

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

System pro správu sportovních lig

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 17. května 2017

Michel Soběhart

Poděkování

Chtěl bych poděkovat vedoucímu diplomové práce, Doc. Ing. Josefu Steinbergerovi, Ph.D. za odborné vedení práce a cenné rady, které mi pomohly při tvorbě diplomové práce. Další velké díky patří vedoucímu futsalové ligy na Západočeské univerzitě, Ing. Jaroslavu Kovaříkovi a vedoucímu Šumavské ligy amatérského hokeje v Klatovech, Luboši Hosnedlovi, za rady při kompletování funkčních požadavků výsledné aplikace. V neposlední řadě také rodině a přítelkyni za podporu a pomoc při studiu.

Abstract

The main goal of the diploma thesis is creation of information system for sports league administration. System should be easily extensible to other sports. Application is primarily designed for administration of futsal league at University of West Bohemia in Pilsen.

In this thesis you can read information about system analyse. This analyse consists of several parts - analyse of existing systems, components useful in implementation and suitable technologies for developing of web applications. The next section provides information about complete database model design and system architecture. Then we find a chapter about implementation of administration interface and informational website. At the end you can read information about final application testing and valorisation of application results. As a final proposal is presented a possible further extension of application.

Abstrakt

Hlavním cílem diplomové práce je vytvoření informačního systému pro spravování sportovních lig. Systém by měl být jednoduše rozšiřitelný pro jiné sporty. Primárně je aplikace určena pro vedení futsalové ligy na Západočeské univerzitě v Plzni.

V této práci se dočtete informace o analýze systému. Ta se skládá z více částí, a to analýzy existujících systémů, jednotlivých komponent využitelných při implementaci a vhodných technologií pro vývoj webových stránek. V další části se nachází informace o návrhu kompletního databázového modelu a architektury systému. Poté nalezneme kapitolu o implementaci administrátorského prostředí a informační webové stránky. Na konci se dozvíte informace o otestování výsledné aplikace a zhodnocení výsledků aplikace. Jako poslední je uveden možný návrh dalšího rozšíření aplikace.

Obsah

1	Úvod	1
2	Motivace	2
2.1	Futsal na Západočeské univerzitě v Plzni	2
2.2	Šumavská liga amatérského hokeje v Klatovech	4
3	Existující systémy a komponenty	5
3.1	Florbal na Západočeské univerzitě	5
3.2	Active Sports	6
3.3	Livesport	7
3.4	Výsledky v1.5.1	10
3.5	Rozpis zápasů	12
3.5.1	Online rozpis zápasů	12
3.5.2	Bergerovy tabulky	13
3.5.3	Kirkmanův algoritmus	14
3.5.4	Frončekovy tabulky	17
3.5.5	Zhodnocení rozpisu zápasů	17
4	Technologie	18
4.1	HTML	18
4.1.1	HTML5	19
4.2	CSS	19
4.3	Bootstrap	20
4.4	PHP	20
4.5	JavaScript	21
4.5.1	Node.JS	22
4.5.2	AngularJS	22
4.6	Java EE	23
4.7	ASP.NET	25
4.8	Ruby on rails	26
4.9	Zhodnocení a výběr technologií	28

5	Potřeby potencionálních uživatelů	30
5.1	Veřejná část webových stránek	30
5.2	Administrátorské prostředí	32
5.2.1	Neregistrovaný uživatel	33
5.2.2	Registrovaný uživatel	33
5.2.3	Hráč	33
5.2.4	Vedoucí týmu	34
5.2.5	Rozhodčí	34
5.2.6	Redaktor	35
5.2.7	Administrátor	35
6	Návrh a implementace systému	37
6.1	Architektura	37
6.2	Model–View–Controller	37
6.2.1	Model	37
6.2.2	Controller	43
6.2.3	View	45
6.3	Databázový model	45
6.4	Tvorba dynamického obsahu	46
6.5	Práva v systému	48
6.5.1	Procedury	62
6.5.2	Triggery	63
6.5.3	Pohledy	64
6.6	Export do PDF	64
6.7	Zabezpečení	65
7	Testování	67
7.1	Funkční testování	67
7.1.1	Vložení a úprava aktuality	67
7.1.2	Vložení a úprava banu	67
7.1.3	Seznam typů pokuty	68
7.1.4	Vložení nového hráče	68
7.1.5	Vložení společenské akce	68
7.2	Testování použitelnosti	69
7.3	Testování bezpečnosti	69
7.4	Zhodnocení testování	69
8	Závěr	70
	Seznam použitých zkratk	72

Seznam literatury	75
Přílohy	78
A Vývojový diagram rozpisu zápasů	79
B Databázový model	80
C UML diagram	81
D Struktura CD	82
E Instalační příručka	84
E.1 Konfigurace aplikace	84
E.2 Překlad aplikace	85
E.3 Databáze	85
F Uživatelská příručka	86
F.1 Informativní část	86
F.1.1 Úvodní stránka aplikace	86
F.1.2 Registrace na stránky	86
F.1.3 Přihlášení na stránky	87
F.1.4 Změna hesla uživatele	88
F.1.5 Přepnutí do administrace ligy	88
F.1.6 Rozpis zápasů	89
F.1.7 Výsledky zápasů	89
F.1.8 Tabulka ligy	90
F.1.9 Kanadské bodování	91
F.1.10 Brankářské statistiky	91
F.1.11 Dynamický obsah	91
F.1.12 Fotogalerie	93
F.1.13 Pokuty	93
F.1.14 Fórum	94
F.1.15 Tým hledá hráče	94
F.1.16 Hráč hledá tým	95
F.1.17 Ke stažení	95
F.1.18 Kontakty	96
F.2 Administrátorské prostředí	96
F.2.1 Přepnutí zpět do informativní části	97
F.2.2 Hlavní strana	97
F.2.3 Přístup zamítnut	98
F.2.4 Seznam uživatelů	99

F.2.5	Seznam týmů	99
F.2.6	Úprava a vložení týmu	100
F.2.7	Seznam hráčů	101
F.2.8	Úprava a vložení hráče	101
F.2.9	Staň se hráčem	102
F.2.10	Seznam rozhodčích	103
F.2.11	Úprava a vložení rozhodčích	103
F.2.12	Seznam typů rozhodčích	104
F.2.13	Úprava a vložení typu rozhodčích	104
F.2.14	Staň se rozhodčím	106
F.2.15	Seznam zápasů	106
F.2.16	Seznam odehraných zápasů	107
F.2.17	Seznam neodehraných zápasů	107
F.2.18	Přihlášení na zápasy	107
F.2.19	Seznam přihlášených hráčů	108
F.2.20	Rozpis zápasů	109
F.2.21	Seznam aktualit	110
F.2.22	Úprava a vložení aktualit	111
F.2.23	Seznam pokut	111
F.2.24	Úprava a vložení pokuty	112
F.2.25	Seznam zaplacených pokut	112
F.2.26	Seznam nezaplacených pokut	113
F.2.27	Seznam typů pokut	113
F.2.28	Úprava a vložení typu pokut	114
F.2.29	Seznam banů	115
F.2.30	Úprava a vložení banu	115
F.2.31	Seznam typů banů	116
F.2.32	Úprava a vložení typu banů	117
F.2.33	Seznam galerií	118
F.2.34	Úprava a vložení galerie	118
F.2.35	Přidání fotografií do galerie	119
F.2.36	Týmový chat	120
F.2.37	Kategorie dynamického obsahu	120
F.2.38	Úprava a vložení kategorie	121
F.2.39	Úprava obsahu kategorie	121
F.2.40	Úprava a vložení podstránky	122
F.2.41	Subkategorie dynamického obsahu	122
F.2.42	Export ligové tabulky	124
F.2.43	Export kanadského bodování	124
F.2.44	Seznam žádostí o tým	124
F.2.45	Úprava a vložení žádosti o tým	126

F.2.46 Seznam žádostí o hráče	126
F.2.47 Úprava a vložení žádosti o hráče	127

1 Úvod

Diplomová práce se zabývá vytvořením systému pro správu sportovních lig. Primárním využitím diplomové práce je vedení Univerzitní futsalové ligy na Západočeské univerzitě v Plzni. Univerzitní futsalová liga je soutěž týmů, která probíhá po celý školní rok. Hrají ji studenti, zaměstnanci i absolventi Západočeské univerzity. Nadřazeným orgánem ligy je Katedra tělesné výchovy a sportu.

Cílem projektu je návrh a implementace takového systému pro správu sportovních lig, který bude možné aplikovat na více sportů, zejména kolektivních. Motivací je usnadnění práce spojené s vedením ligy administrátorovi, největší možná automatizace úkonů, zpřístupnění a prezentace informací všem účastníkům a široké veřejnosti. Postupem času se v projektu objevuje myšlenka rozšíření vedení ligy a možnost přenesení určité části práce na další lidské zdroje.

Tvorba projektu se rozděluje do několika základních částí. První částí je analýza projektu, která se rozděluje do několika skupin. Součástí je analýza technologií pro vývoj podobných systémů a komponent. Dále je nutné získat z potenciálních zákazníků již běžících lig požadavky na systém. Ty udají směr vývoje, a také kompletnější přehled o možnostech i funkcionalitě systému. Následným krokem je navržení splnitelných požadavků z analýzy, vhodná volba technologií pro vývoj a architektury tak, aby byla dále v budoucnu rozšiřitelná. Následuje návrh systému a implementace projektu. Nakonec je nutné otestovat systém a nasadit.

2 Motivace

Původní nápad na vytvoření aplikace pro administraci sportovních lig jsem dostal od vedoucího florbalové ligy na ZČU. Ve stejné době přišel podobný nápad také od vedoucího Šumavské ligy amatérského hokeje v Klatovech. Webové stránky, které jsou momentálně provozovány pro vedení florbalové ligy jsou nedostačující a složitě spravovatelné. Dále jsem se dozvěděl, že Katedra tělesné výchovy a sportu (dále jen KTS) potřebuje systém, který by bylo možné využít nejen pro jeden sport, ale s menšími úpravami také pro další sportovní odvětví. Primární využití je však pro Univerzitní ligu futsalu.

Diplomová práce má za hlavní úkol ulehčení práce, především ušetření času lidem, kteří se starají o správný chod sportovních lig. Na ZČU jsou to z největší části pedagogové a záukolovaní studenti. Dalším úkolem je spojit dohromady lidi zajímaví se o daný sport. Jsou to nejen stabilně hrající lidé, ale i nováčci z řad studentů, kteří by se chtěli ligy také zúčastnit. Zároveň poskytují informace o ligách pro širokou veřejnost.

Pro kompletnější představu výhod souvisejících s úsporou času a práce poukážu na dvě reálně běžící ligy, do kterých jsem se jako hráč nebo jako administrativní pomocník zapojil.

2.1 Futsal na Západočeské univerzitě v Plzni

„Projekt Univerzitní ligy ZČU začal vznikat na podzim roku 2014 za spolupráce nově vzniklého spolku Školní liga futsalu Plzeň, spolku Plzeň pro sport a Katedry tělesné výchovy a sportu. Cílem projektu je podporování mimostudijních sportovních aktivit studentů Univerzity, rozvoj talentu a zdraví u mladých lidí. Prevence proti drogám a kriminalitě prostřednictvím pohybu. Univerzitní liga ZČU je určena pro studenty a zaměstnance univerzity v Plzni. Nově vzniká Univerzitní liga ZČU, tedy zmenšená verze fotbalu, který se hraje 4+1, se čtyřmi hráči v poli a brankářem. Většina prestižních univerzit v ČR již podobný projekt několik let úspěšně provozuje (UK, Masarykova Brno, Olomouc, a další). Vzhledem k tomu, že fotbal/futsal je v České Republice stále nejoblíbenějším sportem, jsme přesvědčeni, že v Plzni nalezne mezi studenty řadu nadšených příznivců at' v podobě hráčů či jejich fanoušků.“ [18]

90% docházky na zápasy, aby získal nárok na udělení zápočtu. Kvůli tomu si vedoucí vytvořil další tabulku, kterou ručně doplňuje.

Je však nutné podotknout, že již taková stránka dodala lize na sledovanosti a přiměla mnoho studentů se také zapojit. Každým rokem dochází k nárůstu o nové hráče a také týmy. V roce 2016/2017 se již soutěže zúčastňuje 10 týmů čítajících okolo 100 hráčů.

2.2 Šumavská liga amatérského hokeje v Klatovech

Druhým příkladem je vedení Š.L.A.H. v Klatovech, kde jsem se dříve účastnil jako hráč a vypomáhal s vedením ligy. Soutěž byla vytvořena pro amatérské hokejisty z Klatov a okolí. Ovšem postupem času docházelo ke zmírňování pravidel, a tak v soutěži začali nastupovat hráči z vyšších lig. V sezóně 2015/2016 hrálo ligu 8 týmů, celkově přes 150 hráčů. V poslední sezóně došlo k rozdělení na ryze amatérskou část a část poloprofesionální, která získá status okresního přeboru.

Pro veškerou administrativu ligy se využívalo zápisů o utkání, které procházely analýzou a přepisováním do mnoha dalších papírových tabulek (tabulka ligy, tabulka střelců a asistentů, tabulka trestných minut týmů a další). Tento systém se s přibývajícími týmy a zápasy stal neudržitelným a muselo dojít k zásadním změnám.

První změnou byla automatizace úkonů, se kterou jsem měl tu možnost vypomáhat. Společně se správcem ligy jsme tvořili v softwarovém programu typu Microsoft Excel tabulky, které uměly dopočítávat mnoho informací. Snažili jsme se o snížení duplikace při zadávání informací, a zároveň o co největší přívětivost a přehlednost pro uživatele. Nevyhnuli jsme se však nutnosti některé informace duplikovat do více různých míst.

Další změnou, která následovala, bylo snížení statistických informací. Mělo to však za následek, že při výsledném vyhodnocení ročníku ligy, kdy bylo nutné veškeré statistiky spočítat, bylo potřeba hodně času pro jejich vytvoření. To byl jeden z mých prvních impulsů, který mě dovedl k nápadu vytvořit složitější systém.

3 Existující systémy a komponenty

V této kapitole si uvedeme podobné existující systémy, které jsem zanalyzoval zejména z hlediska funkcí daného systému. Snažil jsem se také o analýzu technologického řešení, pokud to bylo možné. V této kapitole si popíšeme jednotlivé součásti využitelné v mé výsledné aplikaci. Každá z komponent může být podmnožinou funkčnosti výsledné aplikace. Pro začátek je potřeba zmínit, že neexistuje mnoho aplikací, které řeší podobný problém. Pověštinou jsou ligy, které mají vlastní webové stránky a vše okolo vedení řeší ručně. Mnohdy tak administrátor ligy musí přímo přepisovat veškerý HTML kód.

3.1 Florbal na Západočeské univerzitě

Při analytické části florbalového systému jsem čerpal z bakalářské práce Davida Koška [10], který webové stránky tvořil. Dále jsem díky vedoucímu Katedry tělesné výchovy a sportu Mgr. Pavlu Červenkovi získal přístup k administraci ligy. Mohl jsem si vyzkoušet část systému. Tím jsem získal obecnější přehled o jeho běhu a funkcionalitě.

První analyzovanou částí je databázová vrstva. V práci byl výběr zúžen na dvě databáze – MySQL a PostgreSQL. Nakonec bylo rozhodnuto pro využití MySQL databáze. Databázový model obsahuje 15 tabulek. Pro vedení takové ligy je tabulek velmi málo, jelikož systém by měl být komplexnější. To se také potvrdilo během prvního roku běhu aplikace. Systém byl rozšířen o dalších 13 tabulek, které byly nezbytné pro další funkcionalitu. Pro komunikaci dalších částí systému s databází je používána knihovna DIBI.

Webové stránky jsou vytvořeny pomocí HTML a CSS. Dále je využito v menším množství programovacího jazyku JavaScript, např. pro rozbalovací menu. Jako skriptovací jazyk se využívá PHP. Administrátorské prostředí je založeno na architektuře MVC (Model-View-Controller) a programováno také pomocí PHP.

Základní funkcionalita systému byla v bakalářské práci vytvořena. Ovšem již z popisu funkcí a diagramů případů užití je jasné, že komplexita systému je nedostačující. Stav systému na konci roku 2016 se nedá srovnávat se stavem odevzdané práce. Systém se posunul ve funkčnosti, bylo vytvořeno mnoho

dalších možností. Začal se využívat částečně bootstrap pro responsivní design.

Během této doby se ukázalo, že rozšiřitelnost systému nebyla velmi dobrá. Nové funkce jsou do systému těžko dodělávány a chybí zde jakákoliv abstrakce. Mnoho věcí se muselo doplnit dalšími formuláři, a tak již není možné zadávat informace na jednom místě. Další nevýhodou je vkládání statistik. Statistiku se vkládají tlačítka přidat gól a přidat asistenci. Bohužel zde není žádná odezva na kliknutí na tlačítka. Proto se v jednom roce také stalo, že výsledné kanadské bodování nesedělo s ostatními částmi statistik. Není zde provedena žádná kontrola např. na počet gólů vzhledem k zápasům. Zápas může skončit 10:10 a zapsán pouze jeden gól a naopak může skončit 0:0 a hráči mohou získat body za góly a asistence.

Při rozhovoru s administrátorem ligy bylo dosaženo závěru, že hlavní požadavek, na který by se měl brát zřetel, je automatizace úkonů. Pokud je možné zadat data na více míst, vzrůstá tím časová náročnost pro vedení soutěže, a také nepřehlednost a nekonzistence dat. Občas zapomenuto vložit data na všechna místa, a tím dochází k tomu, že nejsou všude stejná. Další možností by mělo být zadávání gólů a asistencí, aby bylo možné mít uložené statistiky o utkání.

3.2 Active Sports

Active Sports je systém pro administraci sportovních lig. Vytváří ho firma Active, která je na trhu více než deset let. Firma pracovala s počtem přes 1000 sportovních lig, klubů a sdružení. Zabývají se registračním softwarem a softwarem pro správu lig. Pomáhá získat více účastníků, rychle spravovat různé sportovní organizace. Dále podporuje kempy, komunitní centra, školy a podnikové korporace. O technologiích vývoje se nedá mnoho zjistit, a proto se zaměřím na funkcionalitu daného systému. V této části jsem čerpal z webových stránek společnosti [22].

Systém umožňuje administraci lig a jejich management, což znamená vytváření sezón, divizí a týmů. Dále distribuce týmových soupisek a informací elektronicky pomocí notifikací. Slouží také k organizaci rozhodčích a jejich přiřazení na konkrétní zápas. Možnost přiřazení third party¹ nástroje. Kontrola registrací do ligy, možnost nastavení věku, velikosti týmu nebo přestupu

¹Third party – opakovatelně využitelná komponenta, která lze využít v různých programech

hráče mezi týmy. Také je možné shromáždit a ověřit data dle určitých pravidel. Umožňuje nábor a řízení dobrovolníků. Systém je přizpůsobitelný mnoha sportům a jeho automaticky aktualizovány po zadání bodů.

Vytvoření týmu a přiřazení hráčů lze vytvořit v průvodci budování týmu. U týmu lze určit kritéria pro hráče, jako je škola, úroveň dovedností nebo předchozí týmy. Existují zde různé funkce pro přiřazování hráčů. Například si tým přiřazuje hráče sám a nebo se hráči přihlašují do týmu, kde musí dojít k potvrzení vedoucím týmu. Poskytuje informace o týmu, vytisknout soupisky týmu, informace o týmu, hráčích nebo trenérech.

Další možností je generování rozpisů, at' už automaticky s různými parametry (kolik her za týden, jaké hodiny, čas, atd.) nebo ručně. Vytvoření sezónního plánu optimalizovaného pro tým. Je zde možnost rušit nebo změnit zápasy kvůli nepřízní počasí. Poté přijde upozornění všem hráčům a trenérům. Lze snadno vytvořit a naplánovat turnaj v nástroji pro plánování turnajů.

Je možné v systému generovat seznamy kontaktů a adresy pro skupiny hráčů, personálu nebo dobrovolníků. Možnost ukládat dokumenty, jako například zdravotní prohlídky, vysvědčení a další. Výše postavení lidé potom mohou dokumenty analyzovat. Lze snadno komunikovat se svým týmem prostřednictvím emailu.

Systém je hodně rozsáhlý. Tomu také odpovídá jeho komplexní využitelnost. Je možné doplnit určité části vlastními komponentami, což dodává systému další rozšiřitelnost. Je nutné dodat, že systém běží přes 10 let, takže je velmi dlouho otestován a rozšiřován zkušenostmi uživatelů systému.

3.3 Livesport

Livesport s.r.o je česká společnost založená v roce 2006. Od té doby se jim podařilo vybudovat jednu z největších platforem pro sběr a zobrazování sportovních statistik v reálném čase pod globálním brandem Flashscore². Při sportovních utkáních je cílem aplikace dodávat výsledky do celého světa. Nejedná se pouze o skóre zápasu, střelce gólu, ale také o podrobné informace, jako jsou například ofsajdy nebo fauly při fotbalovém klání. Navíc přidává informace o kurzech na jednotlivé zápasy nebo souboje.

²<http://www.flashscore.com/>

Hlavní složkou platformy jsou data. Data se musí efektivně sbírat ze zápasů. K tomu slouží lidé od externí firmy, kteří za pomoci televize a videostreamů sledují dané zápasy živě a zapisují všechny informace do systému. V celé platformě existuje mnoho stránek v mnoha zemích, které jsou různými variantami a mutacemi. Stejně tak existuje mnoho mobilních aplikací. Díky tomu má Livesport přes 11 milionů uživatelských přístupů denně. [8]

Za veškerou podporou reálného dodání dat na trh je široká infrastruktura. Livesport vlastní aktuálně (k 29.2.2016) přes 300 serverů, přičemž třetina z nich je virtuálních. Platforma je ověřená největšími zápasy jako například finále Ligy mistrů. Při takovém zápase je obrovské zatížení, až 375 000 dotazů za vteřinu. Mezi používanými technologiemi je Docker nebo Varnish (cache s vlastními modifikacemi). První vrstva je frontendová cache, druhá vrstva je backendová cache, další jsou aplikační servery (Apache web server, PHP), databáze MySQL i MongoDB. [8]

Při analýze jsem se zaměřil na funkční stránku, rozdíly statistik mezi sporty, a další. Zejména jsem se zaměřil na futsal a fotbal, které spolu blízce souvisí a jsou vzorem pro futsalovou ligu. Dále jsem zanalyzoval ostatní sporty a vyhledával množinu statistik, které se musí ukládat při daném zápase. Obrázky jsou screenshoty z oficiálních stránek zdroje [19].

Sportem číslo jedna mezi komunitou je fotbal. Pokud se jedná o velmi sledované zápasy, statistiky ukládané při jednom zápase jsou zcela kompletní. Na úvodní stránce jsou uvedeny veškeré informace o průběhu utkání, v jaké minutě se stala jaká událost (padl gól, žlutá karta, červená karta, penalta, střídání), informace o rozhodčím a počtu diváků. V další záložce s názvem „Statistiky utkání“ můžeme vidět informace o držení míče, celkovém počtu střel, počtu střel na bránu soupeře, zblokováných střelách, přímých a rohových kopech, ofsajdech, vhazováních, zásazích brankářů, faulech, žlutých a červených kartách a počtech přihrávek, at' úspěšných, tak i celkových. V další záložce můžeme vidět sestavy obou týmů včetně hráčů, kteří čekají na lavičce náhradníků. V neposlední řadě také statistiky jednotlivých hráčů v utkání a LIVE komentář. Zde je možno vidět, kolik práce je nutné vynaložit pro takovou analýzu utkání. Na obrázcích 3.1 a 3.2 je pro představu uvedeny detail a statistiky zápasu.

Dalším velice sledovaným sportem je hokej. Jsou zde velké rozdíly mezi zápasy, které patří do nejsledovanějších lig světa jako NHL, KHL, a které patří do méně známých lig, jako je třeba nejvyšší soutěž z Kazachstánu (je

Přehled zápasu	Statistiky zápasu	Sestavy	Statistiky hráčů	LIVE komentář
1. poločas				
14' Correa J. (Neproměněná penalta)	1 - 0			
25' Sarabia P. (Escudero S.)				
2. poločas				
55' Carrico D. Lenglet C.			Musa A. Gray D. 58'	
62' Correa J. (Jovetic S.)				
63' Iborra Correa J.	1 - 1		(Drinkwater D.) Vardy J. 73'	
77' Escudero S. (Podražení)				
90+2' Carrico D. (Držení)			Albrighton M. Amartey D. 88'	
Informace o zápase				
Rozhodčí: Turpin C. (Fra)				
Počet diváků: 38 834, Stadion: Estadio Ramón Sánchez Pizjuán (Sevilla)				

Obrázek 3.1: Detail fotbalového zápasu

Přehled zápasu	Statistiky zápasu	Sestavy	Statistiky hráčů	LIVE komentář
Zápas	1. poločas	2. poločas		
68%			Držení míče	32%
22			Střely celkem	8
8			Střely na branku	3
10			Střely mimo branku	2
4			Zblokované střely	3
8			Přímé kopy	6
8			Rohové kopy	1
0			Ofsejdy	0
27			Vhazování	25
2			Zásahy brankářů	6
6			Fauly	9
2			Žluté karty	0
654			Přihrávky celkem	234
579			Úspěšné přihrávky	171

Obrázek 3.2: Statistiky zápasu

ukládán pouze konečný výsledek). Rozdíl od fotbalových zápasů je v určitých statistikách, které do hokeje nepatří. Příkladem mohou být žluté a červené karty, které se v hokeji neudělují, nebo rohové kopy. Naopak přibývají jiné informace, které jsou nejčastěji spojené s vyloučením. Jsou to góly v přesilovce, oslabení, ubráněné oslabení, využití a nevyužití přesilovky.

Sportem, o který nám jde především, je futsal. Statistik ohledně futsalových zápasů je ukládáno velice málo. Nejedná se o příliš sledovaný sport a tak většinou lidem stačí pouze vědět konečný výsledek nebo výsledek v poločase zápasu. Není zde informace o průběhu utkání, zda-li padla nějaká karta (žlutá nebo červená), byla kopána penalta nebo který střelec dal gól.

Analýza těchto stránek mi velice pomohla k vytvoření komplexnějšího přehledu o možných sportech, jejich potřebách a rozdílech. Pokud se podíváme na florbalové statistiky, je zde vidět stejný rozdíl jako mezi futsalem a fotbalem. Hodně statistik je podobných ale nejsou u méně sledovaného sportu ukládány.

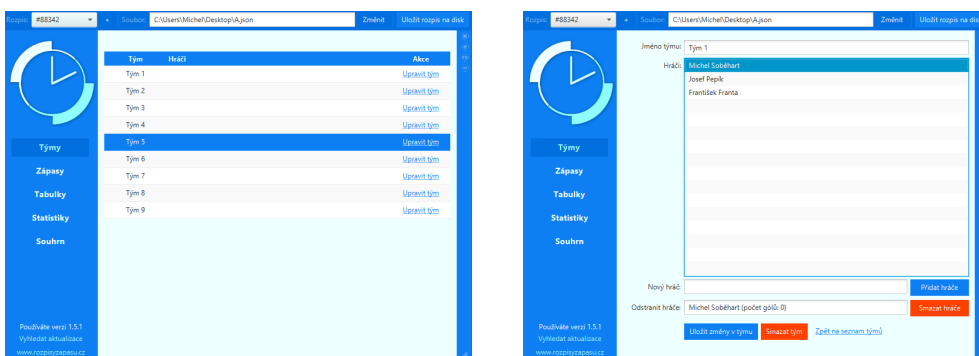
3.4 Výsledky v1.5.1

Jako alternativu pro sběr výsledků k zápasu jsem provedl analýzu aplikace Výsledky v1.5.1 vydané 27.6.2015 panem Zdeňkem Drahošem. Free verzi aplikace si lze bezplatně stáhnout z <http://www.rozpisyzapasu.cz/vysledky/>. Software není vytvořen jako webová stránka, ale jako desktopová aplikace. Základ aplikace byl vytvořen v rámci diplomové práce [2].

Po spuštění aplikace musíme vložit rozpis ve formátu JSON (JavaScript Object Notation). Ten si vytvoříme na stránce s rozpisem utkání, o které se můžete dále dočíst v kapitole 3.5.1. Načtení rozpisu zpřístupní možnost základního vedení ligy nebo turnaje. V menu nalezneme 4 záložky - týmy, zápasy, tabulky, statistiky. Ty si nyní blíže projdeme.

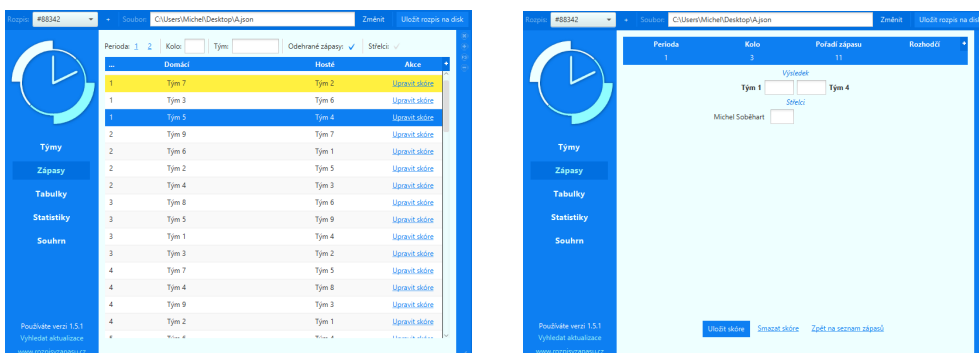
V záložce „Týmy“ se nachází seznam týmů společně s hráči, kteří za daný tým hrají. Je zde možnost úpravy týmu, kdy se program přepne na obrazovku, ve které lze změnit název týmu, přidat nebo odebrat dané hráče. Můžeme zde vidět, že nejsou vedeny žádné další informace ohledně hráčů, ale pouze jména. Ukázky z obou obrazovek jsou vidět na obrázku 3.3

Další záložkou jsou „Zápasy“. Zde je zobrazen seznam zápasů s mož-



Obrázek 3.3: Obrazovky pro seznam a úpravu týmů

ností filtrování. To lze provést např. určením periody zápasů, výběrem názvu týmu nebo kola. Můžeme také zápas upravit nebo smazat. Úprava zápasu poté obsahuje dvě pole s počtem gólů jednotlivých týmů. Pod výsledkem zápasu jsou vypsaní jednotliví hráči týmu. Je možné nastavit, kolik dal daný hráč gólů (popř. měl bodů) v zápase. Jako mínus vidím to, že nemohou být zadány asistence. Naopak jako plus sledávám jednoduché zadávání počtu gólů. Jednotlivé obrazovky je možné vidět na obrázku 3.4.



Obrázek 3.4: Obrazovky pro seznam a úpravu zápasů

Další záložkou je „Tabulka ligy“. V tabulce jsou uloženy všechny informace, které se ze zadaných dat dají vypočítat (počet zápasů, počet výher, remíz a proher, skóre a body). Poslední důležitá záložka zahrnuje tabulku statistik, která obsahuje jméno hráče, tým za který hraje a vstřelené góly. Statistika jsou i pro brankáře. Nakonec poskytuje statistiky ohledně celého rozpisu (kolik se odehrálo utkání, kolik jich ještě zbývá odehrát).

3.5 Rozpis zápasů

Rozpis zápasů patří k důležitým součástem každé ligy, pokud administrátor nechce tvořit zápas po zápasu sám. V této sekci si představíme stránku s rozpisy zápasu a také algoritmy, kterými lze zápasy rozlosovat.

3.5.1 Online rozpis zápasů

Jedním z nejznámějších a nejpoužívanějších stránek používaných k rozlosování zápasů je <http://www.rozpisyzapasu.cz/>. Jak je zmíněno dříve, tento rozpis je úzce spojený s aplikací Výsledky, blíže popsanou v kapitole 3.4. Rozpisy zápasů byly vytvořeny v rámci diplomové práce [2].

Prvním typem rozlosování, které je možné generovat, je tzv. každý s každým. Je možné si navolit tři vlastnosti, které může splňovat rozpis zápasů:

- Obyčejný rozpis (Kirkmanův algoritmus, Bergerovy tabulky).
- Týmy hrají stejný počet zápasů na každém hřišti (max 4 hřiště).
- Týmy čekají stejnou dobu mezi zápasy (jen pro lichý počet týmů).

Pro rozpis každý s každým (round robin) se musí zadat informace o počtu týmů a jejich jménech. Dále počet hřišť a počet kol, které spolu týmy budou hrát. Lze také vypočítat časy zápasů. K tomu je potřeba zadat délku jednoho odehraného zápasu, přestávku mezi jednotlivými utkáními a data, ve kterých se hraje a kdy je volno. Také je možné přidat rozhodčí k jednotlivým zápasům.

Druhým typem je play-off pavouk. Play-off je vyřazovací systém, kdy se odehraje méně zápasů, než když hraje každý s každým. V jedné sérii (čtvrtfinále, semifinále, finále) mezi sebou mohou soupeřit týmy ve více zápasech. Opět při tvorbě se musí zadat počet týmu a jména, dále kolik zápasů se bude hrát v jedné sérii a jestli se uskuteční zápas o bronzovou příčku.

Další možností je vytvoření konferenčního turnaje. Zde je možnost vytvoření dvou konferencí, kde se budou výsledky započítávat stejným způsobem jako u typu každý s každým. Jediným rozdílem je, že jsou vytvořeny dvě tabulky podle toho, v jaké konferenci se daný tým nachází. Navíc se musí zadat

jména konferencí. Maximální možný počet konferencí jsou dvě. Zde mi chybí například možnost, aby týmy hrály mezi sebou v konferenci vícekrát nebo hráli jednou mezi všemi a jednou jen v konferenci. Vedlo by to ke snížení počtu zápasů, pokud by nezbýval čas a bylo hodně týmů (např. akademický rok na univerzitě).

3.5.2 Bergerovy tabulky

Mezi neznámější a nejpoužívanější prostředek pro sestavování rozvrhů sportovních zápasů patří Bergerovy tabulky. Autor tabulek není znám, avšak jejich sestavení se připisuje šachistovi Bergerovi. Primárně byly vytvořené pro šachové turnaje. Zde se nerozdělují zápasy doma a venku, protože se hraje jeden turnaj na jednom místě. Rozlišuje se však, jestli hráč začíná s bílými nebo černými figurkami. Proto mají tabulky široké využití v mnoha sportech. Systém spočívá v tom, že každý tým si před začátkem jednoho turnaje vylosuje číslo od 1 do počtu týmů. Pokud je lichý počet týmů, přidá se jedno číslo fiktivního týmu. Pokud někdo hraje s fiktivním týmem, má v daném kole pauzu. Kapitola převzata z [13].

Realizace algoritmu podle Bergerovo postupu si znázorníme na příkladu šesti týmů. Počet týmů budeme označovat jako $2n=6$, takže $n=3$.

1. Nejvyšší číslo n (v našem případě 6) dáme do levého spodního rohu.
2. Číslo 1 dáme nad nejvyšší číslo do pravého horního rohu.
3. Postupně umístíme všechny další čísla tak, že doplňujeme týmy od nejmenšího po největší proti směru hodinových ručiček. Čísla zapisujeme tak, že od kroku 2 vždy vynecháme jedno volné políčko.
4. Když se dostaneme k číslu n , začneme opět v pravém horním rohu, kde je volné políčko. Tímto dostaneme rozpis pro první kolo.
5. V dalším kroku budeme točit všechny týmy po směru hodinových ručiček o jednu pozici. Jediné pole, které zůstane celou dobu na svém místě je pole v pravém dolním rohu, tedy n .

Pro rozpis na domácím a hostujícím hřišti doplníme informaci, že týmy s lichým číslem hrají jako hosté v případě, že tým protihráče má vyšší liché

1. kolo	4	2	1
	3	5	6
2. kolo	3	4	2
	5	1	6
3. kolo	5	3	4
	1	2	6
4. kolo	1	5	3
	2	4	6
5. kolo	2	1	5
	4	3	6

Tabulka 3.1: Rozpisu po jednotlivých kolech dle Bergerových tabulek

číslo nebo nižší sudé číslo. Týmy se sudým číslem hrají venku s nižším lichým číslem a vyšším sudým týmem. Jedinou výjimkou je tým číslo 2, který začíná jako domácí a poté se každé další kolo střídá.

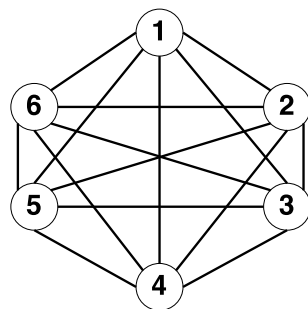
3.5.3 Kirkmanův algoritmus

Kapitola byla vytvořena dle vědeckého článku [9]. V práci nalezneme řešení problému rozlosování zápasů. Využívá se tzv. Kirkmanův algoritmus pro dva problémy - základní a rozšířený. V případě rozlosování ligy je potřeba rozlišovat mezi domácími a hostujícími týmy. Proto jsem se zaměřil na rozšířený problém, který tuto možnost akceptuje.

Problém tedy je: máme dán sudý počet týmů n a $n-1$ hracích kol. Cílem algoritmu je vytvořit rozpis utkání tak, aby každý tým hrál s každým právě jednou, každý tým hrál přesně jeden zápas za kolo. Posledním požadavkem je, aby se každému týmu co nejvíce střídaly zápasy doma a venku.

Při řešení problému je nutné si uvědomit, že neexistuje takový rozpis utkání, který neobsahuje žádné zlomy v pořadí zápasů doma a venku. To lze ověřit pomocí důkazu. Pokud dva týmy budou mít stejnou posloupnost domácích a venkovních zápasů, tak se nikdy nemohou setkat proti sobě (nelze, aby hráli současně proti sobě dva týmy, kde každý z nich je domácí tým a nebo naopak). Je tedy zřejmé, že pro každý rozpis utkání pro n týmů má každý z $n-2$ týmů alespoň jeden zlom domácích a nebo venkovních zápasů. V praxi to znamená, že jeden tým hraje dva zápasy za sebou doma nebo venku.

Pro grafické zobrazení takového algoritmu rozlosování namodelujeme problém na datovou strukturu (graf). Jednotlivé vrcholy představují týmy a hrany mezi nimi představují jednotlivé zápasy. Pro jednodušší představu využijeme počet týmu $n = 6$. Každý tým musí v daném kole hrát s každým, proto bude graf úplný³. Základní graf tedy bude vypadat jako na obrázku 3.5.



Obrázek 3.5: Základní úplný graf

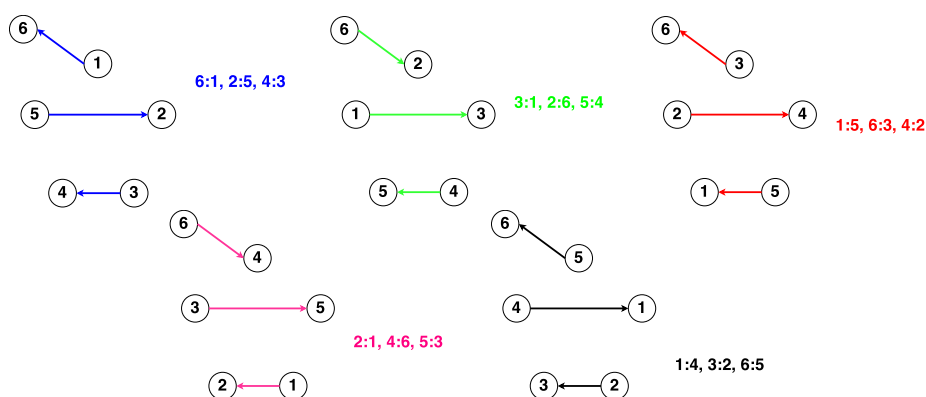
Princip algoritmu spočívá v grafové metodě zvané edge coloring (obarvování hran). Popíšeme si tedy posloupnost kroků, kterými lze takový algoritmus popsat:

1. vytvoříme pravidelný mnohoúhelník z vrcholů $1, 2, \dots, n-1$ a vrchol n umístíme vedle mnohoúhelníku na levou horní stranu,
2. vrchol n spojíme s horním vrcholem v mnohoúhelníku,
3. každý zbývajících vrchol spojíme s vrcholem protějším, který má stejnou výšku. Orientace hran se otáčí s každou jednou výškou protějším vrcholů,
4. obarvíme tyto hrany jednou barvou (celkově jich budeme potřebovat $n-1$),
5. všechny vrcholy kromě vrcholu n otočíme v mnohoúhelníku cyklicky proti směru hodinových ručiček, vždy o jednu pozici. Vrchol n zůstává na stejné pozici,
6. spojíme vrcholy stejně jako v krocích 2 a 3, pouze u kroku číslo 2 otočíme orientaci hrany oproti předchozí iteraci,

³úplný graf - takový, kde každý vrchol je hranou spojený se všemi ostatními vrcholy v grafu

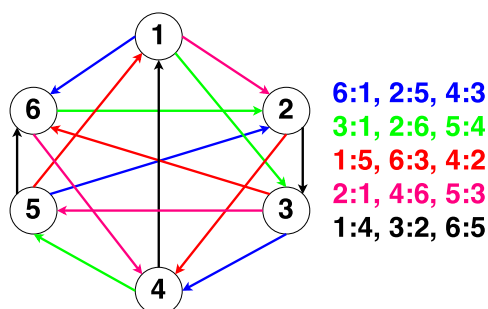
7. vzniklé hrany obarvíme další barvou,
8. opakujeme kroky 5, 6, 7 pro zbývající barvy 3, 4, ..., $n-1$.

Nyní stačí jen zavést pravidlo, kdy je tým domácí a kdy host. Pokud je rozpis jednokolový, můžeme si vybrat jakoukoliv možnost. Pokud je vícekolový, je nutné pravidlo střídat. V našem případě si určíme, že pokud šipka u hrany vede od týmu i k týmu j , tak domácím týmem se stává tým j . Jednotlivé kroky iterace můžete tedy vidět na obrázku 3.6.



Obrázek 3.6: Kroky algoritmu rozpisu zápasů pro 6 týmů

Výsledný rozpis zápasů pomocí grafu a obarvování hran je znázorněn na obrázku 3.7.



Obrázek 3.7: Rozpis zápasů pro 6 týmů

Posledním problémem s nutností specifikace je rozlosování při lichém počtu týmů. Řešení je jednoduché. Při lichém počtu týmů se přidá další imaginární tým, který představuje roli přestávky v daném kole.

3.5.4 Frončekovy tabulky

Tuto metodu vyvinul český profesor matematiky Dalibor Fronček na univerzitě v Minnesotě (University of Minnesota Duluth). Metoda byla využita v české hokejové extralize v roce 2000/2001. Nyní se využívá v rozlosování soutěže české fotbalové ligy od roku 2001/2002. Algoritmus vylepšuje předchozí popsané metody. Frončekovy tabulky zajišťují takový rozpis, kdy žádný tým nekončí poslední dva zápasy v sezóně na stadionu soupeře. Jako předchozí algoritmy také zajišťuje pravidelné střídání domácího a hostujícího týmu. Další výhodou je možnost, aby týmy z blízkých sídel nebo regionů nehrály domácí zápasy v jednom kole, zohlednění nevhodných podmínek na stadionech (např. začátek jarní části sezóny, stadion je potřeba pro jiné účely). Další velkou výhodou v dnešní době je ovlivňování korupce, protože zápasy není možné predikovat. Systém je totiž proměnlivý k posloupnostem zápasů. Více o této metodě se můžete dočíst ve vědeckém článku [4].

3.5.5 Zhodnocení rozpisu zápasů

Požadavkem na rozpis je hlavně rozlosování domácí-hosté, které poskytují všechny tři algoritmy. Dalším požadavkem s hlediskem na implementaci je jednoduché nasazení na jakýkoliv počet týmů - to také splňují všechny tři. Důvodem proč jsem vybral Kirkmanův algoritmus je, že rozlosování je velice jednoduše naprogramovatelné a článek tuto možnost popisuje velmi podrobně. Dalším důvodem je, že pro náš případ jsou Frončekovy tabulky zbytečně komplikované. Zaměření se týká menších lig, které se povětšinou hrají na jednom stadionu nebo více stadionech v blízkém okolí. Proto odpadá nevýhoda posledních dvou zápasů v sezóně na hřišti soupeře.

4 Technologie

Výběr technologie pro implementaci aplikace je důležitý a má mnoho faktorů, které je třeba zhodnotit. Důležitost takových faktorů se může pro vývoj různých softwarů lišit, každý má své přednosti. Pro mou diplomovou práci jsem sestavil posloupnost hodnot, které budu analyzovat:

1. Znalost programovacího jazyka.
2. Rozšiřitelnost aplikace.
3. Komerční využití aplikace, nasazení.
4. Dokumentace jazyka.
5. Dostupnost frameworků a komponent.
6. Časová náročnost implementace.
7. Náklady na provoz.

Z toho je vidět, že přiřazuji vysokou prioritu znalosti programovacího jazyka, kdy se budu moct dále soustředit na vývoj namísto učení nových věcí. Snažím se o aplikaci, která bude robustní a dobře rozšiřitelná. Jelikož je aplikace tvořena primárně pro ZČU, která disponuje vlastními servery, není vysoká důležitost přikládána komerčnímu využití a s tím spojenými náklady na provoz. Vývoj webové aplikace se však neskládá pouze z jednoho programovacího jazyka.

4.1 HTML

HTML (Hypertext Markup Language) je programovací jazyk, který se používá k vytvoření většiny webových stránek. Od počátku webu bylo vytvořeno několik verzí HTML. O vývoj jazyka se zasloužila organizace s názvem W3C¹. Hlavní verzí jazyka byla HTML 4.01 vydaná v prosinci 1999. V lednu roku 2000 byla přidána striktnější pravidla a vznikl XHTML (Extensible Hypertext Markup Language). HTML je značkovací jazyk, který popisuje strukturu

¹W3C – World Wide Web Consortium

dokumentu. Jazyk je charakterizován elementy, které mají specifický název a také atributy. [15]

4.1.1 HTML5

Poslední specifikací jazyka HTML je verze HTML5. Finálně byla vydána 28.10.2014. Přináší nové prvky, které jazyk oživují. Nové elementy v HTML5 jsou:

- **sémantické pro strukturu stránky** – speciální elementy určující strukturu stránky, definuje hlavičku, patičku, určení sekcí, navigací,
- **sémantické pro text** – elementy pro čas nebo značky,
- **webové formuláře** – velké množství podtypů elementu input,
- **audio, video a pluginy** – elementy pro přehrání muziky, videa a jiných zdrojů,
- **canvas** – plátno pro kreslení,
- **podpora jiných jazyků než angličtiny.**

Některé nové funkce fungovaly již dříve, pouze ne byly součástí jazyka a budou bez problémů zobrazitelné (například drag and drop). A také jsou funkce, které mohou být ignorovány. Například jde o zaoblení rohů, které na nových prohlížečích funguje, ale na starších budou rohy zobrazeny hranaté (zobrazení z CSS3). Poslední skupinou může být například uživatelský vstup pro datum. Nelze zobrazit v prohlížeči Google Chrome. [11]

4.2 CSS

CSS (zkratka z Cascading Style Sheets) je jazyk popisující prezentaci webových stránek. Stejně jako HTML, tak i CSS navrhla standardizační organizace W3C. Bylo vytvořeno více verzí jazyka, CSS1, CSS2, v roce 2011 vznikla verze CSS2.1. V dnešní době se již plně používá zatím poslední verze kaskádových stylů, a to CSS3. Ta se blíže pojí s HTML5.

Umožňuje nastavení barev, písma, uspořádání elementů, a mnoho dalšího. V neposlední řadě také přizpůsobit prezentaci různým typům zařízení. Český název pro CSS jsou kaskádové styly. Kaskádové se nazývají proto, že je možné je vrstvit. Technologie není závislá pouze na HTML, ale lze použít s jakýmkoliv značkovacím jazykem založeným na XML. [15]

4.3 Bootstrap

Bootstrap je open source framework pro rychlejší a snadnější vývoj webových stránek. Framework byl vyvinut Markem Ottem a Jacobem Thorntonem ve Twitteru. Poprvé byl zveřejněn jako open-source produkt v srpnu 2011 na GitHub². V červnu 2014 se stal projektem s číslem 1 na GitHub.

Bootstrap obsahuje HTML a CSS šablony založené na typografii, formulářích, tlačítkách, tabulkách, navigacích, obrázkových karuselích. Dále obsahuje volitelné JavaScript pluginy. Jednou z velkých výhod je možnost jednoduché tvorby responzivních designů³ webových stránek.

4.4 PHP

PHP (zkratka pro PHP: Hypertext Preprocessor) je open source skriptovací jazyk pro vytváření dynamických a interaktivních webových stránek. PHP kód se nachází přímo v HTML stránkách, takže je snadné a rychlé vytvořit dynamický obsah. Syntaxe je inspirována dalšími programovacími jazyky, jako jsou C, Perl a Java. K běhu je potřeba PHP parser (CGI nebo modul serveru), webový server a webový prohlížeč. Může běžet na všech hlavních operačních systémech (Linux, MacOS, Windows, mnoho variant UNIX systémů a další). PHP je server-side, což znamená, že programy nebo skripty běží na straně serveru. Skript je pomocí PHP engineu pokaždé, když běží. [1]

Proces běhu skriptu je následovný:

1. požadavek ze strany klienta o webovou stránku,

²GitHub – webová služba, která slouží k hostování zdrojového kódu projektů a pro open-source projekty je zdarma.

³Responzivní design – automaticky se přizpůsobující design na různých obrazovkách

2. webový server rozpozná, že jde o PHP skript a dává příkaz PHP enginu o zpracování a spuštění skriptu,
3. po dokončení skriptu odesílá zpět HTML stránku do webového prohlížeče.

Zde si popíšeme výhody a nevýhody jazyka:

- + podpora většiny hostingů,
- + náklady na provoz,
- + velké množství funkcí a využitelných kódů,
- + multiplatformnost,
- + vhodné i na malé projekty,
- + licence,
- + podporuje mnoho databázových systémů,
 - chybí nástroj pro debugging,
 - kontext aplikace je vytvářen při každém požadavku,
 - nejednotnost a nekonzistence zdrojového kódu,
 - použití výjimek.

4.5 JavaScript

JavaScript byl původně představen v roce 1995 jako způsob, jak přidat programy do webových stránek v prohlížeči Netscape Navigator. Od té doby je možné jazyk využít ve všech významných webových prohlížečích. Název nemá nic společného s programovacím jazykem Java. Standardem je ECMAScript, který je nazvaný po ECMA organizaci. Obě dvě jména jsou pro stejný jazyk a máme možnost je zaměnit. Mezi lety 2000 a 2010 byla nejrozšířenější verzí ECMAScript 3. Nejen webové prohlížeče, ale také některé databáze, například MongoDB nebo CouchDB, využívají JavaScript pro skriptování a dotazování do databáze. [7]

Výhody a nevýhody jsou:

- + nižší zátěž serveru,
- + interaktivita,
- + možnost oddělení skriptu do externího souboru,
- + dovednosti možné pouze na straně klienta,
- + rychlost,
- lze ho zakázat v prohlížeči,
- špatný přístup k souborům,
- ukládání dat,
- možná nekompatibilita mezi prohlížeči.

4.5.1 Node.JS

Node.js je platforma postavená na Google Chrome JavaScript Engine V8. Platforma byla publikována Ryanem Dahlem v roce 2009. Node.js představuje open source, multiplatformní prostředí pro vývoj rychlé a škálovatelné síťové aplikace. Používá neblokující I/O model. Aplikace jsou výkonné, ideální pro datově náročné aplikace běžící v reálném čase, které mohou být spuštěny v distribuovaných zařízeních. [16]

Node.js je asynchronní a řízený událostmi. Znamená to, že všechny knihovny API jsou neblokované, takže server nečeká na vrácení dat. Server přesune volání na API a pomocí notifikačního mechanismu a událostí se snaží získat odpověď. Používá jednovláknový model s opakováním událostí, ale je dobře škálovatelný. Nepochází zde k bufferování dat. Je publikován pod MIT licencí. [16]

4.5.2 AngularJS

AngularJS je strukturální framework použitelný pro dynamické webové stránky. Jako šablonu využívá HTML a umožňuje rozšířit HTML syntax. Díky bindování dat a vkládání závislostí (dependency injection) slouží framework pro zjednodušení psaní kódu. Na rozdíl od Node.js využívá skriptování na straně klienta.

Vývoj aplikací v AngularJS je jednodušší pro vývojáře, protože využívá vyšší úroveň abstrakce. Byl postaven na bázi CRUD (Create, Read, Update, Delete). Není vhodný pro hry a GUI editory. Kapitola převzata z webu [14].

4.6 Java EE

Java EE (zkratka z Java Enterprise Edition) se objevila na trhu koncem roku 1990. Jazyk Java se stal robustní softwarovou platformou pro tvorbu enterprise aplikací. Slouží k tvorbě webových, transakčních nebo bezpečných distribuovaných aplikací. Stejně jako Java Standard Edition (Java SE) poskytuje API pro zpracování kolekcí, tak Java EE poskytuje standardní způsob pro zpracování transakcí, zasílání zpráv nebo perzistentní vrstvu.

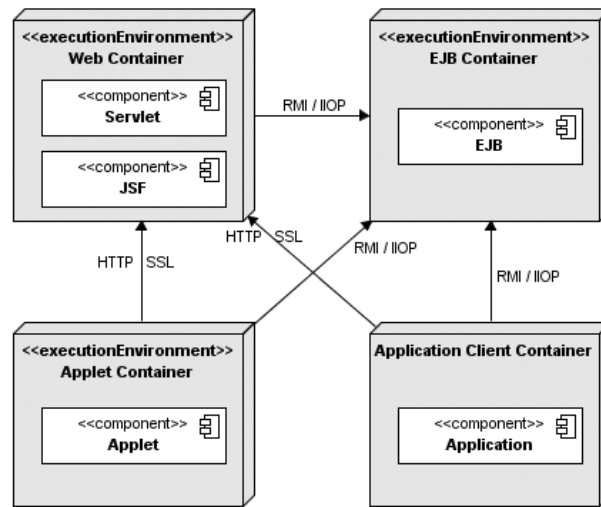
První verze Java EE (původně známá jako J2EE) byla zaměřena na distribuované komponenty. Od té doby se softwarové aplikace musely přizpůsobit novým technickým řešením, jako jsou webové služby SOAP a REST. Platforma se vyvinula v reakci na tyto technické potřeby tím, že poskytuje různé způsoby práce prostřednictvím standardních specifikací. Java EE se v průběhu let stal jednodušším, snadněji použitelným, přenosnějším a integrovatelnějším řešením. Je velmi dobře zdokumentován a má vysokou základnu uživatelů a s tím spojenou vysokou podporu.

Java EE je sada specifikací realizovaných v různých kontejnerech. Kontejner poskytuje služby jako je řízení životního cyklu, dependency injection a další. Tyto komponenty používají jasně definované protokoly, kterými komunikují s dalšími složkami systému. Na obrázku 4.1 můžete vidět logické vztahy mezi kontejnery a komunikační protokoly. Kapitola byla se změnami převzata z [5].

Můžeme vidět, že Java EE platforma definuje čtyři typy komponent:

1. **Applety** – grafické uživatelské rozhraní spouštěné ve webovém prohlížeči. Používají Swing API, která poskytuje silné uživatelské rozhraní.
2. **Aplikace** – programy spouštěné na klientovi. Většinou je to GUI nebo batchprocessing⁴ program, který má přístup ke všem ostatním komponentám střední vrstvy.

⁴provedení série příkazů bez manuálního vstupu (neinteraktivní)



Obrázek 4.1: Architektura platformy Java EE

3. **Webové aplikace** (servlety, servletové filtry, webové posluchače událostí, Java Server Pages (JSP), Java Server Faces (JSF)) – jsou prováděny ve webovém kontejneru a reagují na požadavky HTTP z webových klientů. Servlety také podporují SOAP a REST. Také mohou obsahovat odlehčený Enterprise Java Beans (EJB).
4. **Podnikové aplikace** (Enterprise Java Beans, Java Message Service, Java Transaction API, asynchronní volání, RMI/IOOP) jsou prováděny v kontejneru EJB.

Výhody a nevýhody tohoto jazyka jsou:

- + rozšiřitelnost,
- + mnoho využitelných frameworků,
- + multiplatformnost,
- + existence design patternů pro vývoj,
- + jednoduchý popis tříd pomocí anotací (např. tvorba MVC architektury)
- + platformu podporuje mnoho velkých firem,
- + velká základna uživatelů (např. řešení chyb na Stackoverflow⁵),

⁵<http://www.stackoverflow.com/>

- + dokumentace,
- + jednoduchá implementace MVC architektury (Spring MVC),
 - náročnější implementace na malé projekty,
 - náklady na provoz,
 - méně podporujících hostingů.

4.7 ASP.NET

ASP.NET je webová platforma, která poskytuje veškeré služby potřebné pro vytvoření enterprise serverové webové aplikace nebo webových služeb. Tento jazyk je postaven na .NET Frameworku, proto všechny funkce z frameworku jsou využitelné také pro aplikace v ASP.NET. Aplikace mohou být napsané v jakémkoliv jazyce kompatibilním s Common Language Runtime (CLR), například Visual Basic nebo C#. [3]

Předchůdcem ASP.NET je programovací jazyk ASP. Jednou z hlavních výhod oproti klasickému ASP je změna z interpretovaného kódu do zkompi-lovaného kódu. To umožňuje webovým aplikacím mít větší výkon a rychlejší běh. Nejprve je kód sestaven kompilátorem vysokoúrovňového jazyka, který generuje Microsoft Intermediate Language (MSIL) kód. Ten převezme .NET runtime a generuje nativní strojový kód. Webové aplikace jsou tedy spouš-těny pomocí rozhraní .NET Frameworku a ne pomocí operačního systému. Aplikace se tím stávají bezpečnějšími. [3]

ASP.NET byl představen roku 2002, měl jediný programovací model s názvem Web Forms. Ten umožnil vývojářům vytvářet webové aplikace v krátkém čase. Společnost vyvíjející tento programovací jazyk je Microsoft. Webové stránky jsou běžně reprezentovány dvěma soubory, s příponou .aspx definující uživatelské rozhraní a s příponou .aspx.cs nebo .aspx.vb (v závislosti na jazyku serveru), které uživatelské rozhraní obsluhují. ASP.NET obsahuje tři technologie pro tvorbu webových aplikací: [3]

- ASP.NET WebForms
- ASP.NET Web Pages

- ASP.NET MVC

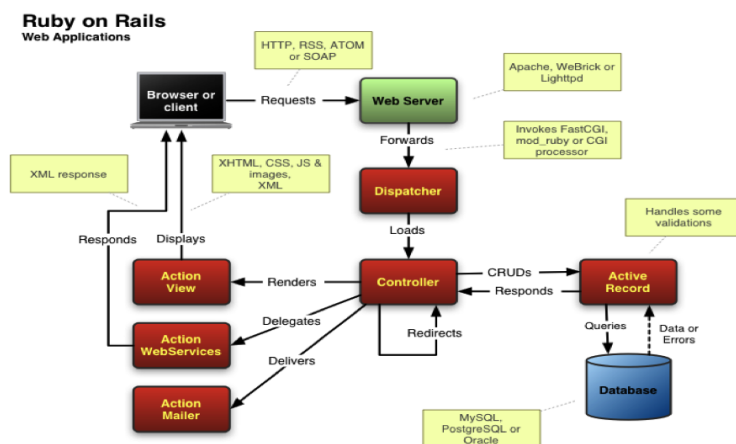
Nyní uvedeme některé výhody a nevýhody programovacího jazyka. Ty jsou:

- + hodně chyb možné zachytit při vývoji,
- + možnost programovat v jakémkoliv jazyce z platformy .NET,
- + rozšiřitelnost,
- + ovládací prostředí a prvky podobné jako OS Windows,
- + existence design patternů pro vývoj,
- + jednoduchá implementace MVC architektury,
- složité naučení,
- malá podpora hostingů,
- složité programování OS mimo Windows,
- náročnější implementace a použitelnost pro malé projekty.

4.8 Ruby on rails

Ruby on Rails (zkráceně také jen Rails) je framework pro vývoj webových aplikací napsaných v programovacím jazyce Ruby. Ruby on Rails byl oficiálně uvolněn v červenci 2004 jako open source. První verze byla vydána v prosinci 2005. Rails byl jeden z prvních jazyků, který byl schopný provádět REST architektonický styl pro strukturování aplikace.

Jednou ze základních myšlenek stojících u vzniku frameworku byla „konvence má přednost nad konfigurací“. Dalším podstatným principem je Don't Repeat Yourself (zkráceně DRY), kdy se programátor snaží pomocí správné úrovně abstrakce eliminovat duplicitu. Základem je návrhový vzor Model-View-Controller. Dalším důležitým prvkem je tzv. scaffolding. Umožňuje v brzkých fázích vývoje bez větších nároků na programátora vyzkoušet spolupráci hlavních komponent. V Ruby on Rails se využívají tzv. gemy. Jsou



Obrázek 4.2: Architektura frameworku Ruby On Rails [20]

to knihovny, které jsou soběstačné a umožňují dodat další funkcionalitu do systému. Na obrázku 4.2 můžete vidět architekturu celého frameworku.

Výhody a nevýhody jazyka jsou:

- + velká škála informačních blogů, diskuzních fór a IRC kanálů,
- + dobrá dokumentace,
- + vhodné i pro malé projekty,
- + multiplatformnost,
- + konvence před konfigurací, výhoda při přebírání aplikace,
- + rozšiřitelnost, gemy,
- + generátory kódu,
- + jednoduchá implementace MVC architektury,
- velmi malá podpora hostingů,
- nutnost Linuxu.

4.9 Zhodnocení a výběr technologií

Po analýze technologií jsem vybral dva kandidáty na programovací jazyk pro vývoj aplikace. U všech ostatních byly nalezeny určité věci, které nebyly pro můj systém vhodné. U JavaScriptu (resp. u frameworků NodeJS a AngularJS) hrála proti zejména absolutní neznalost jazyka, i když v poslední době jeho využití velice vzrůstá. Proto bude využit pouze na určité jednoduché úkoly na webové stránce. ASP.NET byl také z programátorského hlediska novinkou, líbila se mi rozšiřitelnost a jednoduché využití MVC architektury. Ovšem při porovnání s Java EE, se kterou jsem se již setkal, a také kvůli nižší základně uživatelů jsem se rozhodl tento jazyk rovněž vyloučit. Ruby on Rails má dvě hlavní nevýhody, které mě odradily. Podpora hostingů je velmi malá i proti jiným málo podporovaným jazykům a použití Linuxu pro programování.

Rozhodoval jsem tedy nakonec mezi jazykem PHP a Java Enterprise Edition. Dle předem navržených hodnot, podle kterých jsem programovací jazyk vybíral, dopadl výběr takto:

1. Znalost programovacího jazyka – **Java EE**.
2. Rozšiřitelnost aplikace – **Java EE**.
3. Komerční využití aplikace, nasazení – **PHP**.
4. Dokumentace jazyka – **Java EE**.
5. Dostupnost frameworků a komponent – PHP má mnoho využitelných funkcí a Java EE hodně komponent, nedokáží posoudit.
6. Časová náročnost implementace – Nebyl jsem schopen rozhodnout. Aplikace je, řekl bych, středně velká, proto nejspíše časově by se rychleji dala vyvinout v **PHP**.
7. Náklady na provoz – **PHP**.

Každý z jazyků získal v jednotlivých hodnotách po třech výhodách proti jazyku druhému. Avšak vzhledem ke zvoleným prioritám jsem se rozhodl vybrat jazyk Java EE jako primární jazyk pro naprogramování diplomové práce.

Pro databázovou část jsem se rozhodl využít databáze MySQL. Hlavním důvodem je znalost systému a jeho funkcionalita, která plně dostačuje na tvorbu databáze.

5 Potřeby potencionálních uživatelů

Analýza potřeb potencionálních uživatelů systému probíhala ve čtyřech fázích. V první fázi probíhala analýza potřeb z již hotových systémů, a také z vlastních nápadů. Předpokladem pro tvorbu byly již zmíněné zkušenosti s vedením lig.

V další fázi jsem využíval schůzek se zákazníkem, vedoucím futsalové ligy, Mgr. Jaroslavem Kovaříkem, který mi pomohl dovést stav analýzy dle funkční stránky do cílového bodu. Zřetel byl přednostně brán na požadavky, které jsou vhodné přímo pro vedení futsalové ligy. Zejména se jednalo o možnosti administrace se zaměřením na jednoduchost.

Ve třetí fázi jsem specifikoval požadavky za pomoci vedoucího Šumavské ligy amatérského hokeje, Luboše Hosnedla. Vzhledem k jeho dlouholetým zkušenostem a množství produktivních nápadů jsem se rozhodl dát co největší možnost lidem k připojení na naše stránky. Příkladem mohou být komentáře k zápasům, týmový chat, možnost přidání fotografií k zápasům a další.

V poslední fázi došlo z největší části ke slučování nasbíraných požadavků. Bylo nutné některé z požadavků a dobrých nápadů odstranit. Byly to pouze ty, které nevyšly ze schůzek od zákazníka.

5.1 Veřejná část webových stránek

Veřejná část popisuje stránky, které budou viditelné v rámci našeho systému pro všechny uživatele. Jedná se o informativní část systému. Dozvíme se zde informace o lize, průběhu aktuální sezóny, statistikách z aktuální sezóny. Ostatní stránky, které mají zobrazovací privilegia, jsou součástí administrátorského prostředí. Velké množství požadavků na prezenční vrstvu aplikace vznikly v první fázi analýzy, další byly specifikovány se zákazníkem. Zde můžete vidět výpis těchto stránek společně se základním a stručným obsahem:

- **Úvodní stránka** – první stránka při vstupu na web. Bude obsahovat aktuality, obrázky z různých akcí, základní tabulku ligy (pořadí, název týmu a body), prvních 10 míst ve statistice hráčů (pořadí, jméno hráče

a body).

- **Propozice** – informace o lize, jak se liga hraje, pravidla, startovné.
- **Rozpis zápasů** – rozpis zápasů ve formě tabulky, možnost uložení zápisu o utkání.
- **Výsledky zápasů** – výsledky zápasů ve formě tabulky, kdo hraje s kým a daný výsledek. Kliknutím na daný zápas je otevřen detail zápasu.
- **Detail zápasu** – výsledek zápasu, loga týmů, vstřelené góly a asistence.
- **Detail týmu** – základní statistiky týmu.
- **Detail hráče** – statistiky hráče společně s informacemi o uživateli.
- **Detail brankáře** – statistiky brankáře společně s informacemi o uživateli.
- **Tabulka** – kompletní tabulka ligy. Kliknutím na daný tým v tabulce se objeví detail týmu.
- **Kanadské bodování** – kompletní statistiky hráčů. Kliknutím na daného hráče se objeví detail hráče.
- **Statistiky brankářů** – kompletní statistika brankářů. Kliknutím na daného brankáře se objeví detail brankáře.
- **Kontakty** – kontakty na vedoucí ligy, vedoucího KTS, adresa haly ZČU, kde se hrají zápasy společně s mapou.
- **Pokuty** – nezaplacené pokuty týmů.
- **Ke stažení** – dokumenty ke stažení.
- **Fórum** – fórum pro všechny registrované uživatele.
- **Galerie** – galerie fotografií.
- **Reprezentace** – informace o reprezentaci ZČU ve formě článků o daných akcích.

5.2 Administrátorské prostředí

Administrátorské prostředí je hlavní částí systému a je možné do něj vstoupit až po registraci uživatele. Snahou je co nejvíce zjednodušit a zautomatizovat úkony. Nejlépe najít veškerou posloupnost úkonů, kdy uživatel zadá veškeré hodnoty pouze jednou a všechny informace se sami zaktualizují. Další nutností je povolení přístupu pouze uživatelům, kteří mohou na danou stránku přistupovat. Uživatel v administrátorském prostředí přistupuje k databázi, může přidávat, editovat a mazat mnoho důležitých informací.

V systému se vyskytuje několik různých rolí uživatelů. Tyto role jsou:

1. neregistrovaný uživatel,
2. registrovaný uživatel,
3. hráč,
4. vedoucí týmu,
5. rozhodčí,
6. redaktor,
7. administrátor.

Každý uživatel má svoje možnosti v administraci ligy. Zpočátku nebyla nutnost tolik rolí, pokud by administrátor chtěl ligu vést sám. Po zohlednění časové náročnosti na vedení ligy jedním člověkem jsem se pokusil o vytvoření dalších rolí. Tyto role mají méně práv než administrátor, ale mohou velkou mírou pomoci při administraci ligy. Nyní si popíšeme jednotlivé role společně s jejich právy v systému.

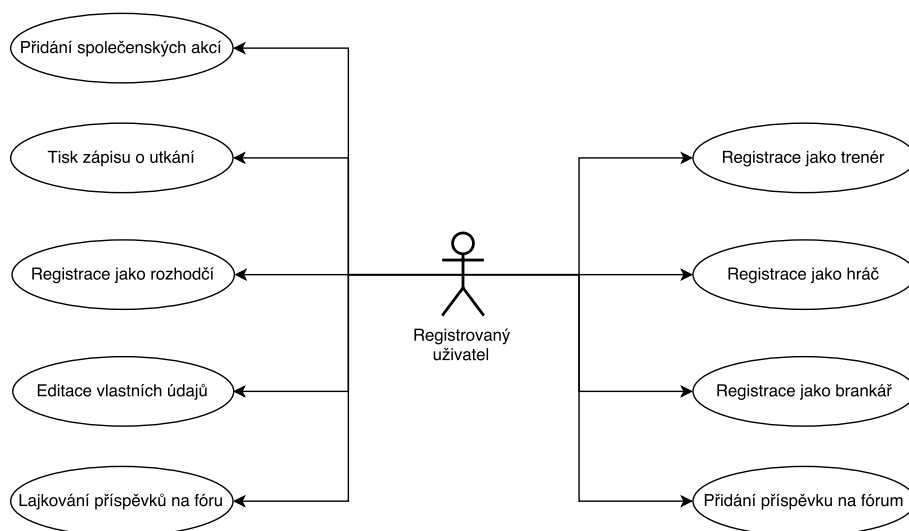
Důležitou možností je slučování rolí. Po registraci na webové stránky je možnost získat více rolí v systému zároveň. Uživatel může být například hráč a také redaktor. Například ve florbalové lize je jeden člověk hráčem, vedoucím týmu a zároveň administrátorem. Proto role budou slučovány akumulativně.

5.2.1 Neregistrovaný uživatel

Neregistrovaný uživatel má přístup k veřejné části aplikace kromě fóra, které je pouze pro registrované uživatele. Jeho jediným případem užití je tisk zápisů o utkání, proto jsem se rozhodl nevytvářet pro něj diagram.

5.2.2 Registrovaný uživatel

Uživateli, který je pouze registrován a není dále specifikován, přibývá další požadovaná funkcionalita oproti neregistrovanému. Registrovaný uživatel může navíc přidat příspěvek do sekce, kde hráč hledá tým. Může se dále registrovat jako hráč, brankář, rozhodčí nebo trenér. Na obrázku 5.1 je možné vidět diagram případů užití.

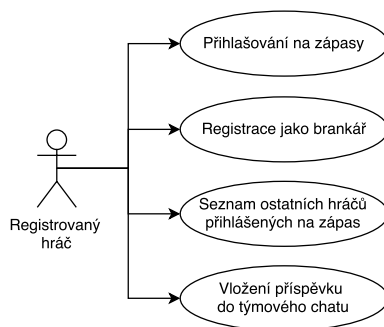


Obrázek 5.1: Diagram případů užití pro roli registrovaného uživatele

5.2.3 Hráč

Hráčem se uživatel stane, pokud se přihlásí na stránkách jako hráč a dojde ke schválení vedoucím týmu. Druhou možností je přidání hráče na soupisku vedoucím týmu nebo administrátorem. Dědí práva od registrovaného uživatele. Tím nabývá možnosti přihlášení k daným zápasům jeho týmu, možnost přidat příspěvek do týmového chatu, a také rozesílat emaily v rámci svého

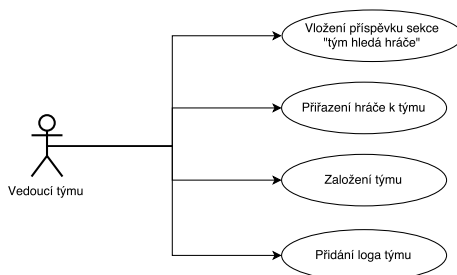
týmu. Také může přidávat příspěvky do sociálních akcí. Hráč jiná práva nepotřebuje, protože všechno ostatní za něj obstará vedoucí týmu nebo administrátor. Pokud se hráč nebude chtít natolik zapojit do dění ligy, je jako běžný registrovaný uživatel. Specifická práva pro tuto roli jsou na obrázku 5.2.



Obrázek 5.2: Diagram případů užití pro roli hráče

5.2.4 Vedoucí týmu

Vedoucím týmu se stane uživatel, pokud ho zvolí administrátor ligy. Vedoucí poté může založit vlastní tým, přiřadit mu logo nebo přidat na soupisku další hráče. Dále může přidat příspěvek do sekce „tým hledá hráče“. Práva vedoucího jsou přímo spojená s administrací svého týmu. Můžete je vidět na obrázku 5.3.



Obrázek 5.3: Diagram případů užití pro roli vedoucího týmu

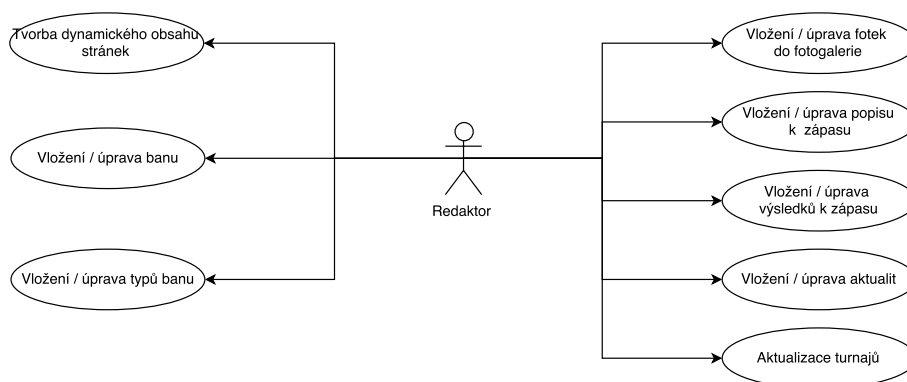
5.2.5 Rozhodčí

Rozhodčím se může stát pouze registrovaný uživatel, který se sám jako rozhodčí přihlásí. Rozhodčí má možnost se přihlásit na pískání zápasů, které si

sám vybere. Rozhodčí nemá v systému další práva.

5.2.6 Redaktor

Redaktor je uživatel, který má za úkol vypomáhat administrátorovi s vedením celé ligy. Redaktor může vypomáhat s úpravou informací ve veřejné části stránek a tvorbou dynamického obsahu. Dále se může starat o zápasy - přidávat popisek a výsledky. Po domluvě s vedoucím futsalové ligy může také zápasy vkládat a měnit data jejich odehrání. Dále může vkládat nebo upravovat aktuality, bany. Redaktorem se stává člověk, kterého administrátor považuje za důvěryhodného. Naopak tato role nemusí být zastoupena, pokud není potřeba. Práva redaktora můžete vidět na obrázku 5.4.

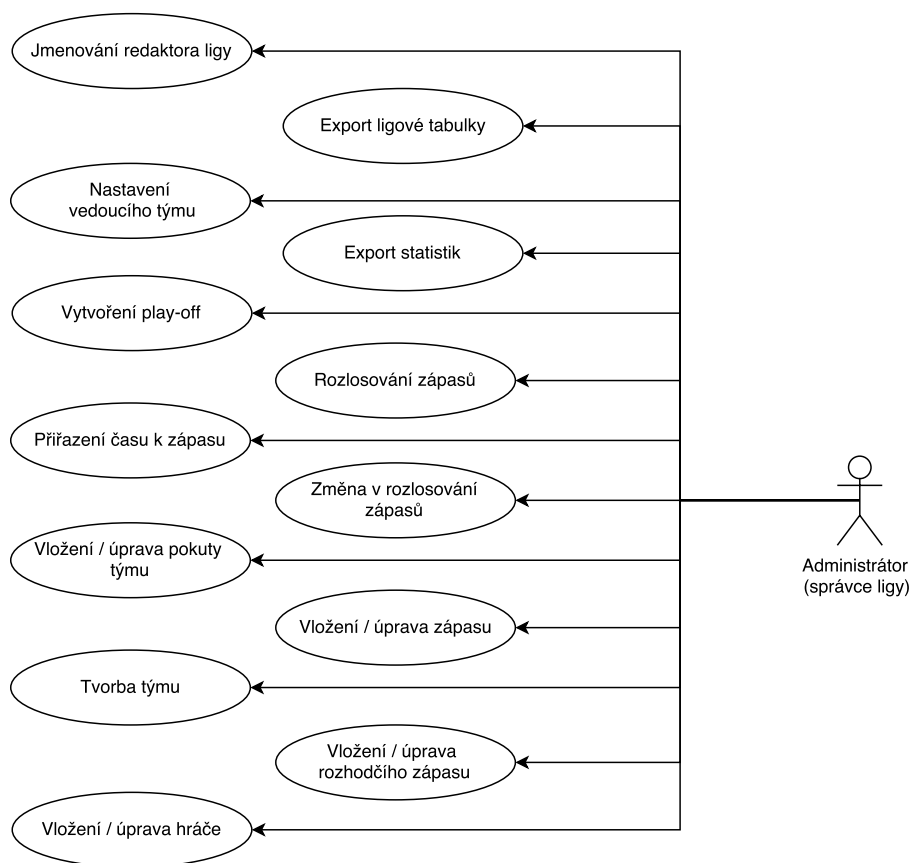


Obrázek 5.4: Diagram případů užití pro roli redaktora

5.2.7 Administrátor

Administrátor je role, která má veškerou kontrolu nad vedením ligy. Může upravovat veškeré informace v databázi za pomoci formulářů. Dále sledovat statistiky ligy v rychlém přehledu. Má veškerá práva jako všichni uživatelé s výjimkou registrovaného hráče, kde nemůže přidávat příspěvky do týmových chatů a přihlašovat se na zápasy. Má možnost využít funkce rozpisu zápasů, přiřadit nebo upravovat nový čas zápasu a také zápas vložit. Může upravovat rozpis zápasů a přidávat rozhodčí k daným zápasům. Vkládá pokuty týmům nebo je upravuje. Dále slouží ke schvalování vedoucího týmu nebo může tým vytvořit sám. Má možnost měnit práva některých uživatelů. Může zabanovat uživatele za nevhodné chování na globálním fóru. Další jeho

možností je jmenovat redaktora ligy. Administrátorská práva najdete na obrázku 5.5.



Obrázek 5.5: Diagram případů užití pro roli administrátora

6 Návrh a implementace systému

V této kapitole je popsán návrh architektury systému a části implementace. Dále jsou zde informace o návrhu struktury datového modelu databáze společně s triggerami a procedurami, které řeší problém statistiky na databázové úrovni. Návrh rozpisu zápisů je uveden formou vývojového diagramu v příloze A.

6.1 Architektura

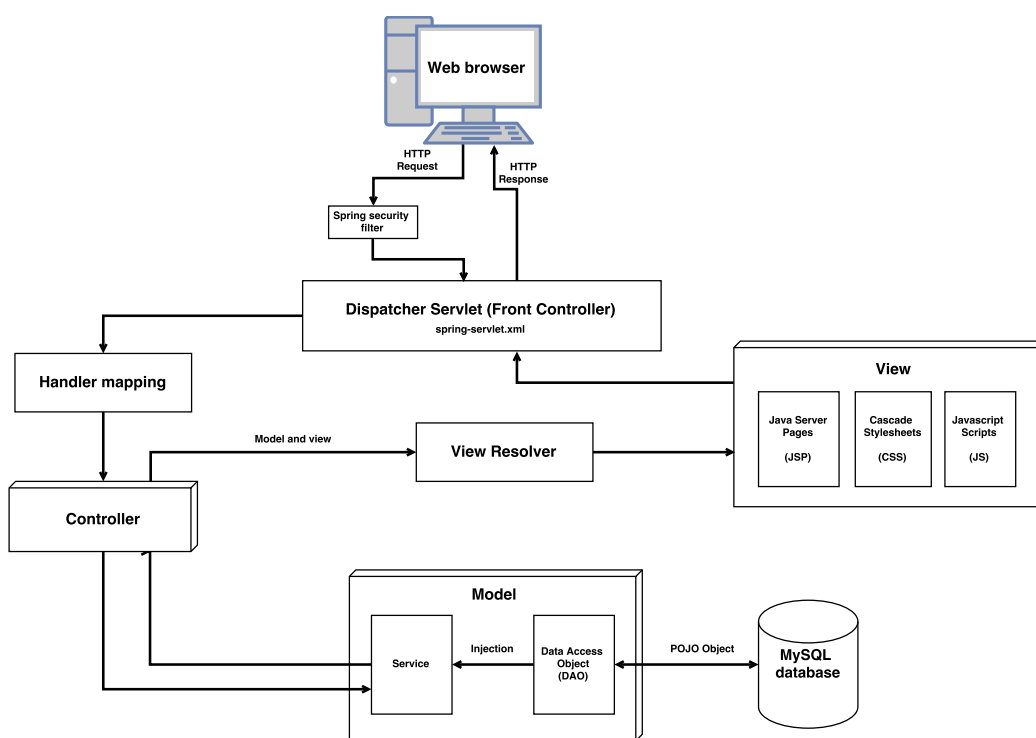
V této sekci si popíšeme návrh architektury systému. Měla by být jednoduše implementovatelná, dobře škálovatelná a rozšiřitelná. Dále musíme zvolit způsob pro komunikaci s databází a vytvořit postup při rozpisu zápasů. Jako hlavní programovací jazyk byla zvolena Java. Pro snazší tvorbu aplikace jsem si vybral framework Spring, který ušetří čas při implementaci, a díky mnoha design patternům pro vývoj bude také dobře čitelný a rozšiřitelný.

6.2 Model–View–Controller

Pro architekturu systému jsem se rozhodl využít architektury MVC (Model-View-Controller). Je to jedna z nejčastěji používaných architektur a součástí frameworku je také Spring MVC, která uzpůsobuje implementaci. Každá ze tří hlavních částí je oddělená a má svůj účel. Na obrázku 6.1 lze vidět strukturu aplikace. Dále v příloze C můžete vidět UML diagram pro rozložení zápasů.

6.2.1 Model

Model je část aplikace, která obsahuje logiku aplikace, komunikuje s databází. Model se skládá z více částí, a to z DAO (Data Access Object) a Service. Dále se stará o propojení s databází.



Obrázek 6.1: MVC architektura aplikace

Data Access Object

DAO je součástí vrstvy, která slouží pro přístup k datům a spojuje služby s databází. Soubor zná zdroj dat (v tomto případě pracuje s doménovým POJO objektem). Je schopna udržovat CRUD požadavky - vytváření datového objektu, čtení dat, jejich aktualizaci a mazání z databáze. Pro každý zdroj dat existuje DAO třída, která je oddělená od základní abstraktní třídy. Ukázka je vidět ve výpisu zdrojového kódu `BaseDAO 6.1`.

```
public abstract class BaseDAO<T, ID extends Serializable>
    implements IBaseDAO<T, ID> {
    @Autowired
    private SessionFactory sessionFactory;

    public Session getSession() {
        return sessionFactory.getCurrentSession();
    }

    public void save(T entity) {
        Session hibernateSession = getSession();
        hibernateSession.saveOrUpdate(entity);
    }

    public void delete(T entity) {
        Session hibernateSession = getSession();
        hibernateSession.delete(entity);
    }

    public void update(T entity) {
        Session hibernateSession = getSession();
        hibernateSession.update(entity);
    }

    // dalsi metody
    public void merge(T entity)
    public List<T> findMany(Query query)
    public T findOne(Query query)
    public T findById(Class<T> clazz, Integer id)
    public List<T> findAll(Class<T> clazz)
    public Integer count(Class<T> clazz)
}
```

Zdrojový kód 6.1: Ukázka generického DAO objektu `BaseDAO`

Ostatní DAO třídy, které jsou v systému, mají také vlastní rozhraní, které dědí od `IBaseDAO`. Sami pak dědí od `BaseDAO` třídy.

Service

Service je vrstva aplikace, která slouží jako vnější vrstva modelu. Controller přes servisní vrstvu přistupuje k datům modelu. Každá třída této vrstvy má automaticky nastavenou property Data Access Objektu přes anotaci `@Autowired`. Každá třída ze servisní vrstvy má anotaci `@Service` a vlastní rozhraní. Ve zdrojovém kódu 6.2 je ukázka ze třídy `RoleService`.

```
@Service
public class RoleService implements IRoleService {
    @Autowired
    @Qualifier(value="roleDao")
    private IRoleDAO roleDAO;

    @Transactional
    @Override
    public List<Role> loadAllRoles() {
        return roleDAO.findAll(Role.class);
    }

    @Transactional
    @Override
    public void saveRole(Role role) {
        roleDAO.save(role);
    }

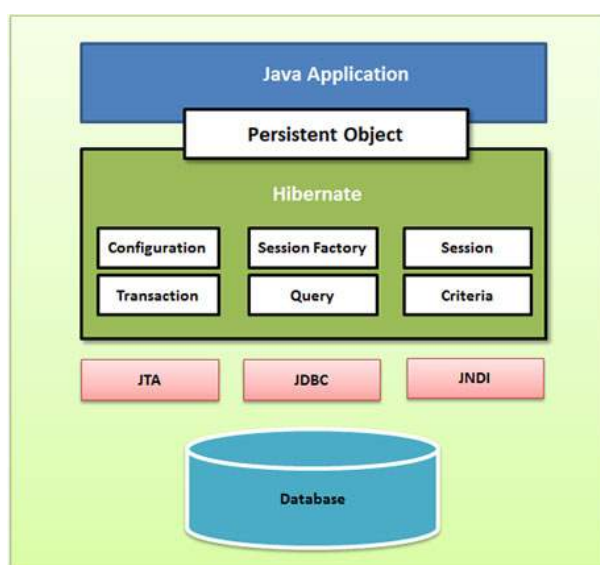
    @Transactional
    @Override
    public List<Role> findRolesLimitOffset(int limit, int offset) {
        Query query = roleDAO.getSession().createQuery("FROM Role
            ORDER BY create_time");
        query.setFirstResult(offset);
        query.setMaxResults(limit);
        return roleDAO.findMany(query);
    }

    // dalsi metody
    public void updateRole(Role role)
    public Role findRoleById(Integer id)
    public void deleteRole(Integer id)
    public Integer getCount()
}
```

Zdrojový kód 6.2: Ukázka třídy ze servisní vrstvy

Propojení s databází

Pro komunikaci s databází jsem se rozhodl využít tzv. ORM (Objektově relační mapování). Nejpoužívanějším frameworkem je Hibernate, který jsem také použil. Ten mapuje objekty z javovských tříd na entity v relační databázi a udržuje ji perzistentní. Mapování lze vytvořit pomocí konfiguračních XML souborů nebo pomocí anotací. Na obrázku 6.2 lze vidět architekturu Hibernate frameworku. Pro připojení k databázi tedy využiji obdobnou strukturu.



Obrázek 6.2: Architektura frameworku Hibernate

Konfigurační objekt je prvním Hibernate objektem vytvořeným v aplikaci s použitím tohoto frameworku. Obsahuje dva základní druhy souborů:

- **Připojení k databázi** – připojení je nastaveno pomocí souboru `hibernate.cfg.xml`. Ukázka ze souboru je popsána ve zdrojovém kódu 6.3.
- **Mapování tříd** – v systému jsem využil mapování pomocí anotací v POJO objektech.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
  "-//Hibernate/Hibernate Configuration DTD//EN"
  "http://hibernate.sourceforge.net/hibernate-configuration-3.0.
  dtd">

<hibernate-configuration>
  <session-factory>
    <mapping class="cz.zcu.kiv.dp.database.domain.Actuality" />
    <mapping class="cz.zcu.kiv.dp.database.domain.Ban" />
    :
    <mapping class="cz.zcu.kiv.dp.database.domain.User" />
    <mapping class="cz.zcu.kiv.dp.database.domain.UserFindTeam" />
  </session-factory>
</hibernate-configuration>
```

Zdrojový kód 6.3: Ukázka souboru `hibernate.cfg.xml`

Konfigurační objekt se používá k vytvoření objektu `SessionFactory`, který nastaví Hibernate pro aplikaci pomocí dodaného konfiguračního souboru. Umožňuje vytvoření instance `Session` objektu. Obvykle je objekt vytvořen jednou při spuštění aplikace. Jeden `SessionFactory` objekt je vytvořen pro jednu databázi pomocí konfiguračního souboru.

Session objekt se využívá k získání fyzického spojení s databází. Vytváří se pokaždé, když je potřeba interakce s databází. Perzistentní objekty jsou ukládány a získávány prostřednictvím tohoto objektu.

Transakční (Transactional) objekt představuje jednotku práce s databází. Většina z RDBMS podporuje funkci transakcí. Transakce jsou prováděny pomocí transakčního manažeru a transakcí (z JDBC nebo JTA).

Dotazovací (Query) objekt se dotazuje pomocí SQL nebo HQL do databáze a vytváří objekty. Instance takového objektu mapuje parametry dotazu, limit vrácených výsledků a také jeho provedení.

Objekt **kritérií (Criteria)** je využíván pro vytváření a provádění objektově orientovaných kritérií a získávání objektů.

Jako perzistentní objekty se využívají tzv. POJO (Plain Old Java Object) objekty. Tyto objekty se jedna ku jedné mapují s databázovou tabulkou. Příklad takového objektu můžete vidět ve zdrojovém kódu 6.4. Příklad je převzat a zkrácen ze třídy `Role.java`.

```
@Entity //Identifikator, ze je trida entitou
@Table(name="ROLE") // mapovani na tabulku
public class Role {
    @Id //je primarnim klicem
    @NotEmpty(message = "{NotEmpty.rights.name}") //nesmi byt
        prazdny
    @Length(max = 64, message = "{Length.rights.name}") //urceni
        delky
    @Column(name = "role_name") //mapovani na sloupec v tabulce
    private String name;

    @Length(max = 5, message = "{Length.rights.shortcut}") //urceni
        delky
    @Column(name="shortcut") //mapovani na sloupec v tabulce
    private String shortcut;

    @ManyToMany(mappedBy="roles") //urceni relace s uzivateli
    private List<User> users;

    @ManyToMany(fetch = FetchType.LAZY) //urceni relace s pravy
    @JoinTable(name = "ROLE_HAS_RIGHTS", joinColumns = {
        @JoinColumn(name = "role_name", nullable = false, updatable =
            false) },
        inverseJoinColumns = { @JoinColumn(name = "right_name",
            nullable = false, updatable = false) }) //propojeni pres
        pomocnou tabulku
    private List<Rights> rights;

    //gettery a settery
}
```

Zdrojový kód 6.4: Ukázka POJO třídy

6.2.2 Controller

Controller (česky kontroler) funguje jako koordinátor mezi modelem a view, a spravuje aplikační logiku. Kontroler dostane požadavek od klienta. Přes model přistupuje k datům do databáze. Výsledná data dále posílá do části view, která je zobrazí na obrazovku klienta. Je tedy zodpovědný za odpověď programu na požadavek klienta. Ukázka je popsána ve zdrojovém kódu 6.5.

```
@Controller
public class BanController {
    @Autowired
    @Qualifier("banService")
    private IBanService banService;

    @Autowired
    @Qualifier("banTypeService")
    private IBanTypeService banTypeService;

    @RequestMapping(value = "/BanList", method = RequestMethod.GET)
    public String getBans(Model model, @RequestParam(value="page",
        required=false) Integer page) {
        Integer bansCount = banService.getCount();
        Integer count = 0;
        if(bansCount % Constants.RECORDS_ON_PAGE_ADMIN != 0) {
            count = 1;
        }
        count += (bansCount / Constants.RECORDS_ON_PAGE_ADMIN);

        if(page == null) {
            page = 1;
        }

        List<Ban> listOfBans = banService.findBansLimitOffset(
            Constants.RECORDS_ON_PAGE_ADMIN, (page - 1) * Constants.
            RECORDS_ON_PAGE_ADMIN);
        model.addAttribute("ban", new Ban());
        model.addAttribute("listOfBans", listOfBans);
        model.addAttribute("pageCount", count);
        return "admin/BanList";
    }

    //dalsi metody
}
```

Zdrojový kód 6.5: Ukázka controlleru

Každý kontroler je označen anotací `@Controller`. Vždy má přístup pouze k servisní vrstvě z modelu. Potřebné třídy z této vrstvy jsou do kontroleru injektovány. Můžete také vidět mapování URL požadavku na danou metodu controlleru. To je zajištěno pomocí anotace `@RequestMapping`, kde hodnota znamená URL adresu a určuje metodu GET nebo POST, podle účelu dané metody. Dále je vidět také namapování na view. Třída kontroleru vrací řetězec, ve kterém se nachází jméno JSP souboru. Ten se zobrazí společně s daty z modelu.

6.2.3 View

View (česky pohled) představuje část aplikace, která obsahuje soubory zobrazující uživateli danou stránku. Výstupem je vyrenderovaná HTML stránka. Pohled se skládá ze tří hlavních druhů souborů:

- **JSP soubory** – obsahová část webových stránek.
- **CSS soubory** – kaskádové styly pro vzhled.
- **JS soubory** – JavaScriptové soubory pro jednoduché dynamické vlastnosti.

V JSP souborech je logika vytvořena pomocí knihovny JSTL (JSP Standard Tag Library). Ta nabízí využití cyklů, podmínek a dalších možností.

6.3 Databázový model

Databázový model aplikace vychází plně z funkčních požadavků na daný systém vzniklých při analýze potencionálních zákazníků v kapitole 5.2. Celkový relační datový model můžete najít v příloze B diplomové práce. Hlavním požadavkem na databázi je její obecnost pro různé sporty. Je potřeba vytvořit takový koncept, který dokáže být napojen na sportovní odvětví s různými statistikami a možnostmi. Ty v jiných sportech využity nebudou.

Návrh databáze byla jedna ze stěžejních úkolů systému. Při špatném návrhu nebo při mapování pouze na jeden sport by mohlo dojít k mnoha nerosvlnalostem při budoucím využití u sportu jiného. Navrženo tedy bylo 43 tabulek, které od důležitých k méně důležitým určují ráz aplikace. Nyní budou popsány jednotlivé tabulky společně s informací, kterou v sobě uchovávají. U všech tabulek existují dva atributy, které uchovávají poslední změnu a také datum vytvoření záznamu. Tyto atributy jsou popsány v tabulce 6.1.

Název	Datový typ	Klíč	Nulový	Popisek
create_time	TIMESTAMP(0)	✗	✗	datum a čas vytvoření záznamu
update_time	TIMESTAMP(0)	✗	✗	datum a čas poslední úpravy záznamu

Tabulka 6.1: Atributy společné pro všechny tabulky

6.4 Tvorba dynamického obsahu

Webové stránky mohou obsahovat kromě statických stránek také jiné informativní stránky. Ty si vytváří administrátor sám. Proto byla vytvořena struktura pro ukládání takových stránek. Ke každé stránce je nutné uchovávat kategorii a zdrojový kód, který se při otevření kategorie zobrazí. Je také možné vytvořit podkategorie, a to maximálně do hloubky 3 (tedy subsubkategorie).

Tabulka CATEGORY

Tabulka ukládá informace o kategoriích, které je možné do systému vložit dynamicky. Struktura entity je znázorněna v tabulce 6.2.

Název	Datový typ	Klíč	Nulový	Popisek
id_category	INT	PK	✗	identifikátor kategorie
title	VARCHAR(100)	✗	✗	titulek kategorie
id_page	INT	FK	✗	identifikátor stránky se zdrojovým kódem

Tabulka 6.2: Tabulka CATEGORY

Tabulka PAGE

Tabulka obsahuje záznamy o stránkách, které jsou přiřazeny kategoriím, subkategoriím nebo subsubkategoriím. Mohou se ještě skládat z podstránek. To určuje flag *subpages*. Struktura je popsána v tabulce 6.3.

Název	Datový typ	Klíč	Nulový	Popisek
id_page	INT	PK	✗	identifikátor zdrojové stránky
source_text	TEXT	✗	✗	zdrojový kód stránky
title	VARCHAR(100)XT	✗	✗	titulek stránky
subpages	BOOL	✗	✗	flag, zda má stránka podstránky
path	INT	✗	✓	cesta k souboru

Tabulka 6.3: Tabulka PAGE

Tabulka SUBCATEGORY

Subkategorie nese podobné informace jako kategorie, pouze má navíc odkaz na svoji rodičovskou kategorii. Struktura je vidět v tabulce 6.4.

Název	Datový typ	Klíč	Nulový	Popisek
id_subcategory	INT	PK	✗	identifikátor subkategorie
title	VARCHAR(100)	✗	✗	titulek subkategorie
id_category	INT	FK	✗	identifikátor nadřazené kategorie
id_page	INT	FK	✗	identifikátor stránky se zdrojovým kódem

Tabulka 6.4: Tabulka SUBCATEGORY

Tabulka SUBSUBCATEGORY

Tabulka subsubkategorie je další v hierarchii kategorií stránek, která je pod subkategorií. Strukturu ukazuje tabulka 6.5.

Název	Datový typ	Klíč	Nulový	Popisek
id_subsubcategory	INT	PK	✗	identifikátor subsubkategorie
title	VARCHAR(100)	✗	✗	titulek subsubkategorie
id_subcategory	INT	FK	✗	identifikátor nadřazené subkategorie
id_page	INT	FK	✗	identifikátor stránky se zdrojovým kódem

Tabulka 6.5: Tabulka SUBSUBCATEGORY

Tabulka TEXT

Tato entita představuje jeden zdrojový kód stránky, pokud se kategorie page skládá z podtextů. Struktura je popsána v tabulce 6.6.

Název	Datový typ	Klíč	Nulový	Popisek
id_text	INT	PK	✗	identifikátor textu
title	VARCHAR(100)	✗	✗	titulek textu
source_text	TEXT	✗	✗	zdrojový kód textu
path	TEXT	✗	✓	cesta k souboru
id_page	INT	FK	✗	identifikátor nadřazené stránky

Tabulka 6.6: Tabulka TEXT

6.5 Práva v systému

Každý návštěvník webových stránek má určitá práva. Musí se vytvořit struktura, která má za úkol uchovávat veškeré informace k privilegiím uživatelů v systému. Dále je nutné umožnit možné přidávání nebo odebrání práv, proto musí být tato definice dynamicky vyřešená.

Tabulka USER

Jedna z nejdůležitějších entit, která slouží k uchování uživatelů. Obsahuje potřebné informace o jednom uživateli, který se do systému zaregistruje. Jako primární klíč bylo zvoleno uživatelské jméno, které musí být unikátní. Všechny ostatní role typu hráč, rozhodčí, trenér nebo brankář musí mít svůj záznam v této tabulce. Struktura je vyznačena v tabulce 6.7.

Název	Datový typ	Klíč	Nulový	Popisek
username	VARCHAR(64)	PK	✗	identifikátor uživatele
name	VARCHAR(32)	✗	✗	jméno uživatele
surname	VARCHAR(64)	✗	✗	příjmení uživatele
email	VARCHAR(128)	✗	✗	email uživatele
password	VARCHAR(255)	✗	✗	heslo uživatele
active	BOOL	✗	✗	flag, zda je uživatel aktivní
phone_number	VARCHAR(14)	✗	✓	telefonní číslo uživatele
notes	TEXT	✗	✓	poznámky k uživateli

Tabulka 6.7: Tabulka USER

Tabulka ROLE

Číselník uchovávající role v systému. Jednotlivé role jsou předdefinovány a nebude možné je změnit v rámci systému. Všechny role jsou popsány v kapitole 5.2. Struktura tabulky je možná vidět v tabulce 6.8.

Název	Datový typ	Klíč	Nulový	Popisek
role_name	VARCHAR(64)	PK	✗	název role
shortcut	VARCHAR(5)	✗	✓	zkratka role

Tabulka 6.8: Tabulka ROLE

Tabulka RIGHT

Číselník uchovávající práva na možné úkony v systému. Jednotlivá práva existují na každou akci, kterou lze v systému vykonat. Každá role má určitá práva, která v budoucnu mohou být obohacována o další nebo naopak. Definice entity je vidět v tabulce 6.9.

Název	Datový typ	Klíč	Nulový	Popisek
right_name	VARCHAR(64)	PK	✗	název práva
shortcut	VARCHAR(5)	✗	✓	zkratka práva

Tabulka 6.9: Tabulka RIGHT

Tabulka BAN

Entita obsahuje informace o banech, které byly přidělené uživateli. Je možné ukončit ban dříve, než je jeho datum vypršení, proto je uchovávána informace o aktivitě (sloupec *active*). Struktura tabulky je popsána v tabulce 6.10.

Název	Datový typ	Klíč	Nulový	Popisek
id_ban	INT	PK	✗	identifikátor banu
maturity	DATETIME	✗	✗	datum vypršení banu
duration	INT	✗	✗	délka trvání
active	BOOL	✗	✗	flag, jestli je ban aktivní
id_ban_type	INT	FK	✗	identifikátor typu banu
username	VARCHAR(64)	FK	✗	identifikátor uživatele

Tabulka 6.10: Tabulka BAN

Tabulka BAN_TYPE

Číselník obsahující typy banů, které je možné přidělit. Bany se udávají například za nevhodné chování na internetu. Jedním z příkladů může být psaní vzkazů se sprostými slovy. Entita má strukturu popsanou v tabulce 6.11.

Název	Datový typ	Klíč	Nulový	Popisek
id_ban_type	INT	PK	✗	identifikátor typu banu
name	VARCHAR(64)	✗	✗	název typu banu
shortcut	VARCHAR(5)	✗	✓	zkratka typu banu

Tabulka 6.11: Tabulka BAN_TYPE

Tabulka ACTUALITY

Tabulka obsahuje aktuality, které se budou zobrazovat na hlavní stránce nebo v sekci aktualit. Slouží k uložení nejdůležitějších informací pro uživatele webu. Entita obsahuje id jako primární klíč, titulek a text aktuality. Struktura je popsána v tabulce 6.12.

Název	Datový typ	Klíč	Nulový	Popisek
id_actuality	INT	PK	✗	identifikátor aktuality
title	VARCHAR(100)	✗	✗	titulek aktuality
text	TEXT	✗	✗	text aktuality

Tabulka 6.12: Tabulka ACTUALITY

Tabulka BANNER

Entita slouží k uložení obrázku banneru určitého sponzora. Pokud má sponzor svůj banner (obrázek) v požadovaném formátu, bude zobrazen na hlavní stránce. Struktura je zobrazena v tabulce 6.13.

Název	Datový typ	Klíč	Nulový	Popisek
id_banner	INT	PK	✗	identifikátor banneru
path	TEXT	✗	✗	cesta k banneru
active	BOOL	✗	✗	flag, zda je banner aktivní
id_sponsor	INT	FK	✗	identifikátor sponzora

Tabulka 6.13: Tabulka BANNER

Tabulka COACH

Entita představuje trenéra týmu. Struktura je popsána v tabulce 6.14.

Název	Datový typ	Klíč	Nulový	Popisek
id_coach	INT	PK	✗	identifikátor trenéra
id_coach_type	INT	FK	✗	identifikátor typu trenéra
username	INT	FK	✗	identifikátor uživatele

Tabulka 6.14: Tabulka COACH

Tabulka COACH_TYPE

Číselník obsahující typy trenérů, které je možné přidělit. Ukládají se záznamy jako hlavní trenér, asistent trenéra. Struktura entity je popsána v tabulce 6.15.

Název	Datový typ	Klíč	Nulový	Popisek
id_coach_type	INT	PK	✗	identifikátor typu trenéra
name	VARCHAR(64)	✗	✗	název typu trenéra
shortcut	VARCHAR(5)	✗	✓	zkratka typu trenéra

Tabulka 6.15: Tabulka COACH_TYPE

Tabulka COMMENT_MATCH

Tato entita slouží pro uložení komentářů k jednotlivým zápasům od návštěvníků webových stránek. Komentáře budou k vidění a přidání u detailu zápasu. Struktura tabulky je zobrazena v tabulce 6.16.

Název	Datový typ	Klíč	Nulový	Popisek
id_comment_match	INT	PK	✗	identifikátor komentáře k zápasu
author	VARCHAR(100)	✗	✗	autor komentáře
text	TEXT	✗	✗	text komentáře
likes	INT	✗	✗	počet liků komentáře
dislikes	INT	✗	✗	počet disliků komentáře
id_match	INT	FK	✗	identifikátor zápasu

Tabulka 6.16: Tabulka COMMENT_MATCH

Tabulka COMMENT_MATCH_PHOTO

Tabulka slouží k ukládání komentářů k fotkám zápasů. Strukturu lze vidět v tabulce 6.17.

Název	Datový typ	Klíč	Nulový	Popisek
id_comment_match_photo	INT	PK	✗	identifikátor komentáře
author	VARCHAR(100)	✗	✗	autor komentáře
text	TEXT	✗	✗	text komentáře
likes	INT	✗	✗	počet líků
dislikes	INT	✗	✗	počet dislíků
id_match_photo	INT	FK	✗	identifikátor zápasové fotografie

Tabulka 6.17: Tabulka COMMENT_MATCH_PHOTO

Tabulka COMMENT_PHOTO

Tato tabulka je poslední, ve které se ukládají komentáře. Tentokrát k fotografiím uložených v galeriích. Struktura je obdobná a ukazuje ji tabulka 6.18.

Název	Datový typ	Klíč	Nulový	Popisek
id_comment_photo	INT	PK	✗	identifikátor komentáře
author	VARCHAR(100)	✗	✗	autor komentáře
text	TEXT	✗	✗	text komentáře
likes	INT	✗	✗	počet líků komentáře
dislikes	INT	✗	✗	počet dislíků komentáře
id_photo	INT	FK	✗	identifikátor fotografie

Tabulka 6.18: Tabulka COMMENT_PHOTO

Tabulka FACULTY

Číselník uchovávající data o fakultách. Pokud se jedná o univerzitní ligu, je nutné mít informace, na které fakultě hráč studuje. Pokud se nejedná o univerzitní ligu, tabulka se nevyužije a zůstane prázdná. Struktura je popsána v tabulce 6.19.

Název	Datový typ	Klíč	Nulový	Popisek
id_faculty	INT	PK	✗	identifikátor fakulty
name	VARCHAR(64)	✗	✗	název fakulty
shortcut	VARCHAR(5)	✗	✓	zkratka fakulty

Tabulka 6.19: Tabulka FACULTY

Tabulka FINE

Tabulka slouží k uložení informací o pokutách, které byly uděleny týmům. Může také být spojena s nějakým zápasem. Struktura je zobrazena v tabulce 6.20.

Název	Datový typ	Klíč	Nulový	Popisek
id_fine	INT	PK	✗	identifikátor pokuty
description	TEXT	✗	✗	popis pokuty
amount	INT	✗	✗	částka k zaplacení
maturity	DATE	✗	✗	datum splatnosti pokuty
paid	BOOL	✗	✗	flag, zda-li je pokuta zaplacená
id_team	INT	FK	✗	identifikátor týmu
id_fine_type	INT	FK	✗	identifikátor typu pokuty
id_match	INT	FK	✓	identifikátor zápasu

Tabulka 6.20: Tabulka FINE

Tabulka FINE_TYPE

Číselník obsahující typy pokut, které je možné vytvořit. Můžou to být například pokuty za špatný zápis, za neodehraný zápas a další. Struktura je popsána v tabulce 6.21.

Název	Datový typ	Klíč	Nulový	Popisek
id_fine_type	INT	PK	✗	identifikátor typu pokuty
name	VARCHAR(64)	✗	✗	název typu pokuty
shortcut	VARCHAR(5)	✗	✓	zkratka typu pokuty

Tabulka 6.21: Tabulka FINE_TYPE

Tabulka FORUM

Jednotlivé záznamy v tabulce označují příspěvky, které uživatelé na fórum přidali. K příspěvkům mohou další uživatelé přidávat „palce nahoru a dolů“. Strukturu můžete vidět v tabulce 6.22.

Název	Datový typ	Klíč	Nulový	Popisek
id_message	INT	PK	✗	identifikátor příspěvku ve fóru
title	VARCHAR(100)	✗	✓	titulek příspěvku
text	TEXT	✗	✗	text příspěvku
author	VARCHAR(100)	✗	✗	autor příspěvku
likes	INT	✗	✗	počet líků příspěvku
dislikes	INT	✗	✗	počet dislíků příspěvku
username	INT	FK	✓	identifikátor uživatele

Tabulka 6.22: Tabulka FORUM

Tabulka GALLERY

Tabulka obsahuje informace o galeriích. Pro přeskakující obrázky na hlavní stránce webovek je vložena hlavní galerie. Ke každé galerii mohou uživatelé také přidávat líky a dislíky. Struktura je popsána v tabulce 6.23.

Název	Datový typ	Klíč	Nulový	Popisek
id_gallery	INT	PK	✗	identifikátor galerie
title	VARCHAR(100)	✗	✗	titulek galerie
description	TEXT	✗	✗	popisek galerie
likes	INT	✗	✗	počet líků galerie
dislikes	INT	✗	✗	počet dislíků galerie

Tabulka 6.23: Tabulka GALLERY

Tabulka GOAL

Tabulka obsahuje góly, které byly vstřelené v rámci zápasů. Zde dochází k rozdílu u ukládání různých sportů. U futsalu, hokeje, florbalu a dalších je možné ukládat jednotlivé góly společně s asistencemi. Ovšem pokud nechceme ukládat statistiky tak podrobně nebo máme jiné sporty (basketbal, házená a další), potřebujeme ukládat pouze kolik hráčů dalo kolik gólů a neukládat další informace. Proto je mnoho parametrů nepovinných a vytvořen sloupec *count*. Strukturu lze vidět v tabulce 6.24.

Název	Datový typ	Klíč	Nulový	Popisek
id_goal	INT	PK	✗	identifikátor gólu
time	VARCHAR(5)	✗	✓	čas, ve kterém gól padl
match_part	INT	✗	✓	část zápasu, ve které gól padl
count	INT	✗	✓	počet gólů
id_goal_type	INT	FK	✗	identifikátor typu gólu
id_match	INT	FK	✗	identifikátor zápasu
id_player_goal	INT	FK	✗	identifikátor hráče, který dal gól
id_player_assist_1	INT	FK	✓	identifikátor hráče, který měl první asistenci
id_player_assist_2	INT	FK	✓	identifikátor hráče, který měl druhou asistenci

Tabulka 6.24: Tabulka GOAL

Tabulka GOALKEEPER

Tabulka obsahuje informace o rozhodčích. Je navázána na tabulku hráče, protože brankář může vypomoci týmu v poli a dávat góly. V tabulce jsou uloženy statistiky související pouze s brankářem. Entita má strukturu popsanou v tabulce 6.25.

Název	Datový typ	Klíč	Nulový	Popisek
id_goalkeeper	INT	PK	✗	identifikátor brankáře
saves	INT	✗	✗	počet úspěšných zákroků
goals	INT	✗	✗	počet obdržných gólů
wins	INT	✗	✗	počet vítězství
active	BOOL	✗	✗	flag, jestli je brankář aktivní
id_player	INT	FK	✗	identifikátor hráče, který je zároveň brankářem

Tabulka 6.25: Tabulka GOALKEEPER

Tabulka GOAL_TYPE

Číselník obsahující typy gólů, které je možné vstřelit. Při sportech, kde neukládáme asistence přímo u gólu, je v této tabulce uložen také typ pro asistenci. Jedním ze sloupečků určeným pro speciální sporty, je *own*. Určuje typ gólu, který je vlastní a je připisován druhému týmu. Struktura je sepsána v tabulce 6.26

Tabulka MATCHES

Tabulka sloužící pro ukládání zápasů. K zápasu jde uložit organizátor zápasu a také maximálně čtyři rozhodčí. Tabulka má název ve množném čísle, pro-

Název	Datový typ	Klíč	Nulový	Popisek
id_goal_type	INT	PK	✗	identifikátor typu gólu
name	VARCHAR(64)	✗	✗	název typu gólu
shortcut	VARCHAR(5)	✗	✓	zkratka typu gólu
own	BOOL	✗	✗	flag, jestli je gól vlastní

Tabulka 6.26: Tabulka GOAL_TYPE

tože MATCH je rezervované slovo jazyka SQL. Struktura entity je popsána v tabulce 6.27.

Název	Datový typ	Klíč	Nulový	Popisek
id_match	INT	PK	✗	identifikátor zápasu
description	INT	✗	✓	popis zápasu
date	DATETIME	✗	✓	datum a čas odehrání zápasu
goals_home	INT	✗	✗	počet gólů domácích v zápasu
goals_away	INT	✗	✗	počet gólů hostů v zápasu
round	INT	✗	✓	kolo, ve kterém se zápas odehraje
basic_part	INT	✗	✗	flag, jestli je zápas v základní části
year	VARCHAR(9)	✗	✓	sezóna, ve kterém se zápas odehraje
match_number	INT	✗	✗	číslo zápasu
played	BOOL	✗	✗	flag, zda je zápas odehraný
team_id_home	INT	FK	✗	identifikátor domácího týmu
team_id_away	INT	FK	✗	identifikátor hostujícího týmu
team_id_organizator	INT	FK	✓	identifikátor organizátorského týmu
id_referee	INT	FK	✓	identifikátor hlavního rozhodčího
id_referee1	INT	FK	✓	identifikátor dalšího rozhodčího
id_referee2	INT	FK	✓	identifikátor dalšího rozhodčího
id_referee3	INT	FK	✓	identifikátor dalšího rozhodčího

Tabulka 6.27: Tabulka MATCHES

Tabulka PHOTO

Entita pro ukládání fotografií. Každá fotografie je přiřazena určité galerii. Struktura je zobrazena v tabulce 6.28.

Tabulka PLAYER

Entita je jednou z dalších velmi důležitých tabulek v databázovém modelu. Obsahuje důležité informace o hráčích a také veškeré statistiky. Opět zde můžeme vidět sloupce jako *red_card* nebo *yellow_card*, které jsou určeny hlavně

Název	Datový typ	Klíč	Nulový	Popisek
id_photo	INT	PK	✗	identifikátor fotografie
description	TEXT	✗	✗	popisek fotografie
title	VARCHAR(100)XT	✗	✗	titulek fotografie
likes	INT	✗	✗	počet líků fotografie
dislikes	INT	✗	✗	počet dislíků fotografie
path	INT	✗	✓	cesta k fotografii
id_gallery	INT	FK	✗	identifikátor galerie, do které fotografie patří

Tabulka 6.28: Tabulka PHOTO

pro sporty jako fotbal nebo futsal. Existuje zde nspecifikovaný flag, který například pro futsalovou ligu znamená, že hráč je členem USK. Struktura je popsána v tabulce 6.29.

Název	Datový typ	Klíč	Nulový	Popisek
id_player	INT	PK	✗	identifikátor hráče
number	INT	✗	✗	číslo na dresu hráče
goal	INT	✗	✗	počet gólů hráče
assist	INT	✗	✗	počet asistencí hráče
penalty	INT	✗	✗	počet trenstných minut hráče
red_card	INT	✗	✗	počet žlutých karet hráče
yellow_card	INT	✗	✗	počet červených karet hráče
active	BOOL	✗	✗	flag, zda je hráč aktivní
flag	BOOL	✗	✓	nespecifikovaný flag
id_team	INT	FK	✗	identifikátor týmu, za který hráč hraje
id_faculty	INT	FK	✓	identifikátor fakulty, kterou hráč studuje
id_status	INT	FK	✓	identifikátor statusu hráče
id_post	INT	FK	✗	identifikátor postu hráče
username	INT	FK	✗	identifikátor uživatele

Tabulka 6.29: Tabulka PLAYER

Tabulka POST

Číselník obsahující informace o typech postů jednotlivých hráčů. Záznamy budou vloženy do databáze staticky. Struktura je sepsána v tabulce 6.30.

Tabulka REFEREE

Tabulka udržuje v systému informace o rozhodčích. Struktura je vidět v tabulce 6.31.

Název	Datový typ	Klíč	Nulový	Popisek
id_post	INT	PK	✗	identifikátor postu
name	VARCHAR(64)	✗	✗	název postu
shortcut	VARCHAR(5)	✗	✓	zkratka postu

Tabulka 6.30: Tabulka POST

Název	Datový typ	Klíč	Nulový	Popisek
id_referee	INT	PK	✗	identifikátor rozhodčího
id_referee_type	INT	FK	✗	identifikátor typu rozhodčího
username	INT	FK	✗	identifikátor uživatele, který se stal rozhodčím

Tabulka 6.31: Tabulka REFEREE

Tabulka REFEREE_TYPE

Číselník obsahující typy rozhodčích, které v daném sportu mohou být. Mohou to být rozhodčí hlavní, čároví, nebo videorozhodčí a další. Struktura je popsána v tabulce 6.32.

Název	Datový typ	Klíč	Nulový	Popisek
id_referee_type	INT	PK	✗	identifikátor typu rozhodčího
name	VARCHAR(64)	✗	✗	název typu rozhodčího
shortcut	VARCHAR(5)	✗	✓	zkratka typu rozhodčího

Tabulka 6.32: Tabulka REFEREE_TYPE

Tabulka ROSTER

Tabulka obsahuje soupisky týmů na daný zápas. Soupisky budou vytvářeny při vzniku zápasů. Následně budou při přidání nebo odebrání hráčů jednotlivé soupisky aktualizovány. Hráč se může na zápas přihlásit, může být nominován. Hlavním cílem je určit, kolik zápasů daný hráč odehrál. Pro vedení futsalové ligy je potřeba jednoduše zjistit počet odehraných zápasů. Strukturu entity je možné vidět v tabulce 6.33.

Název	Datový typ	Klíč	Nulový	Popisek
id_roster	INT	PK	✗	identifikátor záznamu v soupisce
signed	BOOL	✗	✗	flag, jestli je hráč přihlášen na zápas
nominated	BOOL	✗	✗	flag, jestli je hráč nominován na zápas
played	BOOL	✗	✗	flag, jestli hráč odehrál zápas
id_player	INT	FK	✗	identifikátor hráče
id_match	INT	FK	✗	identifikátor zápasu

Tabulka 6.33: Tabulka ROSTER

Tabulka SETTING

Tabulka obsahuje nastavení, které budou prolínány celým systémem. Struktura je zobrazena v tabulce 6.34.

Název	Datový typ	Klíč	Nulový	Popisek
id_setting	INT	PK	✗	identifikátor nastavení
name	VARCHAR(100)	✗	✗	název nastavení
value	VARCHAR(64)	✗	✗	hodnota nastavení
default_value	VARCHAR(64)	✗	✗	defaultní hodnota nastavení

Tabulka 6.34: Tabulka SETTING

Tabulka SOCIAL_EVENT

Tato tabulka ukládá informace o společenských akcích. Jakýkoliv registrovaný uživatel může na stránky vložit společenskou akci pro pozvání ostatních návštěvníků webových stránek. Struktura je popsána v tabulce 6.35.

Název	Datový typ	Klíč	Nulový	Popisek
id_social_event	INT	PK	✗	identifikátor společenské akce
title	VARCHAR(100)	✗	✗	titulek společenské akce
text	TEXT	✗	✗	popisek společenské akce
datetime_from	DATETIME	✗	✗	datum a čas začátku společenské akce
datetime_to	DATETIME	✗	✗	datum a čas konce společenské akce
place	VARCHAR(128)	✗	✗	místo společenské akce
username	INT	FK	✗	identifikátor uživatele, který akci založil

Tabulka 6.35: Tabulka SOCIAL_EVENT

Tabulka SPONSOR

Entita obsahuje informace o sponzorech ligy. Struktura je sepsána v tabulce 6.36.

Název	Datový typ	Klíč	Nulový	Popisek
id_sponsor	INT	PK	✗	identifikátor sponzora
name	VARCHAR(100)	✗	✗	název sponzora
description	TEXT	✗	✗	popisek sponzora

Tabulka 6.36: Tabulka SPONSOR

Tabulka STATUS

Tabulka obsahuje informace o stavech hráčů. Může představovat různé druhy statusů. Jedním z typů může být registrovaný a neregistrovaný hráč, ve které mohou být rozděleni do dalších kategorií. Jinou možností je například rozdělení podle výšky poplatku za hraní ligy. Struktura entity je popsána v tabulce 6.37.

Název	Datový typ	Klíč	Nulový	Popisek
id_status	INT	PK	✗	identifikátor statusu
name	VARCHAR(100)	✗	✗	název statusu
shortcut	VARCHAR(5)	✗	✓	zkratka statusu
price	INT	✗	✓	cena, kterou hráč zaplatí za účast

Tabulka 6.37: Tabulka STATUS

Tabulka TEAM

Tabulka uchovává statistiky a další informace o týmech, kteří hrají ligu. Každý tým musí mít svého vedoucího z role uživatelů systému. Struktura je popsána v tabulce 6.38.

Tabulka TEAM_CHAT

Tabulka slouží pro uložení zpráv do týmového chatu. Informace z tabulky vidí pouze hráči daného týmu. Ostatní uživatelé přístup k informacím nemají (a

Název	Datový typ	Klíč	Nulový	Popisek
id_team	INT	PK	✗	identifikátor týmu
name	VARCHAR(64)	✗	✗	název týmu
shortcut	VARCHAR(5)	✗	✓	zkratka týmu
wins	INT	✗	✗	počet vítězství
loses	INT	✗	✗	počet proher
draw	INT	✗	✗	počet remíz
wins_pp	INT	✗	✗	počet vítězství v prodloužení
loses_pp	INT	✗	✗	počet porážek v prodloužení
wins_so	INT	✗	✗	počet vítězství po samostatných nájezdech
loses_so	INT	✗	✗	počet proher po samostatných nájezdech
goals_home	INT	✗	✗	počet vstřelených gólů
goals_away	INT	✗	✗	počet obdržených gólů
points	INT	✗	✗	počet získaných bodů
penalty	INT	✗	✗	počet trestných minut
active	BOOL	✗	✗	flag, zda je tým aktivní
team_leader	INT	FK	✓	identifikátor vedoucího týmu
id_coach_main	INT	FK	✓	identifikátor hlavního trenéra
id_coach_assitant	INT	FK	✓	identifikátor asistenta trenéra
id_coach_third	INT	FK	✓	identifikátor případného dalšího trenéra

Tabulka 6.38: Tabulka TEAM

to ani administrátor ligy). Struktura je vidět v tabulce 6.39.

Název	Datový typ	Klíč	Nulový	Popisek
id_message	INT	PK	✗	identifikátor zprávy v týmovém chatu
title	VARCHAR(100)	✗	✓	titulek zprávy
text	TEXT	✗	✗	text zprávy
id_team	INT	FK	✗	identifikátor týmu
id_player	INT	FK	✗	identifikátor hráče, který zprávu napsal

Tabulka 6.39: Tabulka TEAM_CHAT

Tabulka TEAM_FIND_PLAYER

Tato tabulka obsahuje informace o týmech, které hledají hráče. Vedoucí týmu může vložit do této sekce příspěvek a tím zvýšit šance k nalezení dalších hráčů do svého týmu. Struktura je zobrazena v tabulce 6.40.

Tabulka USER_FIND_TEAM

Tato entita je opačnou k předchozí. I noví uživatelé mají možnost zvýšit šance na zapojení do ligy prostřednictvím informací uložených v této tabulce. Struktura je popsána v tabulce 6.41.

Název	Datový typ	Klíč	Nulový	Popisek
id_team_find_player	INT	PK	✗	identifikátor záznamu tým hledá hráče
title	VARCHAR(100)	✗	✗	titulek záznamu
description	TEXT	✗	✗	popisek záznamu
active	BOOL	✗	✗	flag, jestli je záznam aktuální
id_team	INT	FK	✗	identifikátor týmu

Tabulka 6.40: Tabulka TEAM_FIND_PLAYER

Název	Datový typ	Klíč	Nulový	Popisek
id_user_find_team	INT	PK	✗	identifikátor záznamu uživatel hledá tým
title	VARCHAR(100)	✗	✗	titulek záznamu
description	TEXT	✗	✗	popisek záznamu
active	BOOL	✗	✗	flag, jestli je záznam aktuální
username	INT	FK	✗	identifikátor uživatele

Tabulka 6.41: Tabulka USER_FIND_TEAM

6.5.1 Procedury

Procedury jsou uložené části kódu, které se mohou používat vícekrát za sebou. Mohou mít parametry a výstupní hodnoty. Hlavním využitím uložených procedur v mé práci je aktualizace statistik. Při získávání statistik by vznikaly velmi složité a časově náročné dotazy. Proto jsem se rozhodl využít uložení některých statistik přímo u entit a pomocí triggerů hodnoty ukládat nebo měnit. Zdrojový kód 6.6 ukazuje proceduru pro uložení statistik z tabulky gólu do tabulky hráč.


```
BEGIN
DECLARE id_goal_player INTEGER;
DECLARE id_goal_assist1 INTEGER;
DECLARE id_goal_assist2 INTEGER;
DECLARE goal_type_own name BOOLEAN;

SELECT a.id_player_goal , a.id_player_assist_1 , a.
id_player_assist_2 , b.own INTO id_goal_player ,
id_goal_assist1 , id_goal_assist2 , goal_type_own FROM goal AS
a LEFT JOIN goal_type AS b ON a.id_goal_type = b.id_goal_type
WHERE a.id_goal = goal_id;

IF goal_type_own = false THEN
UPDATE player SET goal = goal + 1 WHERE id_player =
id_goal_player;
IF id_goal_assist1 IS NOT NULL THEN
UPDATE player SET assist = assist + 1 WHERE id_player =
id_goal_assist1;
END IF;
IF id_goal_assist2 IS NOT NULL THEN
UPDATE player SET assist = assist + 1 WHERE id_player =
id_goal_assist2;
END IF;
END IF;
END
```

Zdrojový kód 6.6: Ukázka procedury pro přidání statistik z gólu

6.5.2 Triggery

Triggery slouží v databázi jako spouště vykonávající určitou operaci. Ta se vykoná v souvislosti s určitou situací, která nastane v databázi. Jedná se například o vložení řádku, aktualizaci řádku nebo jeho odstranění. V každé tabulce je obsažen sloupec pro datum poslední aktualizace záznamu. Právě na tento případ trigger využijí. Ve zdrojovém kódu 6.7 lze vidět ukázkou triggeru pro tabulku `ROLE`. Ten se spustí před aktualizací řádku a pro každý řádek nastaví hodnotu sloupce `update_time` na aktuální čas. Používá se tedy u všech tabulek, které jsou v databázovém schématu.

```
CREATE
TRIGGER 'role_update_time'
BEFORE UPDATE
ON 'ROLE'
FOR EACH ROW
SET NEW.update_time = CURRENT_TIMESTAMP
```

Zdrojový kód 6.7: Ukázka triggeru pro tabulku ROLE

Další využití triggerů v mé práci je při ukládání statistik. Všechny uložené procedury jsou volány pomocí triggerů příkazem CALL.

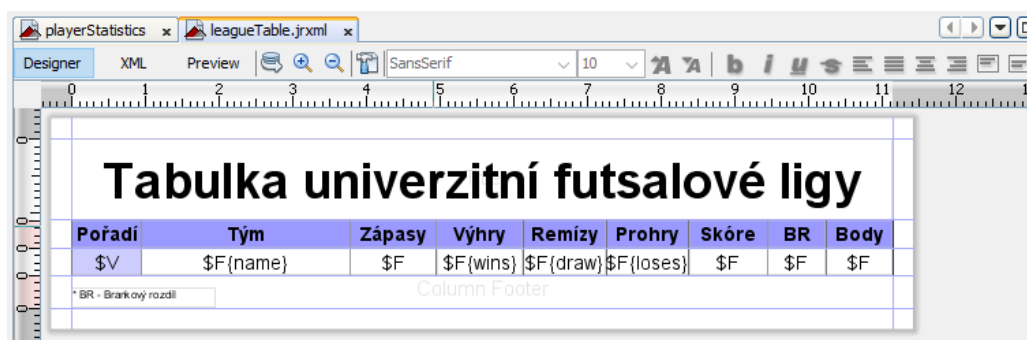
6.5.3 Pohledy

Dále v databázi využívám také pohledy na data (view). Pohledy slouží k výběru složitějších dat a dotazů, jako je například ligová tabulka nebo hráčské statistiky. Většinu informací obstarávají výše zmíněné procedury a triggeru, avšak určité informace jsem se rozhodl získávat až při zobrazování dat. Pohledy jsou namapované v aplikaci samostatně.

6.6 Export do PDF

Export statistik nebo zápisů do PDF je prováděn pomocí Jasper report frameworku. Díky editoru, který umí pomocí Designeru vytvářet reporty vizuálně, se pro mě stal první volbou při výběru. Reporty jsou strukturovány do více částí podle typu obsahu. Příkladem může být hlavička, detail nebo patička. Vstupem do reportu je speciální kolekce `JRBeanCollectionDataSource`, která obsahuje kolekci dat zobrazenou v části detailu. Mapování probíhá pomocí tzv. fieldů, které pomocí názvu proměnné dokážou mapovat jednotlivé proměnné v datovém zdroji.

Report je reprezentován šablonou. To je soubor s koncovkou `.jrxml`. Tato šablona je přeložena do souboru s příponou `.jasper`, která se dále využívá a plní daty. Design šablony můžete vidět na obrázku 6.3.



Obrázek 6.3: Úvodní šablony .jrxml

6.7 Zabezpečení

Zabezpečení aplikace je vytvořeno pomocí Spring Security. Spring Security je framework, který poskytuje celou řadu bezpečnostních služeb aplikacím v Javě. Spring Security je komplexní autentikační/autorizační framework postavený na vrcholu Spring frameworku. Dodržuje postupy ním zavedené. Má otevřený zdrojový kód, velkou komunitu a vysokou uživatelskou základnu [12].

Pro bezpečnost využívám podmnožinu funkcí umožňující přihlášení do systému a také povolení přístupů k částem webových stránek. Při řešení problému s ověřením každé role, zda-li může přistupovat na webovou stránku, se nabízel Spring Security framework jako nejlepší řešení. Obsahuje logiku pro přihlašování do a odhlašování z aplikace. Je potřeba pouze nakonfigurovat a namapovat logiku na formulář nebo tlačítko. Po přihlášení si uživatel udržuje informaci o tom, jaké role a práva v systému má. Poté jsou přes interceptory nastavena práva přístupu na webové stránky. Interceptor je návrhový vzor, který umožňuje třetí straně rozšiřovat aplikaci o další služby. Ze systému publikujeme událost URL, tu zachytí dispatcher a předává ji zaregistrovaným interceptorům. Ve zdrojovém kódu 6.8 můžete vidět přiřazení práv v systému k jednotlivým URL adresám.

```
<beans:beans xmlns="http://www.springframework.org/schema/
  security"
  xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.
  xsd
  http://www.springframework.org/schema/security
  http://www.springframework.org/schema/security/spring-security.
  xsd">

<!-- enable use-expressions -->
<http auto-config="true" use-expressions="true">
  <intercept-url pattern="/ActualityList" access="hasAuthority('
  ACTUALITY_LIST')"/>
  <intercept-url pattern="/ActualityInsertUpdate" access="
  hasAuthority('ACTUALITY_INSERT_UPDATE')"/>
  <intercept-url pattern="/deleteActuality" access="hasAuthority
  ('ACTUALITY_INSERT_UPDATE')"/>
  <intercept-url pattern="/saveActuality" access="hasAuthority('
  ACTUALITY_INSERT_UPDATE')"/>

<!-- access denied page -->
<access-denied-handler error-page="/AccessDenied"/>
  <form-login
    login-processing-url="/login"
    login-page="/Login"
    authentication-failure-url="/Login?error=true"
    username-parameter="username"
    password-parameter="password"/>
  <logout logout-url="/logout"/>
  <!-- enable csrf protection -->
  <csrf />
</http>

<authentication-manager>
  <authentication-provider user-service-ref="
  customUserDetailsService">
    <password-encoder hash="bcrypt"/>
  </authentication-provider>
</authentication-manager>
```

Zdrojový kód 6.8: Ukázka zabezpečení přístupu pomocí Spring Security

7 Testování

V této kapitole popíši principy testování systému, které jsem prováděl sám nebo za pomoci zákazníka.

7.1 Funkční testování

Funkční testování probíhalo pomocí testování **blackbox**. Je znám vstup a výstup při dané akci, uživatel nevidí do kódu. V této kapitole jsou uvedeny pouze případy, které neprošly testováním. Vždy je uveden důvod a změna, která musela být provedena.

7.1.1 Vložení a úprava aktuality

- **Vstup** – upravený vstup z formuláře.
- **Výstup** – změněná data v databázi, tabulce ACTUALITY.
- **Důvod chyby** – úprava mění hodnotu data i času vytvoření podle času úpravy.
- **Oprava** – nepředávání hodnoty sloupce z formuláře pomocí <form:hidden> tagu, formulář doplněn.

7.1.2 Vložení a úprava banu

- **Vstup** – vstup z formuláře.
- **Výstup** – změněná data v databázi, tabulce BAN.
- **Důvod chyby** – nelze vložit ani upravit ban, nepodporovaný formát data a času.
- **Oprava** – změna špatného formátovacího patternu na dd.MM.yyyy.

7.1.3 Seznam typů pokuty

- **Vstup** – kliknutí na tlačítko Pokuty v záložce pokuty.
- **Výstup** – seznam typů pokut, tlačítka na úpravu a vymazání.
- **Důvod chyby** – nelze provést jakékoliv úpravy existujícího typu pokuty, po kliknutí na typ pokuty se nic nestane.
- **Oprava** – změna špatného odkazu na identifikátor accordionu.

7.1.4 Vložení nového hráče

- **Vstup** – vstup z formuláře.
- **Výstup** – uložený hráč v databázi, přiřazená soupiska ke všem zápasům.
- **Důvod chyby** – soupiska se nepřirazuje k zápasům, chybí možnost využití flagu pro člena USK.
- **Oprava** – přidání nového vstupu na formuláři, zavolání procedury pro přidání soupisek pomocí triggeru v databázi.

7.1.5 Vložení společenské akce

- **Vstup** – vložení společenské akce.
- **Výstup** – společenská akce v databázi.
- **Důvod chyby** – nelze vložit společenskou akci v administrátorském prostředí, není ani zobrazena v informativní části.
- **Oprava** – funkcionatita připravena, pouze nedodané odkazy a tlačítka k provedení.

7.2 Testování použitelnosti

Testování použitelnosti aplikace byla prováděna za pomoci osoby, která nese souvisela s vývojem projektu. Po vytvoření uživatelské dokumentace byly zkušeny funkce a hledány chyby související s použitelností a uživatelskou přívětivostí aplikace. Byly nalezeny tyto chyby:

- Špatné řazení podle data vytvoření u všech funkcí v administrátorském prostředí.
- Špatný formát data a času u všech funkcí v administrátorském prostředí.
- Chybí řazení tabulek u statistik a ligové tabulky.

7.3 Testování bezpečnosti

Na zabezpečení aplikaci není kladený vysoký důraz. Je však nutné, aby nebyl umožněn přístup na jednotlivé stránky uživatelům, kteří nemají správná práva. Uživatel se vždy musí přihlásit uživatelským jménem a heslem. Bez přihlášení není možné se dostat na nepovolené stránky. Během testování bezpečnosti byly nalezeny dvě chyby, které neovlivňovaly chod aplikace.

- Na hlavní stránku administrace se dokázal uživatel dostat po zadání URL.

Tato chyba byla odstraněna pomocí úpravy souboru `spring-security.xml`. Pro odstranění chyby byl vložen další interceptor s právy uživatele pro přístup k hlavní stránce administrace.

7.4 Zhodnocení testování

Podmnožina testování

8 Závěr

Cílem tohoto projektu bylo vytvoření systému pro správu sportovních lig. Požadavkem byla komplexita a robustnost systému. Systémová databáze by měla být mapovatelná na více sportů. Primárním zákazníkem diplomové práce je Katedra tělesné výchovy a sportu, přednostně vedoucí futsalové ligy, Mgr. Jaroslav Kovařík.

V první kapitole popisují motivaci k tvorbě projektu, proč jsem se rozhodl aplikaci vyvíjet a jaký by měl být její přínos. V další kapitole jsem provedl analýzu existujících systémů a komponent, které by mohli být využitelné při tvorbě projektu. Také jsem analyzoval různé druhy rozpisu zápasů. V následující kapitole jsem provedl analýzu technologií vhodných pro implementaci, zhodnotil a vybral podmnožinu jazyků, ve kterých bude aplikace vytvořena. Dále jsem provedl analýzu potřeb potencionálních uživatelů. K tomu jsem využíval schůzek nejen se zadavatelem, ale také s dalšími potencionálními uživateli. V této kapitole také byly vytvořeny role v systému. V další kapitole jsem provedl návrh architektury, databázového modelu a rozpisu zápasů společně s implementací projektu. V poslední kapitole jsem provedl více druhů testování systému a opravil chyby, které byly nalezeny.

Práce pro mě byla velice přínosná. Diplomová práce byla můj prozatím největší projekt v životě. Nabyt jsem mnoho zkušeností s tvorbou softwaru, a to nejen z hlediska implementace, ale hlavně z hlediska analýzy. Ta bylo oproti očekávání časově velice náročná. Bylo také nutné komunikovat se zákazníkem, který měl o systému jisté představy a musel jsem se jim přizpůsobovat.

Nasazení systému na komerční prostředí bohužel ještě nebylo provedeno. Na nový systém by se mohlo přecházet ke konci futsalové sezóny, právě proto byla ponechána na serveru stará struktura stránek a nasazení proběhne až před začátkem sezóny 2017/2018. Nyní probíhá komunikace se zákazníkem, který nechává připravit prostředí pro běh aplikace na ZČU. Nasazení by mělo proběhnout před zahájením další sezóny, na začátku akademického roku. Dále byl projeven zájem o nasazení systému také pro vedení florbalové ligy. Bylo předběžně domluveno, že po menších grafických úpravách by mělo dojít k nasazení také před zahájením sezóny 2017/2018.

Systém by měl být rozšiřitelný a tak se nabízí velké množství vylepšení, které však nebyly možné vytvořit v rámci diplomové práce. Hlavně z hlediska časového jsem rozsah překročil o mnoho hodin. Databáze je také připravena

pro další funkcionalitu, která se v systému neprojevila (např. komentáře k fotografiím).

Z výsledků testování vyplývá, že systém splňuje požadavky zákazníka. Bylo potřeba pouze mírných oprav. Funkční testování bylo provedeno na základě diagramu případů užití.

Vzhledem k výše popsaným informacím jsem přesvědčen, že jsem práci splnil diplomovou práci v plném rozsahu.

Seznam použitých zkratek

- **API** – Application Programming Interface
- **CGI** – Common Gateway Interface
- **CLR** – Common Language Runtime
- **CRUD** – Create, Read, Update, Delete
- **CSS** – Cascading Style Sheets
- **DAO** – Data Access Object
- **DRY** – Don't Repeat Yourself
- **EJB** – Enterprise Java Bean
- **GUI** – Graphical User Interface
- **HTML** – Hypertext Markup Language
- **HQL** – Hibernate Query Language
- **I/O** – Input/Output
- **IOOP** – Internet Inter-Orb Protocol
- **IRC** – Internet Relay Chat
- **Java EE** – Java Enterprise Edition
- **Java SE** – Java Standard Edition
- **JDBC** – Java Database Connectivity
- **JS** – JavaScript

- **JSF** – Java Server Faces
- **JSON** – JavaScript Object Notation
- **JSP** – Java Server Pages
- **JSTL** – Java Standard Tag Library
- **JTA** – Java Transaction API
- **KHL** – Kontinental Hockey League
- **KTS** – Katedra tělesné výchovy a sportu
- **MIT** – Massachusetts Institute of Technology
- **MSIL** – Microsoft Intermediate Language
- **MVC** – Model-View-Controller
- **NHL** – National Hockey League
- **ORM** – objektově relační mapování
- **OS** – operační systém
- **PHP** – PHP: Hypertext Preprocessor
- **POJO** – Plain Old Java Object
- **RDBMS** – Relational Database Management System
- **REST** – Representational State Transfer
- **RMI** – Java Remote Method Invocation
- **SOAP** – Simple Object Access Protocol
- **SQL** – Structure Query Language
- **Š.L.A.H.** – Šumavská liga amatérského hokeje
- **UK** – Univerzita Karlova
- **URL** – Uniform Resource Locator
- **USK** – Univerzitní sportovní klub
- **W3C** – World Wide Web Consortium

- **XHTML** – eXtensible Hypertext Markup Language
- **XML** – eXtensible Markup Language
- **ZČU** – Západočeská univerzita v Plzni

Literatura

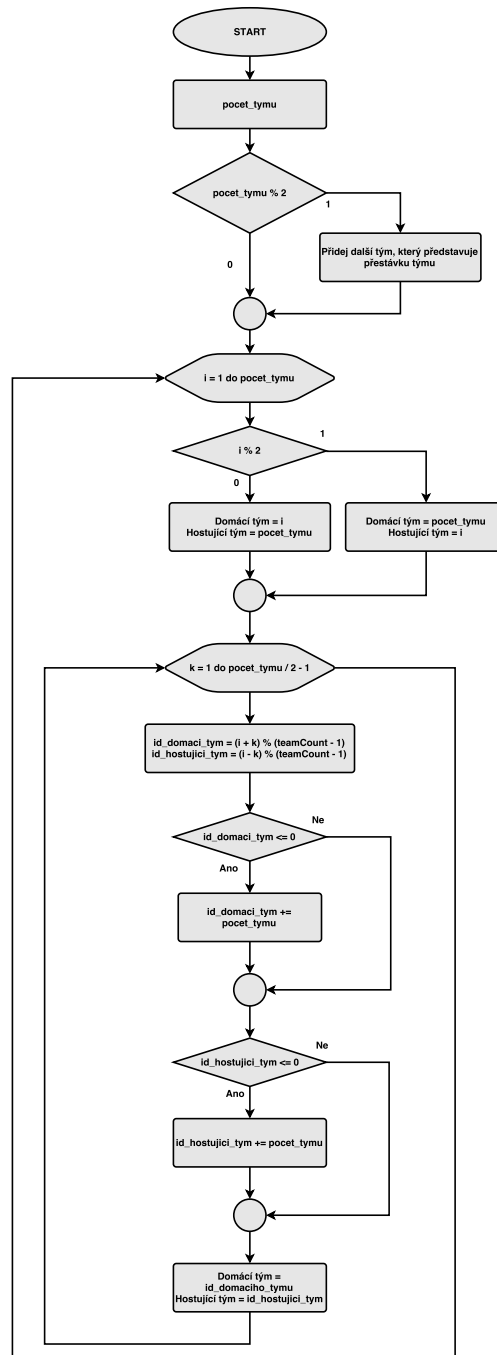
- [1] DOYLE, Matt. *Beginning PHP 5.3*. Indianapolis, IN: Wiley Pub., c2010. Wrox beginning guides. ISBN 978-0-470-41396-8.
- [2] DRAHOŠ, Zdeněk. *RESTful aplikace pro tvorbu a analýzu rozpisů sportovních zápasů*. 2014. Diplomová práce. Univerzita Pardubice, Fakulta elektrotechniky a informatiky. Vedoucí práce Hříděl Jan.
- [3] FREEMAN, Adam. *Pro ASP.NET MVC 5*. 5th ed. New York, N.Y.: Apress, 2013. ISBN 978-1-4302-6530-6.
- [4] FRONČEK, Dalibor. *Scheduling a Tournament* [online]. American Mathematical Society, 2010 [cit. 2017-04-10]. Dostupné z: <http://www.mathaware.org/mam/2010/essays/FroncekTournament.pdf>.
- [5] GONCALVES, Antonio. *Beginning Java EE 7*. Berkeley, CA: Apress, 2013. ISBN 978-1-4302-4627-5.
- [6] HARTL, Michael. *Ruby on rails tutorial: learn web development with rails*. Third edition. Addison-Wesley Professional, 2014. ISBN 978-0134077703.
- [7] HAVERBEKE, Marijn. *Eloquent javascript: a modern introduction to programming. Second edition*. San Francisco: No Starch Press, 2015. ISBN 978-1-59327-584-6.
- [8] JAVŮREK, Karel. *Český Livesport dává sportovní výsledky celému světu, obrat dosahuje na miliardu*. Živě.cz - Vše o počítačích, IT a internetu [online]. 2016 [cit. 2017-04-11]. Dostupné z: <https://connect.zive.cz/clanky/cesky-livesport-dava-sportovni-vysledky-celemu-svetu-obrat-dosahuje-na-miliardu/sc-320-a-181565/default.aspx>

- [9] KNUST, Sigrid. Scheduling of Tournaments or Sports Leagues. *Algorithms Unplugged* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, 267 - 275 [cit. 2013-10-12]. DOI: 10.1007/978-3-642-15328-0_27. Dostupné z: http://www.springerlink.com/index/10.1007/978-3-642-15328-0_27
- [10] KOŠEK, David. *Aplikace pro správu Univerzitní florbalové ligy*. 2014. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Lenc Ladislav.
- [11] MACDONALD, Matthew. *HTML5: the missing manual. Second edition*. O'Reilly Media, 2013. ISBN 978-1-4493-6326-0.
- [12] SCARIONI, Carlo a Manuel Jordan Elera. TECHNICAL REVIEWER. *Pro Spring security*. New York: Apress, 2013. ISBN 978-1-430-24819-4.
- [13] SOPKOVÁ, Magdaléna. *Využití metod operačního výzkumu ři tvorbě rozpisů sportovních zápasů*. 2015. Diplomová práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky. Vedoucí práce Jablonský Josef.
- [14] *AngularJS – Superheroic JavaScript WWW framework* [online]. 2010. [cit. 2017-05-15]. Dostupné z <https://angularjs.org/>.
- [15] HTML & CSS - W3C. *World Wide Web Consortium (W3C)* [online]. [cit. 2017-04-11]. Dostupné z: <https://www.w3.org/standards/webdesign/htmlcss>
- [16] Node.js Introduction. *Tutorials point: Simply Easy Learning* [online]. [cit. 2017-04-11]. Dostupné z: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm
- [17] Hibernate architecture. *Tutorials point: Simply Easy Learning* [online]. [cit. 2017-04-18]. Dostupné z: https://www.tutorialspoint.com/hibernate/hibernate_architecture.htm
- [18] O nás - slpf.cz. *Školní liga futsalu Plzeň - slpf.cz* [online]. [cit. 2017-04-11]. Dostupné z: <https://www.futsal.zcu.cz/o-nas>
- [19] *Livesport.cz* [online]. [cit. 2017-04-11]. Dostupné z: <http://www.livesport.cz/>

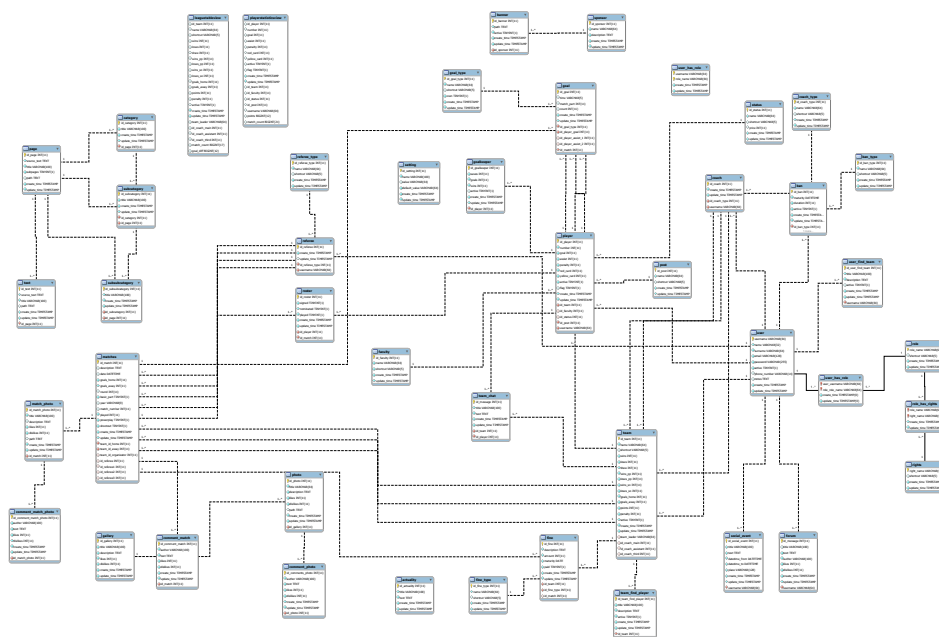
-
- [20] Reference model depicting the overall framework architecture. In: *Adrian Meija Blog* [online]. [cit. 2017-04-11]. Dostupné z: http://adrianmejia.com/images/rails_arch.png
- [21] Ruby on Rails Guides: Začínáme s Rails. *RubyOnRails.cz* [online]. [cit. 2017-04-10]. Dostupné z: <http://guides.rubyonrails.cz/>
- [22] *Sports League Management Software & Online Registration* [online]. [cit. 2017-04-10]. Dostupné z: <http://activesports.com/>
- [23] *ŠUMAVSKÁ LIGA AMATÉRSKÉHO HOKEJE* [online]. [cit. 2017-04-10]. Dostupné z: <http://sumavska-liga-amaterskeho-hokeje.wgz.cz/>

Přílohy

A Vývojový diagram rozpisu zápasů

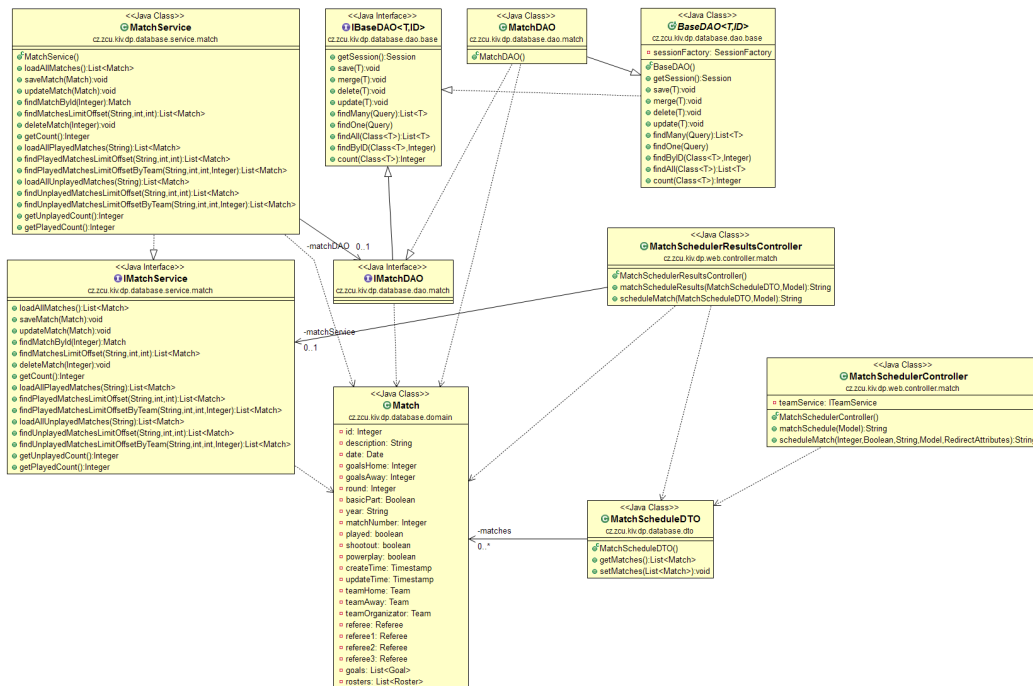


B Databázový model



Obrázek B.1: Schéma databáze

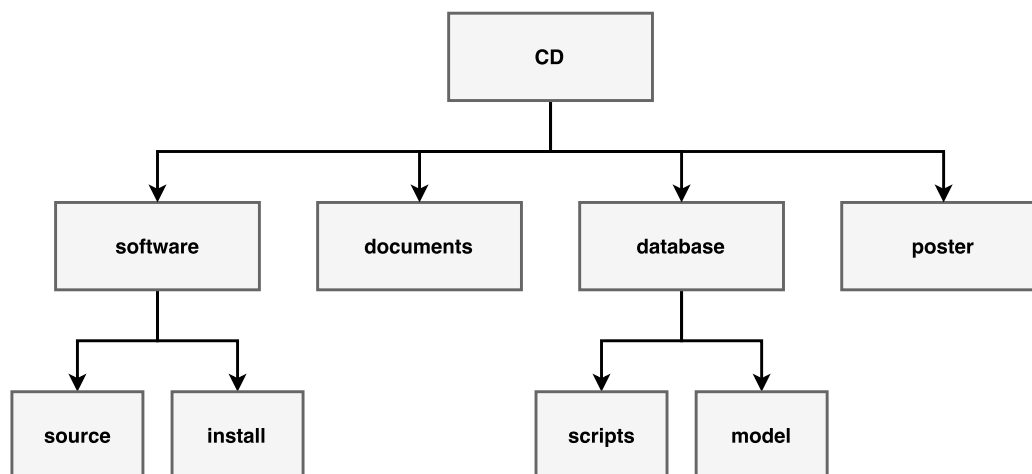
C UML diagram



Obrázek C.1: UML diagram

D Struktura CD

V příloze je popsána kompletní struktura CD, které je přiložené v diplomové práci. Na obrázku D.1 můžete vidět schématické znázornění struktury.



Obrázek D.1: Struktura CD

Jednotlivé části obsahují:

- **software**
 - **source** – veškeré zdrojové soubory vytvořené při implementaci aplikace.
 - **install** – přeložený soubor `.war`, který je možné rovnou nasadit na webový server.
- **documents**
 - **Sobehart_Michel_DP.pdf** – PDF soubor s kompletním textem diplomové práce.
 - **Sobehart_Michel_DP_schema.pdf** – PDF soubor se schématem databáze.
- **database**
 - **create_database.sql** – skript pro vytvoření struktury databáze.

- **base_data.sql** – skript pro vložení základních dat do databáze - jsou pro běh potřebné.
- **test_data.sql** – skript pro vložení testovacích dat do databáze.
- skripty pro tvorbu databáze a naplnění daty, model vytvořený v programu MySQL Workbench
- **poster**
 - **Sobehart_Michel_poster.pub** – poster diplomové práce ve formátu PUB.
 - **Sobehart_Michel_poster.pdf** – poster diplomové práce vyexportovaný do formátu PDF.

E Instalační příručka

Instalační příručka slouží uživateli k nasazení systému na komerční nebo testovací prostředí. Pro nasazení aplikace jsou zapotřebí tyto nástroje:

- **Java JDK 1.8.xx**
- **Apache Maven**
- **MySQL**
- **Apache Tomcat 8**

E.1 Konfigurace aplikace

V aplikaci je nutné nakonfigurovat připojení k databázi. Veškeré nastavení připojení se provádí v souboru `jdbc.properties`. Je důležité provést tři základní nastavení:

- `jdbc.databaseurl` – URL k databázi. Ponechte `jdbc:mysql://`, poté nastavte adresu, kde běží mysql server a název databáze (v mém případě futsal). Nakonec ponechte nastavení kódování.
- `jdbc.username` – uživatelské jméno k přístupu do databáze.
- `jdbc.password` – heslo k přístupu do databáze.

Ve zdrojovém kódu E.1 můžete vidět základní nastavení databáze.

```
jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.dialect=org.hibernate.dialect.MySQLDialect
jdbc.databaseurl=jdbc:mysql://localhost:3306/futsal?
    characterEncoding=utf-8
jdbc.username=root
jdbc.password=
jdbc.show_sql=true
```

Zdrojový kód E.1: Ukázka nastavení databáze pomocí konfiguračního souboru

Druhou nutnou konfigurací je nastavení cesty k uploadu souborů. Aplikace musí mít na složku práva zápisu. Cesta ke složce se nastavuje v souboru `upload.properties`. Klíč je `upload.path`. Stejná `path` se musí nastavit také v souboru `server.xml` v konfiguraci Apache Tomcat webového serveru. Je nutné dodat řádku:

```
<Context docBase="cesta_ke_složce"path="/SportWebApp/images"/>
```

E.2 Překlad aplikace

Celá aplikace je zabalena v balíka `.war`, který lze rovnou nasadit na server. Pokud chceme aplikaci přeložit, využijeme k tomu nástroje Apache Maven. Pomocí příkazu `mvn install` ve složce se zdrojovými kódy je možné aplikaci přeložit. Ta se zabalí do `.war` souboru, který se pak pouze překopíruje na webový server.

E.3 Databáze

Aplikace je nastavena na přístup k databázi MySQL na standartním portu 3306, na localhostu. Výchozí uživatelské jméno je `root` a heslo prázdné. Na MySQL server provedete pomocí DDL `create_database.sql` vytvoření databáze a pomocí `base_data.sql` vložení řádek do databáze.

F Uživatelská příručka

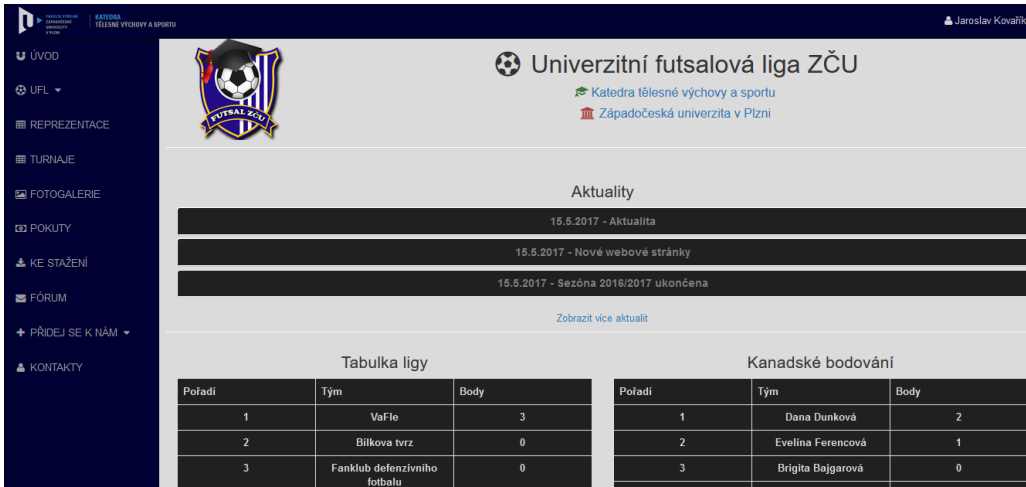
Uživatelská příručka slouží uživateli k zjištění informací o ovládání vytvořeného programu.

F.1 Informativní část

V této sekci je popsáno ovládání informativní části aplikace.

F.1.1 Úvodní stránka aplikace

Úvodní stránka aplikace zobrazuje přehled o lize, ligovou tabulku, kanadské bodování, aktuality, a také obrázky z hlavní galerie. Po levé straně je menu. To umožňuje dostat se kliknutím do dalších částí aplikace.



The screenshot shows the homepage of the University Futsal League ZČU. The page layout includes a dark blue sidebar menu on the left with the following items: ÚVOD, UFL, REPREZENTACE, TURNAJE, FOTOGALERIE, POKUTY, KE STAŽENÍ, FÓRUM, PŘIJEJ SE K NĀM, and KONTAKTY. The main content area has a header with the league logo and name 'Univerzitní futsalová liga ZČU', along with the text 'Katedra tělesné výchovy a sportu' and 'Západočeská univerzita v Plzni'. Below the header is a section titled 'Aktuality' with three news items: '15.5.2017 - Aktualita', '15.5.2017 - Nové webové stránky', and '15.5.2017 - Sezóna 2016/2017 ukončena'. At the bottom of the page are two tables: 'Tabulka ligy' and 'Kanadské bodování'.

Pořadí	Tým	Body
1	VaFle	3
2	Bilkova tvrz	0
3	Fanklub defenzivního fotbalu	0

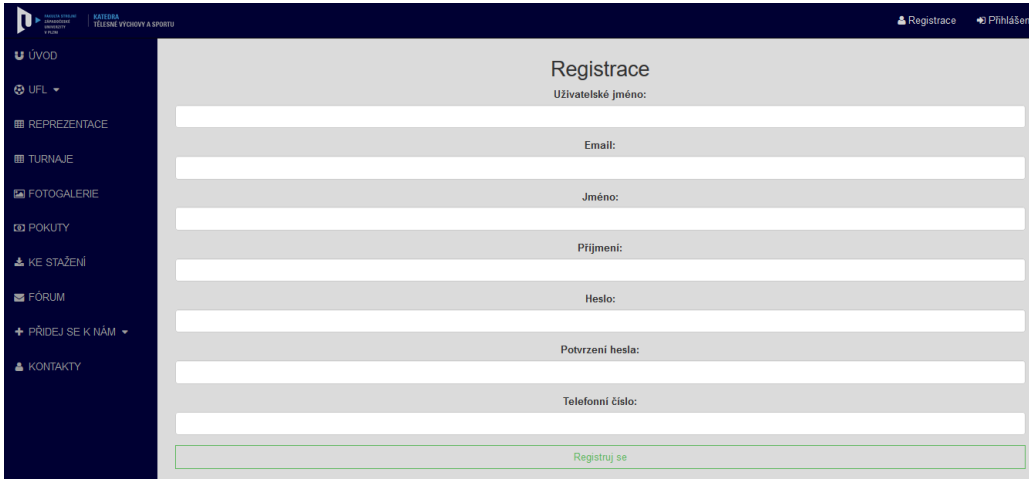
Pořadí	Tým	Body
1	Dana Dunková	2
2	Evelína Ferencová	1
3	Brigita Bajgarová	0

Obrázek F.1: Úvodní stránka aplikace

F.1.2 Registrace na stránky

Pro registraci na stránky slouží v pravé horním rohu tlačítko „Registrace“. Po kliknutí na něj se zobrazí registrační formulář, který můžete vidět na obrázku

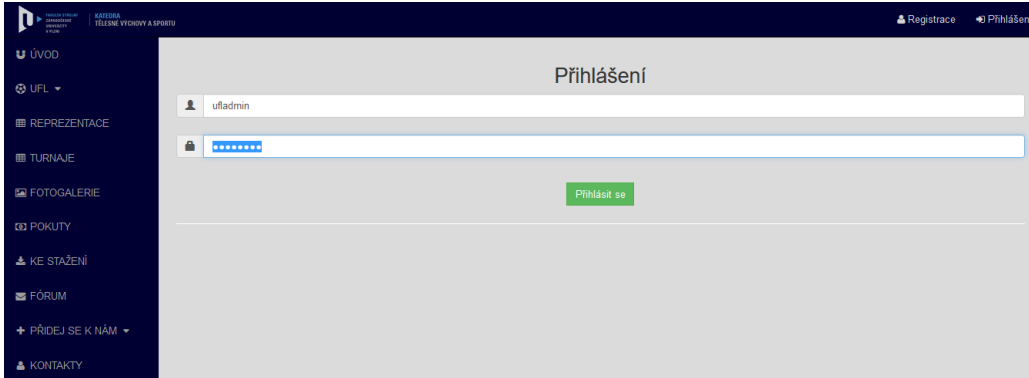
F.2. Ihned po úspěšné registraci jste přesunuti na stránku s přihlášením a profil je aktivní.

The image shows a registration form on a website. The form is titled "Registrace" and is located on a page with a dark blue header and a sidebar. The sidebar contains a navigation menu with items like "ÚVOD", "UFL", "REPREZENTACE", "TURNAJE", "FOTOGALERIE", "POKUTY", "KE STAŽENÍ", "FÓRUM", "PŘIDEJ SE K NÁM", and "KONTAKTY". The registration form itself has a light gray background and contains several input fields: "Uživatelské jméno:", "Email:", "Jméno:", "Příjmení:", "Heslo:", "Potvrzení hesla:", and "Telefonní číslo:". A green button labeled "Registruj se" is positioned at the bottom of the form. The top right corner of the page has links for "Registrace" and "Přihlášení".

Obrázek F.2: Registrační formulář

F.1.3 Přihlášení na stránky

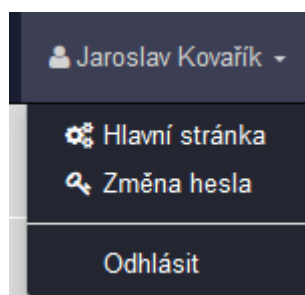
Pro přihlášení na stránky slouží v pravé horním rohu tlačítko „Přihlášení“. Po kliknutí na něj se zobrazí přihlašovací formulář, který můžete vidět na obrázku F.3.

The image shows a login form on a website. The form is titled "Přihlášení" and is located on a page with a dark blue header and a sidebar. The sidebar contains a navigation menu with items like "ÚVOD", "UFL", "REPREZENTACE", "TURNAJE", "FOTOGALERIE", "POKUTY", "KE STAŽENÍ", "FÓRUM", "PŘIDEJ SE K NÁM", and "KONTAKTY". The login form itself has a light gray background and contains two input fields: one for the username "uřadmin" and one for the password, which is masked with blue dots. A green button labeled "Přihlásit se" is positioned at the bottom of the form. The top right corner of the page has links for "Registrace" and "Přihlášení".

Obrázek F.3: Přihlašovací formulář

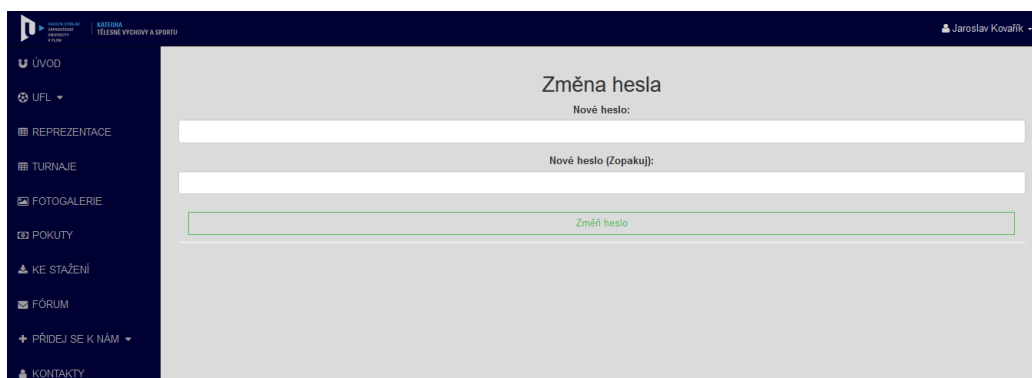
F.1.4 Změna hesla uživatele

Změna hesla se provádí pomocí kliknutí na tlačítko "Změna hesla" v pravém horním rohu stránky. Tlačítko můžete vidět na obrázku F.21.



Obrázek F.4: Ukázka tlačítka pro změnu hesla

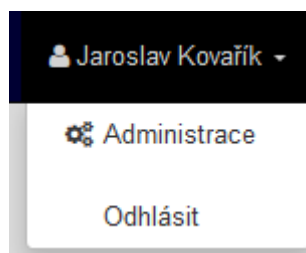
Po kliknutí na tlačítko jste přesměrováni na formulář ke změně hesla. Obě dvě hesla musí být samozřejmě stejná. Formulář můžete vidět na obrázku F.5.

The screenshot shows a web page titled 'Změna hesla'. On the left is a dark blue sidebar with navigation links: ÚVOD, UFL, REPREZENTACE, TURNAJE, FOTOGALERIE, POKUTY, KE STAŽENÍ, FÓRUM, PŘIDEJ SE K NÁM, and KONTAKTY. The main content area has a light gray background. At the top right of the main area, the user name 'Jaroslav Kovařík' is displayed. The form contains two input fields: 'Nové heslo:' and 'Nové heslo (Zopakuj):'. Below these fields is a green button labeled 'Změň heslo'.

Obrázek F.5: Formulář pro změnu hesla

F.1.5 Přepnutí do administrace ligy

Přepnutí do administrace ligy se provádí pomocí kliknutí na tlačítko "Administrace" v pravém horním rohu stránky. Tlačítko můžete vidět na obrázku F.6.



Obrázek F.6: Ukázka tlačítka pro změnu hesla

F.1.6 Rozpis zápasů

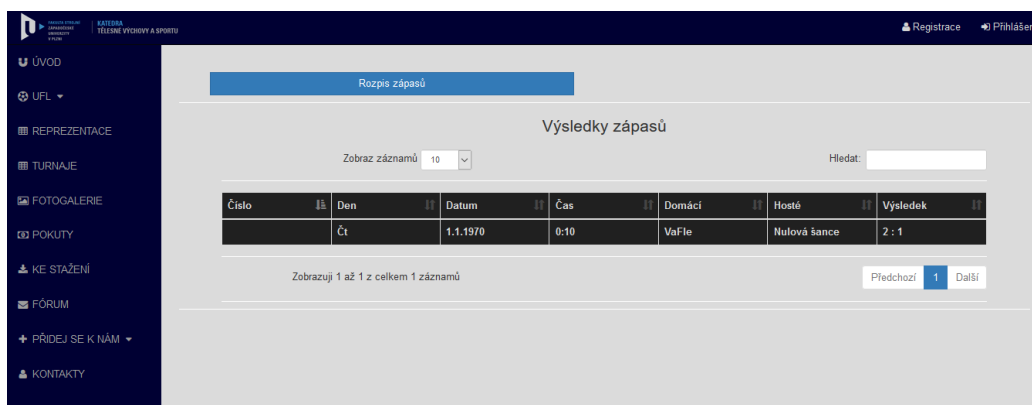
Kliknutím na tlačítko „UFL → Rozpis zápasů“ se dostanete na tabulku rozpisu zápasů. Tabulka lze řadit podle všech polí a filtrovat pomocí uživatelského vstupu „Hledat“. Dále lze také zvolit množství dat zobrazených na stránce. V posledním sloupci se nachází odkaz na tisk zápisu o utkání. Obrázek můžete vidět na obrázku F.7.

Číslo	Den	Datum	Čas	Domáci	Hosté	Pofadatel	Zápis
1				Kluci z plakátu	The Losers	VaFle	Tisk
2				Bílková tvrz	VaFle	The Losers	Tisk
3				The Wankers	Nulová šance	Old Boys Plzeň	Tisk
4				Výběr okresu	Old Boys Plzeň	Nulová šance	Tisk
5				Galantní Jeleni	Žlutý delfin	Fanklub defenzivního fotbalu	Tisk
6				VaFle	Fanklub defenzivního fotbalu	Žlutý delfin	Tisk
7				Nulová šance	Kluci z plakátu	Bílková tvrz	Tisk
8				Old Boys Plzeň	Bílková tvrz	Kluci z plakátu	Tisk

Obrázek F.7: Rozpis zápasů

F.1.7 Výsledky zápasů

Kliknutím na tlačítko „UFL → Výsledky zápasů“ se dostanete na tabulku výsledků zápasů. Tabulka lze řadit podle všech polí a filtrovat pomocí uživatelského vstupu „Hledat“. Dále lze také zvolit množství dat zobrazených na stránce. Obrázek můžete vidět na obrázku F.8.



Obrázek F.8: Výsledky zápasů

F.1.8 Tabulka ligy

Kliknutím na tlačítko „UFL → Tabulka ligy“ se dostanete na tabulku ligy. Tabulka lze řadit podle všech polí a filtrovat pomocí uživatelského vstupu „Hledat“. Dále lze také zvolit množství dat zobrazených na stránce. Obrázovku můžete vidět na obrázku F.9.

Pořadí	Název týmu	Zápasů	Výhry	Remizy	Prohry	BR	+/-	Body
1	VaFle	1	1	0	0	2 : 1	1	3
2	Bilkova tyrz	0	0	0	0	0 : 0	0	0
3	Fanklub defenzivního fotbalu	0	0	0	0	0 : 0	0	0
4	Galantní Jeleni	0	0	0	0	0 : 0	0	0
5	Kluci z plakátu	0	0	0	0	0 : 0	0	0
6	Old Boys Plzeň	0	0	0	0	0 : 0	0	0
7	The Losers	0	0	0	0	0 : 0	0	0
8	The Wankers	0	0	0	0	0 : 0	0	0
9	Výběr okresu	0	0	0	0	0 : 0	0	0
10	Žlutý dellín	0	0	0	0	0 : 0	0	0
11	Nulová šance	1	0	0	1	1 : 2	-1	0

Obrázek F.9: Tabulka ligy

F.1.9 Kanadské bodování

Kliknutím na tlačítko „UFL → Kanadské bodování“, se dostanete na tabulku kanadského bodování. Tabulka lze řadit podle všech polí a filtrovat pomocí uživatelského vstupu „Hledat“. Dále lze také zvolit množství dat zobrazených na stránce. Obrázku můžete vidět na obrázku F.10.

Pořadí	Jméno	Příjmení	Tým	Zápasy	Góly	Asistence	Body	Pozice	PZ (%)
1	Dana	Dunková	VaFle	0	1	1	2	Brankář	0,000
2	Evelina	Ferencová	VaFle	1	1	0	1	Obránce	1,000
3	Brigita	Bajgarová	Fanklub defenzivního fotbalu	0	0	0	0	Útočník	0,000
4	Gizela	Černohousová	Fanklub defenzivního fotbalu	0	0	0	0	Útočník	0,000
5	Lumír	Česka	The Losers	0	0	0	0	Útočník	0,000
6	Tomáš	Daniel	The Losers	0	0	0	0	Útočník	0,000
7	Dobroslav	Dohnálek	The Losers	0	0	0	0	Obránce	0,000
8	Ljuba	Donátová	VaFle	1	0	0	0	Útočník	0,000
9	Jiřina	Drobná	VaFle	1	0	0	0	Obránce	0,000

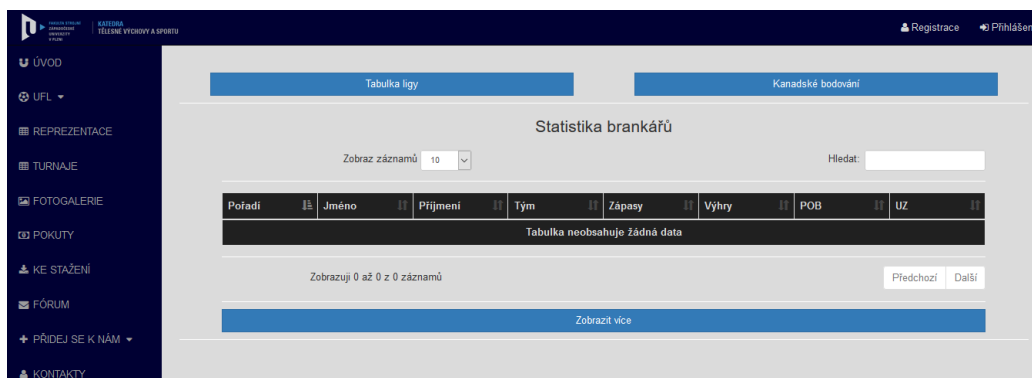
Obrázek F.10: Kanadské bodování

F.1.10 Brankářské statistiky

Kliknutím na tlačítko „UFL → Brankářské statistiky“, se dostanete na tabulku bodování brankářů. Tabulka lze řadit podle všech polí a filtrovat pomocí uživatelského vstupu „Hledat“. Dále lze také zvolit množství dat zobrazených na stránce. Obrázku můžete vidět na obrázku F.11.

F.1.11 Dynamický obsah

Kliknutím na tlačítko „REPREZENTACE → Muži“ se dostanete do informací o mužské reprezentaci. Sekce je vytvořena pomocí tvorby dynamického obsahu. Stejně tak i sekce „TURNAJE“. Obrázku můžete vidět na obrázku F.12.



Obrázek F.11: Brankářské statistiky



Obrázek F.12: Dynamický obsah - reprezentace

F.1.12 Fotogalerie

Kliknutím na tlačítko „FOTOGALERIE“ se dostanete do přehledu galerií s fotografiemi. U každé galerie je vidět počet líků a dislíků. Obrazovku můžete vidět na obrázku F.13.



Obrázek F.13: Fotogalerie

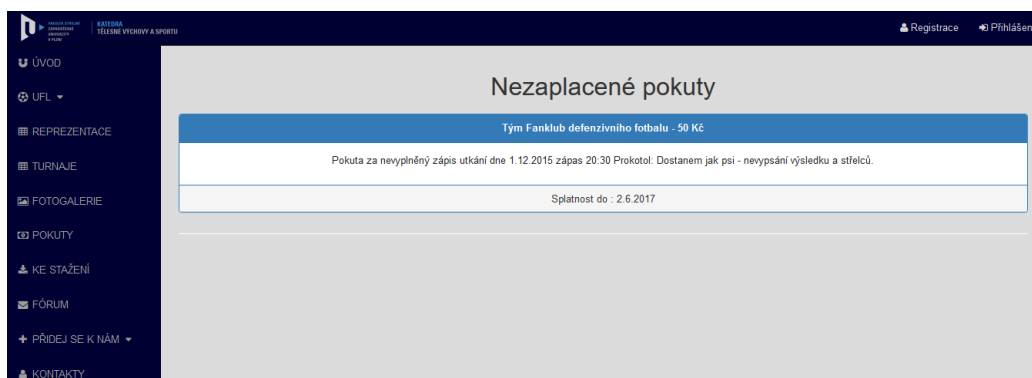
U každé fotogalerie najdete odkaz na zobrazení všech fotografií, které do galerie náleží. Po kliknutí na tento odkaz se přepneme do obrazovky s fotografiemi. To je možné vidět na obrázku F.14.



Obrázek F.14: Fotografie ve fotogalerii

F.1.13 Pokuty

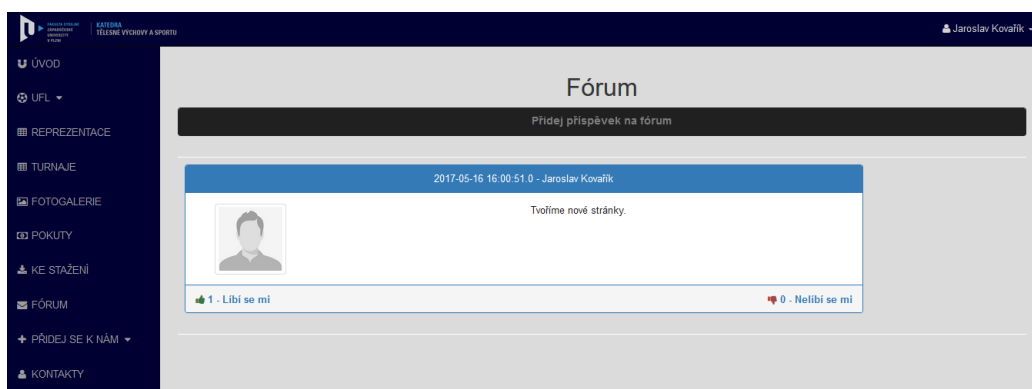
Kliknutím na tlačítko „POKUTY“ se dostanete do přehledu nezaplacených pokut. Obrazovku můžete vidět na obrázku F.15.



Obrázek F.15: Pokuty

F.1.14 Fórum

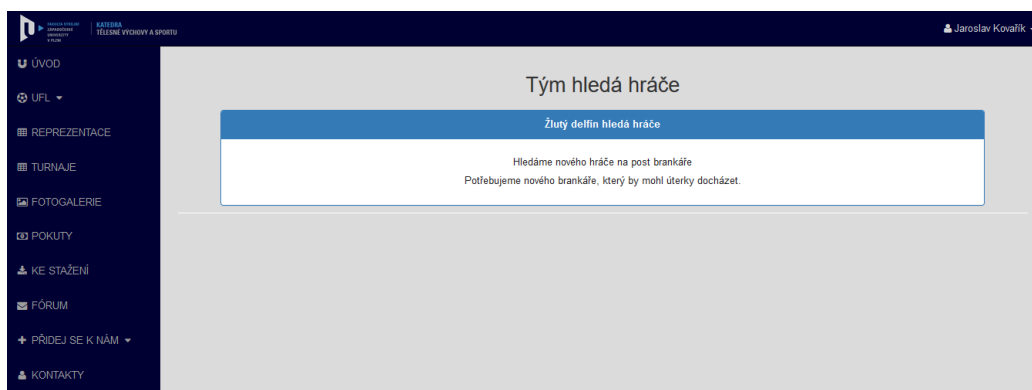
Kliknutím na tlačítko „FÓRUM“ se dostanete do globálního fóra ligy. Zde mohou všichni registrovaní uživatelé přidávat svoje příspěvky. Obrazovku můžete vidět na obrázku F.16.



Obrázek F.16: Fórum

F.1.15 Tým hledá hráče

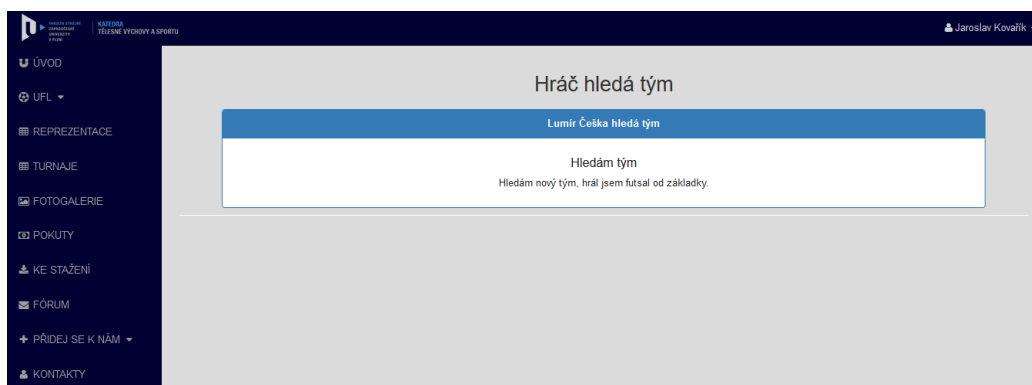
Kliknutím na tlačítko „PŘIDEJ SE K NÁM -> Tým hledá hráče“ se dostanete do sekce, kde jsou zobrazeny příspěvky týmů, které shánějí hráče. Obrazovku můžete vidět na obrázku F.17.



Obrázek F.17: Sekce tým hledá hráče

F.1.16 Hráč hledá tým

Kliknutím na tlačítko „PŘIDEJ SE K NÁM → Hráč hledá tým“ se dostanete do příspěvků uživatelů, kteří hledají nový tým a podali si tímto způsobem žádost. Popis by měl říct vše o hráči a také možnosti kontaktování. Obrazovku můžete vidět na obrázku F.18.



Obrázek F.18: Sekce hráč hledá tým

F.1.17 Ke stažení

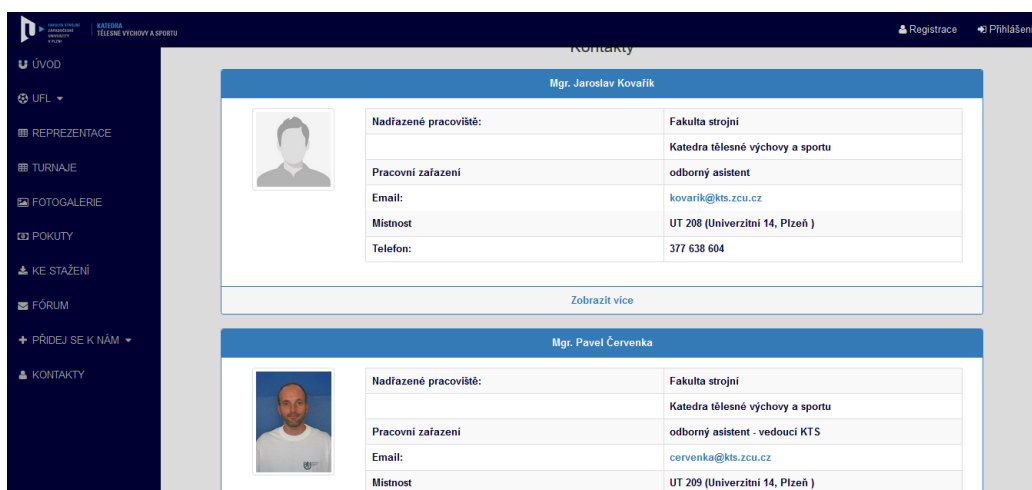
Kliknutím na tlačítko „KE STAŽENÍ“ se dostanete do přehledu dokumentů, které si uživatel může stáhnout a jsou potřebné k určité části vedení nebo hraní ligy. Obrazovku můžete vidět na obrázku F.19.



Obrázek F.19: Ke stažení

F.1.18 Kontakty

Kliknutím na tlačítko „KONTAKTY“ se dostanete do přehledu kontaktů. Jsou zde uvedeni dva hlavní kontakty futsalové ligy, a to vedoucí ligy a vedoucí celé katedry. Obrazovku můžete vidět na obrázku F.20.



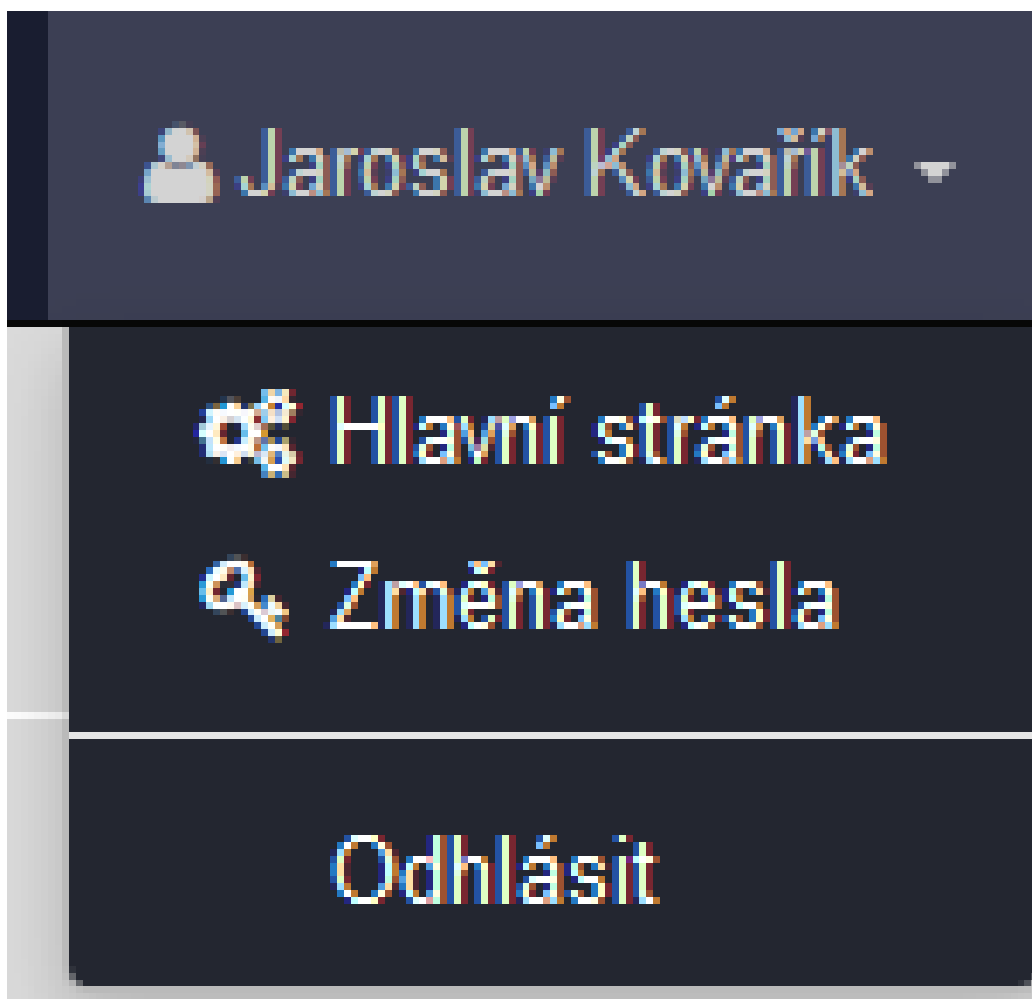
Obrázek F.20: Kontakty

F.2 Administrátorské prostředí

V této sekci je popsáno ovládání administrátorského prostředí aplikace.

F.2.1 Přepnutí zpět do informativní části

Přepnutí do informativní části lze pomocí tlačítka „Hlavní strana“ v možnostech uživatele v pravém horním rohu. Obrazovku hlavní strany můžete vidět na obrázku F.21.

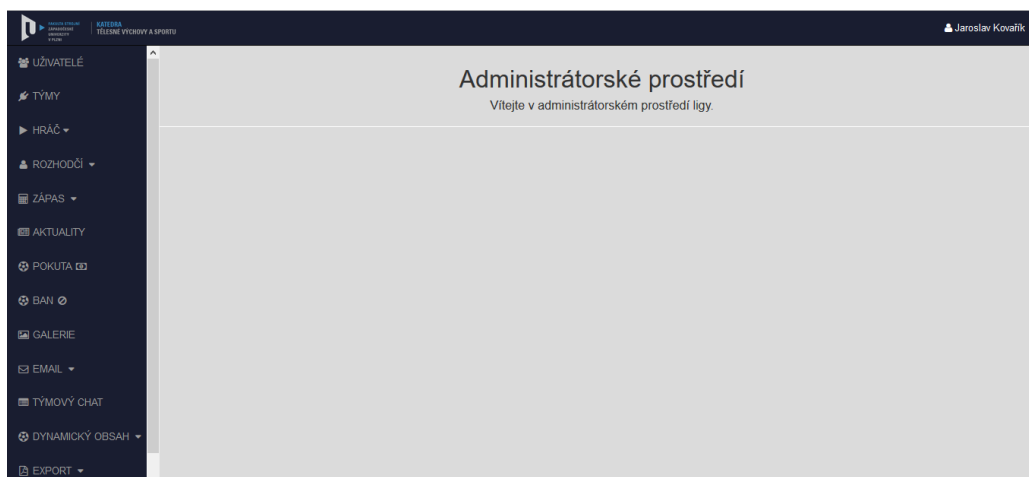


Obrázek F.21: Přístup zamítnut

F.2.2 Hlavní strana

Po přepnutí do administrace se dostanete na hlavní stranu administrátorského prostředí. Na obrazovce lze vidět levé menu, které obsahuje tlačítka

na přepnutí do ostatních administračních funkcí. Některá tlačítka jsou rozbalovací a proto u sebe mají šipku. Obrazovku hlavní strany můžete vidět na obrázku F.22.



Obrázek F.22: Obrazovka hlavní strany

F.2.3 Přístup zamítnut

Při pokusu uživatele o nepovolenou akci dojde k automatickému přesměrování na tuto stránku. Obrazovku hlavní strany můžete vidět na obrázku F.23.

F.2.4 Seznam uživatelů

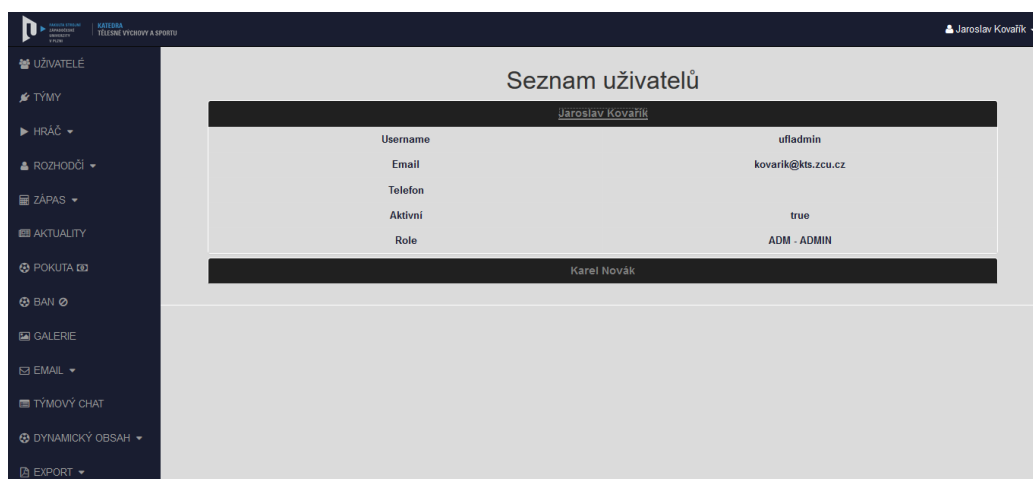
Kliknutím na tlačítko „UŽIVATELÉ“ se dostanete na seznam uživatelů. Slouží pouze jako informace pro administrátora ligy. Kliknutím na jméno uživatele se rozbalí tabulka s informacemi. Obrazovku můžete vidět na obrázku F.24.

F.2.5 Seznam týmů

Kliknutím na tlačítko „TÝMY“ se dostanete na seznam týmů. Kliknutím na jméno týmu se rozbalí tabulka s informacemi. Je možné týmy měnit, přidávat



Obrázek F.23: Přístup zamítnut

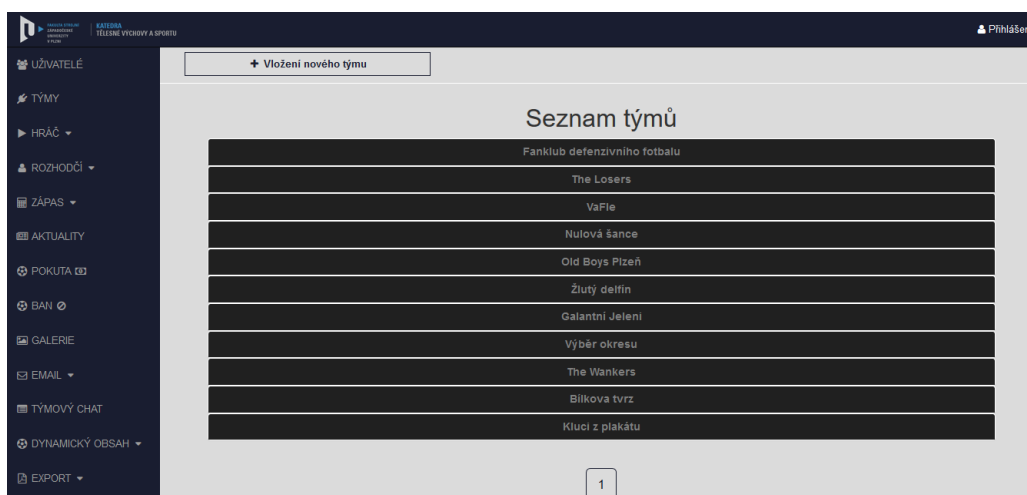


Obrázek F.24: Seznam uživatelů

nebo mazat. K tomu existují tlačítka:

- **Vložení nového týmu** - přepnutí do obrazovky pro úpravu nebo vložení nového týmu, uživatelské vstupy jsou prázdné.
- **Upravit tým** - přepnutí do obrazovky pro úpravu nebo vložení nového týmu, uživatelské vstupy jsou vyplněné podle upravovaného týmu.
- **Smazat tým** - vyskakovací okno pro potvrzení, následné smazání týmu.

Obrazovku můžete vidět na obrázku F.25.



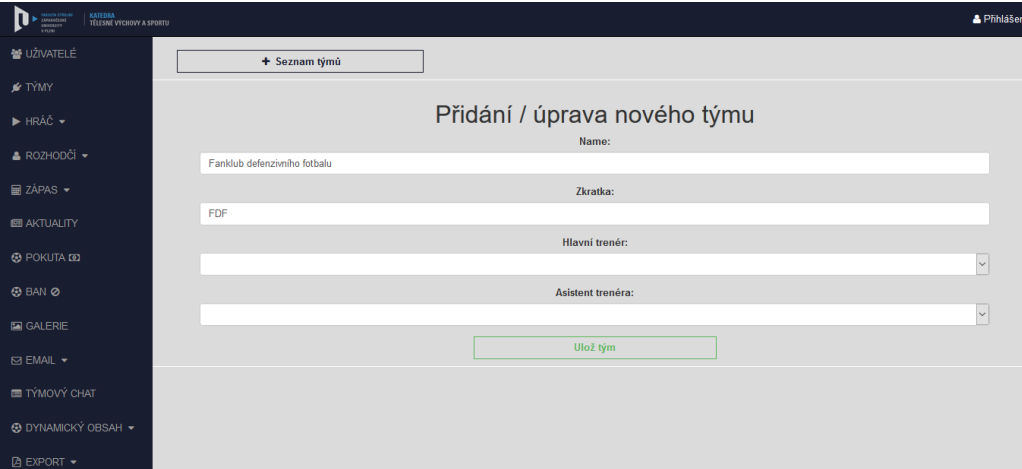
Obrázek F.25: Seznam týmů

F.2.6 Úprava a vložení týmu

Na obrazovku se dostanete pouze z obrazovky pro seznam týmů. Změny se projeví v systému po kliknutí na tlačítko „Ulož tým“. Obrazovku můžete vidět na obrázku F.26.

F.2.7 Seznam hráčů

Kliknutím na tlačítko „HRÁČI → Hráči“ se dostanete na obrazovku seznamu týmů. Kliknutím na jméno týmu se rozbalí tabulka s informacemi. Je možné týmy měnit, přidávat nebo mazat. K tomu existují tlačítka:



The screenshot shows a web application interface for managing teams. On the left is a dark sidebar menu with options: UŽIVATELE, TÝMY, HRÁČ, ROZHODČÍ, ZÁPAS, AKTUALITY, POKUTA, BAN, GALERIE, EMAIL, TÝMOVÝ CHAT, DYNAMICKÝ OBSAH, and EXPORT. The main content area is titled 'Přidání / úprava nového týmu' and contains a form with the following fields: 'Name' (filled with 'Fanklub defenzivního fotbalu'), 'Zkratka' (filled with 'FDF'), 'Hlavní trenér' (empty dropdown), and 'Asistent trenéra' (empty dropdown). A green 'Ulož tým' button is at the bottom of the form. A '+ Seznam týmů' button is located at the top of the main area.

Obrázek F.26: Úprava a vložení týmu

- **Přidání hráče** - přepnutí do obrazovky pro úpravu nebo vložení nového hráče, uživatelské vstupy jsou prázdné.
- **Upravit hráče** - přepnutí do obrazovky pro úpravu nebo vložení nového hráče, uživatelské vstupy jsou vyplněné podle upravovaného týmu.
- **Smazat hráče** - vyskakovací okno pro potvrzení, následné smazání hráče.

Obrazovku můžete vidět na obrázku F.27.



The screenshot shows the 'Seznam týmů' (Team List) view. At the top, there is a '+ Vložení nového týmu' button. The main area contains a table with the following data:

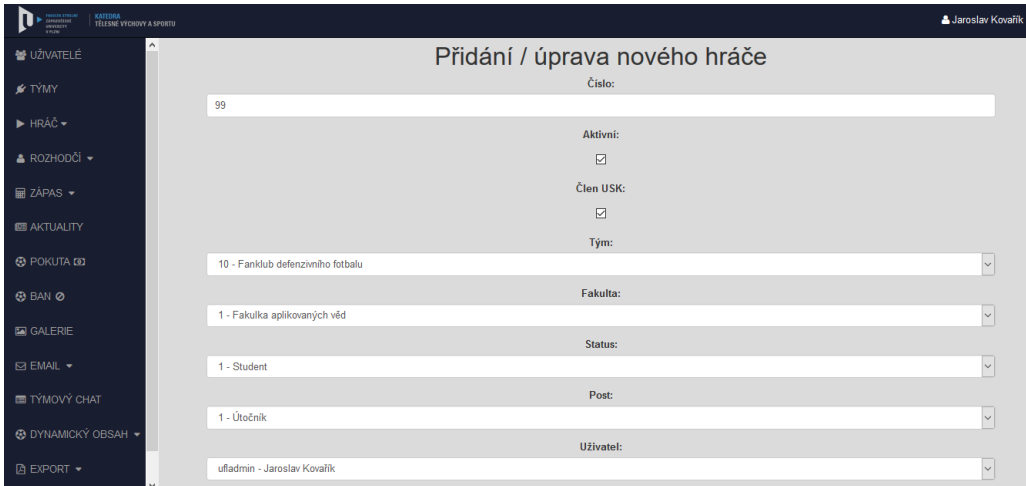
Fanklub defenzivního fotbalu
The Losers
VaFlie
Nulová šance
Old Boys Pizeň
Žlutý delphin
Galantní Jeleni
Výběr okresu
The Wankers
Blíkova tvrz
Kluci z plakátu

At the bottom of the table, there is a pagination control showing '1'.

Obrázek F.27: Seznam hráčů

F.2.8 Úprava a vložení hráče

Na obrazovku se dostanete pouze z obrazovky pro seznam hráčů. Změny se projeví v systému po kliknutí na tlačítko „Ulož hráče“. Obrazovku můžete vidět na obrázku F.28.



The screenshot shows a web form titled "Přidání / úprava nového hráče" (Add / edit new player). The form contains several input fields and checkboxes. On the left, there is a dark sidebar menu with options like "UŽIVATELE", "TÝMY", "HRÁČ", "ROZHODČÍ", "ZÁPAS", "AKTUALITY", "POKUTA", "BAN", "GALERIE", "EMAIL", "TÝMOVÝ CHAT", "DYNAMICKÝ OBSAH", and "EXPORT". The main form fields are: "Číslo:" (Number) with the value "99"; "Aktivní:" (Active) with a checked checkbox; "Člen USK:" (USK member) with a checked checkbox; "Tým:" (Team) with a dropdown menu showing "10 - Fanklub defenzivního fotbalu"; "Fakulta:" (Faculty) with a dropdown menu showing "1 - Fakulta aplikovaných věd"; "Status:" (Status) with a dropdown menu showing "1 - Student"; "Post:" (Position) with a dropdown menu showing "1 - Útočník"; and "Uživatel:" (User) with a dropdown menu showing "uřádní - Jaroslav Kovařík".

Obrázek F.28: Úprava a vložení hráče

F.2.9 Staň se hráčem

Na obrazovku se dostanete kliknutím na tlačítko „HRÁČI -> Staň se hráčem“. Změny se projeví v systému po kliknutí na tlačítko „Staň se hráčem“. Obrazovku můžete vidět na obrázku F.29.

F.2.10 Seznam rozhodčích

Kliknutím na tlačítko „ROZHODČÍ -> Rozhodčí“ se dostanete na obrazovku seznamu rozhodčích. Kliknutím na jméno rozhodčího se rozbalí tabulka s informacemi. Je možné rozhodčí měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidej nového rozhodčího** - přepnutí do obrazovky pro úpravu nebo vložení nového rozhodčího, uživatelské vstupy jsou prázdné.

The screenshot shows a web form for adding or editing a player. The form is titled 'Přidání / úprava nového hráče'. It contains several input fields and dropdown menus:

- Číslo: (empty text input)
- Člen USK: (checkbox, currently unchecked)
- Tým: (dropdown menu with '10 - Fanklub defenzivního fotbalu' selected)
- Fakulta: (dropdown menu with '1 - Fakulta aplikovaných věd' selected)
- Status: (dropdown menu with '1 - Student' selected)
- Post: (dropdown menu with '1 - Útočník' selected)

At the bottom of the form is a green button labeled 'Staň se hráčem'. The left sidebar contains navigation options: UŽIVATELE, TÝMY, HRÁČ, ROZHODČÍ, ZÁPAS, AKTUALITY, POKUTA, BAN, GALERIE, EMAIL, TÝMOVÝ CHAT, DYNAMICKÝ OBSAH, and EXPORT.

Obrázek F.29: Staň se hráčem

- **Upravit rozhodčího** - přepnutí do obrazovky pro úpravu nebo vložení nového rozhodčího, uživatelské vstupy jsou vyplněné podle upravovaného rozhodčího.
- **Smazat rozhodčího** - vyskakovací okno pro potvrzení, následné smazání rozhodčího.

Obrazovku můžete vidět na obrázku F.30.

The screenshot shows a table listing referees. The table has two buttons at the top: '+ Přidej nového rozhodčího' and '+ Přidej druh rozhodčího'. The table title is 'Seznam rozhodčích'. The table content is as follows:

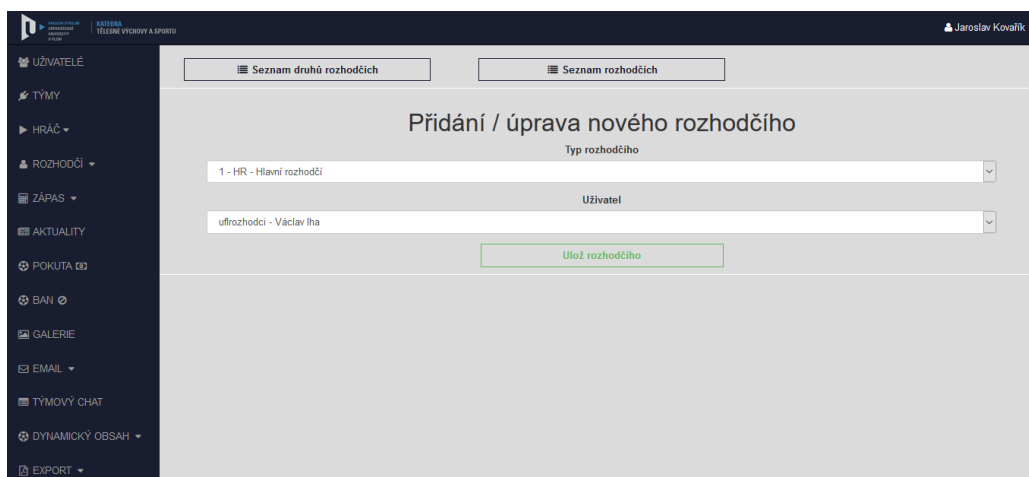
2 - Václav Ina	
Email	vaclaviha@seznam.cz
Aktivní	Ano
Typ rozhodčího	Hlavní rozhodčí
Datum vytvoření	2017-05-16 07:54:15.0
Datum aktualizace	2017-05-16 07:54:15.0

Below the table are two buttons: 'Upravit rozhodčího' (green) and 'Odstranit rozhodčího' (red). At the bottom center is a page number '1'.

Obrázek F.30: Seznam rozhodčích

F.2.11 Úprava a vložení rozhodčích

Na obrazovku se dostanete pouze z obrazovky pro seznam rozhodčích. Změny se projeví v systému po kliknutí na tlačítko „Ulož rozhodčího“. Obrazovku můžete vidět na obrázku F.31.



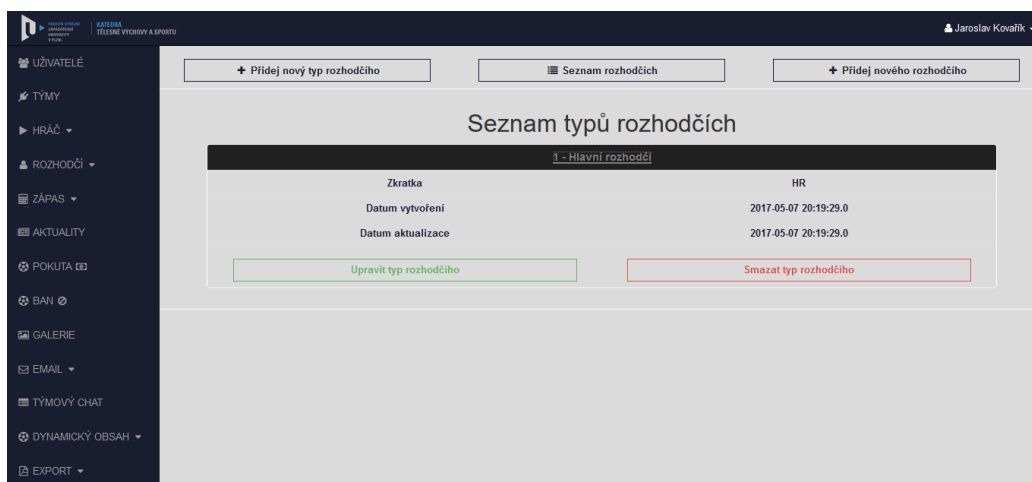
Obrázek F.31: Úprava a vložení rozhodčího

F.2.12 Seznam typů rozhodčích

Kliknutím na tlačítko „ROZHODČÍ ->Typ rozhodčích“ se dostanete na obrazovku seznamu typů rozhodčích. Kliknutím na jméno rozhodčího se rozbalí tabulka s informacemi. V databázi je uložena hodnota hlavního rozhodčího. Je možné rozhodčí měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidej nový typ rozhodčího** - přepnutí do obrazovky pro úpravu nebo vložení nového rozhodčího, uživatelské vstupy jsou prázdné.
- **Upravit typ rozhodčího** - přepnutí do obrazovky pro úpravu nebo vložení nového typu rozhodčího, uživatelské vstupy jsou vyplněné podle upravovaného typu rozhodčího.
- **Smazat typ rozhodčího** - vyskakovací okno pro potvrzení, následné smazání typu rozhodčího.

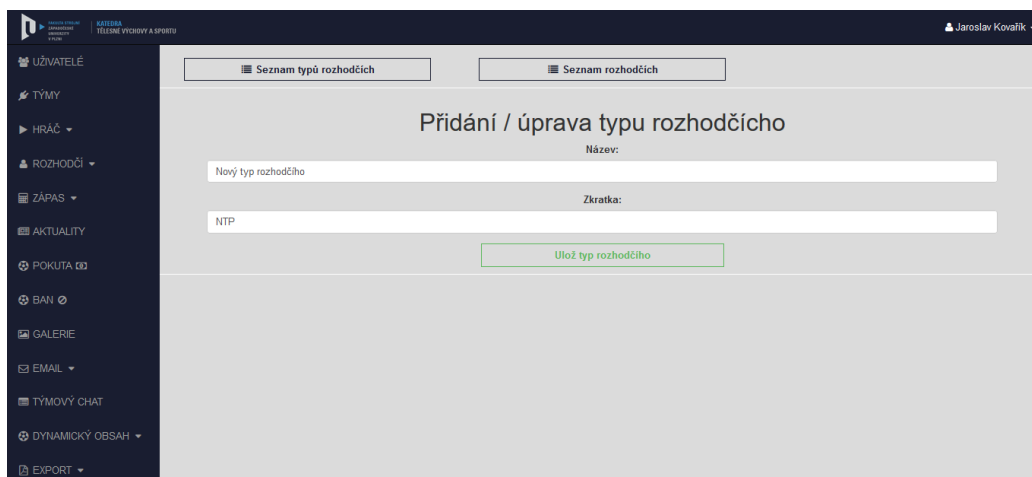
Obrazovku můžete vidět na obrázku F.32.



Obrázek F.32: Seznam typů rozhodčích

F.2.13 Úprava a vložení typu rozhodčích

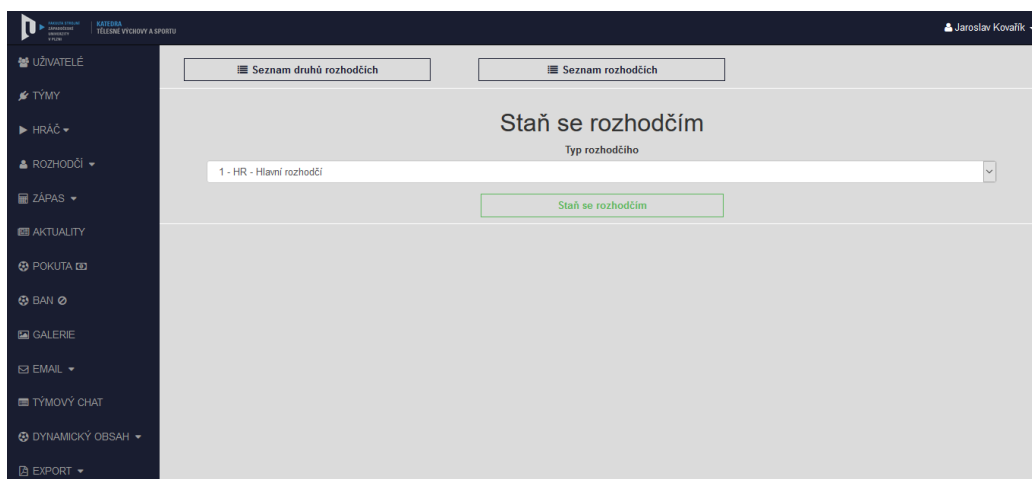
Na obrazovku se dostanete z obrazovky pro seznam typů rozhodčích nebo ze seznamu rozhodčích. Změny se projeví v systému po kliknutí na tlačítko „Ulož typ rozhodčího“. Obrazovku můžete vidět na obrázku F.33.



Obrázek F.33: Úprava a vložení typu rozhodčího

F.2.14 Staň se rozhodčím

Na obrazovku se dostanete kliknutím na tlačítko „ROZHODČÍ → Staň se rozhodčím“. Změny se projeví v systému po kliknutí na tlačítko „Staň se rozhodčím“. Obrazovku můžete vidět na obrázku F.34.



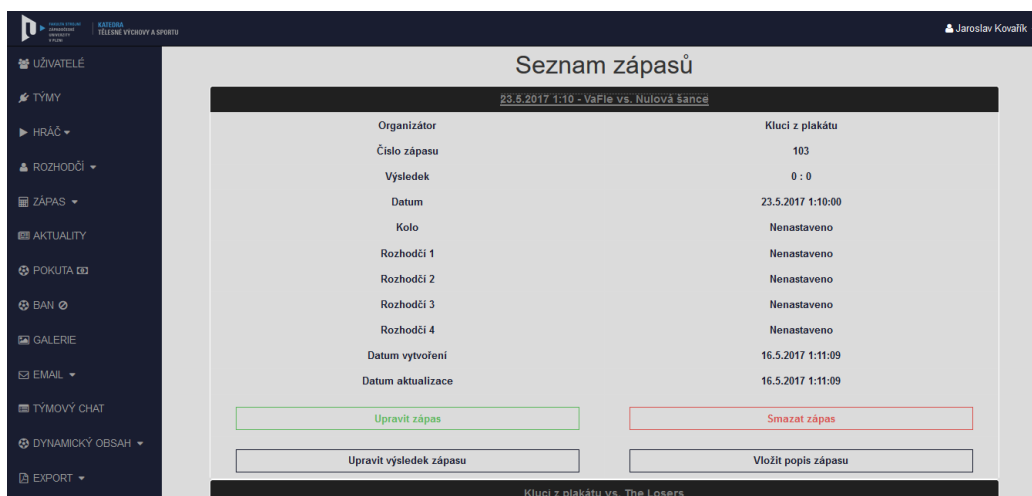
Obrázek F.34: Staň se rozhodčím

F.2.15 Seznam zápasů

Kliknutím na tlačítko „ZÁPAS → Zápasy“ se dostanete na obrazovku seznamu zápasů. Kliknutím na jméno zápasu se rozbalí tabulka s informacemi. Je možné zápas měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidej nový zápas** - přepnutí do obrazovky pro úpravu nebo vložení nového zápasu, uživatelské vstupy jsou prázdné.
- **Upravit zápas** - přepnutí do obrazovky pro úpravu nebo vložení nového zápasu, uživatelské vstupy jsou vyplněné podle upravovaného zápasu.
- **Smazat zápas** - vyskakovací okno pro potvrzení, následné smazání zápasu.

Obrazovku můžete vidět na obrázku F.35.



Obrázek F.35: Seznam zápasů

F.2.16 Seznam odehraných zápasů

Kliknutím na tlačítko „ZÁPAS → Odehrané zápasy“ se dostanete na obrazovku seznamu odehraných zápasů. Zápasy jsou zde filtrované, zobrazeny jsou odehrané zápasy. Obrazovka nabízí stejné možnosti jako obrazovka seznamu zápasů popsaná v kapitole F.2.15.

F.2.17 Seznam neodehraných zápasů

Kliknutím na tlačítko „ZÁPAS → Neodehrané zápasy“ se dostanete na obrazovku seznamu neodehraných zápasů. Zápasy jsou zde filtrované, zobrazeny jsou neodehrané zápasy. Obrazovka nabízí stejné možnosti jako obrazovka seznamu zápasů popsaná v kapitole F.2.15.

F.2.18 Přihlášení na zápasy

Kliknutím na tlačítko „ZÁPAS → Přihlášení na zápasy“ se dostanete na obrazovku, kde se hráč může přihlásit na zápasy jeho týmu. Na zápas je možné se přihlásit pomocí kliknutí na obrázek ANO nebo NE. Zvolená hodnota je zobrazena pomocí tučnějších barev. Na každý zápas lze kliknout a dostat se do seznamu hráčů, kteří jsou na zápas přihlášení. Obrazovku můžete vidět

na obrázku F.36.



Obrázek F.36: Přihlášení na zápas

F.2.19 Seznam přihlášených hráčů

Na obrazovku se dostanete pomocí kliknutí na daný zápas z obrazovky pro přihlášení na zápasy. Je zde vidět, kteří hráči jsou přihlášení na daný zápas, a kteří ne. Obrazovku můžete vidět na obrázku F.37.



Obrázek F.37: Seznam přihlášených hráčů na zápas

F.2.20 Rozpis zápasů

Kliknutím na tlačítko „ZÁPAS → Rozpis zápasů“ se dostanete na obrazovku, kde lze vytvořit rozpis zápasů. Zápasy se berou přímo z uložených týmů. Obrazovku můžete vidět na obrázku F.38.

Obrázek F.38: Tvorba rozpisu zápasů

Po kliknutí na tlačítko „Vytvoř rozpis“ se vytvoří zápasy rozpisu a přejde se na obrazovku, kde je možné výsledný rozpis vidět. Tuto obrazovku můžete vidět na obrázku F.39.

Číslo zápasu	Tým domácí	Tým hosté	Tým organizátor	Kolo
1	Kluci z plakátu	The Losers	VaFle	1
2	Bilkova tvrz	VaFle	The Losers	1
3	The Wankers	Nulová šance	Old Boys Plzeň	1
4	Výběr okresu	Old Boys Plzeň	Nulová šance	1
5	Galantni Jeleni	Žlutý delfin	Fanklub defenzivního fotbalu	1
6	VaFle	Fanklub defenzivního fotbalu	Žlutý delfin	2
7	Nulová šance	Kluci z plakátu	Bilkova tvrz	2
8	Old Boys Plzeň	Bilkova tvrz	Kluci z plakátu	2
9	Žlutý delfin	The Wankers	Výběr okresu	2
10	Galantni Jeleni	Výběr okresu	The Wankers	2
11	The Losers	Nulová šance	Old Boys Plzeň	3
12	Fanklub defenzivního fotbalu	Old Boys Plzeň	Nulová šance	3
13	Kluci z plakátu	Žlutý delfin	Galantni Jeleni	3
14	Bilkova tvrz	Galantni Jeleni	Žlutý delfin	3
15	The Wankers	Výběr okresu	VaFle	3

Obrázek F.39: Vytvořený rozpis zápasů

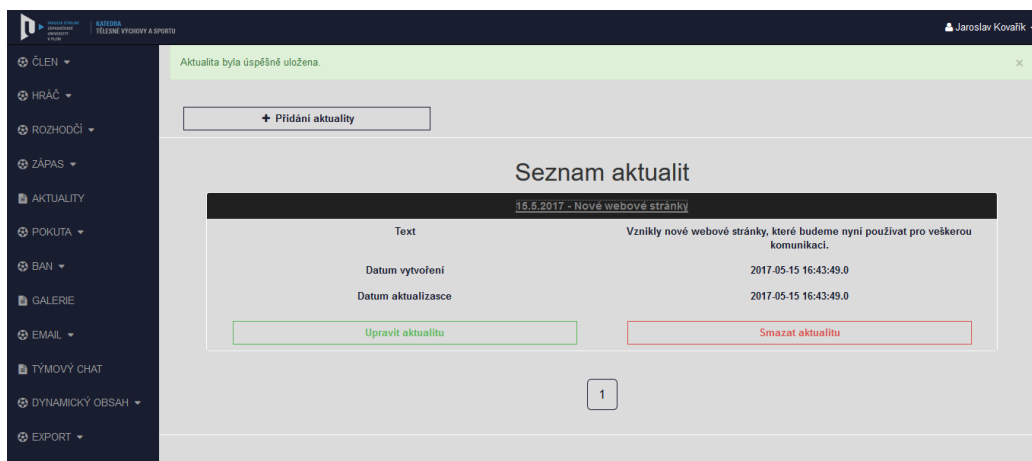
Po kliknutí na tlačítko „Uložit zápasy“ se zápasy uloží do databáze a tím je rozpis vytvořen.

F.2.21 Seznam aktualit

Kliknutím na tlačítko „AKTUALITY“ se dostanete na obrazovku seznamu aktualit. Kliknutím na jméno aktuality se rozbalí tabulka s informacemi. Je možné aktuality měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidání aktuality** - přepnutí do obrazovky pro úpravu nebo vložení nové aktuality, uživatelské vstupy jsou prázdné.
- **Upravit aktuality** - přepnutí do obrazovky pro úpravu nebo vložení nové aktuality, uživatelské vstupy jsou vyplněné podle upravované aktuality.
- **Smazat aktuality** - vyskakovací okno pro potvrzení, následné smazání aktuality.

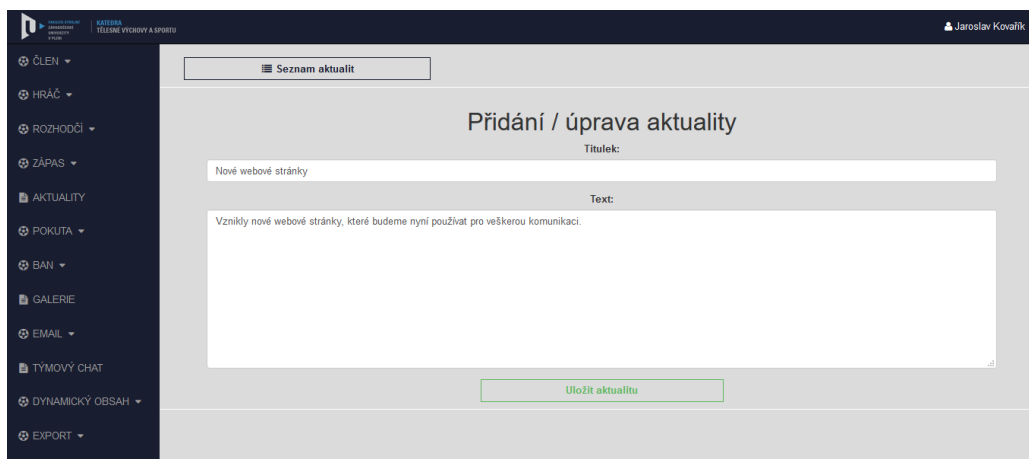
Obrazovku můžete vidět na obrázku F.40.



Obrázek F.40: Seznam aktualit

F.2.22 Úprava a vložení aktualit

Na obrazovku se dostanete pouze z obrazovky pro seznam aktualit. Změny se projeví v systému po kliknutí na tlačítko „Ulož aktualitu“. Obrazovku můžete vidět na obrázku F.41.



Obrázek F.41: Úprava a vložení aktuality

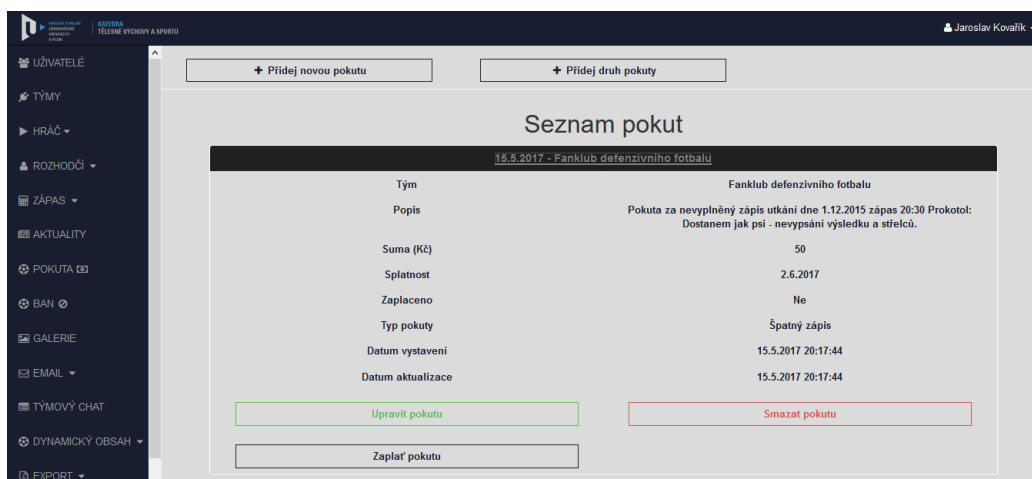
F.2.23 Seznam pokut

Kliknutím na tlačítko „POKUTA -> Pokuty“ se dostanete na obrazovku seznamu pokut. Kliknutím na jméno pokuty se rozbalí tabulka s informacemi. Je možné pokuty měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidej novou pokutu** - přepnutí do obrazovky pro úpravu nebo vložení nové pokuty, uživatelské vstupy jsou prázdné.
- **Upravit pokutu** - přepnutí do obrazovky pro úpravu nebo vložení nové pokuty, uživatelské vstupy jsou vyplněné podle upravované pokuty.
- **Smazat pokutu** - vyskakovací okno pro potvrzení, následné smazání pokuty.
- **Nastav pokutu na zaplacenou** - umožní nastavení pokuty jako zaplacené.

- **Nastav pokutu na nezaplacenou** - umožní nastavení pokuty jako nezaplacené.

Obrazovku můžete vidět na obrázku F.42.



Obrázek F.42: Seznam pokut

F.2.24 Úprava a vložení pokuty

Na obrazovku se dostanete pouze z obrazovky pro seznam pokut. Změny se projeví v systému po kliknutí na tlačítko „Ulož pokutu“. Obrazovku můžete vidět na obrázku F.43.

F.2.25 Seznam zaplacených pokut

Kliknutím na tlačítko „POKUTA → Zaplacené pokuty“ se dostanete na obrazovku seznamu zaplacených pokut. Pokuty jsou zde filtrované, zobrazeny jsou zaplacené pokuty. Obrazovka nabízí stejné možnosti jako obrazovka seznamu pokut popsána v kapitole F.2.29.



Obrázek F.43: Úprava a vložení pokuty

F.2.26 Seznam nezaplacených pokut

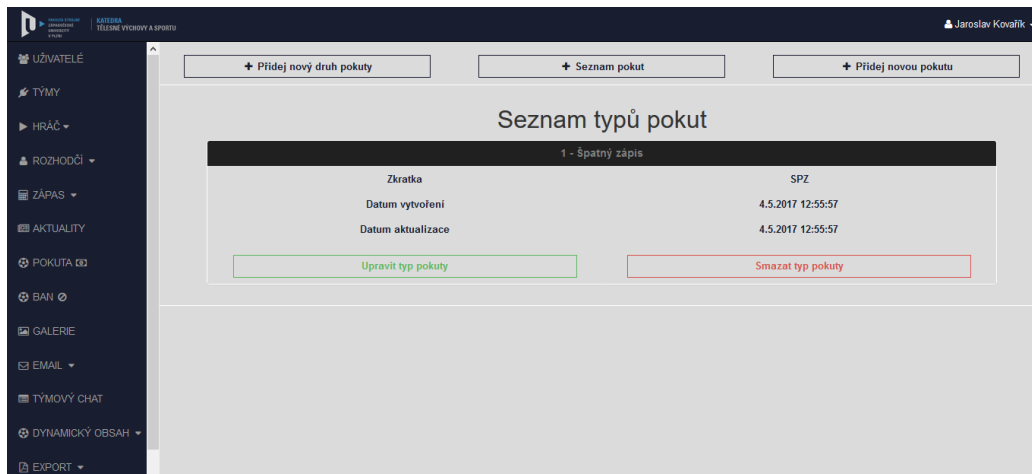
Kliknutím na tlačítko „POKUTA → Nezaplacené pokuty“ se dostanete na obrazovku seznamu nezaplacených pokut. Pokuty jsou zde filtrované, zobrazeny jsou nezaplacené pokuty. Obrazovka nabízí stejné možnosti jako obrazovka seznamu pokut popsána v kapitole F.2.29.

F.2.27 Seznam typů pokut

Kliknutím na tlačítko „POKUTA → Typ pokut“ se dostanete na obrazovku seznamu typů pokut. Kliknutím na jméno pokuty se rozbalí tabulka s informacemi. Je možné typy pokut měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidej nový typ pokuty** - přepnutí do obrazovky pro úpravu nebo vložení nového typu pokuty, uživatelské vstupy jsou prázdné.
- **Upravit typ pokuty** - přepnutí do obrazovky pro úpravu nebo vložení nového typu pokuty, uživatelské vstupy jsou vyplněné podle upravovaného typu pokuty.
- **Smazat typ pokuty** - vyskakovací okno pro potvrzení, následné smazání typu pokuty.

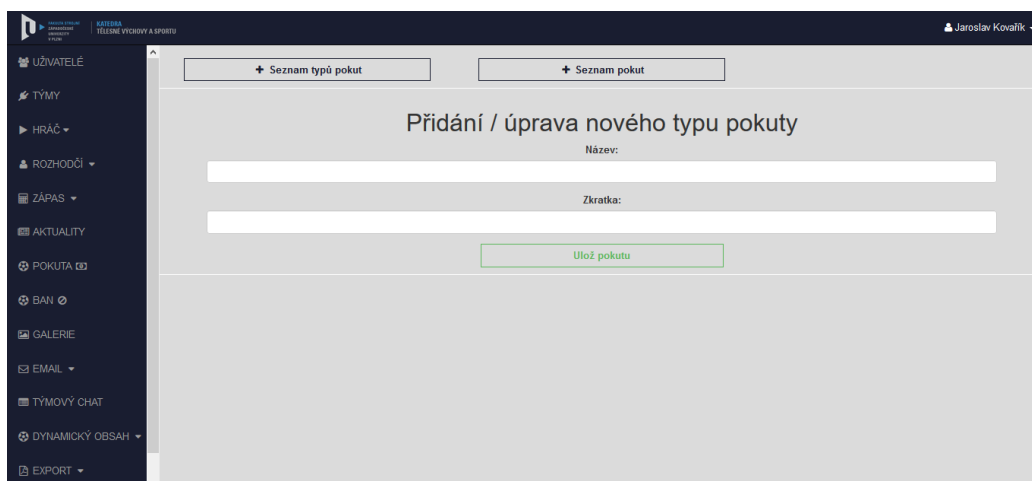
Obrázovku můžete vidět na obrázku F.44.



Obrázek F.44: Seznam typů pokut

F.2.28 Úprava a vložení typu pokut

Na obrazovku se dostanete z obrazovky pro seznam typů pokut nebo ze seznamu pokut. Změny se projeví v systému po kliknutí na tlačítko „Ulož typ pokuty“. Obrazovku můžete vidět na obrázku F.45.



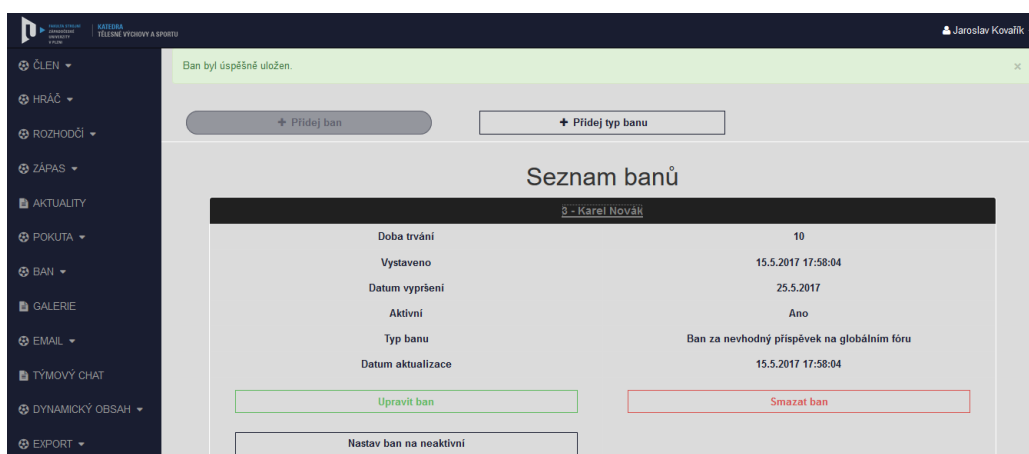
Obrázek F.45: Úprava a vložení typu pokuty

F.2.29 Seznam banů

Kliknutím na tlačítko „BAN → Bany“ se dostanete na obrazovku seznamu banů. Kliknutím na jméno banu se rozbalí tabulka s informacemi. Je možné bany měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidej nový ban** - přepnutí do obrazovky pro úpravu nebo vložení nového banu, uživatelské vstupy jsou prázdné.
- **Upravit ban** - přepnutí do obrazovky pro úpravu nebo vložení nového banu, uživatelské vstupy jsou vyplněné podle upravovaného banu.
- **Smazat ban** - vyskakovací okno pro potvrzení, následné smazání banu.

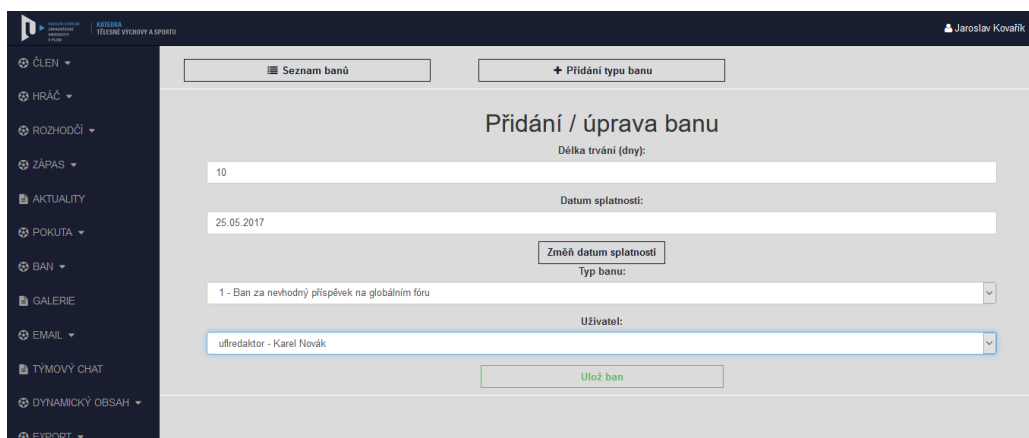
Obrazovku můžete vidět na obrázku F.46.



Obrázek F.46: Seznam banů

F.2.30 Úprava a vložení banu

Na obrazovku se dostanete pouze z obrazovky pro seznam banů. Změny se projeví v systému po kliknutí na tlačítko „Ulož ban“. Obrazovku můžete vidět na obrázku F.47.



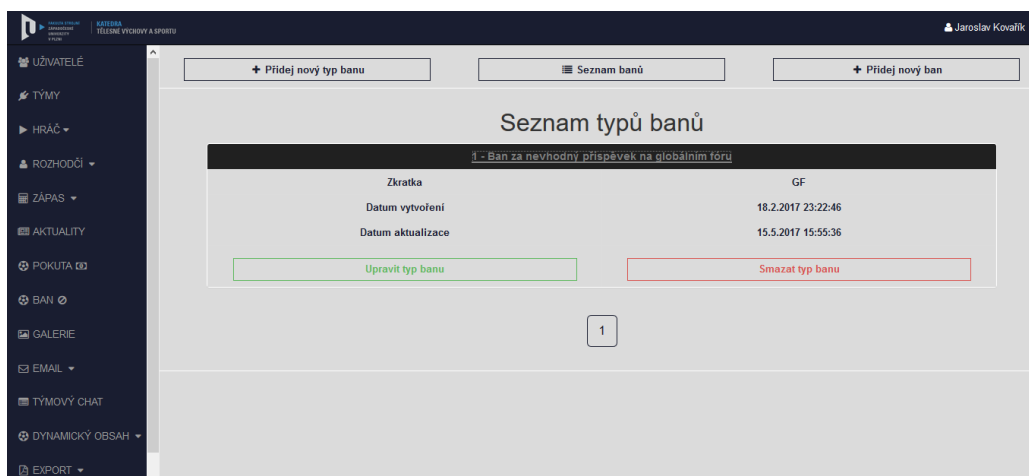
Obrázek F.47: Úprava a vložení banu

F.2.31 Seznam typů banů

Kliknutím na tlačítko „BAN → Typ banu“ se dostanete na obrazovku seznamu typů banů. Kliknutím na jméno pokuty se rozbálí tabulka s informacemi. Je možné typy banů měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidej nový typ banu** - přepnutí do obrazovky pro úpravu nebo vložení nového typu banu, uživatelské vstupy jsou prázdné.
- **Upravit typ banu** - přepnutí do obrazovky pro úpravu nebo vložení nového typu banu, uživatelské vstupy jsou vyplněné podle upravovaného typu banu.
- **Smazat typ banu** - vyskakovací okno pro potvrzení, následné smazání typu banu.
- **Nastav ban na aktivní** - umožní nastavení banu na aktivní.
- **Nastav ban na neaktivní** - umožní nastavení banu na neaktivní.

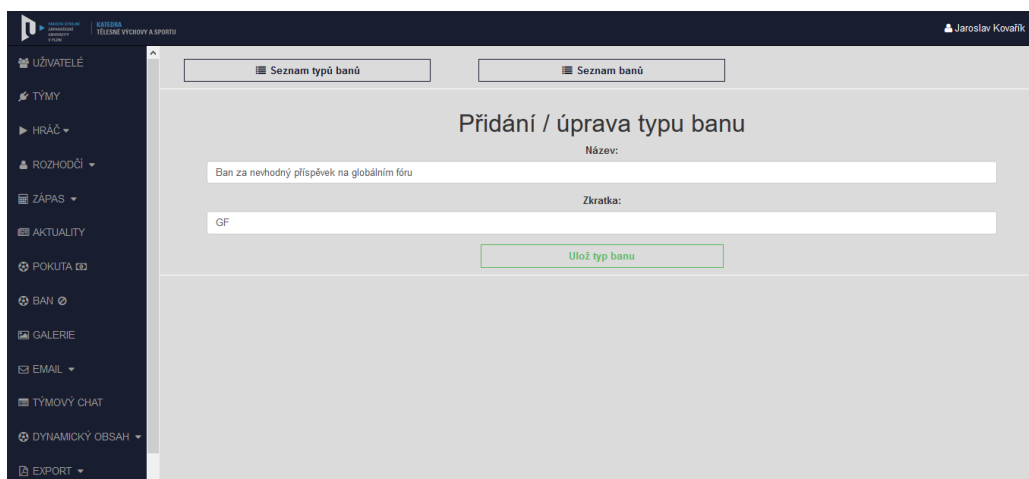
Obrazovku můžete vidět na obrázku F.48.



Obrázek F.48: Seznam typů banů

F.2.32 Úprava a vložení typu banů

Na obrazovku se dostanete z obrazovky pro seznam typů banů nebo ze seznamu banů. Změny se projeví v systému po kliknutí na tlačítko „Ulož typ banu“. Obrazovku můžete vidět na obrázku F.49.



Obrázek F.49: Úprava a vložení typu banu

F.2.33 Seznam galerií

Kliknutím na tlačítko „GALERIE“ se dostanete na obrazovku seznamu galerií. Kliknutím na jméno galerie se rozbalí tabulka s informacemi. Je možné galerie měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Vložení nové galerie** - přepnutí do obrazovky pro úpravu nebo vložení nové galerie, uživatelské vstupy jsou prázdné.
- **Upravit galerii** - přepnutí do obrazovky pro úpravu nebo vložení nové galerie, uživatelské vstupy jsou vyplněné podle upravované galerie.
- **Smazat galerii** - vyskakovací okno pro potvrzení, následné smazání galerie.
- **Přidat nebo odebrat fotografie** - přechod na obrazovku s úpravou fotek ve fotogalerii.

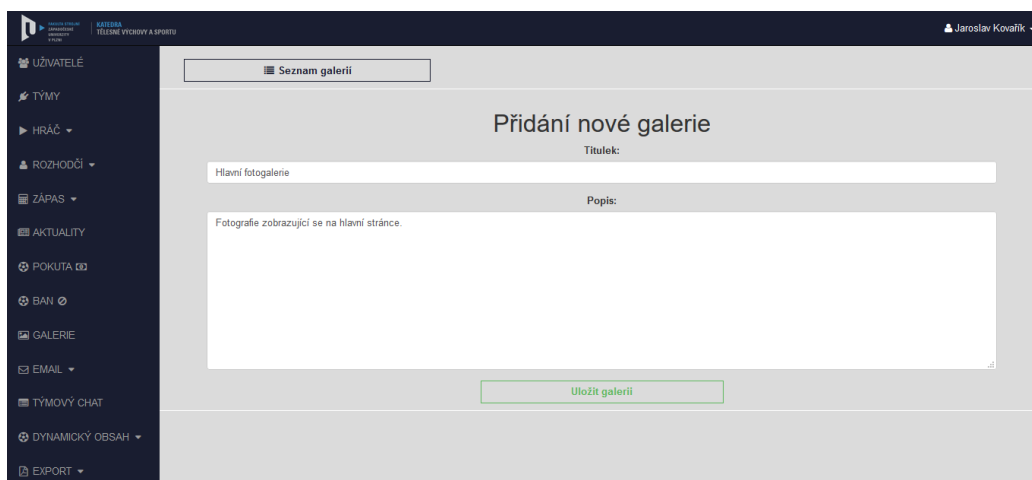
Obrazovku můžete vidět na obrázku F.50.



Obrázek F.50: Seznam galerií

F.2.34 Úprava a vložení galerie

Na obrazovku se dostanete pouze z obrazovky pro seznam galerií. Změny se projeví v systému po kliknutí na tlačítko „Ulož galerii“. Obrazovku můžete vidět na obrázku F.51.



Obrázek F.51: Úprava a vložení galerie

F.2.35 Přidání fotografií do galerie

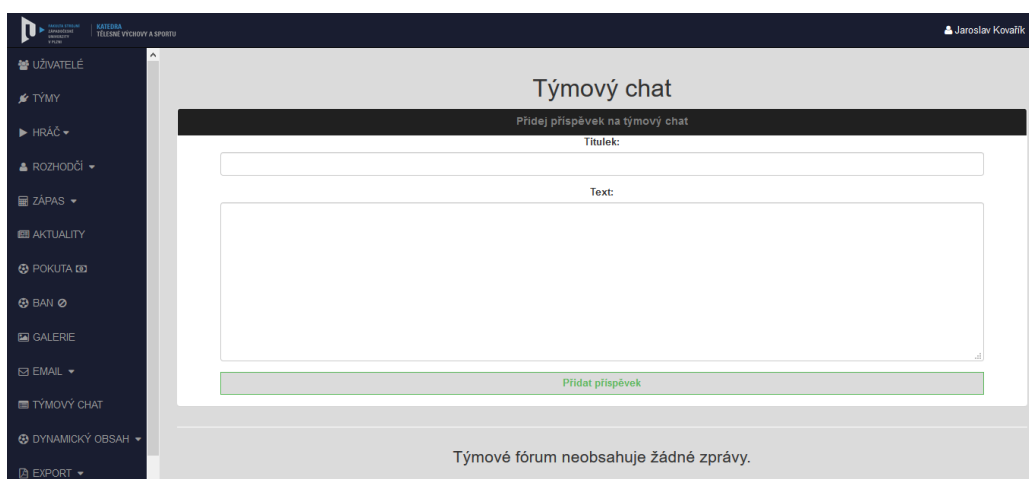
Na obrazovku se dostanete pouze z obrazovky pro seznam galerií. Změny se projeví v systému po kliknutí na tlačítko „Přidej fotografie“. Obrazovku můžete vidět na obrázku F.52.



Obrázek F.52: Přidání fotografií do galerie

F.2.36 Týmový chat

Kliknutím na tlačítko „TÝMOVÝ CHAT“ se dostanete na obrazovku týmového chatu. Týmový chat obsahuje pouze zprávy v týmu. Příspěvky může tedy přidávat pouze hráč týmu. Přidání příspěvku lze provést v horním formuláři. Obrazovku můžete vidět na obrázku F.53.



Obrázek F.53: Týmový chat

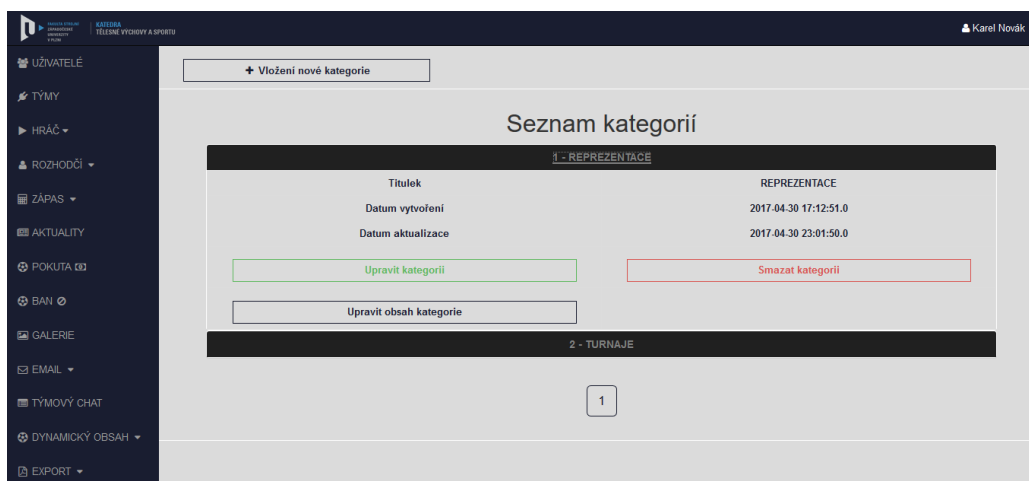
F.2.37 Kategorie dynamického obsahu

Kliknutím na tlačítko „DYNAMICKÝ OBSAH -> Kategorie“ se dostanete na obrazovku seznamu kategorií. Kliknutím na jméno kategorie se rozbalí tabulka s informacemi. Je možné kategorie měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Vložení nové kategorie** - přepnutí do obrazovky pro úpravu nebo vložení nové kategorie, uživatelské vstupy jsou prázdné.
- **Upravit kategorii** - přepnutí do obrazovky pro úpravu nebo vložení nové kategorie, uživatelské vstupy jsou vyplněné podle upravované kategorie.
- **Smazat kategorii** - vyskakovací okno pro potvrzení, následné smazání kategorie.

- **Upravit obsah kategorie** - přechod na obrazovku s úpravou obsahu kategorie.

Obrazovku můžete vidět na obrázku F.54.



Obrázek F.54: Seznam kategorií

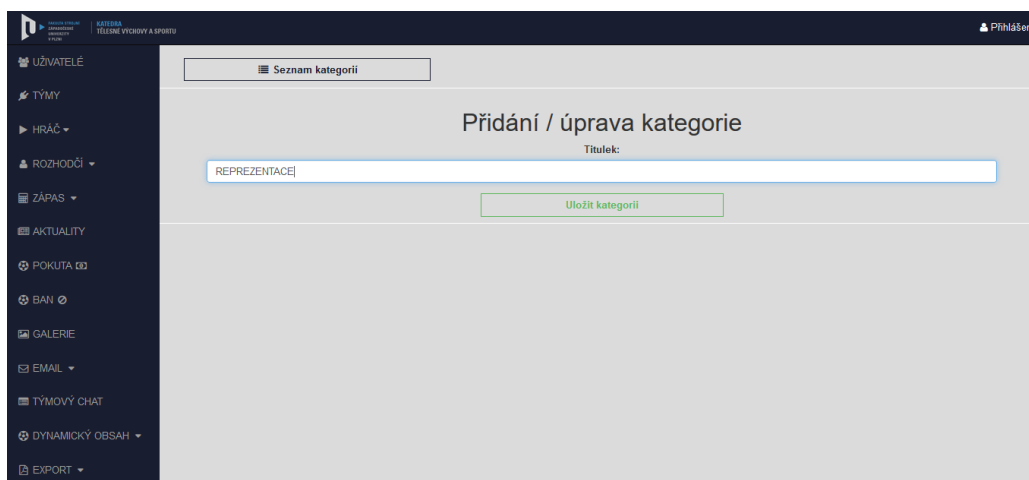
F.2.38 Úprava a vložení kategorie

Na obrazovku se dostanete pouze z obrazovky pro seznam kategorií. Změny se projeví v systému po kliknutí na tlačítko „Ulož kategorii“. Obrazovku můžete vidět na obrázku F.55.

F.2.39 Úprava obsahu kategorie

Na obrazovku se dostanete pouze z obrazovky pro seznam kategorií. Obsah kategorie je spojen s jednou stránkou. Tato část umožňuje vkládání nových textů do kategorie. Pokud kategorie obsahuje více textů budou zobrazeny jednotlivě jako podstránky. Pokud neobsahuje žádné texty, na stránku se zobrazí zdrojový kód. Obsah lze měnit

- **Upravit stránku** - přepnutí do obrazovky pro úpravu nebo vložení nové stránky, uživatelské vstupy jsou vyplněné podle upravované stránky.



Obrázek F.55: Úprava a vložení kategorie

- **Přidat podstránku** - přepnutí do obrazovky pro úpravu nebo vložení nové podstránky, uživatelské vstupy jsou prázdné.
- **Upravit podstránku** - přepnutí do obrazovky pro úpravu nebo vložení nové podstránky, uživatelské vstupy jsou vyplněné podle upravované podstránky.
- **Smazat podstránku** - vyskakovací okno pro potvrzení, následné smazání podstránky.

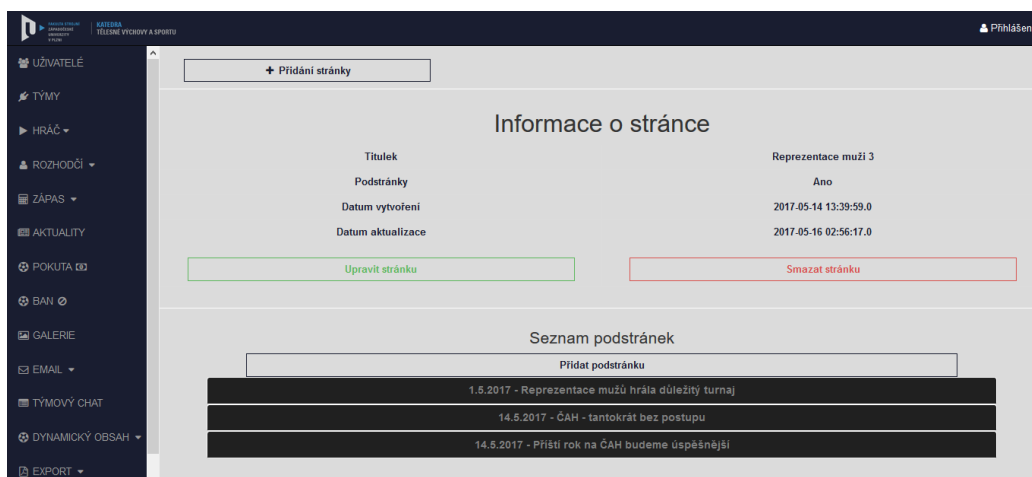
Obrazovku můžete vidět na obrázku F.56.

F.2.40 Úprava a vložení podstránky

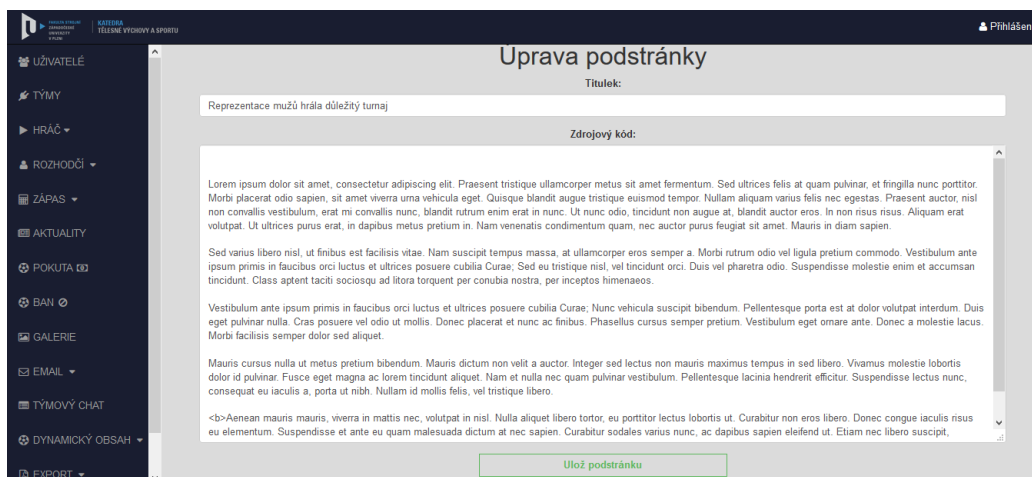
Na obrazovku se dostanete pouze z obrazovky pro úpravu obsahu kategorie nebo podkategorie. Změny se projeví v systému po kliknutí na tlačítko „Ulož podstránku“. Obrazovku můžete vidět na obrázku F.57.

F.2.41 Subkategorie dynamického obsahu

Pro subkategorie existuje stejné prostředí jako pro kategorie.



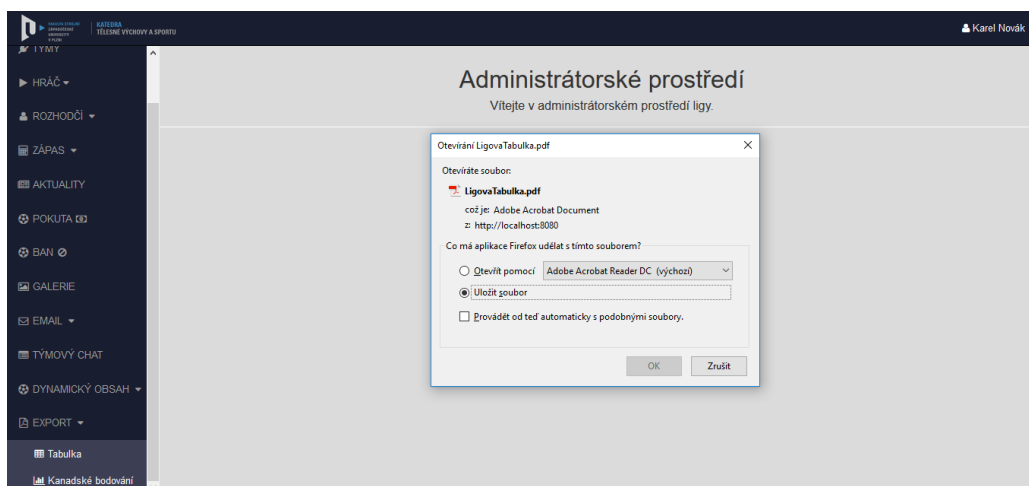
Obrázek F.56: Úprava obsahu kategorie



Obrázek F.57: Úprava a vložení podstránky

F.2.42 Export ligové tabulky

Kliknutím na tlačítko „EXPORT → Ligová tabulka“ lze vygenerovat ligovou tabulku a vytisknout do PDF souboru. Obrazovku můžete vidět na obrázku F.58.



Obrázek F.58: Export ligové tabulky

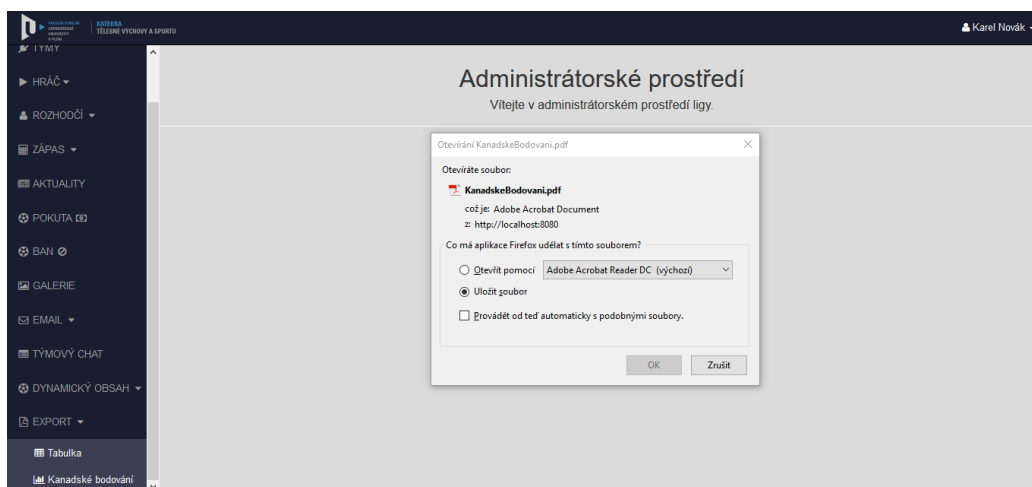
F.2.43 Export kanadského bodování

Kliknutím na tlačítko „EXPORT → Kanadské bodování“ lze vygenerovat kanadské bodování a vytisknout do PDF souboru. Obrazovku můžete vidět na obrázku F.59.

F.2.44 Seznam žádostí o tým

Kliknutím na tlačítko „PŘIDEJ SE K NÁM → Uživatel hledá tým“ lze vidět seznam požadavků od uživatelů, kteří hledají nový tým. Je možné požadavky měnit, přidávat nebo mazat. K tomu existují tlačítka:

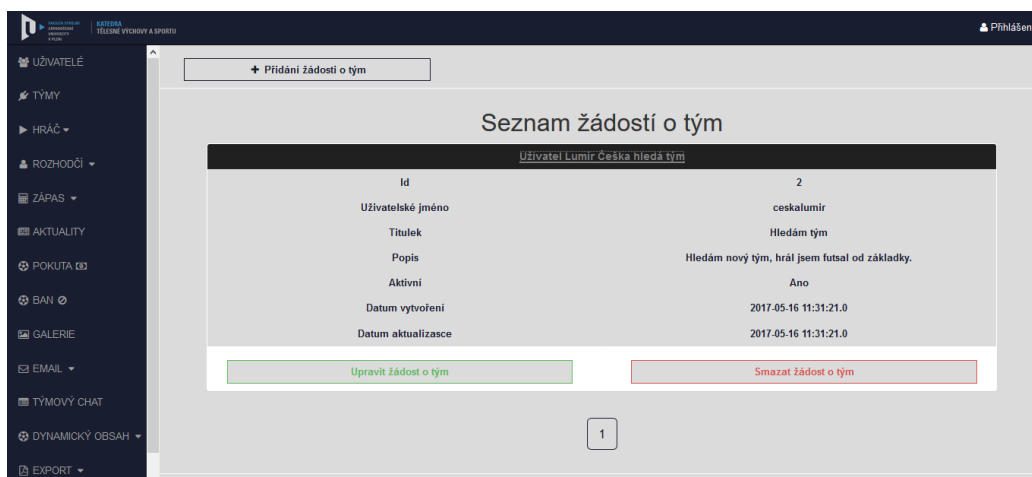
- **Přidání žádosti o tým** - přepnutí do obrazovky pro úpravu nebo vložení nové žádosti o tým, uživatelské vstupy jsou prázdné.



Obrázek F.59: Export kanadského bodování

- **Upravit žádost o tým** - přepnutí do obrazovky pro úpravu nebo vložení nové žádosti o tým, uživatelské vstupy jsou vyplněné podle upravované žádosti o tým.
- **Smazat žádost o tým** - vyskakovací okno pro potvrzení, následné smazání žádosti o tým.

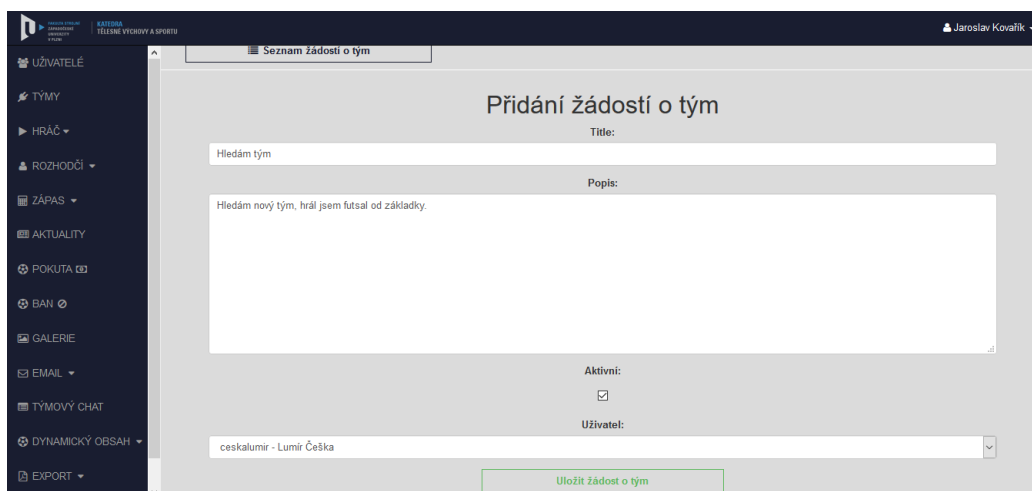
Obrazovku můžete vidět na obrázku F.60.



Obrázek F.60: Seznam požadavků uživatel hledá tým

F.2.45 Úprava a vložení žádosti o tým

Na obrazovku se dostanete pouze z obrazovky pro seznam žádostí o tým. Změny se projeví v systému po kliknutí na tlačítko „Ulož žádost o tým“. Obrazovku můžete vidět na obrázku F.61.

The screenshot shows a web application interface for adding a team request. On the left is a dark sidebar menu with options like 'UŽIVATELE', 'TÝMY', 'HRÁČ', 'ROZHODČÍ', 'ZÁPAS', 'AKTUALITY', 'POKUTA', 'BAN', 'GALERIE', 'EMAIL', 'TÝMOVÝ CHAT', 'DYNAMICKÝ OBSAH', and 'EXPORT'. The main content area is titled 'Seznam žádosti o tým' and 'Přidání žádosti o tým'. It contains a form with the following fields: 'Title:' with the text 'Hledám tým'; 'Popis:' with the text 'Hledám nový tým, hrál jsem fotbal od základky.'; 'Aktivní:' with a checked checkbox; and 'Uživatel:' with a dropdown menu showing 'ceskalumir - Lumir Češka'. At the bottom right of the form is a green button labeled 'Uložit žádost o tým'.

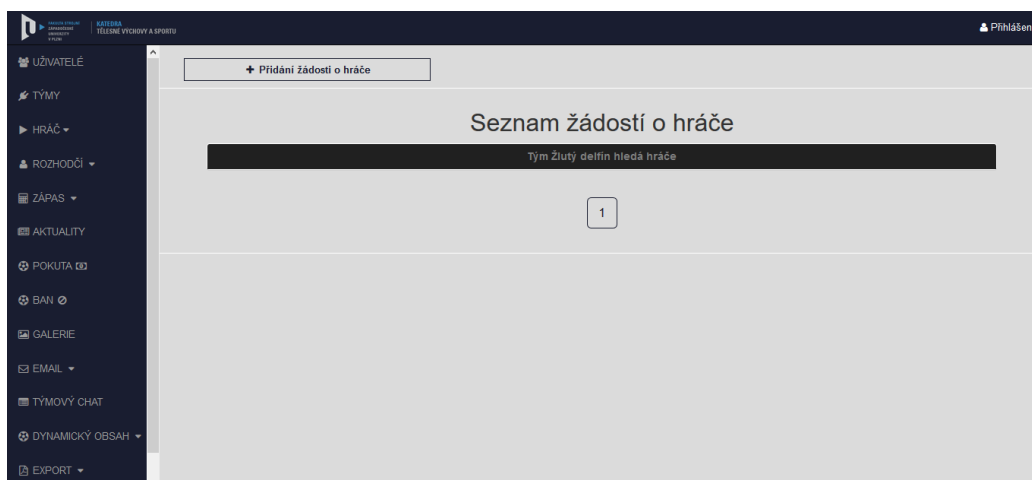
Obrázek F.61: Úprava a vložení žádosti o tým

F.2.46 Seznam žádostí o hráče

Kliknutím na tlačítko „PŘIDEJ SE K NÁM → Tým hledá hráče“ lze vidět požadavky od týmů, které hledají nové hráče. Je možné požadavky měnit, přidávat nebo mazat. K tomu existují tlačítka:

- **Přidání žádosti o hráče** - přepnutí do obrazovky pro úpravu nebo vložení nové žádosti o hráče, uživatelské vstupy jsou prázdné.
- **Upravit žádost o hráče** - přepnutí do obrazovky pro úpravu nebo vložení nové žádosti o hráče, uživatelské vstupy jsou vyplněné podle upravované žádosti o hráče.
- **Smazat žádost o hráče** - vyskakovací okno pro potvrzení, následné smazání žádosti o hráče.

Obrazovku můžete vidět na obrázku F.62.



Obrázek F.62: Seznam požadavků Tým hledá hráče

F.2.47 Úprava a vložení žádosti o hráče

Na obrazovku se dostanete pouze z obrazovky pro seznam žádostí o hráče. Změny se projeví v systému po kliknutí na tlačítko „Ulož žádost o hráče“. Obrazovku můžete vidět na obrázku F.63.



Obrázek F.63: Úprava a vložení žádosti o hráče