



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Informa ní systém pro sportovní klub
Student:	Michal Jirka
Vedoucí:	Ing. Michal Šoch, Ph.D.
Studijní program:	Informatika
Studijní obor:	Informa ní systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Seznamte se s problematikou fungování sportovního klubu. Na základ získaných znalostí prove te analýzu požadavk pro navrhovaný informa ní systém. Vyberte vhodnou technologii pro realizaci internetového informa ního systému, který realizuje požadovanou funkcionalitu. Vyhodno te náklady na vývoj, nasazení a následný provoz. Navržený informa ní systém poté implementujte. Výslednou aplikaci ádn otestujte ve zvoleném sportovním klubu (po dohod s vedoucím práce) a prove te zhodnocení výsledk testování.

Aplikace musí obsahovat minimáln následující funkcionalitu:

- Uživatelské profily len klubu.
- Plánování a kalendá akcí.
- Plánování trénink s trenéry.
- Sdílení fotografií a dokument .

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 24. listopadu 2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Informační systém pro sportovní klub

Michal Jirka

Vedoucí práce: Ing. Michal Šoch, Ph.D.

13. května 2016

Poděkování

Rád bych poděkoval panu Ing. Michalovi Šochovi, Ph.D. za cenné rady a odborné vedení práce. Poděkování patří samozřejmě také mé rodině a přátelům za podporu po celou dobu studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 13. května 2016

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2016 Michal Jirka. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Jirka, Michal. *Informační systém pro sportovní klub*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Hlavní náplní práce je analyzovat a implementovat informační systém pro chod sportovního klubu, který by měl obsahovat některé nové užitečné funkce. V úvodní kapitole je definován samotný pojem informační systém, následuje seznámení s pojmem sportovní klub, základní informace o jeho vedení a managementu. Zároveň byl proveden výzkum formou dotazů na vedoucí sportovních klubů. Z následné analýzy vzešlo několik funkcí, které by uživatelé v tomto systému uvítali. Celý systém je postaven na PHP frameworku Nette, který je v teoretické části také představen, stejně jako další využití technologie. V závěru jsou popsány nedokonalosti systému, návrh na jejich další vylepšení a výsledky testování aplikace. V příloze se nachází zdrojové kódy implementace.

Klíčová slova Sportovní klub, řízení sportovního klubu, správa financí, sdílení souborů, komunikace s veřejností, Nette, MVC

Abstract

The main target of this bachelor thesis is to analyze and implement an information system of sport club which should content some new usefull functions. In the first chapter is the term information system defined. It is followed by definition of sport club, main informations about its direction and management. Concurrently a research was carried out by questionnaire on leadership of any sport clubs. Analysis of this research revealed functions, which users would appreciate. The system is built on PHP framework Nette. It is also described in teoretical part, as well as other used technologies. In the conclusion are written some imperfections of the system, suggestions for its improvements and results of testing. Moreover, source code is enclosed in the attachment.

Keywords Sport club, sport club management, finance administration ,file sharing, public communication, Nette, MVC

Obsah

Úvod	1
1 Cíl práce	3
2 Základní pojmy	5
2.1 Systém, informační systém	5
2.2 Sportovní klub	6
2.3 Sportovní management	7
3 Současná řešení	9
3.1 Tymuj.cz	9
3.2 Spoluhráči.cz	9
3.3 Teamstuff.com	9
3.4 Další závěrečné práce	10
4 Analýza	11
4.1 Požadavky	11
4.2 Možnosti řešení	13
4.3 Zvolené řešení	14
4.4 Technologie	14
4.5 Softwarová architektura	17
4.6 Ekonomicko-manažerský souhrn	17
5 Implementace	21
5.1 Adresář App	22
5.2 Adresář Vendor	23
5.3 Adresář WWW	23
5.4 Databáze	23
5.5 Princip fungování aplikace	24

6 Testování	29
6.1 Testování podle scénářů	29
6.2 Testy použitelnosti	30
6.3 Test GUI v prohlížečích	32
Závěr	35
Literatura	37
A Seznam použitých zkratk	41
B Obsah příloženého média	43

Seznam obrázků

5.1	Adresářová struktura Nette	21
5.2	Ukázka kódu třídy NewsManager.php	25
5.3	Životní cyklus presenteru	26
5.4	Ukázka kódu třídy GroupPresenter.php	27
5.5	Ukázka kódu šablony News\show.latte	28
5.6	Schéma databáze	28
6.1	Screenshot ze zařízení iPhone 5s	33

Seznam tabulek

4.1	Náklady na vývoj	19
4.2	Náklady na provoz	19

Úvod

Od počátku historie sportu se lidé sdružovali do různých kolektivů, ať už podle odvětví činnosti, věku, pohlaví nebo třeba výkonnosti. Vznik prvních oficiálních sportovních klubů je datován do druhé poloviny 19. století. Stejně jako každá jiná organizace má klub své vedení, hospodaření, správu financí a tak dále. Důležitým faktorem je vnitřní komunikace mezi vedením a hráči. S nástupem internetu se samozřejmě všechny procesy zjednodušily. Dnes téměř každý sportovní klub, od malých po ty největší u nás, má své webové stránky. Zde nalezneme většinou základní informace pro veřejnost. Avšak z mého osobního průzkumu vyplynulo, že řešení interních záležitostí stále probíhá pomocí uzavřených skupin na sociálních sítích, případně e-mailů. Ve sportovním prostředí se pohybují spoustu let a je pro mě součástí téměř každého všedního dne. Proto jsem se rozhodl v praktické části mé práce vytvořit informační systém, který by mohl být využíván univerzálně v klubech různého zaměření a zároveň byl finančně dostupný i pro menší oddíly. Domnívám se, že kvalitní systém podobného druhu v české verzi na internetu momentálně chybí. Zároveň bych do řešení rád zahrnul i některé konkrétní funkční požadavky, které jsem dostal od několika vedoucích sportovních klubů. Finální verzi informačního systému plánuji nasadit do provozu v oddílu, jehož jsem členem. V teoretické části se zaměřím na problematiku fungování sportovních klubů. Dále zhodnotím ekonomickou stránku věci, provedu SWOT analýzu a vyčísím náklady na vývoj a provoz. Součástí bude také analýza řešení a popis použitých technologií.

Cíl práce

Cílem této bakalářské práce je analýza, návrh a implementace informačního systému pro sportovní kluby. Provedl jsem průzkum mezi majiteli klubů a dalšími odborníky za účelem stanovení konkrétních funkčních požadavků, které se pokusím při práci splnit. Systém by měl být navržen jako webová aplikace s komplexním využitím, která bude obsahovat jak veřejnou, tak uživatelskou část. Při návrhu aplikace jsem si stanovil základní prvky funkcionality, které by měla aplikace obsahovat. Jedná se především o správu uživatelských profilů, systém událostí a tréninků propojený s osobním kalendářem, sdílení dokumentů a fotografií, správu financí a další. Základem pro celou aplikaci bude oblíbený PHP framework Nette, který je založen na třívrstvé architektuře MVC. Dalším cílem je vytvořit aplikaci, která bude využitelná na různých typech zařízení a ve všech nejpoužívanějších prohlížečích. Responzivního designu dosáhnu využitím sady nástrojů Bootstrap. Předpokládám, že aplikace by měla být nasazena do testovacího provozu ve vybraném sportovním klubu. Z hlediska managementu jsem stanovil cílový segment uživatelů. Tím by měly být především menší až středně velké kluby. Naprostá většina z nich totiž nepoužívá žádný jednotný informační systém. Cílem je zjednodušit jejich interní procesy, z nichž většina aktuálně probíhá neefektivními způsoby. Toto bude podrobněji popsáno v ekonomicko-manažerském pohledu na věc, stejně jako hrubé vyčíslení nákladů na vývoj a provoz aplikace. V souladu se stanoveným segmentem trhu by výsledná webová aplikace měla být cenově dostupná tak, aby její nasazení a provoz výrazně neovlivnil klubový rozpočet.

Základní pojmy

2.1 Systém, informační systém

Systém lze definovat jako množinu vzájemně propojených komponent, které musí vzájemně spolupracovat tak, aby byl naplněn daný cíl. Co se týká informačního systému, jedná se o velmi široký pojem, a proto stanovit jeho jedinou přesnou definici je obtížné. Jedna z mnoha definic říká, že *„informační systém je soubor lidí, technických prostředků a metod, zabezpečujících sběr, přenos, zpracování, uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení.“* [1]

V současné době si lze jen těžko představit chod jakékoliv větší firmy nebo organizace bez kvalitního informačního systému. Ten je základem pro efektivní řízení organizace. Jeho prostřednictvím probíhá většina důležitých interních procesů firmy, ale zároveň slouží pro podporu vztahů s veřejností, či zákazníky. Informační systémy lze rozdělit z hlediska různých kritérií. Níže uvádím dva základní typy dělení.

2.1.1 Rozdělení IS podle zaměření

- **Manažerské**
EIS (Executive IS) jsou systémy pro podporu řízení, vytvořené speciálně pro potřeby řídicích pracovníků.
- **Taktické**
DSS (Decision support system) jsou systémy pro podporu rozhodování na základě určitých vstupních informací a dat.
- **Vedení**
MIS (Management information system) jsou instalovány za účelem zkvalitnění vedení organizace.

- **Expertní**
KWS (Knowledge work system) slouží k získávání a uchování aktuálních odborných dat a znalostí.
- **Kancelářské**
OIS (Office IS) obsahuje nástroje pro podporu standartních kancelářských činností.
- **Operativní**
 - Transakční (TPS)
 - Péče o zákazníka (CRM)
 - Rezervační systémy (RIS)
 - Konstrukční (CAM)
 - Geografické systémy (GIS)
 - Výrobní systémy (MRP)
 - A další

2.1.2 Rozdělení IS podle rozsahu funkčnosti

- **All-in-one**
Pokrývá všechny klíčové procesy, nezaměřuje se tolik na detailní funkčnost.
- **Best-of-breed**
Orientuje se pouze na specifické procesy nebo obory. Vyznačuje se detailní funkčností a oborovou specializací.
- **Lite**
Odlehčené verze systémů využívané v menších a středních firmách.

[2]

2.2 Sportovní klub

Sportovní kluby jsou organizace založené sportovci za účelem sdílení společného zájmu, organizace tréninků a účasti na soutěžích. V naší zemi jsou sportovní kluby právníckými osobami, které fungují většinou jako občanská sdružení. Je však možné narazit i na kluby fungující jako obchodní společnosti. Základním dokumentem každého klubu jsou stanovy, které popisují organizační strukturu.[3, 4]

2.3 Sportovní management

S pojmem sportovní klub samozřejmě úzce souvisí i termín sportovní management. Definujeme ho jako „*způsob uceleného řízení tělovýchovných a sportovních svazů, spolků, klubů, tělovýchovných jednot a družstev, které alespoň z části akcentují podnikatelsky orientované chování.*“ Hlavním cílem sportovního managementu je efektivní využívání finančních částek, které ve světě sportu kolují. Předmětem zkoumání jsou obecné manažerské činnosti ve sportu, ale i činnosti specifické pro konkrétní oblast sportu. Od těchto činností jednotlivých manažerů se následně odvíjí konkrétní způsob řízení daného sportovního klubu.[5]

V dnešní době je sportovní prostředí velmi náročné na kvalitu řídicích funkcí, přesto v České republice stále funguje většina klubů na bázi dobrovolnictví svých členů. Stejně jako ve všech oblastech lidské činnosti, tak i sem stále více proniká vliv informačních technologií. Proto se domnívám, že kvalitně vytvořený informační systém by mohl zefektivnit vedení jakékoliv sportovní organizace.

Současná řešení

Před výběrem tématu jsem se dotázal několika odborníků ze sportovního prostředí. Ze zjištěných informací vyplývá, že v současnosti minimum klubů využívá jednotný informační systém. Interní komunikace většinou probíhá přes sociální sítě a e-mail. Prezentace pro veřejnost je většinou zajištěna pomocí jednoduchých statických webových stránek. Dostal jsem také konkrétní požadavky, které bych rád využil ve své implementaci.

3.1 Tymuj.cz

Nejužívanějším ryze českým systémem podobného druhu je v současné době portál tymuj.cz. Jedná se o internetovou službu pro lepší organizaci aktivit ve sportovních týmech a snadnou komunikaci jejich členů. Její využití je ovšem univerzální a může tedy sloužit i k organizaci jiných jednotek v oblasti sportu, pohybových aktivit a volného času. Podle dostupných informací využívá aktuálně službu kolem 130 000 uživatelů.[6]

3.2 Spoluhráči.cz

Jedná se o systém funkcí velmi podobný předchozímu. Vznikl v České republice v roce 2007. Kromě efektivní komunikace uvnitř klubu obsahuje zároveň funkce i pro širší komunitu sportujících lidí, jako je například hledání spoluhráče, sparingpartnera apod. Dle mého názoru však nemá tak přehledné uživatelské rozhraní, jako třeba Týmuj.cz.[7]

3.3 Teamstuff.com

Teamstuff.com je dalším nástrojem zaměřeným na správu týmů a klubů. Byl vyvinut skupinou sportovních nadšenců v Austrálii a přeložen do dvanácti světových jazyků, mezi nimiž nechybí ani čeština. Jeho základní myšlenkou je

odlehčit manažerům a trenérům od náročné organizační práce a poskytnout možnost soustředit se pouze na samotnou hru. Za velkou výhodu tohoto systému považují mobilní řešení v podobě volně dostupné aplikace. Díky ní lze spravovat svůj tým kdykoliv přímo z přenosného zařízení.[8]

3.4 Další závěrečné práce

Z dostupných zdrojů jsem zjistil, že v minulosti vzniklo několik závěrečných prací na podobné téma například na Vysoké škole ekonomické v Praze nebo Mendelově univerzitě v Brně. Práce obsahovaly analýzu a implementaci systémů podobného zaměření.[9, 10]

Analýza

4.1 Požadavky

SRS (Software requirements specification) je komplexní popis funkcí, které by měl vytvářený software obsahovat. V obecném případě lze říci, že se jedná o dokument, který vzniká vyjednáváním mezi zákazníkem a dodavatelem. Měl by být stručný, jasný a obsahovat vše důležité, na čem se obě strany dohodly. Obvykle je odsouhlasený a podepsaný jak zákazníkem, tak dodavatelem. V pozdější fázi vývoje na něj lze odkazovat v případě jakýchkoliv nejasností. Existuje několik obecně daných pravidel, podle kterých by měl být dokument SRS sepsán:

- **Correct**
Požadavky by měly být obsahově, formálně i gramaticky správně napsané.
- **Unambiguous**
Měly by být formulované přesně a jasně aby nedocházelo ke sporům.
- **Complete**
Jednotlivé požadavky by měly být definovány kompletně. Například by mělo být jasně definováno chování při určitých mezních hodnotách.
- **Consistent**
Není přípustné, aby se požadavky navzájem vylučovaly.
- **Ranked**
Je možné rozdělit požadavky podle priorit.

4.1.1 Funkční požadavky

- **Evidence uživatelů**
Systém by měl umět evidovat uživatele jakožto jednotlivé členy klubu.

Každý z nich bude mít unikátní přístupové údaje a veřejný osobní profil s možností vložení základních informací a fotografie. Samozřejmostí je možnost změny hesla.

- **Uživatelské role**

Uživatelé budou rozděleni do pěti různých rolí podle oprávnění přístupu k jednotlivým funkcím systému. Jedná se o následující role:

- Admin
- Předseda
- Sekretář
- Trenér
- Hráč

Každý uživatel bude mít přiděleno právě jednu z těchto rolí.

- **Skupiny uživatelů**

Kromě rolí poskytne systém možnost rozdělení uživatelů do skupin, které si vedení klubu samo vytvoří. To usnadní například rozesílání zpráv, či pozvánky na události.

- **Soukromé zprávy**

Systém by měl umožňovat odeslat soukromou zprávu jinému uživateli, případně několika uživatelům nebo skupině.

- **Správa financí**

Další z funkcí by měla zajistit vedení klubu přehled v oblasti financí. Jedná se o evidenci závazků a pohledávek a přehled o transakcích v klubové kase.

- **Správa událostí**

Systém bude podporovat tvorbu událostí a pozvánek na ně. Současně bude možné evidovat účasti a sledovat statistiky docházky jednotlivých členů klubu.

- **Správa tréninků**

Bude se jednat o podobný princip jako v případě událostí, ale speciálně modifikovaný pro tréninky.

- **Osobní kalendář**

Funkce, která by měla úzce souviset s událostmi a tréninky. Bude znamenávat tyto akce do předem připraveného kalendáře, což výrazně usnadní přehlednost.

- **Sdílení fotografií a dokumentů**

Systém by měl umět nahrávání a sdílení dokumentů tak, aby k nim měli přístup všichni uživatelé. Zároveň bude možné vytvářet a prohlížet fotogalerie.

4.1.2 Nefunkční požadavky

- Webová aplikace
- Podpora ve všech významných prohlížečích
- Responzivní design pro zobrazení na různých typech zařízení
- Grafické uživatelské rozhraní pomocí HTML, CSS, Bootstrap a dalších doplňkových knihoven.
- Framework PHP Nette
- Databáze MySQL
- Zabezpečení citlivých dat, zejména hesel k uživatelským účtům a údajů v databázi týkajících se financí

4.2 Možnosti řešení

Základním požadavkem v dnešní době je, aby informační systém fungoval jako webová aplikace. Většina sportovních klubů vlastní své domény a tím pádem nebude problém s nasazením do ostrého provozu. Výhodou webové aplikace oproti desktopové je intuitivnější ovládání a navíc odpadají starosti spojené s instalací, synchronizací dat a podobně. V současnosti je známo mnoho způsobů a technologií, kterými lze vyvíjet webové aplikace. Zde uvedu nejčastěji používané a následně zdůvodním svůj výběr.

4.2.1 Front-end

„Do front-end technologií patří všechny technologie, které nějakým způsobem zasahují do webové stránky na straně webového prohlížeče.“ [11]

Základem většiny webových aplikací je značkový jazyk HTML, aktuálně již ve verzi 5.0. Za zmínku ještě stojí technologie XHTML a XML. K HTML se téměř neodmyslitelně váže jazyk CSS. Jedná se o kaskádové styly, které určují samotný vzhled stránky. Dalším hojně využívaným nástrojem je skriptovací jazyk Javascript, který dodává webovým stránkám interaktivitu a dynamiku. K dispozici je také celá řada jeho rozšíření. Rád bych zde ještě zmínil jazyk Dart, který vyvinula v roce 2013 společnost Google s cílem konkurovat Javascriptu.

4.2.2 Back-end

„Jako back-end se označuje část webové aplikace, která slouží k administraci a ke zpracování dat.“ [12]

Na rozdíl od front-endu existuje obrovské množství technologií a způsobů řešení. Za všechny jmenuji trojici dle mého názoru nejoblíbenějších nástrojů – PHP, VB.NET, JAVA.

Další důležitou složkou každé webové aplikace je databáze. V tomto směru máme na výběr opět několik možností řešení. Nejpoužívanějšími jsou MySQL, MSSQL, ORACLE a PostgreSQL.

4.3 Zvolené řešení

Na základě svých předchozích zkušeností jsem se rozhodl využít osvědčené kombinace nástrojů a technologií.

Využití HTML je téměř samozřejmostí. Jedná se o všeobecně známý jazyk, který ve své verzi 5.0 přinesl další vylepšení. Aktuálně neexistuje mnoho webových stránek a aplikací, které by využívaly jiný jazyk pro základní konceptuální rozložení celé stránky.

S HTML se neodmyslitelně pojí CSS a Javascript, které také při implementaci využiju. Z mnoha javascriptových knihoven jsem si vybral oblíbené jQuery. Na základě doporučení se pokusím využít rovněž framework Bootstrap, který zajistí responzivní a moderní design.

Rozhodování o nástroji pro back-end bylo též jednoduché. V minulosti jsem měl možnost setkat se v praxi jak s PHP, tak s VB.NET. Pro svou přehlednost a jednoduchost volím jednoznačně jazyk PHP. Ten samozřejmě také nabízí plno frameworků. Mezi nejznámější patří Symfony, Zend, Lavarel a Nette. Z těchto jmenovaných jsem měl zkušenosti se Symfony a Nette. Rozhodl jsem se využít framework Nette, a to především z důvodu, že byl vyvinut v České republice. V neposlední řadě zmíním databázi. S jazykem PHP se obvykle pojí databáze MySQL a nejinak tomu bude i v mé práci. MySQL je navíc nejčastěji nabízenou databází webových hostingů.

Jednotlivé technologie ještě blíže popíšu v následující části své práce.

4.4 Technologie

V následující části detailněji popíšu všechny technologie, které jsem si vybral pro implementaci.

4.4.1 HTML

HTML (HyperText Markup Language) je značkovací jazyk, který je určen k tvorbě webových dokumentů, které obsahují hypertextové odkazy a pokročilejší formátování. HTML poskytuje elementy, které dokáží nastavit vzhled

a velikost textu, vkládat obrázky do dokumentů, vytvářet formuláře, či tabulky. Samotné HTML je určeno pouze pro dokumenty se statickým obsahem, tudíž nedokáže vytvořit dynamicky se měnící web a nepodporuje interakci s uživatelem. První verze HTML 1.0 spatřila světlo světa v roce 1991 a jejím autorem byl Tim Berners-Lee. Následně se v roce 1993 objevilo rozšíření známé jako HTML+, které obsahovalo 78 elementů. Mnohé z nich se dnes již v HTML nevyskytují. V následujících letech zaznamenával jazyk další a další vývoj.[13] Zatím poslední verzí je HTML 5.0. Ve finální verzi bylo představeno 28. října 2014. Uživatelům přineslo některé zcela nové elementy, pomocí kterých jsme schopni vytvářet webové stránky podle moderních trendů a požadavků.

4.4.2 CSS

Vznik a vývoj jazyka HTML postupně vedl k vytvoření samostatného nástroje pro grafickou úpravu webových dokumentů. V roce 1996 tak vznikly kaskádové styly neboli CSS (Cascading Style Sheet). Jejich první verze byla vytvořena přímo pro spolupráci s tehdejší verzí HTML. O dva roky později pak vyšlo CSS 2, které umožňuje interakci s dokumenty XHTML nebo také XML. Kaskádové styly používají svoji vlastní syntaxi, která se dá poměrně snadno a rychle naučit.[14]

4.4.3 Javascript, jQuery

Javascript je v současnosti jeden z nejrozšířenějších programovacích jazyků a jeho význam stále stoupá. Jedná se o dynamicky typovaný jazyk navržený pro skriptování v prostředí webového prohlížeče. Syntaxe Javascriptu vychází z jazyka C a základem je takzvaná prototypová dědičnost, která bývá velmi obtížná pro programátory, kteří jsou zvyklí na tradiční objektový model.[15] Jinými slovy, Javascript dodává dynamičnost webovým stránkám a zajišťuje interakci s uživatelem.

jQuery je knihovna Javascriptu, která usnadňuje vývojářům práci s Javascriptem. Efektivně využívá předdefinované metody odvozené od původních funkcí Javascriptu. jQuery nepřináší žádnou novou funkcionalitu, nýbrž výrazně zjednodušuje využití Javascriptu a rozšiřuje ho tak mezi širší spektrum programátorů.[16]

4.4.4 PHP

Zkratka PHP původně znamenala „Personal Home Page“, což byla technologie, pomocí které chtěl Rasmus Lerdorf v roce 1994 sledovat návštěvnost svých webových stránek. Postupným vývojem se pak ujal název „PHP: Hypertext Preprocessor“. Určitou zvláštností se může zdát fakt, že se nejedná o jazyk programovací, nýbrž skriptovací. Je navržen tak, aby vykonal určitou činnost v reakci na událost (odeslání formuláře, přechod na danou URL adresu apod.).

Jedná se o serverovou technologii nezávislou na platformě. Tím pádem je PHP využitelné téměř na většině operačních systémů. Mezi hlavní přednosti tohoto jazyka patří výkon, stabilita a přenositelnost skriptů mezi různými servery. Zároveň je syntaxí velmi jednoduchý, a proto je využitelný i pro začínající vývojáře.[17]

4.4.5 MySQL

Jedná se o nejoblíbenější veřejně dostupný databázový server. Byl vyvinut švédskou společností MySQL AB. Světově rozšířený je zejména díky tomu, že je nabízen poskytovateli webhostingů jako součást jejich služeb. Oblíbená je zejména kombinace MySQL s PHP.

4.4.6 Framework Nette

Jedná se o PHP framework stavěný tak, aby byl co nejpoužitelnější a nejvstřícnější. Má velmi srozumitelnou syntaxi a výrazně usnadňuje vývojářskou práci. Mezi hlavní přednosti Nette patří podpora moderních webových technologií jako například PHP 5 nebo AJAX, přehledný šablonovací systém a kvalitní ladící nástroje.[18]

Já sám za největší výhodu považuji českou dokumentaci a aktivní komunitu uživatelů. Díky tomu lze dohledat řešení většiny běžných problémů na českém fóru.

4.4.7 Bootstrap

Bootstrap je populární framework, který kombinuje HTML, CSS a Javascript. Primárně slouží k tvorbě responzivního webu, který lze správně zobrazit na zařízení různých rozměrů. Výrazně zjednodušuje a urychluje vytváření designu jakékoliv webové aplikace. Původně byl vytvořen roku 2010 společností Twitter. Následoval rychlý rozmach a z Bootstrapu se stal jeden z nejpoužívanějších front-end frameworků vůbec. Přesto je stále dostupný pod open-source licencí. Aktuálně je na světě verze 3 a ve vývoji verze 4.[19]

4.4.8 Font Awesome

Font Awesome je open-source knihovna využívající CSS. Slouží k vkládání obrázkových ikon do webu. Zdánilivě se jedná o malý doplněk, ale správným využitím dokáží ikony udělat celkový vzhled modernější a přehlednější, případně mohou usnadnit navigaci.[20]

4.4.9 DateInput

Formulářová komponenta, která slouží k výběru data a času. Je vytvořena jako komponenta pro Nette, pracuje s objektem DateTime a jQuery.[21]

4.4.10 DataTable

Jedná se o velmi užitečný jQuery plugin, který dodává obyčejným HTML tabulkám zcela nové možnosti. Podporuje mimo jiné stránkování, řazení nebo efektivní vyhledávání v rozsáhlých tabulkách.[22]

4.4.11 Fullcalendar

FullCalendar je další z řady jQuery pluginů. Jedná se o nástroj pro zobrazení kalendáře akcí, v mém případě událostí a tréninků. Podporuje zároveň i možnost propojení se známým kalendářem od společnosti Google.[23]

4.5 Softwarová architektura

Aktuálně existuje poměrně velké množství definic termínu softwarová architektura. Většina z nich je založena na principu popisu komponent a jejich chování. Za všechny zde uvádím jednu obecnou. „*Softwarová architektura je struktura komponent programu/systému, jejich vzájemné vazby, principy a předpisy určující jejich návrh a vývoj v průběhu času.*“[24] Jinak řečeno, architektura je obecnější pohled na systém, který zahrnuje hlavní komponenty, jejich chování a vzájemnou komunikaci. Určuje strukturu systému a definuje jeho chování. Důležitým poznatkem je, že každý systém má nějakou architekturu, ne vždy ale bývá zdokumentovaná.[25]

V mém informačním systému jsem se rozhodl využít architekturu, která je typická pro webové aplikace. Jedná se o třívrstvou architekturu MVP (Model-View-Presenter). Model je jádrem aplikace, komunikuje s databází a zajišťuje přístup k datům. Vrstva view prezentuje data uživatelům a presenter propojuje obě předchozí vrstvy, reaguje na události pocházející od uživatele a provádí změny v modelu nebo view.[26]

4.6 Ekonomicko-manažerský souhrn

4.6.1 SWOT analýza

SWOT analýza je základní nástroj managementu. Za vynálezce této metody je považován Američan Albert S. Humphrey, který působil v oblasti managementu především ve druhé polovině 20.století. Samotná zkratka odpovídá prvním písmenům anglických názvů jednotlivých částí této analýzy – strengths (silné stránky), weaknesses (slabé stránky), opportunities (příležitosti), threats (hrozby). Primárním cílem této metody je přimět manažera, autora daného projektu, podnikatele atd. aby se zamysleli právě nad těmito položkami a vyvodili z nich nějaké důsledky. Výhodou SWOT analýzy je, že se jedná o univerzální nástroj, který je využitelný od malých projektů po ty nejrozsáhlejší.[27]

- Strengths - silné stránky
 - Dobře provedená analýza
 - Implementace pomocí jednoduchých principů
 - Snadná modifikace pro potřeby konkrétního klubu
 - Technická podpora po celou dobu užívání
- Weaknesses - slabé stránky
 - Malý projektový tým
 - Nezkoušenost vývojáře s podobným projektem
 - Krátké testovací období
- Opportunities - příležitosti
 - Cenová dostupnost licence
 - Nízké provozní náklady
 - Minimum podobných systémů na trhu
- Threats - hrozby
 - Časové zpoždění dokončení projektu
 - Nedostatek financí na marketingovou propagaci
 - Nedostatek zájemců o koupi licence

Z uvedené SWOT analýzy vyplynul přehledný plán, který říká, čeho by měl projekt využít a na jaké oblasti si naopak dát pozor a snažit se je eliminovat. Tuto konkrétní analýzu zde uvádím jako příklad. U velkých projektů lze samozřejmě vytvořit rozsáhlou SWOT analýzu, která může výrazně ovlivnit úspěšnost celého projektu.

4.6.2 Náklady na vývoj

Náklady na vývoj jsem rozdělil do několika částí, které zároveň odpovídají rozdělení projektu do etap. Jedná se o čísla teoretická a odhadnutá, nikoliv náklady reálné. Pro výpočet jsem použil hodinovou sazbu 300 Kč, což odpovídá mzdě začínajícího programátora webových aplikací. Množství odpracovaných hodin jsem vyjádřil v jednotce mhr (man-hours). Výsledné hodnoty jsou přehledně zapsány do tabulky 4.1.

Důležitým a často uváděným ukazatelem trvání práce je také jednotka man-days. Její hodnotu lze získat vydělením hodnoty man-hours osmi a udává počet pracovních dní jedné osoby, který je potřeba na dokončení daného úseku práce. V tomto případě by tedy kompletní práce na projektu trvala 122 mhr, což je v přepočtu po zaokrouhlení 15 pracovních dní.

Tabulka 4.1: Náklady na vývoj

Etapa	Doba trvání	Náklady
Shromáždování požadavků	6 mhr	1800 Kč
Analýza požadavků a návrh	20 mhr	6000 Kč
Nastudování principů Nette Frameworku	8 mhr	2400 Kč
Implementace	70 mhr	21 000 Kč
Testování	15 mhr	4500 Kč
Nasazení do provozu	3 mhr	900 Kč
Celkově	122 mhr	36 600 Kč

4.6.3 Náklady na provoz

Jak jsem uvedl v předchozí SWOT analýze, náklady na provoz považuji za jednu z výhod celé aplikace. Systém je navržen tak, aby jeho provoz výrazně nezatížil rozpočet sportovního klubu. Ceny za webhosting a registraci domény jsou převzaty z hostingu Endora.cz, který osobně nejčastěji využívám.[28] Položky jsou blíže rozepsané v tabulce 4.2.

Tabulka 4.2: Náklady na provoz

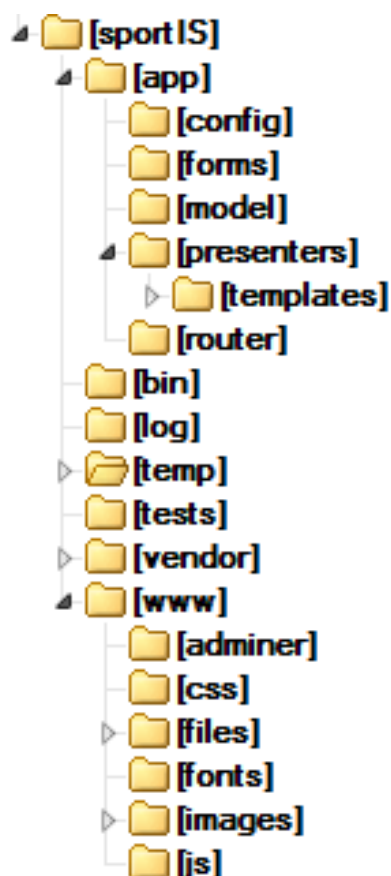
Položka	Cena
Cena licence	12 000 Kč
Náklady na údržbu	1000 Kč za 1 rok
Webhosting	232 Kč za 1 rok
Registrace domény	241 Kč za 1 rok
Celkové náklady	12 000 Kč jednorázově + 1473 Kč za 1 rok

4.6.4 Přínosy pro uživatele

Hlavním přínosem této aplikace pro potenciální uživatele je zefektivnění komunikace, předávání informací a zjednodušení interních procesů klubu. Zároveň systém podporuje komunikaci s veřejností, z čehož mohou plynout například výhody při získávání sponzorů. Systém by měl především ušetřit čas a eliminovat nedorozumění v komunikaci, která mohou vzniknout užíváním mailů, či sociálních sítí.

Implementace

V této kapitole popíšu adresářovou strukturu Nette Frameworku, označovanou také termínem sandbox. Po instalaci vznikne v kořenovém adresáři projektu následující základní struktura:



Obrázek 5.1: Adresářová struktura Nette

5.1 Adresář App

Dalo by se říci, že adresář app je mozkiem celé aplikace. Obsahuje veškeré části zodpovědné za řízení aplikační logiky. Stěžejním souborem je `bootstrap.php`. Ten zajišťuje načtení celé aplikace, definuje konfigurační soubory a spouští tzv. autoloading dalších součástí.

5.1.1 Config

Jak již název napovídá, tato složka v sobě ukrývá dva velice důležité konfigurační soubory. Jedná se o `config.neon` a `config.local.neon`. První z nich obsahuje globální nastavení aplikace a druhý parametry specifické pro aktuální vývojové nebo produkční prostředí. Zásadní informací je nastavení přístupu do databáze. Za zmínku ještě stojí zvláštní typ souborů s koncovkou `.neon`. Jedná se o typ specifický pro Nette a jeho konfiguraci.

5.1.2 Model

Adresář obsahuje třídy modelové vrstvy aplikace. Ty jsou zodpovědné za komunikaci aplikace s databází, získávání dat, manipulaci s nimi a ukládání. V mé implementaci složka Model obsahuje celkem 11 tříd. Z toho je jedna pomocná třída Model, sedm tříd pojmenovaných podle konkrétního presenteru, kterému dodává data, a koncovky Manager (například `NewsManager`, `FinanceManager` atd.) a třech tříd s koncovkou Storage, které jsou využity pro práci se soubory (`FileStorage`, `PictureStorage` a `GalleryStorage`).

Ukázka zdrojového kódu třídy `NewsManager` je na obrázku 5.2.

5.1.3 Presenters

V tomto adresáři se nachází jádro funkčnosti celé aplikace – presentery. Jedná se o vrstvu, která propojuje model a pohledy. Úloha jednotlivých presenterů spočívá v tom, že reagují na podněty od uživatele a na základě nich zajišťují změny buď v modelu, nebo v pohledu. Mohou obsahovat až sedm typů metod, které jsou volány postupně v předem definovaném pořadí. Tento postup je graficky znázorněn v obrázku 5.3 - Životní cyklus presenteru.[29] V mém projektu jsem nejčastěji využíval metody typu `action` a `render`. Zároveň jsem přímo v presenterech implementoval metody typu `createComponent`, které se starají o vytváření formulářů. Celkem mé řešení obsahuje 18 presenterů. Základem je `BasePresenter`, který je nadtřídou pro všechny zbývající, vyjma `ErrorPresenter`. Aplikace je postavená tak, že každý presenter reprezentuje jeden funkční celek. Například `AttendancePresenter` se stará o funkce spojené s docházkou, `FinancePresenter` zase o funkce správy financí a tak dále. Podrobněji je proces zpracování požadavků presentery popsán v sekci 5.5.

Ukázka kódu z třídy `GroupPresenter` je na obrázku 5.4.

5.1.4 Templates

Výše jsem popsal dvě z vrstev třívrstvé architektury. Nakonec jsem si nechal vrstvu View. Ta za pomoci šablon (pohledů) určuje, co konkrétně uvidí uživatel na své obrazovce. Je zodpovědná za převod dat zpracovaných předchozími dvěma vrstvami do vizuální podoby. K tomuto účelu využívá Nette kvalitní šablonovací systém zvaný Latte. Je to pomůcka, která díky množství různých maker a filtrů výrazně usnadňuje programátorovi propojení PHP s HTML. Moje implementace obsahuje celkem 52 pohledů, z nichž každý reprezentuje právě jednu unikátní stránku. To vše zastřešuje soubor `@layout.latte`, který obsahuje stejnou základní kostru každé z těchto stránek.

Ukázka z kódu souboru `News\show.latte` je na obrázku 5.5.

5.2 Adresář Vendor

Další z důležitých složek, která obsahuje knihovny využívané v aplikaci. Velké množství z nich je předdefinovanou součástí samotného Nette Frameworku. Složka rovněž slouží pro instalaci přídatných knihoven vytvořených početnou komunitou programátorů, využívajících tento framework.

5.3 Adresář WWW

Jedná se o adresář, který shromažďuje css styly, javascriptové soubory, obrázky, případně další veřejné soubory. Zároveň obsahuje velmi důležitý soubor `index.php` o jehož významu se dočtete dále.

5.4 Databáze

Pro práci s databází poskytuje Nette Framework tři základní třídy. Připojení k databázi zajišťuje `Nette\Database\Connection` a základní funkcionalitu `Nette\Database\Context`. Pro zjednodušení a optimalizaci výběru dat z databáze je k dispozici třída `Nette\Database\Table`.^[30] Jak již bylo napsáno dříve, pro implementaci mého projektu jsem si zvolil aktuálně nejpoužívanější databázi MySQL. Její největší výhodou je, že je nabízená téměř každým webhostingem. Při návrhu aplikace jsem si nejprve vytvořil kostru databázového modelu, který byl postupně doplňován a upravován podle potřeb v průběhu implementace. Jedním z nefunkčních požadavků bylo šifrování citlivých dat v databázi. Tímto způsobem jsou zabezpečena uživatelská hesla a údaje o financích.

Obrázek 5.6 byl vygenerován nástrojem Adminer, který jsem používal pro lokální vývoj. Znárodnuje databázové schéma a propojení jednotlivých tabulek cizími klíči.

5.5 Princip fungování aplikace

Zásadním souborem celé aplikace je `index.php` v adresáři `www`. Přes něj prochází veškeré požadavky od prohlížeče. Ačkoliv tento soubor je pro chod aplikace nezbytný, v jeho těle moc funkcí nenajdeme. Ve skutečnosti se zde totiž pouze předává řízení do jádra aplikace, konkrétně do zaváděcího souboru `bootstrap.php` ve složce `app`. Ten se postará o načtení celého frameworku, konfiguračních souborů, knihoven a dalších důležitých součástí.

Ve chvíli, kdy se daný požadavek dostane přes oba výše zmíněné soubory, stará se o něj dále objekt `Nette\Application`. Ten ve spolupráci s routerem zpracuje URL a zjistí, pro který presenter je požadavek určen a která akce se má vykonat.

Konkrétním příkladem v mé webové aplikaci může být zobrazení uživatelského profilu. Poté co uživatel klikne na dané jméno v seznamu členů klubu, aplikace si zjistí, že žádá o provedení akce `show` v presenteru `Profile` s parametrem například `id=25`. Nejprve je tedy vytvořen objekt třídy `ProfilePresenter`. Ten ve spolupráci s modelem, který je mu přiřazen (v tomto případě `ProfileManager.php`), získá data z databáze a pomocí metody `renderShow($id)` je předává do šablony. Zde je vidět, že slovo `Show` odpovídá názvu požadované akce a `$id` je parametr předávaný této funkci. Nakonec presenter přistoupí k vykreslení šablony, která má své pevné místo v adresářové struktuře. Nachází se v souboru `templates\Profile\show.latte`. Cesta k němu opět logicky odpovídá názvům presenteru a akce.[29]

```
class NewsManager extends Nette\Object
{
    /** @var Nette\Database\Context */
    private $database;

    public function __construct(Nette\Database\Context $database)
    {
        $this->database = $database;
    }

    public function getNews()
    {
        return $this->database->table('news');
    }

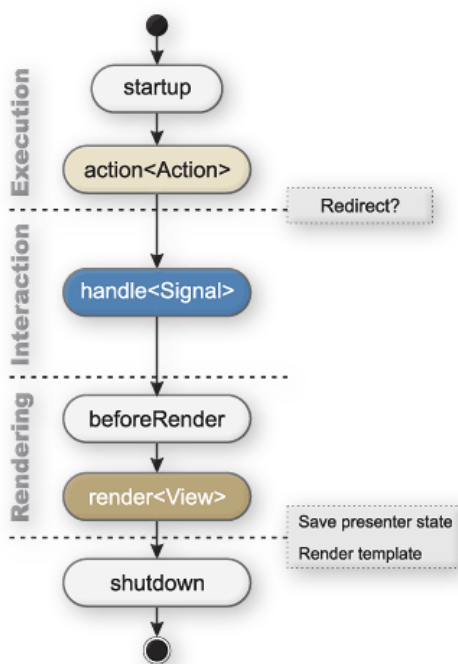
    public function getComments()
    {
        return $this->database->table('comments');
    }

    public function getComment($commentId)
    {
        return $this->database->table('comments')->where('id', $commentId)->fetch();
    }

    public function create($values)
    {
        return $this->database->table('news')->insert($values);
    }

    public function update($id, $values)
    {
        return $this->database->table('news')->get($id)->update($values);
    }
}
```

Obrázek 5.2: NewsManager.php



Obrázek 5.3: Životní cyklus presenteru

```
class GroupPresenter extends BasePresenter
{
    /** @var Nette\Database\Context */
    protected $database;

    public function __construct(Nette\Database\Context $database)
    {
        $this->database = $database;
    }

    public function renderDefault()
    {
        $this->template->groups = $this->database->table('groups');
        $this->template->database = $this->database;
        $this->template->userRole = $this->getUser()->getIdentity()->role;
    }

    protected function createComponentGroupForm()
    {
        $form = new UI\Form;

        $form->addText('name', 'Název: ')->setRequired("Vyplňte název skupiny");
        $form->addTextArea('description', 'Popis: ')
            ->addRule(UI\Form::FILLED, 'Vyplňte informace o skupině.');
```

```
        $form->addSubmit('send', 'Odeslat');
        $form->onSuccess[] = array($this, 'groupFormSucceeded');

        return $form;
    }
}
```

Obrázek 5.4: GroupPresenter.php

5. IMPLEMENTACE

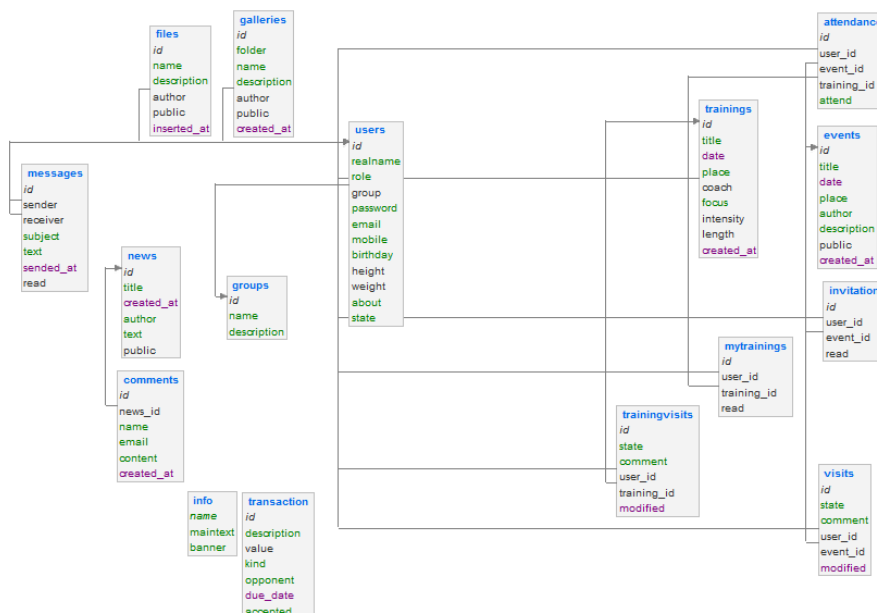
```

<div class="row">

  {var $cols=0}
  {if $news->author==$usrRealname}
  {var $cols=3}
  <div class="col-lg-3">
  <div class="box">
  <div class="caption">Možnosti</div>
  <div class="boxcontent">
  <div class="list-group" style="margin-bottom: 0px;">
  <a n:href="edit $news->id" class="list-group-item">
  <h4 class="list-group-item-heading">
  <i class="fa fa-pencil"></i> Upravit
  </h4>
  </a>
  </div>
  <div class="list-group" style="margin-bottom: 0px;">
  <a n:href="delete $news->id" class="list-group-item">
  <h4 class="list-group-item-heading">
  <i class="fa fa-trash-o"></i> Smazat
  </h4>
  </a>
  </div>
  </div>
  </div>
  </div>
  {/if}

```

Obrázek 5.5: News\show.latte



Obrázek 5.6: Schéma databáze

Testování

6.1 Testování podle scénářů

Jednou z nejčastějších testovacích metod je testování dle scénářů. Ačkoliv je tento způsob efektivní především pro větší projekty, rozhodl jsem se ho zařadit i do mé práce. Záměrně jsem stanovil několik kritických scénářů, které by mohly způsobit neočekávané chování aplikace.

6.1.1 Špatné přihlašovací údaje

Při zadání chybných přihlašovacích údajů se systém správně vrátí do základního stavu. Ovšem testováním bylo zjištěno, že chybí notifikace o nepovedeném přihlášení. Tento problém je způsoben využitím modálního okna Bootstrapu pro přihlašovací formulář. Chyba byla odstraněna a notifikace je nyní funkční.

6.1.2 Rozdílná hesla při registraci nového uživatele

Formulář pro přidání nového uživatele do systému obsahuje dvě pole pro zadání hesla. Zadané heslo je potřeba potvrdit zadáním stejného hesla v následujícím políčku. Aplikace na základě validace formuláře správně vyhodnotí zadání jako chybné, registraci neprovede a upozorní na chybu červeným zvýrazněním formulářového pole.

6.1.3 Smazání skupiny uživatelů

Jednou z možností správy skupin uživatelů je samozřejmě odstranění skupiny. Problém však může nastat, pokud se uživatel pokusí odstranit skupinu, která obsahuje nějaké členy. Testováním byla zjištěna správná funkčnost, kdy jsou všichni uživatelé z odstraněné skupiny automaticky přesunuti do skupiny Nezařazeno. Ta je v systému nastavená implicitně a nelze ji smazat.

6.1.4 Hromadné odesílání zpráv

Dalším testovacím scénářem jsem zvolil možnost, kdy uživatel při psaní soukromé zprávy využije možnost odeslání celé skupině uživatelů, ale zároveň s tím vybere i jednotlivého uživatele z dané skupiny. Testováním bylo zjištěno, že taková zpráva je pak odeslána dvakrát. Tato chyba byla následně odstraněna jednoduchou úpravou v implementaci a při následném otestování proběhl scénář bez problému.

6.1.5 Vložení souboru s již existujícím jménem

Jedním ze základních funkčních požadavků bylo sdílení souborů. Jednotlivé dokumenty jsou vkládány do stejné složky. Problém tedy může nastat, pokud se uživatel pokusí vložit soubor stejného názvu, jaký už v adresáři existuje. Test ukázal, že tento případ je v implementaci vyřešen. Systém zabrání vložení souboru a v notifikačním řádku upozorní uživatele na duplikaci jmen.

6.2 Testy použitelnosti

Test použitelnosti (usability) je metoda, která ve spolupráci se skupinou nezávislých uživatelů dokáže odhalit a následně vyřešit problémy, které samotný vývojář aplikace většinou nepostřehne. Spočívá v tom, že uživatelům jsou zadány úkoly, které mají na webu provést a následně jsou sledovány jejich postupy, reakce a způsoby dosažení zadaného cíle. Pro testování své aplikace jsem oslovil tři lidi různých věkových skupin a zaměření.

6.2.1 Průběh

Prvním testovaným uživatelem byl 45letý muž, majitel sportovního klubu s mírně podprůměrnou znalostí práce s počítačem a internetem. Byly mu poskytnuty přihlašovací údaje pro účet v roli předsedy a následující seznam úkolů:

- Přihlášení
- Vytvoření nového uživatele s libovolnými osobními údaji, rolí hráč a skupinou Nezařazeno
- Vytvoření tréninku o délce trvání 45 minut se zaměřením na kondici, na který bude pozván pouze hráč vytvořený v předchozím bodě
- Zadat údaje o docházce na událost, která se konala 30.dubna 2016 a všichni pozvaní se zúčastnili

Druhým testovaným byl 17letý juniorský hráč, člen sportovního klubu se středně pokročilou znalostí IT a webových technologií. Byly mu poskytnuty přihlašovací údaje pro účet v roli hráče a následující úkoly:

- Přihlášení
- Změna údajů v hráčském profilu a změna profilové fotografie
- Změna přístupového hesla k účtu
- Sdílení libovolného souboru tak, aby byl neveřejný

Třetím respondentem se stala 20letá studentka vysoké školy zaměřené na ekonomiku bez zkušeností ze sportovního prostředí, která disponuje uživatelskými znalostmi na pokročilejší úrovni. Byly jí přiděleny následující úkoly na uživatelském profilu sekretáře klubu:

- Přihlášení
- Zaznamenat na účet klubu příjem částky 2500 Kč jako cenu za umístění v soutěži
- Vytvořit údaje o závazku o hodnotě 1800 Kč, který má klub vůči městskému úřadu za pronájem sportoviště. Splatnost závazku má být 31.května 2016
- Akceptovat příjem financí za pohledávku, kterou měl klub vůči sponzorské firmě v hodnotě 1000 Kč

Nejstarší věkovou kategorii v tomto testování zastupovala 85letá žena, která má nižší uživatelské znalosti. Do testování byla zařazena, aby bylo zjištěno, zda jsou základní funkce aplikace přístupné i takto specifické skupině uživatelů. Jejím úkolem bylo zorientovat se sekci pro nepřihlášené uživatele, okomentovat nejnovější veřejný článek a zjistit datum konání nejbližší veřejné akce.

6.2.2 Výsledek

Během testování jsem pozorně sledoval chování jednotlivých respondentů v rámci aplikace. Ukázalo se, že uživatelé s alespoň základní znalostí práce s internetem a webem nemají problém se rychle zorientovat a řešit zadané problémy efektivně v relativně krátkém čase. Pro uživatele s menšími zkušenostmi s webovými aplikacemi může být zpočátku obtížnější zorientovat se v položkách hlavní nabídky a zjistit, kde se která sekce nachází. Konkrétně mi bylo navrženo přejmenovat záložku Admin, ale nakonec jsem se rozhodl na návrh nepřistoupit.

S potěšením musím konstatovat, že tento typ testů neodhalil žádné závažné nedostatky a všichni respondenti zvládli své úkoly bez větších problémů.

6.3 Test GUI v prohlížečích

Mezi nefunkčními požadavky aplikace najdeme mimo jiné responzivní design a podporu všech významných prohlížečů. Rozdílné vykreslování v různých prohlížečích a typech zařízení je věčným problémem webových programátorů. V dnešní době existují desítky prohlížečů, které používají odlišná renderovací jádra.

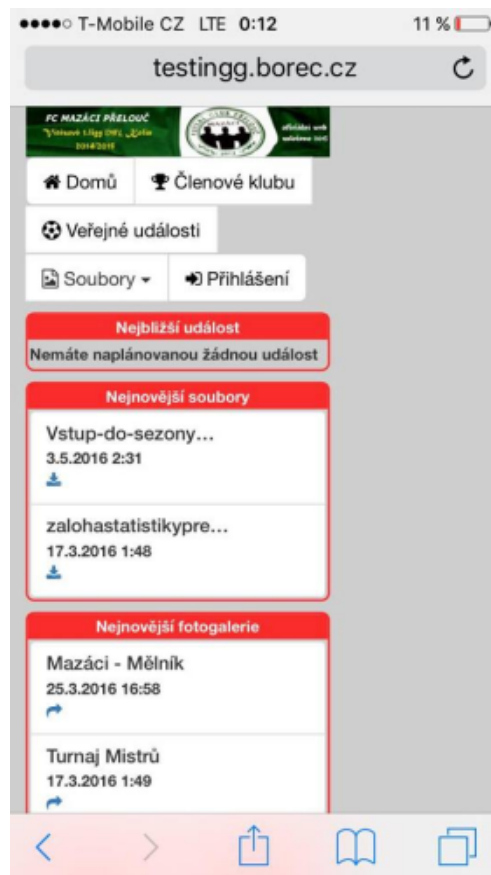
Moje aplikace byla testována na těchto počítačových prohlížečích:

- Mozilla Firefox 46.0.1
- Google Chrome 50.0.2661.94
- Internet Explorer 11
- Safari 9.1 na Mac OS

Následuje seznam testovaných prohlížečů na mobilních zařízeních:

- Chrome 47.0
- Aplikace Internet na Androidu
- Opera Mini 11.0
- Safari na iPhone 5s

Určité problémy během testování jsem zaznamenal u prohlížeče Opera Mini 11.0. Aplikace byla použitelná, ale nechovala se správně a interaktivně. To mohlo být způsobeno například slabší podporou Javascriptu v tomto prohlížeči. V ostatních zmíněných prohlížečích jsem neregistroval žádné vážnější chyby. Na obrázku 6.1 je screenshot ze zařízení iPhone 5s.



Obrázek 6.1: Screenshot ze zařízení iPhone 5s

Závěr

Cílem této práce bylo analyzovat, navrhnout a implementovat informační systém pro sportovní kluby ve formě webové aplikace. Součástí analýzy mělo být také ekonomicko-manažerské shrnutí. Implementace měla zahrnovat funkční požadavky získané během shromažďování poznatků od odborníků.

Nejprve jsem v samostatné kapitole poukázal na současný stav řešení, který se nakonec oproti původním předpokladům ukázal být poměrně rozsáhlý.

Na základě získaných poznatků byly v následné analýze definovány konkrétní funkční i nefunkční požadavky na vznikající systém. Poté byly shrnuty možnosti řešení, které současný svět webových technologií nabízí. Finální výběr konkrétních nástrojů byl vysvětlen a odůvodněn a jednotlivé technologie byly detailně popsány. Ekonomicko-manažerský souhrn pak poukázal na náklady spojené s vývojem a nasazením systému. Stanovený cíl finanční nenáročnosti i pro menší organizace byl splněn.

Cílem implementace bylo vytvořit systém založený na českém PHP frameworku Nette a jeho třívrstvé architektuře. Podle plánu byl také využit systém Bootstrap, který zajistil kompaktnější design. Podařilo se splnit všechny požadavky, které byly konkrétně definovány v analýze 4.1. Vzhledem k nezkoušenosti s Nette jsem cíle dosáhl především využitím jednoduchých implementačních postupů. Během řešení úlohy jsem postupně zjišťoval, že tento framework nabízí daleko více možností, které jsem při práci nevyužil.

Nakonec byla aplikace nasazena v klubu, jehož jsem členem a prošla třemi způsoby testování. Konkrétně se jednalo o test podle kritických scénářů, test použitelnosti a test GUI. Výsledky odhalily určité nedostatky, které byly ve většině případů opraveny. To vše je shrnuto v samostatné kapitole.

Informační systémy jsou aplikace, jejichž potenciál je opravdu velmi široký. Domnívám se, že by bylo možné navrhnout a provést velké množství dalších vylepšení a rozšíření. Přesto bych zde chtěl jmenovat dvě hlavní, kde vidím určité nedostatky. Z funkčního hlediska se jedná především o propojení událostí a tréninků s Google kalendářem, který je v současnosti velmi oblíbený. Z hlediska grafiky a uživatelského rozhraní bych viděl rezervu ve vy-

užití Javascriptu. Jeho větším zapojením do implementace by byla aplikace dynamičtější a uživatelsky přívětivější.

V průběhu řešení této bakalářské práce jsem zjistil, že svět webových technologií je v současné době plný možností. Obohatil jsem své znalosti o základní využití Nette Frameworku a zároveň odhalil jeho další možnosti, které mohu využít v dalších projektech. V neposlední řadě jsem si poprvé vyzkoušel práci na rozsáhlejšímu projektu, který vyžadoval náročnější rozvržení času. Věřím, že tato práce byla pro mě přínosem.

Literatura

- [1] Molnár, Z.: *Podnikové informační systémy*. Praha: Vydavatelství ČVUT, 2004, ISBN 80-01-03079-2, 13 s.
- [2] Náplava, P.: Typy informačních systémů [online]. [cit. 2016-04-14]. Dostupné z: https://edux.fit.cvut.cz/courses/BI-TIS/_media/lectures/06/prednaska06.pdf
- [3] ČUS: Jak funguje sportovní klub? [online]. [cit. 2016-04-14]. Dostupné z: <http://www.cuscz.cz/sluzby-servis/jak-funguje-sportovni-klub.html>
- [4] Kubíček, J.: Sport a právo [online]. 2012, [cit. 2016-04-14]. Dostupné z: <http://www.jirikubicek.cz/data/dokumenty/sportovni-pravo-aktualni.zip>
- [5] Čáslavová, E.: *Management a marketing sportu*. Praha: Olympia, 2009, ISBN 978-80-7376-150-9, 18-19 s.
- [6] Týmuj: O službě Týmuj.cz [online]. [cit. 2016-04-14]. Dostupné z: <http://www.tymuj.cz/o-nas.html>
- [7] Spoluhráči.cz: Spoluhráči.cz – vaše oddílová klubovna a klubové stránky zdarma [online]. [cit. 2016-04-26]. Dostupné z: <http://www.spoluhraci.cz/>
- [8] Teamstuff: Free Sports Team Management Software [online]. [cit. 2016-04-26]. Dostupné z: <http://teamstuff.com/>
- [9] Krug, J.: Informační systém pro sportovní kluby [online]. [cit. 2016-04-26]. Dostupné z: http://www.vse.cz/vskp/35996_informacni_system_pro_sportovni_kluby

- [10] Kočí, J.: Informační systém sportovního klubu [online]. [cit. 2016-04-26]. Dostupné z: http://is.mendelu.cz/zp/portal_zp.pl?prehled=vyhledavani;podrobnosti=37201;zp=31323;download_prace=1
- [11] Bittner, J.: 2. díl – Výběr technologií 1. inetwork.cz [online]. [cit. 2016-04-14]. Dostupné z: <http://www.itnetwork.cz/html-css/webove-aplikace/technologie-pro-vyvoj-webovych-aplikaci-vyber-technologie-1/>
- [12] Štráfelda, J.: Co je Backend. Adaptic [online]. [cit. 2016-04-14]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/backend/>
- [13] Písek, S.: *HTML: začínáme programovat*. 3., aktualiz. vyd. [i.e.] 1. vyd. Praha: Grada, 2010, ISBN 978-80-247-3117-9, 15-17 s.
- [14] Grusová, L.: *CSS pro úplné začátečníky: od běžných k vysoce efektivním a atraktivním WWW stránkám*. Vyd. 1. Brno: Computer Press, 2003, ISBN 80-7226-680-2, 8 s.
- [15] Žára, O.: *JavaScript: programátorské techniky a webové technologie*. 1. vydání. Brno: Computer Press, 2015, ISBN 978-80-251-4573-9, 11-13 s.
- [16] Rutter, J.: *Smashing jQuery*. Hoboken, N.J.: Wiley, 2011, ISBN 978-0-470-97723-1, 5 s.
- [17] Ullman, L. E.: *PHP a MySQL: názorný průvodce tvorbou dynamických WWW stránek*. Vyd. 1. Brno: Computer Press, 2004, ISBN 80-251-0063-4, 12-16 s.
- [18] Nette Foundation: Seznámení s Nette Frameworkem [online]. [cit. 2016-04-12]. Dostupné z: <http://doc.nette.org/cs/2.3/getting-started>
- [19] Otto, M.; Thornton, J.: Bootstrap [online]. [cit. 2016-04-12]. Dostupné z: <http://getbootstrap.com>
- [20] Gandy, D.: Font Awesome, the iconic font and CSS Toolkit [online]. [cit. 2016-04-12]. Dostupné z: <http://fontawesome.io>
- [21] Vodáček, O.: DateInput [online]. [cit. 2016-05-10]. Dostupné z: <http://addons.nette.org/voda/date-input>
- [22] Datatables: Table plug-in for jQuery [online]. [cit. 2016-04-12]. Dostupné z: <http://datatables.net/>
- [23] Shaw, A.: FullCalendar – Javascript Event Calendar (jQuery plugin)[online]. [cit. 2016-04-12]. Dostupné z: <http://fullcalendar.io/>

-
- [24] Smolík, T.: Softwarová architektura: Nežúžený „klasický“ úvod [online]. [cit. 2016-04-13]. Dostupné z: http://www.profinet.eu/wp-content/uploads/2016/03/Profinit_SwArchitectureOverview.pdf
- [25] Čejka, T.: Taktiky pro ladění kvality softwarových architektur [online]. 2011, [cit. 2016-04-13]. Brno: Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Barbora Bühnová. Dostupné z: http://is.muni.cz/th/140490/fi_m/
- [26] Nette Foundation: Model–View–Presenter (MVP) [online]. [cit. 2016-04-12]. Dostupné z: <http://doc.nette.org/cs/0.9/model-view-presenter>
- [27] BusinessVize.cz: Kde se vzala a k čemu všemu je vlastně SWOT analýza [online]. [cit. 2016-04-27]. Dostupné z: <http://www.businessvize.cz/planovani/kde-se-vzala-a-k-cemu-vsemu-je-vlastne-swot-analyza>
- [28] Endora.cz: Freehosting, webhosting - neomezeně domén, multihosting [online]. [cit. 2016-04-27]. Dostupné z: <http://www.endora.cz>
- [29] Nette Foundation: MVC aplikace a presentery [online]. [cit. 2016-05-01]. Dostupné z: <http://doc.nette.org/cs/2.3/presenters>
- [30] Nette Foundation: Databáze [online]. [cit. 2016-05-01]. Dostupné z: <http://doc.nette.org/cs/2.3/database>

Seznam použitých zkratk

IS Informační systém

GUI Grafické uživatelské rozhraní

MVC Architektura Model-View-Controller

XML Rozšiřitelný značkovací jazyk

XHTML Kombinace XML a HTML

URL Jednoznačné zapsání umístění souboru na internetu

Obsah přiloženého média

readme.txt.....	stručný popis obsahu CD
src	
_ impl.....	zdrojové kódy implementace
_ create.sql.....	skript pro vytvoření databáze
_ BP_Jirka_Michal_2016.tex..	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
_ assignment.pdf	zadání práce ve formátu PDF
_ BP_Jirka_Michal_2016.pdf	text práce ve formátu PDF