

**Realizace webové aplikace  
sportovního klubu s podporou  
CASE nástroje WebRatio a jazyka  
IFML**

**Diplomová práce**

**Vedoucí práce:**

**doc. Ing. Ivana Rábová, Ph.D.**

**Bc. Ondřej Matulík**

**Brno 2016**



## **Poděkování**

Chtěl bych poděkovat vedoucí mé práce doc. Ing. Ivaně Rábové, Ph.D. za odborné konzultace, cenné připomínky a odborné rady, které mi pomohly při zpracování práce.



### **Čestné prohlášení**

Prohlašuji, že jsem tuto práci: **Realizace webové aplikace sportovního klubu s podporou CASE nástroje WebRatio a jazyka IFML**

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

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

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

V Brně dne 4. ledna 2016

---



## **Abstract**

Matulík, O. Implementation of web application for the sport club with the support of IFML notation. Diploma thesis. Brno: Mendel University in Brno, 2015.

This diploma thesis deals with implementation of web application for the sport club with the support of IFML notation. The thesis describes software development methodologies with a main focus on web applications and IFML notation. The practical part of this thesis deals with the implementation of a web application based on analysis of the current state of the problem domain and defined business requirements. Development process of entire application was conducted in accordance with the IFML notation. For the analysis and system modelling were used diagrams and models of UML2 and IFML languages. Models and diagrams were made by CASE tool WebRatio and Enterprise Architect. The application was implemented using ASP.NET MVC technology, Microsoft SQL Server database tool and C# language.

## **Keywords**

IFML, WebRatio, ASP.NET MVC, web application, Microsoft SQL Server, UML

## **Abstrakt**

Matulík, O. Realizace webové aplikace sportovního klubu s podporou CASE nástroje WebRatio a jazyka IFML. Diplomová práce. Brno: Mendelova univerzita v Brně, 2015.

Diplomová práce se zabývá problematikou realizace webové aplikace sportovního klubu pomocí notace IFML. Práce popisuje metodiky vývoje aplikací se zaměřením na webové aplikace a notaci IFML. Praktická část práce se věnuje realizaci aplikace, která je implementována na základě podrobné analýzy současného stavu problémové domény a definovaných požadavků na nově vzniklou aplikaci. Proces vývoje celé aplikace probíhal v souladu s notací IFML. K analýze a modelování systému byly využity diagramy a modely jazyka UML 2.0 a IFML, vytvořené pomocí CASE nástrojů Enterprise Architect a WebRatio. Následně byla aplikace implementována pomocí technologií ASP.NET MVC, Microsoft SQL Server a jazyka C#.

## **Klíčová slova**

IFML, WebRatio, ASP.NET MVC, webová aplikace, Microsoft SQL Server, UML





# Obsah

<b>1</b>	<b>Úvod</b>	<b>13</b>
<b>2</b>	<b>Cíl práce</b>	<b>15</b>
<b>3</b>	<b>Teoretická část</b>	<b>17</b>
3.1	Informační systém .....	17
3.2	Webové aplikace.....	17
3.2.1	Výhody webových aplikací.....	18
3.2.2	Nevýhody webových aplikací.....	19
3.3	Metodiky vývoje informačních systémů.....	19
3.3.1	Rigorózní metodiky .....	20
3.3.2	Agilní metodiky .....	20
3.4	Metody vývoje webových aplikací.....	20
3.5	UML.....	21
3.5.1	Diagramy UML.....	22
3.6	IFML.....	25
3.6.1	Sběr a analýza požadavků.....	28
3.6.2	Základní prvky jazyka IFML.....	29
3.6.3	Doménový model .....	30
3.6.4	Hypertextový model.....	32
3.6.5	Prezentační model .....	34
3.7	ASP.NET MVC 5 .....	35
3.8	MS SQL SERVER 2014.....	36
3.9	WebRatio .....	37
<b>4</b>	<b>Metodika a postup práce</b>	<b>39</b>
<b>5</b>	<b>Vlastní práce</b>	<b>41</b>
5.1	Profil klubu .....	41
5.2	Stav ICT .....	41

---

5.3	Možnosti řešení IS .....	42
5.4	Stav procesů .....	42
5.5	Specifikace požadavků .....	43
5.6	Sběr požadavků .....	44
5.6.1	Funkční požadavky .....	44
5.6.2	Uživatelé systému .....	46
5.6.3	Datové požadavky .....	46
5.6.4	Nefunkční požadavky .....	47
5.7	Analýza požadavků .....	48
5.7.1	Uživatelské role .....	48
5.7.2	Diagramy a specifikace případů užití .....	50
5.8	Doménové modely .....	54
5.9	IFML – Hypertextové modely .....	58
5.9.1	Neregistrovaný uživatel .....	59
5.9.2	Uživatel .....	60
5.9.3	Hráč .....	60
5.9.4	Trenér .....	62
5.9.5	Pokladník .....	63
5.9.6	Sekretář .....	64
5.9.7	Předseda .....	65
5.10	Prezentační návrh .....	65
5.11	Implementace a nasazení .....	67
5.11.1	Vytvoření databáze .....	68
5.11.2	Vývoj a implementace aplikace .....	68
5.12	Testování a nasazení .....	70
5.13	Návrhy na vylepšení .....	71
<b>6</b>	<b>Diskuze</b> .....	<b>73</b>
<b>7</b>	<b>Závěr</b> .....	<b>75</b>
<b>8</b>	<b>Literatura</b> .....	<b>77</b>

Obsah	11
<b>9 Seznam obrázků</b>	<b>79</b>
<b>A Diagramy případů užití</b>	<b>81</b>
<b>B Ukázky webové aplikace</b>	<b>85</b>



# 1 Úvod

Informační systémy dnes již pronikly do celé řady oborů. Dříve se IS implementovaly zejména do podniků za účelem zvýšení konkurenceschopnosti a získání konkurenční výhody. V současné době se již začínají využívat i v neziskových sektorech, kde slouží především ke zvýšení efektivity, automatizaci procesů a úspoře času. Do těchto sektorů jsou zahrnuta různá sdružení, spolky či sportovní kluby.

Řízení sportovního klubu na amatérské úrovni je vykonáváno obvykle v rámci dobrovolnosti, a proto je třeba mít procesy maximálně zoptimalizované, aby nebylo nutné zabývat se věcmi, které lze jednoduše zautomatizovat. To se týká i florbalového klubu TJ Sokol Břeclav, který by chtěl zefektivnit řízení klubu za použití právě informačního systému.

V současnosti se velice často nasazují informační systémy jako webové aplikace, a to díky schopnosti webových serverů provozovat již komplexní podnikové či jiné informační systémy. Aby byl takový systém schopen obsloužit velké množství uživatelů současně, je nutné klást velký důraz na správnou architekturu systému. K tomu účelům slouží mnoho pokročilých metodik pro vývoj webových aplikací.

Dnes již lze zakoupit informační systém pro sportovní klub jako například již hotové „krabicové“ řešení. Vzhledem k odlišnostem jednotlivých klubů, nedokáže tato řešení splnit požadavky týkající se procesů a struktury všech klubů. Další možností je pořízení informačního systému, který je implementován „na míru“. Výhodou těchto IS je především jejich přímé navržení podle organizační struktury daného klubu a pokrytí všech jeho procesů. Taková řešení jsou ovšem z pravidla velmi finančně náročná a menší kluby si nemohou takovou investici dovolit.

Sportovní klub TJ Sokol Břeclav, kterého se tato diplomová práce týká, disponuje pouze webovou prezentací sloužící k informování veřejnosti o činnosti klubu. Některé z administrativních procesů jsou řešeny pomocí tabulkových procesorů, avšak většina stále pomocí „tužky a papíru“. Díky zvyšující se hráčské základně klubu, se stává tento způsob řízení klubu neefektivním a časově náročným.



## 2 Cíl práce

Cílem této diplomové práce je navrhnout a implementovat webový informační systém pro florbalový klub TJ Sokol Břeclav. Jeho úkolem bude zautomatizovat a usnadnit administrativní činnosti klubu, jeho řízení a také prezentování klubu veřejnosti. Tato práce se zaměřuje především na analýzu a implementaci systému pro administraci a řízení klubu. Systém ulehčí práci s organizováním tréninků či zápasů, vytvářením statistik a reportů z docházky. Dále bude sloužit k evidenci příjmů a výdajů klubu a v neposlední řadě také k evidenci cestovních příkazů.

Webová aplikace by měla být vytvořena takovým způsobem, aby ji bylo možné, v případě změn procesů v klubu, upravovat.

Aby bylo možné splnit výše uvedené cíle, je nezbytné vyřešit několik dílčích úkolů. První úkolem je seznámení se s metodami vývoje webových aplikací pomocí notace IFML a s principy CASE nástroje WebRatio. Dále je nutné provést analýzu a namodelování současného stavu procesů v klubu. Tyto procesy jsou spojeny s evidencí členů, pokladnou, docházkou a týmovou komunikací. Realizace aplikace bude provedena pomocí notace IFML s podporou UML a nástroje WebRatio. Na závěr musí dojít k implementaci webové aplikace v jazyce C# a technologie ASP.NET MVC.





## 3 Teoretická část

### 3.1 Informační systém

System je chápán jako soubor více prvků nacházejících se ve vzájemné interakci a tvořící komplexní celek.

Informace jsou taková data, která mají pro uživatele určitý význam a která uspokojují konkrétní objektivní informační potřebu svého příjemce. Transformace dat na informace probíhá až v okamžiku jejich užití. Charakteristikou informací je jejich vypovídající schopnost a je cílem zpracování dat (Rábová, 2010).

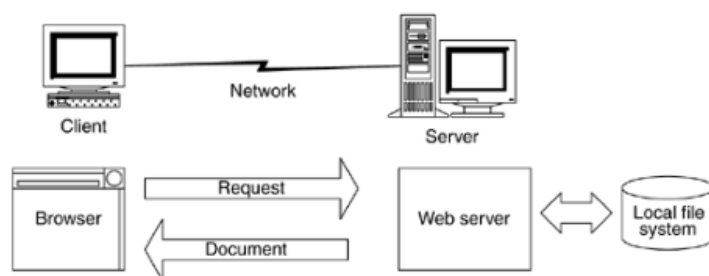
Informační systém je definován jako systém metod, lidí a prostředků zabezpečující informační procesy. Základním úkolem informačních systémů je získat informace, zpracovat je a v potřebný čas, požadovaném rozsahu a formě, je poskytovat na rozhodovací místa (Konečný, 2010).

Jiná definice říká, že informační systém je komplex informací, lidí, použitých technologií a technologických prostředků, který slouží ke sběru, přenosu, uchování a zpracování dat za účelem tvorby a prezentace informací. Správně fungující informační systém je schopen zautomatizovat a zjednodušit procesy, které byly dosud řešeny neefektivně (Rábová, 2010)(Konečný, 2010).

Obsah IS lze vyměřit hlavními charakteristikami, mezi které patří funkce, procesy a funkcionalita IS, která představuje hierarchicky uspořádaný souhrn všech operací s daty. IS by měl v co největší míře podporovat optimalizaci podnikových procesů neboli procesní řízení firmy (Rábová, 2010).

### 3.2 Webové aplikace

Webové aplikace se vyvinuly z webů a webových systémů. Pojem webové aplikace není striktně vymezen. Od klasických webů se webové aplikace odlišují především využitím složitější aplikační logiky a možnosti přístupu k „obchodní“ logice aplikace (Conallen, 2002).



Obr. 1 Princip webového systému  
Zdroj: Conallen, 2002.

Architektura webových aplikací je založena na systému klient-server. Klientovi postačí většinou jen schopnější prohlížeč. Webové aplikace zpracovávají požadavky od uživatelů na webovém serveru a poté jsou za pomoci webového prohlížeče a internetu zobrazena klientovi. Prohlížeč klienta považujeme za „front-end“ rozhraní představující prezentační vrstvu. Zobrazuje obsah uživateli a zároveň přijímá vstupy od uživatele. Tyto vstupy poté zasílá serveru ke zpracování. Webový server (back-end) následně odpovídá uživateli nejčastěji pomocí HTML, XHTML nebo XML stránek s instrukcemi, jak má webový prohlížeč klienta vygenerovat uživatelské rozhraní. Výstupy webových aplikací jsou nezávislé na platformě. Odpadá tak starost o to, jakou platformu a jaký software má uživatel k dispozici. Architektura webových aplikací je často skoro totožná s dynamickými weby. Princip webového systému znázorňuje obrázek výše (Obr. 1) (Conallen, 2002).

V poslední době se stále více začínají využívat nové technologie jako AJAX nebo RIA, které se snaží eliminovat omezení týkající se vzhledu a to za použití stávajících prezentačních jazyků. Díky těmto technologiím je dnes možné vyvíjet webové aplikace, které se vzhledem podobají desktopovým aplikacím.

### 3.2.1 Výhody webových aplikací

Stanley (2010) radí mezi výhody webových aplikací tyto skutečnosti:

- Uživatel pracuje vždy s aktuální verzí aplikace. Aplikační logika je uložena na straně serveru a aktualizace probíhá zde.
- K aplikaci je možné přistupovat odkudkoli, je-li dostupné internetové připojení.
- Uživateli stačí pouze podporovaný webový prohlížeč.
- Nejsou náročné na HW prostředky a technické vybavení klienta.
- Data jsou uložena na jednom místě, z čehož plyne jednodušší provádění záloh a zabezpečení dat.

### 3.2.2 Nevýhody webových aplikací

Mezi nevýhody webových aplikací patří: (Stanley, 2010)

- Vyžadují připojení k internetu. Není-li toto připojení k dispozici, je možné pracovat offline, ale jen v omezené míře.
- V případě nekvalitního připojení k internetu může být zpomalen tok dat a práce s aplikací. Rychlost odezvy ovlivňuje také množství připojených uživatelů, technické vybavení serveru a architektura aplikace.
- Dostupnost webové aplikace závisí na dostupnosti serveru, na kterém aplikace běží. Jakýkoliv výpadek serveru způsobí odepření přístupu uživatelům k aplikaci.
- Data jsou často uchovávána u třetí osoby, z čehož vyplývají možná bezpečnostní rizika.
- Některé webové prohlížeče mohou interpretovat přijaté informace ze serveru odlišně, popřípadě nemusí podporovat použité technologie.
- Vzhledem k tomu, že jsou webové aplikace dostupné prakticky odkudkoliv, je nutno věnovat velkou pozornost zabezpečení.

### 3.3 Metodiky vývoje informačních systémů

Informační systémy se mohou lišit svým rozsahem. Existují jak jednoduché aplikace, které slouží například k evidenci rezervací, tak také rozsáhlé integrované systémy. Tyto integrované systémy jsou ve firmách skládány z dalších subsystémů jako ERP či CRM, které podporují veškeré procesy v podniku. Vývoj takového systému vyžaduje využití pokročilých přístupů pro řízení softwarových projektů (Bružina, 2010).

S vývojem informačních systémů, které se postupem času stávaly stále složitějšími, rostly požadavky na vývoj metodik, které by usnadnily a zdokonalily vývoj IS. Dle Řepy, Synáčka a Hamerníka (2006, str. 6) je metodika tvorby IS definována jako: *„doporučený souhrn etap, přístupů, zásad, postupů, pravidel, dokumentů, řízení, metod, technik a nástrojů pro tvůrce informačních systémů, který pokrývá celý životní cyklus informačních systémů. Určuje kdo, kdy, co a proč má dělat během vývoje a provozu IS. Metodika by se měla vztahovat na všechny prvky IS (pracovníky, organizační procedury, data, SW a HW a další), organizační vlivy IS, ekonomické otázky spojené s vývojem a provozem IS a doporučené dokumenty a případně způsob řízení v jednotlivých fázích životního cyklu IS“*.

Definice samotné metody je dle Brucknera, Voříška a Buchalcekové (2012, str. 302) následující: „Metody určují, co je třeba dělat v určité fázi vývoje či provozu IS/ICT. Metoda je vždy spojena s určitým přístupem, jako je funkční, datový, nebo například objektový přístup“. Z toho vyplývá, že buď každá metoda řeší posloupnost činností v určité fázi procesu vývoje systému, nebo pouze z určitého pohledu na systém (SW, data, funkce).

V minulosti tvořil IS každý projektant většinou na základě vlastních zkušeností. Postupem času začala vznikat jistá doporučení, která se týkala struktury programů

a využití vývojových diagramů. Následně se zvyšoval důraz na strukturované programování (Pascal). Důsledkem toho byl odklon od využívání vývojových diagramů. Časté chyby v SW vedly ke zdokonalování analýzy a stále častějšímu využití grafických modelů. Začala se rozvíjet strukturovaná analýza a metodiky pokrývající celý životní cyklus projektu IS. Druhá polovina 80. let stanovila mohutný rozvoj objektového přístupu k tvorbě SW. Mezi dva základní přístupy ke tvorbě metodologií patří strukturovaný a objektový přístup. V 90. letech se po velkém nárůstu objektových metodik začala projevovat snaha o jejich standardizaci, čímž vznikl jazyk UML. V zájmu co nejrychlejší tvorby spolehlivého a jednoduše rozšiřitelného SW, se začaly vyvíjet metodiky pokrývající všechny fáze životního cyklu tvorby IS (Konečný, 2010).

Nelze obecně říci, že určité metody jsou spojeny s konkrétní metodikou, nebo že každá technika slouží výhradně k podpoře specifické metody. Vztahy mezi metodami, technikami a nástroji nejsou pevně dané a mohou se vzájemně prolínat. Některé metody mohou být více specifické a jiné mají naopak univerzálnější charakter. Metody datového modelování využívají prakticky všechny metodiky. U technik a nástrojů to platí obdobně. Technika může vyžadovat specifický nástroj, bez kterého by neměla smysl nebo využívá obecněji používaný nástroj. Příkladem takového nástroje je diagram procesů nebo ER diagram. Existují také nástroje natolik univerzální, že nejsou vázány k žádným technikám a využívá je mnoho metod. Takto je koncipován celý jazyk UML (Brambilla, Fraternali et al., 2014).

### 3.3.1 Rigorózní metodiky

Mezi základní přístupy k procesu vývoje softwaru patří metodiky rigorózní a agilní. Rigorózní metodiky vycházejí z předpokladů, že procesy a požadavky vytvářeného IS/ICT lze jednoduše specifikovat a popsat. Zpravidla jsou založeny na vodopádovém vývoji. Existují ale rigorózní metodiky založené na iterativním a inkrementálním vývoji. Mezi ně patří metodiky OPEN, RUP (Rational Unified Proces) a EUP (Enterprise Unified Process). Nejčastěji jsou používány ve velkých až středních projektech (Mittner, 2010)(Buchalcevoá, 2005).

### 3.3.2 Agilní metodiky

Agilní metodiky naopak předpokládají, že softwarové procesy není možné předem popsat, což umožní určit specifikaci požadavků jen částečně. Metodika pracuje jen s obecnými pravidly a praktikami. Jsou využívány především u menších týmů a projektů, kde je prioritou co nejrychlejší dodání použitelného produktu (Mittner, 2010).

## 3.4 Metody vývoje webových aplikací

Webové aplikace se od klasických informačních systémů v mnoha ohledech odlišují a mají svá specifika. Pro tvorbu standardní webové prezentace (firemní webové stránky), není potřeba využívat sofistikovaných vývojových metod. Pro vývoj složitějších webových aplikací se v minulosti využívaly klasické metodologie vývoje soft-

waru. Složitost webových aplikací se ale postupem času stále zvyšovala. V současnosti, kdy firmy již nasazují své informační systémy na web, je použití klasických metodologií pro analýzu a návrh takto komplexních webových systémů nevhodné. Postupně docházelo k různým úpravám stávajících metodologií a způsobů modelování, právě pro potřeby vývoje webových aplikací. Jednalo se zejména o OpenUP, UML Web Application Extension a UML-based Web Engineering. Tyto přístupy však opomíjely jeden z nejdůležitějších aspektů webových aplikací, a to zachycení uživatelské interakce a navigace mezi stránkami. V důsledku toho se začaly vyvíjet nové způsoby vývoje založené na hypertextu a hypermédiích (HDM-Object Oriented Hypermedia Design Method, RMM, EORM atd.). Vznikly metodologie určené pro specifikaci, analýzu a návrh webových aplikací, které se využívají především pro datové a navigační modelování. Mezi tyto aplikace patří zejména IFML (WebML) (Zelenka, 2003)(Bružina, 2010).

Bruckner, Voříšek a Buchalceková (2012) předpokládají, že prvotními impulsy pro vytvoření nového jazyka a procesu vývoje byly následující nedostatky při vývoji webových aplikací:

- využívání nevhodných metodologií,
- nedostatečná podpora modelem řízeného vývoje,
- enormní úsilí vkládané do prototypování,
- rostoucí komplexnost moderních internetových stránek,
- rostoucí náklady na vývoj,
- potřeba zvýšení úrovně abstrakce návrhu,
- zaměření sil na analýzy a kreativitu, místo na manuální kódování stránek.

### 3.5 UML

Jazyk UML (Unified Modeling Language) je univerzální jazyk pro vizuální modelování systémů. UML nepopisuje jak při analýze a návrhu postupovat, ale poskytuje nástroje pro tuto činnost. Není vázaný na žádnou konkrétní metodiku a lze jej tedy využít s většinou existujících metodik. Je explicitně navržen zejména pro použití v CASE nástrojích. V roce 1997 byl jazyk přijat sdružením OMG (Object Management Group) jako standard objektově orientovaného jazyka pro vizuální modelování (Arlow, 2007).

Dle Arlowa (2007) se jazyk UML snaží o unifikaci zejména těchto domén:

- **Vývojový cyklus** – UML nabízí vizuální syntaxi pro modelování během celého vývojového cyklu projektu.
- **Aplikační domény** – jazyk umožňuje modelování jakýchkoliv systémů.
- **Implementační jazyky a platformy** – je nezávislý na programovacím jazyce a platformě. Nejčastěji je však využíván v rámci objektově orientovaných jazyků.

- **Vývojové procesy** – podporuje mnoho metodik zabývajících se vývojem systémů. Nejvíce je upřednostňován metodikou UP (Unified Process).

Arlow (2007) také uvádí, že základem UML je modelování softwaru jako kolekce spolupracujících objektů. Aspekty modelů UML:

- **Statická struktura** – popisuje typy objektů, které jsou pro modelovaný systém důležité a jak spolu vzájemně souvisejí.
- **Dynamické chování** – modeluje životní cyklus těchto objektů a způsob jejich interakce pro dosažení požadované funkcionality navrhovaného systému. Oba tyto aspekty je nutné chápat jako jeden celek.

### 3.5.1 Diagramy UML

Diagramy jsou nejznámější a nejpoužívanější částí standardu UML. Jedná se o grafickou reprezentaci množiny elementů, které jsou propojeny vzájemnými cestami (vztahy). Každý diagram zachycuje pohled na nějakou část modelovaného systému, nikoliv na celek. Mezi nejčastěji používané modely pro popis statických částí systému patří (Wazlawick, 2013):

- diagram tříd,
- diagram komponent,
- objektový diagram,
- diagram nasazení.

Pro pohled na dynamické části systému jsou často využívány následující diagramy:

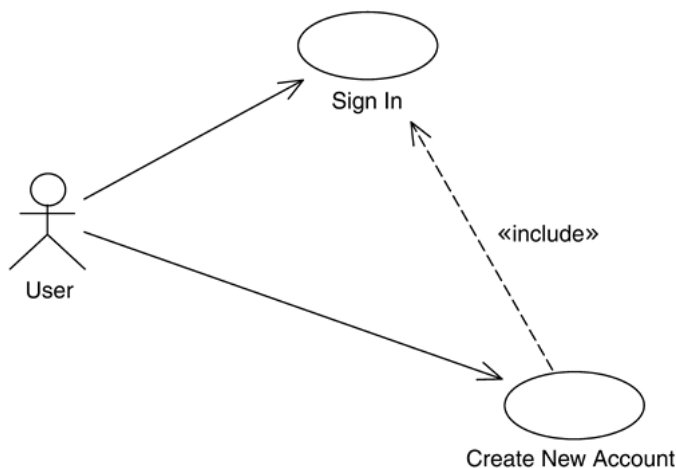
- diagram případů užití,
- sekvenční diagram,
- komunikační diagram,
- stavový diagram,
- diagram aktivit.

#### 3.5.1.1 Diagram případů užití

Diagramy případů užití (Use Case) jsou nejčastěji používanými diagramy v rámci UML. Popisují jeden ze způsobů použití systému a jednoznačně zachycují funkčnost, kterou bude systém disponovat. Vyvinutý systém poskytuje jen takovou funkcionality, jež je popsána diagramy případů užití. Vytváří se v počáteční fázi návrhu systému, kdy získáváme informace od zákazníka ohledně funkčnosti systému.

Diagram se skládá ze čtyř komponent: (Wazlawick, 2013)(Arlow, 2007)

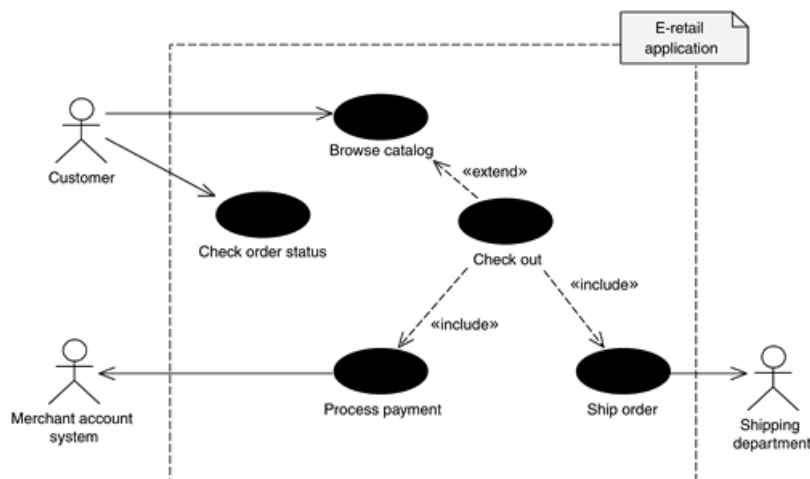
- **Hranice systému** – zobrazují rozsah systému pomocí hranic. Ohraničení představuje případy užití náležící do modelovaného systému, a které naopak nikoliv. Znárodnuje se jako obdélník, ve kterém se nachází případy užití.
- **Aktéři** – určují, jaké role uživatel v rámci své komunikace se systémem reprezentuje. Uživatel může ovšem zastupovat více rolí současně. Aktérem je například fyzická osoba, externí systém nebo čas, který jakkoliv interaguje s modelem systémem.
- **Případy užití (Use Case)** – jedná se o množinu hlavních a alternativních scénářů, vyjadřující funkcionalitu systému nabízenou uživateli. Jinými slovy jde o tok událostí mezi uživatelem a systémem, jež představuje nezbytný proces pro poskytnutí dané funkcionality.
- **Relace (vztah)** – relace definuje vztah (komunikaci) aktérů s případy užití. Relace můžeme rozdělit na následující druhy:
  - **Asociace** – představuje vztah, ve kterém je aktér v interakci s případem užití. Relace je zobrazena pomocí plné čáry.
  - **Include** – jedná se o vztah mezi dvěma případy užití. Nejčastěji se využívá v případě společného chování dvou případů užití. Tato vlastnost se poté vyčlení do samostatného případu užití. Zobrazuje se přerušovanou čarou se šipkou u závislého případu s textem „Include“. Následující obrázek (Obr. 2) znázorňuje vazby v diagramu.



Obr. 2 Use Case Model - vazba "Include"  
Zdroj: Wazlawick, 2013.

- **Extend** – vyjadřuje vztah mezi dvěma případy užití, kde je k danému hlavnímu případu užití přidáno další rozšiřující chování. Hlavní případ užití toto rozšíření obvykle nezaznamená. Znárodnuje se podobně jako vztah „Include“, jen s rozdílným textem – „Extends“. Na obrázku níže lze vidět ukázkou "Top Level" Use Case Modelu (Obr. 3).

- **Dědičnost** – jedná se o zobecnění společného chování případů užití do „rodičovského“ případu užití. Z tohoto případu se následně dědí další případy užití (potomci) a tím je docíleno zjednodušení diagramů.



Obr. 3 "Top Level" Use Case Modelu

### 3.5.1.2 Diagram tříd (class diagram)

Diagram tříd slouží k zobrazení či definici statické struktury celého systému. Je jedním ze základních postupů při analýze navrhovaného systému. Hlavními stavebními bloky jsou třídy a jejich vzájemné vztahy. Třída popisuje množinu objektů se stejnými vlastnostmi. Jsou znázorněny obdélníkem a každá třída obsahuje několik parametrů: název, atributy (mohou mít nastavenou viditelnost a datový typ) a metody. Metody třídy mohou mít specifikovanou viditelnost (stejně jako atributy), návratové typy a vstupní parametry. Atributy vyjadřují informace o objektu. Chování třídy je popsáno pomocí metod, které mohou pracovat s atributy nebo s jinými objekty (Wazlawick, 2013).

Vztahy mezi třídami dělíme na:

- **Asociace** – popisuje relaci mezi dvěma třídami. U této relace definujeme pojem násobnost omezující počet objektů dané třídy, které mohou být v relaci s objektem jiné třídy. Asociace může existovat i v rámci jedné třídy. Objekty dané třídy odkazují na objekty stejné třídy.
- **Generalizace** – umožňuje využívat dědění v rámci vztahu rodič - potomek.
- **Agregace** – je speciální typ asociace vypovídající o tom, že třída je částí jiné třídy (vztah celek – část). Třída, která je součástí jiné třídy, může existovat nezávisle na celku a může být částí více tříd zároveň.
- **Kompozice** – je silnější forma agregace, kde část celku nemůže existovat samostatně. Instance této třídy vzniká a zaniká s instancí celku.



### 3.6 IFML

Modelování uživatelského rozhraní aplikací je nejkomplikovanější fází softwarového vývoje. Do příchodu IFML (WebML) byla tato část v softwarovém inženýrství často přehlížena. Právě proto vznikla tato metodika, jež dokáže vývojářům usnadnit velké množství práce.

Jazyk IFML (Interaction Flow Modeling Language) je jazyk určený pro modelování toku interakcí systému s uživatelem. V roce 2013 byl přijat jako standart OMG (Object Management Group) spadající do kategorie „Modeling and Metadata Specifications“ podobně jako jazyk UML. IFML je podporován společností WebRatio, zaměřující se na vývoj podnikových webových aplikací. Společnost vyvíjí nástroj také zvaný WebRatio, původně založený na jazyce WebML, ze kterého právě IFML vychází a přímo jej nahrazuje. IFML zobecňuje koncept jazyka WebML a odstraňuje omezení týkající se použití pouze na webovou doménu. Jedná se o rozšíření jazyka UML pro modelování datově náročných uživatelských rozhraní (Data Intensive Applications). Nejčastěji jsou to aplikace typické sběrem a uchováváním velkého množství strukturovaných záznamů. Modelovací jazyk WebML se vyvíjel více než 10 let a v době psaní této práce je IFML v počáteční verzi 1.0. Tato diplomová práce je založena právě na verzi 1.0. Není tedy vyloučeno, že se od novějších verzí může částečně lišit (Webratio, 2013)(Wazlawick, 2013).

Jazyk je určen k modelování toků a interakcí systému s uživatelem, k vyjádření obsahu a chování front-end aplikací náročných na přenos dat. Nejčastěji se jedná o podnikové informační systémy nebo webové aplikace. Mezi hlavní oblasti zájmu patří: (IFML, 2015)

- tradiční webové aplikace založené na HTML a HTTP,
- „Rich Internet Applications“ podporované standardem HTML 5,
- mobilní aplikace,
- klient-server aplikace,
- desktopové aplikace,
- více okenní a kontextové aplikace.

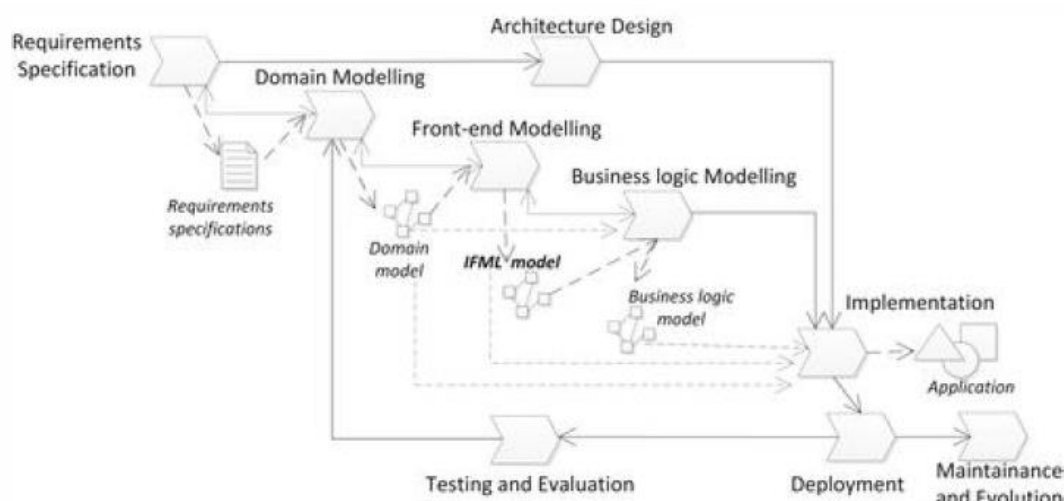
Ve výše uvedených aplikacích se IFML zaměřuje na tyto hlavní cíle: (IFML, 2015)

- obsahovou část,
- navigační prvky,
- události,
- vazby na obchodní logiku,
- vazby na perzistenci.

IFML se nezabývá modelováním konečného vzhledu aplikace (grafika, stylování) a uživatelského rozhraní, proto také není vhodný pro vývoj systémů, které jsou založeny na grafických prvcích, jako jsou videohry či vysoce interaktivní aplikace (Brambilla, Fraternali et al., 2014).

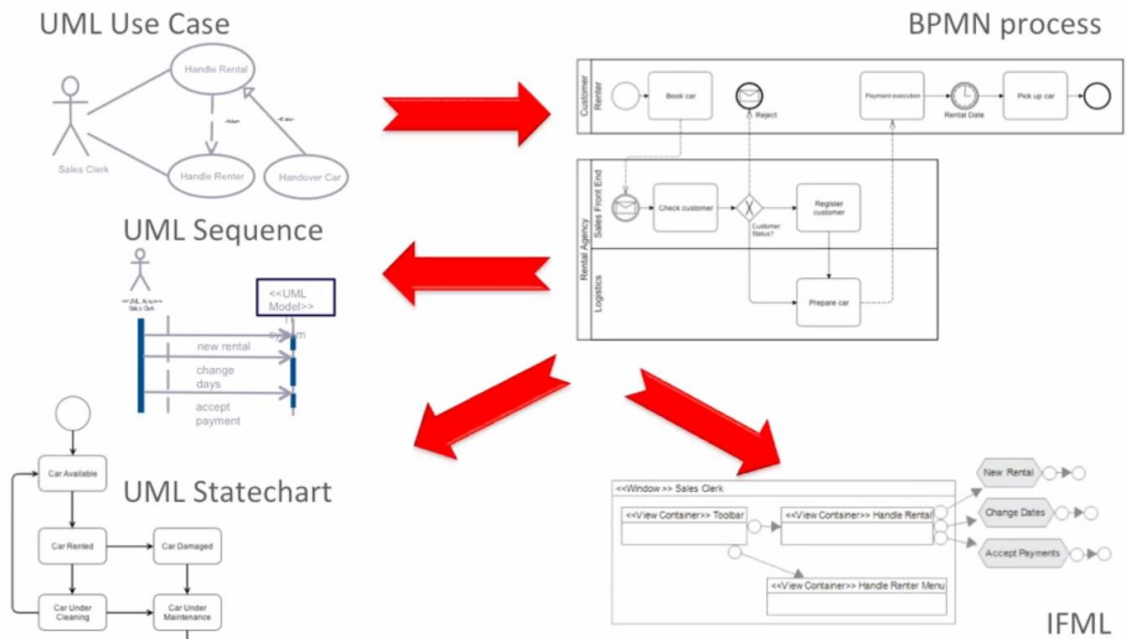
Mezi hlavní koncepty patří abstrakce od použitých technologií. Což znamená, že se vývojář nemusí zabývat konkrétní syntaxí programovacího jazyka, jenž bude použit pro konečnou implementaci. Základem je namodelování domény, kterou je možno následně převést do libovolného programovacího jazyka (IFML, 2015).

Metodika je založena na třech hlavních modelech – datový, hypertextový a prezentační. Z pohledu dnes populární architektury MVC (Model-View-Controller) se IFML zaměřuje převážně na „View“ model, což je prezentační část aplikace. Popisuje závislosti „View“ modelu na zbývajících modelech. IFML umožňuje designerům a vývojářům namodelovat, jak uživatelské interakce ovlivňují chování systému. S výjimkou prezentačního modelu jsou všechny modely ukládány ve značkovacím jazyce XMI (Brambilla et al., 2013).



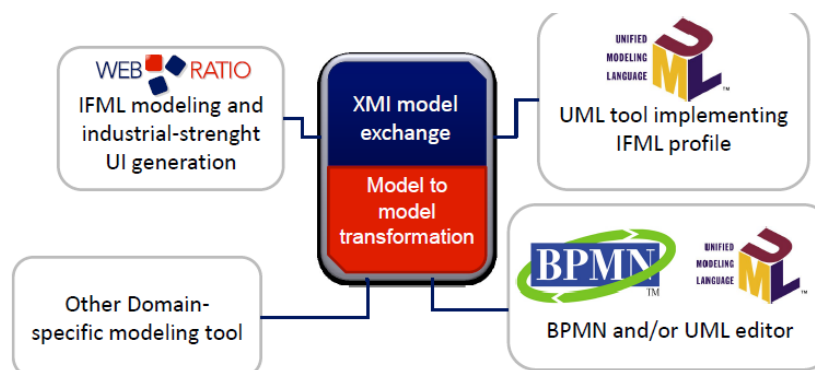
Obr. 4 Cyklus vývoje dle notace IFML  
Zdroj: Brambilla, Fraternali, 2014.

IFML se řídí procesem vývoje dle standardu MDDE (Modelově řízený proces vývoje). Staví na již zaběhlých konceptuálních modelech a kopíruje běžný iterativní a inkrementální cyklus vývoje softwaru metodiky UP (Unified Process). Proces vývoje dle notace IFML je znázorněn na obrázku výše (Obr. 4). Autoři se snažily provázat IFML modely s již existujícími metodikami pro analýzu a specifikaci požadavků (BPMN, UML) tak, aby byl IFML schopen podporovat kompletní vývojový proces aplikací (Obr. 5). Právě využitím IFML modelů spolu s BPMN diagramy, diagramy aktivit (UML), sekvenčními diagramy a diagramy tříd je umožněno využití všech výhod pro modelování rozsáhlých „front-end“ aplikací jako jsou webové či mobilní aplikace (Brambilla, 2013).



Obr. 5 Propojení IFML modelů s již existujícími metodikami  
Zdroj: Object Management Group, 2014.

Vývojář ovšem není omezen jen na vybrané metody, ale může využít jakékoliv metody určené pro analýzu a popis procesů. Na obrázku níže (Obr. 6) je znázorněno schéma propojení a integrace jednotlivých nástrojů využívající formát XMI. Jazyk IFML se zaměřuje především na popis uživatelských rozhraní. Specifikaci a popis požadavků nebo jiných omezení nechává již na volbě samotného vývojáře (Brambilla, Fraternali et al., 2014).



Obr. 6 Integrace nástrojů skrze model XMI 0

Na základě výše uvedených skutečností lze říci, že IFML popisuje kompletní proces vývoje webové aplikace včetně životního cyklu projektu. Proces vývoje aplikace na základě notace IFML se skládá z těchto hlavních částí:

- Specifikace požadavků.
- IFML modely a diagramy:
  - doménový model,
  - kompoziční model,
  - model business logiky (navigační model).
- Realizace aplikace.
- Testování.
- Nasazení a údržba.

### 3.6.1 Sběr a analýza požadavků

Prvním krokem každé analýzy je popsání problémové domény a definování požadavků na nový systém. Tyto požadavky je nutné následně analyzovat a schválit, jak zadavatelem, tak také zpracovatelem projektu. Jakákoliv změna zadání v průběhu projektu může způsobit nejen navýšení nákladů, ale také nedodržení termínů. Jeli-kož mohou tyto skutečnosti v konečném důsledku ovlivnit spokojenost zákazníka, je nutné věnovat této fázi projektu dostatek prostředků.

IFML využívá pro tuto fázi projektu již existujících a prověřených metodik. Ne-definuje, podle jaké metodiky by měla specifikace požadavků probíhat, ale dává jen určitá doporučení. Dle Arlowa a Neustadta (2007) se jednotlivé fáze dělí na sběr a popis požadavků a analýzu požadavků.

#### 3.6.1.1 Sběr a popis požadavků

Jedná se o soupis základní funkční specifikace, které se vytváří na základě komuni-kace se zákazníkem a studiem firemních dokumentů.

#### 3.6.1.2 Analýza požadavků

Po sběru požadavků je nutné provést jejich vyhodnocení a analýzu. Dále jsou shro-mážděné požadavky sjednoceny a převedeny do formální podoby. Výstupem jsou funkční a nefunkční požadavky.

### Funkční požadavky a případy užití

Popisují, jaké funkce by měly mít jednotlivé skupiny uživatelů k dispozici. Nejčastěji se vyjadřují pomocí UML diagramů a případů užití. Funkční požadavky dělíme na:

- Identifikace uživatelů – rozděluje jednotlivé uživatele do skupin se stejnými charakteristikami v rámci firemních procesů.
- Požadavky na personalizaci – popisuje, které skupiny uživatelů budou přístu-povat ke stejnému obsahu. Vytváří se tzv. Site Views (základní pohledy) pro jednotlivé uživatele a skupiny.

- Požadavky na data – z těchto požadavků se následně vychází při tvorbě datového modelu. Jde o popis základních objektů systému pomocí datového slovníku.

### Nefunkční požadavky

Mezi nefunkční požadavky patří například požadavky týkající se dostupnosti služeb, bezpečnosti a možné rozšiřitelnosti.


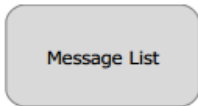

### 3.6.2 Základní prvky jazyka IFML

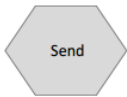

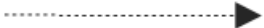
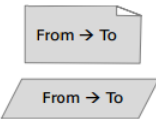
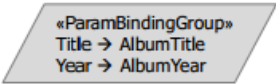
Mezi základní prvky IFML patří zejména diagramy. Diagram obsahuje jeden nebo více základních kontejnerů, které představují webovou stránku, formulář a okno aplikace. Tento kontejner je poté vnitřně hierarchicky rozdělen do dalších, do sebe vnořených kontejnerů webových stránek nebo do dalších částí formuláře.

Každý jednotlivý kontejner může obsahovat prvky neboli komponenty, zobrazující obsah nebo sloužící k získávání vstupních dat. Komponenty mohou mít jak vstupní, tak výstupní parametry. Výstup jedné komponenty může být vstupem pro jinou komponentu. Tento vztah je zobrazován pomocí vazby.

Jakýkoliv prvek (komponenta či kontejner) je schopen reagovat na různé události jak od uživatele, tak také od externí aplikace nebo od jiného prvku. Reakci na událost označovanou jako interakční tok znázorňujeme pomocí propojení zúčastněných prvků zobrazených v tabulce níže (Tab. 1)(IFML, 2015).

Tab. 1 Základní pojmy z oblasti IFML

Prvek	Význam	IFML notace	Příklad užití
<b>ViewContainer</b>	Prvek rozhraní, které obsahuje prvky zobrazující obsah a podporuje interakci s dalšími ViewContainers		Okno webové stránky, okno ve windows, podokno, panel
<b>ViewComponent</b>	Zobrazuje obsah nebo vstupní formulář		HTML seznam, JavaScript galerie, vstupní - výstupní formulář, zaškrťovací tlačítko
<b>Event</b>	Událost ovlivňující stav aplikace		

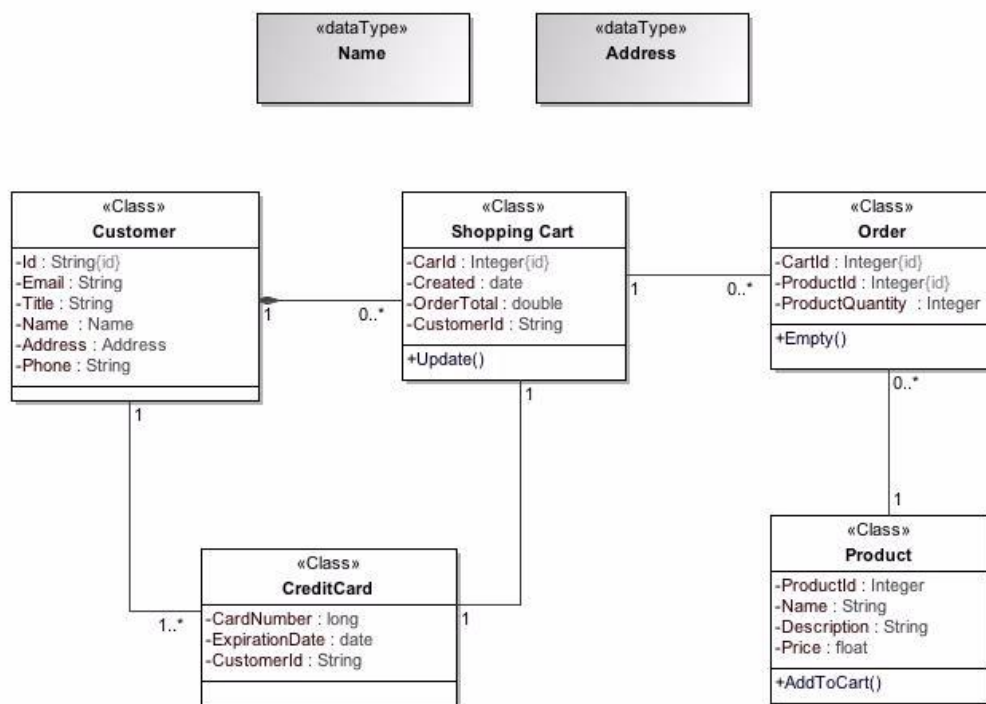
<b>Action</b>	Část obchodní logiky spuštěné určitou událostí		Aktualizace databáze, odeslání emailu, kontrola pravopisu
<b>NavigationFlow</b>	Závislost vstupů a výstupů. Zdroj odkazu má nějaký výstup, který je spojen s výstupem cíle		Přijetí nebo odeslání parametru v HTTP požadavku
<b>Data Flow</b>	Data, která procházejí mezi ViewComponents, nebo akce jako důsledek předchozí interakce uživatele		
<b>Parameter Binding</b>	Určuje, že vstupní parametr zdroje je spojen s výstupním parametrem cíle		
<b>Parameter Binding Group</b>	Množina prvků ParameterBinding spojená do InteractionFlow (navigace nebo datový tok)		

Zdroj: IFML, 2015.

### 3.6.3 Doménový model

Metodika vývoje pomocí IFML definuje modely, které popisují řešenou doménu. Mezi zmíněné modely se řadí doménový, odvozený, kompoziční, navigační a prezentační model.

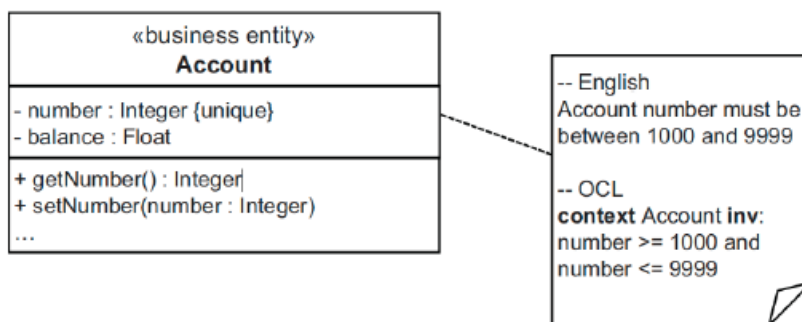
Prvním z modelů je doménový neboli strukturální, jenž reprezentuje datovou strukturu a organizaci dat, se kterou aplikace pracuje. Pro vyjádření doménového modelu je možné v rámci metodiky využít libovolný prostředek pro datové modelování (ERD, objektový model ODMG či model tříd UML). Model je složen ze základních datových konstrukčních prvků, mezi které patří: třídy, atributy a vztahy včetně jejich datových typů (text, integer, date). Ukázkou doménového modelu představuje obrázek níže (Obr. 7) (Wazlawick, 2013)(IFML, 2015).



Obr. 7 Strukturální model jazyka IFML  
Zdroj: Object Management Group, 2014.

### 3.6.3.1 Odvozený model

Model vzniká rozšířením doménového modelu odvozením konstrukčních datových prvků. Jedná se o modelování pohledů na data z hlediska určité webové stránky. Odvozený model je možné si představit jako tzv. pohled „View“ do původního strukturálního modelu. Odvozený model můžeme chápat jako množinu odvozených atributů a vztahů (asociací), které jsou většinou definovány v doménovém modelu a mohou být napsány v libovolném jazyce OCL. Na následujícím obrázku (Obr. 8) lze vidět grafické znázornění odvozeného modelu jazyka IFML. Model definuje také tzv. výčty (population) tříd nebo vztahů a to na základě atributových omezení (Wazlawick, 2013)(IFML, 2015).



Obr. 8 Odvozený model jazyka IFML  
Zdroj: Object Management Group, 2014.

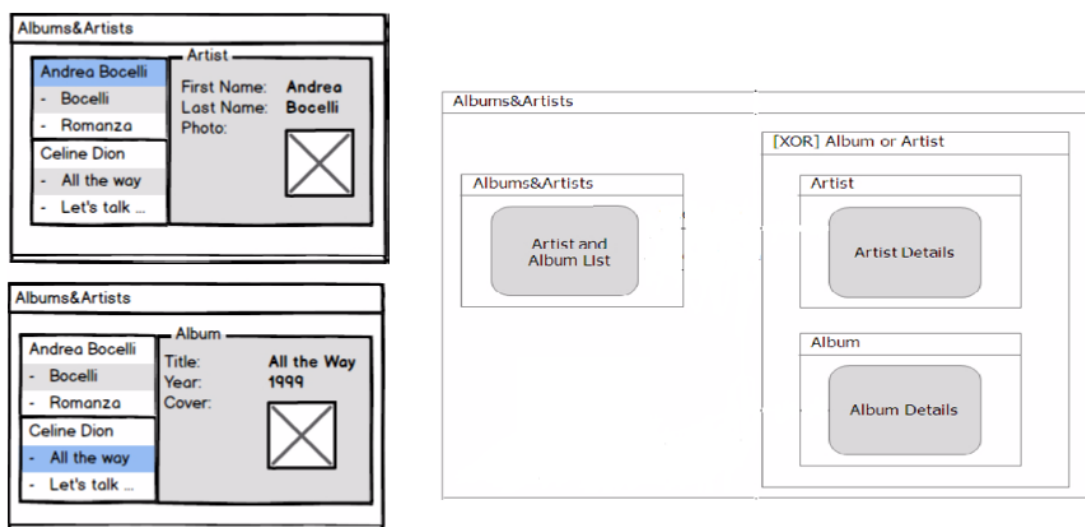
### 3.6.4 Hypertextový model

Hypertextový model představuje nejdůležitější model v jazyce IFML. Pomocí tohoto modelu je možné namodelovat chování a strukturu celé aplikace na konceptuální úrovni. Model definuje tzv. pohledy na systém (tzv. „Site Views“). Můžeme jej rozdělit na dva modely – model kompoziční a model navigační. Zmíněné modely se obvykle zakreslují do jednoho diagramu. Pro jeden strukturální model je možné vytvořit i několik hypertextových modelů (Wazlawick, 2013)(IFML, 2015).

#### 3.6.4.1 Model kompozice

Model kompozice určuje strukturu stránky a popisuje, ze kterých logických prvků je stránka složena. Mezi zmíněné prvky patří index, data, multi-data, pages, areas, filters atd.





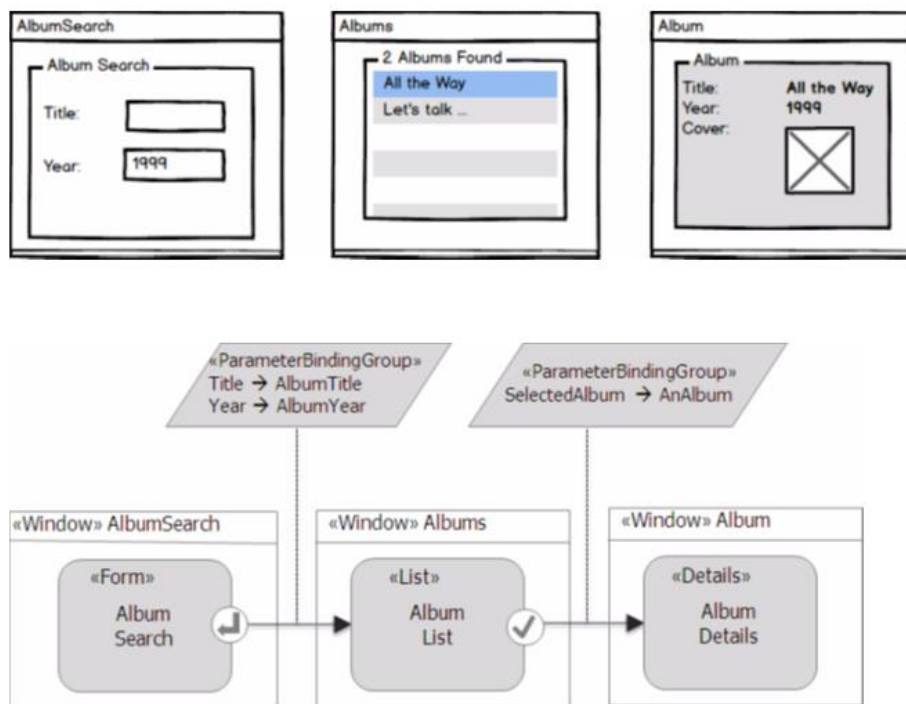
Obr. 9 Kompoziční model aplikace - IFML

Zdroj: IFML, 2015.

Na levé straně obrázku (Obr. 9) je znázorněn příklad uživatelského rozhraní s odpovídajícím IFML modelem na pravé straně. Hlavní kontejner obsahuje tři pohledy zobrazující přehled autorů s jejich alby, detaily autora a detaily konkrétního alba. Z posledních dvou pohledů se zobrazuje vždy jen jeden a to v závislosti na volbě uživatele.

### 3.6.4.2 Navigační model

Model popisuje navigaci mezi jednotlivými stránkami, které jsou propojeny pomocí kontextových a bezkontextových odkazů. Zobrazuje pohledy pomocí interakčních toků a modeluje také způsob pohybu uživatele v systému.

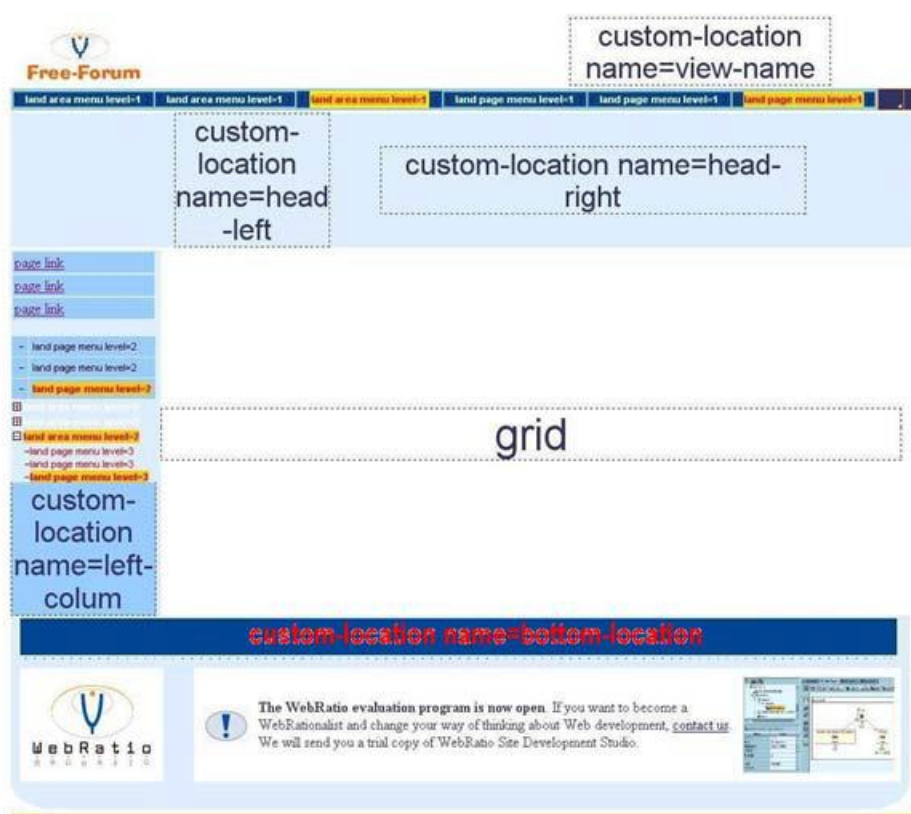


Obr. 10 Navigační model aplikace - IFML  
Zdroj: IFML, 2015.

Na obrázku (Obr. 10) lze vidět příklad uživatelského rozhraní převedeného do podoby IFML modelu. Po zadání uživatelských dat do vstupního formuláře a následném potvrzení, se zobrazí požadovaný „view“ kontejner se seznamem objektů, který odpovídá hledané frázi.

### 3.6.5 Prezentační model

Prezentační model je transformací předešlých modelů a umožňuje vizuální pohled na modelovanou doménu. Určuje vzhled a rozmístění jednotlivých prvků uživatelského rozhraní. Model je vyjádřen pomocí syntaxe XML a je tedy nezávislý na reálném výstupním zařízení. Pro generování kódu daného výstupního zařízení je použita technologie XSLT. K tvorbě layoutu jednotlivých stránek můžeme využít i oficiální nástroj WebRatio, který dokáže navíc kontrolovat návaznost na předešlé modely. Ukázka prezentačního modelu aplikace je na obrázku níže (Obr. 11) (Wazlawick, 2013)(IFML, 2015).



Obr. 11 Prezentační model aplikace - IFML  
Zdroj: Object Management Group, 2014.

### 3.7 ASP.NET MVC 5

Technologie ASP.NET společnosti Microsoft je nástupcem starší technologie Active Server Pages (ASP). Jedná se o framework umožňující vytváření dynamických webových stránek. Technologie je založena na platformě .NET, jehož součástí je i CLR. To umožňuje psát kód v jakémkoliv podporovaném jazyce (Visual Basic, C#, C++ a další). CLR poskytuje další služby, jako je správa paměti či zabezpečení. Framework nabízí rozsáhlou knihovnu předpřipravených nástrojů, díky kterým je možné vyvíjet aplikace s menším množstvím psaného kódu, tudíž i rychleji a efektivněji. Je založen na objektově orientovaných principech. Zaměřuje se především na aplikační logiku a nikoli na infrastrukturu aplikace. Mezi základní knihovny patří LINQ (sada komponent ADO.NET), umožňující jednoduše přistupovat k relačním databázím a jiným zdrojům dat. AJAX slouží pro rychlou tvorbu webových stránek pomocí nástrojů podporujících funkci „drag and drop“ a obsahuje další knihovny pro vývoj webových aplikací a síťovou komunikaci (Evjen, 2009) (Millett, 2010).

Aplikace tvořené pomocí ASP.NET jsou kompilovány do DLL souborů předem, a proto jsou mnohem rychlejší než klasické jazyky, které je nutno při každém přístupu interpretovat. To zajišťuje mnohem vyšší rychlost zpracování oproti staršímu

ASP. Odpovědi na požadavky odeslané skrz protokol HTTP jsou zpracovávány prostřednictvím IIS (Internet Information Server) (Galloway, 2012).

Programování v ASP.NET můžeme rozdělit podle použitého frameworku na MVC a WebForms. Framework MVC se stal v poslední době velmi populární a to především pro tvorbu webových aplikací. Soustředí se na vývoj aplikací dle struktury MVC, která rozděluje aplikaci na - Model, View a Controller. Tyto části jsou na sobě částečně nezávislé, což zajišťuje oddělení kódu (aplikační logiky) od vzhledu aplikace, což usnadňuje znovu použitelnost kódu a dílčích částí. ASP.NET MVC umožňuje kodérovi naprostou volnost ve volbě nástrojů, knihoven, návrhových vzorů a JavaScriptových frameworků. To vše zvyšuje produktivitu a kvalitu vytvořených aplikací. Mezi standartní syntaxe patří Razor View Engine, který převádí prvky do HTML kódu na základě šablon (Helpers). Programátor může využít v kódu i standartní HTML konstrukce (Galloway, 2012)(Pecinovský 2007).

### 3.8 MS SQL SERVER 2014

Databázové řešení od společnosti Microsoft je jedním z nejpoužívanějších a nejoblíbenějších nástrojů pro práci s daty a databázemi. Může se pochlubit také nejlepším zabezpečením mezi srovnatelnými produkty. SQL server je integrovaným řešením, které je navrženo tak, aby podnikům všech velikostí usnadnil správu a analýzu dat. Umožňuje vytvářet aplikace, využívající velké objemy dat pomocí výkonných technologií pro uchování dat v paměti napříč úlohami OLTP, datového skladu, Business Intelligence a analytickými úlohami. Samozřejmostí je správa zálohování, virtualizace a dynamických virtuálních disků. Pomocí běžné sady nástrojů je možné spravovat jak místní, tak také cloudové databáze jako například Microsoft Azure (Mistry, 2014)(www.businessit.cz, 2014).

Oproti starším verzím byly přidány tyto novinky a vylepšení: (Mistry, 2014)(www.businessit.cz, 2014)

- **In-memory OLTP** – kompletní databáze je uložena v operační paměti, což dokáže rapidně zvýšit transakční výkon databáze až desetkrát.
- **In-memory buffer pool** – místo, kde SQL Server ukládá nejčastěji používané objekty (paměťová cache), lze jej rozšířit o některé rychlé úložiště (SSD disk).
- **Vlastní role zabezpečení** – umožňuje pracovat s rolemi na úrovni serveru a vytvářet vlastní role s různými možnostmi oprávnění. Příkladem může být nastavení rolí pro databázového administrátora tak, aby mu bylo znemožněno čtení citlivých dat z databáze.
- **Integrace a zálohování do Microsoft Azure** – systém je schopen zálohovat databáze nejen na disky, ale i do cloudového úložiště Microsoft Azure Storage. Výhodou tohoto řešení jsou nízké náklady a jednoduchá konfigurace. Samozřejmostí je možnost šifrování dat pomocí certifikátů.

- **In-memory ColumnStore a BI** – je funkce, zajišťující jednodušší aktualizaci, vyšší rychlost a kompresi dat pro datové sklady. K dispozici je rovněž i in-memory BI s nástrojem Power Pivot pro získání rychlého přehledu o situaci i z rozsáhlých dat.

### 3.9 WebRatio

CASE nástroj WebRatio byl vytvořen v roce 2001 na univerzitě Politecnico Di Milano. V aktuální verzi plně podporuje standard jazyka IFML, který byl detailně popsán v kapitole 3.6. Jedná se o nástroj pro vizuální modelování aplikace a podnikových procesů, který zefektivňuje vývoj webových aplikací. WebRatio umožňuje automatické generování webových a mobilních aplikací z IFML a BPMN modelů. Lze generovat kód, jak pro serverovou, tak i klientskou část. Platforma WebRatio je škálovatelná a je v souladu s nejpřísnější firemní bezpečnostní politikou. Vytvořené aplikace jsou kompatibilní s HTML5, CSS3 a Java. Webové aplikace postavené na platformě WebRatio splňují standardy Java/JSP 2.0+ a mohou být nasazeny na libovolném aplikačním serveru. Aplikace mohou být rovnou nasazovány do „Cloudu“. K dispozici je také verzování projektů (CVS nebo Sub-verze). Aplikace vytvořené pomocí WebRatia je možné integrovat s dalšími systémy a službami (SAP, Tibco, SaaS, Dropbox apod.) (Brambilla, Butti 2014) (WebRatio, 2014).



## 4 Metodika a postup práce

Diplomová práce se zabývá vývojem webových aplikací a informačních systémů se zaměřením na modelovací jazyk IFML.

Nezbytným krokem pro získání komplexního přehledu v oblasti vývoje webových aplikací a IS, bylo seznámení se s principy vývoje a možnými technologiemi jejich implementace. Tato oblast je součástí teoretické části práce.

Po seznámení se s oblastí principů a metodik vývoje webových aplikací a informačních systémů, které bylo nezbytné pro další práci, bylo nutné z těchto metodik vybrat nejvhodnější řešení pro vlastní práci. Z nabízených metodik byl vybrán modelovací jazyk IFML, kterému se tato práce bude věnovat podrobněji.

Teoretická část představuje výchozí bod pro získání nutných poznatků. Hlavní náplní teoretické části je prozkoumání veškerých možností využití tohoto jazyka pro účely vývoje vlastní aplikace, zejména principy a postupy související s touto metodikou, včetně popisu jednotlivých modelů. Mezi následující kroky patřilo důkladné nastudování nezbytných nástrojů, které byly potřeba pro analýzu a implementaci webové aplikace v praktické části této práce. Jedná se především o CASE nástroj WebRatio, jež je primárním nástrojem pro modelování a vývoj aplikací pomocí notace IFML.

V praktické části práce budou získané poznatky využity pro vývoj a implementaci aplikace řešené domény. Implementovaný systém bude využívat moderních webových technologií, využívající objektový přístup a inkrementální způsob vývoje. Inkrementální vývoj byl vybrán, jelikož je velmi pravděpodobné, že v průběhu vývoje i provozu dojde ke změnám a rozšiřování funkcionality aplikace.

Vývoj aplikace bude obsahovat veškeré náležitosti vývoje softwaru a to od analýzy, návrhu, fyzické implementace až po konečné testování a nasazení aplikace.

Mezi hlavní kroky v počáteční fázi analýzy patří popis modelovaného systému, zmapování současné úrovně ICT, a také zejména analýza a popis procesů v klubu. Následně dojde ke sběru požadavků.

V následující části bude provedena analýza těchto požadavků, včetně jejich namodelování pomocí Use Case diagramů (UML), které vzniknou pomocí nástroje Enterprise Architect.

Nejdůležitější fází vývoje bude návrh aplikace, kde se využije všech možností CASE nástroje WebRatio pro modelování obsahu, struktury a chování aplikace dle notace IFML.

Na základě předešlých fází vývoje se provede implementace aplikace za využití technologií a vývojových nástrojů společnosti Microsoft.

V závěru práce dojde k zhodnocení přínosů systému v praxi. Následně budou také shrnuty výhody vývoje webové aplikace pomocí nástroje WebRatio dle notace IFML.





## 5 Vlastní práce

Tato kapitola se zabývá analýzou, návrhem a implementací vytvářené webové aplikace. Pro návrh aplikace bude využit vývojový proces aplikace dle doporučení modelovacího jazyka IFML, který byl popsán v teoretické části této práce.

### 5.1 Profil klubu

Sportovní florbalový klub COMMIX TJ Sokol Břeclav, pro který bude v rámci této práce vytvořen webový informační systém, sídlí v Břeclavi a nyní eviduje téměř 50 aktivních hráčů ve věku od 15 do 40 let. Díky většímu povědomí o tomto sportu se počet hráčů v klubu COMMIX TJ Sokol Břeclav každým rokem zvyšuje. V současné době vytvořil florbalový klub tři družstva. Jedno z nich se účastní oficiální soutěže pod záštitou České florbalové unie (ČFBU). Zbývající dvě družstva hrají Jihomoravský přebor TJ Sokol.

O plynulý chod klubu se stará předseda, sekretář, pokladník a hlavní trenér. Předseda a sekretář se nyní starají o veškerou administrativu. Jejich pravomocí nejsou zcela přesně definované, čímž se částečně prolínají. Vzhledem ke zvyšující se hráčské základně, dochází současně ke zvýšení náročnosti na řízení klubu. Klub se proto rozhodl rozšířit řídicí pozice o další členy.

Finanční prostředky získává klub převážně z dotací svazu TJ Sokol. Mezi další podstatnou složku rozpočtu lze zařadit dary od sponzorů. Zbývající výdaje jsou hrazeny z členských příspěvků a udělených pokut. Mezi největší klubový výdaj patří pronájem hal a následně také poplatky za start na soutěžích. V souvislosti s návštěvami soutěží musí klub vynaložit i náklady na dopravu.

Již z povahy sportovního klubu vyplývají činnosti, kterými se tento klub musí zabývat. Jedná se tedy především o organizování tréninků, plánování zápasů či turnajů, správa financí a vedení pokladního deníku, výběr členských příspěvků a pokut, kontrola jejich úhrad, zajišťování pronájmu hal, získávání dotací a sponzorů, komunikace se členy klubu a s veřejností.

### 5.2 Stav ICT

V současné době spravuje klub pouze webové stránky, jejichž úkolem je prezentovat klub veřejnosti. Stránky jsou umístěné na hostingu Endora.cz a web funguje na redakčním systému WordPress (WP) s pluginem pro sportovní klub s názvem SportClub. Klub na svých stránkách prezentuje dosažené výsledky a aktuální umístění v zúčastněných soutěžích. Dále jej využívá ke správě výsledků soutěže sokolské ligy. V rámci statistik se zde zveřejňují výsledky jednotlivých zápasů, umístění týmů a také statistiky hráčů. Výše zmíněné údaje jsou zpracovány pomocí pluginu ve WP. Toto řešení není bohužel nejvhodnější. Plugin SportClub neumožňuje úpravy ve větší míře, což má za následek uchovávání velkého množství dat, které klub ke své

činnosti nepotřebuje. Další nevýhodou je ta skutečnost, že plugin nepodporuje zpracování údajů, které jsou pro klub důležité.

Nyní klub nevyužívá žádný komplexní informační systém, který by usnadnil chod a řízení klubu, především po administrativní stránce.

### 5.3 Možnosti řešení IS

Při pořizování nového IS pro sportovní klub existuje v dnešní době více možných variant řešení. Mezi ně patří:

- zakoupení komerčního „krabicového“ řešení,
- implementace IS na míru od externí firmy,
- využití OPEN SOURCE řešení,
- vlastní implementace.

Na trhu s IS, potažmo CRM, je dnes mnoho produktů, které jsou určeny přímo pro sportovní kluby. Jsou k dispozici jak komerční, tak nekomerční řešení. Kvůli omezenému finančnímu rozpočtu klubu není možné využít implementace IS na míru od externí firmy. Je tedy nutné využít již hotových řešení, která budou umožňovat alespoň částečnou parametrizaci. Po analýze všech dostupných řešení nebyl nalezen produkt, který by splňoval veškeré požadavky klubu na informační systém. Zkoumaná řešení ve většině případů nesplňovala požadavky týkající se komplexnosti a pokrytí všech oblastí klubu. Nejvhodnějším řešením je vlastní implementace, kterou se bude následující část práce zabývat.

### 5.4 Stav procesů

Již z obecné povahy sportovního klubu vyplývají některé procesy, které jsou klíčové pro fungování klubu. Mezi tyto procesy se řadí přihlašování na tréninky či zápasy, placení příspěvků, žádosti o proplacení nákladů (za dopravu, nákup sportovních pomůcek), evidence pokladny, evidence údajů o členech a sdílení dokumentů.

V případě přihlašování na události, tréninky či zápasy probíhá nyní komunikace především skrze emailovou komunikaci. Bohužel tento způsob řešení velice znesnadňuje tvoření jakýchkoli reportů o docházce za určité časové období. Za další nevýhodu lze považovat to, že ostatní spoluhráči nejsou informováni o tom, kdo přijde na danou událost. Často se vyskytuje problém s emailem ohledně událostí, kdy nebyl email hráči doručen. V důsledku toho dochází k situaci, kdy se hráč o události nedozvěděl, tudíž se na ni nemohl přihlásit a ani se na ni dostavit.

Členské příspěvky jsou povinni všichni hráči uhradit vždy jednou za sezónu. Tento příspěvek je tvořen částkou za registraci hráčů v soutěži a poplatky za organizování tréninků, pronájem hal, odměny rozhodčím a další náklady. Výše příspěvků se liší v závislosti na počtu hráčů v týmu, počtu tréninků a dalších okolnostech. Informace o tom, kolik má každý hráč zaplatit a do jaké doby má být poplatek uhrazen,

je hráči zasílána na email, popřípadě sdělena osobně. Většina hráčů ovšem tyto poplatky hradí se zpožděním, jelikož na tyto poplatky zapomínají nebo doručený email omylem odstraní. Tato skutečnost je pro klub poměrně zásadní, jelikož zpožděná úhrada poplatků ovlivňuje jeho finanční zajištění.

Obdobným způsobem jsou vyřizovány také žádosti o proplacení nákladů, které jsou řešeny přes email, telefonicky či pomocí tištěných dokumentů, které jsou předávány osobně. Se zvyšujícím se počtem členů je způsob tištěných dokumentů zcela nevhodný, jelikož dochází ke zvýšení nároků na administrativní činnost. Jedinou činností, kterou nyní zefektivňuje současný redakční systém, je vedení záznamů o utkání, ze kterých lze generovat statistiky.

Evidence všech záznamů o členech, zaplacených příspěvcích a dalších dokumentech, je prováděna ručně sekretářem za pomoci tabulkových editorů, což je nepraktické a časově velice náročné. S tím také souvisí ukládání tištěných dokumentů, které musejí být vždy uloženy u pověřených osob. Tento nedistribuovaný systém zamezuje jakékoli sdílení dokumentů mezi více uživateli či možnost k nim přistupovat kdykoliv a odkudkoliv.

## 5.5 Specifikace požadavků

Pro správnou činnost klubu je potřeba optimalizovat a zefektivnit fungování základních procesů, které souvisejí zejména s administrativou klubu a tokem finančních prostředků. Jelikož členové pracují v klubu ve svém volném čase na základě dobrovolnosti, je jakákoli úspora času a materiálních prostředků velmi žádána.

Základem je vytvoření webového informačního systému, který zefektivní práci týmu s organizováním výše uvedených činností, umožní komunikaci a sdílení informací mezi členy klubu. Koncoví uživatelé by tak měli získat požadované informace rychle a to bez nutnosti přímé komunikace s vedením klubu. Systém umožní práci především s neveřejnými a interními informacemi. Do systému bude mít povolen přístup jak vedení klubu, tak i jeho hráči. Je ovšem nutno specifikovat uživatelské role s rozdílnými administrátorskými právy. Každá role poskytne uživateli přístup jen do předem určených částí systému a umožní provádět jen určité změny.

Dále bude vytvořena webová prezentace klubu, která poslouží primárně veřejnosti a poskytne návštěvníkům informace o dění klubu. Ve zveřejněné části nalezneme výsledky utkání, pořádané akce, statistiky soutěží i profily členů klubu. Velice důležité je, aby systém umožňoval dobrou rozšiřitelnost, kterékoli jeho části v případě potřeby, jelikož může dojít v budoucnosti ke změně procesů provádění některých činností nebo se změní požadavky na funkcionalitu.

## 5.6 Sběr požadavků

### 5.6.1 Funkční požadavky

Jelikož bude systém představovat komplexní řešení, je nutné jej rozdělit na několik menších subsystémů, jejichž popis je nutno rozvinout. Funkční požadavky na jednotlivé subsystémy budou specifikovány pro každý jednotlivý modul. Sběr požadavků byl proveden na základě analýzy současného stavu a na základě konzultací s vedením klubu a budoucími uživateli systému.

#### **Webová prezentace klubu**

Veřejná část aplikace bude určena především pro návštěvníky, kteří se zajímají o dění v klubu, o plánovaných akcích a také o umístění klubu v zúčastněných soutěžích.

#### **Přihlašování**

Uživatelé se budou přihlašovat do interní (neveřejné) části IS za pomoci svého uživatelského jména a hesla, skrze webovou prezentaci klubu. Přihlašovací údaje jim poskytne sekretář nebo předseda, který jej zaregistroval do systému.

#### **Práce s uživateli**

Uživatelem interní části aplikace může být jak člen klubu bez dalších práv, tak také hráč nebo osoba ve vedení klubu. Oprávněná osoba, sekretář nebo předseda klubu, bude moci vytvářet, mazat či upravovat uživatele. Přihlašovací údaje budou uživatelům zaslány na email. Heslo i uživatelské jméno si budou moci uživatelé kdykoliv měnit. Také si mohou přidat profilovou fotografii či vyplnit osobní údaje. Při mazání uživatele ze systému zůstává uživatel stále v databázi. Uživateli se změní pouze jeho stav na neaktivní, a to pro uchování statistik či případný návrat do klubu. Jelikož se uživatelé veřejné části do aplikace nepřihlašují, nelze s těmito uživateli nijak pracovat.

#### **Statistiky událostí**

Aplikace bude zpracovávat a uchovávat statistiky ze sokolské ligy na základě zadaných výsledků utkání. Jedná se o statistiky jednotlivých hráčů, kde bude sledován počet vstřelených branek, odehraných utkání a počet udělených trestných minut. Co se týče statistiky týmů, zde vznikne statistika ohledně počtu bodů vstřelených branek, odehraných utkání, počet trestných minut, počet získaných bodů, umístění v tabulce, počet vyhraných a prohraných utkání, vstřelené a obdržené góly. Pro zobrazení statistik ze soutěže ČFBU budou využity statistiky z oficiálních stránek soutěže ([www.cfbu.cz](http://www.cfbu.cz)), které si může každý uživatel zobrazit přímo v systému.

#### **Tvorba událostí (trénink, zápasy)**

Aplikaci umožní plánovat pravidelné či mimořádné tréninky. Pravidelné tréninky zveřejní systém automaticky a zajistí současně zaslání pozvánek na tyto události. Správce bude mít možnost nastavit, kolik dní předem se má trénink automaticky

zveřejnit a po jakou dobu bude možné se na trénink přihlašovat. K události lze nastavit místo konání a zda budou proplaceny náklady za dopravu.

### **Přihlašování na události a pozvánky**

Hráči se budou moci přihlašovat na nadcházející události buď pomocí pozvánek zasláných na email, které jim budou zaslány při každé nově vytvořené události, nebo skrze interní informační systém. Pozvánka na událost, kterou hráči obdrží prostřednictvím emailu, bude obsahovat mimo jiné následující tři hypertextové odkazy do systému související s přihlášením na událost:

- přijdu na událost,
- přijdu pozdě na událost,
- nepřijdu na událost.

Výhodou tohoto řešení je zejména rychlost přihlášení se na událost bez nutnosti zdoluhavého přihlašování do systému. Hráči bude také umožněno nastavení dlouhodobé absence, kdy ho systém automaticky omluví ze všech událostí v daném období. Systému tak umožní kontrolu aktuální účasti na vybrané události. Administrátor nebo sekretář může sám přihlašovat či odhlašovat hráče z budoucích událostí.

### **Členské příspěvky a pokuty**

Systém umožní hráčům jednoduše zjistit, zda mají uhrazené veškeré členské příspěvky a pokuty, popřípadě jakou částku musejí ještě uhradit a do jakého termínu. Systém bude pravidelně informovat hráče o nezaplacených příspěvcích či pokutách, pokud se již blíží doba jejich splatnosti. Jestliže hráč částku neuhradí do doby splatnosti, systém bude emailem informovat pokladníka i sekretáře. Bude také k dispozici přehled jednotlivých pokut, včetně jejich detailů jako je datum udělení, událost a kdo pokutu udělil.

### **Požadavky na náhradu nákladů**

Každý uživatel či hráč má nárok na podání žádosti o proplacení nákladů, jako jsou náklady za dopravu či jiné náklady spojené s činností klubu. Systém umožní žadatelům sledovat stav jejich žádosti. O schválení žádosti rozhoduje sekretář nebo předseda. Dojde-li ke schválení, pokladník je povinen proplatit náklady ve stanoveném termínu.

### **Evidence plateb**

Do evidence plateb bude mít přístup celé vedení klub, ovšem práva ke změnám údajů či vytvoření záznamu dostane pouze pokladník a předseda klubu. Systém umožní evidenci všech příjmů a výdajů klubu, u kterých bude umožněno vyhledávat a zobrazovat jen požadované záznamy.

### **Udělování pokut**

Trenér, sekretář a předseda získají oprávnění udělit hráči pokutu za neúčast na tréninku, pozdní příchod či nepřihlášení se na ohlášenou událost. Výši pokuty za konkrétní prohřešek bude možné jednotně nastavit, aby ji nebylo nutné při každém udělení zadávat. Částku bude ovšem možné v případě potřeby individuálně změnit.

### **Diskuzní fórum**

Členové klubu budou mít k dispozici diskuzní fórum, které mohou využít k týmové komunikaci. Fórum se rozdělí na témata týkající se tréninků, zápasů a připomínek k vedení klubu.

#### **5.6.2 Uživatelé systému**

Mezi nejpočetnější skupinu uživatelů systému lze zařadit hráčskou a členskou základu klubu. Tito uživatelé budou nejčastěji pracovat v neveřejné části informačního systému, do kterého se musí přihlásit pod svými přihlašovacími údaji.

Za další početnou uživatelskou skupinu lze považovat fanoušky týmu a veřejnost. Zmínění uživatelé budou mít přístup pouze do veřejné části, tedy do webové prezentace klubu, jelikož nemají nutnost se přihlašovat do systému. Nejmenší skupinou, ale o to významnější, budou členové vedení klubu.

Pro zjednodušení přidělování práv jednotlivým uživatelům budou v systému definovány uživatelské role, které budou vytvořeny na základě uživatelských skupin popsaných výše. Role budou mít předem nastavena oprávnění a přístupy k jednotlivým částem systému. Uživatelům systému budou přiřazeny role na základě jejich postavení v klubu.

#### **5.6.3 Datové požadavky**

Po definování všech požadovaných funkcí informačního systému, včetně popisu všech uživatelů, kteří budou aplikaci používat, je možné přistoupit ke specifikaci datových požadavků. Jedná se o popis dat, které by měly být systémem zpracovávány a uchovávány. Tyto požadavky budou rozděleny do skupin na základě uživatelských rolí, typu statistických údajů a informačního obsahu aplikace. Detailní soupis dat jednotlivých skupin je obsažen v modelu datové struktury. Datová struktura aplikace bude rozdělena do 2 nezávislých částí. První pro webovou prezentaci a druhá pro interní, neveřejnou část systému.

### **Osobní údaje**

Osobní údaje představují informace o uživatelích aplikace. Pro klub jsou podstatné především kontaktní údaje a přiřazené role.

### **Údaje o jednotlivých týmech klubu**

Dále budou shromažďována data týkající se jednotlivých týmů v rámci klubu. Data představují soupisky jednotlivých týmů a jejich odpovědné osoby. Dále je také nutno sbírat informace o účasti v soutěžích za dané období.

### **Údaje o pořádaných akcích**

Největší skupinou, z hlediska datové náročnosti, bude uchovávání informací, které se týkají pořádaných akcí jako zápasy, tréninky a jiné akce. Do této skupiny dat také patří údaje o termínu a místě pořádání akce, včetně skupiny uživatelů s oprávněním k přihlášení na tuto událost. Získané údaje budou využity v dalších částech aplikace.

### **Pokladna a platby**

Aplikace dále umožní práci s pokladnou, pomocí které budou evidovány veškeré platby a pokuty jednotlivých hráčů či týmů. Jak již bylo výše uvedeno, pokladny se týkají údaje o příjmech a výdajích z pokladny s vazbou na konkrétní osobu nebo tým. Důležitým identifikátorem bude účel platby, datum pohybu na pokladně a částka. Pro klub je nutné udržovat informaci o dlužných částkách hráčů vůči klubu včetně doby splatnosti.

### **Statistické údaje**

Jedním z hlavních požadavků klubu je uchovávání statistických údajů ze všech částí aplikace, což představuje statistiky odehraných zápasů, (pořadí) týmů v zúčastněných soutěžích a evidenci návštěvnosti jednotlivých tréninků a dalších akcí.

### **Cestovní příkazy a náhrady nákladů**

Poslední skupinou spravovaných dat budou interní požadavky uživatelů na úhradu nákladů. Mohou to být jednak žádosti o úhradu cestovních nákladů, ale také žádosti o úhradu nákladů za nakoupené pomůcky. Co se týče úhrady cestovních nákladů, je nutné evidovat identifikační znaky vozidla žadatele. Za identifikační znaky lze považovat typ automobilu včetně SPZ, ujetou vzdálenost a požadovanou částku.

#### **5.6.4 Nefunkční požadavky**

- Aplikace bude dostupná přes webové rozhraní formou HTML stránek.
- Aplikace musí být přehledná a jednoduše obsluhovatelná pro všechny zainteresované skupiny uživatelů.
- Rozdělení na veřejnou část (prezentace klubu s výsledky a články) a neveřejnou část (interní informační systém).
- Aplikace musí být dostupná kdykoli a odkudkoli 24 hodin denně.
- Systém musí spolupracovat s běžně používanými prohlížeči jako IE 8, Firefox, Opera a další.

- Vzhledem k uchování osobních dat je nutné zajistit důsledné zabezpečení proti úniku dat a jejich případné zneužití.
- Provádět pravidelné systémové zálohy a to minimálně 1x týdně.
- Databáze musí být spolu s informačním systémem lehce upravitelná, dojde-li v budoucnosti k rozšiřování struktury aplikace.
- Realizace webové aplikace proběhne pomocí programovacího jazyka ASP.NET. Jako databázový stroj bude sloužit Microsoft SQL Server 2014.

## 5.7 Analýza požadavků

Ze získaných požadavků z předchozích kapitol bude nyní provedena analýza požadavků a dojde tak k formalizování těchto požadavků. V průběhu projektu může dojít k rozšíření či změně těchto požadavků na základě konzultace s vedením klubu (zadavatelem), což je ale součástí iterativního vývoje pomocí IFML a nejedná se tak o negativní jev. Analýza požadavků bude provedena pouze pro první iteraci. Pro každý modul systému budou specifikovány případy užití (Use Case) jazyka UML, včetně diagramů. Specifikace případů užití umožňuje definovat interakce uživatelů s aplikací včetně jejich kompletního průběhu.

### 5.7.1 Uživatelské role

Jak již bylo uvedeno, v rámci organizační hierarchie klubu můžeme definovat čtyři základní uživatelské role, které řídí chod klubu. K těmto řídicím skupinám je nutné specifikovat také ostatní role, které budou v systému definovány. Mezi zmíněné role patří například hráč, návštěvník či člen klubu. Každá role bude disponovat určitými pravomocemi a privilegii.

#### Návštěvník

Návštěvník je uživatel, který vstoupil na webovou aplikaci bez přihlášení. Je mu umožněno pouze prohlížení klubové statistiky, čtení článků a prohlížení soupisky týmů. Po přihlášení do aplikace se z něj stává již registrovaný uživatel a má přístup do interní (neveřejné) části IS, kde jsou mu poskytnuta další práva.

#### Uživatel

Uživatele lze definovat jako návštěvníka aplikace, který se přihlásil do interního informačního systému a není zatím členem žádného týmu. Po přihlášení do systému má právo spravovat svůj profil, přispívat do týmové diskuze nebo prohlížet pořádané události.

#### Hráč

Za hráče je považován uživatel, který je členem jednoho nebo více týmů. Hráč má povinnost platit příspěvky, přihlašovat se na události, které jsou mu určeny a platit případné pokuty. Dále má právo podávat žádosti na úhradu nákladů za cestovné.



**Předseda**

Osoba řídící klub a vystupující jako zástupce klubu. Zastupuje klub při jednání se sponzory, komunikuje s vedením soutěže ČFBU a účastní se pravidelných schůzí TJ Sokol. Předseda klubu disponuje nejvyššími právy a má přístup do celého systému bez omezení.

**Sekretář**

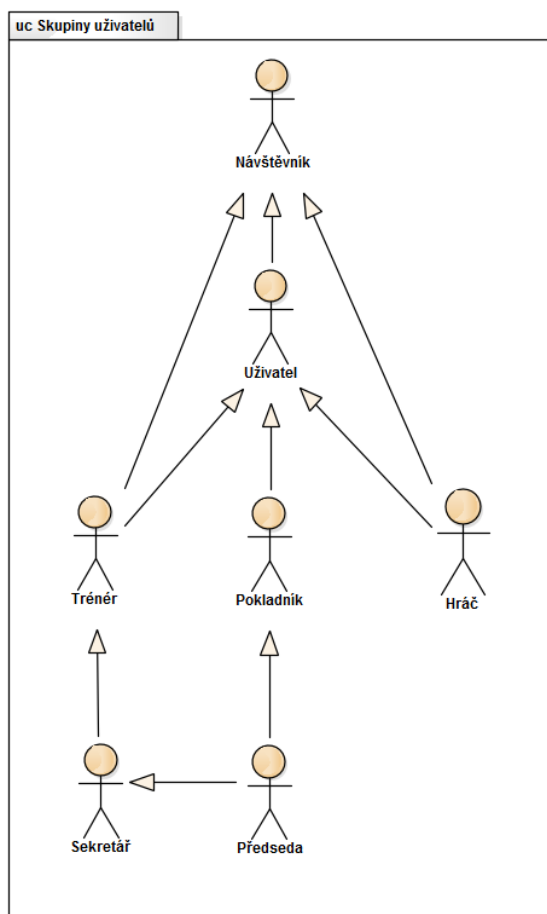
Osoba zajišťující většinu administrativní činnosti v klubu. Vyřizuje například dokumenty, které jsou potřebné pro účast v soutěžích. Dále vytváří profily hráčů a udržuje aktuální informace o členech klubu. Zajišťuje také komunikaci se zástupci ostatních týmů a má právo vytvářet nové události (tréninky a zápasy), na které se hráči přihlašují.

**Pokladník**

Pokladník se stará o veškeré činnosti související s tokem finančních prostředků. Eviduje platby členských příspěvků, proplácí ostatním členům klubu náhrady nákladů.

**Trenér**

Mezi činnosti trenéra patří vytváření soupisek mužstev, udělování pokut jednotlivým hráčům a také metodické vedení přidělených týmů. Dále má trenér právo vyhodnocovat docházku za již proběhlé události.



Obr. 12 Uživatelské skupiny

Na obrázku (Obr. 12) jsou zobrazeny uživatelské skupiny webové aplikace.

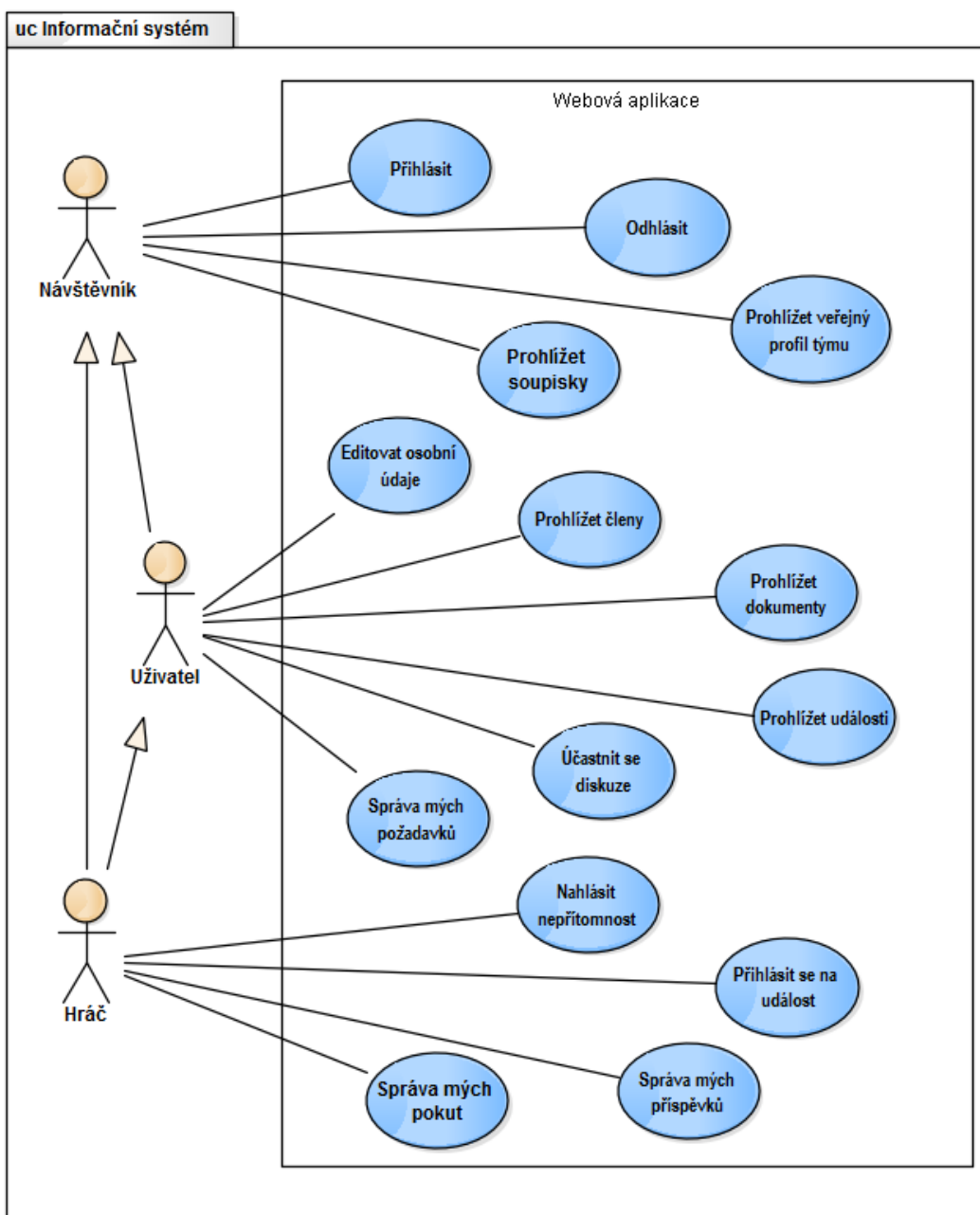
### 5.7.2 Diagramy a specifikace případů užití

Diagramy případů užití zobrazují využití systému a jeho funkcí jednotlivými uživateli. Jedná se o definování interakce systému s jeho uživateli. Pro důkladnou analýzu a poznání systému je nutné znát všechny případy užití pro každou roli uživatelů. Pro potřeby této práce budou využity prvky jazyka UML, tedy diagramy případů užití. Diagramy případů užití je nutné vzhledem ke struktuře aplikace rozdělit do několika podsystémů. V této kapitole budou uvedeny jen základní diagramy. Ostatní diagramy je možné najít v příloze A této práce.

S diagramy úzce souvisí také specifikace těchto případů užití. Specifikace popisuje, jakým způsobem se bude daný případ užití vykonávat a k čemu slouží. Skládá se z názvu, stručného popisu, definice vstupních podmínek, konečného stavu a kompletního scénáře případu.

Jedním z hlavních a prvotních případů užití při interakci uživatele se systémem, je vstup neregistrovaného návštěvníka do veřejné části aplikace, který má pouze

omezené možnosti. Může si prohlížet veřejnou část webové aplikace nebo se přihlásit pod svým uživatelským jménem. Po splnění podmínek pro přihlášení se role návštěvníka mění na roli uživatele. V rámci hierarchie rolí disponuje návštěvník nejnižšími právy, která následně dědí všechny ostatní role. Proto je tento případ užití společný pro všechny role. Po úspěšném přihlášení uživatele do systému má již uživatel k dispozici další případy užití. Přehled těchto případů užití, včetně rozdělení na základní role je zobrazen na následujícím obrázku (Obr. 13). Specifikace těchto případů užití je popsána v tabulkách níže (Tab. 2 a Tab. 3). Další případy užití je možné najít v příloze.



Obr. 13 Use Case Uživatelé

Tab. 2 Příklad užití: Přihlášení uživatele

Příklad užití: Přihlášení návštěvníka aplikace
ID:1
Stručný popis: Návštěvník aplikace se přihlašuje do interní části aplikace
Hlavní aktéři: Neregistrovaný návštěvník aplikace
Vedlejší aktéři: Žádní
Vstupní podmínky: Žádné
Hlavní scénář: <ol style="list-style-type: none"> <li>1. Nepřihlášený návštěvník aplikace si vybere příkaz přihlásit se do systému.</li> <li>2. Aplikace ho vyzve k zadání přihlašovacích údajů</li> <li>3. Aplikace zkontroluje validitu zadaných dat a poté ověřuje identitu uživatele v databázi.</li> <li>4. Pokud systém nenalezne daného uživatele v databázi, vyzve uživatele k opakovanému zadání.</li> <li>5. V případě splnění všech podmínek je návštěvníkovi změněna role na <i>Uživatel</i> a je přihlášen do systému.</li> </ol>
Výstupní podmínky: Návštěvník se úspěšně přihlásil do interní části aplikace
Alternativní scénáře: Žádné

Příklad užití - přihlášení uživatele, je první interakcí návštěvníka s interní částí webové aplikace. Po tomto případu užití se návštěvník stává uživatelem a na základě dalších přiřazených rolí jsou mu k dispozici odpovídající akce systému. Specifikace tohoto případu je uvedena v tabulce výše (Tab. 2).

Tab. 3 Příklad užití - Prohlížet události

Příklad užití: Prohlížet události
ID:2
Stručný popis: Uživatel si zobrazuje přehled událostí klubu
Hlavní aktéři: Uživatel
Vedlejší aktéři: Žádní
Vstupní podmínky: Tento případ užití je k dispozici pouze pro přihlášené uživatele do interní části IS.
Hlavní scénář: <ol style="list-style-type: none"> <li>1. Uživatel zahajuje případ užití volbou „zobraz události“</li> <li>2. Aplikace uživateli zobrazí stránku s tabulkou všech událostí, mezi kterými může uživatel filtrovat na základy typu události, data a dle určení události konkrétním skupinám uživatelů. Dále může události řadit podle kritérií.</li> </ol> Místo rozšíření: Zobrazit detaily události, Upravit událost. Vytvoř novou událost
Výstupní podmínky: Žádné
Alternativní scénáře: Žádné

Jedním ze základních a nejpoužívanějších případů užití bude prohlížení událostí. Příklad je podrobně popsán v tabulce výše (Tab. 3) Tato funkcionality systému umožní uživateli získat přehled o budoucích a již proběhlých akcích. Bude mít možnost mezi záznamy vyhledávat, filtrovat a řadit záznamy podle různých kritérií. Za návazné případy užití lze považovat přihlašování se na tyto události, případně zobrazení podrobností či editace události. Po zobrazení podrobností by měl uživatel získat kromě základních údajů o události také informaci o tom, komu je událost určena a kdo se na událost již přihlásil.

## 5.8 Doménové modely

Doménový model neboli strukturální představuje datovou strukturu aplikace. Jak již bylo uvedeno výše ve specifikaci datových požadavků, tak z hlediska datové struktury bude aplikace rozdělena na dvě části. Z tohoto důvodu je nutno vytvořit dva

datové modely. První pro webovou prezentaci klubu a druhý pro neveřejnou část, která představuje interní informační systém. Jelikož se tato práce zaměřuje především na neveřejnou část systému, je tato část systému popsána podrobněji.

Modely byly navrženy pomocí nástroje WebRatio na základě objektů a vazeb, které byly zachyceny v části analýzy požadavků a diagramů případů užití. Diagramy obsahují jak návrhové třídy, tak i vzájemné vazby mezi těmito třídami včetně datových typů jednotlivých atributů. Doménový model vytvořený pomocí nástroje WebRatio je ukládán ve formátu XML a je možné z něj vygenerovat datovou strukturu pro konkrétní databázový systém. Ve zmíněném případě se bude jednat o syntaxi pro databázi MS SQL. Doménový model nástroje WebRatio však nezobrazuje cizí klíče a to z důvodu nezávislosti na použité databázi. Ty je nutné při následné tvorbě databáze vytvořit „ručně“ nebo je zaznačit do modelu pomocí „společných“ atributů, ze kterých WebRatio při generování SQL souboru vytvoří odpovídající cizí klíče. WebRatio používá při tvorbě některých modelů odlišné způsoby interpretace než jiné nástroje pro toto použití. Jedním ze způsobů je zobrazení kardinality mezi jednotlivými třídami. Kardinalita je ve WebRatio zobrazována v opačném směru než je obvyklé při tvorbě ERD diagramů.

V doménových modelech je rozlišováno velké množství entit, které lze rozdělit do kategorií podle typu uchovávaných dat. V rámci vytvářené aplikace budou tyto kategorie odpovídat jednotlivým sekcím tedy skupinám stránek.

Za nejdůležitější kategorie v interní části aplikace lze považovat tréninky a zápasy, uživatele, evidenci pokladny, správu požadavků na náhradu cestovních nákladů, evidenci pokut, správu registračních poplatků, evidenci docházky a diskuzní fórum.

#### **Tréninky a zápasy:**

Entita: *Event* – ukládá veškeré informace o historických nebo budoucích událostech v klubu.

#### **Uživatelé**

Entita: *AspNetUsers* – spravuje veškeré informace o členech klubu. Mezi ně patří například osobní a přihlašovací údaje.

#### **Evidence pokladny**

Entita: *Payment* – popisuje jednotlivé platby v klubu.

#### **Správa požadavků na náhradu cestovních nákladů**

Entita: *TravelCompensation* – ukládá veškeré žádosti o úhradu cestovních nákladů. Entita obsahuje veškeré potřebné informace pro stanovení výsledné částky a identifikaci jízdy. Každý požadavek na úhradu nákladů nabývat dvou stavů a to schváleno/neschváleno a zapláceno/nezapláceno.

### **Evidence pokut**

Entita: *Penalty* – eviduje pokuty udělované členům klubu. Pokuty mají vazbu na pokladníka, uživatele a případně na událost, ke které se vztahují.

### **Správa registračních poplatků**

Entita: *RegistrationPayments* – spravuje registrační poplatky v klubu. Registrační poplatek se vztahuje vždy k dané sezóně ligové soutěže. Návazné tabulky jsou *AspNetUsers*, *Payment* a *Season*

### **Evidence docházky**

Entita: *AttendanceToEvent* – jedná se o vazební entitu, která propojuje členy klubu s událostí. Uchovává informaci o docházce uživatele na konkrétní událost.

### **Diskuzní fórum**

Entita: *Forum* – tato entita spravuje veškeré informace týkající se diskuzního fóra. Aplikace využívá funkcionality nástroje NET Forum společnosti JitBit. Datová struktura je převzata z výchozí datové struktury tohoto nástroje, a za pomoci předpřipravených konfiguračních souborů bylo fórum upraveno k potřebám této práce. Diskuzní fórum bylo nakonfigurováno tak, aby využívalo uživatelské účty, pod kterými se členové klubu přihlašují do systému. Z důvodu lepší přehlednosti datového modelu nebyly entity diskuzního fóra aplikace podrobněji popsány, ale v rámci struktury jsou znázorněny pouze jedinou entitou Forum.

Webová aplikace využívá pro správu uživatelů, rolí a práv nástroje Asp.NET Identity 2.1. Výchozí datová struktura tohoto systému byla pro potřeby aplikace částečně upravena. Mezi hlavní entity pro práci s uživateli a jejich právy patří *AspNetUsers*, *AspNetUserRoles*, *AspNetRoles* a také *AspNetUserLogins*.

### **AspNetUsers**

Eviduje údaje o uživateli. Jedná se především o osobní údaje a údaje o aktivitě uživatele v týmu.

### **AspNetUserRoles**

Tato třída umožňuje spravovat přiřazené role jednotlivých uživatelů. Každý člen klubu může mít přiřazeno více rolí.

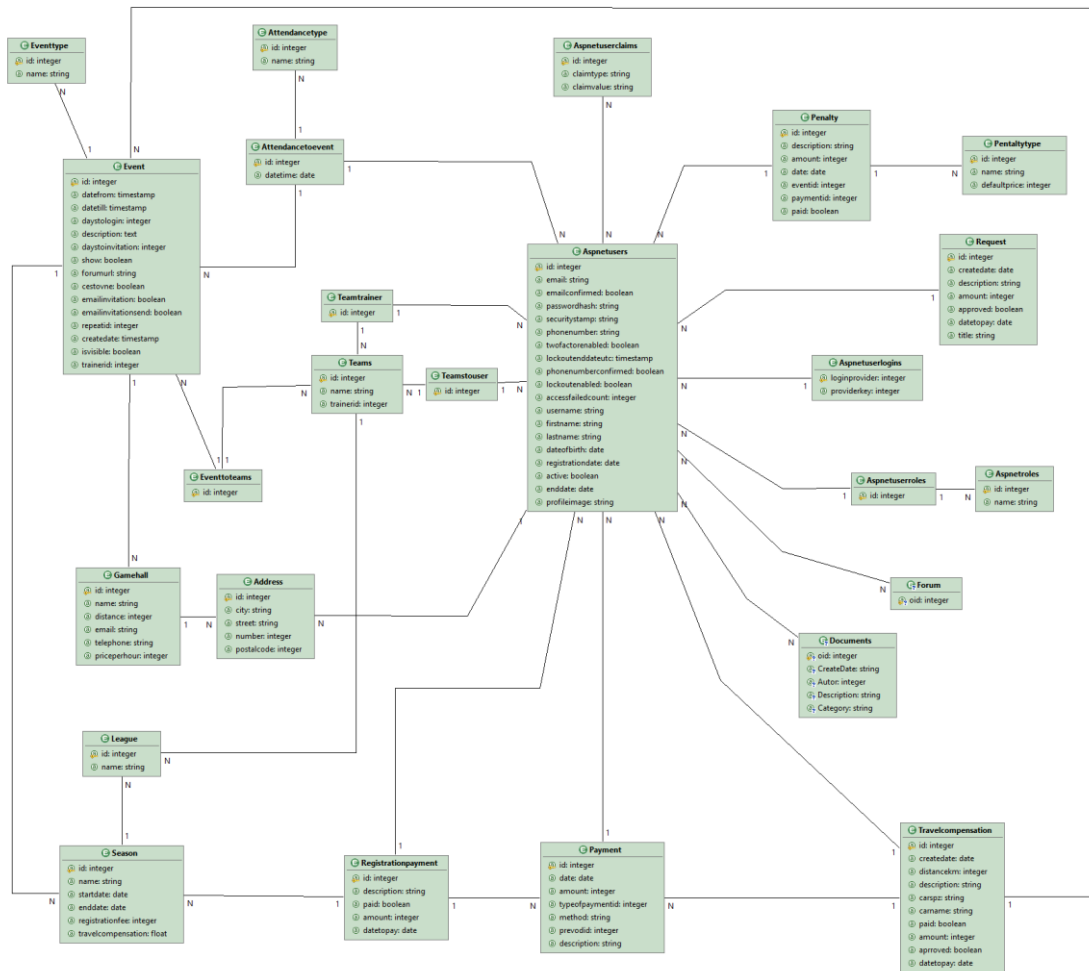
### **AspNetRoles**

Tato entita uchovává role v systému. Mezi role patří například Trenér, Předseda, Pokladník a Hráč.

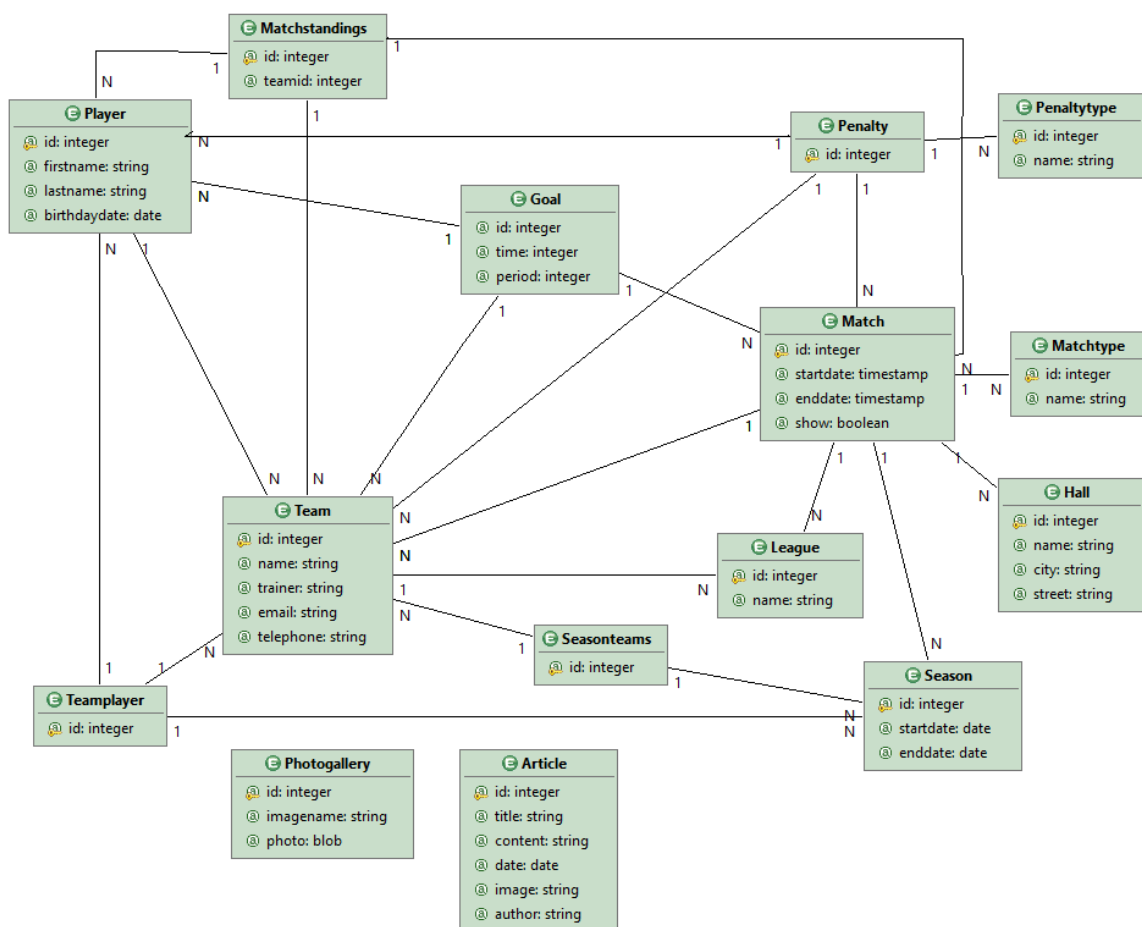
### **AspNetUserLogins**

Jedná se o standardní tabulku nástroje Asp.NET Identity. Entita loguje přihlášení jednotlivých uživatelů.





Obr. 14 Doménový model interní části aplikace



Obr. 15 Doménový model webové prezentace

## 5.9 IFML – Hypertextové modely

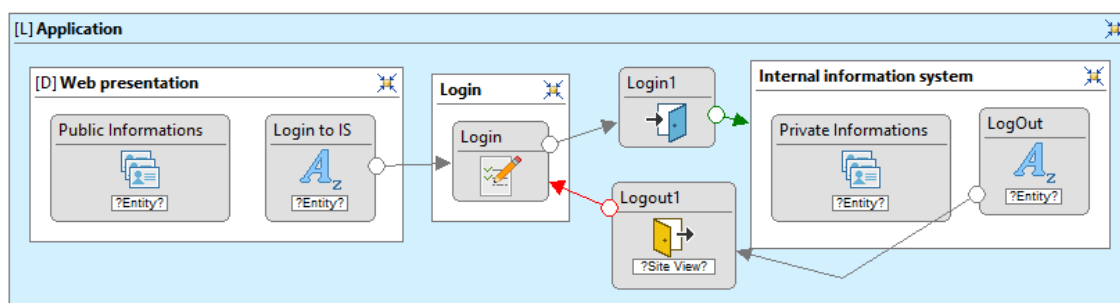
Jak již bylo zmíněno v teoretické části práce, tak IFML modely, a konkrétně hypertextové modely, patří mezi nejdůležitější modely pro návrh struktury a chování aplikace. Pro modelování budou použity pohledy na data tzv. „Site Views“, které umožňují nahlédnutí na systém z různých pohledů. Jelikož je struktura této aplikace velmi rozsáhlá, je nutné rozdělit hypertextové modely do skupiny na základně přiřazených rolí uživatelů.

Vzhledem k tomu, že každému z uživatelů může být přiřazeno více rolí současně, bude každý diagram znázorňovat pouze ty části systému, které jsou z pohledu daného uživatele nejpodstatnější. Ucelený pohled na pracovní prostředí daného uživatele je tedy možné následně jednoduše odvodit. Kompletní hypertextové modely dle jednotlivých rolí jsou součástí příloženého CD.

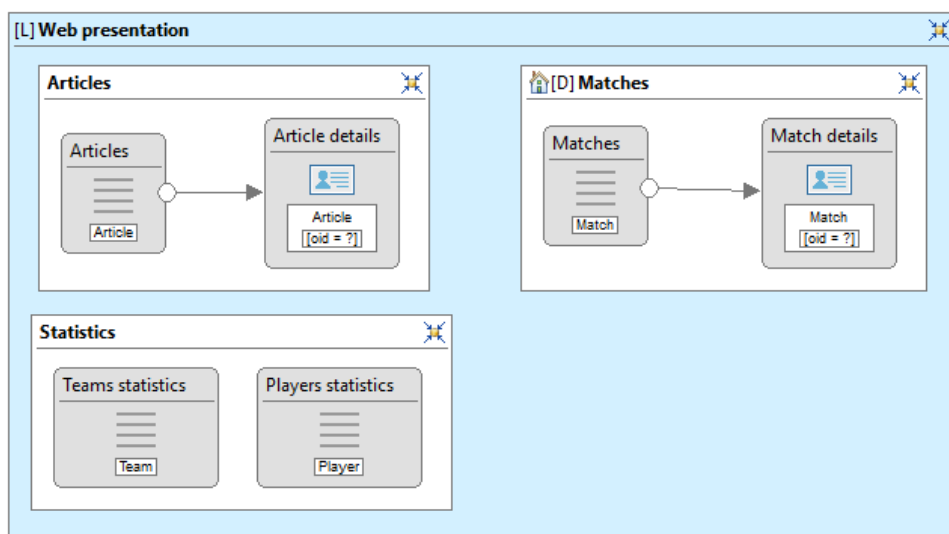
### 5.9.1 Neregistrovaný uživatel

Stejně jako ostatní pohledy, tak i tento vychází z případů užití pro odpovídající roli uživatele. Tento pohled (Obr. 16) se naskytne všem uživatelům při prvotním přístupu k aplikaci. Neregistrovaný uživatel má velice omezené možnosti – může přistupovat k veřejné části aplikace (webové prezentaci klubu) nebo se přihlásit do interní části aplikace (za předpokladu již vytvořeného účtu). Webová prezentace klubu, představující obrázek níže (Obr. 17), slouží k informování veřejnosti o aktualitách v klubu a dosažených výsledcích. Jsou zde k nahlédnutí články, statistiky a výsledky zápasů jednotlivých týmů klubu.

Uživatel se přihlásí vyplněním formuláře (uživatelské jméno a heslo). V případě úspěšného přihlášení dojde k přesměrování do neveřejné části aplikace a je mu zobrazen odpovídající pohled na základě přiřazených rolí. Registraci uživatelů provádí sekretář nebo předseda klubu.



Obr. 16 IFML model - neregistrovaný uživatel



Obr. 17 IFML model - webová prezentace

### 5.9.2 Uživatel

Pohled role *Uživatel* je omezen pouze na prohlížení údajů o členech klubu, pořádaných událostí, přispíváním do týmové diskuze a editací osobních údajů.

### 5.9.3 Hráč

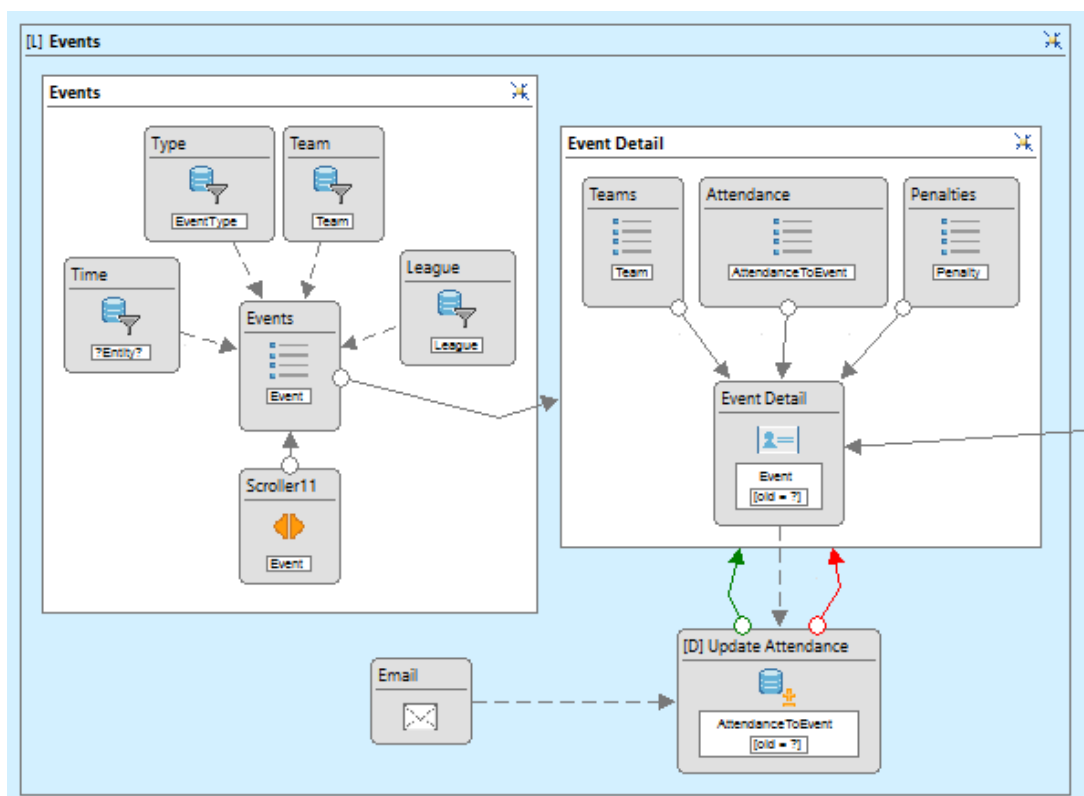
Uživatel s přiřazenou rolí *Hráč* má navíc oproti roli *Uživatel* tyto možnosti:

- přihlašování na tréninky a jiné události jemu určené (události, týkající se týmů, jehož je členem),
- podávání žádostí k proplacení nákladů za dopravu na konkrétní událost,
- nastavení dlouhodobých absencí.

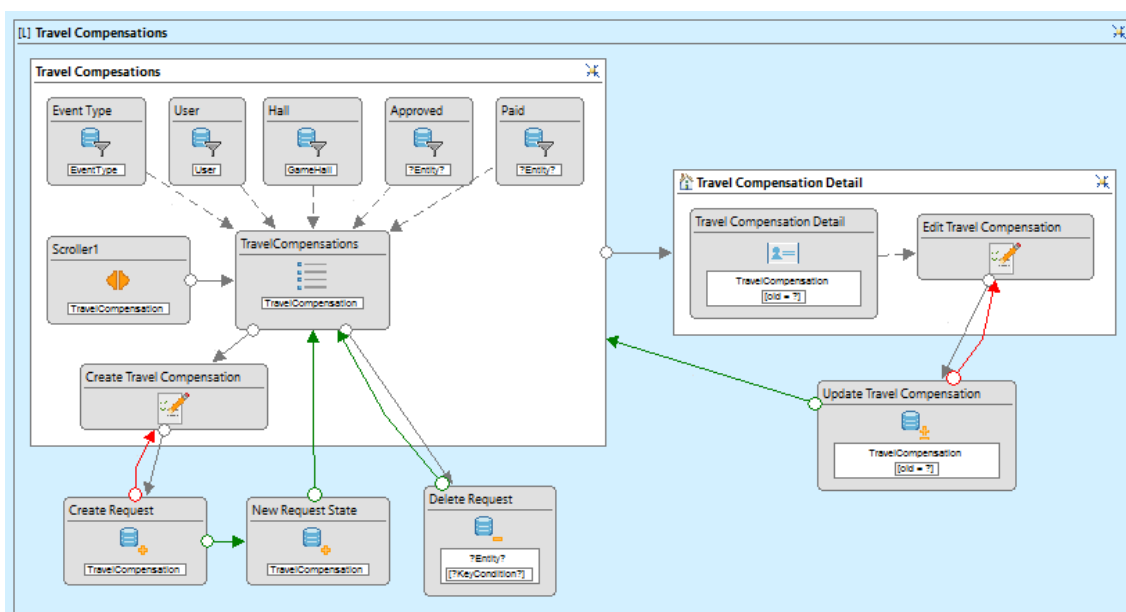
Pro *Hráče* je nejdůležitější sekci *Tréninky a zápasy*. Při vstupu do této sekce se *Hráči* zobrazí výpis všech budoucích událostí, na které se může přihlásit. *Uživatel* si může následně zvolit filtr událostí, který umožní zobrazení událostí dle jeho požadavků. Má možnost si zobrazit nejen události, na které se může přihlásit, ale také všechny události, které klub pořádá. Případně si může události filtrovat dle týmů nebo ligových soutěží. *Hráči* budou také k dispozici podrobnosti, které se týkají dané události. Mezi zmíněné podrobnosti patří typ zvolené události, datum, čas a místo konání, dále seznam týmů, pro které je událost určena, seznam již přihlášených osob a seznam udělených pokut vztahujících se k této události. IFML model přihlášení na událost je znázorněn na obrázku níže (Obr. 18).

Za další důležitou sekci lze považovat *Cestovní příkazy*, ve které mohou *hráči* zadávat své požadavky na proplacení cestovních příkazů. Jelikož se požadavek vždy vztahuje k určité události, budou *hráči* nabídnuty události, kterých se zúčastnil

a systém mu automaticky vypočítá výši náhrady dle vzdálenosti a sazby Kč/km. Sazbu stanovuje sekretář nebo předseda klubu, vždy pro danou ligovou soutěž a rok. Po vytvoření požadavku může uživatel sledovat jeho stav, tedy zda je již schválen či nikoliv. Je-li žádost schválena pověřenou osobou (sekretář či předseda), může pokladník tuto žádost proplatit. Vytvoření požadavku znázorňuje obrázek níže (Obr. 19).



Obr. 18 IFML model - přihlášení na událost

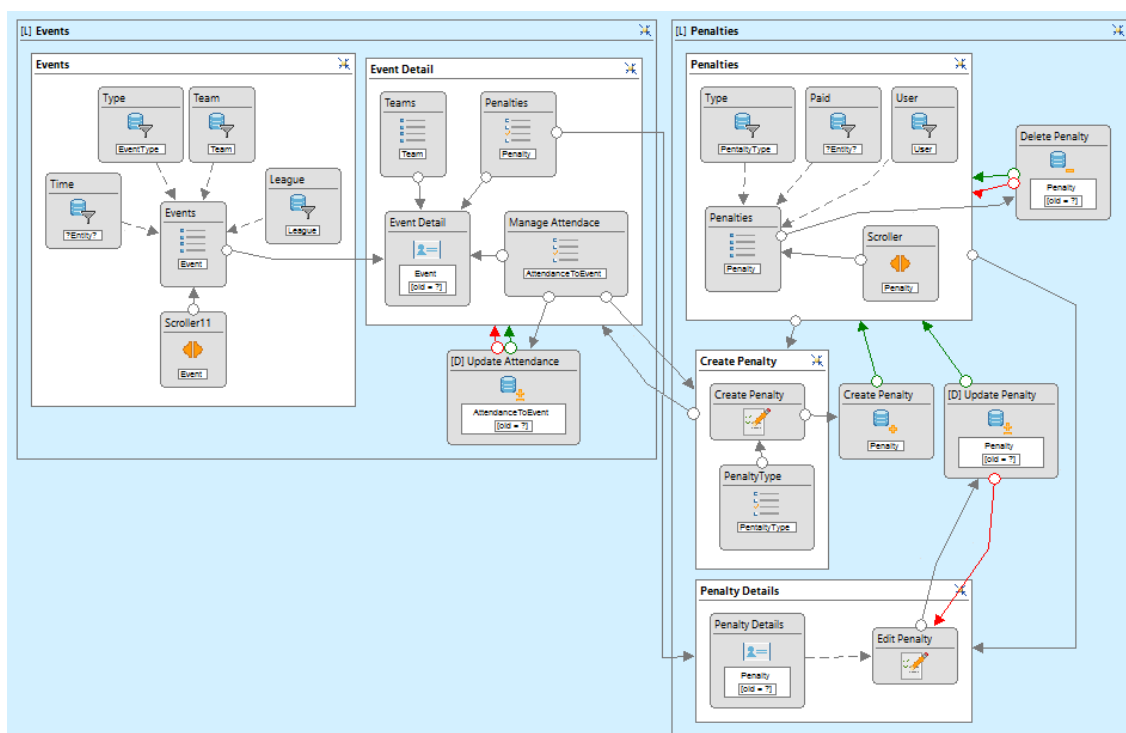


Obr. 19 IFML model - vytvoření požadavku na úhradu cestovního

#### 5.9.4 Trenér

Pro roli *Trenér* patří mezi nejdůležitější procesy v rámci aplikace vyhodnocování docházky k vybrané události a také „ruční“ přihlašování jednotlivých hráčů. V případě, že se jedná o budoucí událost, má trenér oprávnění měnit stav již přihlášených hráčů (stavy: přijde, možná přijde, nepřijde), nebo může ještě nepřihlášené hráče přidat „ručně“. Dále má k dispozici možnost opětovného zaslání emailové pozvánky na událost. Tato pozvánka bude doručena na email všem hráčům, kteří se k danému okamžiku k události nevyjádřili.

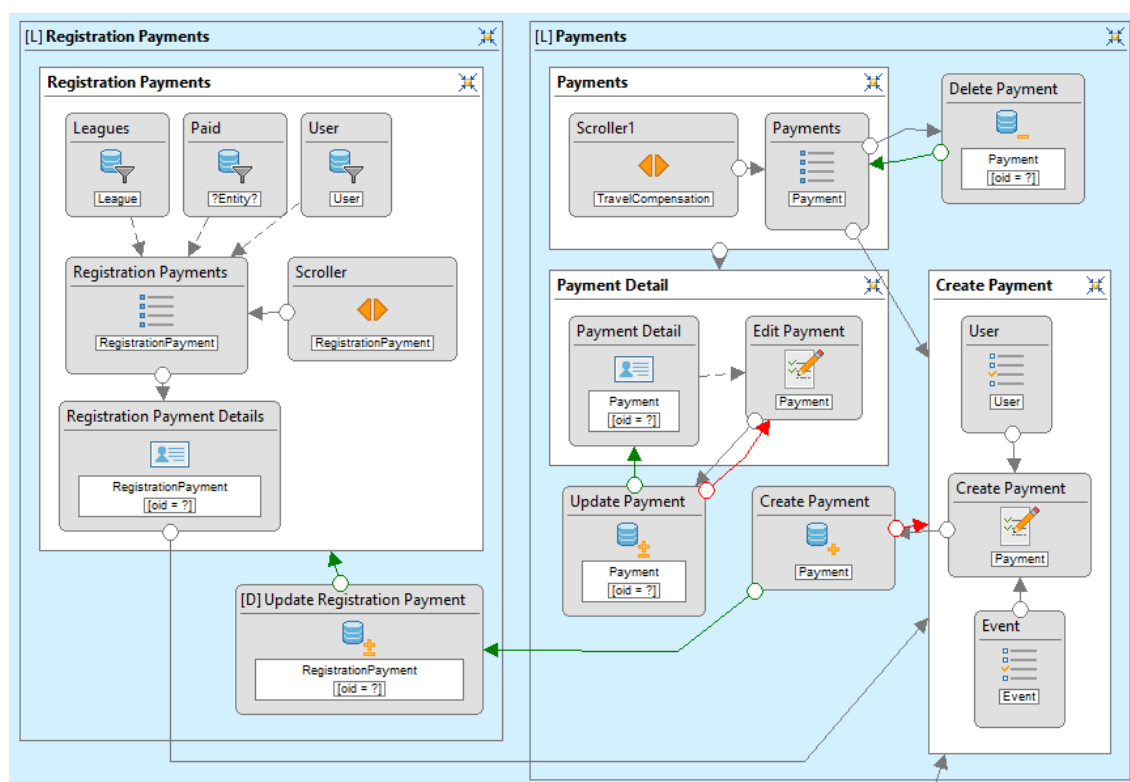
Trenér má dále možnost po proběhnutí události, upravit docházku hráčů dle skutečnosti. Dojde-li k neshodě či jiným prohřeškům, smí udělit hráči pokutu. Při vytváření zvolí typ pokuty, ke kterému mu bude následně automaticky nabídnuta částka, která byla předem stanovena pro daný typ pokuty. Má ovšem právo hodnotu zvolené pokuty podle potřeby změnit. Po zadání částky může také přidat poznámku. Jakmile dojde k vytvoření pokuty, bude trenér přesměrován zpět na událost. Proces vyhodnocování docházky a přidávání pokut je znázorněn na obrázku níže (Obr. 20). Trenér může také upravovat soupisky přiřazených týmů přidáváním a odebráním jednotlivých hráčů.



Obr. 20 IFML model - vyhodnocení docházky

### 5.9.5 Pokladník

Pohled pro roli *Pokladník* má oproti roli *Uživatel* dostupnou sekci *Pokladna*. V této sekci vidí pokladník aktuální peněžní stav pokladny. Dále je zde seznam veškerých provedených plateb, které proběhly v rámci klubu. Smí zde filtrovat platby podle druhu (příjem nebo výdaj), typu (za co byla platba provedena) a také podle pokladníka, který vybranou platbu provedl. Z hlavní strany sekce může také vytvořit novou platbu. Ovšem ve většině případů bude pokladník přidávat údaj o platbě přímo z konkrétní sekce. Podívá-li se pokladník například do sekce *Cestovní příkazy* a některé záznamy jsou již schváleny sekretářem či předsedou, má možnost ihned přidat platbu k tomuto požadavku. Tento krok je znázorněn na obrázku (Obr. 21). Zvolí-li si tento postup, budou mu ve formuláři pro přidání nové platby automaticky vyplněny veškeré potřebné údaje. Obdobný postup může zvolit v sekci *Pokuty a Registrační příspěvky*.



Obr. 21 IFML model - vytvoření platby

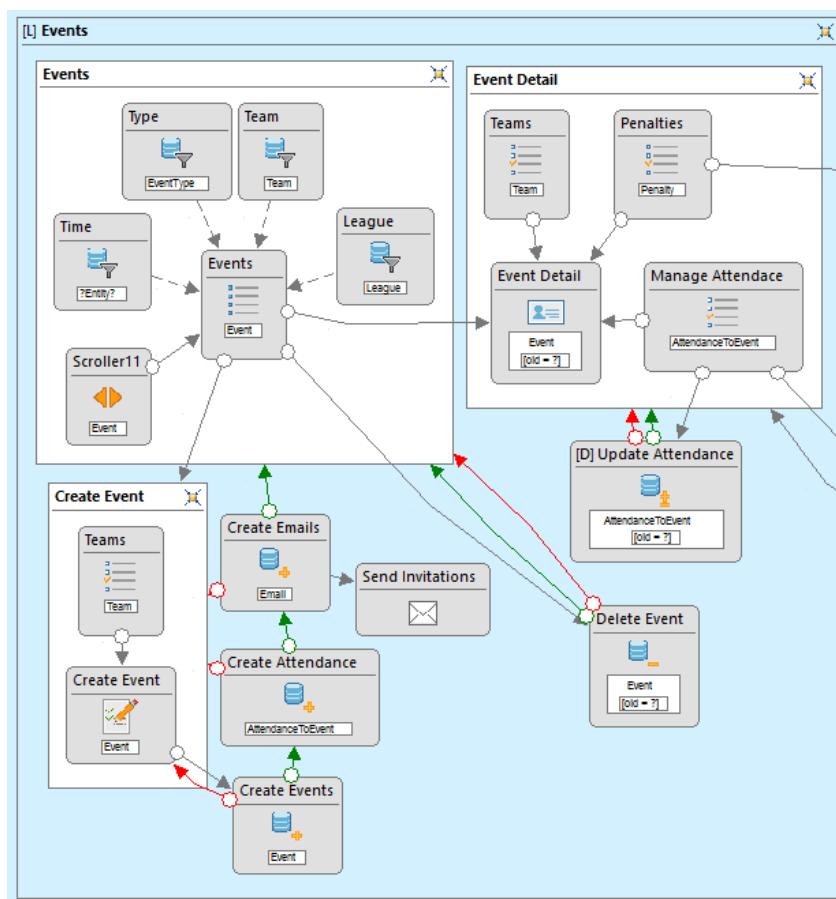
### 5.9.6 Sekretář

Uživatelská role *Sekretář* přebírá pohledy z role *Trenér* a přidává další administrativní pohledy na aplikaci. Sekretář má možnost přidávat nové týmy, ligové soutěže k jednotlivým týmům, sezóny, vytvářet nové typy pokut a k nim odpovídající částky, stanovovat registrační poplatky, zakládat události a v neposlední řadě má na starosti správu hal (vzdálenosti, ceníky). Mezi nejdůležitější sekce pro roli *Sekretář* lze považovat *Tréninky a zápasy*, *Registrační příspěvky* a *Cestovní příkazy*. Jak již bylo zmíněno, Sekretář má právo vytvářet nové události (Obr. 22). Kromě jednorázové události má zde také možnost vytvoření více událostí najednou, což se týká událostí, které se pravidelně opakují. Při zadávání události si volí počet dnů stanovujících, kolik dnů před jejím začátkem bude událost zveřejněna pro přihlašování.

Dále musí sekretář vybrat konkrétní týmy, pro které bude událost určena. Na základě vybraných týmů se po vytvoření události vygeneruje seznam hráčů, kterým může být po zveřejnění události zaslána pozvánka na email. Pokud má být emailová pozvánka zaslána automaticky, je nutné tuto volbu označit při vytváření události. Sekretář má také možnost stanovování registračních poplatků na konkrétní sezónu a ligu. Sekretář může hráčům přiřadit poplatky jak hromadně, tak také jen pro určitého člena. Mezi další činnost, kterou má sekretář v rámci role přiřazenou, patří proces schvalování požadavků pro náhradu nákladů na cestování. V této sekci je nutné, aby sekretář individuálně každou žádost posoudil a v případě souhlasu schválil.



V opačném případě musí kontaktovat žadatele a po vzájemné dohodě buď požadavek schválit, nebo jej smazat. Schvalování požadavků provádí sekretář přes akci „Upravit“ v seznamu požadavků, kde následně změní stav požadavku na schválený.



Obr. 22 IFML model - vytvoření události

### 5.9.7 Předseda

Pohled pro roli *Předseda* přebírá práva a pohledy rolí *Sekretář* a *Pokladník*. Předseda má tedy přístup ke všem pohledům v systému a z toho důvodu nebude tento pohled podrobněji popisován.

## 5.10 Prezentační návrh

Jediným požadavkem ze strany klubu pro prezentační návrh aplikace bylo ponechat design a barevnou škálu již existující webové prezentace klubu. Je tedy nutné, aby vizuální část nově vytvořené aplikace byla přizpůsobena těmto požadavkům. Důvodem k tomuto požadavku bylo především to, že stávající návštěvníci stránek si již na vzhled a rozvržení aplikace zvykli. Interní část systému by měla vzhledově a graficky

korespondovat s veřejnou částí. Oproti veřejné části, kde je grafický vzhled důležitý, je ale v interní části systému nutná především přehlednost a snadná orientace. Do interní části práce budou mít přístup pouze členové klubu, a proto zde není kladen vysoký důraz na grafickou část. Prezentační model veřejné části aplikace je znázorněn na obrázku níže (Obr. 23) spolu s ukázkou interní části aplikace (Obr. 24). Po přihlášení do aplikace je uživateli vygenerováno navigační menu dle rolí, které mu byly v systému přiděleny. V rámci tohoto menu může mít ale přístupné jen určité ovládací prvky. Další ukázky aplikace lze najít v příloze B.

Přihlášení do IS

# COMMIX TJ SOKOL BŘECLAV

FLORBALOVÝ KLUB

NOVINKY ČLÁNKY O NÁS TÝM VÝSLEDKY A STATISTIKY TRÉNINKY  
FOTOGALERIE KONTAKTY

## NOVINKY

**ÚŽASŇÁKOVI OVLÁDLI MEMORIÁL BR. JOSEFA JANKA**  
01.01.0001 0:00:00 | Vytvořil: **Administrátor**

Poslední neděli v **uplynulém** roce se v hale TJ Lokomotiva Břeclav sešli florbaloví nadšenci všech věkových kategorií, aby odehráli druhý ročník Memoriálu br. Josefa Janka. Nejlépe si v něm vedl tým...

[ČTI DÁLE](#)

**STRÁNKY V PROVOZU !!**  
01.01.0001 0:00:00 | Vytvořil: **Administrátor**

Dnešním dnem byli úspěšně vytvořeny internetové stránky **Dnešním dnem byly úspěšně spuštěny stránky klubu. Nedostatky budou časem odstraněny**

...

[ČTI DÁLE](#)

### POŘADÍ - SOKOL

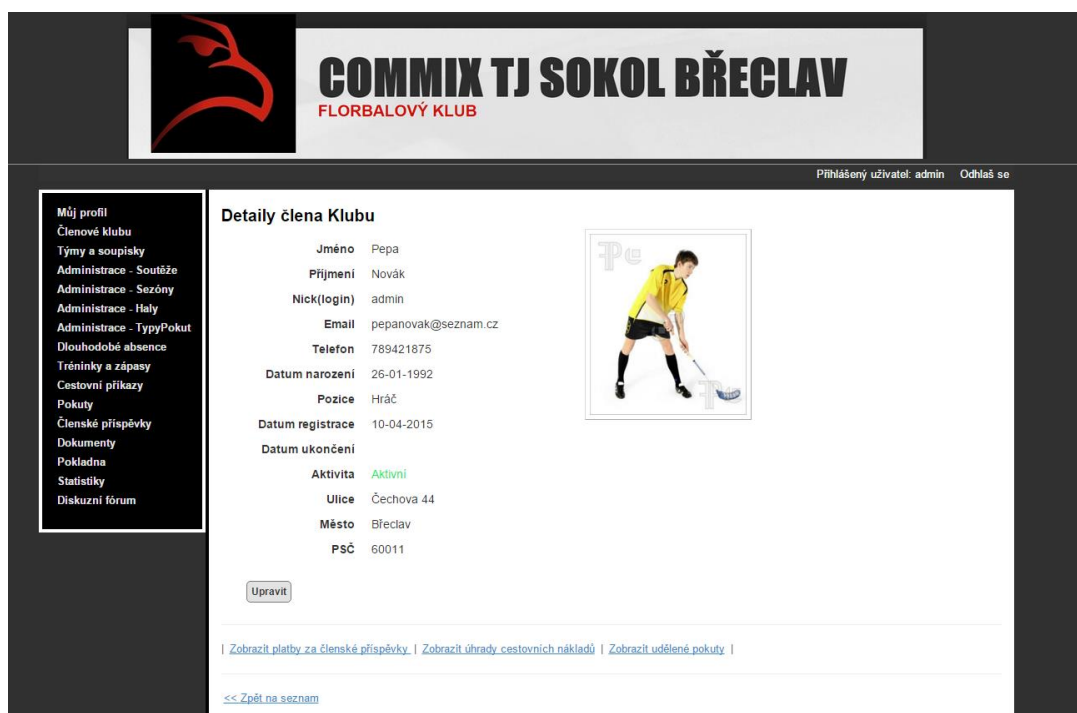
Tým	Body
Břeclav A	1
Břeclav I	0
Hovorany	0
Lanžhot	0
Velké Bílovice	0
Valtice	0
Lokomotiva	0

### PŘÍŠTÍ ZÁPASY

07.07.2016 0:00:00  
Hovorany - Lanžhot

[ZOBRAZ VŠE](#)

Obr. 23 Veřejná část aplikace - úvodní strana



**COMMIX TJ SOKOL BŘECLAV**  
FLORBALOVÝ KLUB

Přihlášený uživatel: admin Odhlaš se

**Můj profil**  
Členové klubu  
Týmy a soupisky  
Administrace - Soutěže  
Administrace - Sezóny  
Administrace - Haly  
Administrace - TypyPokut  
Dlouhodobé absence  
Tréninky a zápasy  
Cestovní příkazy  
Pokuty  
Členské příspěvky  
Dokumenty  
Pokladna  
Statistiky  
Diskuzní fórum

**Detaily člena Klubu**

Jméno	Pepa
Příjmení	Novák
Nick(login)	admin
Email	pepanovak@seznam.cz
Telefon	789421875
Datum narození	26-01-1992
Pozice	Hráč
Datum registrace	10-04-2015
Datum ukončení	
Aktivita	Aktivní
Ulice	Čechova 44
Město	Břeclav
Psč	60011

[Zobrazit platby za členské příspěvky](#) | [Zobrazit úhrady cestovních nákladů](#) | [Zobrazit udělené pokuty](#) |

[<< Zpět na seznam](#)

Obr. 24 Interní část aplikace - detaily uživatele

## 5.11 Implementace a nasazení

Implementace webové aplikace proběhla na základě nefunkčních požadavků a případů užití v prostředí ASP.NET MVC 5 za pomoci nástroje Microsoft Visual Studio. Pro realizaci aplikace byl využit programovací jazyk C# a pro uložení dat databáze Microsoft SQL Server.

Základní struktura databáze byla vytvořena na základě doménových modelů z nástroje WebRatio, zatímco hypertextové modely sloužily jako základ pro funkční a logickou strukturu aplikace.

Celá aplikace byla realizována pomocí nástrojů společnosti Microsoft:

- Visual Studio 2014,
- ASP.NET MVC 5,
- IIS EXPRESS,
- SQL SERVER 2014,
- Programovací jazyk C#.

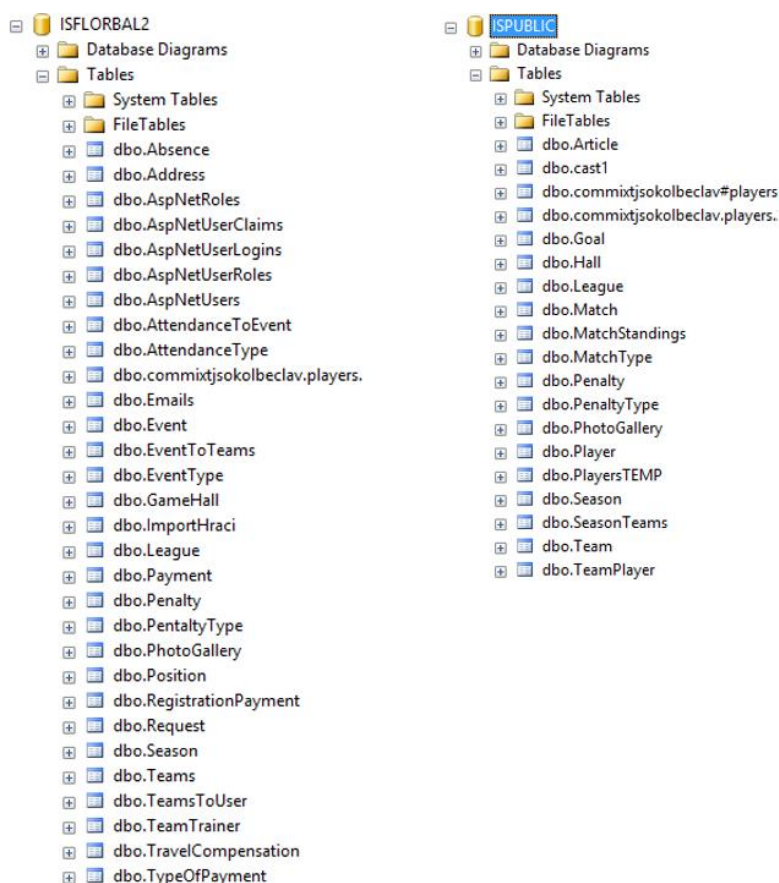
Implementace samotné aplikace lze rozdělit do dvou fází a to vytvoření databáze a vývoj aplikace. Jednotlivé fáze budou podrobněji popsány v následujících kapitolách.

### 5.11.1 Vytvoření databáze

První fází implementace bylo vytvoření databáze, která vznikla pomocí nástrojů WebRatio a Microsoft Management Studio. Tvorba databáze proběhla v několika krocích – fyzická definice tabulek, doplnění integritních omezení a nahrání samotných dat. V prvním kroku bylo nutné z doménového modelu nástroje WebRatio vygenerovat SQL soubor, který se následně spustil v Microsoft Management Studiu. Po tomto kroku byly vytvořeny tabulky v databázi.

V následujícím kroku byly doplněny integritní omezení, které bylo nutné doplnit „ručně“ v Management Studiu. Jako další možnost se nabízelo využití uživatelského rozhraní GUI nebo SQL Syntaxe (příkaz „New Query“).

Naplnění dat proběhlo načtením vygenerovaného SQL souboru a dávkovým spuštěním příkazu, tedy stejně jako v první fázi. Zmíněný soubor byl vygenerován ze stávající webové prezentace klubu.



Obr. 25 Struktura databáze

### 5.11.2 Vývoj a implementace aplikace

Po vytvoření databáze a naplnění daty se přestoupilo k samotné implementaci aplikace. Jak bylo již zmíněno v přechozí kapitole, byl v rámci platformy ASP.NET využit

vzor MVC, který pracuje se striktně definovanou tří vrstvou architekturou. Architektura konceptu MVC rozděluje vrstvy na Model, View a Controller. Mezi výhody tohoto řešení patří především vyšší kontrola nad vývojem aplikace a mnohem lepší znovu-použitelnost kódu. Implementace proběhla v prostředí Microsoft Visual Studio 2014. Pro přístup k databázovým údajům z aplikace byl využit Entity Framework (součást frameworku .NET) a speciální dotazovací jazyk LINQ. Ten umožňuje přistupovat k údajům v relační databázi a zpětně je načítat do aplikace ve formě objektů (objektově-relační mapování). Načtené objekty je možné poté v databázi jednoduše upravovat nebo mazat. Implementace bude probíhat pomocí Entity Frameworku metodou Database – First. Tato metoda je vhodná v momentě, kdy je již vytvořeno databázové schéma a je nutné převést strukturu do aplikačního kódu. Framework umožňuje vygenerování definic „datových“ tříd pro jednotlivé entity databáze. Aplikace využívá diskuzního fóra .NET Forum společnosti JitBit, které plně splňuje veškeré požadavky na něj kladené. Z důvodu nalezení již existujícího řešení, které je zdarma a plně vyhovuje všem požadavkům, nebylo potřeba vyvíjet vlastní řešení. Fórum je plně integrováno do aplikace a využívá stávajících uživatelských účtů. Celý projekt včetně ukázek aplikace je možné nalézt na přiloženém CD.

#### 5.11.2.1 Autentizace a role aplikace

Aplikace využívá pro správu autentizace a rolí, již existujícího řešení v podobě ASP.NET Identity 2.1.. Toto řešení umožňuje využívat uživatelsky přizpůsobené providery. Stávající providery bylo nutné pro potřeby práce upravit. Stejně tak bylo nutné upravit strukturu odpovídajících tabulek v databázi – tabulky *AspNetUsers* a *AspNetUserRoles*. Toto řešení umožňuje snadnou práci s uživatelskými účty a rolemi.

#### 5.11.2.2 Struktura a dělení kódu

Samotná webová aplikace se skládá z projektů: *Florbal\_IS* (interní informační systém) a *Public\_IS* (webová prezentace klubu). Struktura adresářů obou projektů je shodná. Dle vzoru MVC jsou projekty rozděleny do složek odpovídající jednotlivým vrstvám. Jedná se například o adresáře controllers, DAL, models a views.

#### Controllers

Složka obsahující třídy, jejichž nastavení ovlivňuje chování jednotlivých komponent a logiku celé aplikace. Při každé interakci uživatele s aplikací je volána odpovídající akce Controlleru – tzv. *GET* nebo *POST* událost. Událost zajišťuje, že jsou uživateli vrácena požadovaná data ve formě HTML stránky, včetně provedení dalších akcí na pozadí systému – např. uložení či smazání dat z databáze. Každá logická sekce aplikace odpovídá jedné třídě (Controlleru). Pro evidenci registračních poplatků slouží například Controller *RegistrationPaymentController*.

## DAL

Adresář DAL představuje třídy obstarávající práci s daty v databázi. Každá třída zabezpečuje přístup ke konkrétní entitě datového modelu. Tento přístup k datům odpovídá návrhovému vzoru Repository.

## Models

Obsahuje definice tříd pro jednotlivé entity z doménového modelu. Třída obsahuje odkazy na použité knihovny, výčet atributů s jejich datovými typy a dalšími vlastnostmi. Může se jednat například o způsob zobrazení daného atributu nebo omezení v rámci povolených hodnot atributu.

## Views

Tento adresář obsahuje oddělené HTML stránky obsahující odkazy na prvky (objekty) zpracovávané na serveru. Každá podsložka adresáře Views slouží pro zobrazení příslušného Controlleru (entity modelu). Pro interpretaci stránek je využit Razor View Engine

## 5.12 Testování a nasazení

Aplikace byla testována vždy při každé iteraci tzv. vývojářským testováním, které obsahovalo tzv. UNIT testing, což představuje testování různých částí kódu. V tomto případě se testují možné varianty průchodů a následně je provedena kontrola reakcí systému na tyto průchody. Jedná se o uměle vytvářené případy, které se mohou během ostrého provozu vyskytnout.

Dále se kontrolovala vzájemná komunikace a funkčnost všech komponent v rámci celé aplikace.

Aplikace byla po vývojářském testování předána budoucím uživatelům systému a nyní probíhá tzv. akceptační testování, kdy využívají aplikaci při reálném použití. Pro testování v rámci klubu byla aplikace nasazena na již existující server, který má klub k dispozici. Uživatelé považují aplikaci za přínosnou, jelikož velmi oceňují možnost přihlášení se na událost skrze zasílané pozvánky na email. Není nutné se přihlašovat do aplikace zdlouhavým zadáváním jméno a hesla. Stačí jen kliknout na odpovídající odkaz v zaslané emailové pozvánce. Uživatelé dále vítají rychlé zobrazení statistik docházky pro jednotlivé hráče či sledování stavu jejich požadavků na náhrady cestovného.

Po otestování a akceptaci aplikace vybranými uživateli systému bude převeden provoz ze stávající webové prezentace na již novou aplikaci. Souběžně s tím budou všem členům klubu vytvořeny uživatelské účty do interní části aplikace a začnou již aplikaci využívat v ostrém provozu.

Z hlediska funkčních požadavků aplikace splnila stanovená zadání a všechny předpoklady.

### 5.13 Návrhy na vylepšení

Již v průběhu vývoje aplikace se objevily ze strany uživatelů návrhy na rozšíření aplikace z hlediska funkčnosti. Další požadavky také vznikaly ve fázi testování aplikace. Žádný z těchto nových požadavků však nebyl popsán v prvotní fázi analýzy. Z tohoto důvodu bude jejich analýza a případná realizace probíhat až v další iteraci po testování a odsouhlasení analyzovaných požadavků uvedených v této práci. Mezi další plánovaná rozšíření funkcionality aplikace patří:

- Předvyplnění a tisk formulářů – možnost tisku již předvyplněných formulářů přímo z aplikace.
- Zobrazení událostí ve formě kalendáře – zobrazení naplánovaných událostí prostřednictvím zabudovaného kalendáře.
- Zaslání reportů – automatického zaslání vybraných reportů na email.





## 6 Diskuze

První část této práce je věnována problematice vývoje webových aplikací a to zejména jejich metodikám a metodám. V současné době lze sledovat zvyšující se trend jak ohledně zavádění informačních systémů do menších firem, tak také do neziskových sektorů. Důvodem nárůstu zavádění IS u zmíněných subjektů je především uvědomění si jeho výhod a přínosů. Za hlavní výhody považují zvýšení efektivity své práce, úsporu času a mimo jiné daleko rychlejší přístup k potřebným informacím. Díky stále se zvyšujícímu pokrytí internetem, jeho dostupnosti a také schopnosti webových serverů spravovat složité aplikace, je možné dostat tyto systémy na Web.

Spolu s rozvojem IS rostou také požadavky na vznik nových metodik, které jsou vhodné pro vývoj webových aplikací. Na poli klasických desktopových aplikací již existují desítky metodik zabývajících se jejich vývojem. Mezi zmíněné metodiky patří například Rational Unified Process (RUP) či Enterprise Unified Process (EUP). Většina metodik je svázána s jazykem UML, nebo s některými jeho rozšířeními. Ovšem metodik, zabývajících se vývojem webových aplikací, není prozatím mnoho. Některé stávající metodiky používané pro klasické aplikace byly upraveny tak, aby je bylo možné použít s webovými aplikacemi, jako například UML Web Application Extension nebo UML-based Web Engineering. Většina těchto metodik ovšem nedisponuje vhodnou dokumentací a nástroji, které by usnadnily vývoj aplikací. Mezi příčiny vzniku metodik přímo pro webové aplikace patří zejména nedostatky klasickým desktopových metodologií. Za nevýhody je považována například špatná podpora modelem řízeného vývoje, stále rostoucí komplexnost moderních internetových stránek či nedostatečná podpora abstrakce návrhu.

Rozdíl mezi desktopovými a webovými aplikacemi je právě z pohledu vývoje potřeba definování tzv. hypertextového paradigmatu. Začaly se tedy vyvíjet metody založené právě na hypertextu jako například HDM (Object Oriented Hypermedia Design Method). V práci jsem se podrobně zaměřil zejména na novou notaci IFML, která je založena na svém předchůdci WebML. IFML byla v roce 2013 přijata jako standart sdružením OMG (Object Management Group).

IFML definuje v rámci modelování aplikace několik typů modelů, kde za nejdůležitější model můžeme považovat Doménový model a Hypertextový model. V případě doménových modelů, IFML přesně nedefinuje, jaký model se má využít. Je tedy zcela na uživateli, jaký datový model se rozhodne použít. Sami autoři doporučují využít objektový model notace UML, který využívá i nástroj WebRatio.

Metodika IFML příliš nepodporuje modelování ve fázi analýzy a to konkrétně specifikaci požadavků. Pro namodelování těchto požadavků je nutné využít jiných přístupů než IFML. V této práci byly pro toto použití zvoleny Use Case diagramy jazyka UML. Use Case diagramy ve fázi analýzy jsou následně podpořeny detailnější specifikací pomocí doménového modelu a hypertextových modelů. Tato kombinace nástrojů a metodik se jeví při vývoji jako velice vhodná.

Přínosem IFML je zejména možnost navigačního modelování neboli tvorba hypertextového modelu aplikace. Při porovnání s modely ostatních metodologií, využívající hypertextové paradigma, je tento model uživatelsky velmi srozumitelně navržen a popsán v rámci dokumentace. Při porovnávání jednotlivých prvků pro modelování webových aplikací bylo zjištěno, že většina metodik využívá srovnatelné sady těchto prvků, které může daný uživatel při modelování využít.

Většina metodik také definuje tzv. pohledy na data. IFML využívá pro tyto účely vlastní dotazovací jazyk WebML-OQL. Většina metodik naopak příliš neřeší prezentační část modelování, která vždy závisí na konkrétním použitém vývojovém nástroji. IFML disponuje pro tento model nástrojem WebRatio, který podporuje celý vývojový životní cyklus aplikace.

Nástroj WebRatio umožňuje vygenerování nejen struktury doménového modelu do použité relační databáze, ale disponuje také komponenty pro vygenerování prototypu fungující aplikace na základě definovaných modelů. Vygenerování funkční aplikace je v aktuální verzi WebRatio omezeno pouze na jazyk JAVA, což se může jevit jako velké omezení v případě implementace aplikace v jiném programovacím jazyce. Mezi výhody patří ukládání veškerých modelů ve formátu XML, s čímž souvisí snadná editace jinými nástroji. Modelování doménového modelu není ve WebRatiu pro začínající uživatele snadné, a to zejména pokud přecházíte z nástrojů podporující UML modely jako například Enterprise Architect. WebRatio definuje doménové modely trochu odlišným způsobem. Jedním ze způsobů je nezobrazování cizích klíčů, nebo také opačný způsob zobrazování kardinalit. Mezi další omezení patří složitý způsob definování cizích klíčů v rámci doménového modelu a jeho následný export do relačních DB.

Na základě zjištěných poznatků lze říci, že notace IFML spolu s podporou nástroje WebRatio je pro návrh a vývoj složitějších webových aplikací velice vhodná. Je ovšem nutno zvážit použití konkrétních modelů v řešeném projektu. Výhody IFML se nejvíce projeví právě u velkých projektů, jejichž implementace bude probíhat pomocí nástroje WebRatio a v jazyce JAVA. Nevýhodou je, jak již bylo zmíněno, nepodpora fáze analýzy – specifikace požadavků. Nedostatečnou podporu této fáze vývoje lze vyřešit využitím UML diagramů a případů užití. Tato kombinace technologií se jeví jako nejvhodnější variantou pro kompletní životní cyklus projektu.

Další vhodnou variantou pro vývoj webových aplikací se může jevit například metodika UWE, která stejně jako IFML implementuje hypertextový a navigační model a podporuje celý životní cyklus projektu. Tato práce se však touto metodikou podrobněji nezabývala, a proto není možné určit, zda by byla pro řešení problémové domény vhodnější volbou nežli metodika IFML.

V době psaní této práce nebylo dosud zveřejněno mnoho publikací, které by se zabývaly notací IFML a to především z důvodů zatím relativně nového přístupu, ale také díky stále probíhajícímu vývoji. Lze tedy očekávat, že se v budoucnosti začnou vyvíjet nová rozšíření a nástroje pro podporu této notace, včetně publikací zabývajících se tímto přístupem.

## 7 Závěr

Cílem práce bylo vytvoření webové aplikace florbalového klubu sloužící k automatizaci administrace klubu a to především pokladny, docházky, evidence dopravy a statistik za pomoci CASE nástroje WebRatio a jazyka IFML.

Práce se dělí na dvě části a to jak na teoretickou, tak také na praktickou. Pro splnění stanoveného cíle této práce bylo nejdříve nutno nastudovat problematiku zabývající se webovými aplikacemi a jejich vývojem. Literární řešerši k danému tématu se věnuje teoretická část, ve které byly zmíněny výhody i nevýhody webových aplikací a také metodiky pro jejich vývoj. Dále byly podrobně popsány nástroje a technologie, pomocí kterých došlo k realizaci výsledné aplikace. Uvedená část práce se zabývala také popisem procesu vývoje aplikací dle notace IFML, a to včetně jednotlivých modelů odpovídající daným fázím vývoje.

Za nejpodstatnější část práce lze považovat praktickou část, která se věnovala realizaci aplikace pro sportovní klub. Proces vývoje aplikace v praktické části probíhal v souladu s notací IFML a to od specifikace požadavků, přes tvorbu doménových modelů, navigačních modelů, prezentačních modelů, implementace až po testování a údržbu. Pro analýzu a tvorbu modelů aplikace bylo využito kombinací IFML a UML diagramů. Po úvodním představení klubu proběhla analýza procesů a sběr požadavků, na základě konzultace jak s vedením klubu, tak také s jeho členy. Dále byla popsána hierarchie všech rolí v rámci klubu. Analýza procesů byla završena specifikací případů užití, které byly následně namodelovány pomocí Use Case diagramů. Dalším krokem byla tvorba doménového modelu v nástroji WebRatio dle notace UML. Na základě vytvořených modelů byla aplikace následně implementována ve vývojovém nástroji Visual Studio a frameworku ASP.NET MVC spolu s databázovým nástrojem MS SQL Server. Po dokončení byla aplikace otestována a nasazena do zkušebního provozu.

Vytvořená aplikace obsahuje jak veřejnou část, která slouží k propagaci klubu veřejnosti, tak interní část, která je určena k podpoře klubové administrativy. Veřejná část aplikace informuje o činnosti klubu, pořádaných akcích a výsledcích zápasů. Interní část slouží k automatizaci administrativních činností.

V rámci diskuze byly shrnuty možnosti notace IFML, včetně jejich výhod a nevýhod. Byly popsány praktické zkušenosti při používání a popsána doporučení, která vyplynula z praktické části práce. Nakonec byly uvedeny možnosti využití notace IFML a diagramů jazyka UML.

Závěrem práce lze říci, že stanovený cíl byl splněn.



## 8 Literatura

- ARLOW, J., NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2. vyd. Brno: Computer Press, 2007, 567 s. ISBN 978-80-251-1503-9.
- BRAMBILLA, M. MD\*: The Model-Driven Star blog. *Videos of the Webinars on IFML and its integration with BPMN and UML* [online]. 2011 [cit. 2015-03-10]. Dostupné z: <http://www.modeldrivenstar.org/2013/11/IFML-BPMN-UML-videos-of-webinars.html>.
- BRAMBILLA, M., BUTTI, S. *Fifteen Years of Industrial Model-Driven Development in Software Front-End: from WebML to WebRatio and IFML* [online]. Milán: Politecnico di Milano, 2013 [cit. 2015-03-24]. Dostupné z: <http://dbgroup.como.polimi.it/brambilla/sites/dbgroup.como.polimi.it.brambilla/files/referencematerials/WebML-IFML-WebRatio-Novatica-EN.pdf>.
- BRAMBILLA, M., COMAI, S., FRATERNALI, P., MATERA, M. *Designing Web Applications with WebML and WebRatio* [online]. Milán: Politecnico di Milano, 2013 [cit. 2015-05-15]. Dostupné z: <http://www.webml.org/webml/upload/ent5/1/Chapter%209%20-%20WebML.pdf>.
- BRAMBILLA, M., FRATERNALI, P. *Interaction Flow Modeling Language: Model-Driven UI Engineering of Web and Mobile Apps with IFML*. Burlington: Morgan Kaufmann Publishers, 2014, 422 p., ISBN 978-0-12-800108-0.
- BUCHALCEVOVÁ, A. *Metodiky vývoje a údržby informačních systémů*. 1. vyd. Praha: Grada Publishing, 2005, 163 s. ISBN 80-247-1075-7.
- BRUCKNER, T., VOŘÍŠEK J., BUCHALCEVOVÁ A. a kol. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012, 357 s. ISBN 978-80-247-4153-6.
- BRUŽINA, M. *Návrh informačního systému pro internetový obchod*. Brno, 2010. Diplomová práce. Vysoké učení technické v Brně. Fakulta podnikatelská.
- CONALLEN, J. *Building Web applications with UML*. 2nd ed. Boston: Addison-Wesley, 2003, 468 p. ISBN 02-017-3038-3.
- EVJEN, B., HANSELMAN, S. *ASP.NET 3.5 v jazycích C# a Visual Basic: programujeme profesionálně*. 1. vyd. Brno: Computer Press, 2009, 791 s. ISBN 978-80-251-2069-9.
- GALLOWAY, J., WILSON, B., ALLEN, K. *Profesional APS.NET MVC 4*. Indianapolis: Wiley Publishing, 2012, 624 p. ISBN 978-1-118-34846-8.
- KONEČNÝ, V. *Integrované informační systémy: přednášky*. [online] [cit. 2015-05-1]. <http://www.mendelu.cz/user/konecny>
- MITTNER, J. *Metodika pro vývoj webových aplikací*. Praha, 2010. Diplomová práce. Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky

- MISTRY, R., MISNER, S. *Introducing Microsoft SQL Server 2014*. Published: Microsoft Press, 2014, 125 p. ISBN 978-073-5684-751.
- MS SQL Server 2014: Základ datové platformy s brutálním výkonem. *Business IT*. [online]. 2014 [cit. 2015-03-25]. Dostupné z: <http://www.businessit.cz/cz/ms-sql-server-2014-zaklad-datove-platformy-s-brutalnim-vykonem.php>.
- MILLETT, S. *Professional ASP.NET Design Patterns*. Indianapolis: Wiley Publishing, 2010, 720 p. ISBN 978-0-470-29278-5.
- OMG adopts the IFML standard, designed by WebRatio. *Webratio* [online]. 2013 [cit. 2015-03-10]. Dostupné z: <https://www.webratio.com/portal/content/en/news-detail/omg-adopts-ifml-standard-designed-by-webratio?link=oln203z.redirect&nav=page36.32>.
- OBJECT MANAGEMENT GROUP, INC. *Interaction Flow Modeling Language* [online]. 2015 [cit. 2015-03-10]. Dostupné z: <http://www.omg.org/spec/IFML/1.0/PDF>.
- PECINOVSKÝ, R. *Návrhové vzory*. 1. vyd. Brno: Computer Press, 2007, 527 s. ISBN 978-80-251-1582-4.
- RÁBOVÁ, I. *Podnikové informační systémy a technologie jejich vývoje*. V Tribun EU 1. vyd. Brno: Tribun EU, 2010, 139 s. ISBN 978-807-3995-997.
- ŘEPA, V., SYNÁČEK, V., HAMERNÍK, P. a kol. *Metodika vývoje informačního systému s pomocí nástroje Power Designer*. [online]. Praha: Vysoká škola ekonomická, 2006, 186 s. [cit. 2015-03-08]. Dostupné z: [http://is.vsfs.cz/el/6410/leto2009/N\\_PIS/687221/Metodika\\_vyvoje\\_IS\\_06\\_2006.pdf](http://is.vsfs.cz/el/6410/leto2009/N_PIS/687221/Metodika_vyvoje_IS_06_2006.pdf).
- ADVANTAGES OF WEB APPLICATIONS. *Paul Stanley Software*. [online]. [cit. 2015-06-03]. Dostupné z: <http://www.pssuk.com/AdvantagesWebApplications.htm>.
- WAZLAWICK, R. *Object-oriented analysis and design for information systems: modeling with UML, OCL, and IFML*. Waltham: Morgan Kaufmann Publisher, 2014, 376 p. ISBN 978-012-4186-736.
- WEBRATIO. *oficiální prezentace*. [online]. Milano: WebRatio, 2013. Dostupné z: <http://www.webratio.com/portal/content/en/how-it-works>.
- ZELENKA, P. *WebML projektování webových aplikací*. Interval.cz [online]. 2003 [cit. 2015-03-5]. Dostupné z: <http://interval.cz/clanky/webmlprojektovaniwebovychaplikaci/>.

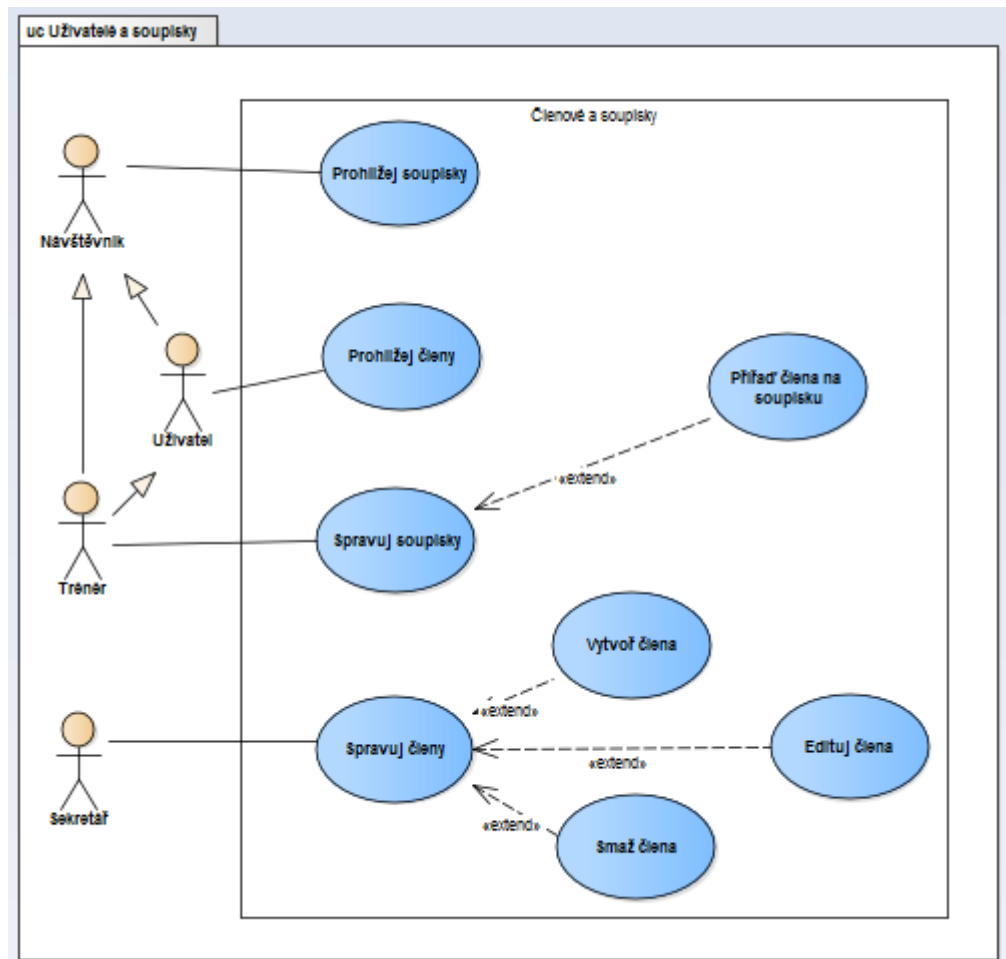
## 9 Seznam obrázků

Obr. 1 Princip webového systému .....	18
Obr. 2 Use Case Model - vazba "Include" .....	23
Obr. 3 "Top Level" Use Case Modelu .....	24
Obr. 4 Cyklus vývoje dle notace IFML .....	26
Obr. 5 Propojení IFML modelů s již existujícími metodikami.....	27
Obr. 6 Integrace nástrojů skrze model XMI 0 .....	27
Obr. 7 Strukturální model jazyka IFML .....	31
Obr. 8 Odvozený model jazyka IFML.....	32
Obr. 9 Kompoziční model aplikace - IFML .....	33
Obr. 10 Navigační model aplikace - IFML.....	34
Obr. 11 Prezentační model aplikace - IFML .....	35
Obr. 12 Uživatelské skupiny.....	50
Obr. 13 Use Case Uživatelé .....	52
Obr. 14 Doménový model interní části aplikace .....	57
Obr. 15 Doménový model webové prezentace .....	58
Obr. 16 IFML model - neregistrovaný uživatel .....	59
Obr. 17 IFML model - webová prezentace .....	60
Obr. 18 IFML model - přihlášení na událost .....	61
Obr. 19 IFML model - vytvoření požadavku na úhradu cestovního.....	62
Obr. 20 IFML model - vyhodnocení docházky .....	63
Obr. 21 IFML model - vytvoření platby .....	64
Obr. 22 IFML model - vytvoření události .....	65
Obr. 23 Veřejná část aplikace - úvodní strana .....	66
Obr. 24 Interní část aplikace - detaily uživatele .....	67
Obr. 25 Struktura databáze .....	68
Obr. 26 Diagram případů užití - práce s uživateli a soupiskami.....	81
Obr. 27 Diagram případů užití - události.....	82
Obr. 28 Diagram případů užití - členské příspěvky a pokuty .....	83
Obr. 29 Diagram případů užití - Webová aplikace .....	84
Obr. 30 Veřejná část aplikace - úvodní strana .....	85
Obr. 31 Veřejná část aplikace - detaily události .....	86
Obr. 32 Interní část aplikace - editace uživatele .....	87
Obr. 33 Interní část aplikace - editace události.....	87
Obr. 34 Interní část aplikace - výpis žádostí o cestovní náhrady .....	88

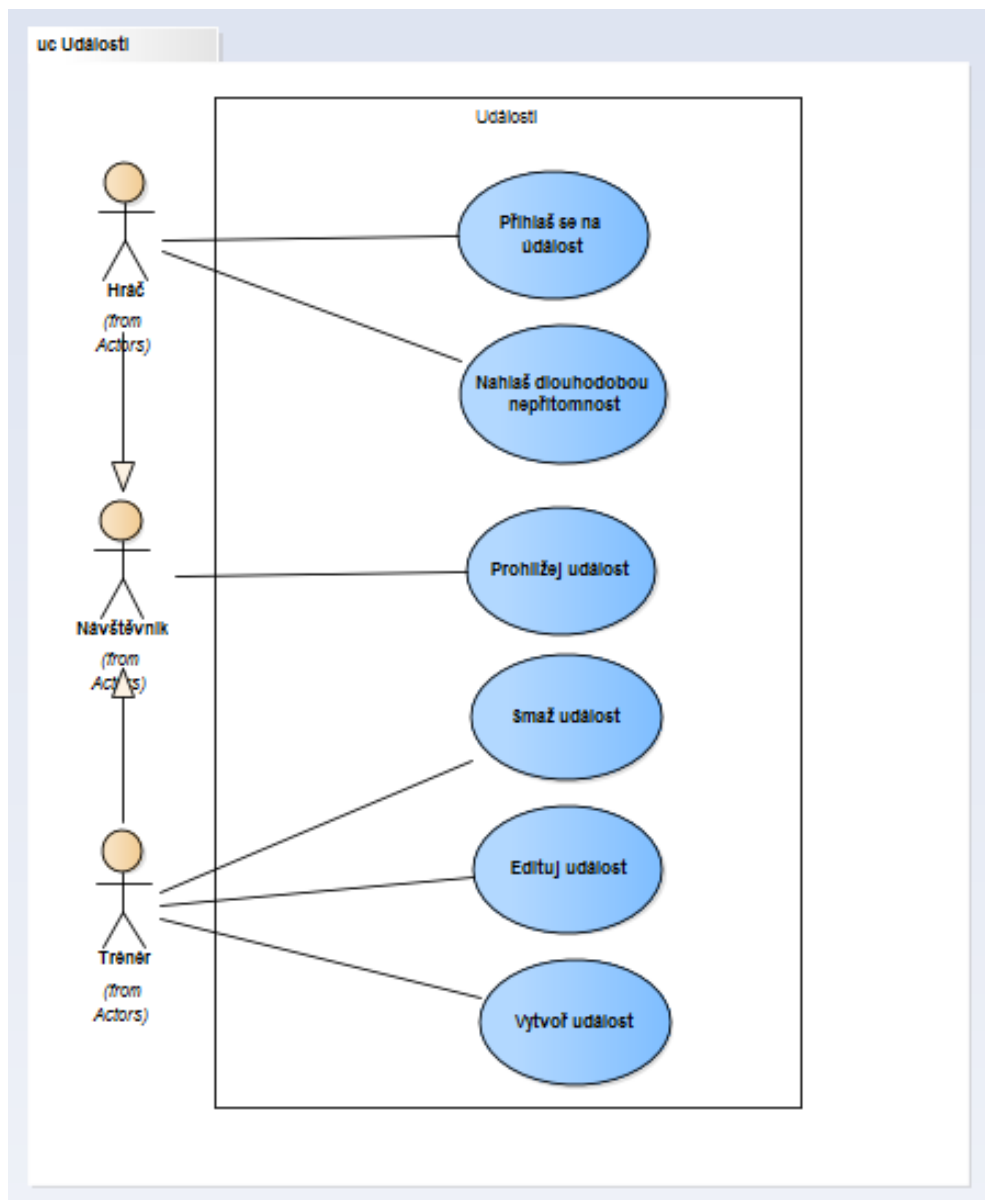
## **Přílohy**



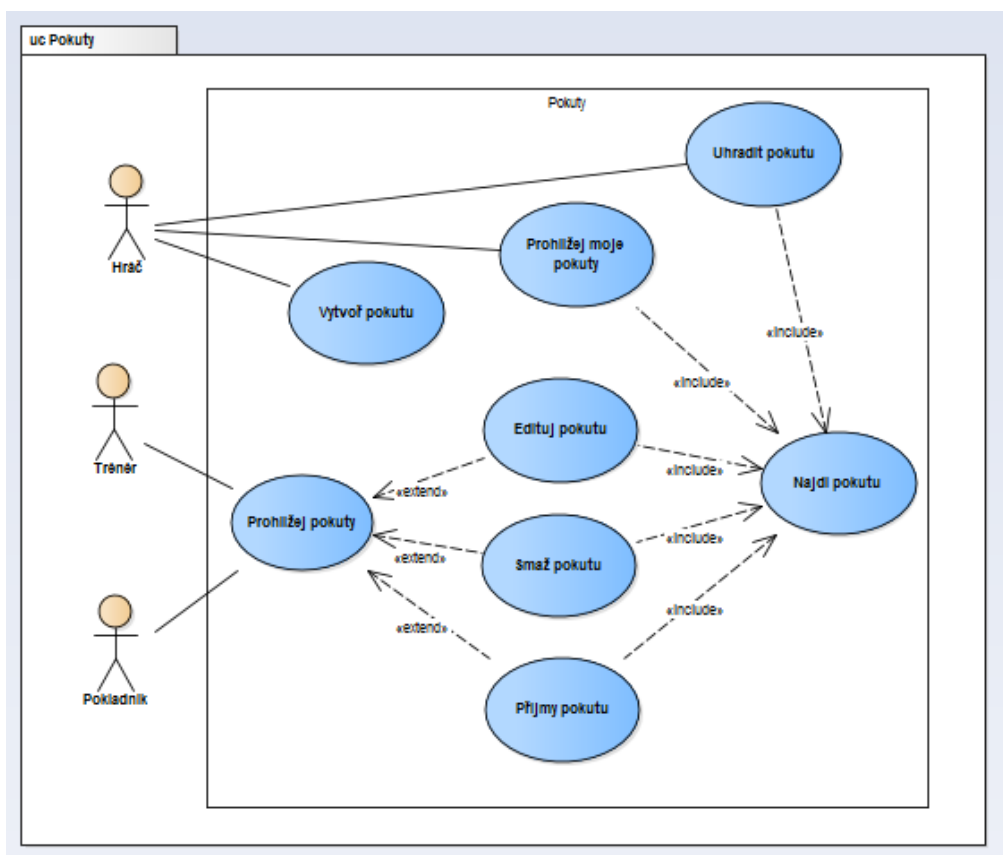
## A Diagramy případů užití



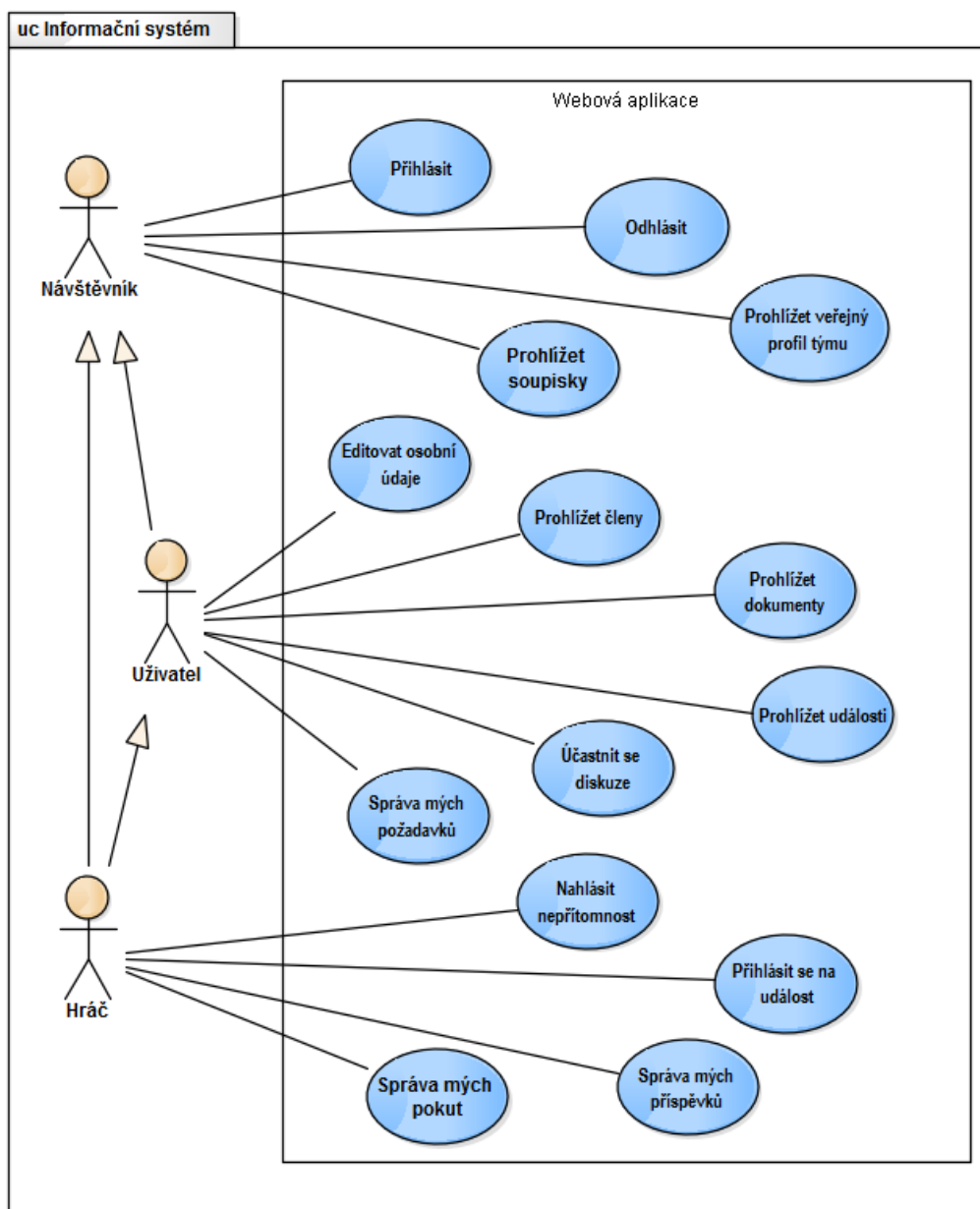
Obr. 26 Diagram případů užití - práce s uživateli a soupiskami



Obr. 27 Diagram případů užití - události



Obr. 28 Diagram případů užití - členské příspěvky a pokuty



Obr. 29 Diagram případů užití - Webová aplikace

## B Ukázky webové aplikace

Přihlášení do IS




# COMMIX TJ SOKOL BŘECLAV

FLORBALOVÝ KLUB

[NOVINKY](#)
[ČLÁNKY](#)
[O NÁS](#)
[TÝM](#)
[VÝSLEDKY A STATISTIKY](#)
[TRÉNINKY](#)  
[FOTOGALERIE](#)
[KONTAKTY](#)

### NOVINKY



**ÚŽASŇÁKOVI OVLÁDLI MEMORIÁL BR. JOSEFA JANKA**  
01.01.0001 0:00:00 | Vytvořil: **Administrátor**

Poslední neděli v **uplynulém** roce se v hale TJ Lokomotiva Břeclav sešli florbaloví nadšenci všech věkových kategorií, aby odehráli druhý ročník Memoriálu br. Josefa Janka. Nejlépe si v něm vedl tým... [ČTI DÁLE](#)



**STRÁNKY V PROVOZU !!**  
01.01.0001 0:00:00 | Vytvořil: **Administrátor**

Dnešním dnem byli úspěšně vytvořeny internetové stránky **Dnešním dnem byly úspěšně spuštěny stránky klubu. Nedostatky budou časem odstraněny**

... [ČTI DÁLE](#)

#### POŘADÍ - SOKOL

Tým	Body
Břeclav A	1
Břeclav I	0
Hovorany	0
Lanžhot	0
Velké Bílovice	0
Valtice	0
Lokomotiva	0

#### PŘÍŠTÍ ZÁPASY

07.07.2016 0:00:00  
**Hovorany - Lanžhot**

[ZOBRAZ VŠE](#)

#### POSLEDNÍ ZÁPASY

05.05.2015 0:00:00  
**(0 - 0)**  
**Commix Břeclav - Hovorany**

Obr. 30 Veřejná část aplikace - úvodní strana



**COMMIX TJ SOKOL BŘECLAV**  
FLORBALOVÝ KLUB

NOVINKY ČLÁNKY O NÁS TÝM VÝSLEDKY A STATISTIKY TRÉNINKY  
FOTOGALERIE FOTOGALERIE KONTAKTY ADMINISTRACE

### STATISTIKY

#### STATISTIKA UTKÁNÍ COMMIX BŘECLAV - HOVORANY

Domáci	05.05.2015 12:00	Hosté
<b>Commix Břeclav</b>		<b>Hovorany</b>
0	Hala: Lokomotiva Břeclav Začátek: 05.05.2015 12:00 Konec: 05.05.2015 12:00	0

#### PREHLED BRANEK

Commix Břeclav			Hovorany				
#	Třetina	Střelec	Asistence	#	Třetina	Střelec	Asistence

#### SOUPISKY

soupiska týmu Commix Břeclav		soupiska týmu Hovorany	
#	Hráč	#	Hráč

#### POŘADÍ - SOKOL

Tým	Body
Břeclav A	1
Břeclav I	0
Hovorany	0
Lanžhot	0
Velké Bilovice	0
Valdice	0
Lokomotiva	0

#### PŘÍŠTÍ ZÁPASY

7.7.2018 0:00:00  
Hovorany - Lanžhot

[ZOBRAZ VŠE](#)

Obr. 31 Veřejná část aplikace - detaily události

- Můj profil
- Členové klubu
- Týmy a soupisky
- Administrace - Soutěže
- Administrace - Sezóny
- Administrace - Haly
- Administrace - TypyPokut
- Dlouhodobé absence
- Tréninky a zápasy
- Cestovní příkazy
- Pokuty
- Členské příspěvky
- Dokumenty
- Pokladna
- Statistiky
- Diskuzní fórum

### Edituj člena

**Jméno**

**Příjmení**

**Nick**

**Email**

**Telefon**


**Datum narození**

**Ulice**

**Město**

**PSČ**

[Zpět na seznam](#)



**Administrace:**

**Pozice v týmu:**

**Aktivita uživatele:**

Obr. 32 Interní část aplikace - editace uživatele

- Můj profil
- Členové klubu
- Týmy a soupisky
- Administrace - Soutěže
- Administrace - Sezóny
- Administrace - Haly
- Administrace - TypyPokut
- Dlouhodobé absence
- Tréninky a zápasy
- Cestovní příkazy
- Pokuty
- Členské příspěvky
- Dokumenty
- Pokladna
- Statistiky
- Diskuzní fórum

### Docházka a informace o události

**Typ události** Trénink

**Hala** Hala - Valtice

**Začátek** 08-12-2015 02:18 odp.

**Konec** 08-12-2015 02:18 odp.

**Zveřejněna**

**Pozvánky na email**

**Popis**

Odešli znovu pozvánku

Trénink je určený pro týmy:

- Sokol - Břeclav A
- Sokol - Břeclav I
- CFBU - Commix

**Docházka na událost:**

Přihlásit uživatele:

---

**Prijdou:** Možná přijdou: Neprijdou:

Matulík Ondřej


Praška Pavel


Ficová Martina

Novák Pepa

Nevyjádřily se

**Seznam pokut za událost**

[<< Zpět na seznam](#)

Obr. 33 Interní část aplikace - editace události

- Můj profil
- Členové klubu
- Týmy a soupisky
- Administrace - Soutěže
- Administrace - Haly
- Administrace - TypyPokut
- Dlouhodobé absence
- Tréninky a zápasy
- Cestovní příkazy
- Pokuty
- Členské příspěvky
- Dokumenty
- Pokladna
- Statistiky
- Diskuzní fórum

### Prehled žádostí o cestovní náhrady

Vytvoř nový požadavek za cestovné

Typ události: Všechny události Hala: Všechny haly Schváleno: Vše Uhrazeno: Vše

Hledat podle příjmení:  Hledat

Datum:	Jméno:	Příjmení:	Liga:	Typ události:	Cíl cesty:	Částka:	Schváleno:	Stav:			
10-04-2015	<a href="#">Ondřej</a>	<a href="#">Matulík</a>	Sokol	Tumaj	Hala - Valtice	500	Schváleno	Nezaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>
10-04-2015	<a href="#">Pepa</a>	<a href="#">Novák</a>	Sokol	Tumaj	Hala - Valtice	500	Schváleno	Zaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>
10-04-2015	<a href="#">Pepa</a>	<a href="#">Novák</a>	Sokol	Tumaj	Hala - Valtice	500	Schváleno	Zaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>
10-04-2015	<a href="#">Pepa</a>	<a href="#">Novák</a>	Sokol	Tumaj	Hala - Valtice	500	Schváleno	Nezaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>
10-04-2015	<a href="#">Martina</a>	<a href="#">Ficová</a>	Sokol	Tumaj	Hala - Valtice	500	Schváleno	Zaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>
10-04-2015	<a href="#">Martina</a>	<a href="#">Ficová</a>	Sokol	Tumaj	Hala - Valtice	500	Neschváleno	Nezaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>
10-04-2015	<a href="#">Pepa</a>	<a href="#">Novák</a>	Sokol	Tumaj	Hala - Valtice	500	Neschváleno	Nezaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>
10-04-2015	<a href="#">Martina</a>	<a href="#">Ficová</a>	Sokol	Tumaj	Hala - Valtice	500	Neschváleno	Nezaplaceno	<span>Detaily</span>	<span>Upravit</span>	<span>Smazat</span>

Celkem žádostí: 8                      4 000,00 Kč

Obr. 34 Interní část aplikace - výpis žádostí o cestovní náhrady