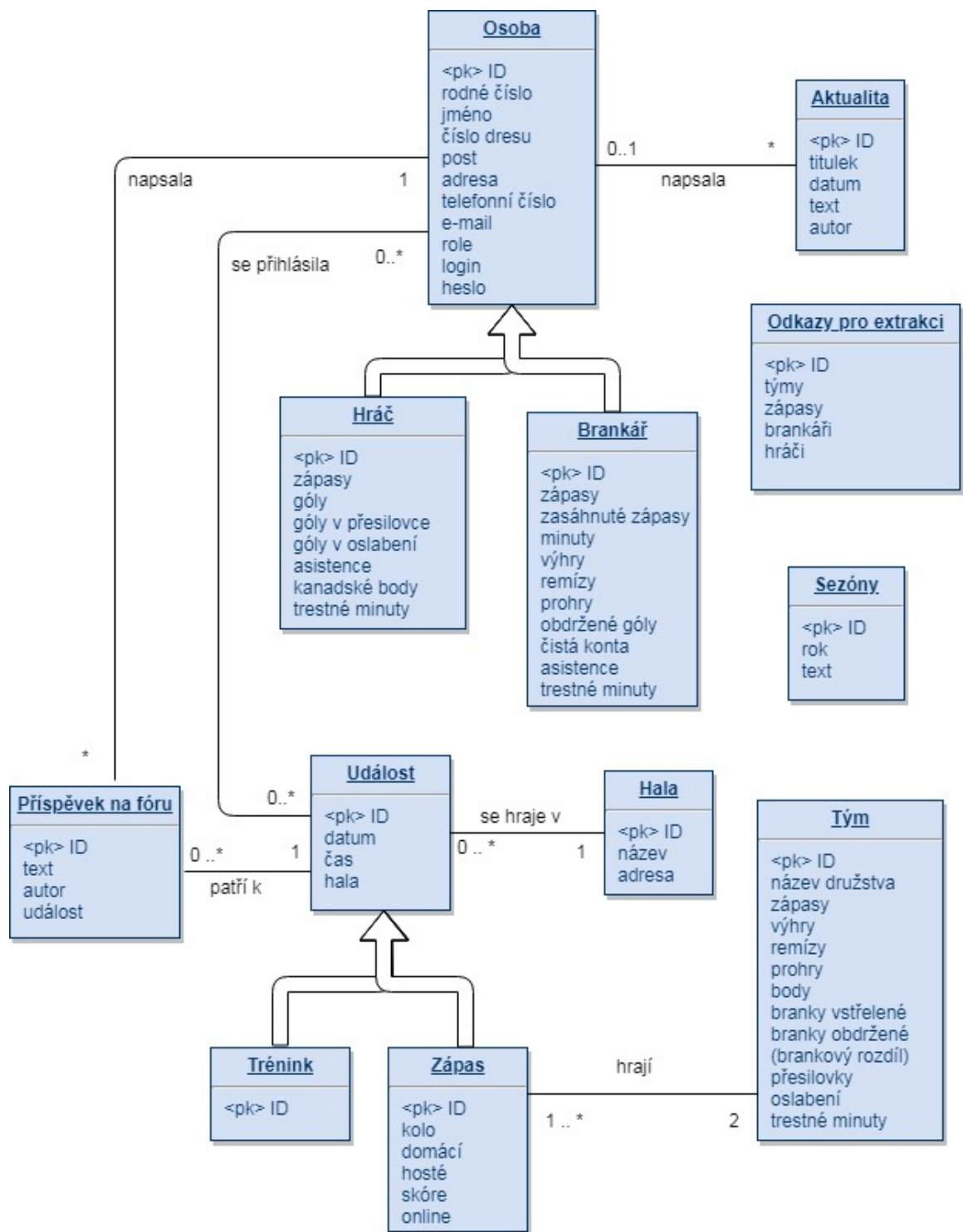
Ačkoliv může být manipulace s extrakčními odkazy nebezpečná a může vést ke stažení irelevantních dat, musí být možnost, jak tyto odkazy upravit bez zásahu v kódu. Na druhou stranu toto uložení umožňuje určitou automatizaci správy informačního systému, jelikož se extrakční odkazy mohou v různých sezónách lišit – konkrétně v případě postupu či sestupu do jiné ligy.

## 4.4 ER Diagram

Na základě požadavků byl vytvořen model požadavků formou ER diagramu. ER diagram byl vytvořen pomocí online editoru<sup>1</sup>. Výsledný ER Diagram lze vidět na obrázku 4.2.

---

<sup>1</sup>[www.draw.io](http://www.draw.io) – online software pro vytváření diagramů zdarma



Obrázek 4.2: ER diagram



## Osoba

Entitní množina osoba označuje osoby, které buď za tým pouze hrají, nebo většinou osoby mající také přístup do informačního systému a aktivně se přihlašující na události. U každé osoby je tak uchovááno:

- rodné číslo, jméno pro identifikaci,
- adresa, telefonní číslo, e-mail pro jednodušší komunikaci trenéra s danou osobou,
- číslo dresu a post sloužící pouze jako doplněk atributů „florbalové“ osoby,
- role, login a heslo pro přihlašování a ověřování přístupových práv.

Jako primární klíč bylo zvoleno umělé číselné ID. Jako primární klíč by se dal použít i atribut rodné číslo, číslo dresu nebo login. Kvůli možným problémům se zveřejňováním osobních údajů a pro větší abstrakci byl zaveden primární klíč ID.

## Hráč

Pomocí vztahu specializace lze zjistit, zda daná osoba za tým hraje i ligové zápasy. Atributy hráče jsou pomocí cizího klíče navázány na osobu s určitým jménem, jíž statistiky patří.

U hráče jsou pak uchovávány všechny informace, které jsem schopen ze stránek českého florbalu automaticky zjistit, tj. počet zápasů, gólů (celkem), gólů v přesilovce a oslabení, asistencí, kanadských bodů a trestných minut.

## Brankář

Obdobou entitní množiny hráč je entitní množina brankář. Atributy brankáře jsou na entitu osoba navázány cizím klíčem v případě, že k dané osobě s daným jménem existují statistiky brankáře v informačním systému českého florbalu.

U brankáře jsou samozřejmě statistiky odlišné, konkrétně počet zápasů (na soupisce), zápasů (do kterých zasáhnul), minut, výher, remíz, proher, obdržných gólů, čistých kont, asistencí a trestných minut.

## Aktualita

Entitní množina aktualita slouží k ukládání aktualit. Jako primární klíč je vedeno ID. Dále obsahuje atributy titulek, datum, text a autor. Autor je cizím klíčem do tabulky osoba.

## Hala

Entitní množina hala umožňuje ukládat haly, ve kterých se odehrávají zápasy či tréninky. U hal je evidováno pouze ID, název a adresa.

## Událost

Entitní množina událost je stejně jako osoba entitou, která zobecňuje více druhů entit. Obecně se jedná o akci pořádanou týmem, která se koná v určité datum, určitý čas a v určité hale. Primárním klíčem je umělé ID.

## **Trénink**

Entitní množina událost se dělí na dvě entitní množiny pomocí vztahu specializace. Jednou z nich je trénink. O tréninku se uchovává pouze ID, které je cizím klíčem do tabulky událost. Tím je daná událost označena jako trénink.

## **Zápas**

Pokud je daná událost zápas, je na ni navázána entitní množina zápas pomocí cizího klíče ID. O zápase je uchováváno kolo, domácí tým, hostující tým, skóre a odkaz na online přenos.

## **Příspěvek na fóru**

Aby mohla osoba psát komentáře k jednotlivým událostem, byla zavedena entitní množina příspěvek na fóru pro ukládání těchto komentářů. Primárním klíčem je ID. Dalším atributem je text daného komentáře.

## **Tým**

V entitní množině tým jsou uchovávány statistiky všech týmů z ligy florbalového týmu kvůli porovnávání a určení pořadí v soutěži. O každém takovém týmu je uchováván jeho název, počet zápasů, remíz, proher, bodů, branek vstřelených a obdržených, trestných minut a úspěšnost přesilovek a oslabení. Primárním klíčem je ID.

## **Sezóna**

Pro ukládání statistik z minulé sezóny byla zavedena entitní množina sezóna. Jedná se o nejjednodušší možné uchování statistik z předcházejících sezón. U sezón je uložen pouze rok (respektive dva roky, ve kterých se sezóna odehrává) a všechny statistiky.

## **Odkazy pro extrakci**

Poslední neméně důležitou entitní množinou jsou odkazy pro extrakci. Tato entitní množina obsahuje pouze jediný záznam se čtyřmi odkazy, které využívá extrakce pro stahování dat. Konkrétně se jedná o odkazy pro stahování statistik týmů, zápasů, brankářů a hráčů.

## Kapitola 5

# Návrh a implementace informačního systému

V předchozí kapitole byly popsány všechny požadavky na informační systém. Z daných požadavků je třeba sestavit návrh informačního systému, který tyto požadavky bude splňovat.

### 5.1 Návrh relační databáze

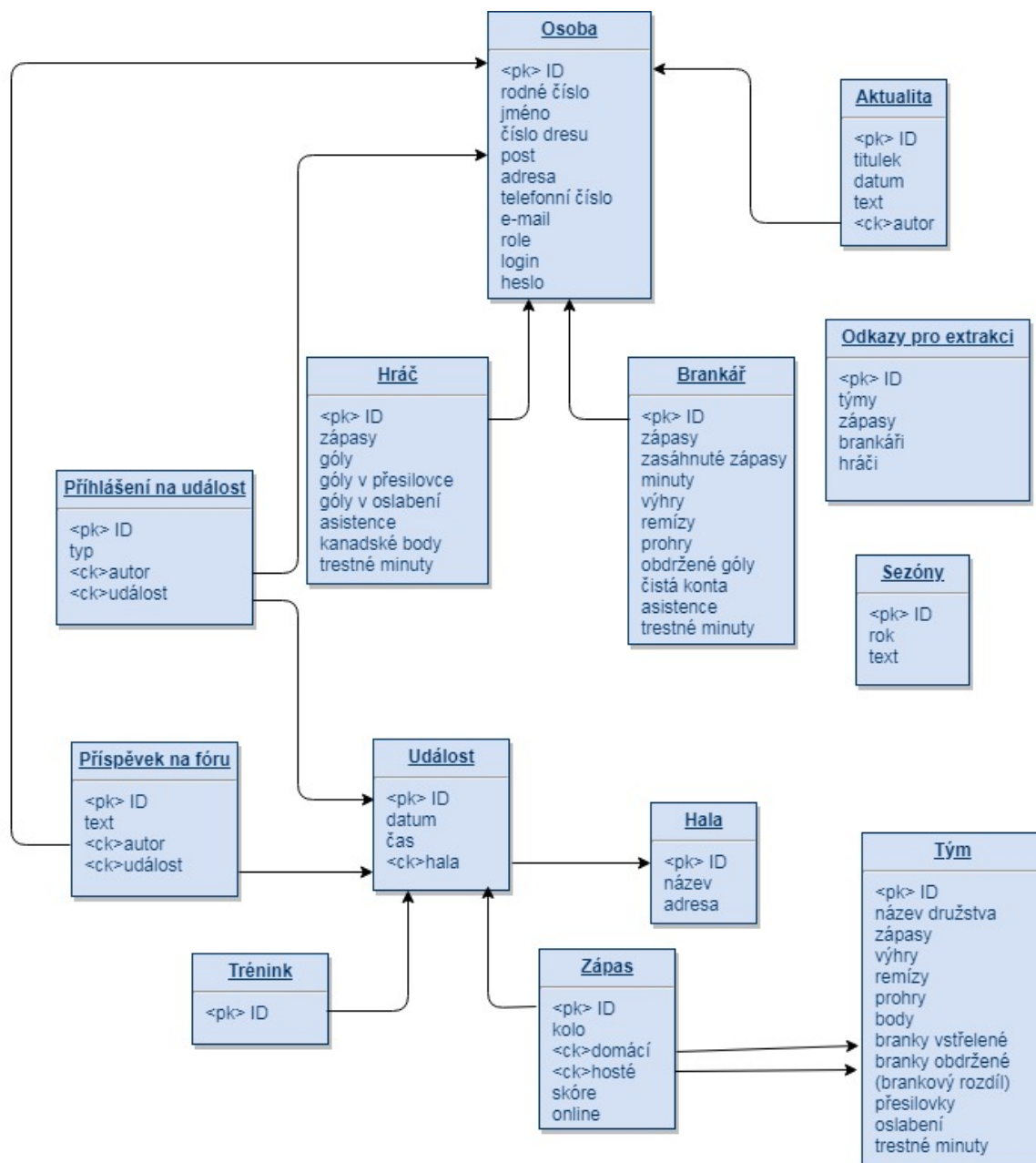
Při transformaci ER diagramu na schéma relační databáze byly vazby typu 1:N převedeny na cizí klíč, který spojuje dané dvě tabulky. Vazba typu M:N představující přihlášení osoby na událost byla převedena pomocí vazební tabulky. Nakonec se ukázalo, že tato vazební tabulka bude muset ještě obsahovat typ přihlášení – konkrétně text s možnostmi „Přijdu“, „Přijdu chytat“ a „Nepřijdu“. Vztahy generalizace specializace byly spojeny pomocí primárního klíče podřadných tabulek. Tyto primární klíče zároveň plní funkci cizího klíče do nadřazené tabulky.

Šipky na diagramu zobrazují cizí klíče. Jejich počátek je na řádku cizího klíče v dané tabulce. Konec šipky ukazuje na tabulku, na jejíž primární klíč je cizí klíč z původní tabulky navázán.

Schéma relační databáze lze vidět na obrázku 5.1. Schéma relační databáze bylo vytvořeno pomocí online editoru.<sup>1</sup>

---

<sup>1</sup>[www.draw.io](http://www.draw.io) – online software pro vytváření diagramů zdarma



Obrázek 5.1: Schéma relační databáze

## 5.2 Návrh uživatelského rozhraní

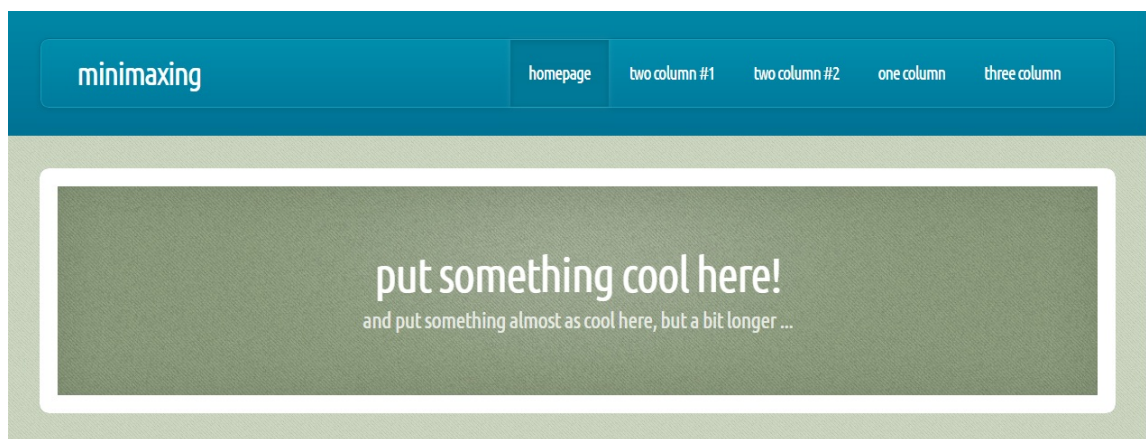
Je potřeba také navrhnout uživatelské rozhraní pro interakci mezi uživatelem a informačním systémem. Cílem bylo vytvořit jednoduchý přehledný vzhled včetně menu, kterým lze přepínat jednotlivé stránky.

Jelikož patřím mezi programátory, kteří by pravděpodobně nedokázali vytvořit profesionálně vypadající vzhled stránky, stáhnul jsem si šablonu<sup>2</sup> pro základní vzhled. Z této šablony jsem využil hlavně vzhled menu a tlačítek, jejichž barvu jsem změnil do klubové zelené barvy. Příložený CSS soubor pak má samozřejmě vliv také na další prvky, jako jsou například nadpisy. Dále jsem použil vzhled banneru a zápatí stránky. Tímto se podařilo vytvořit dobře vypadající jednotný vzhled. Jednotný vzhled jsem musel ručně dodat tabulkám, ve kterých vypisují data z databáze.

Web by měl být responzivní, tj. přizpůsobivý displeji zařízení. O tuto vlastnost web bohužel přišel přidáním tabulek. Jelikož šířka tabulky (hlavně když jde o statistiky) převyšuje výšku, na svislém displeji např. mobilního telefonu web nevypadá použitelně, i když rozměry tabulek na webu uvádím relativně v procentech.

Dalším problémem na mobilních zařízeních je úplné zmizení menu. V obyčejném PHP je menu funkční. Bohužel ten samý kód v Nette již menu nezobrazí a nepodařilo se zjistit proč.

Část původního vzhledu lze vidět na obrázku 5.2.



Obrázek 5.2: Vzhled využití šablony

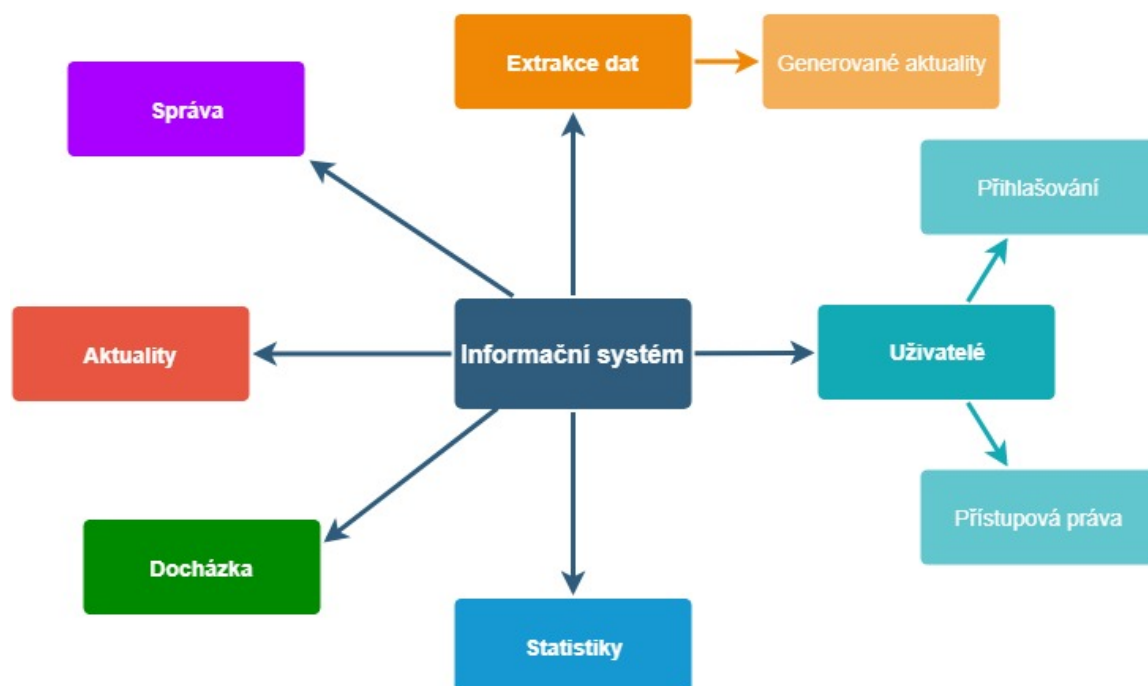
<sup>2</sup>Dostupné z: <https://html5up.net/minimaxing>

### 5.3 Návrh podproblémů

Je žádoucí rozdělit informační systém na podproblémy. Tyto podproblémy se následně dají řešit odděleně a nevzniká zmatek.

Podproblémy byly navrženy podle různých skupin funkcí informačního systému. Každá taková skupina funkcí obsahuje více záležitostí. Např. podproblém aktualit obsahuje čtení aktuality a možnost napsání aktuality. Tyto dílčí záležitosti již byly znázorněny pomocí diagramu případů užití.

Diagram podproblémů je znázorněn na obrázku 5.3. Při vytváření tohoto diagramu byl využit online editor<sup>3</sup>.



Obrázek 5.3: Podproblémy

<sup>3</sup>[www.draw.io](http://www.draw.io) – online software pro vytváření diagramů zdarma

## 5.4 Implementace

Na základě stanovených požadavků, zvolení implementačních technologií a návrhu byl vytvořen informační systém pro florbalový tým. V této kapitole je popsána stručně implementace jednotlivých částí informačního systému.

### 5.4.1 Databáze

Pro fungování informačního systému je třeba vytvořit databázi, do které budou ukládána data. Pro základní inicializaci databáze byl vytvořen skript, který databázi připraví pro prvotní používání.

Pro každou tabulku ze schématu relační databáze byla vytvořena tabulka v databázi. To samozřejmě obnáší i definici atributů včetně typu (datum, integer, varchar atd.). Pro některé tabulky bylo potřeba vytvořit i cizí klíč, kterým se určuje, který atribut právě definované tabulky je navázán na jaký atribut jiné tabulky.

Pro základní fungování a testování byla databáze také naplněna vzorkovými daty. Toto vzorkové naplnění se týká tabulek osob, aktualit, hal a extrakčních odkazů. Tabulky hráč, brankář, tým a zápas (ve spojení s událostí) budou samy naplněny pomocí extrakce dat. Pomocí akcí v uživatelském rozhraní půjde naplnit ostatní tabulky.

### 5.4.2 Extrakce dat

Pro automatizaci naplnění databáze a také okamžitý přístup k aktualizovaným statistikám bylo zvoleno, že bude naprogramován skript, který sám stáhne data z webové stránky Českého florbalu a naplní databázi informačního systému.

Pro extrakci byla použita knihovna PHP Simple HTML DOM Parser<sup>4</sup>. Tato knihovna slouží pro stažení stránky pomocí DOM. Funkce, která toto provádí, vrací HTML kód ve formě řetězce.

V rámci studování extrakce dat jsem si extrakci vyzkoušel v čistém PHP. Tato studijní část probíhala před implementací, a proto je výsledný kód napsaný v čistém PHP.

Ačkoliv to má jisté nevýhody, v podstatě to ničemu nevadí. Jak již bylo zmíněno, Nette samo ošetřuje pokusy o SQL Injection. Toto ošetřování se uplatní zejména v URL a formulářích, ale zdaleka to není potřeba při extrakci dat, kde se stahují předem jasná data ze stránky Českého florbalu. Stahovaná data pouze obsahují statistiky, to znamená čísla a řetězce. Aby nastal útok, musel by nějaký příkaz být přímo v buňce stahované tabulky. Jelikož se statistiky na stránkách Českého florbalu automaticky dopočítávají z elektronického zápisu o utkání, nic takového nehrozí. Jediná hrozba je ruční naplnění SQL příkazem, což je ale neočekávané a prestižní stránka Českého florbalu si toto nemůže dovolit.

Napsání skriptu v PHP bez propojení se zbytkem aplikace způsobuje ještě jeden problém. Ve skriptu není dostupný přihlášený uživatel. V tom případě nelze testovat oprávnění uživatele. Odkaz na stažení je však dostupný pouze přihlášeným uživatelům. V případě, že by neoprávněný uživatel zadal ručně URL skriptu, nemůže tím nic pokazit. Buď data zůstanou stejná, nebo se stáhnou nová, což by stejně někdo musel udělat.

Před začátkem extrakce bylo potřeba určit správné pořadí, ve kterém se budou statistiky extrahovat. Z logických souvislostí vyplynulo pořadí týmy, zápasy, brankáři a nakonec hráči.

Týmy jsou extrahovány jako první hlavně s ohledem na automatické tvoření záznamu aktuality a tvoření textu aktuality. Při extrakci týmů je možnost zjistit rozdíl mezi pů-

---

<sup>4</sup>Dostupné z: <https://sourceforge.net/projects/simplehtmldom/>

vodním a novým počtem zápasů týmu FbC ReActive Aligators E. Pokud je tento rozdíl větší než nula, startuje tvoření aktuality. Když něco startuje tvoření aktuality, měla by tím začínat extrakce dat.

Zápasy jsou extrahovány jako druhé hlavně z hlediska aktuality. Když se u týmů zjistí, že nastala změna zápasů a je generována aktualita, je určitě důležitější výsledek zápasů než statistiky hráčů. Zápasy musí být každopádně extrahovány po týmech, jelikož obsahují dva cizí klíče do tabulky týmů (domácí a hostující tým).

Statistiky brankářů jsou extrahovány před hráči z jednoho prostého důvodu. Pokud je nalezena při extrakci brankářů nová osoba, je dost pravděpodobné, že její post bude brankář. V tomto případě bude nově vytvořené osobě do postu automaticky vloženo, že je brankář („G“). Pokud je osoba jen brankářem z nouze (např. při zápase se zraní všichni brankáři), může se poté post ručně změnit ve správu osob. Z hlediska tvoření aktuality je jedno, zda je před hráči či ne. Někoho zajímají více brankáři a někoho hráči.

Extrakci završují statistiky hráčů. U těchto statistik není jediný důvod, proč by měly být extrahovány před jinou ze zmíněných statistik.

## Extrakce týmů

Před začátkem extrakce je z databáze vybrán záznam, který obsahuje extrakční soubory. Je kontrolováno, zda je odkaz funkční a jestli vůbec zadaná stránka existuje. Pomocí zmíněné knihovny je stažen kód extrahované stránky. Pokud se extrahování podaří, měla by být jednoduchá cesta, jak z extrahovaného HTML kódu vybrat požadovanou tabulku. A to sice vybráním pomocí názvu *divu*. Český florbal bohužel u chtěné tabulky nemá žádný název, a tak je třeba se dostat k tabulce ručně.

Jelikož je potřeba vybrat tabulku, jako první je využita funkce *find* s parametrem *table*. Tímto je dosaženo toho, že pomocí cyklu *foreach* lze procházet všechny tabulky. Pomocí proměnné je vybrána druhá tabulka (se statistikami týmů) a s ní se dále pracuje. Tabulku je třeba samozřejmě dále rozdělit na řádky, které jsou opět procházeny cyklem *foreach*. Každý řádek je rozdělen na buňky a jeho hodnoty jsou vloženy do pole *tym*. Od třetího a dalšího řádku jsou zpracovávány buňky tabulky. První dva řádky obsahují nechtěné informace (nějaký nepodstatný skrytý řádek a hlavička tabulky). Od třetího řádku výše probíhá zpracování pole a zpracování do databáze.

Probíhá kontrola, zda v tabulce je již tým s extrahovaným názvem. Pokud takový záznam není nalezen, je tým i s jeho statistikami vložen do tabulky. Pokud takový tým již existuje, pouze se aktualizují jeho statistiky.

Zároveň je kontrolováno, zda došlo ke změně počtu zápasů u týmu FbC ReActive Aligators E. Pokud taková změna nastane, znamená to, že byly odehrány nějaké zápasy od poslední extrakce. V takovém případě začíná generování aktuality. Je připraven titulek aktuality s počtem bodů, které tým získal. Takový titulek může vypadat například takto: „Aligátoři získali 3 body!“ Ostatní statistiky z této tabulky nemají smysl, buď budou logicky odvozeny čtenářem, nebo jsou neobjektivní. Tím je myšleno, že čtenář si dokáže odvodit z výsledků zápasů, kolik výher, remíz a proher tým uhrál. Také si dokáže odvodit vstřelené a obdržené góly. Neobjektivní informací je pak myšleno například rozdíl úspěšnosti přesilovek a oslabení (před zápasem a po zápasu). Podávání takové informace nemá naprosto žádný smysl.

Na závěr je vypsán řetězec, zda se vložení či úprava týmu podařila nebo nepodařila.



## Extrakce zápasů

Extrakce zápasů probíhá podobně až po rozdělení tabulky na buňky. Rozdílem je pouze pořadí při vybírání požadované tabulky, ta je zde až desátá. Následující zpracování řádků je už zajímavější.

Jako první je kontrolováno, zda v databázi jsou týmy, které zápas hrají. Pokud nejsou, nemůže na ně být navázán cizí klíč. Takový zápas se přeskakuje s vypsáním varování. Dále je kontrolováno, zda již daný zápas existuje. To se děje pomocí výběru z databáze s daným kolem, domácím a hostujícím týmem. Pokud takový zápas neexistuje, je potřeba zápas do databáze vložit.

Datum zápasu na stránkách Českého florbalu obsahuje pouze den a měsíc. Kvůli řazení událostí podle data je potřeba zjistit i rok. Z extrahovaného data si vyberu řetězec s měsícem. Pokud je aktuální měsíc v rozsahu leden až srpen, znamená to, že je extrakce spuštěna v druhém roce sezóny. V tomto případě je k datu připojen řetězec s rokem - aktuální rok -1 pro měsíce zápasu září až prosinec a aktuální rok pro měsíce zápasu leden až srpen. V případě, že je extrakce spuštěna v prvním roce sezóny (září až prosinec), je doplněn aktuální rok pro měsíce zápasu září až prosinec a aktuální rok -1 pro měsíc leden až srpen. Všechny statistiky zápasů včetně upraveného data jsou vloženy do databáze.

Pokud již daný zápas je v databázi, kontroluje se aktuálnost skóre a času. Skóre se změní při odehrání zápasu (bez odehrání je skóre zápasu :-). V případě změny skóre a tvoření aktuality je výsledek zápasu přidán do textu aktuality. Čas u některých pozdějších zápasů nemusí být na začátku sezóny znám, a proto je případně aktualizován.

Úspěšnost či neúspěšnost vložení či úpravy je zveřejněna uživateli vypsanou hláškou.

## Extrakce brankářů

Proces stažení stránky se statistikami brankářů až po rozdělení na buňky tabulky je stejný jako v minulých případech. Proces zpracování řádků tabulky a uložení do databáze je zde o něco málo složitější, jelikož se pracuje se dvěma tabulkami databáze zároveň – osoba a brankář. Je třeba zkontrolovat, zda osoba s daným jménem vůbec existuje v tabulce databáze. Pokud neexistuje, je vytvořen nový záznam osoby s daným jménem, postem brankáře a rolí hráče. Do tabulky brankář jsou statistiky osoby přidány se stejným ID, jako má ID jeho osoba v tabulce osob. Na závěr je vypsána hláška, zda se přidání nové osoby a brankáře podařilo či nikoliv.

I když je v tabulce osoba mnohem více atributů, zbylé není možno při této extrakci zjistit. Např. rodné číslo, telefonní číslo nebo adresu lze nastavit dodatečně ručně. Při extrakci by se dal vytvořit přístup vložím přihlašovacího jména a vygenerovaného hesla. Od této možnosti bylo ustoupeno, jelikož u některých osob není potřeba, aby měly přístup do informačního systému. Tím je myšleno, že s námi sice hrály párkrát zápas, ale na tréninky chodí s jiným mužským týmem či juniory. Pokud by extrakcí vygenerovaná osoba měla mít přístup, může jí ho nastavit později ručně administrátor či trenér.

Ale pokud již je osoba v databázi, je třeba zkontrolovat, zda k dané osobě již existuje záznam v tabulce brankář. Pokud záznam brankáře neexistuje, je statistika osoby přidána do tabulky a navázána cizím klíčem. Pokud již existuje záznam brankáře, je záznam pouze aktualizován. U každého vložení či úpravy je uživateli vypsáno, zda se akce podařila či nikoliv.

V případě generování aktuality jsou změny ve statistikách (popřípadě úplně nové statistiky) konkatenovány do řetězce textu této aktuality.

## Extrakce hráčů

Proces extrahování hráčů do databáze je téměř stejný jako u brankářů. Při vytváření nové osoby je do atributu uložen post, který může mít hráč nastaven v informačním systému Českého florbalu. Pokud post nastaven nemá, je zde pouze prázdný řetězec a může si ho nastavit později sám (včetně dalších údajů o osobě).

Jinak je extrahování kromě atributů naprosto stejné. Pokud osoba není v databázi, je vytvořen její záznam a k ní záznam v tabulce hráče. Pokud osoba již existuje, jsou její statistiky hráče aktualizovány či přidány (podle existence).

Pokud se tvoří aktualita, je rozdíl statistik (popřípadě i nové statistiky) konkatenován do textu aktuality. Na závěr je vypsána hláška, zda se související úpravy databáze podařily.

## Generované aktuality

Velkým usnadněním, co se týče informování hráčů a fanoušků, je automatické tvoření aktuality při extrakci. Toto generování bylo již stručně vysvětleno u každé z extrahovaných statistik. Zde bude shrnut celý průběh generování aktuality při extrakci.

Jak již bylo zmíněno, informace o statistikách hráčů a brankářů v jednotlivých zápasech nejsou uchovávány. Je to zbytečné, jelikož by bylo jen staženo a zobrazeno to, co může fanoušek vidět na odkaze pro online přenos. Zde také může vidět rozšířenější statistiky, jako góly po třetinách nebo průběh zápasu v čase. V takovém případě by byly staženy informace (jen o základních statistikách v utkání) méně cenné než online přenos. Náhračkou za uchovávání statistik v utkání je právě generovaná aktualita.

Základní myšlenkou je zjistit rozdíl stavu tabulek se statistikami před extrakcí a po extrakci. To je celkem lehce proveditelné – odečtením původní hodnoty od nové extrahované hodnoty. Pokud v databázi původně určitá statistika nebyla, znamená to, že jsou celé statistiky z dané doby mezi extrakcemi. Tento popsany algoritmus využívají statistiky zápasů, brankářů a hráčů. Další věcí k zamyšlení je, jak se pozná, že došlo ke změně statistik vlivem odehraného zápasu.

Ke zjištění, zda se bude tvořit aktualita, dochází již při extrakci týmů. Konkrétně u týmu s názvem FbC ReActive Aligators E a jeho stavem zápasů. Pokud se stav zápasů od poslední extrakce změní, spouští se celé tvoření aktuality. Pro aktualitu je z této tabulky využít pouze atribut bodů. Vygenerovaný titulek seznamuje čtenáře s počtem získaných bodů za období mezi extrakcemi.

Další důležitou informací je, s jakým výsledkem dopadly zápasy. Pokud je doplněno skóre zápasu, znamená to, že v čase mezi extrakcemi byl odehrán tento zápas. Skóre, domácí a hostující tým se pak objeví v textu aktuality. Ostatní atributy (datum, čas, kolo, hala a online přenos) nejsou tak důležité. Fanoušek si je případně bude moct dohledat v záložce zápasů.

Když už jsou uvedeny týmové statistiky, nesmí být opomenuty ty individuální. Pokud se brankář nebo hráč zúčastnil v daném rozmezí mezi extrakcemi nějakého zápasu, projeví se to v jeho počtu zápasů. V takovém případě je rozdíl statistik přidáván do aktuality.

Aby generovaná aktualita dobře vypadala, jsou statistiky vloženy do tabulky pomocí HTML kódu v ukládaném textu. U hráčů i brankářů je zjištěn rozdíl mezi předchozím a novým stavem vybraných atributů. Vybrané (důležitější) statistiky u brankáře jsou zápasy, minuty, obdržené góly, asistence a trestné minuty. U hráče jsou těmito statistikami zápasy, góly, asistence, kanadské body a trestné minuty. Tyto statistiky se samozřejmě neobejdou bez jména brankáře nebo hráče. Pokud daná osoba ještě žádné statistiky hráče nebo bran-

káře neměla, je jasné, že je také získala za dané období mezi extrakcemi. V takovém případě jsou její statistiky přidány bez odečítání předchozího stavu statistik.

Po proběhnutí extrakce je potřeba přichystaný titulek a text aktuality vložit do databáze. Do tabulky aktualit je vkládán titulek s počtem bodů a složený text se statistikami. Do data je vloženo aktuální datum. Autor aktuality není naplněn, jelikož je aktualita automaticky vygenerovaná a nemá autora.

### 5.4.3 Statistiky

Extrahované statistiky jsou uloženy v databázi. Nicméně jsou k ničemu, pokud nejsou zobrazeny uživateli. Jelikož tyto statistiky budou veřejně přístupné každému, není třeba ověřovat práva ani přihlášení uživatele.

Z databázové tabulky pro týmy je vytvořena tabulka soutěže. Týmy jsou seřazeny podle získaných bodů, případě vstřelených a obdržných branek. Tím se určuje i jejich pořadí v lize, které je uměle doplněno počítadlem v prvním sloupci. Zobrazení této tabulky lze vidět na obrázku 5.4.

Statistiky zápasů jsou při zobrazování seřazeny podle data a času. Všechny atributy tabulky jsou vypsané. Pokud jsou ručně doplněny atributy halý a online odkazu, jsou vypsané i tyto atributy. Online odkaz je vypsán přímo pomocí odkazu, nikoliv jen samotná cesta URL.

Statistiky hráčů jsou zobrazeny seřazené podle kanadských bodů a případně gólů. Jelikož i toto řazení přineslo v některých případech rovnocenné výsledky, výše v pořadí jsou v tabulce zobrazení hráči s menším počtem zápasů. Jelikož stejného počtu kanadských bodů dosáhli v méně zápasech, zaslouží si být lepší v pořadí. Pořadí statistik hráčů je určeno v prvním sloupci umělým počítadlem. Ke každému hráči je pro zajímavost doplněn počet kanadských bodů na zápas.

O řazení statistik brankářů by se dalo diskutovat. Nakonec bylo zvoleno řazení podle výher po vzoru internetových stránek Českého florbalu. Pro zajímavost je ke každému brankáři při zobrazování dopočítán počet obdržných branek na čas utkání.

Pod statistikami je odkaz na TOP statistiky, které jsou inspirovány internetovými stránkami Českého florbalu. TOP statistiky pouze doplňují základní statistiky a seřazují pět hráčů, kteří mají nejvyšší počet dané statistiky v různých kritériích. Těmito kritérii jsou kanadské body, zápasy, góly, asistence, výhry (pro brankáře) a trestné minuty.

Pořadí	Název týmu	Zápasy	Výhry	Remízy	Prohry	Body	Branky vstřelené	Branky obdržené	Brankový rozdíl	Úspěšnost přesílovek	Úspěšnost oslabení	Trestné minuty
1.	FbC ReActive Alligators E	21	17	2	2	53	168	70	98	25.00	91.30	54
2.	MAH Košér Orel Telnice C	21	16	2	3	50	181	97	84	40.00	73.33	38
3.	FBC Sokol Ivanovice na Haně B	21	11	3	7	36	124	96	28	38.46	82.14	75
4.	FBK Orel Borítov "B"	21	8	2	11	26	91	136	-45	14.29	80.00	53
5.	Brankovice Black Arrows	21	8	1	12	25	81	105	-24	10.00	90.00	91
6.	TJ SOKOL LETOVICE	21	6	3	12	21	115	134	-19	16.00	55.56	30
7.	F.A.T.R. B	21	5	2	14	17	74	127	-53	12.50	77.78	66
8.	FBC BISON SLAVKOV B	21	4	3	14	15	85	154	-69	16.67	47.83	128

Obrázek 5.4: Zobrazení statistik týmů

#### 5.4.4 Uživatelé

V informačním systému je třeba rozlišit rozdílná přístupová práva pro různé akce. Typicky se přístupová práva rozlišují pomocí přihlašovacího systému a atributu, který udává pravomoc.

##### Přihlašování

V Nette se přihlašování řeší pomocí vytvoření modelu Authenticator. Ten se vytváří ve složkové struktuře /app/model/Authenticator.php. Tento skript kontroluje, zda je zadané přihlašovací jméno a heslo v databázi. Pokud není, vypisuje se chybová hláška. Pokud ano, je naplněna globální proměnná user do session. V této proměnné jsou dostupné všechny atributy přihlášeného uživatele. Výjimkou je heslo, které je po přihlášení smazáno.

Díky tomuto vytvořenému modelu je možné kdekoliv v aplikaci volat funkci *login* (*login*, *heslo*), která automaticky zkontroluje, zda jsou přihlašovací údaje v pořádku. Tato funkce je volána v controlleru app/presenters/LoginPresenter, který zpracovává přihlašovací údaje zadané do formuláře.

V přihlašovacím formuláři je i checkbox s možností zapamatovat si přihlášeného uživatele. Pokud není zatržen, session vyprší po 30 minutách neaktivity. Pokud je požadování zapamatování, session by měla vypršet po 14 dnech. Tuto funkci kazí nastavení hostingu, který session maže po pár desítkách hodin.

V případě zapomenutého hesla je zde možnost automatické obnovy. Podmínkou je mít předem vyplněný funkční e-mail. Pokud je správně vyplněno uživatelské jméno a e-mail, je na danou e-mailovou adresu odeslán e-mail s novým vygenerovaným desetimístním heslem. Pokud zadané údaje nejsou správné, vypíše se telefonní číslo administrátora, který případně ve správě osob nastaví nové heslo.

Pokud se uživatel odhlásí, je uživatel odhlášen pomocí metody *logout*, která je volána na uživatele. Tato metoda uvolní session s uživatelem a ten je odhlášen.

## Přístupová práva

Jakmile je uživatel přihlášen, je možnost ošetřovat, zda má přístup na danou stránku. K tomu slouží atribut `ROLE`, jehož číselná hodnota značí výši pravomocí. V informačním systému se následně zobrazuje obsah stránky jenv případě, že je role větší než nějaké číslo. Pokud uživatel nemá pravomoc zobrazit obsah stránky, je zobrazena pouze odpovídající hláška s upozorněním. Takové stránky lze dosáhnout pouze ručním zadáním cesty URL, jelikož odkazy, na které by se uživatel neměl dostat, jsou skryty stejným způsobem.

V informačním systému pro florbalový tým jsou tři různé role uživatelů – hráč, trenér a administrátor. Základní funkce může používat i nepřihlášený uživatel. Mezi tyto funkce patří prohlížení aktualit, tabulky soutěže, statistik hráčů, zápasů a minulých sezón.

**Hráč** dědí pravomoci nepřihlášeného uživatele. Navíc se může přihlašovat na události nebo k nim psát komentáře. Ve správě může spustit extrakci dat, upravit svůj profil a zobrazit přehled hal kvůli zjištění adresy.

**Trenér** má stejné pravomoci jako hráč. Navíc může psát aktuality a vytvářet tréninky. Ve správě může spravovat databázi. Konkrétně tréninky, osoby a haly. U zápasů může upravovat pouze halu a online přenos, zbytek je automaticky stahován. Odstraňovat může pouze tréninky, jelikož odstraněním haly, zápasu nebo osoby by mohlo dojít ke ztrátě některých údajů. Smazání haly a osoby by mohlo způsobit, že v databázi budou smazána důležitá data, jelikož jsou na tyto tabulky navázány jiné tabulky pomocí cizího klíče. Tyto akce, které mohou mít nebezpečný vliv na databázi, může vykonávat pouze **administrátor**.

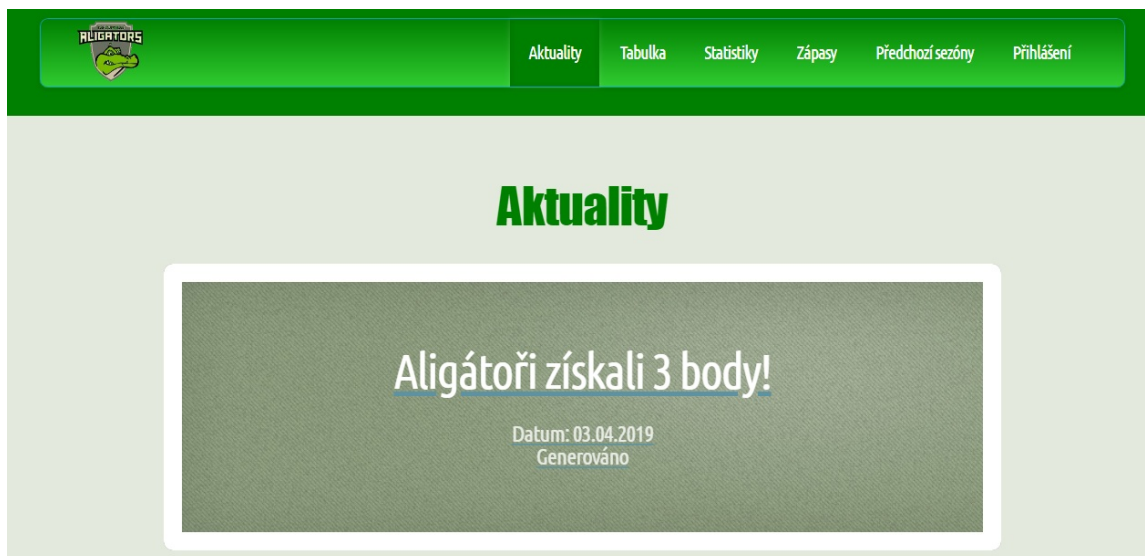
Pro testování byly vytvořeny následující účty s danými právy:

- přihlašovací jméno „Hrac“ s heslem „Hrac12345“,
- přihlašovací jméno „Trenér“ s heslem „Trenér12346“,
- přihlašovací jméno „Admin“ s heslem „Admin12347“.

### 5.4.5 Aktuality

Aktuality zobrazují deset nejnovějších napsaných aktualit. Řazení při výběru z databáze je nastaveno na výběr od nejvyššího po nejnižší ID s limitem na 10 záznamů. Na hlavní stránce aktualit jsou aktuality zobrazeny pomocí banneru s titulkem, který funguje jako odkaz. Po kliknutí na odkaz je zobrazen detail aktuality včetně textu. Zobrazení hlavní stránky s odkazem na aktualitu lze vidět na obrázku 5.5.

Trenér a administrátor může vložit aktualitu. Pokud se tak rozhodne, musí vložit datum, titulek a text aktuality. Jako autor bude vložen přihlášený uživatel. Aktualita po vložení dostane nejvyšší ID a starší aktuality se posunou směrem dolů.



Obrázek 5.5: Zobrazení odkazů na aktuality

#### 5.4.6 Docházka

V docházce je zobrazeno pět následujících událostí podle data a času. Pro zobrazení těchto událostí je využit stejný banner jako v případě aktualit. Na banneru je zobrazen typ události (zápas nebo trénink), hala, datum a čas. Bannery fungují jako odkazy. Po otevření nějaké události je možnost vyjádřit, zda se daná osoba události zúčastní nebo ne. Odpověď v anketě je uložena (případně aktualizována) a vypsána do tabulky pod anketu. Zároveň je zde součet jednotlivých odpovědí. Komentáře pod anketou jsou vkládány do databáze a zobrazovány od nejnovějšího po nejstarší. Ukázkou docházky na jednu událost lze vidět na obrázku 5.6.

Trenér a administrátor mohou vytvářet tréninky. Ke každému tréninku je třeba vyplnit datum, čas a halu. V takovém případě je vytvořen pouze jeden trénink. Pro vytvoření více tréninků je třeba zatrhnout checkbox. Počáteční datum je v tom případě použito z první části formuláře. Cyklem jsou vkládány tréninky každý týden až po druhé uvedené datum.

**Událost 12.05.2019 18:30**

Varianta:  Přijdu  
 Přijdu chytat  
 Nepřijdu

**Uložit**

Přijde: 1.  
Přijde chytat: 0.  
Nepřijde: 0.

Jméno	Účast
Jakub Zmek	Přijdu

**Komentáře**

Komentář:

**Uložit komentář**

Jméno	Komentář
Jakub Zmek	+1 hráč do pole

Obrázek 5.6: Formulář pro přihlašování na událost

#### 5.4.7 Správa

Ačkoliv se databáze dá spravovat pomocí příkazů, taková správa je uživatelsky nepřijemná. Proto je třeba vytvořit uživatelské rozhraní, které bude umožňovat přidávat, upravovat a mazat záznamy určitých tabulek v databázi.

Ve správě je možnost upravovat obsah databáze co se týká zápasů, tréninků, osob, haly a extrakčních odkazů. Typická správa tabulky databáze vypisuje všechny záznamy z tabulky i s jejich atributy. U každého takového záznamu je možnost jej upravit nebo smazat. Navíc je možnost přidat nový záznam. Mezi popsany typ správy patří správa hal, osob a tréninků. Správa osob navíc obsahuje jednu funkci navíc – možnost vytvořit nové heslo. V případě ztráty hesla uživatelem tak může administrátor stisknout toto tlačítko u daného záznamu a vyplnit nové heslo. Ve správě osob je navíc skryto heslo čtyřmi hvězdičkami. Tyto hvězdičky značí, že je heslo nastavené a osoba má přístup (v kombinaci s přihlašovacím jménem). Pokud osoba nemá nastavené heslo, je kolonka prázdná. Ukázkou správy osob lze vidět na obrázku 5.7. Zmíněné databázové operace jsou umožněny pomocí ikon, které byly staženy ze serverů<sup>5</sup> pro stahování ikon zdarma.

Trošku jinou možnost správy nabízí zápasy a extrakční odkazy. Je to dáno tím, že počet těchto záznamů je předem dán. Počet zápasů je dán pomocí extrakce zápasů a extrakční odkazy obsahují čtyři odkazy – pro týmy, zápasy, brankáře a hráče. V tomto případě se počet nedá změnit a odpadají možnosti přidávání a mazání záznamů. U extrakčních odkazů tak lze pouze změnit dané čtyři odkazy pro extrakci. U zápasů je většina atributů pravdivě předvyplněná díky extrakci, a proto není třeba tyto atributy měnit. Mezi nestažené atributy patří hala a odkaz na online přenos. Proto je možnost upravovat pouze tyto dva atributy u každého záznamu zápasu.

Kromě základní správy tabulek je zde možnost, aby si každý přihlášený uživatel upravil své údaje. Pokud každý hráč této možnosti využije a bude si pravidelně aktualizovat svůj profil, bude tím ušetřeno mnoho práce trenérovi nebo administrátorovi.

<sup>5</sup>Použité ikony byly staženy ze serverů <http://www.iconarchive.com> a <http://www.softicons.com>

Dalšími funkcemi jsou naplnění a vyčištění statistik z databáze. Pro naplnění slouží odkaz na již zmíněnou extrakci. Aby byla možnost nějak vyčistit databázi před extrahováním statistik z nové sezóny, je k dispozici funkce ukončit sezónu. Tato funkce smaže veškeré statistiky týmů, zápasů, brankářů a hráčů. Aby nedošlo k úplné ztrátě těchto statistik, je zde možnost statistiky uchovat. Po zatrhnutí checkpointu jsou statistiky uloženy ve formě textu do předchozích sezón. Ukončení sezóny je povinná funkce před extrahováním statistik z nové sezóny, jelikož je většina statistik při extrakci pouze upravována, nikoliv mazána. Tím pádem by se např. v nové sezóně nové týmy přidaly k týmům z minulé sezóny a vznikla by nepravdivá tabulka soutěže.

## Správa osob



Jméno	Rodné číslo	Číslo dresu	Post	Adresa	Tel.č.	E-mail	Role	Login	Heslo			
Admin							Admin	Admin	****			
František Chmela							Hráč					
Hráč							Hrac	****				
Jakub Gremmel							Hráč					
Jakub Zmek	123456/7890	35	G	Voriskova 2	607833917	jakub.zmek@seznam.cz	Admin	Zmek	****			
Jan Oliva	543454/4530	48	O	Kosmákova 32	608743453	Oliva@seznam.cz	Hráč	Oliva	****			
Jiří Sahan	154357/1321	17	U	Vondrákova 14	609045345	Sahan@seznam.cz	Hráč	Sahan	****			
Ladislav Dufek							Hráč					
Lukáš Gregr	7854132/4515	87	U	Bohunice	774213547	Gregr@seznam.cz	Trenér	Gregr	****			
Lukáš Jerábek	947818/4561	14	O	Horní 22	774125874	JerabekL@seznam.cz	Hráč	JerabekL	****			
Martin Krumpolc							Hráč					
Michal Dubský	453421/4652	28	G	Bozetechova 89	774024458	Dubsky@seznam.cz	Hráč	Dubsky	****			
Ondřej Frank							Hráč					
Ondřej Slezák	453578/3512	96	U	Šlapanice	774323456	Slezak@seznam.cz	Hráč	Slezak	****			

Obrázek 5.7: Správa osob



## Kapitola 6

# Testování

Práce na programu nekončí implementací. Je třeba ještě otestovat, zda jsou implementované části funkční a zda dělají to, co mají. Zjištěné chyby je třeba opravit. Také je třeba zjistit, zda je implementovaný informační systém podle představ zákazníka nebo uživatelů. Těmto požadavkům je třeba následně vyhovět nějakým návrhem řešení.

Testování funkčnosti probíhalo v internetových prohlížečích Google Chrome, Mozilla Firefox a Microsoft Edge. V těchto prohlížečích se informační systém chová očekávaným způsobem.

### 6.1 Testování programátorem

Testování funkčnosti bylo prováděno většinou ihned po naprogramování dané funkce. Bylo třeba zjistit, zda se daná implementovaná funkce chová tak, jak je očekáváno. Pokud ne, tyto problémy byly řešeny okamžitě, což je výhodné hlavně z hlediska paměti programátora. Přesto došla řada na konečné testování funkčnosti, do které bylo zahrnuto několik experimentů.

Ačkoliv existuje podpora pro testování v Nette<sup>1</sup>, nebyla v této práci využita. Toto testování umožňuje vytvořit si různé testy v jazyku PHP. Tyto testy je možné spustit a zjistit, zda je dosaženo očekávaného výsledku. Testování tímto způsobem je spíše vhodné pro aplikace, které využívají funkce, objekty apod. Tyto podsystémy se zde dají testovat a ověřovat, zda vrací očekávané výsledky.

V informačním systému pro florbalový tým je třeba testovat jiným způsobem. Testuje se ručním způsobem - tj. zda dané tlačítko odkazuje tam kam má, zda funguje odeslání formuláře apod. Je zde třeba také otestovat, zda jsou z databáze vybrány očekávané výsledky. Pro všechno zmíněné testování naprosto stačí testování v uživatelském rozhraní.

### Simulace tréninku

Prvním menším experimentem byla simulace tréninku. Přihlašoval jsem se z náhodných účtů a zadával jsem za danou osobu účast na vytvořený trénink. Za některé osoby jsem napsal dodatečný komentář k tréninku. Odpovědi se přidávaly do tabulky, v případě změny účasti byla odpověď upravena i v tabulce. Komentáře byly seřazeny správně podle přidání od nejnovějšího po nejstarší.

---

<sup>1</sup>Stručný popis: <https://tester.nette.org/cs/guide>

Ačkoliv vše fungovalo tak jak mělo, rozhodl jsem se pro změnu. Výpis účasti hráčů se mi zdál chaotický. Rozhodl jsem se tak řadit účast podle abecedy odzadu. Z toho vyplynulo pořadí „Přijdu chytat“, „Přijdu“ a „Nepřijdu“. Toto pořadí je dle mého názoru logické, jelikož více by tým měly zajímat odpovědi s účastí, než ty s neúčastí.

Kromě seřazení odpovědí tomu stále něco chybělo. Konkrétně usnadnění zjištění, kolik lidí přijde a zda má tudíž trénink vůbec cenu. Nad tabulkou výpisů odpovědí je tak sečten i celkový počet jednotlivých druhů odpovědí.

## Generování aktuality

Jelikož funkce generování aktuality byla implementována později než samotná extrakce a mezitím skončila sezóna, nebyl jsem schopen spustit extrakci po nějakém zápase týmu. Musel jsem tvoření aktuality vyvolat ručně a řádně tuto funkci otestovat. Abych tuto funkci otestoval, bylo třeba ručně změnit data v databázi. K tomu jsem využil uživatelské prostředí pro práci s databází phpMyAdmin.

Pro testování vygenerované aktuality jsem si vybral první hrací den 22. září 2018 s dvěma zápasy. Našemu týmu jsem odečetl dva zápasy a ubral 3 body. Výsledek obou zápasů jsem smazal. Jelikož jsem dané zápasy chytal já, ubral jsem si daný počet zápasů a nějaký počet obdržených gólů. Jelikož jsem u hráčů až tak nutně nepotřeboval určovat přesné statistiky v daných zápasech, ubral jsem náhodným hráčům náhodné atributy. I když vstřelené ani obdržené góly nesouhlasí s výsledky zápasů, při normálně vytvořené aktualitě by samozřejmě seděly.

Výsledek tohoto testování generování aktuality byl uložen do skriptu pro naplnění databáze jako ukázka, jak by vygenerovaná aktualita mohla vypadat. Tato aktualita je tudíž dostupná na hlavní stránce. U aktuality je místo autora popis, že je generována.

## Extrakce dat s nežádoucím odkazem

Úprava extrakčních odkazů je velmi nebezpečná věc a před změnou by se měl každý uživatel několikrát ujistit, že je odkaz v pořádku. Tímto experimentem bude zjištěno, co se stane, když si administrátor nedá pozor a naplní databázi nežádoucím extrakčním odkazem.

Míra nebezpečnosti záleží hlavně na počtu tabulek na dané webové stránce. U extrakce statistik týmů je riziko vysoké, jelikož se pracuje s druhou tabulkou na stránce. U extrakce statistik zápasů, hráčů a brankářů je vyšší šance, že proběhne beze změn, jelikož pracuje až s desátou tabulkou. Když už je tabulka v daném pořadí na stránce, hrozí, že bude vypsána hromada varování, jelikož se pracuje s očekávaným počtem buněk v tabulce. Nakonec hrozí, že tabulka bude naplněna nesmyslnými hodnotami.

Pro ověření těchto úvah byla vybrána webová stránka fakulty<sup>2</sup>. Zatímco tabulka s týmy byla naplněna nesmyslnými hodnotami, extrakce ostatních statistik proběhla bez povšimnutí díky neexistenci desáté tabulky.

Dospěl jsem k závěru, že tomuto problému se nedá vyhnout. Administrátor musí být opatrný a platnost odkazu pečlivě prověřit. V případě nežádoucích dat v databázi lze tato data smazat pomocí funkce ukončení sezóny (bez uložení dat). Bylo možné ošetřit pouze skutečnost, zda je odkaz platný, tj. existuje stránka se zadanou URL.

---

<sup>2</sup><http://www.fit.vutbr.cz/>

## Univerzálnost informačního systému (extrakce dat)

Jelikož budeme postupovat o ligu výše, je třeba zjistit, zda je struktura webu se statistikami stejná či nikoliv. Pro tento experiment byl vybrán náhodný tým z vyšší ligy, jehož extrakční odkazy byly vloženy do databáze.

Všechny statistiky tohoto týmu byly vloženy v pořádku do databáze. Proto se dá informační systém označit za univerzální mezi ligami. Po drobných změnách se se dá označit i jako univerzální mezi týmy. Pokud by měl informační systém požívat jiný tým, mělo by být změněno logo a v případě zájmu i barva menu na barvu týmu. Jelikož generování aktuality reaguje na název týmu „FbC ReActive Aligators E“, bylo by vhodné změnit i název týmu.

## 6.2 Testování uživatelů

Pro uživatelské testování jsem zvolil osoby, které tento informační systém budou využívat. Jedná se o spoluhráče z týmu.

Díky tomuto testování bylo možné získat nějaké podněty od budoucích uživatelů. Tyto podněty jsou nejčastěji vzhledové – ať už se uživatel něco nelíbí nebo se někomu zdá informační systém nepřehledný. Testování uživateli je velmi důležité i při výrobě programu na zakázku. Že byl nějaký program vytvořen podle představ programátora, neznamená, že se všem musí zdát stejně dokonalý.

Testování spoluhráči probíhalo po kompletní implementaci informačního systému. Ačkoliv funkčnost již byla testována programátorem, bylo dost možné, že se mohla objevit nějaká chyba. Hlavně co se týká neotestovaných internetových prohlížečů, jelikož nějaké funkce nemusí jiný prohlížeč podporovat. Každému hráči byl vytvořen přístup do informačního systému. Dobrovolníci si vyzkoušeli přihlášení na trénink, čtení aktualit, úpravu svých údajů či extrakci dat. Trenér si vyzkoušel vytvoření aktuality a tréninků.

Reakce byly vesměs pozitivní s drobnými výhradami. V zásadě se jednalo pouze o drobné vzhledové výhrady. Např. proč nebyla použita jiná barva pro danou věc, proč logo není větší apod. S celkovou přehledností informačního systému byli všichni dobrovolníci spokojeni. Nikdo neobjevil žádnou chybu funkčnosti v informačním systému.

# Kapitola 7

## Závěr

Cílem této práce bylo vytvořit informační systém pro správu florbalového týmu. Informační systém byl implementován pro florbalový tým FbC ReActive Aligators E. Tento tým chtěl mít přehledně dostupnou správu týmu na jedné webové stránce.

Nastudoval jsem si HTML, CSS, MySQL, PHP, Nette, webové technologie a extrahování dat z jiné webové stránky. Po této studijní části jsem byl schopen informační systém implementovat.

Před implementací bylo třeba provést analýzu požadavků. To znamená, že mi bylo řečeno, co by daná stránka měla umět. Výsledkem této analýzy byl diagram případů užití, který znázorňuje možnosti jednotlivých typů účtu. Tento diagram byl předveden týmu a jím odsouhlasen. Jako model požadavků byl sestaven ER diagram.

Na základě požadavků bylo možné sestavit návrh informačního systému. Návrh relační databáze byl vytvořen transformací z ER diagramu. Byl navržen vzhled informačního systému. Informační systém byl rozdělen na podproblémy.

Na základě schématu databáze byl vytvořen skript pro inicializaci databáze a naplnění vzorkem hodnot. Následně byly implementovány postupně jednotlivé podproblémy informačního systému.

Testování funkčnosti jsem prováděl osobně při implementaci programu. Po celkové implementaci jsem provedl několik zajímavých experimentů, ze kterých vyšlo určité vylepšení. Uživatelské testování probíhalo se spoluhráči, kteří si zkusili použít různé funkce informačního systému a dále mi zpětnou vazbu.

Implementovaný informační systém umožňuje základní správu florbalového týmu. Existuje několik dalších zajímavých funkcí, které by mohly v budoucnu rozšířit informační systém. Mohly by být extrahovány i statistiky ze zápasů. Tím jsou myšleny nejen kanadské body, ale i průběh zápasu, jednotlivé časy gólů apod. Z toho by se daly odvodit zajímavé statistiky týkající se průběhu zápasu. Mezi takové statistiky může patřit celkové skóre týmu po jednotlivých třetinách, po jaké době se průměrně podaří využít přesilovku apod.

Další užitečnou funkcí by mohla být správa financí. K tomuto účelu by měla být vytvořena nová funkce pokladníka. Ten by mohl vytvářet nové položky k zaplacení (licence, příspěvek klubu, pronájem haly). K hráčům by následně doplňoval, zda mají danou položku zaplacenou či nikoliv.

Mohlo by být využito mnohem větší množství akcí ke generování aktuality. Kdyby při extrakci byla nalezena nová osoba, byla by vygenerována aktualita s novými posilami. Aktualita s odchody hráčů by mohla být tvořena při odstranění osoby z databáze.

Informační systém bude využíván hráči florbalového týmu FbC ReActive Aligators E. Přes léto bude probíhat podrobnější testování pro získání podrobnější zpětné vazby.



# Literatura

- [1] *Beginner's guide to Web Scraping with PHP*. [Online; navštíveno 18.04.2019].  
URL <http://www.prowebscraper.com/blog/web-scraping-using-php/>
- [2] *Top 10 Programming Languages For Web Development*. [Online; navštíveno 11.05.2019].  
URL <https://www.rswebsols.com/tutorials/programming/top-10-programming-languages-web-development>
- [3] *Programovací jazyky*. 2016, [Online; navštíveno 15.04.2019].  
URL <http://www.jasnyweb.cz/programovaci-jazyky/>
- [4] *Seznámení se s Nette Frameworkem*. 2018, [Online; navštíveno 16.04.2019].  
URL <https://doc.nette.org/cs/3.0/getting-started>
- [5] *MVC aplikace presentery*. 2019, [Online; navštíveno 17.04.2019].  
URL <https://doc.nette.org/cs/3.0/presenters>
- [6] *Píšeme první aplikaci*. 2019, [Online; navštíveno 16.04.2019].  
URL <https://doc.nette.org/cs/3.0/quickstart>
- [7] Adobe: *Informace o webových aplikacích*. 2017, [Online; navštíveno 14.04.2019].  
URL <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>
- [8] Břešťan, J.: *Rozdíly mezi MVC a MVP*. devel.cz, 2014, [Online; navštíveno 14.04.2019].  
URL <https://devel.cz/otazka/rozdily-mezi-mvc-a-mvp>
- [9] Croft, J.; Lloyd, I.; Rubin, D.: *Mistrovství v CSS: pokročilé techniky pro webové designéry a vývojáře*. Brno: Computer Press, 2007, ISBN 978-80-251-1705-7.
- [10] Dalton: *Difference between asp.net MVC and MVP? Are they both same?* stackoverflow.com, 2013, [Online; navštíveno 14.04.2019].  
URL <https://stackoverflow.com/questions/19996963/difference-between-asp-net-mvc-and-mvp-are-they-both-same>
- [11] Kosek, J.: *PHP - tvorba interaktivních internetových aplikací: podrobný průvodce*. Praha: Grada, 1998, ISBN 80-7169-373-1.
- [12] Maslakowski, M.: *Naučte se MySQL za 21 dní*. Praha: Computer Press, 2001, ISBN 80-7226-448-6.

- [13] Zoner Software, a.: *10 nejlepších frameworků pro vývojáře* . 2015, [Online; navštíveno 15.04.2019].  
URL <https://www.interval.cz/clanky/10-nejlepsich-php-frameworku-pro-vyvojare/>

# Příloha A

## DVD

DVD se zdrojovými kódy a dokumentací. Složková struktura DVD obsahuje:

- Zdrojové kódy programu jsou ve složce `/sources`.
  - Složka `/sources/app` obsahuje konfiguraci (`app/config`), kódy vrstev MVC controller (`app/presenters`) a view (`app/presenters/templates`).
  - Složka `/sources/www` obsahuje soubory přístupné z webu - CSS soubory, obrázky a skript pro extrakci.
- Dokumentace ve zdrojové podobě je ve složce `/doc`.
- Dokumentace ve formátu PDF je v souboru `xzmekj00.pdf`.



## Příloha B

# Návod k instalaci

Pro běh aplikace je třeba inicializovat databázi. Při vytváření na hostingu je databáze většinou přístupná přes uživatelské prostředí phpMyAdmin. V záložce import je třeba nahrát skript pro inicializaci databáze `database.sql`. Před nahráním je žádoucí zapoznámkovat smazání záznamů z tabulek (sada příkazů *DELETE FROM*). Protože při prvotní inicializaci tyto tabulky neexistují, způsobí to chybu.

Je třeba s vytvořenou databází spojit informační systém. Toto spojení je vytvářeno v souborech `www/init.php` a `app/config/config.neon`.

Na závěr je třeba složkovou strukturu informačního systému překopírovat na vytvořený hosting pro webovou stránku. V případě linuxového serveru je třeba nastavit, aby byla složka `www` veřejně přístupná.

## Příloha C

# Uživatelská příručka

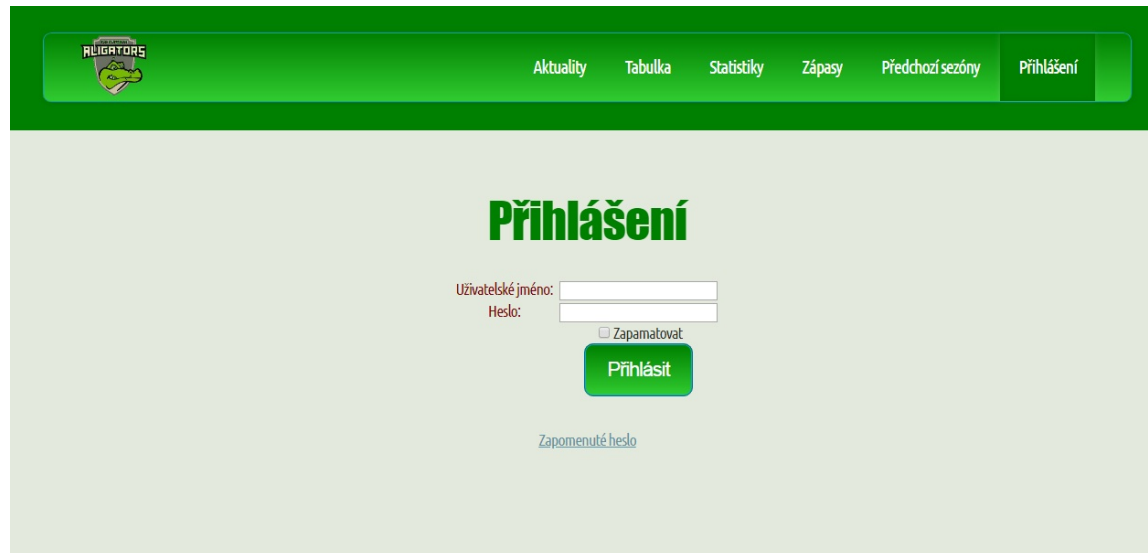
### C.1 Přihlášení

Pokud máte přístupové údaje, jako první je žádoucí se přihlásit. Poté budete mít přístup k více funkcím.

Možnost přihlášení najdete v záložce "Přihlášení". Vyplňte přihlašovací jméno a heslo. Zatrhnete zapamatovat v případě, že chcete zůstat na daném počítači déle přihlášení.

V případě, že jste zapomněl heslo, klikněte na odkaz "Zapomenuté heslo" a vyplňte formulář. V případě neúspěchu můžete kontaktovat administrátora na zobrazeném telefonním čísle.

Pokud se přihlášení podaří, zobrazí se Vám, že jste přihlášen. Formulář pro přihlášení lze vidět na obrázku C.1.

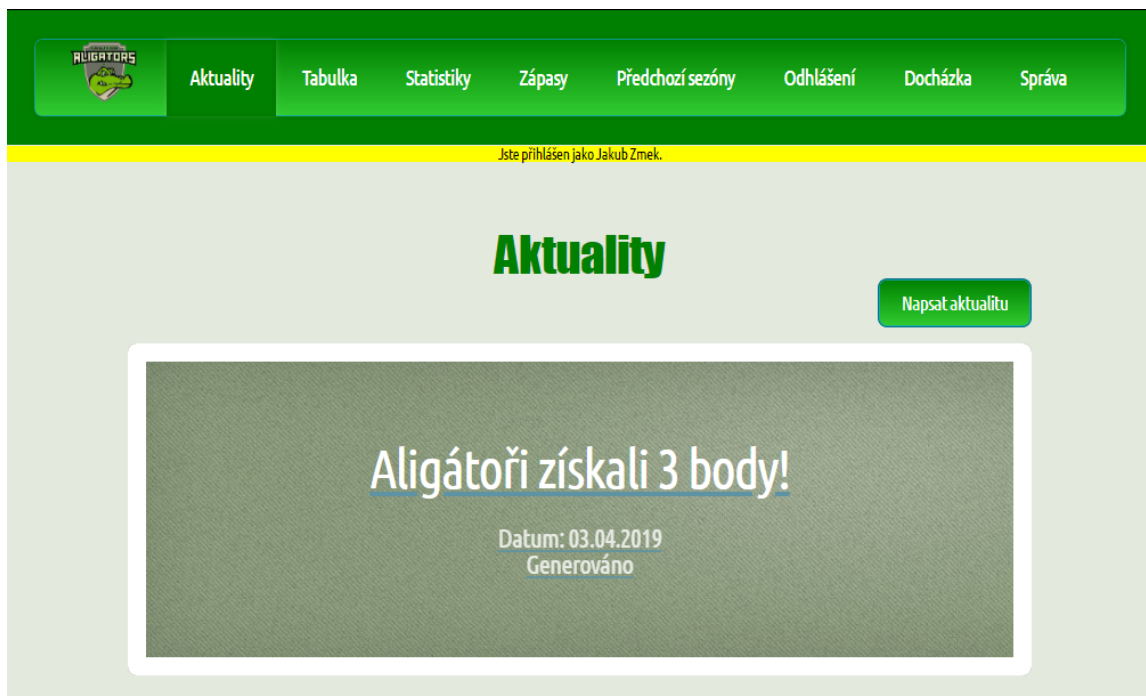


Obrázek C.1: Přihlášení

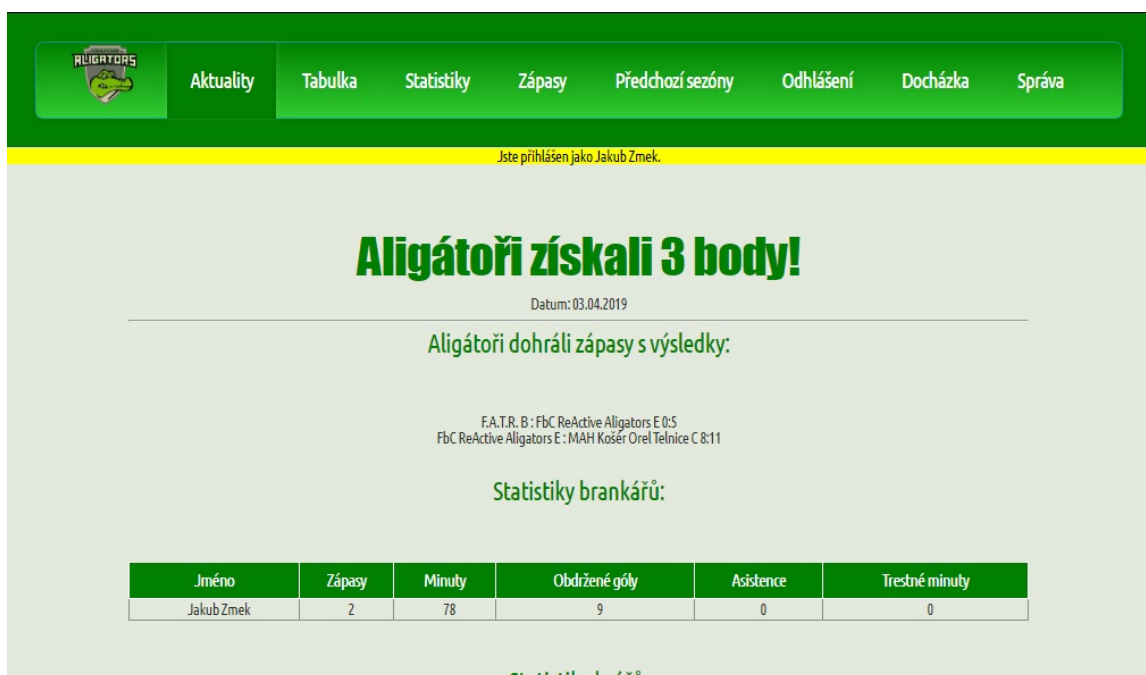
Po přihlášení Vám budou zpřístupněny funkce, které máte nastavené. Pokud některou z možností nemůžete vykonávat, přeskočte ji.

## C.2 Aktuality

Pokud si chcete přečíst nějakou aktualitu, klikněte na daný banner aktuality. Po kliknutí na danou aktualitu se Vám zobrazí text aktuality. Toto lze vidět na obrázcích C.2 a C.3.



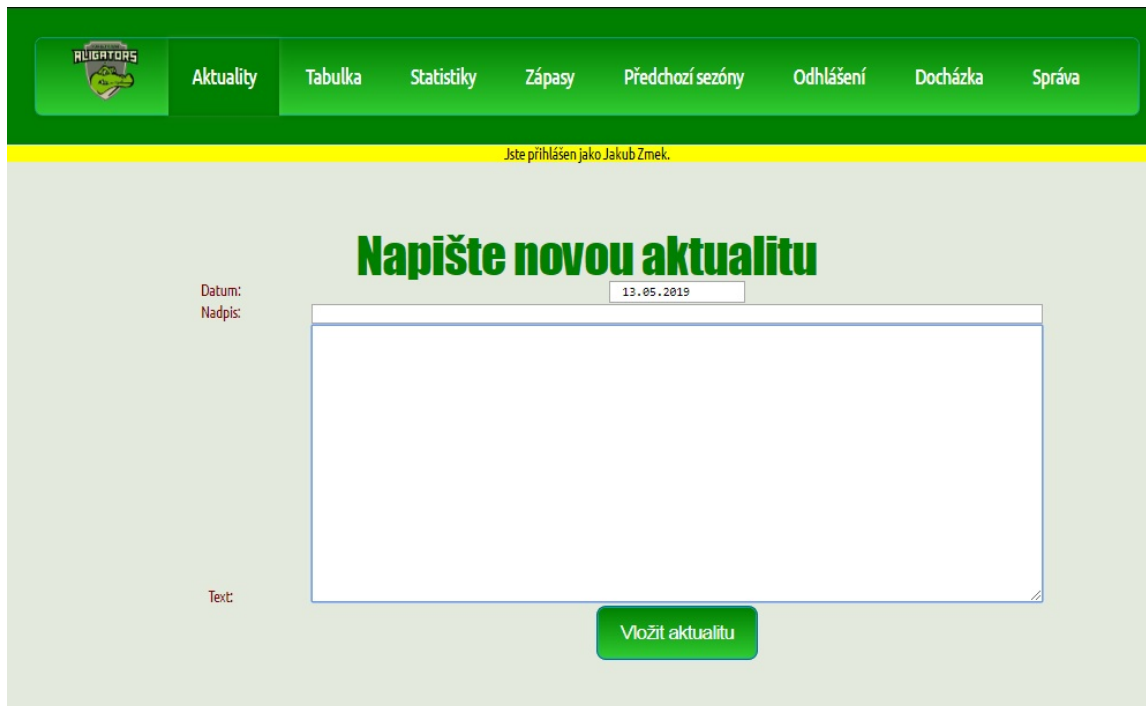
Obrázek C.2: Aktuality



Obrázek C.3: Text aktuality

## Vytvoření aktuality

Po stisknutí tlačítka "Napsat aktualitu" je možné vytvořit novou aktualitu. Vyplňte ve formuláři datum, titulek a text aktuality a potvrďte tlačítkem. Formulář lze vidět na obrázku C.4.

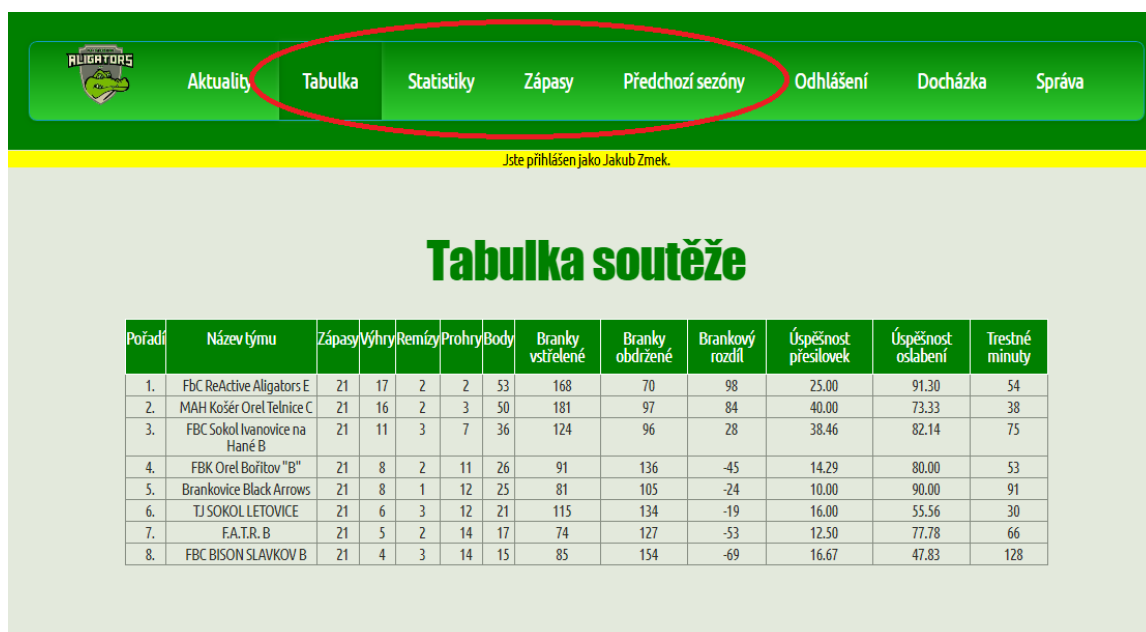


The screenshot shows a web interface for creating a news item. At the top, there is a green navigation bar with a logo on the left and several menu items: "Aktuality", "Tabulka", "Statistiky", "Zápasy", "Předchozí sezóny", "Odhlášení", "Docházka", and "Správa". Below the navigation bar, a yellow status bar indicates the user is logged in as "Jakub Zmek". The main content area has a light green background and features the heading "Napište novou aktualitu" in large green text. To the left of the form, there are labels for "Datum:", "Nápis:", and "Text:". The "Datum:" label is positioned above a date input field containing "13.05.2019". The "Nápis:" label is positioned above a large, empty text input area. The "Text:" label is positioned below the text input area. At the bottom center of the form, there is a green button labeled "Vložit aktualitu".

Obrázek C.4: Vytvoření aktuality

## C.3 Statistiky

Pokud se chcete dozvědět něco o tom, jak se daří týmu, použijte záložky menu "Tabulka", "Statistiky" a "Zápasy". V těchto záložkách se nachází statistiky týmů v lize, statistiky hráčů a brankářů týmů a výsledky zápasů. V případě, že by Vás zajímaly statistiky z minulých sezón, zvolte záložku "Předchozí sezóny". Ukázku z programu naleznete na obrázku C.5.



Pořadí	Název týmu	Zápasy	Výhry	Remízy	Prohry	Body	Branky vstřelené	Branky obdržené	Brankový rozdíl	Úspěšnost přesílovek	Úspěšnost oslabení	Trestné minuty
1.	FbC ReActive Aligators E	21	17	2	2	53	168	70	98	25.00	91.30	54
2.	MAH Kožér Orel Telnice C	21	16	2	3	50	181	97	84	40.00	73.33	38
3.	FBC Sokol Ivanovice na Hané B	21	11	3	7	36	124	96	28	38.46	82.14	75
4.	FBK Orel Bořitov "B"	21	8	2	11	26	91	136	-45	14.29	80.00	53
5.	Brankovice Black Arrows	21	8	1	12	25	81	105	-24	10.00	90.00	91
6.	TJ SOKOL LETOVICE	21	6	3	12	21	115	134	-19	16.00	55.56	30
7.	F.A.T.R. B	21	5	2	14	17	74	127	-53	12.50	77.78	66
8.	FBC BISON SLAVKOV B	21	4	3	14	15	85	154	-69	16.67	47.83	128

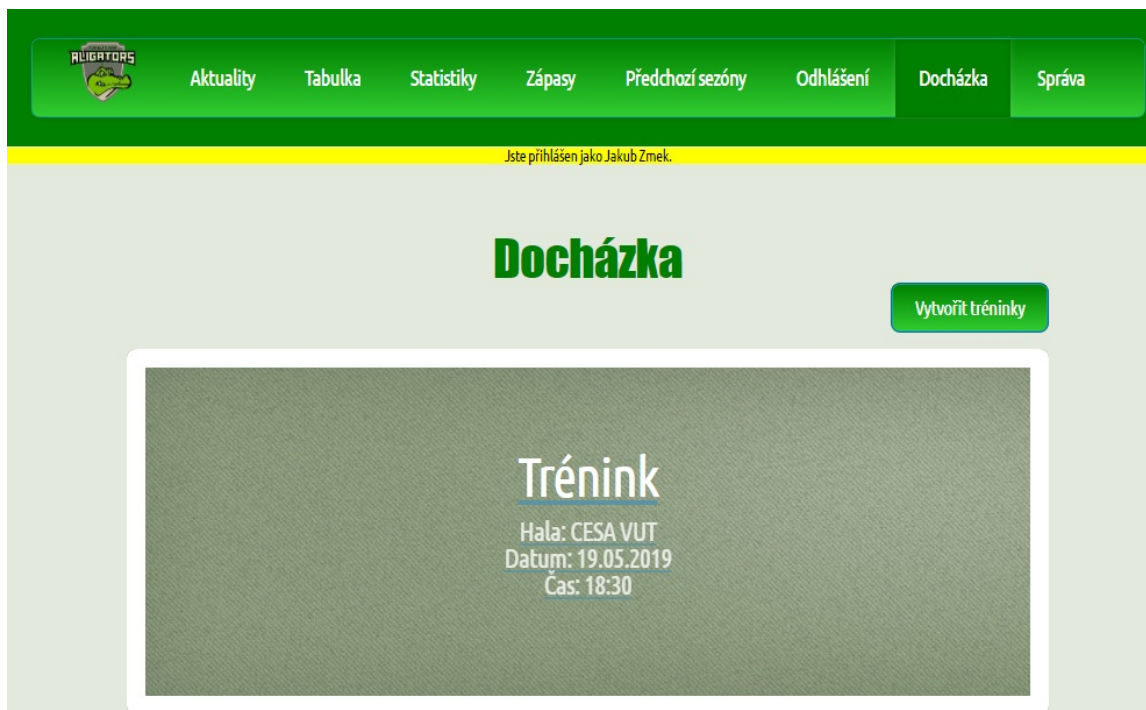
Obrázek C.5: Ukázka statistik

## C.4 Docházka

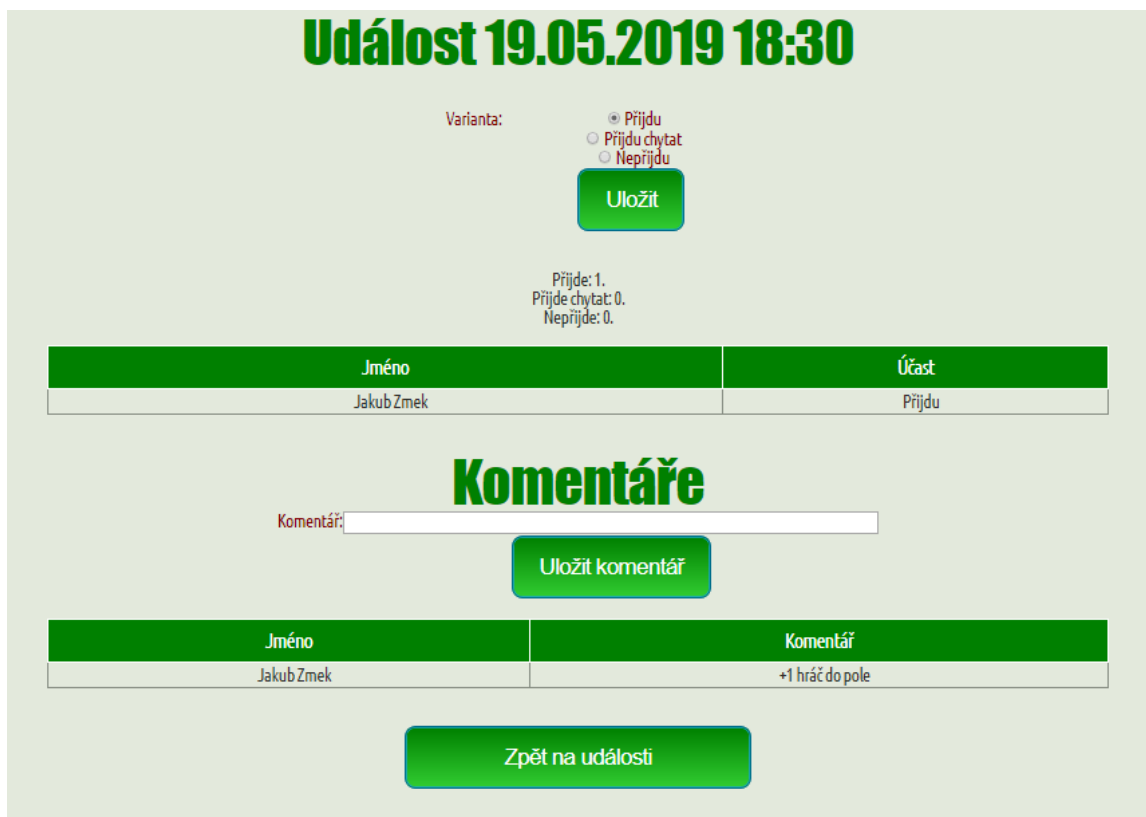
Stiskněte v menu záložku "Docházka". Vypíšou se Vám následující události (tréninky, zápasy). Po stisknutí na vybranou událost se Vám zobrazí formulář pro docházku na událost.

Zatrhněte položku formuláře a potvrďte. V případě, že chcete vložit komentář, napište komentář a vložte tlačítkem.

Tento postup je znázorněn na obrázcích C.6 a C.7.



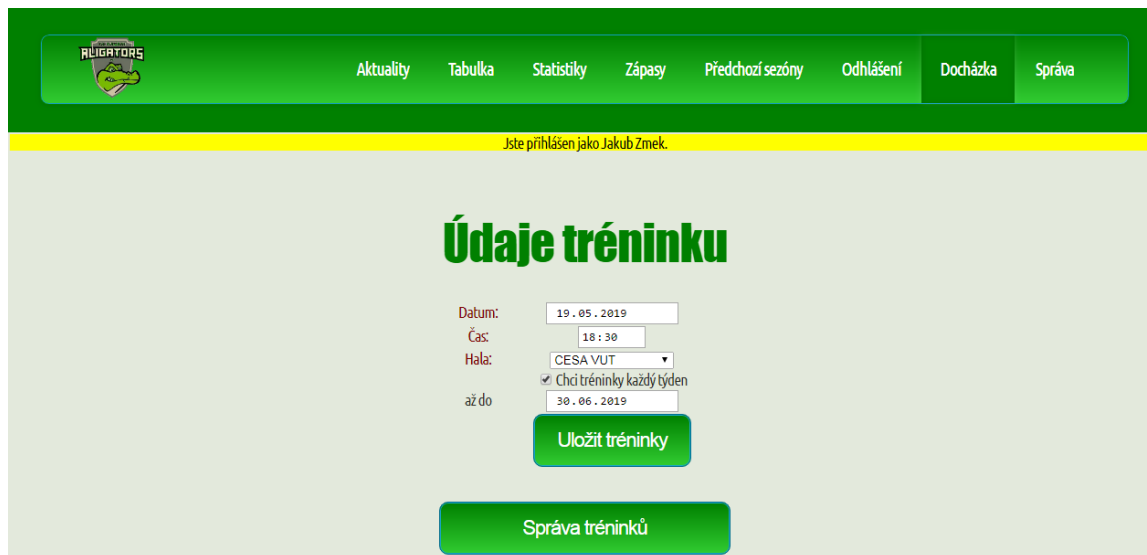
Obrázek C.6: Následující události



Obrázek C.7: Přihlašování na událost

## Vytvoření tréninku

Pokud chcete vytvořit trénink, klikněte na tlačítko "Vytvořit tréninky". Zadejte datum, čas a halu. Pokud chcete vytvořit více tréninků zároveň, zatrhněte následné pole a zadejte mezní datum. Tím se vytvoří tréninky po týdnu od prvního data až po druhé datum. Příklad vyplnění formuláře lze vidět na obrázku C.8.



The screenshot shows the 'Údaje tréninku' (Training Data) form. At the top, there is a navigation bar with the ALIGATORS logo and menu items: Aktuality, Tabulka, Statistiky, Zápasy, Předchozí sezóny, Odhlášení, Docházka, and Správa. Below the navigation bar, a yellow bar indicates the user is logged in as 'Jakub Zmek'. The main content area is titled 'Údaje tréninku' and contains the following fields:

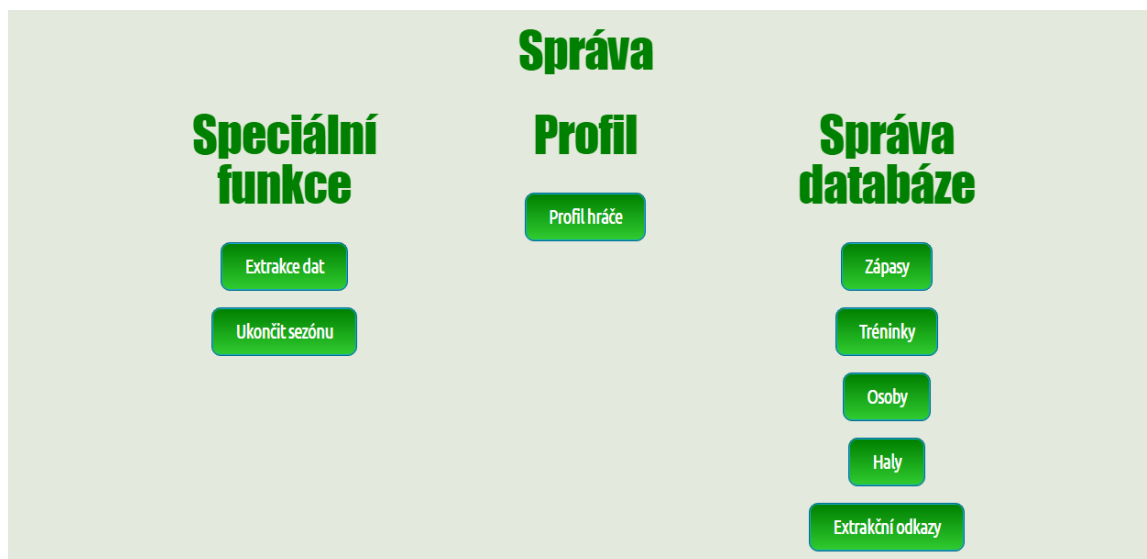
- Datum: 19. 05. 2019
- Čas: 18 : 30
- Hala: CESA VUT
- Chci tréninky každý týden:
- až do: 30. 06. 2019

Below the form are two buttons: 'Uložit tréninky' (Save training) and 'Správa tréninků' (Training management).

Obrázek C.8: Vytvoření tréninků

## C.5 Správa

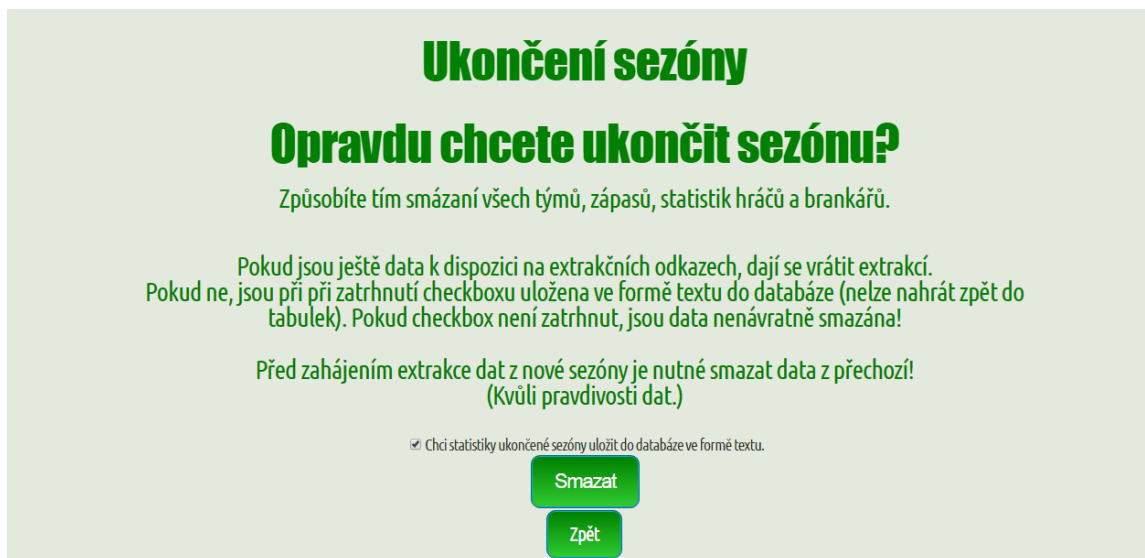
K ovládání správy slouží menu zobrazené na obrázku C.9. Do správy se dostanete pomocí záložky v menu "Správa".



Obrázek C.9: Menu správy

## Speciální funkce

V tomto podmenu je možnost extrahovat data a ukončit sezónu. Extrakci spusťte pomocí kliknutím na tlačítko. Statistiky týmu se automaticky aktualizují. V případě, že skončila sezóna, ukončete sezónu stisknutím odpovídajícího tlačítka. Přečtěte si, jaký má dopad na informační systém. Tento text je možno vidět na obrázku C.10. Pokud je vše v pořádku, potvrďte ukončení sezóny tlačítkem.



Obrázek C.10: Ukončení sezóny

## Profil

V podmenu profil můžete kliknutím na tlačítko upravit údaje o své osobě. Vyplňte (případně upravte) své osobní údaje podle aktuálního stavu. Ačkoliv Vaše heslo můžete vidět jen Vy, kvůli bezpečnosti **vždy používejte různá hesla na různých stránkách!** Ukázku úpravy profilu můžete vidět na obrázku C.11.



Logo: RUGBYSTARS

Menu: Aktuality | Tabulka | Statistiky | Zápasy | Předchozí sezóny | Odhlášení | Docházka | Správa

Jste přihlášen jako Jakub Zmek.

## Profil hráče Jakub Zmek

Rodné číslo:	<input type="text" value="1234567890"/>
Číslo dresu:	<input type="text" value="35"/>
Post:	<input type="text" value="G"/>
Adresa:	<input type="text" value="Voriskova 2"/>
Telefonní číslo:	<input type="text" value="607833917"/>
Email:	<input type="text" value="jakub.zmek@seznam.cz"/>
Login:	<input type="text" value="Zmek"/>
Heslo:	<input type="text" value="0"/>

Obrázek C.11: Úprava údajů

## Správa databáze

V této sekci můžete spravovat obsah jednotlivých tabulek databáze. Stačí kliknout na dané tlačítko. Některé záznamy<sup>1</sup> lze pouze upravovat (zápasy a extrakční odkazy). U dalších lze i přidávat a mazat záznamy. Jakékoliv osobě lze vytvořit nové heslo.

Zmíněné operace jsou znázorněny ikonami. Přidání nového záznamu je umožněno zeleným obrázkem "+" nad tabulkou. Úprava je umožněna stisknutím ikony s tužkou a papírem. Smazání je umožněno červeným křížkem. Nové heslo je možné nastavit pomocí ikony s klíčem a zámkem.

V případě úpravy extrakčních odkazů je zobrazen rovnou formulář s odkazy. V případě, že chcete upravovat jinou tabulku, vyberte požadovaný řádek a klikněte na danou ikonu. Při úpravě, přidání nebo vytvoření nového hesla vyplňte formulář a potvrďte. Smazání je třeba ještě potvrdit tlačítkem.

Ukázku správy osob můžete vidět na obrázcích [C.12](#), [C.13](#), [C.14](#) a [C.15](#).

<sup>1</sup>Záznam je (jednoduše řečeno) řádek tabulky databáze.

## Správa osob



Jméno	Rodné číslo	Číslo dresu	Post	Adresa	Tel.č.	E-mail	Role	Login	Heslo			
Admin							Admin	Admin	****			
František Chmela							Hráč	Hrac	****			
Hráč							Hráč	Hrac	****			
Jakub Gremmel							Hráč					
Jakub Zmek	123456/7890	35	G	Voriskova 2	607833917	jakub.zmek@seznam.cz	Admin	Zmek	****			
Jan Oliva	543454/4530	48	O	Kosmákova 32	608743453	Oliva@seznam.cz	Hráč	Oliva	****			
Jiří Šahan	154357/1321	17	U	Vondrákova 14	609045345	Sahan@seznam.cz	Hráč	Sahan	****			
Ladislav Dufek							Hráč					
Lukáš Gregr	7854132/4515	87	U	Bohunice	774213547	Gregr@seznam.cz	Trenér	Gregr	****			
Lukáš Jerábek	947818/4561	14	O	Horni 22	774125874	JerabekL@seznam.cz	Hráč	JerabekL	****			
Martin Krumpolc							Hráč					
Michal Dubský	453421/4652	28	G	Bozetechova 89	774024458	Dubsky@seznam.cz	Hráč	Dubsky	****			
Ondřej Frank							Hráč					
Ondřej Slezák	453578/3512	96	U	Šlapanice	774323456	Slezak@seznam.cz	Hráč	Slezak	****			

Obrázek C.12: Správa osob



[Aktuality](#)
[Tabulka](#)
[Statistiky](#)
[Zápasy](#)
[Předchozí sezóny](#)
[Odhlášení](#)
[Docházka](#)
[Správa](#)

Jste přihlášen jako Jakub Zmek.

## Údaje osoby

Rodné číslo:

Číslo dresu:

Post:

Adresa:

Telefonní číslo:

E-mail:

Role:

Login:

Obrázek C.13: Úprava osoby

Aktuální menu: [Aktuality](#) [Tabulka](#) [Statistiky](#) [Zápasy](#) [Předchozí sezóny](#) [Odhlášení](#) [Docházka](#) [Správa](#)

Jste přihlášen jako Jakub Zmek.

## Údaje osoby

Rodné číslo:   
 Číslo dresu:   
 Post:   
 Adresa:   
 Telefonní číslo:   
 Email:   
 Role:   
 Login:

Obrázek C.14: Vytvoření nového hesla

Aktuální menu: [Aktuality](#) [Tabulka](#) [Statistiky](#) [Zápasy](#) [Předchozí sezóny](#) [Odhlášení](#) [Docházka](#) [Správa](#)

Jste přihlášen jako Jakub Zmek.

## Opravdu chcete smazat osobu Lukáš Jeřábek?

Způsobíte tím smázaní statistik, které na osobu mohou být navázány!

Obrázek C.15: Smazání osoby

## C.6 Odhlášení

Po skončení prohlížení informačního systému se můžete odhlásit pomocí záložky "Odhlášení" v menu. Odhlášení potvrďte tlačítkem.