

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra Počítačů

## IS pro správu letních táborů

**Jindřich Regál**

Vedoucí: Ing. Božena Mannová, Ph.D.

Obor: Softwarové technologie a management

Zaměření: Softwarové inženýrství

Prosinec 2017



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Regál Jindřich

Studijní program: Softwarové technologie a management  
Obor: Softwarové inženýrství

Název tématu: IS pro správu letních táborů

Pokyny pro vypracování:

Seznamte se s problematikou informačních systémů a s konkrétním systémem, používaným v klubu pro správu a organizaci letních táborů. Na základě získaných znalostí navrhnete nový informační systém pro přihlašování na tábory. Tento systém by měl snížit zátěž dobrovolníků v klubu, měl by zvýšit informovanost koordinátorů a usnadnit komunikaci a sdílení informací. Systém by měl být maximálně uživatelsky přívětivý. Proveďte specifikaci požadavku na systém a na základě jejich analýzy navrhnete systém nový. Navržený informační systém implementujte ve vhodně zvoleném jazyce. Použijte i vhodné aplikace Googlu. Výslednou aplikaci dokumentujte včetně uživatelské příručky a otestujte včetně testů použitelnosti.

Seznam odborné literatury:

- 1) Ian Somerville: Software Engineering, Global Editio, Pearson Higher Ed, 2016
- 2) Ondřej Žára, JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015

Vedoucí: Ing. Božena Mannová, Ph.D.

Platnost zadání do konce zimního semestru 2018/2019

prof. Dr. Michal Pěchouček, MSc.  
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 13.7.2017



## Poděkování

Děkuji vedoucí své bakalářské práce Ing. Boženě Mannové, Ph.D. za cenné rady, připomínky a odborné vedení. Dále bych chtěl poděkovat dobrovolníkům a vedení z klubu SHM Praha - Počernice za čas strávený konzultacemi a testováním aplikace.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 22. prosince 2017

## Abstrakt

Tato práce se zabývá návrhem a implementací informačního systému pro správu letních táborů a podporu chodu neziskové organizace. Funkce a požadavky byly určeny dle potřeb organizace SHM Klub Praha - Počernice, z. s. Byla zpracována analýza procesů a požadavků klubu. Na základě analýzy byl vytvořen návrh struktury aplikace, případů užití a databázového modelu. Dle návrhu byla provedena implementace v jazyce PHP.

**Klíčová slova:** informační systém, PHP, Nette Framework, Javascript, jQuery, Google Drive API

## Abstract

This thesis deals with design and implementation of information management system for summer camps and a non-profit organization. Required features were determined by the needs of SHM Klub Praha - Pocernice, z. s. Process and requirements analysis was made. Based on analysis, application structure, use cases and database model were designed. The implementation was made using PHP programming language.

**Keywords:** Information system, PHP, Nette Framework, Javascript, jQuery, Google Drive API

**Title translation:** IS for camp registration

## Obsah

|   |           |  |  |
|---|-----------|--|--|
| <b>1 Úvod</b>                                     | <b>1</b>  |  |  |
| <b>2 Současný stav a požadavky</b>                | <b>3</b>  |  |  |
| 2.1 Popis současného stavu                        | 3         |  |  |
| 2.1.1 Evidence členů                              | 3         |  |  |
| 2.1.2 Registrace účastníků na akce                | 3         |  |  |
| 2.1.3 Vedení kroniky                              | 3         |  |  |
| 2.1.4 Evidence a půjčování majetku                | 4         |  |  |
| 2.1.5 Informace o akcích                          | 4         |  |  |
| 2.2 Požadavky klubu na nový systém                | 4         |  |  |
| 2.3 Omezení                                       | 4         |  |  |
| 2.3.1 Omezení z uživatelského pohledu             | 5         |  |  |
| 2.3.2 Technická omezení a nároky                  | 5         |  |  |
| <b>3 Analýza</b>                                  | <b>7</b>  |  |  |
| 3.1 SWOT  | 7         |  |  |
| 3.2 Požadavky                                     | 8         |  |  |
| 3.2.1 Funkční požadavky                           | 8         |  |  |
| 3.2.2 ICT požadavky                               | 9         |  |  |
| 3.2.3 Podpora                                     | 9         |  |  |
| 3.3 Identifikace rizik                            | 10        |  |  |
| 3.3.1 Rizika obchodního dopadu                    | 10        |  |  |
| 3.3.2 Riziko v souvislosti s charakterem aplikace | 10        |  |  |
| 3.3.3 Procesní rizika                             | 10        |  |  |
| 3.3.4 Technologická rizika                        | 10        |  |  |
| 3.3.5 Rizika komponent                            | 10        |  |  |
| 3.4 Tabulka rizik                                 | 10        |  |  |
| 3.5 Alternativní řešení                           | 11        |  |  |
| <b>4 Návrh</b>                                    | <b>13</b> |  |  |
| 4.1 Uživatelská část                              | 13        |  |  |
| 4.2 Administrátorská část                         | 14        |  |  |
| 4.3 Role  | 15        |  |  |
| 4.4 Případy užití                                 | 16        |  |  |
| 4.4.1 Scénáře případů užití                       | 16        |  |  |
| 4.5 Model   | 18        |  |  |
| 4.5.1 Oprávnění, komunikace a export              | 18        |  |  |
| 4.5.2 Události, formuláře a přihlášky             | 20        |  |  |
| 4.5.3 Platby a správa členů                       | 22        |  |  |
| <b>5 Implementace</b>                             | <b>23</b> |  |  |
| 5.1 Volba technologie                             | 23        |  |  |
| 5.1.1 Programovací jazyk PHP                      | 23        |  |  |
| 5.1.2 Databáze MySQL                              | 24        |  |  |
| 5.1.3 Client-side technologie                     | 24        |  |  |
| 5.1.4 Vývojové prostředí                          | 24        |  |  |
| 5.2 Nette framework                               | 25        |  |  |

|  |           |  |           |
|--|-----------|--|-----------|
| 5.2.1 Latte šablony . . . . .                      | 25        | 7.1 Budoucnost projektu . . . . .                        | 45        |
| 5.2.2 Nette database . . . . .                     | 25        | <b>A Použité zkratky</b>                                 | <b>47</b> |
| 5.3 MVC . . . . .                                  | 26        | <b>B Literatura</b>                                      | <b>49</b> |
| 5.3.1 Model . . . . .                              | 26        | <b>C Kompletní schéma databáze</b>                       | <b>53</b> |
| 5.3.2 View . . . . .                               | 26        | <b>D Uživatelská příručka</b>                            | <b>55</b> |
| 5.3.3 Controller . . . . .                         | 26        | D.1 Softwarové požadavky . . . . .                       | 55        |
| 5.4 Struktura aplikace . . . . .                   | 27        | D.2 Instalace aplikace . . . . .                         | 55        |
| 5.5 Klíčové komponenty . . . . .                   | 28        | D.3 Uživatelská část . . . . .                           | 56        |
| 5.5.1 Entity a oprávnění . . . . .                 | 28        | D.4 Administrátorská část . . . . .                      | 58        |
| 5.5.2 Tabulkový výpis . . . . .                    | 30        | <b>E Vyjádření předsedy klubu Mgr.<br/>Pavla Heriana</b> | <b>61</b> |
| 5.5.3 Univerzální formulář . . . . .               | 30        |  |           |
| 5.6 Implementační zajímavosti . . . . .            | 32        |  |           |
| 5.6.1 Editor přihlašovacích<br>formulářů . . . . . | 32        |  |           |
| 5.6.2 E-maily a šablonovací systém                 | 34        |  |           |
| <b>6 Testování</b>                                 | <b>37</b> |  |           |
| 6.1 Nette tester . . . . .                         | 37        |  |           |
| 6.2 Testy použitelnosti . . . . .                  | 38        |  |           |
| 6.3 Bezpečnost . . . . .                           | 39        |  |           |
| 6.4 Akceptační test . . . . .                      | 39        |  |           |
| 6.4.1 Klientská část . . . . .                     | 39        |  |           |
| 6.4.2 Administrátorská část . . . . .              | 41        |  |           |
| <b>7 Závěr</b>                                     | <b>45</b> |  |           |



## Obrázky

|   |    |  |    |
|---|----|--|----|
| 3.1 Swot analýza . . . . .  | 7  | D.1 Výchozí rozcestník uživatelské části . . . . .             | 56 |
| 4.1 Vizualizace uživatelských rolí . . .  | 15 | D.2 Postup přihlášení na akci s registrací uživatele . . . . . | 57 |
| 4.2 Diagram případů užití . . . . .   | 17 | D.3 Výchozí obrazovka administrace                             | 58 |
| 4.3 Databázový model 1. část . . . . .  | 19 | D.4 Obvyklý vzhled tabulkového výpisu v administraci . . . . . | 59 |
| 4.4 Databázový model 2. část . . . . .  | 21 |  |    |
| 4.5 Databázový model 3. část . . . . .  | 22 |  |    |
| 5.1 Tabulkový výpis <code>OperationsDataGrid</code> v IS . . . . .                    | 30 |  |    |
| 5.2 Výsledný formulář vygenerovaný s použitím metody <code>addInputs</code> . . . . . | 31 |  |    |
| 5.3 Původní návrh modelu . . . . .  | 32 |  |    |
| 5.4 Zobrazení dat přímo v databázi pomocí <code>PhpMyAdmin</code> . . . . .           | 33 |  |    |
| 5.5 Ukázka formulářového editoru . . . . .  | 33 |  |    |
| 5.6 Vložení zástupné proměnné v editaci šablony pro členství . . . . .                | 34 |  |    |
| 5.7 Zmařený pokus o smazání přihlášek přes šablonu . . . . .                          | 35 |  |    |
| 5.8 Příklad použití šablony . . . . .   | 35 |  |    |
| 5.9 Vygenerovaný soubor . . . . .   | 35 |  |    |
| 5.10 Zdrojové přihlášky . . . . .   | 35 |  |    |
| 6.1 Přidané propojovací tlačítko . . . . .  | 38 |  |    |
| 6.2 Přidání funkce párování dle závěru z testování . . . . .                          | 38 |  |    |

## Tabulky

|   |    |
|---|----|
| 3.1 Tabulka rizik .....                                       | 11 |
| 5.1 Statistika podpory prog. jazyků<br>webhostingy v ČR ..... | 23 |



# Kapitola 1

## Úvod

Od roku 1991 působí v České Republice nezisková organizace Salesiánské hnutí mládeže, z. s.[1], jejíž cílem je nabízet dětem a mládeži kvalitní využití volného času a pomoci jim v osobním rozvoji[2]. Organizace je rozdělena do 33 klubů rozprostřených po celém území České republiky a její provoz je zajišťován téměř výhradně dobrovolníky, kteří se o chod starají ve svém volném čase. To je i případ SHM Klub Praha - Počernice (dále jen SHM Počernice). Důsledkem chodu na dobrovolnické bázi je absence jasně stanovených procesů a pravidel při přípravě jednotlivých akcí. Každý dobrovolník musí v současné době hledat své postupy přípravy, komunikace s účastníky atd. To zhoršuje přehled vedení klubu o všech aktivitách a je tedy třeba vynakládat větší úsilí při dohledu nad splněním všech zákonných povinností, které jsou na klub kladeny.

Cílem tohoto projektu je zanalyzovat nejčastější procesy v klubu a na základě analýzy navrhnout informační systém, který nejčastějším úkonům nastaví jasná pravidla a umožní automatizaci alespoň části nezbytných kroků. Při řešení je kladen maximální důraz na snížení administrativní zátěže dobrovolníků. K dosažení tohoto cíle byly potřeby klubu konzultovány s předsedou klubu Mgr. Pavlem Herianem a s několika dobrovolníky, kteří v klubu působí.

Motivací pro vypracování informačního systému je požadavek klubu na modernizaci jejich přihlašovacího systému, osobní zkušenost autora s prací pro klub, snaha o ušetření administrativní práce dobrovolníkům a seznámení s možnostmi využití služeb Google ve vlastní aplikaci.



## Kapitola 2

### Současný stav a požadavky

#### 2.1 Popis současného stavu

Klubu SHM Počernice jakožto zájmovému spolku čerpajícího dotace MŠMT<sup>1</sup>, MHMP<sup>2</sup> vznikají zákonné povinnosti, které musí plnit. Rovněž je podřízen pokynům celorepublikové zastřešující organizace. V této kapitole je stručně popsán postup jejich plnění před nasazením nového informačního systému.

##### 2.1.1 Evidence členů

V současné době je realizována formou Google Tabulky, kde jsou členové zapsáni, data jsou ve špatném stavu (různé formátování adres, dat narození, . . .), data nejsou chráněna proti náhodnému zásahu, k informacím o členech má přístup jen část vedení, statistiky o členech jsou vytvářeny ručně, platby členských příspěvků jsou kontrolovány a urgovány ručně.

##### 2.1.2 Registrace účastníků na akce

Je využíván starý systém, který umožňuje pouze užití jednoho neměnného formuláře pro všechny akce, což znemožňuje univerzálnější použití. Není možné měnit přihlášky bez zásahu správce, není možná obnova zapomenutého hesla k administraci ani další základní funkce nutné pro správu.

##### 2.1.3 Vedení kroniky

Kronika je spravována jednou osobou, která aktivně zjišťuje, jaké akce proběhly a následně od jednotlivých koordinátorů ručně vyžaduje zaslání zápisu do kroniky. Není k dispozici rychlý přehled o chybějících zápisech, stav kroniky je veden v tabulce,

<sup>1</sup>Ministerstvo školství, mládeže a tělovýchovy České republiky.

<sup>2</sup>Magistrát hlavního města Prahy

přičemž neexistuje rychlé provázání se zápisem v kronice a není jasné, zda stav kroniky v kontrolní tabulce odpovídá skutečnému stavu.

#### ■ 2.1.4 Evidence a půjčování majetku

Byla sepsána tabulka majetku, není ale vedeno místo uložení (klub nedisponuje skladem a věci k zapůjčení jsou na více místech) ani termíny, kdy je majetek volný k zapůjčení. Není jasný postup při zapůjčení, vše je řešeno ad hoc.

#### ■ 2.1.5 Informace o akcích

Střídavě vznikají a zanikají Google Kalendáře s přehledem akcí, ty jsou pak ručně přepisovány na web. Podle kalendáře pak správce kroniky hlídá a kontaktuje jednotlivé koordinátory.

### ■ 2.2 Požadavky klubu na nový systém

Business požadavky klubu na nový systém jsou výsledkem jednání s vedením klubu SHM Počernice:

1. Vytváření a úprava přihlášek na akci.
2. Přehled a správa akcí klubu.
3. Možnost změnit formu přihlašování dle akce.
4. Export dat pro další zpracování.
5. Správa členské základny.
6. Snadné informování členské základny o dění v klubu.
7. Automatické generování nezbytných dokumentů.
8. Sledování a zadávání zápisů do kroniky.
9. Možnost nastavení práv k zobrazení informacím dle uživatele pro ochranu osobních údajů.
10. Automatické párování plateb u přihlášek a u členských příspěvků.
11. Sdílení užitečných informací mezi koordinátory.

### ■ 2.3 Omezení

Klub SHM Počernice vznikl na základě aktivit společenství, které fungovalo již před založením klubu. Existence klubu by tedy měla lidem přinášet hlavně benefity. Na některé úkony jsou lidé zvyklí tak, jak jsou (byť z analytického pohledu fungují

chaoticky a neřízeně). Dobrovolníci, členové a účastníci akcí nejsou zaměstnanci klubu a je tedy třeba přistupovat k zavádění nových systémů velice citlivě a s ohledem na současné fungování. Z toho vyplývají omezení pro nový systém.

### ■ 2.3.1 Omezení z uživatelského pohledu

1. Zavedení systému nesmí nijak komplikovat přihlášení a placení jak táborů, tak členských příspěvků.
2. Pro členy a účastníky musí být navrhované řešení intuitivní.

### ■ 2.3.2 Technická omezení a nároky

1. Pro minimalizaci provozních nákladů musí být řešení bez zvláštních technických požadavků oproti běžným webovým aplikacím např. typu CMS (Wordpress atp.).
2. Očekávané zatížení je do 1 000 registrovaných uživatelů, současný přístup maximálně desítky uživatelů.
3. IT řešení musí fungovat bez nutnosti instalace jako webová aplikace pro minimalizaci požadavků na technickou zdatnost uživatelů a maximální kompatibilitu se zařízeními.
4. Systém musí minimalizovat počet úkonů pro aktualizaci dat o akcích na oficiálním webu (napojení do systému není možné, web provozován centrální klubovou organizací v rámci aplikace třetí strany).
5. Maximální možná integrace se službami Google Kalendář a Google Disk.





# Kapitola 3

## Analýza

### 3.1 SWOT

SWOT analýza umožňuje identifikovat, na co se při dalším návrhu zaměřit a čeho je potřeba se vyvarovat[3].

|               | Silné stránky   | Slabé stránky  |
|---------------|---|--|
| Vnitřní původ | <ul style="list-style-type: none"><li>▪ Ochota dobrovolníků starat se o administrativu v klubu.</li><li>▪ Dlouholetá zkušenost v oblasti organizací volnočasových aktivit.</li></ul>  | <ul style="list-style-type: none"><li>▪ Fungování klubu závislé na konkrétních osobách.</li><li>▪ Špatná komunikace mezi koordinátory (sdílení zkušeností, ...).</li><li>▪ Opakované řešení stejných problému dokola.</li></ul>  |
|               | Příležitosti  | Hrozby   |
| Vnější původ  | <ul style="list-style-type: none"><li>▪ Noví účastníci ze satelitních oblastí s nedostatkem volnočasových aktivit (Újezd nad Lesy, Zeleneč, ...).</li><li>▪ Rozšiřování aktivit díky dotacím MČ Praha 20, MHMP a MŠMT</li></ul> | <ul style="list-style-type: none"><li>▪ Pokles účastníků z důvodu špatné informovanosti.</li><li>▪ Krytí termínů s dalšími akcemi ostatních organizací v oblasti.</li><li>▪ Ztráta motivace dobrovolníků z důvodu špatné informovanosti a nutnosti administrativy.</li><li>▪ Malá část dotací použitelná na zajištění provozu klubu.</li></ul> |

Obrázek 3.1: Swot analýza

## 3.2 Požadavky

Ze zadání a analýzy současného stavu vyplývá řada požadavků na nový systém. Ty jsou pro přehlednost rozděleny do 3 kategorií.

### 3.2.1 Funkční požadavky

Funkční požadavky popisují služby, které by měl systém poskytovat. Dále definují reakci a chování systému na konkrétní situace, které mohou nastat[4]. Pro potřeby realizovaného IS byly stanoveny tyto funkční požadavky:

- Akce
  - Evidence pořádaných akcí.
  - Nastavení oprávnění k akcím.
  - Přihlašování uzpůsobeno dle typu akce.
  - Třídění akcí dle typu a času.
  - Editace a zobrazení stavu přihlášek pro účastníky.
  - Kalendář akcí pro koordinátory.
  - Informace o akcích dalších organizací v oblasti.
- Kronika
  - Evidence kroniky.
  - Editace zápisů.
  - Rychlý přehled chybějících zápisů.
  - Automatické připomenutí doplnění kroniky.
- Členská základna
  - Evidence členské základny.
  - Editace detailu členství a přehled plateb pro členy a ustanoveného správce členské základny.
  - Hromadné kontaktování členů s možností filtrování dle věku, lokalit, nebo stavu členství.
- Správa majetku
  - Evidence majetku klubu.
  - Elektronické žádosti o zapůjčení pro aktivní členy klubu.
  - Kalendář obsazenosti majetku k zapůjčení.
- Zpracování dat a export
  - Export dat z přihlášek pro další zpracování.

- Automatické generování povinných dokumentů.
- Možnost úpravy šablon pro automatické dokumenty.
- Automatické statistiky pro vedení klubu.
- Možnost propojení akcí s kalendáři standardem iCalendar nebo jiným.
- Platby
  - Automatické zpracování plateb z bú klubu u ČSOB.
  - Zadávání manuálních plateb a korekce párování.
- Uživatelé a komunikace
  - Definování rolí v klubu a oprávnění přístupu k datům.
  - Sdílení důležitých dokumentů pro koordinátory.
  - Snadné rozeslání informací koordinátorům.

### ■ 3.2.2 ICT požadavky

- Běh všech IT řešení v rámci webového prohlížeče (s podporou MS Edge, Google Chrome v. 35+, Mozilla Firefox v. 30+, Safari v. 8+ v rámci systémů Windows 7 a 10, macOS 10.5 a vyšší).
- V mobilních zařízeních se systémem Android a OS X musí být zachována funkcionality klientské části.
- Dostupnost 99,9 % (vychází ze statistik současných poskytovatelů webhostingu).
- Řešení postaveno na freeware technologiích.
- Počítat se zatížením v řádu max. tisíce registrovaných uživatelů.

### ■ 3.2.3 Podpora

- Zajištění běhu systému minimálně do roku 2022.
- Poskytování technické podpory pro účastníky a členy klubu.
- Řádné školení koordinátorů a dalších dobrovolníků.
- Seznámení kompetentních dobrovolníků v klubu o systému v maximálním rozsahu pro zajištění udržitelnosti do budoucna.

## 3.3 Identifikace rizik

Identifikace rizik pomáhá eliminovat interní vady a hrozby poškození prostředí vlivem systémového selhání[4]. Tato sekce definuje nebezpečné události, které mohou nastat.

### 3.3.1 Rizika obchodního dopadu

Aplikace vzniká jako nekomerční projekt a vychází z reálných potřeb klubu. Vzhledem k tomu, že realizátoři se zřekli své finanční odměny za realizaci, nehrozí žádná penalizace za pozdní dodání.

### 3.3.2 Riziko v souvislosti s charakterem aplikace

Vzhledem k tomu, že motivace vzniku aplikace pochází od dobrovolných spolupracovníků sdružení, bude problematické udržet formální správnost postupu výroby aplikace, kvalitu zadání, zejména jasnou specifikaci skutečných potřeb.

### 3.3.3 Procesní rizika

Protože projekt řídí stejná osoba, která zároveň projekt realizuje, je zde riziko, že v rámci návrhu nebudou některé případy užití navrženy dostatečně optimálně a intuitivně. Je tedy třeba v průběhu vývoje práci průběžně konzultovat s koncovými uživateli aplikace (jednotliví organizátoři akcí).

### 3.3.4 Technologická rizika

Jedním z cílů aplikace je usnadnění a sjednocení způsobu organizace klubu. To je navázáno na další aplikace třetích stran a bude tedy nutné pracovat s mnoha API (banka, internetová galerie, cloudové úložiště). Zde může dojít k problémům způsobeným nedostatečným pokrytím funkcionalit. Práce bude probíhat na ověřeném softwaru a hardwaru. Nepředpokládají se žádná nestandardní řešení.

### 3.3.5 Rizika komponent

Počítá se, že bude nutné v průběhu upravovat původní návrh jak pohledů, tak i klíčových funkcionalit. Harmonogram byl vytvořen s ohledem na možnost vzniku drobných změn a byla ponechána rezerva na dokončení.

## 3.4 Tabulka rizik

Tabulka uvádí nejzávažnější identifikovaná rizika realizace a nasazení IS. Navrhuje opatření pro zmírnění pravděpodobnosti výskytu, popř. následků v případě nastalého problému. Pravděpodobnost vychází z realizace předchozích aplikací podobného charakteru.

| Riziko   | Kategorie             | Pravděpodobnost | Dopad      | Opatření  |
|--|-----------------------|-----------------|------------|---|
| Problémy s implementací metamodelu pro registrační formuláře | Vývoj / technologické | 30%             | kritický   | Kvalitní návrh databáze a pohledů, které podstatně zjednoduší implementaci                            |
| Problémy s nasazením/užíváním                                | Organizační           | 20%             | kritický   | Dostatečná prezentace a školení ke správnému používání, důraz na jednoduché UI při vývoji             |
| Špatné pokrytí potřeb organizátorů/vedení                    | procesní              | 15%             | kritický   | Pravidelné konzultování s klubem, důraz na rozšiřitelnost při implementaci                            |
| Chybějící API pro propojení s jinými aplikace                | technologické         | 30%             | marginální | Drobná úprava procesů ve fungování klubu/ kvalitní zpracování importu/exportu dat v aplikaci Orgament |
| Špatný časový odhad  | vývoj                 | 20%             | marginální | Ponechání časové rezervy (dokončení), odložení nasazení   |
| Nedůvěra v aplikaci  | vnější                | 10%             | marginální | Komunikovat zabezpečení uživatelům poskytujícím osobní informace, registrace na úoou, ssl komunikace  |

Tabulka 3.1: Tabulka rizik

## 3.5 Alternativní řešení

Administrativu klubu by šlo řešit i v již hotových podnikových informačních systémech, nicméně tyto nástroje bývají často velice robustní a pro fungování klubu zbytečně rozsáhlé. Výběr, nasazení a úpravy jiného systému by sebou nesly také mnoho rizik a úskalí. Lze také předpokládat, že vzhledem k univerzálnosti komerčních řešení by bylo třeba věnovat více času zaškolení dotčených osob.



# Kapitola 4

## Návrh

Z požadavků vyplývá, že k aplikaci budou přistupovat dva základní okruhy uživatelů. Prvním okruhem jsou aktivní dobrovolníci a koordinátoři, kteří budou využívat poměrně velkou část funkcionality aplikace dle svých kompetencí (dále jen administrátoři). Druhou skupinou jsou „zákazníci“ klubu, tedy zpravidla účastníci akcí, v případě dětí jejich zákonní zástupci (dále jen uživatelé). Z tohoto důvodu bude aplikace rozdělena na dvě části.

### 4.1 Uživatelská část

Uživatelé jsou často rodiče dětí, kteří nemusí být zblhlí v používání online aplikací. Proto je třeba se v této části aplikace zaměřit na jednoduché UI<sup>1</sup>. Uživatelská část bude obsahovat pouze jednoduchý rozcestník na akce, které je třeba aplikací obsloužit. Jsou to tyto úkony:

- Podat přihlášku
- Seznam přihlášek
- Přihlásit se do klubu
- Spravovat členství

Pro přehlednost bude rozcestník doplněn ikonami symbolizujícími jednotlivé úkony. Pro editaci přihlášek a správu členství je nutné uživatele identifikovat a vytvářet jim uživatelské účty. Z důvodu jednoduchosti použití nebude aplikace vyžadovat poskytování více informací, než je e-mail a heslo. Uživatel bude naváděn v jednotlivých krocích, aby nutnost vytvoření registrace před provedením požadovaného úkonu téměř nepostřehl a neobtěžovala ho.

<sup>1</sup>User Interface - uživatelské rozhraní; způsob, jakým lidé ovlivňují chování aplikace

## 4.2 Administrátorská část

Administrátorská část bude sloužit klubu pro podporu jejich činnosti a správu většiny dat souvisejících s klubem. Z tohoto důvodu je třeba dbát důraz na možnost podrobného nastavení oprávnění k jednotlivým sekcím, aby dobrovolníci neměli přístup k datům, která mají být dostupná pouze užšímu vedení. Zároveň to usnadní orientaci v aplikaci, protože pro méně zainteresované osoby se část funkcionality kompletně skryje.

Vzhledem k rozsahu funkcí musí být administrátorská část strukturována do několika sekcí. Pro navigaci mezi sekcemi bude administrace obsahovat menu po levé straně obrazovky. Maximum plochy monitoru bude využito pro prezentaci uložených dat.

Pro rychlý přehled o nadcházejících akcích a nových událostech bude aplikace vybavena úvodním rozcestníkem obsahujícím výtah nejdůležitějšího z jednotlivých sekcí. Zde bude také možnost informovat organizátory o důležitých událostech, případně zde zpřístupňovat užitečné materiály.

Většina požadavků na funkčnost implikuje tzv. CRUD<sup>2</sup> operaci nad skupinou záznamů. Typická sekce aplikace bude tedy tvořena tabulkou obsahující seznam záznamů. V rámci této tabulky bude možné provádět zmiňované CRUD operace. Podle rozsahu záznamu bude editace probíhat buď inline, tedy přímo v tabulce, nebo zobrazením editačního formuláře. Editační formulář bude shodný s formulářem pro přidání záznamu. Mimo tyto operace bude možné pro daný typ záznamu přidat také další komplexnější operace, které usnadní často prováděné úkony.

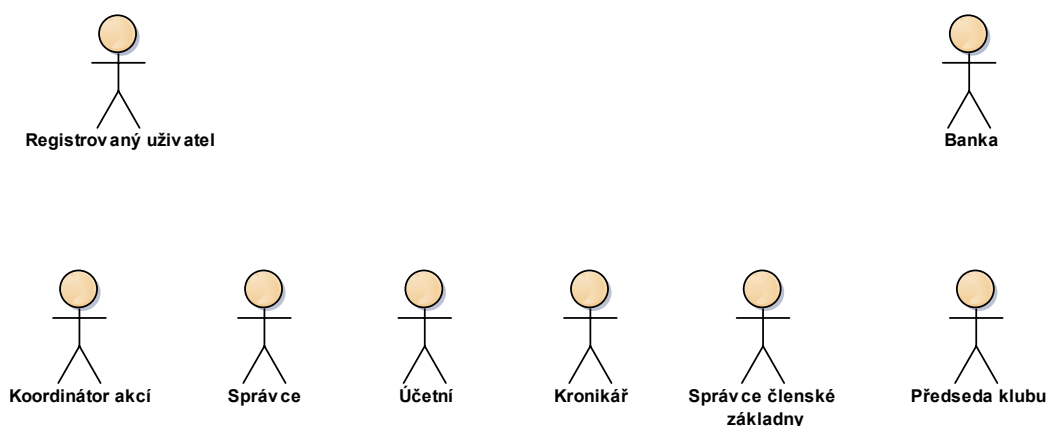
---

<sup>2</sup>Create, Read, Update, Delete - Shrnouje 4 základní operace nad záznamem, vytvoření, čtení, úprava a smazání.



## 4.3 Role

Aplikace bude umožňovat do velké míry role definovat přímo v administraci, protože v rámci vedení klubu se role často prolínají a je tedy třeba s tím počítat. Přesto je pro správné rozdělení sekcí dobré rámcově stanovit role, které budou v aplikaci figurovat.



**Obrázek 4.1:** Vizualizace uživatelských rolí

|                         |  |
|-------------------------|--|
| Banka                   | Banka symbolizuje informační systém banky, který prostřednictvím e-mailu může aplikaci odesílat informace.             |
| Koordinátor akcí        | Koordinátory jsou hlavní vedoucí táborů, organizátoři jednodenních akcí a lektori kroužků.                             |
| Kronikář                | Kronikář zodpovídá za úplnost kroniky a zajišťuje formální správnost zápisů.   |
| Předseda klubu          | Předseda klubu je hlavní osoba v klubu, která schvaluje důležité procesy směrem ven.                                   |
| Registrovaný uživatel   | Registrovaný uživatel je každý, kdo vytvořil přihlášku v rámci systému.  |
| Správce                 | Správce zajišťuje chod informačního systému klubu.   |
| Správce členské základy | Správce členské základy má na starosti evidenci členů v jednotlivých letech a získávání povinných členských příspěvků. |
| Uživatel                | Uživatel je jakýkoli návštěvník aplikace.  |
| Účetní                  | Účetní zajišťuje ekonomické záležitosti klubu a koordinuje získávání grantů.   |

## 4.4 Případy užití

Případ užití definuje část funkcionality systému. Je využíván jednou nebo více rolemi a plní jeden nebo více funkčních požadavků. Pokud funkcionalita nevyplývá plně z názvu případu užití, vytvoří se posloupnost interakcí mezi systémem a uživatelem. Ta se označuje jako scénář případu užití[5].

### 4.4.1 Scénáře případů užití

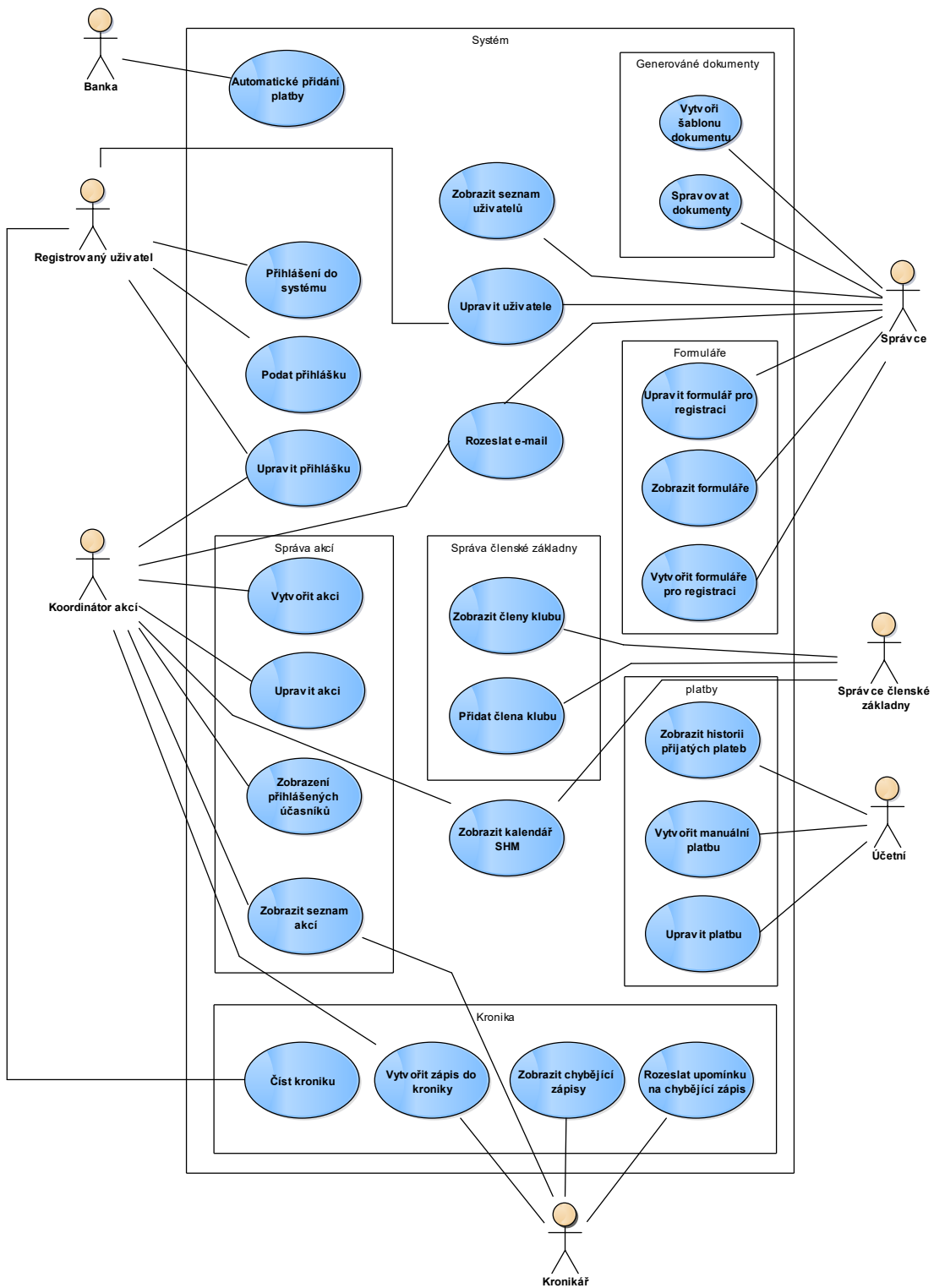
Vzhledem k tomu, že návrh i implementace jsou realizovány stejným autorem a běžné operace nad daty není třeba scénářem popisovat, jsou zde uvedeny pouze scénáře komplexnějších případů užití.

#### Podat přihlášku

1. Systém zobrazí input pro zadání e-mailu.
2. Uživatel vyplní e-mail.
3. Systém zkontroluje, zda je uživatel registrovaný. (Pokud uživatel není zaregistrovaný, pokračuje se bodem 8)
4. Systém zobrazí input pro zadání hesla
5. Uživatel zadá heslo.
6. Pokud heslo souhlasí, pokračuje se bodem 8.
7. Systém zobrazí tlačítko vygenerovat nové heslo a konec.
8. Systém zobrazí seznam akcí, na které lze přihlásit.
9. Uživatel vybere akci.
10. Systém zobrazí příslušný formulář.
11. Uživatel formulář vyplní a odešle.
12. Systém zvaliduje formulář a v případě chyby vrátí na pod 11.
13. Systém zapíše novou registraci.
14. Systém odešle informaci koordinátorům akce a potvrzení uživateli, který přihlášku podal.
15. Systém vypíše informaci o úspěšném zaregistrování.

#### Rozeslat e-mail

1. Systém zobrazí formulář s polem předmět a text zprávy.
2. Uživatel vyplní formulář a klikne na tlačítko vybrat příjemce.
3. Systém uloží e-mail do databáze jako rozepsaný a zobrazí zadávání příjemců.
4. Uživatel vybere okruh příjemců. Má možnost vybrat jako příjemce uživatele, které mají přihlášku na akce, u nichž je koordinátor. Správce může odesílat e-mail komukoli.
5. Systém zobrazí e-maily, na které bude e-mail odeslán.
6. Uživatel potvrdí rozeslání e-mailu.
7. Systém rozešle e-maily (kontrolní pošle také uživateli).
8. Systém zapíše do databáze příjemce a nastaví e-mail jako odeslaný.



Obrázek 4.2: Diagram případů užití

## 4.5 Model

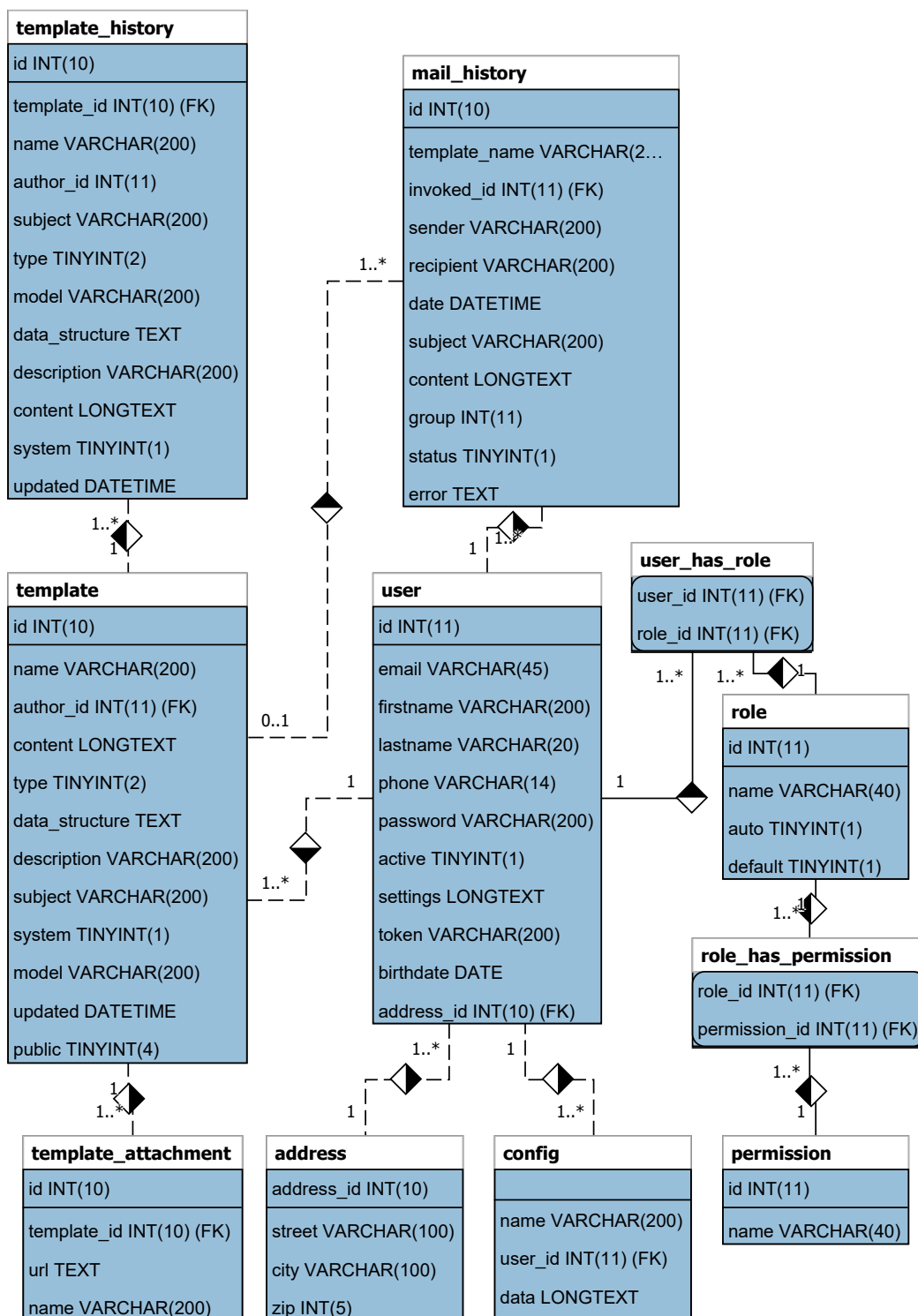
Z důvodu požadavků na univerzální použitelnost (meta model pro formuláře, administrace oprávnění) je základní databázový model pro aplikaci poměrně robustní. Pro přehlednost je zde rozdělen do 3 schémat.

Základem modelu je databáze **user**, která slouží k autentifikaci uživatelů. Obsahuje jak administrátory, tak uživatele spravující členství a přihlášky na tábory. Pro lepší orientaci bude figurovat ve všech částech (v modelu je přítomna pouze jednou).

### 4.5.1 Oprávnění, komunikace a export

Část modelu řešící autentifikaci a řízení oprávnění, dále pak definování šablon souborů pro export a e-mailovou komunikaci. Toto řešení umožňuje maximální flexibilitu použití bez nutnosti zásahu do zdrojového kódu aplikace. Tabulka `template` bude obsahovat i veškeré systémové šablony, které jsou automaticky použity např. při registraci nového uživatele, registraci na událost apod.

|                                  |  |
|----------------------------------|--|
| <code>user</code>                | Obsahuje základní informace o účtech.  |
| <code>address</code>             | Adresa uživatele. Vyčleněno do zvláštní tabulky s relací 1:1 kvůli budoucímu rozšíření aplikace o další entity obsahující adresu.                                  |
| <code>role</code>                | Definuje zastávané role v aplikaci. Příznak <code>default</code> určuje, které role má automaticky každý uživatel.   |
| <code>permission</code>          | Do této tabulky aplikace zapíše každé oprávnění, o které žádá. Následně je možné toto oprávnění přiřadit k příslušným rolím.                                       |
| <code>config</code>              | Univerzální tabulka pro uživatelská nastavení aplikace.  |
| <code>template</code>            | Tabulka obsahující šablony pro generování souborů, exportů a e-mailů.  |
| <code>template_history</code>    | Obsahuje historii změn šablony.  |
| <code>template_attachment</code> | Umožňuje řetězení šablon a příkládání souborů např. pro potřeby přílohy generovaného e-mailu.  |
| <code>mail_history</code>        | Tabulka pro ukládání všech odeslaných e-mailů z aplikace. E-mail může být jak bez šablony, tak může být vygenerován na základě systémové nebo uživatelské šablony. |



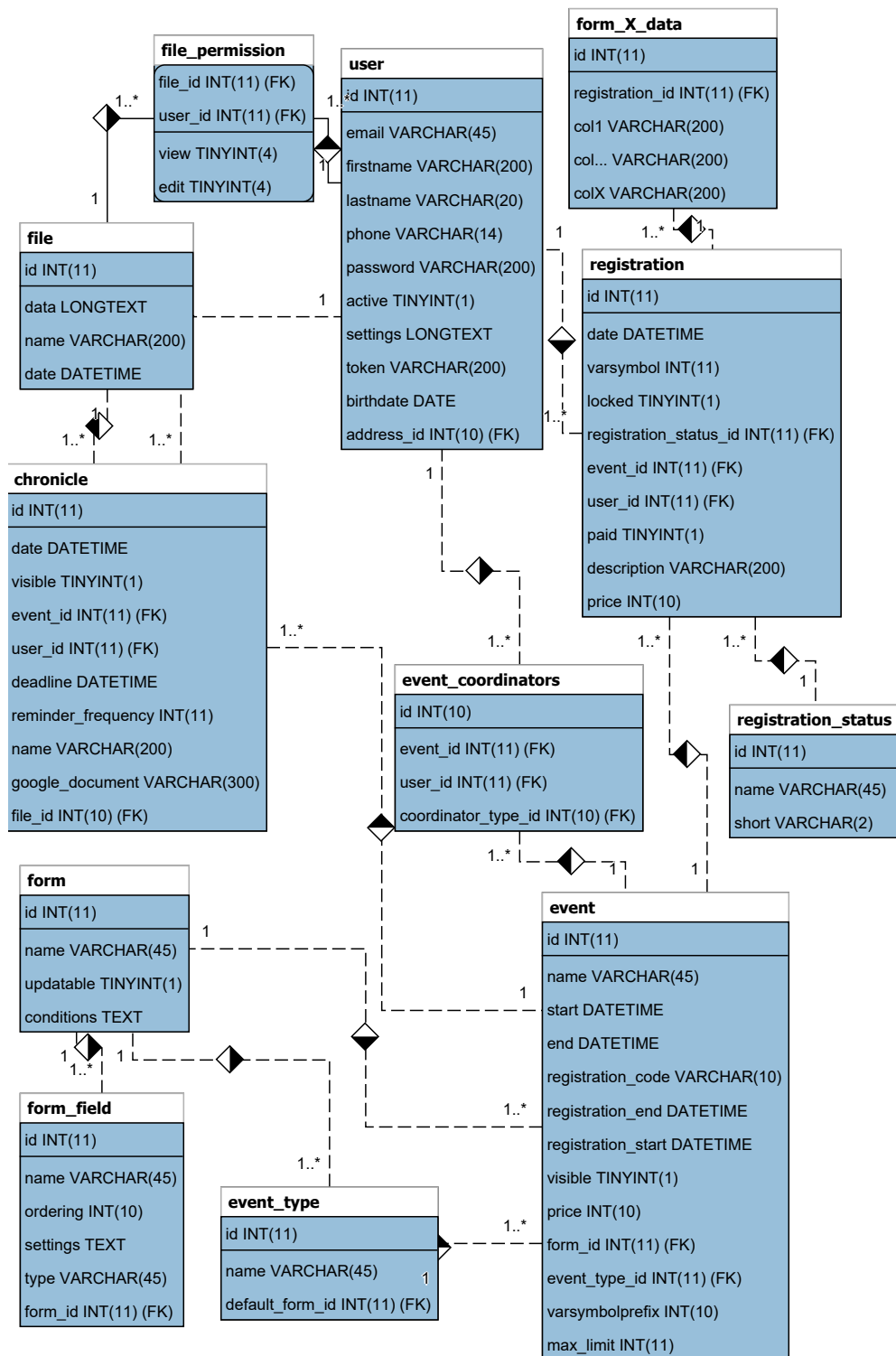
Obrázek 4.3: Databázový model 1. část

### 4.5.2 Události, formuláře a přihlášky

Tento díl modelu představuje strukturu nejdůležitější části aplikace. Nahrazuje původní jednoúčelový přihlašovací systém používaný v klubu, který byl tvořen dvěma tabulkami (události a přihlášky). Na modelu je zajímavé, jak s požadavkem na univerzální použití rapidně narůstá složitost databázové struktury.

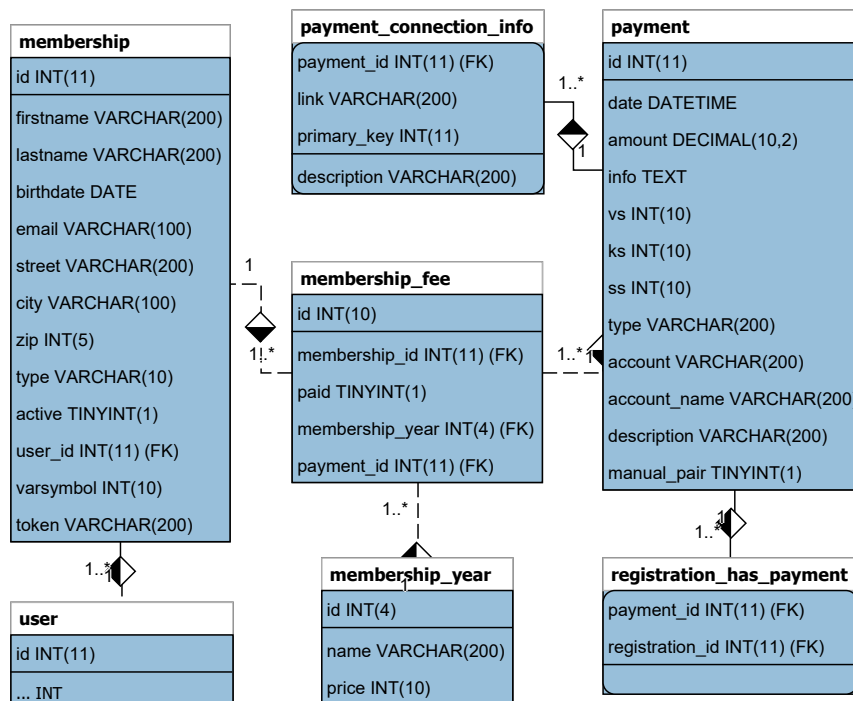
Dalším zajímavým prvkem je hybridní definování tzv. meta modelu pro přihlášky na události. Model obsahuje tabulky pro definování struktury dat, samotná data jsou ale uložena v dynamicky generované tabulce. Podrobněji o řešení pojednává kapitola 5.6.1.

|                     |   |
|---------------------|---|
| event               | Tabulka událostí, které klub organizuje. Zejména jde o tábory a víkendové akce, na které je třeba se přihlašovat.                                   |
| event_coordinators  | Tabulka definující koordinátory konkrétní akce pro účely nastavení oprávnění k události a komunikaci s účastníky.                                   |
| event_type          | Umožňuje rozřídění událostí dle typu.   |
| registration        | Záznam registrace na událost. Každá registrace má svůj stav a je spravována uživatelským účtem.   |
| registration_status | Umožňuje nastavit stavy registrace.   |
| chronicle           | Zápis v kronice, kterou musí klub vést  |
| file                | Tabulka pro ukládání zápisů vytvořených přímo v aplikaci. Vyčleněno z tabulky chronicle pro možnost dalšího využití souborů při rozšíření aplikace. |
| file_permission     | Definuje oprávnění k souboru (zobrazit/editovat).   |
| form                | Reprezentuje administrátory vytvořený formulář, který je možné použít pro registraci na událost.  |
| form_field          | Specifikuje jednotlivé položky registračního formuláře  |
| form_X_data         | Symbolizuje tabulku pro ukládání dat z registrací vytvořených formulářem s id X (viz kapitola 5.6.1).   |



Obrázek 4.4: Databázový model 2. část

### 4.5.3 Platby a správa členů



Obrázek 4.5: Databázový model 3. část

Poslední část modelu zajišťuje evidenci členů a evidenci plateb. Členství jako taková by bylo možné evidovat i v rámci systému univerzálních formulářů. Speciální tabulka ale umožní snadno doplnit evidenci o pokročilejší funkce a tím vytvořit přidanou hodnotu oproti stávající evidenci prostřednictvím Google tabulky.

Tabulka plateb obsahuje sloupce nezbytné pro evidenci bankovních převodů, aby bylo možné do systému platby importovat přímo z bankovníctví a následně automatickými „párovači“ plateb označit příslušné záznamy za zaplacené.

|                          |   |
|--------------------------|---|
| membership               | Tabulka členů klubu.  |
| membership_year          | Seznam roků členství evidovaných v aplikaci.  |
| membership_fee           | Poplatky za jednotlivé roky členství.   |
| payment                  | Zaznamenané platby  |
| registration_has_payment | Tabulka pro relaci N:N mezi platbami a registracemi umožňující variabilitu při hrazení účasti na akcích (více registrací jednou platbou, jedna registrace více platbami). |
| payment_connection_info  | Tabulka s doplňujícími informacemi ke spárované platbě.   |



# Kapitola 5

## Implementace

### 5.1 Volba technologie

Na začátku implementace je třeba vhodně zvolit technologie pro realizaci. Výběr je přizpůsoben požadavkům klubu a osobní zkušenosti autora s různými webovými technologiemi.

#### 5.1.1 Programovací jazyk PHP

Ke splnění požadavku na levný nenáročný provoz byl pro realizaci zvolen programovací jazyk PHP[6]. Ten patří mezi nejčastěji podporované programovací jazyky pro tvorbu webových aplikací. Dle online rozcestníku [hostingy.cz](http://hostingy.cz)[7] je podpora jednotlivých jazyků následující:

| programovací jazyk | počet hostingů v ČR |
|--------------------|---------------------|
| PHP                | 135                 |
| Python             | 25                  |
| Java               | 7                   |

**Tabulka 5.1:** Statistika podpory prog. jazyků webhostingy v ČR

Vzhledem k rozšířenosti PHP existuje nespočet frameworků, které umožní rychlé vystavění základní struktury aplikace. Framework je nadstavba pro softwarové prostředí, která programátorovi usnadňuje práci prostřednictvím hotových částí programu, návrhových vzorů a doporučených postupů. Výhoda použití frameworku tkví v tom, že zpravidla nabízí řešení pro nejčastěji požadované funkce v daném prostředí. V rámci webových aplikací se jedná např. o autentifikaci, komunikaci s databází, práci se soubory atd.

### ■ 5.1.2 Databáze MySQL

Veškerá data aplikace budou ukládána v rámci relační databáze. Zde je opět kladen důraz na snadné nasazení a nízkou cenu, proto aplikace využívá open source<sup>1</sup> databázi MySQL. Z hlediska dostupnosti je na tom lépe[8] robustnější systém PostgreSQL. Předpokládané vytížení aplikace je ale maximálně desítky uživatelů online. Proto bude aplikace primárně navržena pro práci s MySQL databází, která je nejodpojovanějším databázovým systémem v rámci webhostingů v ČR[6].

### ■ 5.1.3 Client-side technologie

Jako client-side se u webové aplikace označuje část, která je zpracovávána na straně uživatele, tedy pro svůj běh využívá výkonu a zdrojů přímo v jeho zařízení. Pro vykreslení ve webovém prohlížeči je použit značkový jazyk HTML, společně s kaskádovými styly CSS. Pro dynamické změny aplikace na straně uživatele se používá javascript.

Nejen u serverové části, ale i v klientské sekci je dobré použít frameworky, které obstarají základní požadavky na aplikaci a programátor může tak více času věnovat specifitějším částem implementace. Pro rychlé zhotovení vizuální stránky aplikace byl použit framework Twitter Bootstrap[9]. Jeho výhodou je masivní rozšíření po celém světě. Díky tomu existuje řada placených i volně dostupných šablon, které lze pro vystavení aplikace použít. Pro IS klubu byla využita šablona Matrix Admin[10], která přesně odpovídá požadavkům na administrátorskou část webu.

Jednodušší provedení dynamických operací umožňuje javascriptový framework jQuery[10], který je opět v rámci webových aplikací hojně rozšířen. Řada jak šablon pro Twitter Bootstrap, tak komponent pro framework Nette rovnou počítá s užitím knihovny jQuery. Zvolené technologie tedy společně fungují ve vzájemné synergii a jedná se o velmi časté spojení při realizaci webové aplikace.

### ■ 5.1.4 Vývojové prostředí

Pro realizaci bylo použito vývojové prostředí IntelliJ IDEA[11] (licence k dispozici v rámci studia na ČVUT). K podpoře syntaktických specifik Nette frameworku byl použit plugin Latte od Jana Tvrdíka[12]. Ke správě PHP knihoven byl použit Composer[13]. Pro získávání závislostí html a javascriptových knihoven bylo užito nástroje Bower[14]. Vývoj a testování probíhalo pomocí softwarového balíčku XAMPP[15] na operačním systému Windows 10. Ten slouží k běhu testovacího Apache serveru a MySQL serveru přímo na počítači programátora.

---

<sup>1</sup> Open source - počítačový software s otevřeným zdrojovým kódem; technická dostupnost kódu; legální dostupnost

## 5.2 Nette framework

Pro realizaci je použit PHP framework Nette[16]. Jedná se o nejpoužívanější PHP framework v ČR[17] a 3. nejpopulárnější PHP framework na světě[18]. Je pro něj k dispozici přes 600 doplňků a rozšíření[19], které lze využít k efektivnímu dosažení požadované funkčnosti. Framework využívá MVC architekturu (viz kapitola 5.3), klade důraz na bezpečnost aplikace, obsahuje nástroje na hledání a odstranění chyb, umožňuje cachování<sup>2</sup> aplikace pro zrychlení běhu, zjednodušuje vytváření šablon uživatelského rozhraní. Obsahuje celou řadu hotových součástí pro nejběžnější operace v aplikaci, jako je vytváření formulářů, autentifikace, odesílání e-mailů a přístup k databázi (Nette Database).

### 5.2.1 Latte šablony

Důležitou součástí Nette je šablonovací systém Latte. Ten umožňuje zkrátit kódování finální podoby webu na minimum. Latte zavádí speciální makra ve složených závorkách, která obstarávají nejběžnější funkce, jako je výpis cyklem `foreach` atd. Při první běhu aplikace se šablony zkompilují do obyčejného PHP souboru. Nedochází tedy ke zpomalení aplikace.

#### ■ Příklad úspornosti Latte na výpisu seznamu

**Ukázka kódu 5.1:** Příklad Latte zápisu [20]

```
<ul n:if="$items">
{foreach $items as $item}
  <li id="item-{$iterator->counter}">{$item|capitalize}</li>
{/foreach}
</ul>
```

**Ukázka kódu 5.2:** Srovnání s PHP kódem bez použití Latte

```
<?php if ($items): ?>
  <?php $counter = 1 ?>
  <ul>
    <?php foreach ($items as $item): ?>
      <li id="item-<?php echo $counter++ ?>"><?php
        echo htmlspecialchars(mb_convert_case($item,
        MB_CASE_TITLE)) ?>
      </li>
    <?php endforeach ?>
  </ul>
<?php endif?>
```

### 5.2.2 Nette database

Nette database je součástí Nette Frameworku, která tvoří most pro práci s databází. Díky tomu je možné provozovat aplikaci snadno s různými databázovými systémy bez nutnosti výrazných zásahů do implementace. Plně podporuje databázové systémy MySQL, PostgreSQL, SQLite 3 a MS SQL.

<sup>2</sup>Cache - uchování dat pro zrychlení běhu při dalším použití

## 5.3 MVC

Model View Controller je softwarová architektura, která od sebe odděluje aplikační logiku, obsahu aplikace a způsob prezentace dat. To napomáhá srozumitelnosti kódu, předchází bezpečnostním chybám, či nechtěným zásahům do dat a usnadňuje práci v týmu. Tento přístup funguje v Nette Framework následujícím způsobem:

### 5.3.1 Model

Model obsahuje aplikační logiku a propojení s databází. V Nette je model tvořen zpravidla objekty typu `\Nette\Object`, které sdružují operace určitého logického celku a vytvářejí službu, kterou je možné v aplikaci dále použít. Taková služba je pak zaregistrována v Nette v rámci konfiguračního souboru `config.neon`, kde je možné jí nastavit název.

**Ukázka kódu 5.3:** Příklad definování služby (s názvem / bez názvu)

```
services:
    user: Model\Security\AuthorizableUser
    - Model\Address
```

### 5.3.2 View

View definuje, jak se mají data získaná z aplikace zobrazit. V Nette tuto funkci plní šablonovací systém Latte, který je podrobněji popsán v kapitole 5.2.1.

### 5.3.3 Controller

Controller propojuje model a view. Získává data z modelu, případně je upravuje pro potřeby view. Slouží k vytvoření požadovaných komponent výsledné rozhraní aplikace. V Nette tuto funkci plní tzv. presenter. Presenterů je typicky v aplikaci několik, přičemž každý presenter má na starosti část spolu souvisejících pohledů (use case) aplikace. V informačním systému klubu jsou definovány tyto presentery:

- Administrátorská část
  - BankPresenter - Obsluha plateb pro účetní
  - CalendarPresenter - Zobrazení kalendáře akcí
  - ChroniclePresenter - Správa kroniky
  - ConfigPresenter - Nastavení aplikace
  - ErrorPresenter - Zpracování chybových hlášení
  - EventPresenter - Události organizované klubem
  - FormsPresenter - Vytváření a správa formulářů
  - HomepagePresenter - Úvodní přehled v aplikaci
  - LoginPresenter - Zpracování přihlášení a obnova hesla

- MailPresenter - Rozesílání e-mailů
- MemberPresenter - Správa členské základny
- TemplatePresenter - Administrace šablon a generování souborů
- UserPresenter - Vytváření a správa uživatelů
- Uživatelská část
  - FrontPresenter - Přihlašování na události
  - MemberPresenter - Správa členství uživatele
  - TemplatePresenter - Generování souborů dostupných pro uživatele
  - UserPresenter - Detail uživatelského účtu

## 5.4 Struktura aplikace

Adresářová struktura vychází z doporučení Nette dokumentace. Pro přehlednost jsou související komponenty forms a grid, kterých je na webu větší množství, vyčleněny do zvláštních složek.

```

+—app
|   +—AdminModule - presentery a šablony administrátorské části
|   |   —presenters
|   |   —templates
|   +—components - znovupoužitelné komponenty pro použití v presenterech
|   |   —templates
|   +—config - konfigurační soubory
|   +—forms - definice formulářů pro použití v presenterech
|   +—FrontModule - presentery a šablony uživatelské části
|   |   —presenters
|   |   —templates
|   +—grids - definice tabulkových výpisů pro použití v presenterech
|   +—model - aplikační logika a propojení s databází
|   —router - definice url adres aplikace
+—doc - Nette dokumentace
+—log - záznamy o chybách a výjimkách
+—temp - dočasné soubory (cache, zparsované šablony,...)
+—tests - Nette tester skripty
|   —output - výstupy testů
+—vendor - knihovny
—www - Client-side soubory aplikace
+—bower_components - knihovny získané prostřednictvím bower
+—css - kaskádové styly aplikace
+—js - javascriptové aplikace
+—source - prostor pro uživatelská data nahraná v aplikaci
+—thumbs - náhledy pro uživatelská data
  
```

## 5.5 Klíčové komponenty

V rámci této kapitoly jsou rozebrány základní součásti aplikace a jejich specifika, na kterých je postavena celá další implementace.

### 5.5.1 Entity a oprávnění

Z analýzy vyplynulo, že aplikace bude obsahovat řadu entit, nad kterými je třeba provádět CRUD operace s kontrolou oprávnění. Pro tyto účely je v modelu navržena abstraktní třída `UniversalModel`. Terminologií jazyka se jedná v podstatě o entity manager[21] spojený s funkcí služebníka[22]. Entitou je pak konkrétní záznam v databázi reprezentován objektem `IRow` z `Nette Database`. `UniversalModel` řeší řízení oprávnění a další základní operace s entitou, které jsou využívány tabulkovými výpisy a exporty. Pro práci s oprávněními definuje následující metody:

- `hasPrivilege($privilege)`
- `hasInstancePrivilege($privilege, $item, $allowGlobalPermission = false)`
- `getAllowedInstances()`

Mimo tyto metody umožňuje `UniversalModel` číst, vkládat, upravovat a mazat data s kontrolou oprávnění následujícími metodami:

- `get($id, $secured = true)`
- `insert($values, $secured = true)`
- `update($primaryKey, $values)`
- `delete($id)`

Získání záznamů a přidávání je možné i bez oprávnění nastavením parametru `secured` na `false`. Tato možnost je určena pro specifické případy, např. pokud aplikace potřebuje získat seznam dat, která ale uživateli nevypisuje a pouze s nimi provádí další operace. Při tomto kroku by kontrola práv operaci zbytečně zpomalovala, a proto je možné použít parametr `secured`. Pro úpravu a mazání tato možnost z bezpečnostních důvodů není, protože se jedná o nevratné změny v datech aplikace.

Další funkcí struktury `UniversalModel` je připravovat data pro použití v šablonách pro export a e-mail. Např. při generování dokumentu k události je dobré mít možnost použít data o jednotlivých přihláškách. K tomu slouží metoda `getDataForRowTemplate`, která vrací data použitelná v šabloně pro konkrétní záznam v databázi. Tato metoda je pro každou entitu přepsána v příslušném manageru.

**Ukázka kódu 5.4:** Příprava dat pro šablony k entitě členství

```
public function getDataForRowTemplate(IRow $row)
{
    return [
        "cLen" => $row,
        "spravce_clena" => $row->ref("user"),
        "roky_clenstvi" => $row->related($this->memberFee)
    ];
}
```

Pro možnost využít data z výpisu k rozeslání e-mailu je v třídě `UniversalModel` metoda `getEmailParser`, která ze záznamů v databázi získává e-maily k nim relevantní. To je následně využíváno v tabulkovém výpisu (viz kapitola 5.5.2).

**Ukázka kódu 5.5:** Získání relevantního e-mailu k členství

```
public function getEmailParser() {
    return [
        "E-mail_clena" =>
            function(IRow $row) {
                return [$row->email =>
                    $row->firstname." ".$row->lastname];
            },
        "E-mail_spravce_clenstvi" =>
            function(IRow $row) {
                $usr = $row->ref("user");
                return [
                    $usr->email => ($usr->lastname == "" ?
                        $usr->email :
                        $usr->firstname." ".$usr->lastname)
                ];
            }
    ];
}
```

## 5.5.2 Tabulkový výpis

V administraci je opakovaně potřeba poskytovat přehledně seznam dat v databázi. Pro tyto účely se využívá zpravidla tabulkový výpis (tzv. datagrid). V aplikaci byla použita komponenta Ublaboo datagrid[23]. Mezi její hlavní výhody patří filtrování, řazení, velká škála podporovaných formátů vstupních dat, export dat, hromadné operace a funkce editace záznamů přímo v tabulce.

Vybraný datagrid je rozšířen o základní nastavení pro všechny výpisy pomocí objektu `MyDataGrid`, kde jsou definovány překlady, počty položek na stránku a export. Takto rozšířený výpis umožňuje dále zapnout úpravu záznamů nad daty využívající práci s oprávněními prostřednictvím struktury `UniversalModel`, jak je popsána v kapitole 5.5.1. Vše zařídí metoda `inlinize`, která datagrid napojí na model, sama zjistí práva aktuálního uživatele a dle toho zpřístupní/nezpřístupní přidávání, editaci a mazání záznamů.

V požadavcích na informační systém je kladen důraz na exporty a ušetření času při vyplňování nezbytných dokumentů. Tyto požadavky systém plní univerzálně prostřednictvím možnosti vytvořit šablonu pro libovolná data využívající `UniversalModel`. Pro generování šablon a odesílání e-mailů pak slouží další rozšíření datagridu na třídu `OperationsDataGrid`. Ta přidává do výpisu hromadné operace pro vytvoření šablon, vygenerování souboru dle šablony, nebo odeslání hromadného e-mailu.

| Hromadné akce: Odeslat hromadný e-mail |          | seznam členství | Provést           | Export do CSV                       |
|--|----------|-----------------|-------------------|-------------------------------------|
| Příjmení                               | Telefon  | počet čl.       | Role              | Akce                                |
| test1@kablík...                        | Taddeo   | Gashion         | Kronikář, Správce | <input checked="" type="checkbox"/> |
| test2@kablík...                        | Nicole   | Sweating        | Vedoucí           | <input checked="" type="checkbox"/> |
| test3@kablík...                        | Yorgo    | Yakobowitch     |                   | <input type="checkbox"/>            |
| test4@kablík...                        | Doralynn | Jeannot         |                   | <input type="checkbox"/>            |
| test5@kablík...                        | Biddy    | Izard           |                   | <input type="checkbox"/>            |

( Položky: 0 - 5 z 5 ) Resetovat filtr 10

Obrázek 5.1: Tabulkový výpis `OperationsDataGrid` v IS

## 5.5.3 Univerzální formulář

V administrátorské části se nachází také mnoho formulářů pro zadávání dat. Nette samo o sobě nabízí velkou podporu pro vytváření formulářů a validaci vstupních dat. V aplikaci je navíc použito rozšíření Nextas Forms[24], které přidává dynamickou validaci některých prvků formuláře a přidává podporu stylů Twitter Bootstrap[9]. I tak je objekt formulářů v aplikaci dále rozšířen o spolupráci s `UniversalModel`.

`UniversalForm` nabízí několik vylepšení, která snižují množství kódu v presenterech. Jedná se např. o metodu `addInputs`, která pouze na základě pole ve tvaru (`[column] => [label]`) vytvoří formulářové prvky dle typu sloupce v databázi,



včetně základních validačních pravidel (např. povinnost/nepovinnost pole). Obsahuje také výchozí chování pro vytvoření, či editaci záznamu. Propojení s databází je pak možné pomocí jediné metody `setInsertOnSuccess` (případně `setUpdateOnSuccess` pro úpravu záznamu).

V následující ukázce z klientské části je vidět, jak `UniversalForm` automaticky doplní varování o povinnosti vyplnit pole, nebo přidá klikací výběr data pro sloupec typu `date`.

#### Ukázka kódu 5.6: Kód pro vytvoření formuláře

```
$form = new UniversalForm("member", $this->members);
$form->addInputs([
    "firstname" => "Jméno",
    "lastname" => "Příjmení",
    "birthdate" => "Datum narození",
    "email" => "E-mail",
    "street" => "Ulice",
    "city" => "Město",
    "zip" => "PSČ",
]);
$form->enableAjax();
$form->addSubmit('add', "Přidat členství do klubu")
    ->setAttribute('class', 'btn-primary');
```

The screenshot shows a dark-themed web form for adding a membership. The form is titled "Přidat členství" and has the "Orgament" logo in the top left. It contains several input fields with red error messages below them: "Jméno" (Name), "Příjmení" (Surname), "Datum narození" (Date of birth) which is filled with "04.02.1995", "E-mail", "Ulice" (Street), "Město" (City), and "PSČ" (Postal code). A calendar widget is open over the date field, showing the month of February 1995. The calendar has a header "únor 1995" and a grid of days. The 4th of the month is highlighted in blue. The form also includes a submit button labeled "Přidat členství do klubu" with the class "btn-primary".

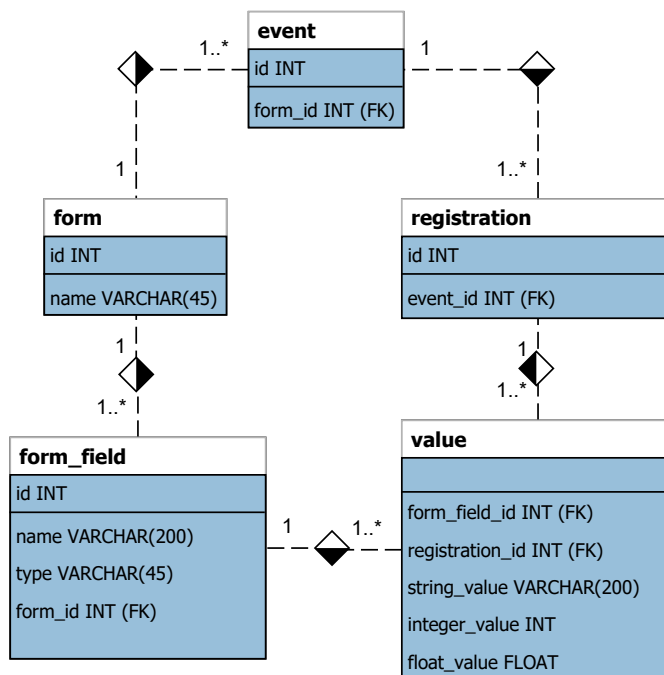
Obrázek 5.2: Výsledný formulář vygenerovaný s použitím metody `addInputs`

## 5.6 Implementační zajímavosti

Následující kapitola se podrobněji věnuje složitějším problémům, ne příliš často řešeným funkcím a zajímavým řešením.

### 5.6.1 Editor přihlašovacích formulářů

Největším úkolem v rámci implementace systému je vytváření dynamických formulářů pro registrace na události. Původní záměr byl vytvořit v databázi meta model a následně k datům přistupovat např. pomocí vytvoření view[25].



Obrázek 5.3: Původní návrh modelu

V průběhu implementace se ale ukázalo, že toto řešení není zcela ideální. View by tak jako tak bylo nutné vytvářet zvláště pro každý formulář. Jejich použití podstatně komplikuje úpravu záznamů prostřednictvím hotových komponent. Data jsou v případě problémů, pádu systému atd. velice nepřehledná. Jak naznačuje schéma 5.3, pro použití různých datových typů by buď v rámci form\_field bylo zapotřebí několik sloupců, přičemž data by obsahoval vždy pouze jeden datový sloupec na řádek, nebo by pro každý typ prvku ve formuláři musela být v databázi další tabulka. To by situaci dále komplikovalo. Ze znalosti problematiky vyplývá, že použitých formulářů budou v rámci aplikace jednotky, maximálně desítky. Naopak některé mohou obsahovat i velmi mnoho polí. View by pak obsahovalo např. 10-15 left join spojení, což se nejevilo jako nejvhodnější řešení.

Pro zmíněné důvody je nakonec finální řešení jiné, jak je vidět v kapitole 4.5.2. Aplikace po vytvoření formuláře vygeneruje tabulku form\_X\_data, kde X označuje

id formuláře. Zde jsou prvky formuláře reprezentovány sloupci. V tomto řešení může být datový typ sloupce definován přesně podle potřeb prvku formuláře a o základní validaci dat se tedy postará samotná databáze. Veškeré operace s daty jsou pak mnohem jednodušší, protože se v rámci výpisu účastníků pracuje pouze s tabulkou registrace spojenou s konkrétní tabulkou dat dle formuláře. Navíc data zůstávají snadno čitelná i v případě, že by bylo třeba s daty na úrovni databáze dále pracovat (viz obrázek 6.2).

| id | ref | col651<br>Jméno | col652<br>Příjmení | col653<br>Datum narození | col655<br>Telefon účastníka | col656<br>Telefon matka | col657<br>Telefon otec | col659<br>Ulice | col660<br>Číslo Popisné | col661<br>Město | col662<br>PSC | col66<br>Další kontakt e-mail | col666<br>Poznámky   |
|----|-----|-----------------|--------------------|--------------------------|-----------------------------|-------------------------|------------------------|-----------------|-------------------------|-----------------|---------------|-------------------------------|----------------------|
| 1  | 90  | Lorem           | Ipsum              | 1995-02-04               | 123 456 789                 | 123 456 789             |                        | Testovací       | 123                     | Praha           | 11000         |                               | Poznámka k přihlášce |
| 2  | 91  | Lorem           | Ipsum              | 1995-02-04               | 123 456 789                 | 123 456 789             |                        | Testovací       | 123                     | Praha           | 11000         |                               | Poznámka k přihlášce |
| 3  | 92  | Lorem           | Ipsum              | 1995-02-04               | 123 456 789                 | 123 456 789             |                        | Testovací       | 123                     | Praha           | 11000         |                               | Poznámka k přihlášce |

**Obrázek 5.4:** Zobrazení dat přímo v databázi pomocí PhpMyAdmin

Samotný editor formulářů pak disponuje řadou předpřipravených prvků, které umožňují přesnou validaci vstupních dat. Dále umožňuje přidávat nadpisy a dodatečné vysvětlivky přímo do formuláře. V administraci je také možné definovat složitá validační pravidla, která pracují s více poli najednou apod. V rámci validace lze používat i funkce count, strlen, strpos, is\_int, preg\_match.

#### ■ Příklady validačních pravidel

- `*1* != *2*` : Pole se musí schodovat.
- `strlen(*3*) < 5` : Pole musí mít alespoň 5 znaků.
- `strlen(*3*) < 5 || strlen(*3*) > 10` : Pole musí mít mezi 5 - 10 znaky.
- `strpos(*3*, "slovo")` : Pole 3 nesmí obsahovat "slovo".

**Vytvořit formulář**

1 Textové pole   povinné pole

2 Číslo   povinné pole  
Minimální hodnota:   
Maximální hodnota:

3 E-mail   povinné pole

4 Výběr   povinné pole  
Hodnoty k výběru:  Možnosti odděluje čárkou (,).

**Náhled formuláře Nový formulář 7**

Název týmu   
Zadejte prosím Název týmu.

Počet členů týmu   
Maximální hodnota je 10.

Kontaktní e-mail   
Zadejte prosím platný e-mail.

Kdy dorazíte?

**(a)** : Editor formulářů

**(b)** : Výsledný formulář

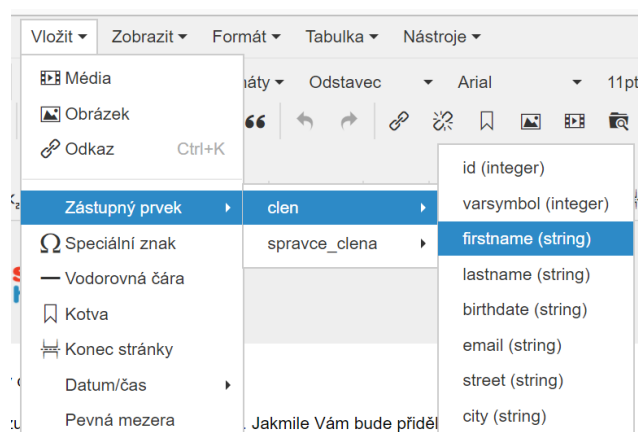
**Obrázek 5.5:** Ukázka formulářového editoru

### 5.6.2 E-mail a šablonovací systém

Protože je předpoklad, že informační systém bude uživateli používán velice pasivně (pouze několikrát za rok pro přihlášení na akci apod.), je nezbytné, aby o veškerých informacích informoval uživatele a administrátory e-mailem. Proto je v aplikaci přítomen univerzální šablonovací systém, prostřednictvím kterého lze jednoduše přímo v administraci změnit veškeré e-maily, které systém zasílá. Pokud se v průběhu vývoje objeví další systémová zpráva, kterou je třeba zasílat, vytvoří se systémová šablona a administrátorovi webu je automaticky zaslán e-mail, že je třeba definovat e-mail pro novou operaci.

Pro každou šablonu (ať už e-mailovou, nebo šablonu dokumentu) je možné poskytnout libovolná data ve tvaru pole, Selection[26], nebo iRow[27]. Systém z nich při prvním vyžádání šablony vytvoří schéma zástupných proměnných, která lze při psaní šablony používat.

Pro editaci šablon je na webu použit wysiwyg[28] editor TinyMCE[29]. Do něj je přidána možnost vložení zástupné proměnné prostřednictvím kontextové nabídky.

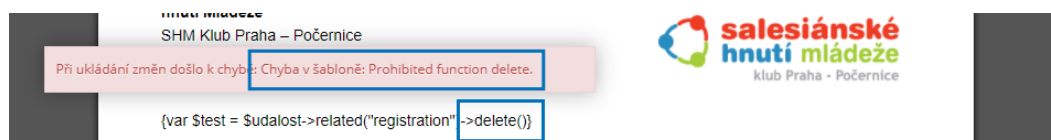


Obrázek 5.6: Vložení zástupné proměnné v editaci šablony pro členství

### SafeEngine

Aby byl systém psaní šablon reálně použitelný, je zapotřebí, aby bylo možné provádět s daty základní operace typu procházení, sčítání hodnot, apod. Nette disponuje šablonovacím jazykem Latte (viz kapitola 5.2.1), proto se nabízí jej využít i v případě šablon. Problém použití je ale bezpečnost aplikace. Prostřednictvím Latte by bylo možné přistupovat k libovolné funkci jazyka PHP, což je z bezpečnostního hlediska nepřijatelná varianta. Z tohoto důvodu byla původní třída zpracovávající šablony rozšířena na třídu **SafeEngine**. Ta při kompilování zjišťuje, zda je šablona bezpečná a přistupuje jen k datům, která jsou do šablony vložena systémem. Povoluje jen základní PHP funkce `is_array`, `strlen`, `strpos` a funkce pro procházení dat v `Selection`. Jednoduchá kontrola bezpečnosti je možná díky PHP funkci `token_get_all`, která detekuje PHP syntaxi a vytvoří pole použitých funkcí, proměnných, textů a dalších konstruktů.

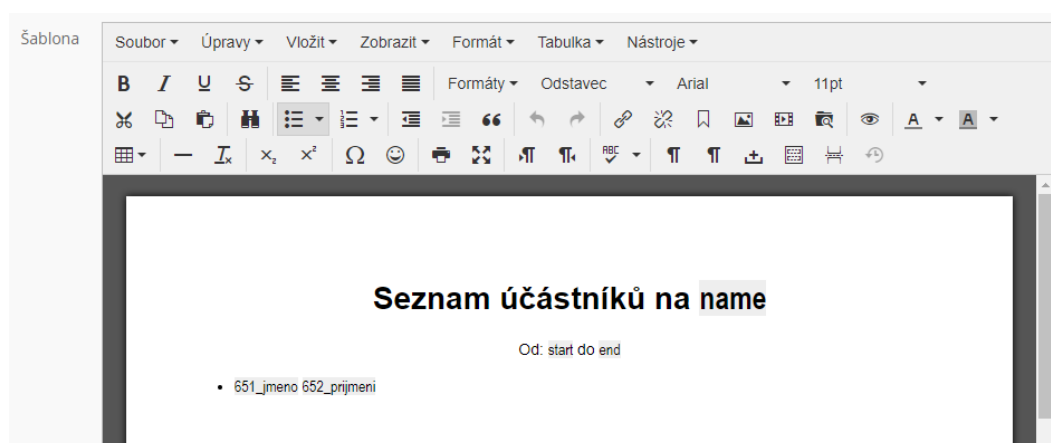
Následující příklad ukazuje, jak by bylo možné bez příslušných oprávnění smazat záznamy v databázi prostřednictvím šablony, pokud by byl použit původní Nette Latte engine. Jak je vidět, systém funkci delete detekoval a nedovolil šablonu upravit.



Obrázek 5.7: Zmařený pokus o smazání přihlášek přes šablonu

## ■ Detekce výpisů

Mimo základních Latte maker aplikace dále v šablonách usnadňuje výpis v seznamech a tabulkách. Automaticky detekuje zda se v seznamu, nebo řádku vyskytuje proměnná obsahující průchod přes více entit a v takovém případě je průchod realizován na html prvku `li` (položka seznamu), resp. `tr` (řádek tabulky).



Obrázek 5.8: Příklad použití šablony



Obrázek 5.9: Vygenerovaný soubor

| VS       | Stav     | Jméno     | Příjmení | Datum narození | Telefon účastníka | Telefon matka | Telefon otoc | Ulice     | Číslo Poplání | Město    | PSČ      | Další kontaktní e-maily | Poznámky          | Cena     | Zapl. | Registrován  | E-mail           | Akce |
|----------|----------|-----------|----------|----------------|-------------------|---------------|--------------|-----------|---------------|----------|----------|-------------------------|-------------------|----------|-------|--------------|------------------|------|
| Filtr... | Filtr... | Filtr...  | Filtr... | Filtr...       | Filtr...          | Filtr...      | Filtr...     | Filtr...  | Filtr...      | Filtr... | Filtr... | Filtr...                | Filtr...          | Vše ▾    |       | Filtr...     |                  |      |
| 1708001  | P        | Kurt      | Kiersten | 4. 2. 1995     | 123 456 789       | 123 456 789   |              | Testovací | 123           | Praha    | 11000    |                         | Poznámka k při... | 1 600 Kč | Ne    | 28. 11. 2017 | regy42@gmail.com |      |
| 1708002  | P        | Anastasia | Larrie   | 4. 2. 1995     | 123 456 789       | 123 456 789   |              | Testovací | 123           | Praha    | 11000    |                         | Poznámka k při... | 1 600 Kč | Ne    | 28. 11. 2017 | regy42@gmail.com |      |
| 1708003  | P        | Noreen    | Kian     | 4. 2. 1995     | 123 456 789       | 123 456 789   |              | Testovací | 123           | Praha    | 11000    |                         | Poznámka k při... | 1 600 Kč | Ne    | 28. 11. 2017 | regy42@gmail.com |      |

Obrázek 5.10: Zdrojové přihlášky



## Kapitola 6

### Testování

#### 6.1 Nette tester

Nette obsahuje nástroj pro testování Nette tester. Ten umožňuje základní testování výstupů funkcí (`same`, `equal`, `true`, apod.), ale také složitější testování presenteru a jeho výstupu. Pro testování výsledné HTML stránky lze použít třídu `DomQuery`. S její pomocí lze testovat přítomnost prvků na webu dotazováním v css notaci.

Aplikace jako taková, mimo zmíněné implementační zajímavosti neobsahuje složitou aplikační logiku. Typickým příkladem, kdy je vhodné používat Nette tester, jsou funkce výpočtů (např. dopočítávání DPH v e-shopech, vypočítávání průměrů, statistik). Nic takového ale navržený informační systém neobsahuje. Jedná se zpravidla pouze o načítání a ukládání dat z databáze, jejich modifikace a reprezentace. Nejčastěji se chyba při implementaci objevovala v Latte šablonách, kdy např. byl použit jiný datový sloupec apod. Takové chyby jsou nejsnáze odhalitelné prostým zkontrolováním aplikace v prohlížeči již při vývoji.

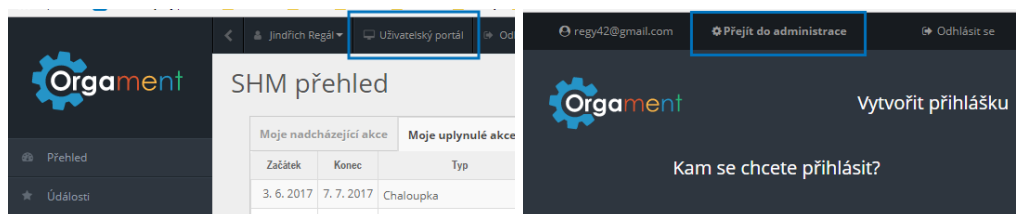
Složitější konstrukty, které se v systému vyskytují, spočívají ve funkcích mající silnou vazbu na client-side část. Pro tyto funkce (odesílající ajax requesty, atd.) testování přes Nette Tester není možné. Existují nástroje (např. Selenium[30]), které umožňují napsat test běžící přímo v prohlížeči. Čas pro vytváření takových testů by byl však neúměrný jak době implementace, tak času, za který lze funkce aplikace otestovat dobrovolníky v klubu.

Nette se samo o sobě snaží předejít chybám a odbourávat typické problémy PHP prostřednictvím debugovacího nástroje Nette Tracy. Ten v případě objevené chyby vypíše chybu přehledně včetně zdrojového kódu, kde k chybě došlo. v produkční verzi pak tento log uloží na server a administrátorovi aplikace pošle e-mail o nastalé chybě. Je tak možné rychle odhalit a opravit i chybu v již nasazené aplikaci.

## 6.2 Testy použitelnosti

Pro odstranění drobných chyb, případně chyb v návrhu mající vliv na ergonomii aplikace se jako vhodné jevílo uživatelské testování[31] dobrovolníky v klubu SHM Praha - Počernice. Ti byli seznámeni s jednotlivými funkcemi dle jejich agendy v klubu. Jejich podněty byly postupně do aplikace začleňovány.

Jedním z odhalených nedostatků bylo např. nedostatečné propojení uživatelské a administrátorské části. Dobrovolníci byli informováni, že administrátorská část je přístupná pod relativní adresou /admin. Uživatelé s menší znalostí webových aplikací se ale v případě, že se dostali do uživatelské části, neuměli vrátit do administrátorské části. Proto do obou variant přibylo tlačítko, které obě části propojuje. To je pochopitelně viditelné jen pro účty mající zvláštní oprávnění.

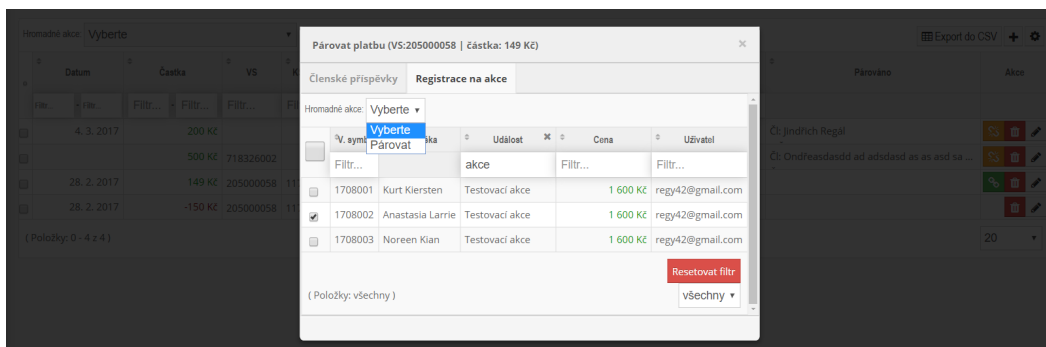


(a) : Administrátorská část

(b) : Klientská část

Obrázek 6.1: Přidané propojovací tlačítko

Dalším problémem bylo párování plateb. Systém se snaží automaticky rozpoznat, které přihlášky a členství je možné zaplatit danou platbou. Např. při platbě za členství může uživatel spravující více členství (zpravidla rodič platí členství nejen své, ale i svých dětí) stačí poslat jednu platbu z VS jednoho ze členství a v případě vyšší částky se automaticky zaplatí i dalším spravované stejným účtem. Podobně funguje párování i u plateb za tábory. Přesto je někdy potřeba platbu ručně spárovat. v původní verzi bylo možné platbu spárovat dodatečným vyplněním VS. Tím je ale zasahováno do importovaných dat, která neodpovídají stavu na bankovním účtu a navíc bylo nutné nejdříve dohledat variabilní symbol v přihláškách. Z tohoto důvodu byla do systému na žádost účetní klubu přidána možnost manuálního párování, aby bylo možné párování plateb pohodlně opravovat.



Obrázek 6.2: Přidání funkce párování dle závěru z testování



## 6.3 Bezpečnost

Při tvorbě webové aplikace je třeba dbát na ochranu před XSS a CSRF[32]. Před XSS, tedy podstrčením nebezpečného kódu v GET, POST, nebo cookie parametrech, chrání Nette aplikaci automaticky. Při generování šablony automaticky všechny proměnné obalí do příslušné „escapovací“ funkce dle kontextu šablony[20].

Před CSRF Nette chrání formuláře pomocí metody `addProtection`. Ta je volána v konstruktoru třídy `MyForm`, přes kterou jsou vytvořeny všechny formuláře aplikace. Navíc všechna oprávnění k přístupu a manipulaci s daty jsou zabudována přímo v modelu díky třídě `UniversalModel`. Pokud by tedy došlo k neoprávněnému signálu ze strany uživatele, který by se začal zpracovávat, nejpozději v modelu dojde k výjimce a zapsání chyby do logu.

## 6.4 Akceptační test

Vzhledem k tomu, že návrh a rozsah funkcionalit byl ve spolupráci s klubem realizován přímo autorem (jako zainteresovanou osobou v klubu), neočekává se možnost zásadního odklonění od požadavků. Akceptační test byl proveden v závěru projektu autorem pro ověření správně funkcionality všech komponent.

### 6.4.1 Klientská část

#### Rozcestník

|                 |   |
|-----------------|---|
| Popis           | Rozcestník umožňuje vybrat ze 4 nabízených akcí.  |
| Postup testu    | Uživatel klikne postupně na nabízené možnosti a následně se vrátí zpět na rozcestník tlačítkem zpět v prohlížeči. |
| Správné chování | zobrazí očekávanou podstránku dle popisu tlačítek.  |

#### Přihlášení na akci

|              |  |
|--------------|--|
| Popis        | Formulář pro přihlášení postupně zobrazuje tyto části: <ul style="list-style-type: none"> <li>■ Zadejte e-mail <ul style="list-style-type: none"> <li>■ Pokud jde o registrovaný e-mail, vyzve k zadání hesla</li> <li>■ Pokud jde o neregistrovaný e-mail vyzve pro vytvoření hesla pro případnou editaci přihlášek</li> </ul> </li> <li>■ Vyberte událost</li> <li>■ Vyplňte formulář</li> </ul> |
| Postup testu | Uživatel vyplní všechna pole jako nový uživatel a vytvoří přihlášku. Poté se odhlásí a pokusí se vytvořit novou přihlášku pod stejným účtem.   |

**Správné chování** Aplikace vytvoří dvě nové přihlášky, při první zaregistruje nového uživatele, při druhé vyzve po zadání e-mailu k přihlášení.

### ■ Úprava přihlášky

|                        |  |
|------------------------|--|
| <b>Popis</b>           | Zobrazení všech přihlášek a možnost editace pro neuzavřené přihlášky.  |
| <b>Postup testu</b>    | Uživatel přes úvodní stránku zvolí možnost zobrazit moje přihlášky. Následně se přihlásí a pokusí se upravit přihlášku na akci z předchozího bodu. |
| <b>Správné chování</b> | Aplikace změní přihlášku a odešle notifikační e-mail uživateli a organizátorům akce.   |

### ■ Zapomenuté heslo

|                        |   |
|------------------------|---|
| <b>Popis</b>           | Slouží pro obnovu přístupu k účtu po zapomenutí hesla.  |
| <b>Postup testu</b>    | Uživatel přes úvodní stránku zvolí možnost vytvořit přihlášku, následně vyplní e-mail registrovaného uživatele a bude vyzván k přihlášení. Po vyzvání k zadání hesla klikne na tlačítko zapomněl/a jsem heslo. Poté bude následovat instrukce v e-mailu, které povedou k obnově hesla |
| <b>Správné chování</b> | Aplikace odešle instrukce pro obnovu hesla. v e-mailu bude odkaz obsahující ověřovací token. Po kliknutí na odkaz se uživatel vrátí do fáze, ve které se nacházel při neúspěšném pokusu o přihlášení.   |

### ■ Přihlášení do klubu

|                        |   |
|------------------------|---|
| <b>Popis</b>           | Formulář pro přidání členství postupně zobrazuje tyto části: <ul style="list-style-type: none"> <li>■ Zadejte e-mail           <ul style="list-style-type: none"> <li>■ Pokud jde o registrovaný e-mail, vyzve k zadání hesla</li> <li>■ Pokud jde o neregistrovaný e-mail vyzve pro vytvoření hesla pro případnou editaci přihlášek</li> </ul> </li> <li>■ Vyplňte formulář</li> </ul> |
| <b>Postup testu</b>    | Uživatel vyplní všechna pole jako nový uživatel a vytvoří přihlášku do klubu. Poté se odhlásí a pokusí se vytvořit novou přihlášku do klubu pod stejným účtem.  |
| <b>Správné chování</b> | Aplikace vytvoří dvě nové členství, při prvním zaregistruje nového uživatele, při druhém vyzve po zadání e-mailu k přihlášení.  |

## ■ Úprava členství

|                 |  |
|-----------------|--|
| Popis           | Zobrazení všech členství, stavu zaplacených poplatků a možnost editace adresy a kontaktního e-mailu člena.       |
| Postup testu    | Uživatel přes úvodní stránku zvolí možnost zobrazit členství. Následně se přihlásí a pokusí se upravit členství. |
| Správné chování | Aplikace změní členství a uložení změny informuje notifikací.  |

## ■ 6.4.2 Administrátorská část

Vzhledem k obsáhlosti systému je akceptační test napsán v kondenzované formě. Veškeré části webu obsahují následující funkcionalitu, pokud není uvedeno jinak:

### ■ Operace na entitě a oprávnění

Každá entita umožňuje přidat, zobrazit a upravovat záznamy. Entitou se rozumí typ záznamu, který aplikace umožňuje spravovat (např. uživatel, přihláška, členství, uživatelská role, e-mail, šablona e-mailu, ...). Ke každé entitě rozhraní lze nastavit přístupová práva jednotlivých uživatelů v těchto kategoriích:

- View – zobrazení veškerých entit daného typu
- View\_instance – zobrazení entit souvisejících s uživatelem (např. přihlášky, kde jsem organizátorem, přihlášky, které jsem vytvořil)
- Add – vytvoření nové entity
- Edit – upravení jakékoli entity
- Edit\_instance – upravení entity související s uživatelem (např. přihlášky, kde jsem organizátorem, přihlášky, které jsem vytvořil)

### ■ Zobrazování a filtrování

Výpis entit je realizován tabulkou umožňující stránkování, filtrování a řazení dle vybraných atributů entity.

### ■ Přidávání entit

Formuláře pro přidávání dat obsahují validační pravidla po ověření správnosti formátu dle typu atributu. Po přidání entity se zobrazuje oznámení o úspěšnosti/neúspěšnosti vytvoření záznamu.

### ■ Obecný postup pro testování administrátorské části

- Přihlásí se uživatel s odpovídajícím oprávněním k testované části
  - Pokusí se přidat záznam
  - Pokusí se upravit záznam
  - Pokusí se zobrazit záznamy
- Přihlásí se uživatel, který má oprávnění pouze edit\_instance a nikoli edit
  - Pokusí se upravit záznam kontextově svázan s účtem
  - Pokusí se upravit cizí záznam
- Přihlásí se uživatel, který má pouze oprávnění view
  - Pokusí se zobrazit záznamy
- Přihlásí se uživatel, který má pouze oprávnění view\_instance
  - Pokusí se zobrazit záznam kontextově svázan s účtem
  - Pokusí se zobrazit cizí záznam

### ■ Seznam entit testovaných dle obecného postupu

V závorce jsou uvedeny atributy, které lze zobrazit/upravit dle obecného postupu.

- Uživatelé (e-mail, příjmení, jméno, telefon, role, heslo)
- Uživatelské role (název, seznam oprávnění)
- E-maily (od, pro, předmět, obsah, datum odeslání, šablona)
- Šablony e-mailů (předmět, obsah)
- Členská základna (jméno, příjmení, datum narození, typ členství, e-mail, ulice, číslo, psč)
- Akce (název, ročník, organizátoři, registrace od, registrace do, zobrazovat, trvání od, trvání do, místo, cena)
- Typ akce (název, výchozí formulář)
- Přihlašovací formulář k akcím (název, seznam polí)
- Šablona pro dokumenty (název, typ, obsah)
- Platby (datum, částka, VS, KS, SS, typ transakce, název protiúctu, protiúčet, poznámka, párováno)

## ■ Doplnkové informace k entitám

### Uživatelé

Roli u uživatelů mohou měnit pouze uživatelé mající oprávnění edit u rolí.

### Uživatelské role

Nepoužívají oprávnění edit\_instance. Lze mazat role.

### Přihlašovací formulář k akcím

Z důvodu konsistence dat lze upravovat jen formuláře, které nejsou aktuálně používány k jiným akcím. Z toho důvodu je možnost vytvořit nový formulář kopií. Při vytváření formuláře lze vytvořit tyto typy polí a prvků: textové pole, e-mail, telefonní číslo, číslo, Výběr, výběr z více položek, dlouhý text, nadpis, popisek, datum, zaškrtačací políčko, více hodnot (tagy) a více e-mailů.

Formulářové prvky lze řadit a lze je označovat jako povinná pole. U jednotlivých prvků lze nastavit, zda se mají zobrazovat ve výpisu přihlášek.

### Šablony e-mailů

Při editaci šablony se automaticky zobrazuje seznam použitelných kontextových dat. K šablonám e-mailu je možné připojit přílohu buď nahráním souboru, nebo zvolením jiné šablony dokumentu určené pro stejný model.

### Šablony dokumentů

Pokud byl před vytvářením šablony učiněn výběr dat, je možné šablonu pro kontrolu ihned vygenerovat. Vygenerovaný soubor lze před uložením dodatečně modifikovat. Při editaci šablony se automaticky zobrazuje seznam použitelných kontextových dat.



# Kapitola 7

## Závěr

V rámci projektu vznikl informační systém s robustním základem, který lze v budoucnu snadno rozšiřovat o další funkcionality. Projekt je nyní ve fázi zkušebního provozu, kdy je již klubem užíván. Postupně jsou doplňovány funkce, které práci se systémem zrychlují a usnadňují ji.

Aplikace je přístupná na adrese <https://shm.cableek.com>, kterou zdarma včetně webhostingu poskytl autor projektu nejméně do roku 2020. Během zkušebního provozu první sezónu nedošlo k větším problémům. v aplikaci je nyní registrováno 185 uživatelů, je evidována členská základna o velikosti 144 aktivních členů, kteří zaplatili členský příspěvek. Prostřednictvím systému bylo vytvořeno v první táborevé sezóně 206 přihlášek. Skrze aplikaci bylo rozesláno přes 2000 informačních e-mailů.

Největším přínosem projektu pro autora byla přímá spolupráce s budoucími reálnými uživateli aplikace a řešení nastalých komplikací při ovládání aplikace. Tato zkušenost může výrazně pomoci při návrhu veškerých dalších softwarů. Ne vždy totiž člověk se znalostí IT technologií dokáže správně odhadnout chování běžného uživatele.

### 7.1 Budoucnost projektu

Struktura projektu byla koncipována tak, aby na jejím základě bylo do budoucna možné vystavět jakoukoli webovou aplikaci. Je tedy možné jí pohodlně dále rozšiřovat o další funkce. První funkcí, která do systému přibude, bude evidence majetku klubu a možnost rezervace zápujčky členskou základnou. Záměrem autora je informační systém dále integrovat především s oficiálním webem klubu a oficiálním facebookovým profilem, aby byla aplikace pro uživatele co nejdostupnější. Pokud se v rámci klubu systém osvědčí, bude možné systém nabídnout také dalším klubům Salesiánského Hnutí Mládeže, potažmo dalším táborevým organizacím.





# Příloha A

## Použité zkratky

|         |  |
|---------|--|
| CRUD    | Create Read Update Delete                                    |
| CSRF    | Cross-site Request Forgery                                   |
| CSS     | Cascading Style Sheets                                       |
| CTU     | Czech Technical University                                   |
| ČVUT    | České Vysoké Učení Technické                                 |
| FEL     | Fakulta elektrotechnická                                     |
| HTML    | HyperText Markup Language                                    |
| HTTP    | Hypertext Transfer Protocol                                  |
| HTTPS   | Hypertext Transfer Protocol Secure                           |
| IDE     | Integrated development environment                           |
| MHMP    | Magistrát hlavního města Prahy                               |
| MŠMT    | Ministerstvo školství, mládeže a tělovýchovy České republiky |
| MVC     | Model view controller  |
| ORM     | Object-relational mapping                                    |
| PHP     | Personal Home Page   |
| URL     | Uniform Resource Locator                                     |
| WYSIWYG | What You See Is What You Get                                 |
| XSS     | Cross-site scripting   |



## Příloha B

### Literatura

- [1] Veřejný rejstřík a Sbírka listin. Úplný výpis ze spolkového rejstříku. Justice.cz [online]. Praha: Ministerstvo spravedlnosti České republiky, ©2016 [cit. 2017-11-22]. Dostupné z: <https://or.justice.cz/ias/ui/rejstrik-firma.vysledky?subjektId=739630&typ=UPLNY>.
- [2] Salesiánskéhnutí mládeže z. s. Co děláme. www.shm.cz [online]. [cit. 2017-11-22]. Dostupné z: <http://www.shm.cz/co-delame/>.
- [3] Michaela Tureckiová. *Klíč k účinnému vedení lidí - odemkněte potenciál svých spolupracovníků*. Grada Publishing a.s., 2007.
- [4] Ian Sommerville. *Softwarové inženýrství*. ALBATROS MEDIA A.S., BRANCH BRNO, 2013.
- [5] Bruckner Tomáš, Voříšek Jiří, Buchalceková Alena a kolektiv. *Tvorba informačních systémů: Principy, metodiky, architektury*. Grada Publishing a.s., 2012.
- [6] PHP Group. Webhosting. PHP: Hypertext Preprocessor[online]. [cit. 2017-11-24]. Dostupné z: <http://php.net/>.
- [7] PUEBLO s.r.o. Webhosting. hostingy.cz [online]. [cit. 2017-11-24]. Dostupné z: <http://www.hostingy.cz/webhosting.html>.
- [8] Lukáš Košárek. Convexity. *Masarykova univerzita, Fakulta informatiky*, 2010. [cit. 2017-11-24]. Dostupné z: [https://is.muni.cz/th/256433/fi\\_b\\_b1/Vykonnostni\\_srovnani\\_relacnich\\_databazi.pdf](https://is.muni.cz/th/256433/fi_b_b1/Vykonnostni_srovnani_relacnich_databazi.pdf).
- [9] Mark Otto Jacob Thornton. Bootstrap, 2017. [online]. [cit. 2017-11-28]. Dostupné z: <http://getbootstrap.com/>.
- [10] The jQuery Foundation. jquery, 2017. [online]. [cit. 2017-11-29]. Dostupné z: <https://jquery.com/>.
- [11] JetBrains. IntelliJ idea: The java ide for professional developers by jetbrains, 2017. [online]. [cit. 2017-11-25]. Dostupné z: <https://www.jetbrains.com/idea/>.

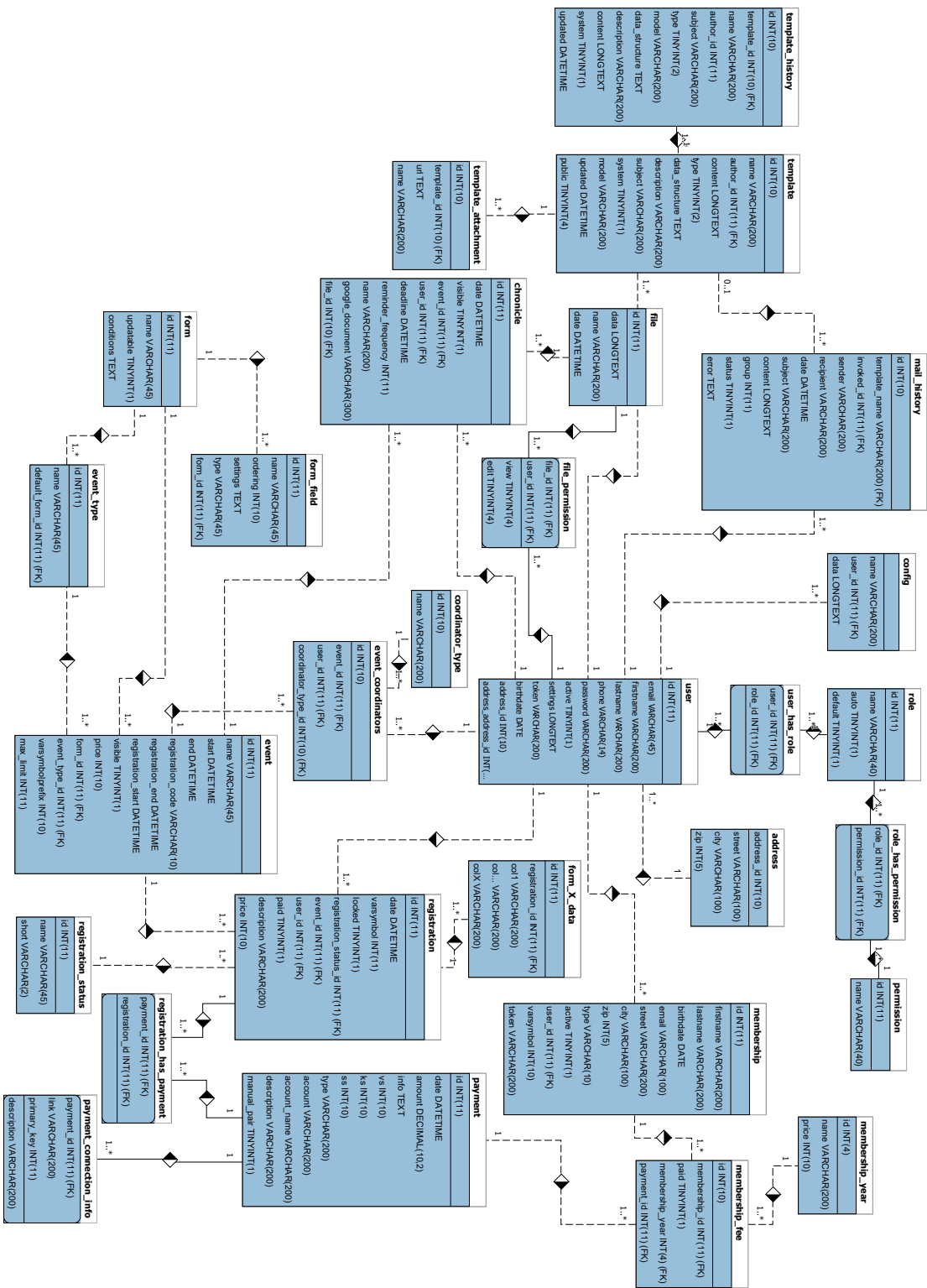


- [28] Dušan Janovský. Html editory, 2017. [online]. [cit. 2017-11-29]. Dostupné z: <https://www.jakpsatweb.cz/editory.html#wysiwyg>.
- [29] Ephox Corporation. Tiny mce, 2017. [online]. [cit. 2017-11-29]. Dostupné z: <https://www.tinymce.com/>.
- [30] Selenium Contributors. Selenium - web browser automation, 2017. [online]. [cit. 2017-12-10]. Dostupné z: <http://www.seleniumhq.org/>.
- [31] H1.cz s.r.o. Uživatelské testování, 2015. [online]. [cit. 2017-11-29]. Dostupné z: <http://www.h1.cz/pouzitelnost>.
- [32] Nette Foundation. Zabezpečení před zranitelnostmi, 2017. [online]. [cit. 2017-11-29]. Dostupné z: <https://doc.nette.org/cs/2.4/vulnerability-protection>.



# Příloha C

## Kompletní schéma databáze







# Příloha D

## Uživatelská příručka

### D.1 Softwarové požadavky

Pro běh aplikace je třeba mít k dispozici Apache HTTP Server s podporou PHP a MySQL databázi. Doporučená konfigurace je následující:

|                        |        |
|------------------------|--------|
| PHP verze              | 135    |
| MySQL innodb_version   | 5.6.17 |
| MySQL protocol_version | 10     |

### D.2 Instalace aplikace

1. Zkopírujte obsah složky **src** na server.
2. v souboru **app/config/config.local.neon** vyplňte údaje pro připojení na **MySQL server** a údaje pro odesílání e-mailů přes **SMTP server**.
3. v souboru **app/config/config.neon** vyplňte **superuser e-mail**, který bude výchozím účtem v aplikaci a bude mít automaticky všechna práva.
4. Importujte soubor **app/config/default.sql** na do MySQL databáze.
5. Spustíte server a běžte na adresu **[adresa\_serveru]/admin** pro zadání hesla administrátorského účtu.

**Upozornění:** Složky *app/*, *temp/* a *log/* nesmí být přímo přístupné přes http, jinak hrozí napadení serveru a získání citlivých údajů!

## D.3 Uživatelská část

Hlavní strana uživatelské části je tvořena rozcestníkem. v případě přihlášení uživatele se v horní části zobrazí pruh s e-mailem uživatele a tlačítko pro odhlášení. Přihlášení se provádí až v případě interakce s aplikací (viz obr. D.2).



Obrázek D.1: Výchozí rozcestník uživatelské části

1. přihlášení na akce pořádané klubem
2. zobrazení aktuálních přihlášek, informace pro zaplacení, možnost úpravy přihlášek
3. podání přihlášky do klubu
4. správa členů klubu (obvykle členové rodiny pod jedním účtem)



Obrázek D.2: Postup přihlášení na akci s registrací uživatele

## D.4 Administrátorská část

The screenshot shows the 'SHM přehled' (SHM Overview) page. It features a dark sidebar menu on the left with the 'Orgament' logo and various navigation options. The main content area is divided into several sections:

- 1**: A button in the top right corner of the main content area, used to toggle the main menu.
- 2**: A user profile dropdown menu in the top right corner, showing the user's name and email.
- 3**: A link in the top right corner that leads to the user management section.
- 4**: The main sidebar menu on the left, containing options like 'Přehled', 'Údálosti', 'Kalendáře', 'Kronika', 'Členská základna', 'Banka', 'Soubory', 'Uživatelé', 'E-mail', and 'Nastavení aplikace'.
- 5**: A section titled 'Důležité soubory pro vedoucí' (Important files for leaders) which lists documents such as '17\_Cesták.xls', '17\_Náhrady za cestovné.doc', '17\_Manual\_pro\_taborove\_vedouci.doc', and '17\_Smlouva\_o\_jednorazovem\_poskytnuti\_objektu.doc'.

Below the screenshot, the text '2017 © Jindřich Regál' is visible.

Obrázek D.3: Výchozí obrazovka administrace

1. tlačítko pro skrytí hlavního menu
2. nastavení detailu uživatele
3. odkaz na uživatelskou část
4. hlavní menu
5. důležité soubory pro koordinátory ve speciální složce na Google Disku klubu

The screenshot shows the 'Seznam událostí' (Event List) page in the Orgament administration system. The interface is dark-themed with a sidebar on the left containing navigation links like 'Přehled', 'Údálosti', 'Kalendáře', 'Kronika', 'Členská základna', 'Banka', 'Soubory', 'Uživatelé', 'E-maily', and 'Nastavení aplikace'. The main area features search filters for 'VS', 'Viditelná', 'Konec', 'Název', 'Začátek', and 'Konec', along with a 'Filtrovat...' dropdown. A table of events is displayed with columns for 'VS', 'Název', 'Účastníci', 'Viditelná', 'Začátek', 'Konec', 'Registrace od', 'Registrace do', 'Cena', 'Formulář', 'Typ', 'Reg. kód', and 'Akce'. The table contains several rows of event data, including 'Testovací tábor' and 'Celoroční aktivita'. A 'Hromadné akce:' dropdown is positioned above the table. At the bottom, there is a pagination control showing '(Položky: 0 - 7 z 7)' and a dropdown for the number of items per page, currently set to 20.

Obrázek D.4: Obvyklý vzhled tabulkového výpisu v administraci

1. navigační lišta
2. název aktuálního pohledu
3. prostor pro tlačítka (obvykle vytváření entit a specifické operace pohledu)
4. zobrazení/skrytí velkého filtru
5. export filtrovaných záznamů do csv pro otevření např. v tabulkovém editoru Excel
6. výběr akce nad řádkem podporující více řádků najednou
7. nejdůležitější akce pro daný řádek
8. zaškrtačací pole pro výběr více záznamů k hromadné akci
9. okno pro zobrazení dodatečných informací o poli v tabulce
10. výběr počtu záznamů k zobrazení najednou



## Příloha E

### Vyjádření předsedy klubu Mgr. Pavla Heriana

Klub SHM Praha - Počernice používal k přihlášení na tábory jednoduchý, narychlo vytvořený systém z roku 2012, který postupem času přestával splňovat naše požadavky. Všichni vedoucí měli přístup k osobním údajům všech dětí, navíc úpravy v přihláškách, přesuny dětí na jiné termíny atd. mohl provádět pouze správce systému. Po roce 2015, kdy navíc do naší správy přešli i tábory klubu SHM Karlín, bylo jasné, že systém bude třeba modernizovat. Proto jsem velice uvítal nabídku Jindřicha Regála na realizaci systému v rámci jeho bakalářské práce. Jako dlouholetý aktivní dobrovolník naší organizace má přehled o fungování klubu, což výrazně usnadnilo práci při stanovení požadavků na nový systém. Dílčí problematiku řešil přímo s dotčenými zástupci klubu, což se ukázalo jako velice efektivní.

Nový systém byl v základní podobě zprovozněn již pro přihlášení na tábory v roce 2017, což sebou neslo drobné zpoždění spuštění přihlašování. Ze strany rodičů jsme ale nezaznamenali při přihlašování žádný problém a systém se tedy osvědčil. Rovněž členské příspěvky se podařilo prostřednictvím nového systému vybrat bez větších obtíží.

Rozhraní pro rodiče a členy klubu je přehledné a jsem s ním spokojen. Vítám možnost zažádání o členství online, protože řada lidí dosud o možnosti stát se členem a podpořit tak činnost klubu dosud nevěděla.

Administrace vypadá na první pohled složitě, nicméně systém obsahuje řadu nových funkcí, což se muselo na vzhledu administrace projevit. I když v současné době systém „pouze“ nahradil původní systém a zautomatizoval správu členů, do budoucna je skvěle připraven pro rozšíření, které plánujeme. Jedná se zejména o možnost rezervace zápujček vybavení pro naše členy, maximální zjednodušení vedení (u vedoucích neoblíbené) kroniky a její zpřístupnění veřejnosti online.

V Praze dne 15. 12. 2017

 salesiánské  
bratři mládeže

SHM Klub Praha - Počernice IČO: 250 07 00  
Městská 1000/15 TEL: +420 731 600 074  
193 AG Praha 9 WWW: [www.pocerniceshm.cz](http://www.pocerniceshm.cz)

