

Studentský klub FIT++, sdružující aktivní studenty Fakulty informačních technologií ČVUT v Praze, by rád vytvořil nový web s rozhráním pro prezentaci své činnosti a rozhráním pro správu uživatelů, akcí a komunikací se svými členy.

Cíl práce:

- Analyzujte aktuální web klubu FIT++.
- Na základě komunikace s členy klubu analyzujte jejich nároky na web.
- Navrhněte architekturu a funkce webu tak, aby splňovala požadavky klienta.
- Web naimplementujte (frontend i backend).
- Proveďte nasazení a uživatelské testy webu.





**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

Bakalářská práce

**Web studentského klubu FIT++**

*Martin Pilný*

Katedra softwarového inženýrství  
Vedoucí práce: Ing. Tomáš Nováček

27. června 2021



---

## Poděkování

Chci poděkovat své rodině a přátelům za podporu a Tomášovi, že mi byl skvělým vedoucím.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, 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 27. června 2021

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2021 Martin Pilný. 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 a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Pilný, Martin. *Web studentského klubu FIT++*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.



---

# Abstrakt

Problematika tvorby webových aplikací na míru klientovi je tu stále. V rámci této práce je klientem studentský klub FIT++, který sdružuje aktivní studenty na fakultě informačních technologií ČVUT. V rámci činnosti klubu FIT++ se pravidelně řeší přijímání nových členů, organizování akcí či volby svých představitelů. Čtenář se tak může dozvědět více o tom, jakým způsobem provádět analýzu požadavků, pokud není klientem jeden člověk, ale celý klub lidí. Práce dále rozebírá problematiku online voleb a porovnání nástrojů, které by tento problém mohli lépe řešit. V návrhu řešení pak práce přináší možné přístupy k řešení několika požadavků členů klubu jako je například organizace akcí či proces registrace nového člena. Výstupem implementační části práce je aplikace s napojením na databázi, veřejnou částí pro prezentaci klubu a soukromou částí pro administraci. Práce dále vysvětluje postupy při řešení požadavků klienta, které mohou být inspirací pro podobné projekty. V poslední části je pak rozebráno uživatelské testování aplikace společně s jeho vyhodnocením.

**Klíčová slova** studentský klub, FIT++, analýza požadavků dotazníkovým šetřením, webová aplikace, Symfony, online volby, uživatelské testování

# Abstract

The issue of creating web applications tailored to the client's needs is still there. In this thesis, the client is the student' club FIT++, which regularly deals with admission of new members, organizing events and the election of their representatives. The reader can learn more about how to analyze requirements, unless the client is one person, but the whole club people. The thesis also analyzes the online elections and a comparison of tools that could better address this issue. In the design of the solution, the thesis brings possible approaches to solving several requirements of club members, such as the organization of events or registration process new member. Outcome of the implementation part of the thesis is an application with a connection to database, a public part for the presentation of the club and a private part for administration. The thesis then explains solution procedures of client requirements that could be inspiration for similar projects. In the last part is discussed user testing of application along with its evaluation.

**Keywords** students' club, FIT++, analysis of requirements by questionnaire survey, web application, Symfony, online elections, user testing

---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza</b>	<b>5</b>
2.1 Analýza současného řešení . . . . .	5
2.2 Analýza požadavků členů FIT++ . . . . .	7
2.3 Analýza podobných projektů . . . . .	12
2.3.1 International student club (isc.cvut.cz) . . . . .	12
2.3.2 +/- FEL (pm.fel.cvut.cz) . . . . .	13
2.3.3 ŠTUK (stuk.fsv.cvut.cz) . . . . .	14
2.3.4 eFDrive (efd.su.cvut.cz) . . . . .	14
2.4 Analýza webových technologií . . . . .	15
2.4.1 Konfigurace prostředí webové aplikace . . . . .	16
2.4.2 Webový framework . . . . .	16
2.4.3 Nástroje pro automatické testování a nasazení . . . . .	17
2.5 Analýza řešení pro online volby . . . . .	17
2.5.1 Google Forms [22] . . . . .	18
2.5.2 SurveyMonkey [24] . . . . .	18
2.5.3 LimeSurvey [26] . . . . .	19
<b>3 Návrh</b>	<b>21</b>
3.1 Databázový model . . . . .	21
3.1.1 User . . . . .	21
3.1.2 Article . . . . .	22
3.1.3 Event . . . . .	22
3.1.4 Photo a Album . . . . .	22
3.2 Use case diagramy . . . . .	22
3.2.1 Správa uživatelů . . . . .	22

3.2.2	Online volby . . . . .	24
3.2.3	Organizace akcí . . . . .	24
3.3	Procesy . . . . .	26
3.3.1	Registrace nového člena klubu . . . . .	27
3.3.2	Online volby . . . . .	27
3.4	Výběr nástrojů a technologií . . . . .	29
<b>4</b>	<b>Realizace</b>	<b>31</b>
4.1	Docker kontejnery . . . . .	31
4.2	Symfony aplikace . . . . .	32
4.2.1	Model-View-Controller . . . . .	33
4.2.2	EasyAdmin . . . . .	33
4.2.3	Správa členů . . . . .	34
4.2.4	Organizace akcí . . . . .	38
4.2.5	Galerie . . . . .	40
4.2.6	Aktuality . . . . .	43
4.3	Online volby . . . . .	44
4.4	Nasazení aplikace . . . . .	44
4.5	Testování aplikace . . . . .	44
4.5.1	Uživatelské testování . . . . .	45
4.5.2	Automatické testování . . . . .	47
	<b>Závěr</b>	<b>49</b>
	<b>Bibliografie</b>	<b>51</b>
	<b>A Seznam použitých zkratk</b>	<b>57</b>
	<b>B Obsah příloženého média</b>	<b>59</b>

---

## Seznam obrázků

2.1	Hlavní stránka webu FIT++ . . . . .	6
2.2	Tabulka s členy klubu pro registrátory a adminy . . . . .	6
2.3	Průchod dotazníkem znázorněný grafem . . . . .	8
2.4	Graf otázky na funkce veřejné části webu . . . . .	9
2.5	Graf otázky na funkce webu pro členy . . . . .	12
2.6	Stránka s kontaktními informacemi na webu ISC . . . . .	13
2.7	Stránka s novinkami na webu +/- FEL . . . . .	14
2.8	Stránka s akcemi na webu ŠTUK . . . . .	15
2.9	Stránka s dokumenty na webu eFDrive . . . . .	16
2.10	Nastavení Google formuláře v rámci organizace FIT ČVUT . . . . .	19
3.1	Databázový model . . . . .	23
3.2	Use case diagram správy uživatelů . . . . .	25
3.3	Use case diagram online voleb . . . . .	25
3.4	Use case diagram správy akcí a jejich organizace . . . . .	26
3.5	Activity diagram registrace nového člena . . . . .	28
3.6	Activity diagram online voleb . . . . .	29
4.1	Úvodní obrazovka administrace . . . . .	35
4.2	Seznam členů v EasyAdmin administraci z pohledu administrátora . . . . .	36
4.3	Detail zatím neschváleného člena z pohledu registrátora . . . . .	37
4.4	Detail události v administraci . . . . .	39
4.5	Fotografie zarovnané knihovnou <i>justfiedGallery</i> . . . . .	41
4.6	Editace článku v administraci . . . . .	43
4.7	Detail průzkumu v nástroji LimeSurvey . . . . .	45



---

# Seznam tabulek

2.1 Průměry a mediány hodnocení jednotlivých funkcí, vedení klubu .	11
---	----





---

# Úvod

Studenti vysokých škol se většinou mimo studia věnují i mnoha mimoškolním aktivitám. Jednou z takových aktivit může být i činnost v některém ze studentských klubů. Mezi ně se řadí i studentský klub FIT++, který sdružuje aktivní studenty Fakulty informačních technologií ČVUT v Praze a je součástí Studentské Unie ČVUT. Mezi jeho hlavní činnosti patří organizace nejrůznějších akcí, jak ve spolupráci s fakultou, tak i akce vlastní. Klub své aktivity sdílí především na sociálních sítích, nicméně má i vlastní web. V poslední době ovšem v klubu vzrostla poptávka po novém webu, který bude lépe reflektovat požadavky od současných členů a bude lépe reprezentovat klub a jeho činnost.

Cílovou skupinou pro tuto práci jsou především studentské kluby či jiné spolky, které se zaměřují na organizaci akcí a zvažují zavedení vlastního webu či přechod na web nový. Důvodů pro takové rozhodnutí může být mnoho. Tyto organizace možná chtějí (nebo jim dokonce zákon ukládá za povinnost) mít přehled o svých členech, svém majetku či svých financích a dosud se tím nezabývaly. Dalším důvodem může být, že počet členů vzrostl natolik, že pro vedení organizace je neudržitelné spravovat klub dosavadními prostředky anebo má spolek velmi specifické požadavky na svou správu a řízení, které současná nabídka služeb a nástrojů nemůže uspokojit.

V teoretické části je práce zaměřena na analýzu skládající se z několika částí a také výběrem nástrojů a webových frameworků. Analyzuje se zde současné řešení webu FIT++, požadavky členské základny na nový web a také zvolená řešení u webů studentských klubů, které mají podobné charakteristiky jako FIT++. Na základě poznatků získaných z této analýzy je proveden výběr nástrojů a frameworků pro vytvoření nového webu.

Velká část praktické části práce se věnuje implementaci webové aplikace. V první fázi jde o návrh databáze, případů užití a procesů. Následuje implementace backendu a frontendu pomocí nástrojů a frameworků zvolených v teoretické části. Součástí jsou i zmínky o dalších cestách, kterými se může implementace ubírat. Dále se zde řeší nasazení aplikace na server a výhledově

## ÚVOD

---

zavedení CI/CD. Poslední věcí, kterou se praktická část zabývá, je uživatelské a částečně i automatické testování.

---

## Cíl práce

Cílem práce je navrhnout a vytvořit webovou práci pro fakultní klub FIT++, který sdružuje aktivní studenty Fakulty informačních technologií ČVUT. Web by měl obsahovat rozhraní pro prezentaci své činnosti a rozhraní pro správu uživatelů, akcí a komunikaci se členy klubu.

V teoretické části práce je jedním z úkolů vypracovat analýzu současného řešení webu a nároky členů klubu na web. Na základě výstupů z této analýzy pak bude nutné vybrat nástroje pro implementaci (frameworky, druh databáze, apod.), navrhnout architekturu a funkce webu tak, aby splňovala požadavky klienta.

Hlavní cíl praktické části je implementace frontendu a backendu webové aplikace na základě návrhu, který vychází z analýzy. Dále je potřeba provést nasazení webové aplikace na server. Nakonec je třeba aplikaci otestovat v rámci uživatelského testování.



---

## Analýza

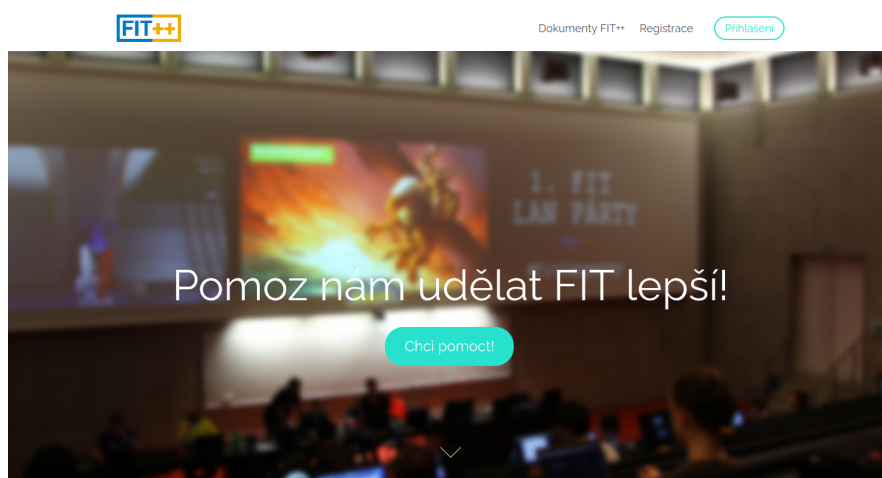
V této kapitole se práce zabývá analýzou, která se skládá z několika částí. První část se zabývá současným webem klubu a jeho zpracováním. V další části práce sbírá a vyhodnocuje požadavky členů FIT++ na nový web. Využilo se zde především dotazníkového šetření a osobní komunikace s členy klubu. Následující část zaměřuje pozornost na projekty podobného typu, tedy webové aplikace pro studentské kluby či spolky. Poté se práce ubírá k analýze webových technologií, které by mohly být použity v rámci práce. Poslední část se pak věnuje definováním problémů online voleb, které vzešly jako požadavek z analýzy požadavků, a možným nástrojům, se kterými by tento problém mohl být vyřešen.

### 2.1 Analýza současného řešení

Návštěvníkovi webu se po příchodu na adresu <https://klubfitpp.cz> zobrazí hlavní stránka, která obsahuje základní informace o klubu, kontaktní informace nebo odkazy na weby, které by mohly případného návštěvníka zajímat. Je zde také navigační menu, ze kterého se lze dostat na stránku s dokumenty, stránku s registračním formulářem nebo stránku s přihlášením do aplikace. Stránka s dokumenty FIT++ má, podobně jako hlavní stránka, jednoduchý design a obsahuje několik pod sebou vložených seznamů s odkazy na jednotlivé dokumenty. Stránky s registračním a přihlašovacím formulářem jsou opět velmi jednoduché, ovšem designově se liší od ostatních, veřejně přístupných stránek, zejména použitou barevnou paletou nebo zvoleným fontem.

Po vyplnění registračního formuláře je uživateli na jím zadanou emailovou adresu odeslán email s dalšími informacemi ohledně dokončení registrace. Následně po přihlášení do aplikace se běžnému uživateli zobrazí stránka s navigačním menu, odkud je možné se dostat na stránku editace profilu nebo se odhlásit. Zde si může uživatel změnit své heslo i veškeré údaje, které zadal během registrace. V případě, že uživatel zapomene své heslo, může si na

## 2. ANALÝZA



Obrázek 2.1: Hlavní stránka webu FIT++

stránce s přihlašovacím formulářem nechat vygenerovat nové heslo, pokud zná své přihlašovací jméno. Po zadání přihlašovacího jména bude na přiřazenou emailovou adresu odeslán email obsahující nově vygenerované heslo.

ID	Příjmení	Jméno	E-mail	Login	Telefon	Datum narození	Pomoc PR	Aktivní	Podpis	Akce
1							ANO	ANO	ANO	Deaktivovat   UDP
2							ANO	ANO	ANO	Deaktivovat   UDP
3							ANO	ANO	ANO	Deaktivovat   UDP
5							ANO	ANO	ANO	Deaktivovat   UDP
6							ANO	ANO	ANO	Deaktivovat   UDP
8							ANO	ANO	ANO	Deaktivovat   UDP
15							ANO	ANO	ANO	Deaktivovat   UDP
20							ANO	ANO	ANO	Deaktivovat   UDP
22							NE	ANO	ANO	Deaktivovat   UDP
24							ANO	ANO	ANO	Deaktivovat   UDP
27							ANO	ANO	ANO	Deaktivovat   UDP

Obrázek 2.2: Tabulka s členy klubu pro registrátory a adminy

Pokud má uživatel vyšší práva v aplikaci, tak se mu zobrazí i tabulka se členy klubu, skrze kterou je možné spravovat některé atributy člena, vygenerovat registrační formulář pro vybrané členy nebo odeslat vybraným členům hromadný email. Design stránek administrace je stejný jako u stránek s registračním a přihlašovacím formulářem.

Co se implementace týče, jedná se jednoduchou webovou aplikaci, která využívá framework Nette pro backend aplikace, šablonovací engine Latte a několika dalších pomocných knihoven např. phinx pro snadnější databázové mi-

grace nebo tracy pro ladění chyb. Celá aplikace je v modelu MVP, takže ve zdrojových kódech aplikace lze mimo jiné najít entitní třídy, různé presentery, a také třeba Latte šablony.

Aplikace běží v několika docker kontejnerech, které se vzájemně propojí pomocí nástroje Docker Compose [1]. V základním nastavení se při spuštění aplikace vytvoří následující kontejnery:

**nginx**    webová aplikace

**adminer** nástroj pro přístup do databáze

**mailhog** nástroj pro posílání e-mailů

Databáze obsahuje jedinou entitu, a tou je User, která obsahuje informace o členovi jako např. jeho přihlašovací údaje, adresu bydliště či příznaky, zda je aktivní nebo zda chce pomáhat na PR akcích fakulty. Uživatelská role, podle které se zpřístupňují dostupné funkce webu, není samostatnou entitou, ale je textovým atributem entity User.

## 2.2 Analýza požadavků členů FIT++

Tato analýza má za úkol zjistit, jak členové klubu hodnotí současný web a jaké funkcionality by očekávali od webu nového. K tomu se využilo online dotazníku, který byl zveřejněn členům klubu 19. října 2020 a uzavřen 25. října 2020. K vyhotovení formuláře se využilo aplikace Google Forms. Dotazník obsahuje jak kvantitativní, tak kvalitativní druhy otázek. Kvantitativními otázkami rozumíme ty, kde odpovědí je číselná hodnota či více číselných hodnot, které většinou vyjadřují hodnocení na škále či řazení a lze z nich vytěžit nějaké znalosti pomocí statistických metod. Na druhou stranu kvalitativními označujeme otázky, které nabízejí respondentovi odpovědět písemně, a tak lépe vyjádřit své myšlenky. Tyto otázky mají ovšem tu nevýhodu, že se obtížně zpracovávají strojově. Z toho důvodu dotazník kombinuje tyto dva druhy otázek, které se tak navzájem doplňují. Většina kvantitativních otázek v dotazníku jsou totiž doplněné o jednu či více kvalitativní otázku, která umožňuje respondentovi upřesnit své předchozí odpovědi v otázkách kvantitativních. Je pochopitelné, že ne vždy bude mít respondent odpověď na kvalitativní otázku, a proto je tento typ otázek nastaven jako nepovinný. Na druhou stranu kvantitativní jsou povinné, aby měl dotazník alespoň nějakou vypovídající hodnotu.

Struktura formuláře je rozdělena na několik sekcí, které se specializují na určité druhy otázek. Názvy sekcí byly zvoleny následovně:

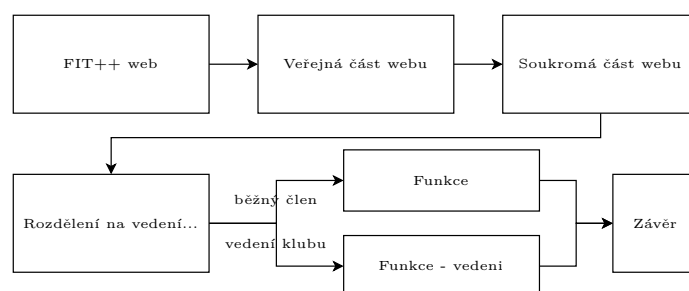
- FIT++ web,
- Veřejná část webu,
- Soukromá část webu,

## 2. ANALÝZA

---

- Rozdělení na vedení klubu a běžné členy,
- Funkcionality – vedení,
- Funkcionality,
- Závěr.

V dotazníku je mimo jiné použito i větvení na základě odpovědi v sekci Rozdělení na vedení klubu a běžné členy. Zde se dotazník rozděluje zvláště pro členy klubu, kteří jsou nebo dřív byli součástí představenstva, a zvláště pro členy, kteří žádnou pozici ve vedení klubu nezastávali.



Obrázek 2.3: Průchod dotazníkem znázorněný grafem

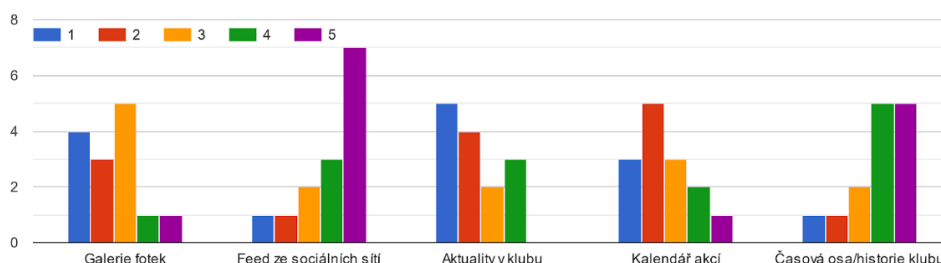
Zde práce rozebírá jednotlivé otázky dotazníku a odpovědi od respondentů. Celkově dotazník vyplnilo 14 respondentů, z toho 6 uvedlo, že bylo nebo je součástí vedení klubu. Zbývajících 8 respondentů uvedlo, že jsou bez současného či minulého působení ve vedení.

Sekce FIT++ web je úvodní sekcí do celého dotazníku. Zde je pouze jediná otázka, která zjišťuje, jak často respondenti navštěvují současný web. Polovina respondentů uvedlo, že web navštěvuje alespoň jednou za půl roku a necelých 43 % alespoň jednou za měsíc. Jeden respondent uvedl, že web za poslední půl rok nenavštívil. Otázka obsahovala i odpověď popisující návštěvnost alespoň jednou týdně, kterou ovšem žádný z respondentů ne zvolil.

Další sekce se zabývá veřejnou částí současného webu, tedy stránkami <https://klubfitpp.cz> a <https://klubfitpp.cz/informace>. První otázka zde je opět kvantitativní a ptá se na to, jak by respondenti ohodnotili současnou veřejnou část webu po obsahové stránce, a to na škále od 1–4, kde 1 je výborné hodnocení a 4 je nedostatečné. Zde stojí za zmínku, že stejná stupnice se využívá i v dalších podobných otázkách v dotazníku, které zjišťují nějaké hodnocení. Výsledky se kloní spíše do nedostatečné části na škále. Následující otázka se doptává respondenta na jeho vysvětlení, proč hodnotil právě takto. Většina odpovědí zmiňuje nedostatek obsahu, zejména by požadovala detailnější popis klubu a klubové činnosti, kalendář akcí či galerii fotek. Opět zde je záhodno zmínit, že tento druh otázky se vyskytuje po každé následující otázce v dotazníku, která se zabývá hodnocením na číselné škále. Třetí



Které z následujících prvků by podle Vás měly být součástí veřejné části webu?



Obrázek 2.4: Graf otázky na funkce veřejné části webu

otázka v této sekci se ptá na hodnocení nefunkčních vlastností veřejné části webu jako např. UX či rychlost načítání obsahu. Hodnocení je v této otázce spíše pozitivní. Všichni respondenti, kteří odpověděli na zdůvodňující otázku, uvedli, že rychlost webu je v pořádku, ale rozcházeli se v hodnocení UX, kde necelá třetina hodnotila UX jako dobré a zhruba polovina k němu měla větší či menší výhrady. Objevila se zde i odpověď, že UX nelze dobře ohodnotit, protože na webu není příliš obsahu. Poslední skupina otázek se zaměřuje na možné webové prvky, které by mohly být součástí webu. Jsou zde zmíněné následující prvky:

- Galerie fotek,
- Feed ze sociálních sítí,
- Aktuality v klubu,
- Kalendář akcí,
- Časová osa / historie klubu.

Respondenti v této otázce odpovídali způsobem, že bylo zapotřebí seřadit výše zmíněné prvky podle důležitosti. Nejlépe byly ohodnoceny prvky Galerie fotek, Aktuality v klubu a Kalendář akcí. Za nejméně důležité považují respondenti prvky Feed ze sociálních sítí a Časovou osu / historii klubu. Poslední dvě otázky se opět doptávají, ale už ne na odůvodnění proč respondenti hodnotili tímto způsobem, ale jestli jim připadá nějaký prvek z výše zmíněných zbytečný a jestli je napadá nějaký jiný prvek, který by měl být součástí veřejné části webu a nebyl zmíněn. Z celkem 7 odpovědí na první z těchto dvou otázek se objevují, s velmi podobnou četností, tři:

- respondent považuje za zbytečný feed ze sociálních sítí,

## 2. ANALÝZA

---

- respondent považuje za zbytečnou časovou osu / historii klubu,
- respondent považuje všechny zmíněné prvky za velmi důležité.

Z toho vyplývá, že by i na tyto prvky bylo vhodné myslet při návrhu. Nejedná se ovšem o nikterak složité prvky a při dobrém návrhu aplikace by mohlo být snadné tyto prvky doimplementovat bez větších zásahů do architektury aplikace. Na poslední otázku z této sekce odpovědělo 5 respondentů a většina zmínila, že žádný další prvek je nenapadá. Další odpovědi nabízejí nápady jako info o členech či propagační video. Jedna z odpovědí také navrhuje podrobnější členění webu na více, příslušně pojmenovaných, stránek. Práce tuto připomínku vnímá, ale spíše jako záležitost UX veřejné části webu.

Třetí sekce obsahuje otázky týkající se stránek, které jsou přístupné pouze pro přihlášené uživatele. Přihlášení probíhá na stránce <https://klubfitpp.cz/prihlaseni>. První 4 z 5 otázek jsou velmi podobné prvním 4 otázkám z předchozí sekce. Opět se zde hodnotí obsah a nefunkční požadavky a obě otázky jsou doplněné o zdůvodňovací otázky. Výsledky jsou poměrně podobné jako výsledky veřejné části webu. Hodnocení obsahu je z větší části negativní. Respondenti většinou uváděli, že v soukromé části webu toho kromě editace účtu příliš není a mohlo by tam být více funkcí. Mezi odpovědmi byly ovšem i názory, že nedostatek obsahu v této části webu není až takový problém, protože tam toho víc nemusí být. Hodnocení nefunkčních požadavků soukromé části webu je spíše pozitivní. Mezi nejčastější výtky patří to, že design soukromé části nesedí k designu veřejné části webu. Rychlost zde zmiňována není. Poslední otázka v sekci se ptá respondentů, co by mohlo být součástí jejich profilu na webu klubu, popř. kde vidí prostor pro zlepšení této stránky. Padly zde návrhy na online volby, možnost přihlásit se na události, které jsou v plánu, či možnost přidat více informací na svůj profil např. fotografii, zájmy nebo dovednosti, aby bylo možné cíleně oslovit členy ohledně budoucích projektů.

Následují dvě paralelní sekce dotazníku, jedna pro vedení klubu, druhá pro ostatní členy. Obě sekce se zabývají funkcemi, které by mohl nový web členům nabízet. Obě sekce začínají otázkou s několika funkcemi, které je nutno seřadit podle priority, podobně jako v sekci o veřejné části webu. Následují otázky na zbytečnost některé ze zmíněných funkcí a na absenci funkcí, které nebyly zmíněny.

V sekci pro vedení obsahovala první otázka následující funkcionality:

- Správa majetku klubu,
- Správa akcí a jejich organizace,
- Online volby,
- CMS (content management system),
- Správa členů,

- Správa financí (faktury, rozpočet, ...),
- Správa dokumentů,
- Možnost koupit si FIT++ merch.

Nejlépe byly ohodnoceny funkce Online volby a Správa členů. CMS bylo, společně se Správou akcí a jejich organizace, v pořadí uprostřed. Nejmenší prioritu měly podle respondentů zbývající funkcionality, tedy Správa financí, dokumentů, majetku klubu a Možnost koupit si FIT++ merch. Tabulka 2.1 znázorňuje přehled průměry a mediány hodnocení v této otázce.

Funkce	Průměr	Medián
Správa majetku klubu	5,67	6,5
Správa akcí a jejich organizace	4	3,5
Online volby	2,33	1
CMS	4	4
Správa členů	3,17	2
Správa financí	5,5	5,5
Správa dokumentů	5,67	5,5
Možnost koupit si FIT++ merch	5,67	6

Tabulka 2.1: Průměry a mediány hodnocení jednotlivých funkcí, vedení klubu

V následující otázkách se objevily odpovědi, které zmiňují, že správa dokumentů, majetku, financí jsou všechny velmi potřebné a že respondent ohodnotil CMS nízkou prioritou, protože s ním nemá zkušenosti. Mezi nápady na novou funkci se zde objevila pouze jedna, a to In-game hra na lovení Apakrychlí [2].

Druhá z těchto dvou sekcí měla omezený výběr funkcionalit. Důvod omezení těchto možností byl ten, že členů, kteří nezastávali pozici v představenstvu klubu, je procentuálně mnohem více, a mohlo by dojít k tomu, že funkce důležité pro vedení klubu by se v hodnocení propadly, protože by pro ostatní členy nebyly podstatné. Dotazník ovšem počítá s tím, že tato větší část členské základny může mít zkušenosti z vedení jiných studijních klubů či spolků. Z toho důvodu obsahuje tato sekce doplňující otázku. První otázka obsahovala následující funkce:

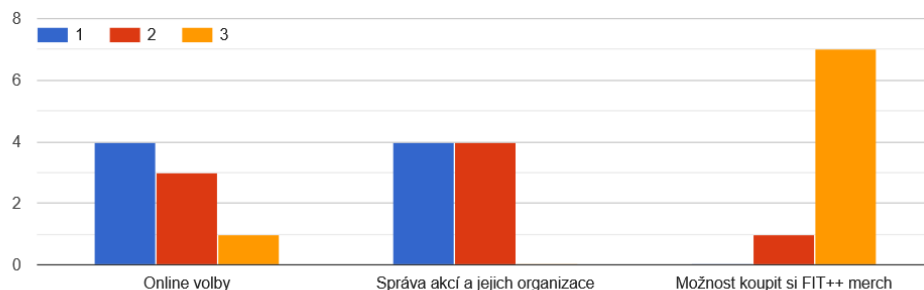
- Správa akcí a jejich organizace,
- Online volby,
- Možnost koupit si FIT++ merch.

Zde získala nejlepší hodnocení funkce Správa akcí a jejich organizace, následně Online volby a pak Možnost koupit si FIT++ merch. V odpovědích na doplňující otázku bylo ve dvou odpovědích zmíněno, že všechny z uvedených

## 2. ANALÝZA

---

Které funkce na webu považujete za nejdůležitější?



Obrázek 2.5: Graf otázky na funkce webu pro členy

funkcí jsou potřeba, a v jedné odpovědi, že zbytečnou funkcí je Možnost koupit si FIT++ merch. Poslední otázkou je již zmíněná doplňující otázka, která se ptá, jakou funkci by respondent na webu uvítal, pokud by byl součástí vedení klubu. Zde se objevila pouze jediná odpověď navrhuující dát možnost v rámci webu přidávat nápady na zlepšení, které by pak vedení klubu schválilo či zamítlo. Popř. možnost vytvořit anketu, kde by se mohli členové vyjádřit k probíranému tématu.

Závěrečná sekce obsahovala pouze jedinou otázku, která se ptala respondentů, zda se chtějí závěrem podělit o nějaký názor nebo připomínku, např. k webu či dotazníku. Odpovědí na tuto otázku bylo několik a šlo především o děkovné vzkazy a přání hodně štěstí při psaní bakalářské práce.

## 2.3 Analýza podobných projektů

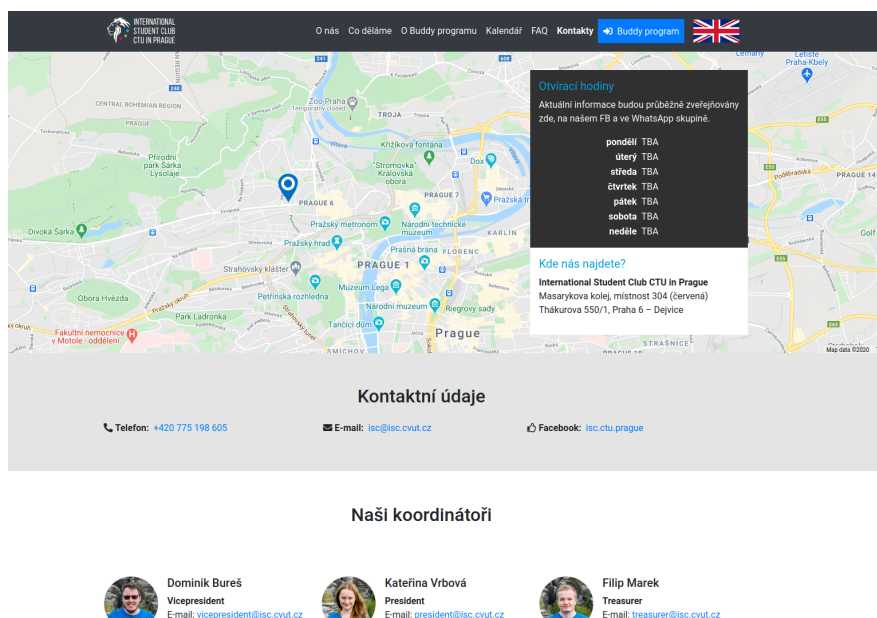
V této části se práce zaměřuje na webové aplikace podobného zaměření, porovnává jejich přístupy k návrhu a funkcionalitě. Všechny vybrané projekty jsou weby studentských klubů či spolků. Jedná se o kluby ISC [3], +/- FEL [4], ŠTUK [5] a eFDrive [6].

### 2.3.1 International student club (isc.cvut.cz)

Web klubu ISC je vytvořený zapomocí PHP frameworku Laravel pro backend aplikace a kombinace frameworku Vue.js a šablony Blade pro frontend. Databáze je relační MySQL a architektura aplikace je standardní MVC (model-view-controller). Na veřejné části webu lze najít souhrnný popis klubu a klubové činnosti, kalendář akcí, FAQ či stránku s kontakty. Obsah veřejné části webu je z části napsán staticky přímo v Blade šablonách (např. popis klubu) nebo

## 2.3. Analýza podobných projektů

získávan dynamicky z databáze (např. kontaktní informace či akce v kalendáři). V administrační části webu lze spravovat uživatele, vytvářet události



Obrázek 2.6: Stránka s kontaktními informacemi na webu ISC

či upravovat kontaktní informace. Dále je zde možné exportovat informace o uživateli v CSV formátu, zobrazit si různé statistiky uživatelů nebo logy aplikace. V aplikaci existuje několik typů uživatelů a to člen, zahraniční člen a buddy. Každá z těchto skupin o sobě ukládá lehce odlišné informace a pro každou je tedy dedikovaná vlastní tabulka v databázi. Události jsou dvojího typu – event a trip. Event se zobrazí na veřejné části webu v kalendáři akcí a nelze se na něj přihlásit. Trip se na druhou stranu zobrazí pouze v neveřejné části webu a lze se na něj přihlásit jako účastník. Online volby se na webu neřeší, klub k tomuto účelu využívá externí aplikaci.

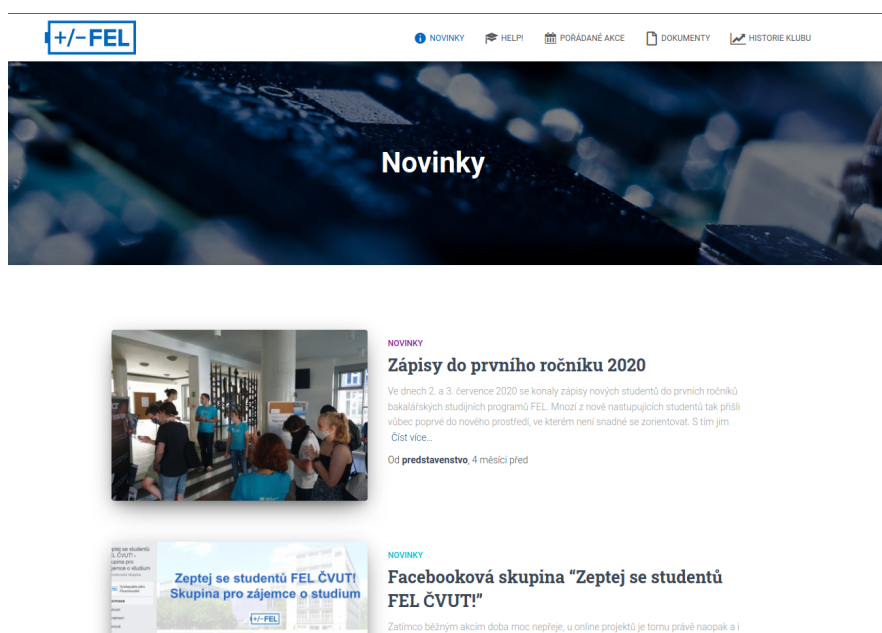
### 2.3.2 +/- FEL (pm.fel.cvut.cz)

Klub +/- FEL zvolil pro vytvoření svého webu nástroj WordPress [7]. Důvodem výběru bylo, že člověk neznalý webových frameworků dokáže pomocí tohoto nástroje vyrobit slušně vypadající web. Úpravy webu se provádějí skrze šablony a pluginy. Použité pluginy jsou např. Elementor (vizuální prvky na stránce), Jetpack (sledování návštěvnosti) nebo Akismet Anti-Spam (filtrace spam registrací a komentářů u novinek). Web ve své veřejné části zobrazuje mimo jiné novinky formou článků, informace o klubu, kalendář akcí, seznam dokumentů či časovou osu historie klubu. Aplikace nijak neukládá informace o členech klubu, a tedy nenabízí správu uživatelů. Události jsou řešeny buď pomocí importu Google kalendáře přes plugin nebo pomocí statického vložení

## 2. ANALÝZA

---

na stránku. Novinky lze vytvářet v administraci WordPressu pomocí webového editoru, který je podobný WYSIWYG [8]. Dokumenty jsou uloženy na odděleném FTP serveru. Není zde žádná synchronizace mezi tímto serverem a aplikací, takže odkazy na stránce je nutné přidávat ručně v administraci WordPressu.



Obrázek 2.7: Stránka s novinkami na webu +/- FEL

### 2.3.3 ŠTUK (stuk.fsv.cvut.cz)

Velmi podobný přístup jako klub +/- FEL zvolil i klub Fakulty stavební ČVUT, ŠTUK. Nepoužívá ovšem tolik pluginů a má poměrně jednoduchý design. Stejně jako +/- FEL nemá nějakou databázi členů, ale seznam členů lze najít v sekci O nás. Události a jejich správa se řeší jako články v rubrikách Plánované akce a Proběhlo. Funkcionality, kterými se tento web odlišuje od webu +/- FEL, je např. fulltextové vyhledávání nebo feed ze sociální sítě Facebook.

### 2.3.4 eFDrive (efd.su.cvut.cz)

Posledním webem, na který se práce zaměřila, patří klubu eFDrive pod Fakultou dopravní ČVUT. Tento web využívá PHP, jQuery, Bootstrap, Ajax a MySQL databázi. Ani zde se neukládají data o členech a klub to řeší mimo web. Správa událostí v aplikaci existuje a v minulosti se používala, ale momentálně již nikoliv. Místo toho je na webu vložen prvek, který je napojený



Obrázek 2.8: Stránka s akcemi na webu ŠTUK

na Google kalendář klubu a jsou zde zobrazeny události pro daný měsíc. Web dále obsahuje i fotogalerii obsahující dlaždicově znázorněná fotoalba. Každé fotoalbum je reprezentováno úvodní fotografií a po kliknutí je uživatel přesměrován na stránky služby Google Photos, kde jsou umístěny fotografie daného alba. Fotoalba na webu klubu jsou tvořena pomocí upravené knihovny Unite-Gallery, která se především stará o responzivní rozmístění fotoalb na stránce. Nové album je možné přidat po přihlášení přes formulář, kde uživatel zadá jméno alba, úvodní fotografii a odkaz na stránky Google Photos. Další důležitou stránkou jsou Dokumenty, která se dělí do několika sekcí podle toho, o jaké dokumenty se jedná. Každý dokument má svůj identifikátor, název, informaci o tom, kdo a kdy dokument nahrál a zda je dokument platný. Dokumenty jsou dostupné ke stažení a jsou uloženy na serveru webu. Web lze také přepínat mezi českou a anglickou verzí. Implementačně je toto řešeno přes PHP funkci lang, která je volána v HTML souborech aplikace. Úpravy textů na webu je možné přímo v prohlížeči po přepnutí do editačního módu, který nastaví upravitelným HTML prvkům na stránce atribut contenteditable a po provedení změn dojde potvrzením k odeslání změn na server.

## 2.4 Analýza webových technologií

Zde se práce zabývá analýzou webových frameworků a technologií, které by mohly být použity pro implementaci. V první části se porovnává způsoby, jakými lze provozovat prostředí pro webovou aplikaci. Dále rozebírá možné

## 2. ANALÝZA

**DOKUMENTY KLUBU EFDRIVE**

**STANOVY A INTERNÍ PŘEDPISY**

ID	Název souboru	Stažení	Vložil	Kdy	Plati
5	Stanovy klubu eFDrive (verze 2013)	↓ stáhnout	Petr	2017-12-02 00:46:04	✖
1	Stanovy klubu eFDrive (verze 2016)	↓ stáhnout	Petr	2017-12-01 20:05:04	✖
85	Stanovy klubu eFDrive (verze 2020)	↓ stáhnout	klub	2020-10-14 21:29:59	✔
30	2017-EFD-IP2 Interní předpis Volební řád	↓ stáhnout	adam	2018-11-28 02:06:21	✖
84	2020-EFD-IP3 Interní předpis Volební řád	↓ stáhnout	klub	2020-10-13 11:56:16	✔

**ZÁPISY ZE SCHŮZÍ PŘEDSTAVENSTVA**

ID	Název souboru	Stažení	Vložil	Kdy	Plati
15	Zápis ze schůze Představenstva 13. 06. 2013	↓ stáhnout	adam	2018-11-25 00:36:59	✔
16	Zápis ze schůze Představenstva 02. 12. 2013	↓ stáhnout	adam	2018-11-25 00:37:07	✔

Obrázek 2.9: Stránka s dokumenty na webu eFDrive

webové frameworky, které by se daly použít k implementaci samotné aplikace. V neposlední řadě porovná databáze, ve kterých by byla uložena uživatelská data. Nakonec práce analyzuje nástroje pro automatické testování a nasazení a jejich možné použití v rámci aplikace.

### 2.4.1 Konfigurace prostředí webové aplikace

Jedním způsobem, jakým lze provozovat webové aplikace na serveru, je v rámci operačního systému serveru spustit uzavřené prostředí, které bude pro danou webovou aplikaci poskytovat potřebné nástroje a knihovny. Na jednom serveru tak lze provozovat více webových aplikací, kdy každá může používat trochu jiné prostředí. Tato část kapitoly porovnává dva způsoby konfigurace těchto prostředí, na kterých jsou spuštěny webové aplikace. Jedná se o Docker [9] a virtualizaci operačního systému. Obě tyto varianty jsou schopné v rámci hostujícího operačního systému vytvořit uzavřené prostředí, které může mít nainstalované jiné knihovny, jinak nastavené oprávnění uživatelů či mít přiřazeno různé množství výpočetních prostředků. Nicméně webové aplikace, které jsou spuštěny v rámci systému Docker, si oproti těm spuštěným uvnitř virtuálního stroje vedou lépe z hlediska výkonu, lépe využívají hardwarové prostředky hostujícího stroje (menší část prostředků je zapotřebí k režii) a tyto aplikace také mají nižší latenci [10, 11, 12].

### 2.4.2 Webový framework

Mezi uvažované frameworky pro základ aplikace byli vybráni dva kandidáti, Laravel [13] a Symfony [14]. Oba tyto frameworky jsou napsány v PHP a v současné době patří mezi jedny z populárních PHP frameworků pro tvorbu webu



[15, 16, 17]. Laravel v tomto vede před Symfony, zejména díky plošší křivce učení a jednodušší syntaxí než má Symfony, a tím je vhodnější pro začínající vývojáře. Co se týče výkonnostních kritérií jako počet požadavků za sekundu, využití paměti či doby odezvy, tak zde Laravel opět překonává Symfony [18]. Na druhou stranu Symfony může být vhodnější pro větší (až enterprise) projekty a je považován za jeden ze stabilnějších PHP frameworků [19]. Symfony je také flexibilnější, má lépe rozšiřitelný kód a jeho komponenty jsou opakovaně použitelné, což šetří čas vývoje.

### 2.4.3 Nástroje pro automatické testování a nasazení

Součástí vývojového cyklu aplikace bývá standardně i integrace nového kódu do současné aplikace (nová funkce či oprava chyby), nasazení aplikace a následné otestování funkčnosti aplikace. Oba tyto kroky obsahují velké množství rutinních činností, které při kratších vývojových cyklech mohou v součtu zabrat mnoho času. Toto platí zejména u testování aplikace, kdy s přidáváním nových či opravou starých funkcí aplikace je vhodné ověřit, že provedené změny nepoškodily jiné části aplikace. Dnes již existují nástroje, které určitou část těchto rutinních záležitostí jsou schopné provádět automaticky. Proces, který tyto automatické nástroje využívá k bezpečné a rychlé integraci změn do aplikace a následného nasazení nové verze aplikace, se pak označuje CI/CD (Continuous Integration/Continuous Deployment) [20].

V rámci práce ovšem nebyla řádně započata analýza nástrojů pro CI/CD, návrh a integrace těchto nástrojů je tedy odsunuta na pozdější termín.

## 2.5 Analýza řešení pro online volby

Jak již vyplynulo z analýzy požadavků, jednou z nejvíce požadovaných funkcí nového webu je možnost online hlasování, neboli elektronické volby. Jedním z možných přístupů bylo celou logiku elektronických voleb implementovat bez využití již hotových řešení. Takové řešení se nezdálo jako optimální, protože by mohlo obsahovat chyby, které ověřené a zaběhnuté systémy mají již ošetřené. Cílem tedy bylo najít nástroj, který by dokázal zajistit logiku elektronických voleb. Jedná se o splnění následujících vlastností, které jsou výběrem z vlastností zmíněných v *Security in Large-Scale Internet Elections: a Retrospective Analysis of Elections in Estonia, The Netherlands, and Switzerland* [21]

### Identita

Musí být možné ověřit identitu člena klubu, a tedy i to, zda má právo hlasovat nebo zda již svůj hlas neodevzdal.

### Dostupnost

Volby se konají standardně jednou ročně a období hlasování je dopředu známo. Je tedy nutné, aby byla v tento čas aplikace dostupná a umožňovala členům klubu hlasovat.

### **Autentifikace a anonymita**

Přestože je třeba vědět, kdo může odevzdat hlas, je důležité zaručit anonymitu hlasujícího člena klubu.

### **Monitorování a kontrola správnosti uložení hlasu**

Odevzdaný hlas vypovídá o volbě člena klubu. Je nutné mít možnost zkontrolovat, že odevzdaný hlas skutečně patřil členovi s hlasovacím právem a zároveň ochránit anonymitu člena klubu.

Volební hlas v klubových volbách lze popsat jako velmi jednoduchý formulář (popř. dotazník), který obsahuje tolik otázek, jaký je počet kandidátů. Každá z otázek se dotazuje na postoj voliče vůči zvolení kandidáta do funkce (v případě klubu se jedná o Představenstvo klubu) a volič na ni musí odpovědět jednou ze tří možností: pro, proti, zdržuji se. Lze vybrat jen jednu z těchto možností, výběr více možností by zneplatnilo odevzdaný hlas. Také by nikdo neměl mít přístup k průběžným výsledkům do konce hlasovacího období, ani členové volební komise, kteří dohlíží na průběh voleb.

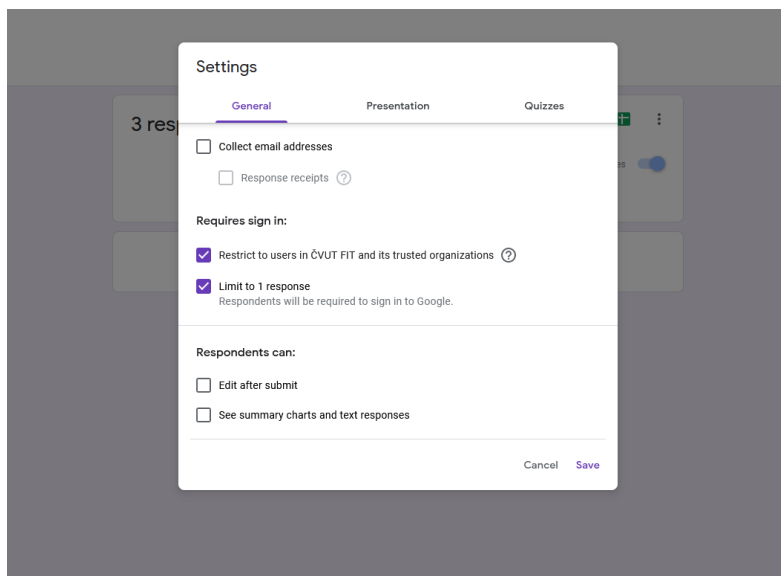
Z toho plynou další požadavky na hledaný nástroj. Možnost vytvořit dostatek otázek, na které lze odpovědět jednou ze tří, již zmíněných, možností a zamezit přístup k průběžným výsledkům voleb, dokud neskončí období hlasování.

#### **2.5.1 Google Forms [22]**

Jedná se velmi známý a rozšířený nástroj pro vytváření online dotazníků a průzkumů. Je možné pomocí něho vytvořit dotazník s různými druhy otázek a následně sbírat odpovědi od respondentů. Z hlediska implementace poskytuje API [23] v jazyce JavaScript, která dokáže vytvořit dotazník s požadovanými otázkami. Má ovšem několik nedostatků. Například, aby bylo možné zaručit anonymitu a identitu současně, musela by být vytvořena organizace v rámci Google a následně ji napojit na webovou aplikaci. Důvodem je, že pouze formulář vytvořený v rámci organizace dovoluje omezit přístup pouze pro členy organizace, tedy zajištění identity. To se jeví jako příliš komplikované vzhledem k rozsahu práce.

#### **2.5.2 SurveyMonkey [24]**

Tato služba se primárně specializuje na vytváření průzkumů v populaci. Oproti Google Forms zde lze upravit formulář více rozmanitými způsoby, ať už po grafické či funkční stránce, např. více druhy otázek. Dále oproti Google Forms je k dispozici REST API [25], místo JavaScript API, které opět dokáže obsloužit základní operace s dotazníkem. Neplacená verze je ovšem omezená na deset otázek v rámci jednoho dotazníku. To by se nemuselo zdát jako problém, poněvadž počet kandidátů do představenstva tento počet nikdy nepřesáhl, ale



Obrázek 2.10: Nastavení Google formuláře v rámci organizace FIT ČVUT

v pokud by došlo k zvýšené aktivitě v klubu, bylo by to zapotřebí buď procesně, nebo implementačně ošetřit.

### 2.5.3 LimeSurvey [26]

Posledním z uvažovaných nástrojů je LimeSurvey, který podobně jako zmiňovaní předchůdci umožňuje tvořit dotazníky s možností rozsáhlé parametrizace. Odlišuje se ovšem svou open-source filosofií a vývojáři této aplikace poskytují balíčky, které si mohou ostatní vývojáři následně zakomponovat do svých webových aplikací, společně s manuály pro správné vyřešení závislostí, připojení k databázi a celkově pohodlnější průběh integrace. Výhodou integrace nástroje do vlastní aplikace je kontrola nad verzí nástroje a také nad daty, se kterými LimeSurvey pracuje. Mimo jiné lze do LimeSurvey importovat informace o účastnících dotazníku nebo účastníky autorizovat pomocí protokolu LDAP (Lightweight Directory Access Protocol) [27]. Samozřejmostí je, že LimeSurvey splňuje vytyčené požadavky na nástroj pro online volby. Je zde možné vytvořit dotazník, který anonymizuje odpovědi účastníků, a zároveň pomocí přístupových tokenů zajišťuje přístup pouze pověřeným účastníkům. Vzhledem k tomu, že lze aplikaci provozovat jako komponentu vlastní aplikace, nejsou zde žádné omezení, co se týče počtu otázek v dotazníku nebo počtu odeslaných emailů. Jediné omezení, které může u některých aplikací nastat je případná nekompatibilita nebo nedostatečné výpočetní prostředky na straně serveru. V případě, že by se během implementační části práce vyskytly problémy s integrací LimeSurvey a nebylo by možné zajistit bezpečně

## 2. ANALÝZA

---

a důvěryhodné ověření uživatele, je možné vytvořit vlastní autorizační plugin, který by to již zajistit měl.

### **Shrnutí**

Z výše zmíněných důvodů byl vybrán nástroj LimeSurvey. Hlavním důvodem je, že lze zajistit již vytyčené požadavky jako je zajištění anonymity a ověření identity voliče. Je možné, že by mohly nastat problém s dostupností aplikace v případě cíleného DDOS (Distributed Denial-Of-Service) útoků [28], který by mohl vyřadit z chodu celý web a s ním i tu část pro online volby. Zde by byla výhoda mít aplikaci voleb zajištěnou jinou službou, která by byla provozována na jiném serveru. Nicméně s ohledem na velikost, majetek a vliv klubu FIT++ je velmi nepravděpodobné, že by takový útok mohl nastat.

---

## Návrh

Tato kapitola se zabývá, jakým způsobem vyřešit požadavky, které vyplynuly z analýzy. V první řadě je třeba si určit, se kterými daty bude naše aplikace pracovat. K tomu poslouží databázový model obsahující tzv. entity, které reprezentují objekty z reálného světa (člen, událost, volební hlas,...). Následně práce pomocí use case digramů [29] znázorňuje některé případy užití aplikace společně s aktéry, kteří mohou s aplikací pracovat, včetně toho, co všechno v ní mohou dělat. Tyto diagramy jsou pak v rámci implementační části práce vstupem pro návrh uživatelských rolí. Procesy, ve kterých tito aktéři následně participují, se pak zobrazují v diagramech aktivit [30], včetně toho, jakým způsobem se v rámci těchto procesů manipuluje s vytvořenými entitami.

Mimo výše zmíněné úkoly, které jsou základem pro návrh softwaru, je třeba navrhnout řešení problémů, které jsou specifické pro tuto práci. Jedná se zejména o to, jakým způsobem realizovat online volby v rámci aplikace.

### 3.1 Databázový model

Pomocí tohoto modelu jsou znázorněny jednotlivé databázové entity, které definují, s jakými daty aplikace pracuje a jaké vazby mezi sebou jednotlivé entity mají. Celý databázový model je zobrazen na obrázku 3.1.

#### 3.1.1 User

Jedná se o hlavní entitu, která reprezentuje člena klubu. Obsahuje především unikátní identifikátor v rámci klubu tzv. *UID*, které se uvádí na důležitých (např. pověřovacích) dokumentech v rámci Studentské unie. Dále obsahuje přihlašovací údaje do aplikace a osobní údaje člena jako např. adresa trvalého bydliště či datum narození. Nakonec drží informace o stavu přihlášky (viz. Registrace nového člena klubu) nebo další příznaky, třeba zda má člen ověřenou emailovou adresu nebo chce pomáhat na PR akcích FIT ČVUT.

### 3.1.2 Article

Tato entita představuje článek o aktuálním dění v rámci klubu. Může se jednat třeba o informační příspěvek či text popisující uplynulou akci. Součástí této entity je kromě titulku článku a samotného obsahu také reference na entitu User, která označuje autora článku. Jsou zde i dva atributy obsahující informaci o času. Prvním je čas vytvoření článku, který udává, kdy byl článek vytvořen v rámci aplikace. Druhým je čas zveřejnění článku na veřejné části webu, který je již nepovinným atributem. V případě nevyplnění je článek zveřejněn ihned po vytvoření.

### 3.1.3 Event

Entita Event znázorňuje událost, která probíhá v rámci klubu. Událost má svůj *název*, který je jediným povinným atributem (kromě id – primárního klíče). Mezi další nepovinné atributy patří časy začátku a konce události, místo konání a textový popis události. Zde je vhodné zmínit, že entita Event má s entitou User vazbu m:n, která je reprezentována tabulkou `users_organize_events`. Tato tabulka obsahuje kromě identifikátorů události a člena klubu také atribut rolí, který dovoluje upřesnit konkrétní relaci (např. zda je uživatel pouze účastník nebo organizátor).

### 3.1.4 Photo a Album

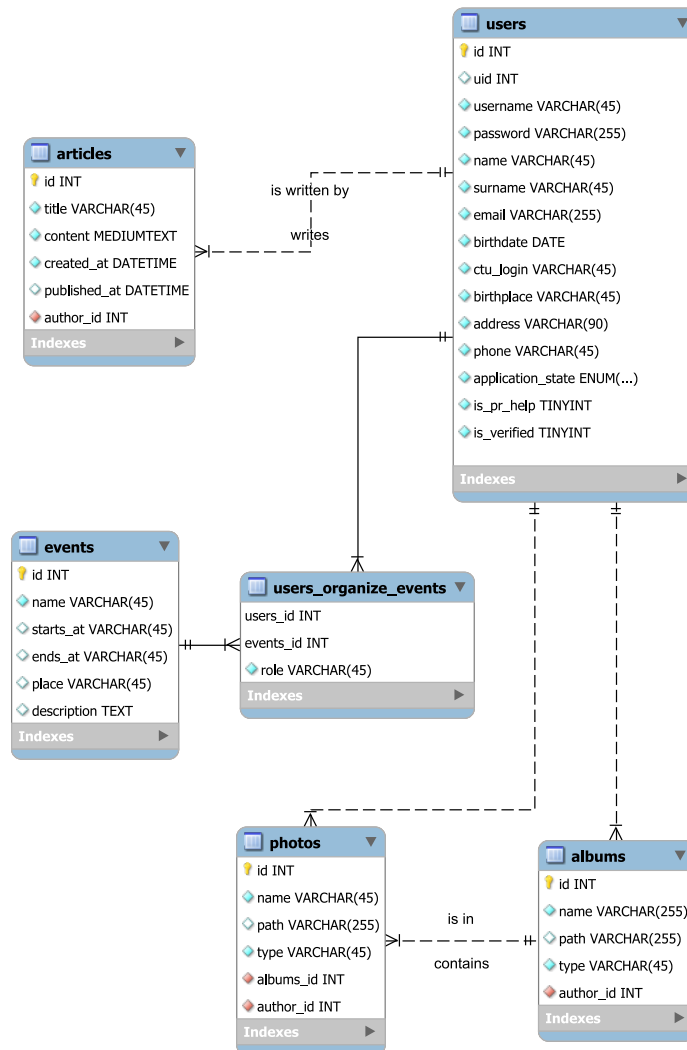
Tyto dvě entity reprezentují fotografie a alba v rámci galerie na webu. Obě entity mají *název*, *typ* a *cestu* k souboru nebo k složce. Atribut *typ* je zde pro případ, že by zdrojem fotografie či alba bylo něco jiného než úložiště na serveru (např. platforma Google Photos). Entita Photo v sobě má navíc referenci na entitu Album, která vyjadřuje, že daná fotografie je součástí nějakého alba. Obě entity mají taktéž referenci na entitu User, vyjadřující autora fotografie či alba.

## 3.2 Use case diagramy

Na základě požadavků, které byly popsány v předchozí kapitole, je třeba navrhnout případy užití aplikace, a následně je vizualizovat pomocí use case diagramů. V rámci práce jsou vypracovány dva diagramy případů užití – správa uživatelů a online volby.

### 3.2.1 Správa uživatelů

Správa uživatelů je pro aplikaci velmi důležitá, protože bez členů klubu, kteří jsou uživateli aplikace, by klub a jeho činnost nemohly existovat. V aplikaci je z hlediska správy uživatelů několik uživatelských rolí:



Obrázek 3.1: Databázový model

### 3. NÁVRH

---

- Neregistrovaný uživatel
- Běžný uživatel
- Registrátor
- Administrátor

Neregistrovaný uživatel je ten, kdo se zatím neregistroval do klubu. Jediné, co by měl takový uživatel mít možnost dělat, je registrovat se do klubu (popř. navštěvovat veřejné části aplikace).

Po dokončené registraci je se z potenciálního uživatele stává běžný uživatel. Tomu následně přibývá možnost přihlásit se svými přihlašovacími údaji či editovat své osobní údaje.

Někteří členové zastávají v klubu určité funkce. Jednou z těchto funkcí je standardně nabírání nových členů. V rámci aplikace se tito členové označují jako registrátoři. Registrátor si může v neveřejné části aplikace zobrazit ostatní členy spolu s jejich osobními údaji, které během procesu registrace nového člena kontroluje. Společně s tím by měl být schopný měnit stav přihlášky do klubu. V případě, že je přihláška validní, registrátor může vyexportovat registrační formulář v PDF souboru.

Administrátor je v aplikaci nejvyšší role, která má stejná práva jako všechny nižší role. Oproti registrátorovi je schopný nastavovat role jednotlivým uživatelům, a také by měl být schopný anonymizovat uživatele. Tato funkce je zde zahrnuta pro případ, že by člen klubu chtěl uplatnit své právo na výmaz [31].

Use case diagram pro správu uživatelů je znázorněn na obrázku 3.2.

#### 3.2.2 Online volby

Jedná se opět o podstatnou funkci, která by měla být součástí aplikace. Podobně jako u správy uživatelů i zde existuje více rolí v rámci aplikace. Je jimi člen volební komise a volič. Obě tyto skupiny mají možnost volit ve volbách v rámci aplikace. Člen volební komise má navíc možnost administrovat samotné volby. Jde především o vytvoření voleb a zveřejnění výsledků voleb. Během vytváření voleb je nutné mít možnost nastavit, v jakém časovém rozmezí se volby konají, a kdo ve volbách kandiduje.

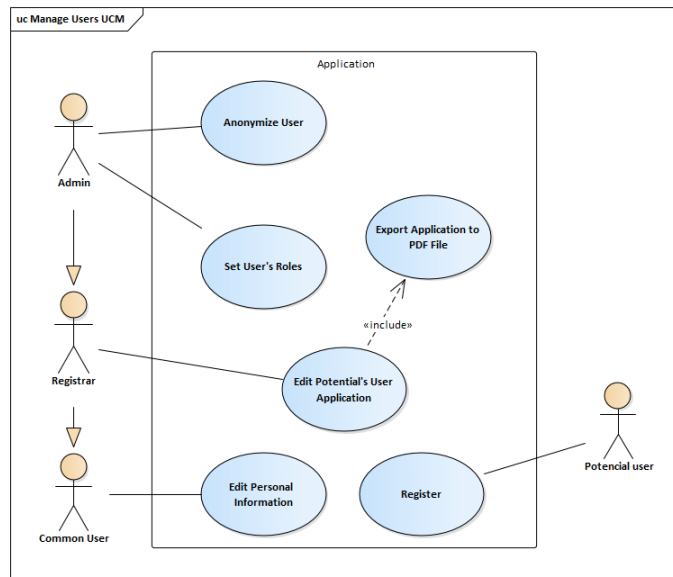
Na obrázku 3.3 se pak nachází use case diagram pro online volby.

#### 3.2.3 Organizace akcí

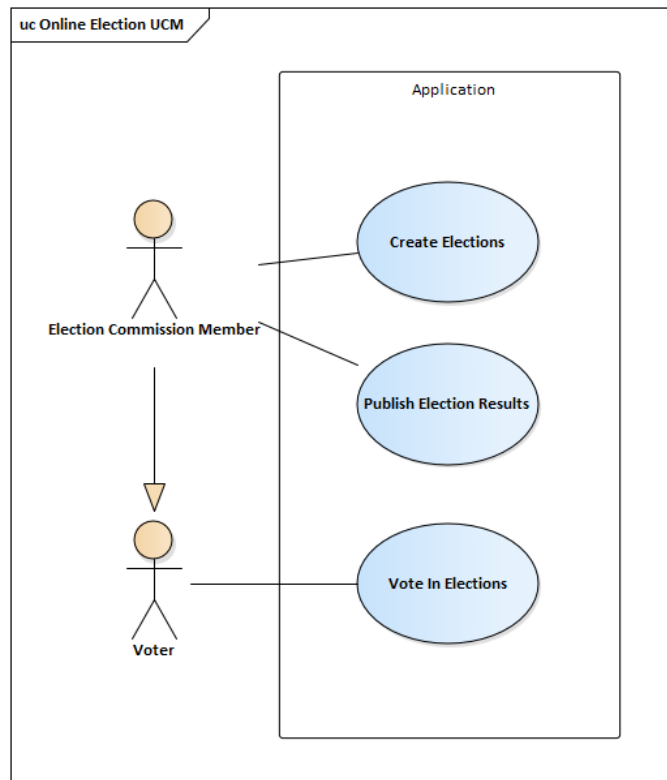
Jedná se o poslední z vybraných případů užití pro vizualizaci pomocí use case diagramů. Organizátorem akce může být jakýkoliv člen klubu, ale jen někteří by měli být schopní vytvářet veřejné akce, které by pak byly viditelné na veřejné části webu. Po vytvoření akce v aplikaci je možné, aby se ostatní členové klubu zapojili do dané akce. Mohou se zapojit buď jako pomocníci, nebo jako



### 3.2. Use case diagramy



Obrázek 3.2: Use case diagram správy uživatelů



Obrázek 3.3: Use case diagram online voleb

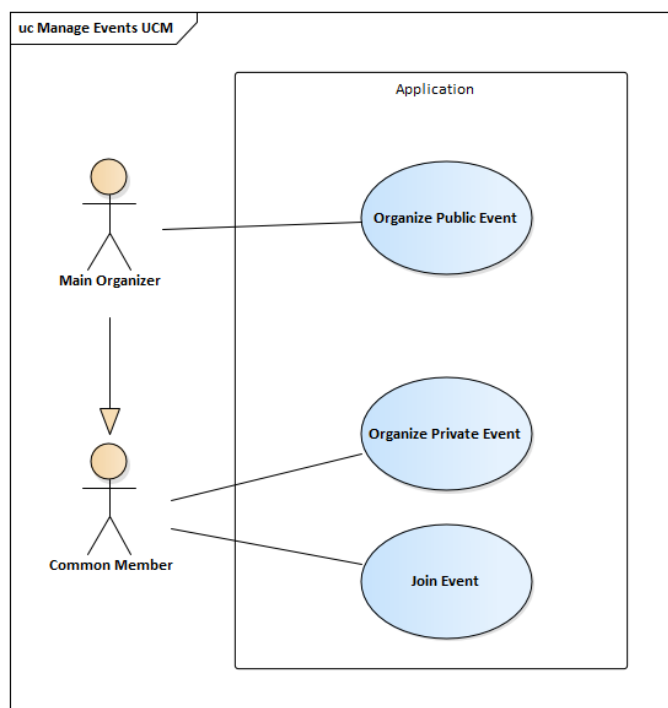
### 3. NÁVRH

---

účastníci. Pokud se člen zapojí do akce jako pomocník, dává tím organizátorovi akce najevo, že je ochotný pomoci s organizací akce. Takovou pomocí bývá obvykle dřívější příchod a pomoc u přípravných prací nebo pozdější odchod a pomoc s úklidem po akci. Pochopitelně, pokud se člen zapojí jako účastník, tak je to signálem, že člen dorazí až na akci samotnou, a bude si především užívat organizátory připravený program.

Podrobnější členění organizátorů (např. moderátor, obsluha pípy, mistr grilu, ...) momentálně není zahrnuto v rámci návrhu a implementace, ale je to něco, co by mohlo být přidáno v dalších verzích aplikace. Podobně by se dala organizace akcí rozšířit o časové možnosti jednotlivých organizátorů, protože v případě celodenní akce mohou mít jednotliví organizátoři akce čas v jinou část dne.

Obrázek 3.4 zobrazuje diagram pro tento případ užití.



Obrázek 3.4: Use case diagram správy akcí a jejich organizace

### 3.3 Procesy

Následně bylo potřeba znázornit důležité procesy v rámci klubu, které je potřeba přenést do aplikace. K tomu slouží tzv. activity diagramy, které dokáží v daném procesu popsat jednotlivé kroky, aktéry či vstupní/výstupní entity. Tento typ diagramu byl v práci použit pro znázornění následujících procesů:

- registrace nového člena klubu,
- online volby.

### 3.3.1 Registrace nového člena klubu

Proces registrace nového člena se odehrává mezi dvěma aktéry, registrátorem a zájemcem o vstup do klubu. Proces začíná tím, že zájemce vyplní a odešle registrační formulář na stránkách klubu. Tím se vytvoří záznam o podání nové přihlášky. Registrátor si následně novou přihlášku zobrazí a zkontroluje správnost osobních údajů. Pokud je přihláška v pořádku, registrátor změní stav přihlášky z *Nová* na *Ověřena*, v opačném případě na *Zamítnuta*. Pokud je přihláška ve stavu *Zamítnuta*, může být odstraněna. Pokud je přihláška ve stavu *Ověřena*, registrátorovi přibývá možnost přihlášku stáhnout ve formátu PDF. Pokud této možnosti využije, přihláška je uložena na jeho počítač a po vytisknutí dochází k fyzické schůzce registrátora a zájemce. Zde zájemce společně s registrátorem vyplní a oba podepíší vytištěnou přihlášku. Následně registrátor přepne přihlášku do stavu *Podepsaná*. Posledním krokem je přepnutí do stavu *Archivována*, který registrátor učiní po doručení přihlášky do kanceláře Studentské Unie ČVUT.

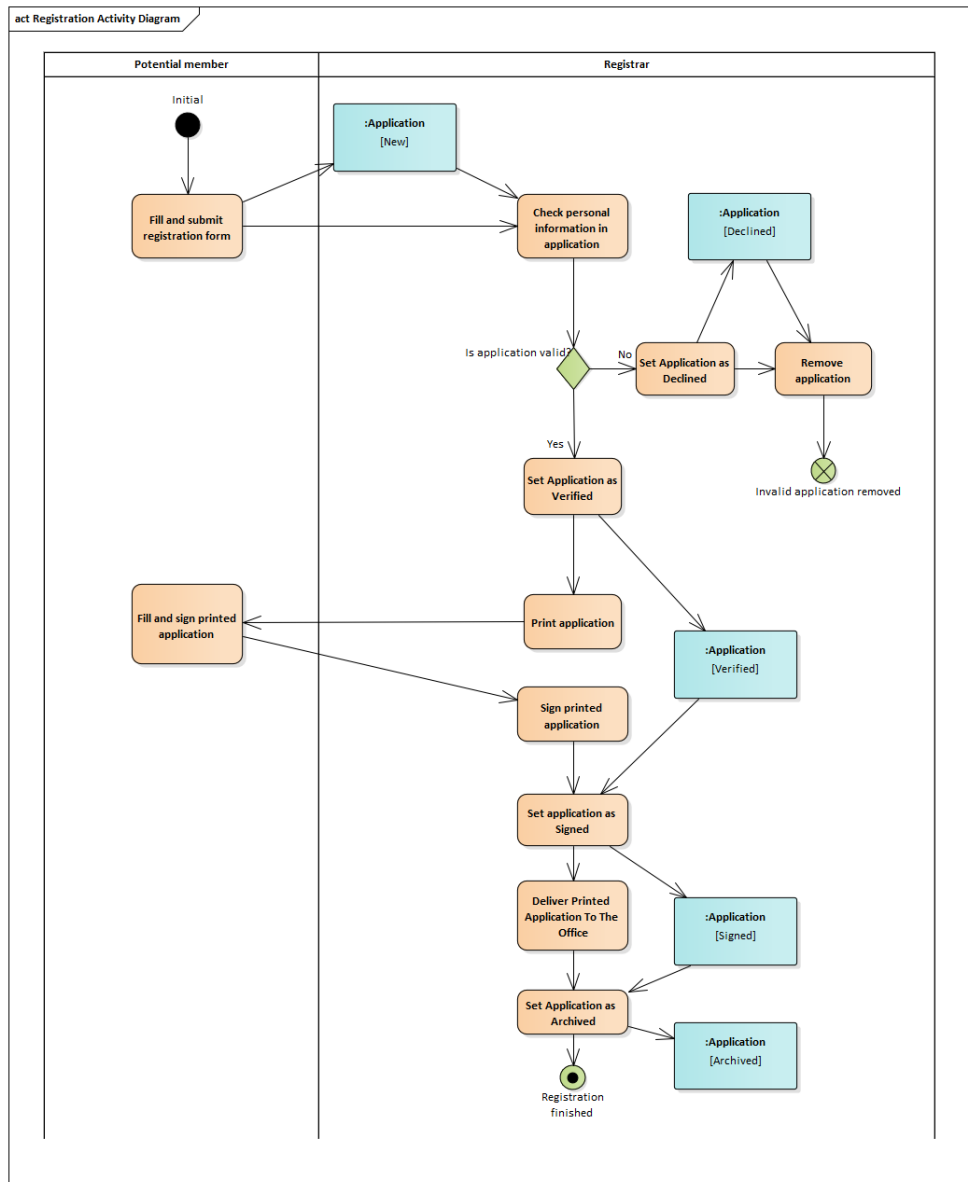
### 3.3.2 Online volby

Druhým procesem jsou online volby, které slouží ke zvolení nového Představenstva klubu. Působí zde celkem 4 aktéři:

- Člen Představenstva klubu,
- Administrátor,
- Člen Volební komise,
- Volič.

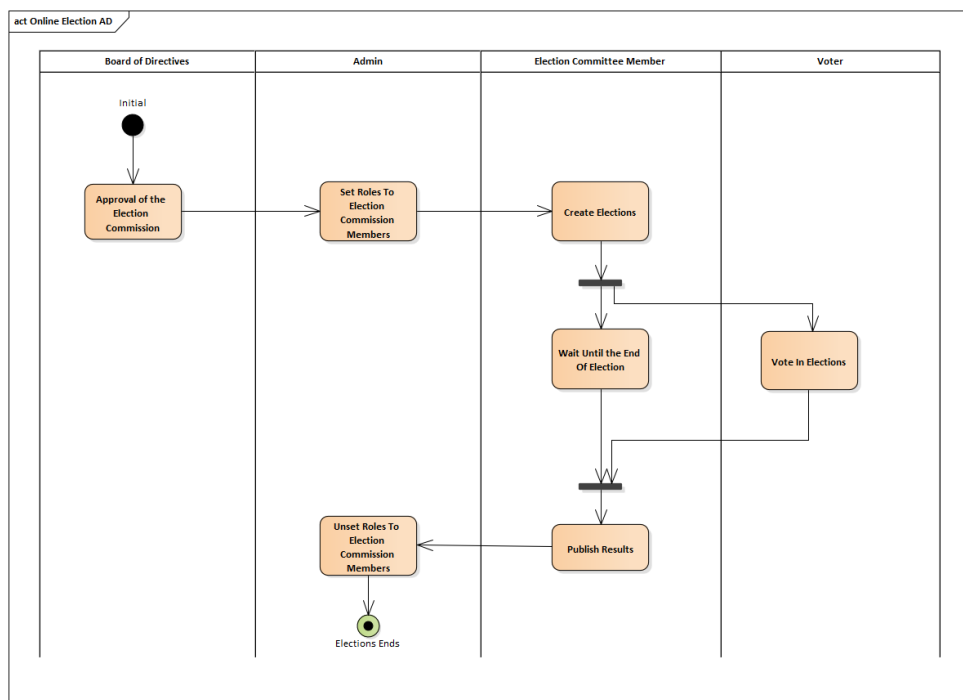
Mezi těmito skupinami neexistují restrikce v rámci aplikace. Mimo aplikaci existuje ještě jeden aktér, a tím je kandidát do nového Představenstva. Zde existuje restrikce taková, že člen Volební komise nesmí zároveň kandidovat. Role kandidáta se v aplikaci momentálně neřeší. Na začátku procesu Představenstvo klubu zvolí členy Volební komise pro nadcházející volby. Administrátor aplikace následně přiřadí daným členům roli Člen Volební komise. Předseda klubu poté vyhlásí volby a definuje povinné termíny voleb. Jedním z těchto termínů je Období podávání kandidátek, během kterého kandidáti zasílají kandidátky na emailovou adresu volebnikomiseFIT@gmail.com a členové Volební komise tyto kandidátky kontrolují. Následuje Období hlasování, kdy dochází k samotnému hlasování voliči. Každý volič se svým hlasem vyjádří ke všem kandidátům, a to buď pro, proti, či zdržením se. Poté, co skončí

### 3. NÁVRH



Obrázek 3.5: Activity diagram registrace nového člena

období hlasování, jsou zveřejněny předběžné výsledky voleb. Během následujících 48 hodin je možné podat stížnost k výsledku voleb, a tím napadnout legitimitu voleb. Pokud není vznesena žádná stížnost, jsou výsledky voleb konečné. V případě napadení voleb je celý případ prošetřen odpovědnými orgány. Během návrhu aplikace bylo cílem tento proces zachovat a pouze připravit rozhraní pro distanční formu voleb.



Obrázek 3.6: Activity diagram online voleb

### 3.4 Výběr nástrojů a technologií

Dle poznatků, získaných v analytické části, je vhodnější zvolit framework Symfony. Vzhledem k tomu, že se na webu klubu neočekává návštěvnost stovek lidí v jeden okamžik, tak rychlost, jakou má Symfony, bude dostačující. Je také možné, že v budoucnu se bude role webu FIT++ měnit a budou se měnit požadavky od členů, ať už funkční či strukturální, a bude nutné aplikaci rozšiřovat či modifikovat. Zde se opět více hodí použít Symfony oproti Laravelu. Je pravdou, že pro začínající webové vývojáře může být Symfony náročnější na pochopení než Laravel, a tedy je zde možnost, že by se web hůře předával dalším správcům, kteří by ho dále rozvíjeli. Nicméně i tak by mělo být pro následující správce zvládnutelné web převzít. Už jen tím, že má

### 3. NÁVRH

---

Symfony poměrně podrobnou dokumentaci a guidelines a také, podobně jako Laravel, širokou, celosvětovou komunitu.

Společně se Symfony se pojí i Twig [32], šablonovací systém primárně určený pro Symfony. V případě databáze bylo zvoleno MariaDB [33], které je opět standardním a dostačujícím řešením.

V analytické části práce se také rozebíralo, jak nastavit prostředí webové aplikace. Na základě zjištěných poznatků byl vybrán Docker, který zejména vyniká z hlediska výkonu.

Základ technologické sady nástrojů je tedy následující:

- Symfony – logika aplikace,
- Twig – šablonovací systém,
- MariaDB – databáze,
- docker – konfigurace prostředí aplikace,
- git – verzování.

---

## Realizace

Zde se práce věnuje samotné implementaci aplikace. Implementace se opírá o návrh, který je popsán v předchozí kapitole. První část této kapitoly popisuje realizaci kontejnerů pomocí nástrojů Docker a Docker Compose, které zajišťují konfiguraci aplikace. Další část rozebírá samotnou Symfony aplikaci. Jsou zde zmíněné obecné principy a návrhové vzory Symfony aplikací, jejich použití v kódu aplikace a postupy zvolené k vyhotovení jednotlivých případů užití. Ve třetí části této kapitoly je pozornost zaměřena na implementaci online voleb, protože jeho realizace je řešena mimo hlavní Symfony část. Předposlední část popisuje nasazení aplikace, a možné budoucí zlepšení pomocí CI/CD. Poslední část této kapitoly se věnuje testování aplikace, a to zejména uživatelskému, následně i okrajově automatickému.

### 4.1 Docker kontejnery

Jak již bylo zmíněno v kapitole Návrh, tak pro konfiguraci prostředí webové aplikace je zvolen nástroj Docker. Ten umožňuje pomocí konfiguračního souboru, který se nazývá *Dockerfile* [34], vytvořit předpis pro *image*, ze kterého se pak generují samotná prostředí tzv. *kontejnery*. Mimo Docker je použit i nástroj Docker Compose, který dokáže podle konfiguračního souboru (nejčastěji pojmenovaného *docker-compose.yaml*) vytvořit několik různých kontejnerů, a např. nastavit komunikaci mezi jednotlivými kontejnery.

Práce se inspiruje veřejným GitHub repozitářem *ger86/symfony-docker* [35], který slouží jako *boilerplate* [36] pro tvorbu Symfony aplikací s MySQL databází a nástrojem Nginx [37] v roli webserveru. Ze zmíněného repozitáře je částečně převzata struktura projektu, konkrétně složky *docker* a *symfony*, které obsahují konfigurační soubory pro Docker kontejnery a kód Symfony aplikace. Dále jsou použity konfigurační soubory pro Docker kontejnery Symfony aplikace a nástroje Nginx, které jsou z velké části podobné jako v repozitáři *ger86/symfony-docker*.

V souboru *docker-composer.yaml*, který je umístěn kořenovém adresáři, lze nalézt celkem pět konfigurací pro jednotlivé Docker kontejnery, kde každý je umístěn do jedné ze dvou *networks*. *Network* je v kontextu nástroje Docker Compose prostředí, kde jsou jednotlivé kontejnery navzájem dosažitelné [38]. Každý kontejner dostane přiřazenou vlastní soukromou IP adresu, přes kterou pak může probíhat komunikace s ostatními kontejnery.

Jak již bylo zmíněno, v aplikaci jsou dvě *networks*, jmenovitě *symfony* a *limesurvey*. *Symfony* obsahuje tři kontejnery, *database*, *php* a *nginx*.

Kontejner *database* je poměrně jednoduchý, vychází z *image* pro MariaDB a v práci je modifikován zejména nastavením časové zóny, aby byla zajištěna konzistence časových údajů ve všech kontejnerech.

O něco složitější je kontejner *php*, který značnou část své konfigurace do souboru *Dockerfile* v již zmíněné složce *docker*, která umístěna v kořenovém adresáři. V tomto souboru je oproti jeho původní verzi navíc instalace dalších knihoven a nástrojů, které jsou nezbytné pro vývoj a provoz aplikace. Jde např. o nástroje *composer*, *yarn* či různé PHP knihovny. Také pomocí souboru *php.ini* konfiguruje PHP v tomto kontejneru.

Posledním kontejnerem v rámci *symfony network* je *nginx*. Nástroj Nginx je zde v roli webserveru, přes který prochází komunikace mezi aplikací a klientem. Pomocí konfigurace tohoto nástroje lze např. nastavit dobu, po kterou se má čekat na odpověď aplikace, a v případě překročení spojení ukončit s chybovou hláškou v odpovědi. U známějších a navštěvovanějších aplikacích pak může Nginx mimojiné zastávat funkci jako tzv. *load balancer* [39], tedy rozložit provoz v aplikaci na více serverů či kontejnerů.

V *network limesurvey* jsou dva kontejnery, *limesurvey* a *mysql*. Zápis, kterým jsou tyto dva kontejnery uvedeny v souboru *docker-composer.yaml*, je inspirován zápisem v souboru *docker-compose.yaml* z GitHub repozitáře *adamzammitt/limesurvey-docker* [40]. Podobně jako u kontejnerů v *network symfony*, i tyto jsou upraveny tak, aby měly správně nastavené časové pásmo. Kontejner *limesurvey* je tvořen z Docker *image aospri/limesurvey*, který obsahuje komunitní verzi nástroje LimeSurvey. Z názvu je pak patrné, že kontejner *mysql* je MariaDB databáze, se kterou pak nástroj LimeSurvey pracuje.

## 4.2 Symfony aplikace

V této sekci je upřena pozornost na Symfony aplikaci, která je hlavním prvkem praktické části práce. Jsou zde popsány základní principy Symfony aplikace a realizace jednotlivých funkcionalit. V další podsekci je vysvětlena implementace administrační části aplikace pomocí knihovny EasyAdmin. Zbylé části této sekce se věnují realizaci implementovaných funkcí aplikace.

V kořenovém adresáři Symfony aplikace lze nalézt několik složek a souborů. Jedny ze základních souborů jsou konfigurační soubory *composer.json* a *composer.lock* pro nástroj *composer*. Tyto soubory v sobě drží informace



o použitých PHP knihovnách, a společně s nástrojem composer zajišťují, aby byly nainstalované všechny potřebné závislosti jednotlivých knihoven, popř. aby bylo zamezeno potenciálním konfliktům mezi knihovnamí. Soubor *symfony.lock* podobně jako *composer.lock* drží verze PHP knihoven, ovšem další data se týkají tzv. *recipes, které umožňují automatizovat konfiguraci composer balíčků přes Symfony Flex Composer plugin* [41].

Z adresářů je zde určitě třeba zmínit *src*, ve kterém je hlavní PHP kód celé aplikace, *config* obsahující zpravidla yml soubory s nastavením jednotlivých knihoven, *assets* pro CSS a Javascript soubory či *translations*, kde se obvykle nacházejí soubory mapující překlady do různých cizích jazyků. Dále zde je trojice adresářů *vendor*, *var* a *node\_modules*, do který se instalují zdrojové kódy knihoven stažených na základě dříve zmíněných souborů *composer.lock* a *yarn.lock*.

### 4.2.1 Model-View-Controller

Jedná se o aplikační návrhový model, který se skládá ze tří propojených částí:

- *model* – data používaná aplikací např. databáze či soubor,
- *view* – slouží k zobrazování objektů v aplikaci např. tlačítka,
- *controller* – pracuje s model i view částmi, které upravuje na základě vstupu [42].

Symfony aplikace pracují také podle tohoto modelu, konkrétně části typu *model* lze nalézt v adresáři *src/Entity*, *src/Controller* obsahuje komponenty typu *controller* a v *templates* jsou Twig šablony, které poskytují *view*.

### 4.2.2 EasyAdmin

Jak již název napovídá, tak tato knihovna slouží k vytváření administračních částí webových, konkrétně Symfony, aplikací. Tato knihovna poskytuje dvě abstraktní třídy typu *Controller*, a potomci obou těchto tříd reprezentují určité druhy stránek v rámci administrace, které pak může programátor využít. Jedná se o:

- *AbstractCrudController*,
- *AbstractDashboardController*.

Třídy, které dědí z *AbstractCrudController*, představují stránky poskytující CRUD [43] pro určitou databázovou entitu (např. *UserCrudController*). V základním nastavení tyto třídy poskytují čtyři druhy stránek v administraci:

- *Index*,
- *Detail*,

- *Edit*,
- *New*.

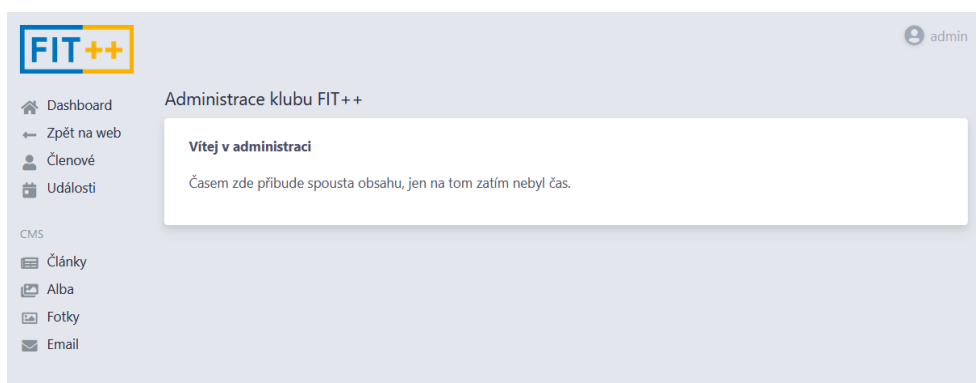
Názvy těchto stránek jsou poměrně návodné ke svému účelu. Tyto stránky pokrývají všechny CRUD operace, dokonce i operaci *delete*, která nemá samostatnou stránku, protože je realizována pomocí modálního okna, které se standardně zobrazí při stisknutí tlačítka *Smazat*. V rámci těchto tříd je možné přepsat nebo rozšířit metody třídy rodiče, a tím si konkrétní stránky uzpůsobit podle sebe. Metoda *configureFields* vrací seznam objektů, které standardně představují pole (textová, časová, check-boxy, ...) pro atributy jednotlivých záznamů databázové entity. Pomocí metody *configureActions* se nastavují operace, které lze provádět s jednotlivými záznamy databázové entity. V základním nastavení jde o již zmiňované operace CRUD, ale je zde i možnost vytvořit operace vlastní (v práci využito zejména u správy uživatelů a organizace akcí). Poslední z konfiguračních metod, která byla v práci použita, je *configureFilters*, a jak název opět intuitivně napovídá, tak nastavuje filtrování na stránce druhu *Index*.

Druhou zásadní třídou, kterou poskytuje knihovna EasyAdmin, je *AbstractDashboardController*. Potomci této třídy se oproti potomkám třídy *AbstractCrudController* neváží k žádné databázové entitě, a jsou určeny pro všechny ostatní stránky v administraci. Může se jednat o uvodní stránku, profil přihlášeného uživatele nebo nastavení administrace. V současnosti je tento typ tříd funkční pouze u třídy *DashboardController*, která momentálně obsahuje pouze uvítání pro právě přihlášené uživatele (obrázek 4.1). Ve vývoji je třída *ProfileController* jejíž účelem je zobrazit profil právě přihlášeného uživatele, kde by mohla být možnost editace osobních údajů, změna hesla, nastavení notifikací apod. Další třída, která zatím není plně funkční, je *EmailController*, sloužící k rozeslání hromadného emailu členům (např. na základě určitého klíče). Tato funkce je součástí i předchozího řešení, a vzhledem k jejímu využití by měla být zachována.

### 4.2.3 Správa členů

Realizaci této i ostatních funkcionalit v rámci Symfony aplikace lze z hlediska implementace rozdělit na dvě části – veřejná a administrační. Veřejná popisuje implementaci standardního MVC v Symfony, zejména části *View* a *Controller*, protože část *Model* je sdílená pro obě části a v zásadě reflektuje databázový model v kapitole Návrh.

Veřejná část se zde věnuje procesu přihlášení a registrace. Jedná se o standardní formuláře, které slouží přímo svému účelu. Po úspěšném odeslání registračního formuláře je uživatel přihlášen, a na emailovou adresu zadanou v registračním formuláři mu je zaslán email ohledně potvrzení emailové adresy. Posílání emailů je realizováno pomocí knihovny *symfony/mailer*, která je nastavená tak, že posílá emaily přes již založenou emailovou adresu. Během



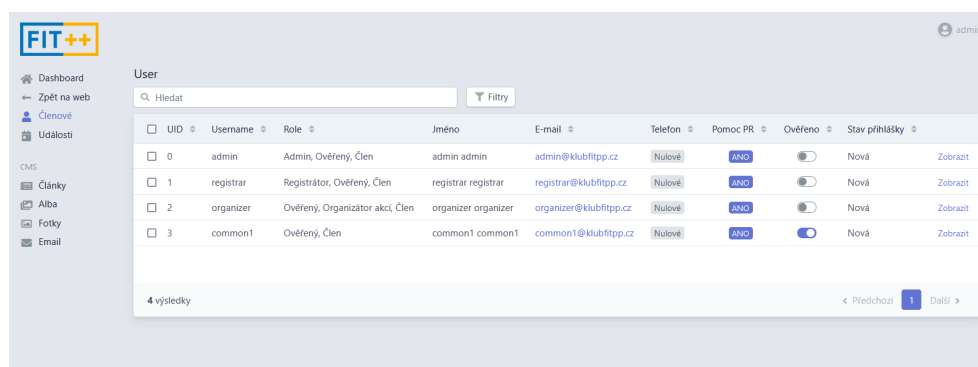
Obrázek 4.1: Úvodní obrazovka administrace

sestavování emailu je pak nahrazen odesílatel za `welcome@klubfitpp.cz`. Email je standardně v čistě textovém formátu, v dalších fázích vývoje by mohl být obohacen o více grafických a stylistických úprav.

Zajímavostí je, že v rámci emailu je adresát osloven správně v 5. pádu. To je zajištěno pomocí služby `www.sklonovani-jmen.cz` [44], která pomocí REST API rozhraní vrací správné oslovení podle zadaných parametrů (jméno, pád, tvar, ...). Při založení účtu dostane uživatel přiřazený API klíč, který je pak nutné vkládat do každého REST API požadavku. Uživatel dostane také kredit v hodnotě 500 kreditů, kde jeden kredit znamená jeden převod jména do správného tvaru. Navíc každý den má uživatel 10 bonusových kreditů, takže v případě, že registrací do klubu nebude denně o moc více než 10, tak základních 500 kreditů by mělo vydržet poměrně dlouho. Nedostatkem tohoto řešení je oslovení neregistrovaných členů, kteří pocházejí ze Slovenské republiky, protože slovenština nemá *vokativ*, a v oslovení se používá 1. pád. Zda v uvítacích emailech nechat oslovení jménem či nikoliv, by pak mělo být součástí širší diskuze specialistů v rámci klubu.

Stěžejní část správy uživatelů v rámci administrace je vytvořena v rámci třídy `UserCrudController`. Pole, která se definují v metodě `configureFields`, jsou rozdělena podle oprávnění. Běžný člen na detailech ostatních členů vidí pouze uživatelské jméno, jméno a příjmení. Registrátoři, neboli členové s rolí `ROLE_REGISTRAR`, vidí navíc ostatní osobní údaje člena a také stav přihlášky do klubu. V dalších fázích vývoje by mohlo být umožněno uživatelům odhalit své osobní údaje, aby je třeba mohli ostatní členové snáze kontaktovat. Ostatní operace CRUD jsou značně omezené. Operaci `edit` smí provádět pouze člen sám na sobě, výjimkou je možnost nastavit ověření uživatele pro administrátora pro případ, že by neregistrovanému uživateli zapadl verifikační email do nevyžádané pošty či se jiným způsobem ztratil. Jedná se o dočasné řešení, které by mělo být později nahrazeno možností znovu odeslat verifikační email. Operaci `delete` smí provádět pouze administrátor, a to pouze v případě,

## 4. REALIZACE



Obrázek 4.2: Seznam členů v EasyAdmin administraci z pohledu administrátora

že je stav přihlášky ve stavu *Zamítnutá*. Poslední ze základních CRUD operací, *new*, je zakázaná všem uživatelům. V případě, že by časem vznikla poptávka po systémových účtech, které se nevytváří procesem registrace, operaci *new* by určitě bylo možné přidat. Na obrázku 4.2 lze vidět seznam členů z pohledu administrátora.

Kromě standardních CRUD operací je správa uživatelů obohacena o několik vlastních operací. Jedná se o:

- *approve\_application*,
- *sign\_application*,
- *decline\_application*,
- *archive\_application*,
- *generate\_pdf\_application*,
- *batch\_generate\_pdf\_application*.

První čtyři operace ze seznamu manipulují se stavem přihlášky a zajišťují správné přechody mezi jednotlivými stavy, které jsou popsány v návrhu procesu registrace. Detail uživatele s přihláškou ve stavu *Nová* zachycuje obrázek 4.3. Při přechodu ze stavu *Nová* do *Ověřena* je navíc uživateli nastaven atribut *UID*, který je poté součástí papírové přihlášky. A právě k papírové přihlášce se váží poslední dvě operace, které jsou specifické pro administraci členů, *generate\_pdf\_application* a *batch\_generate\_pdf\_application*. Obě tyto operace dokáží vygenerovat předvyplněný PDF soubor. Operace *batch\_generate\_pdf\_application* se liší tím, že se jedná o speciální druh, protože narozdíl od ostatních operací ji lze použít pouze na stránce *Index*, a díky

User (#136)		Zamítnout přihlášku	Schválit přihlášku	Zpět na výpis
UID	4			
Username	hanusmi			
ČVUT login	hanusmi			
Jméno	Mistr			
Příjmení	Hanuš			
E-mail	<a href="mailto:hanusmi@klubfitpp.cz">hanusmi@klubfitpp.cz</a>			
Telefon	Nulové			
Datum narození	1. 1. 2000			
Místo narození	FIT ČVUT			
Adresa trvalého bydliště	Thákurova 9, 160 00 Praha 6			
Pomoc PR	ANO			
Ověřeno	NE			
Stav přihlášky	Nová			

Obrázek 4.3: Detail zatím neschváleného člena z pohledu registrátora

ním je možné označit více záznamů současně a spustit operaci nad všemi označenými záznamy. V případě operace `batch_generate_pdf_application` se jedná o vygenerování jednoho PDF souboru, kde každá stránka obsahuje informace o jiném členovi.

Pro generování PDF souboru se používají knihovny *Setasign/FPDF* [45] a *Setasign/FPDI* [46]. Knihovna *Setasign/FPDF* dokáže generovat PDF soubory a libovolně do nich vkládat text, obrázky či HTML. *Setasign/FPDI* doplňuje *Setasign/FPDF* o možnost použít již vygenerovaný PDF soubor či jen některé jeho stránky jako šablonu. Při generování předvyplněných registračních přihlášek se nejdříve pomocí knihovny *Setasign/FPDI* vloží prázdná PDF přihláška na novou stránku, a následně za pomoci metod knihovny *Setasign/FPDF* vloží příslušné údaje do kolonek přihlášky. V základní verzi knihovny *Setasign/FPDF* ovšem není žádný font, který by dokázal použít specificky české znaky s diakritikou. Z toho důvodu jsou do adresáře `vendor/setasign/fpdf/font`, obsahující fonty, které tato knihovna může použít, přidány soubory `Titillium-Regular.z` a `titillium_regular.php`. Tyto soubory lze za

pomoci OTF souboru pro font vygenerovat na webu [www.fpdf.org/makefont](http://www.fpdf.org/makefont). Font Titillium [47] byl vybrán, protože je primárním fontem Studentské unie ČVUT [48],

Vnitřní logika generování PDF přihlášek a přepínání stavů přihlášek je pak delegována do třídy *ApplicationManager*, která se nachází ve složce *src/Service*.

### 4.2.4 Organizace akcí

V rámci implementace veřejné části toho není tolik, jako u správy uživatelů. Nicméně, i tak by se to chtělo zmínit o třídě *EventController*, která poskytuje metody *index* a *detail*. Metoda *Index* zajišťuje vykreslení stránky */event*, na které lze vidět tři seznamy veřejných akcí:

- *Probíhající události*,
- *Nadcházející události*,
- a *Uplynulé události*.

V těchto seznamech jsou jednotlivé akce zobrazeny kompaktně a pouze s nejdůležitějšími informacemi a jsou zkráceny o část popisu. Na tyto zobrazené akce lze následně kliknout pro zobrazení detailu akce.

Administrační část organizace akcí se opět opírá o svou *CrudController* třídu, tedy *EventCrudController*. V rámci oprávnění u jednotlivých polí mají oproti ostatním uživatelům navrh pouze ti s rolí *ROLE\_EVENT\_ORGANIZER*. Tito uživatelé reprezentují klubové organizátory akcí, a mají možnost nastavit, zda je událost veřejná či nikoliv, tedy zda se má zobrazit ve veřejné části aplikace. Mimo to je pro seznam přihlášených uživatelů k události vytvořena šablona *list\_users\_association.html.twig*, která na stránce detailu události zobrazuje seznam členů a jejich organizačních rolí. Toto zachycuje obrázek 4.4.

Před popisem implementace operací, by bylo záhodno vysvětlit jednotlivé role, které zde figurují. V tomto případě ovšem není řeč o rolích čistě uživatelských, ale těch, které jsou součástí vztahu entit *User* a *Event*. Jsou to role:

- *ROLE\_ORGANIZER* – hlavní organizátor akce, který vytvořil událost v aplikaci,
- *ROLE\_HELPER* – člen, který chce pomoci s organizací akce,
- *ROLE\_ATTENDEE* – člen, který se chce pouze dostavit na akci a nepomáhat.

Vysvětlení vzniku těchto rolí je vysvětleno v popisu případu užití organizace akcí.

Event (#203)		Zpět na výpis		Chci pomáhat		Jen jako účastník	
Název	Testovací událost č.0						
Veřejná	ANO						
Začátek události	27. 6. 2021 11:27:45						
Konec události	28. 6. 2021 11:27:45						
Místo konání	Testerov						
Popis	Vygenerovaný popis						
Účastníci	<b>Člen</b>			<b>Role</b>			
	hanusmi		Mistr Hanuš	Hlavní organizátor			
	kraldomi		Dominik Král	Organizátor			
	kulickre		René Kulička	Účastník			

Obrázek 4.4: Detail události v administraci

Co se týče nastavení operací nad jednotlivými akcemi, podobně jako ve třídě *UserCrudController* zde bylo nutné přidat vlastní operace nad rámec CRUD operací. Konkrétně jde o možnosti přihlášení se na akce metodami *join\_event\_as\_attendee* a *join\_event\_as\_helper* a odhlášení se z akcí pomocí *disjoin\_event*. Logika těchto operací je opět přesunuta do oddělené třídy, která se jmenuje *EventOrganizer* a opět ji lze nalézt ve složce *src/Service*. Operace *join\_event\_as\_attendee* přihlásí přihlášeného uživatele k události jako účastníka, *join\_event\_as\_helper* jako pomocníka a *disjoin\_event* přihlášeného uživatele z akce odhlásí. Všechny tyto operace se zobrazují pouze, pokud by jejich provedení mělo za následek změnu vztahu mezi přihlášeným uživatelem a danou událostí.

Kromě nových operací je v metodě *configureActions* třídy *EventCrudController* také lehce upraveno chování CRUD operací *edit* a *delete*. Je zde totiž problém, že upravovat a mazat události by měli mít možnost pouze ti, kteří danou událost vytvořili, nebo uživatelé s rolí *ROLE\_EVENT\_ORGANIZER*. Libovolné operace lze upravit zavoláním metody *update*, která přijímá tyto parametry:

- název stránky, kde se má úprava metody aplikovat,
- název operace, které se úprava týká,
- a funkce, která přijímá v parametru původní operaci a vrací upravenou operaci.

Zda má být operace dostupná na dané stránce se následně upravuje zavolání metody *displayIf*, která přijímá funkci s entitní třídou jako parametrem a vracející hodnotu typu bool. Aby ovšem bylo možné zjistit, zda se má být operace dostupná, musí být v rámci funkce znám identifikátor přihlášeného uživatele. Ten je do funkce metody *displayIf* předán pomocí klíčového slova *use*, které dovoluje předat do funkcí vnější objekty. V tomto případě se jedná o uživatelské jméno přihlášeného uživatele a instanci již zmíněné třídy *EventOrganizer*, která poskytuje metodu *canEditOrDelete* vracející, zda uživatel s tímto uživatelským jménem může upravovat či mazat tuto událost. Pro lepší představu je část kódu na ukázce 1.

```
1  ...
2  return Actions::new()
3  ...
4  ->update(
5      Crud::PAGE_INDEX,
6      Action::EDIT,
7      function (Action $action)
8          use ($eventOrganizer, $logged_in_username)
9      {
10         return $action->displayIf(
11             function (**@var Event $event */ $event)
12                 use ($eventOrganizer, $logged_in_username)
13             {
14                 return $eventOrganizer->canEditOrDelete(
15                     $logged_in_username,
16                     $event
17                 );
18             });
19     });
20  ...
```

Listing 1: Pokročilé použití metody *update* pro operace

V dalších fázích vývoje by pak mělo dojít k nahrazení polí pro začátek a konec události, protože současné řešení pomocí nativních widgetů není v některých prohlížečích uživatelsky přívětivé.

#### 4.2.5 Galerie

Veřejná část této funkcionality je zajištěna pomocí třídy *GalleryController*, která zprostředkovává metody *index* a *detail*. Metoda *index* zajišťuje stránku se seznamem všech alb. U této stránky je potřeba dopracovat grafické zpraco-



Obrázek 4.5: Fotografie zarovnané knihovnou *justfiedGallery*

vání, protože současná podoba je velmi strohá (podobně jako většina veřejné části aplikace). Metoda *detail* následně poskytuje stránku s albem jeho fotografiemi. K vykreslení galerie jsou použity dvě knihovny – *lightGallery* [49] a *justfiedGallery* [50]. Obě knihovny poskytují metody, které lze spustit na statické HTML galerie a následně z nich vytvořit galerie dynamické. Díky *justfiedGallery* jsou všechny fotky rovnoměrně zarovnané (obrázek 4.5), a *lightGallery* zobrazí prezentaci s ovládacími prvky jako automatické přehrávání či možnost sdílet danou fotografii na sociální síť (momentálně není funkční, protože aplikace je dostupná pouze přes fakultní VPN).

V administrační části aplikace je správa fotografií a alb momentálně umožněna pouze správcům obsahu aneb uživatelům, kteří mají přiřazenou roli *ROLE\_CMS*. Stěžejní jsou zde opět *CrudController* třídy, tedy *PhotoCrudController* a *AlbumCrudController*. Atribut *type* v rámci nastavení polí obou tříd má zatím pouze hodnotu *file*, tedy souborové úložiště. Později by mohlo být možné přidat více druhů úložiště např. Google Photos. Obě třídy jsou poměrně jednoduché, co se týče nastavení polí a operací.

Zajímavější je ovšem práce se soubory fotografií. Po vytvoření alba a následném vytvoření fotografie v rámci alba je fotografie uložena do složky *public/images/uploads*, a pak je pomocí metody *setPhotoPath* v třídě *EasyAdminSubscriber* přesunuta do správné složky alba. V této třídě je více metod, které podobně jako tato provádějí operace s fotografiemi či alby na souborové úrovni. V současné chvíli je funkční pouze vytváření a mazání fotografií a alb. Změna alba u fotografie je součástí momentálního vývoje aplikace.

## 4. REALIZACE

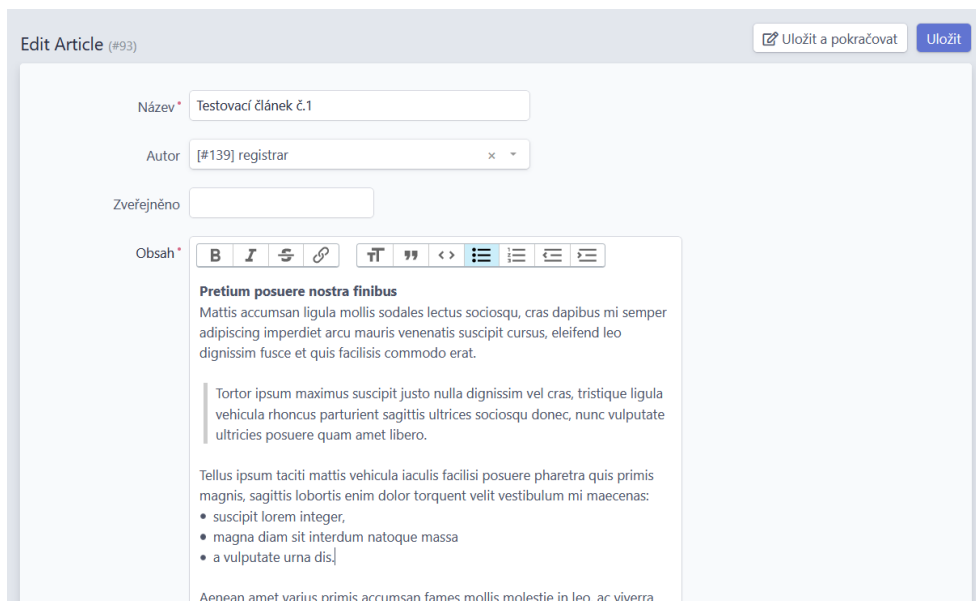
---

```
1 ...
2 public function setPhotoPath(AfterEntityPersistedEvent $event): void
3     {
4         $entity = $event->getEntityInstance();
5
6         if (!$entity instanceof Photo) {
7             return;
8         }
9         if ($entity->getType() !== 'file') {
10            return;
11        }
12        $files = $this->finder->files()
13            ->in('/var/www/symfony/public/images/uploads')
14            ->sortByModifiedTime()
15            ->reverseSorting();
16        /**@var SplFileInfo $last_file */
17        $last_file = $files->getIterator()->current();
18        $new_filename = $entity->getId() . '_' . $entity->getName()
19            . '.' . $last_file->getExtension();
20        $new_path = '/var/www/symfony/public/'
21            . $entity->getAlbum()?->getPath() . '/' . $new_filename;
22        if ($this->filesystem->exists($last_file->getRealPath())) {
23            $this->filesystem->rename($last_file->getRealPath(), $new_path);
24            $this->logger->info('File ' . $last_file->getRealPath()
25                . ' was successfully moved.');
```

```
26            $entity->setPath($new_filename);
27            $this->entityManager->persist($entity);
28            $this->entityManager->flush();
29        } else {
30            $this->logger->critical('File ' . $last_file->getRealPath()
31                . ' don\'t exist.');
```

```
32        }
33    }
34 ...
```

Listing 2: Metoda *setPhotoPath* přesouvající nahranou fotografii do alba



Obrázek 4.6: Editace článku v administraci

### 4.2.6 Aktuality

Zde je implementace poměrně jednoduchá. Ve veřejné části aplikace je sekce *Aktuality*, která obsahuje seznam všech článků seřazených podle času zveřejnění od nejnovějších po ty nejstarší. Jednotlivé články v seznamu obsahují název, autora, datum zveřejnění a začátek samotného článku. Kliknutím na nadpis je pak uživatel přesměrován na detail článku, který už obsahuje text celý.

Administrativní část pro články je nastavena třídou *ArticleCrudController*, která jediné co, tak nastavuje jednotlivá pole pro atributy entitní třídy *Article*. Podobně jako v případě administrace fotografií a alb, tak i do této části administrace je omezen přístup uživatelům, kteří nejsou správci obsahu. Na obrázku 4.6 lze vidět editaci článku.

Implementace z počátku vycházela z návrhu, který definuje zveřejnění článku v čas, který je uložený v atributu *published\_at*, a v případě, že hodnota tohoto atributu není nastavena, tak je článek okamžitě zveřejněn. Současná implementace místo toho přidává operace *publish* a *hide*, které nastaví či odstraní čas zveřejnění automaticky. Důvodem pro změnu bylo zejména to, že automatické zveřejnění by pro někoho mohlo být matoucí až nepříjemně překvapivé.

### 4.3 Online volby

Realizace tohoto požadavku vychází z analýzy řešení pro online volby. Na základě výsledků analýzy byl vybrán nástroj LimeSurvey, který nabízí široké možnosti nastavení průzkumů a voleb (obrázek ??). Momentálně je nástroj nasazen jakožto samostatný Docker kontejner s MySQL databází v kontejneru druhém (viz. Docker kontejnery). Již je možné vytvořit volby, ručně či importem CSV souboru nahrát účastníky průzkumu, kterým lze následně vygenerovat přístupový kód pro vstup k urnám, a pomocí kódu mohou voliči vložit svůj hlas. Ovšem stále nefunguje několik důležitých věcí jako třeba rozesílání emailových pozvánek a upomínek k hlasování, kde se autorovi práce stále nepodařilo lokalizovat problém. Také tu je vkládání voličů, které by bylo vhodné automatizovat a provázat se Symfony aplikací, která drží informace o členech, a tedy i o legitimních voličích. Nakonec ještě zbývá vyřešit to, že nástroj LimeSurvey běží na portu 8082, protože port 80 zabírá kontejner *nginx*. Je tedy nešikovné do takového nástroje přecházet. Možným řešením by mohlo být překonfigurovat kontejner *nginx* tak, aby směřoval určité požadavky do kontejneru *limesurvey* (např. z adresy */lime*), a ty ostatní do kontejneru *php*. Nicméně řešení těchto problémů bude vyžadovat další analýzu v následujících fázích vývoje.

### 4.4 Nasazení aplikace

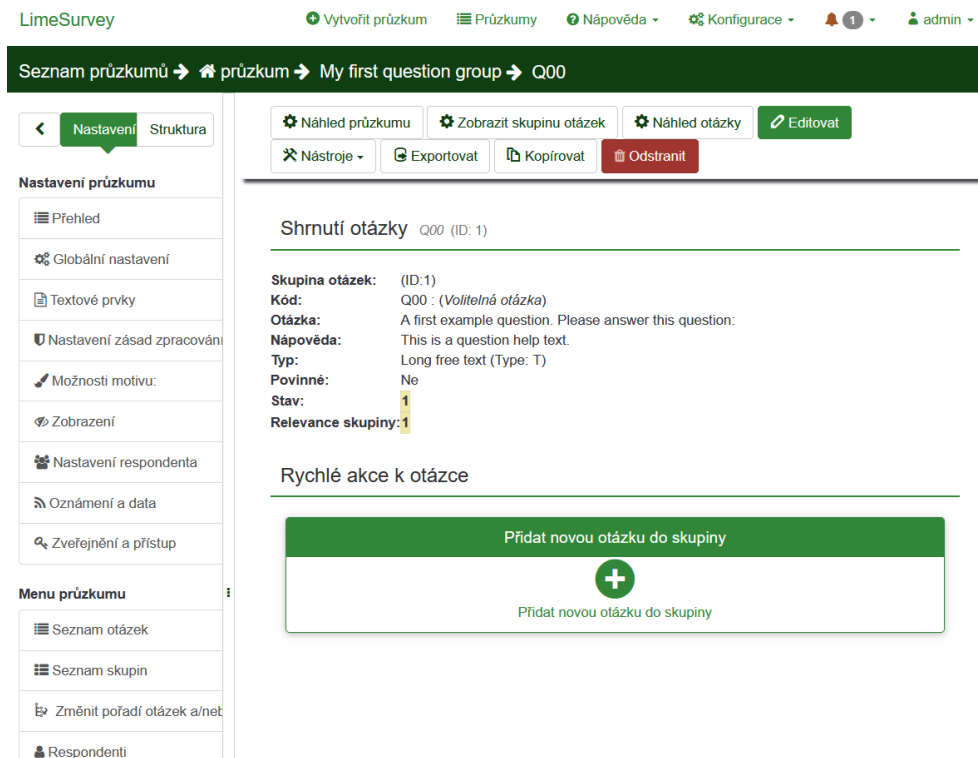
Nasazení aplikace je v současné chvíli manuální záležitost. Aplikaci je nutné stáhnout přes nástroj git na stroj, kde by měla aplikace běžet (popř. přesunout v archivovaném souboru). Poté je nutné nastavit přístupové údaje a jiné *environment variables* v souborech *.env.local* a *symfony/.env.local*, aby byla např. správně nastavené spojení mezi Symfony aplikací a databází. Následně stačí spustit skript *install.sh*, který stáhne potřebné Docker *images* a vytvoří Docker kontejnery. Poté nainstaluje PHP a NodeJs knihovny v kontejneru *php*, a po nainstalování NodeJs knihoven spustí příkaz *yarn run webpack build*, který sestaví css soubory do adresáře *public*.

Dalším milníkem v nasazování aplikace je zavedení CI/CD, které by bylo v původním plánu práce, tak se bohužel zavést a nakonfigurovat nezvládlo. Ideálně by poté mělo být možno spouštět automatické testy, statické analýzy kódu či jiné integrace.

Aplikace je v současné době dostupná za fakultní VPN [51] na adrese 10.38.6.201.

### 4.5 Testování aplikace

V rámci testování aplikace má velký podíl uživatelské testování, které odhalilo několik nedostatků aplikace, a získalo konstruktivní zpětnou vazbu s možnými



Obrázek 4.7: Detail průzkumu v nástroji LimeSurvey

cestami ke zlepšení. Co se týče automatického testování, tak to je spíše v počátcích, a je nutné na něm výrazněji zapracovat.

#### 4.5.1 Uživatelské testování

Tento typ testování aplikace se zaměřuje na to, jak aplikaci vnímá a používá koncový uživatel. V rámci práce bylo uživatelské testování provedeno s čtyřmi respondenty, autor práce během testování zástával funkci průvodce testováním. Všichni respondenti byli v době testování studenty Fakulty informačních technologií ČVUT, popř. byli čerstvými absolventy této fakulty. Testování probíhalo na Strahovských kolejích na pokoji autora práce. Zde byl každý z respondentů usazen za psací stůl, kde byly k dispozici dva monitory. Jeden na provádění samotného testu a druhý s testovacími scénáři ve formě PDF. Během celého testování byli respondenti a jejich akce nahrávány. Zaznamenávaly se tři druhy nahrávek: *webkamera*, *obrazovka* a *klávesnice*. *Webkamera* snímala respondenta, *obrazovka* snímala monitor respondenta, kde prováděl testy, a pohled *klávesnice* zaznamenával klávesnici, myš a pohyb rukou respondenta na mobilní telefon umístěný na stativu. Inspirací pro tyto tři pohledy byla La-

boratoř pro testování uživatelského rozhraní tzv. *UsabilityLab* [52], která při testování využívá téměř totožnou sestavu pohledů. Bohužel v rámci práce již nebyl dostatek času na hlubokou analýzu těchto nahrávek, zejména z hlediska vyhodnocení neverbálního vyjadřování respondentů v reakci na aplikaci. Také se v případě testování s prvním a posledním respondentem stalo, že kvůli technické chybě přestal nahrávat pohled *webkamera*, nicméně zvukový záznam je stále k dispozici ze zbývajících dvou pohledů. Všechny pořízené nahrávky jsou součástí příloženého média.

V úvodní fázi každého testování byl respondent seznámen s tím:

- co je to uživatelské testování a proč se dělá,
- že je testována aplikace, nikoliv respondent,
- že může volně komentovat, co právě dělá či co ho napadá,
- že je nahráván, nahrávka bude přiložena k výsledné práci, a zveřejněna (každý respondent s tímto vyslovil souhlas).

Během samotného testování byli respondenti krátce uvedeni do každého scénáře, a následně byla aplikace průvodcem uvedena do počátečního stavu pomocí konzolového příkazu `yes | sf doctrine:fixtures:load --group=basic` (`sf` je alias pro `php bin/console`). Po dokončení scénáře následoval krátký *debriefing*, neboli shrnutí právě provedeného scénáře, kde se většinou zopakovaly poznámky, které padly během provádění testu, nebo byly zmíněny nápady na zlepšení či přidání nových funkcí.

Celkem se prováděly čtyři scénáře:

- Proces registrace z pohledu neregistrovaného člena,
- Proces registrace z pohledu registrátora,
- Proces organizace akce z pohledu organizátora,
- Proces organizace akce z pohledu pomocníka.

Každý scénář obsahuje krátký popis, počáteční podmínky testu a seznam kroků, které je třeba provést k dokončení testu. Soubor s obsahem jednotlivých scénářů je součástí příloženého média. Celkové testování všech respondentů probíhalo tak, že první tři respondenti testovali v podstatě stejnou aplikaci, ale čtvrtý respondent testoval aplikaci, kde byly zahrnuty některé opravy na základě zpětné vazby od předchozích respondentů. V následujících odstavcích jsou shrnuty především podněty prvních tří respondentů.

Podněty k prvnímu scénáři, které zaznívaly, se často vztahovaly k doděláním vizuální stránky veřejné části webu, zejména registračního formuláře, kde třeba u některých polí chybělo vysvětlení či nebylo vidět tlačítko. Dále dva respondenti vyjádřili podiv, že byli po registraci ihned přihlášení, a také, že

po kliknutí na ověřovací odkaz byly přesměrováni na stránku registrace a nedostali zpětnou vazbu od aplikace.

Běhy druhého scénáře byly méně chybové než u toho prvního. Časté zmatení zde bylo kvůli odlišným textacím v rámci scénáře oproti aplikaci, či úvodní stránce administrace, která zůstala ve stejném stavu jako při počátečním vygenerování.

U třetího scénáře byly nejčastější nedostatky v použitelnosti polí pro výběr datumu a času, protože dovolovalo až 6-ti ciferný rok, který respondenti často vybrali, protože psali datum pomocí klávesnice. Dalším nedostatek se objevil u špatné či nulové viditelnosti tlačítka *Zpět na web*. Poslední vadou bylo nezvýrazňování odkazů v rámci popisu události na veřejné části webu.

Poslední scénář probíhal pravděpodobně nejlaději ze všech, a spíše při něm respondenti přicházeli nápady na zlepšení. Např. možnost notifikace organizátorů, že se někdo přidal/odebral z akce, možnost omezit počet účastníků nebo interní popis akce (pro organizátory).

Jak již bylo řečeno, poslední respondent testoval lehce upravenou verzi aplikace. Nová verze měla upravený registrační formulář po vizuální stránce (zvýrazněné tlačítko, české popisky, vysvětlivky), a také byla nahrazena úvodní stránka administrace. Důvodem bylo právě ověřit, že oprava nalezených chyb již nevyvolá jejich zmínku u dalšího respondenta. Po provedení testů s tímto respondentem lze pozorovat, že až na chyby, které budou pravděpodobně muset být řešeny v dalších fázích vývoje, tak ty opravené již nebyly zmíněny, a zároveň nebyly zmíněny nedostatky nové.

## 4.5.2 Automatické testování

Tento typ testování často realizuje cíle tzv. *regresního testování*, které kontroluje, zda nové změny v aplikaci nezapříčinily chyby v jiné části aplikace [53]. Práce obsahuje základ pro automatické testování ve složce *automated\_tests*, která obsahuje projekt testovacího frameworku cypress.io [54]. Práce obsahuje jeden UI test napsaný v tomto frameworku, a tím je registrace uživatele. Tento test jako uživatel dorazí na stránku Registrace, kterou vyplní a odešle registrační formulář. Určitě by stálo za snahu vytvořit i další průchody aplikací, obzvláště pokud by se vylepšovaly některé funkcionality. Bylo by také záhodno vyzkoušet jiné frameworky než cypress.io., protože je možné, že existují mnohem vhodnější než právě tento.

Kde má práce ovšem mnohem větší nedostatek, tak jsou unit a integrační testy, které kontrolují správnost tříd, například typu *Controller* či ty, které obsahují složitější logiku (např. *EventOrganizer*).

Před případným nasazením do produkčního prostředí a zejména zveřejněním by určitě neškodilo vyzkoušet zátěžové testy. Ano, webová aplikace fakultního klubu pravděpodobně nebude mít takovou návštěvnost jako např. internetové obchody, ale jistota je jistota.





---

## Závěr

Práce měla za cíl vytvořit webovou aplikaci pro klub FIT++. Základem bylo vypracování analýzy současného řešení, obdobných projektů a požadavků klienta. Z výstupů analýzy měla práce za úkol vybrat vhodné frameworky a nástroje pro implementaci a udělat návrh řešení. Dále bylo nutné implementovat aplikaci podle návrhu, provést nasazení aplikace a podrobit web uživatelským testům.

Práce zpracovala analýzu zabývající se několika oblastmi. První oblastí bylo současné řešení webové aplikace. Následovaly požadavky členů klubu a následně část podobných projektů. V této části bylo vybráno několik webových aplikací jiných studentských klubů a spolků. Předposlední oblast se zaměřila na možné webové technologie pro celou aplikaci, ta poslední na možná řešení online voleb v aplikaci.

V další fázi práce byl vypracován návrh implementace na základě provedené analýzy. Hlavním výstupem byl databázový model. Dalšími výstupy z této části byly use case a activity diagramy popisující případy užití a procesy v klubu. V poslední části této fáze došlo v rámci práce k rozhodnutí ohledně použitých technologií. Jako základní framework bylo vybráno Symfony s napojením MariaDB databáze a využitím platformy Docker pro konfiguraci prostředí. Jakožto specializovaný nástroj pro online volby byl vybrán nástroj LimeSurvey.

V rámci implementační fáze byla naprogramována Symfony webová aplikace s napojením na MariaDB databázi, která poskytovala správu uživatelů, organizaci a správu akcí, galerii fotografií a články na webu. Požadavek na online volby zatím nebyl dořešen. Aplikace byla nasazena server CloudFIT a bylo provedeno uživatelské testování. Z toho vyplynulo, že aplikace potřebuje dořešit grafické a funkční požadavky před nasazením do produkčního prostředí.



---

## Bibliografie

1. *Overview of Docker Compose* [online]. Palo Alto: Docker, ©2013-2021 [cit. 2021-06-21]. Dostupné z: <https://docs.docker.com/compose/>.
2. VALENTA, Michal; KALVODA, Tomáš; NOVÁČEK, Tomáš [online] [cit. 2020-11-26]. Dostupné z: <https://apakrychle.ninja/>.
3. LE THANH, Hung. *IT Coordinator ISC* [[ústní sdělení]. [N.d.]. audiohovor, 4.11.2020.
4. VESELY, Petr. *správce webu +/- FEL* [[ústní sdělení]. [N.d.]. audiohovor, 5.11.2020.
5. VLASÁKOVÁ, Anna. *správce webu ŠTUK* [[ústní sdělení]. [N.d.]. audioghovor, 5.11.2020.
6. HNYK, Petr. *správce webu eFDrive* [[ústní sdělení]. [N.d.]. audiohovor, 5.11.2020.
7. *Wordpress.com: Create a Free Website or Blog* [online]. Automattic [cit. 2020-11-26]. Dostupné z: <https://wordpress.com/>.
8. RAYMOND, Eric S. *WYSIWYG* [online]. Eric S. Raymond [cit. 2020-12-06]. Dostupné z: <http://catb.org/~esr/jargon/html/W/WYSIWYG.html>.
9. *Docker* [online]. Palo Alto: Docker [cit. 2021-04-19]. Dostupné z: <https://www.docker.com/>.
10. SPOIALA, Cristian; CALINCIUC, Alin; TURCU, Cornel; FILOTE, Constantin. Performance comparison of a WebRTC server on Docker versus Virtual Machine. *13th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 19-21, 2016*. 2016, s. 295–298.

11. SPOIALA, Cristian; CALINCIUC, Alin; TURCU, Cornel; FILOTE, Constantin. Performance comparison of a WebRTC server on Docker versus Virtual Machine. *13th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 19-21, 2016*. 2016, s. 295–298.
12. POTDAR, Amit M; D G, Narayan; KENGOND, Shivaraj; MULLA, Mohammed Moin. Performance Evaluation of Docker Container and Virtual Machine. *Procedia Computer Science*. 2020, roč. 171, s. 1419–1428. ISSN 1877-0509. Dostupné z DOI: <https://doi.org/10.1016/j.procs.2020.04.152>. Third International Conference on Computing and Network Communications (CoCoNet'19).
13. *Laravel - The PHP Framework For Web Artisans* [online]. Arkansas City: Laravel, ©2011-2021 [cit. 2021-04-20]. Dostupné z: <https://laravel.com/>.
14. *Symfony, High Performance PHP Framework for Web Development* [online]. Paris: SensioLabs, ©2005–2021 [cit. 2021-04-20]. Dostupné z: <https://symfony.com/>.
15. *Symfony, Laravel, CodeIgniter, CakePHP, Zend Framework - Prozkoumat - Trendy Google* [online]. Mountain View: Google [cit. 2021-06-10]. Dostupné z: <https://trends.google.com/trends/explore?cat=31&date=2015-01-01%202020-12-31&q=%2Fm%2F09cjc1,%2Fm%2F0jwy148,%2Fm%2F02qgdkj,%2Fm%2F09t3sp,%2Fm%2F0cdvjh>.
16. SHARMA, Rashmi. *8 Popular PHP Frameworks For Web Development in 2020* [online]. Eagle (Colorado): David Smooke, 2020 [cit. 2021-06-15]. Dostupné z: <https://hackernoon.com/8-popular-php-frameworks-for-web-development-in-2020-od3f38ez>.
17. NARAYAN, Vishnu. *10 Best PHP frameworks for website development in 2020* [online]. 2020 [cit. 2021-06-15]. Dostupné z: <https://meetadeveloper.com/10-best-php-frameworks-for-website-development-in-2020-ad1507a0f9f4>.
18. LAAZIRI, Majida; BENMOUSSA, Khaoula; KHOULJI, Samira; KERKEB, Mohamed Larbi. A Comparative study of PHP frameworks performance. *Procedia Manufacturing*. 2019, roč. 32, s. 864–871. ISSN 2351-9789. Dostupné z DOI: <https://doi.org/10.1016/j.promfg.2019.02.295>. cit. [2020-11-20].
19. LAAZIRI, Majida; BENMOUSSA, Khaoula; KHOULJI, Samira; LARBI, Kerkeb Mohamed; EL YAMAMI, Abir. A comparative study of laravel and symfony PHP frameworks. *International Journal of Electrical and Computer Engineering*. 2019, roč. 9, č. 1, s. 704. Dostupné také z: <http://ijece.iaescore.com/index.php/IJECE/article/view/11601>. cit. [2020-11-20].

20. SACOLICK, Isaac. *What is CI/CD? Continuous integration and continuous delivery explained* [online]. Massachusetts: IDG Communications, 2020 [cit. 2021-05-20]. Dostupné z: <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html>.
21. SCHRYEN, G.; RICH, E. Security in Large-Scale Internet Elections: A Retrospective Analysis of Elections in Estonia, The Netherlands, and Switzerland. *IEEE Transactions on Information Forensics and Security*. 2009, roč. 4, s. 729–744.
22. *Google Forms: Free Online Surveys For Personal Use* [online]. Mountain View: Google [cit. 2021-03-21]. Dostupné z: <https://www.google.com/forms/about/>.
23. *Forms Service / Apps Script / Google Developers* [online]. Mountain View: Google [cit. 2021-03-29]. Dostupné z: <https://developers.google.com/apps-script/reference/forms>.
24. *SurveyMonkey: The World's Most Popular Free Online Survey Tool* [online]. San Mateo: SVMK, ©1999-2021 [cit. 2021-03-18]. Dostupné z: <https://www.surveymonkey.com/>.
25. *API Docs / SurveyMonkey API Developer Portal* [online]. San Mateo: SVMK, ©1999-2021 [cit. 2021-03-29]. Dostupné z: <https://developer.surveymonkey.com/api/v3/>.
26. *LimeSurvey - Easy online survey tool* [online]. Hamburg: LimeSurvey, ©2006-2021 [cit. 2021-04-02]. Dostupné z: <https://www.limesurvey.org/>.
27. J. SERMERSHEIM, Ed. *Lightweight Directory Access Protocol (LDAP): The Protocol* [Internet Requests for Comments]. 2006. RFC, 1654. Dostupné také z: <https://www.hjp.at/doc/rfc/rfc4511.html>.
28. DOULIGERIS, Christos; MITROKOTSA, Aikaterini. DDoS attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*. 2004, roč. 44, č. 5, s. 643–666. ISSN 1389-1286. Dostupné z DOI: <https://doi.org/10.1016/j.comnet.2003.10.003>.
29. *What is Use Case Diagram?* [Online]. Kowloon, Hong Kong: Visual Paradigm International, ©2021 [cit. 2021-06-03]. Dostupné z: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>.
30. *What is Activity Diagram?* [Online]. Kowloon, Hong Kong: Visual Paradigm International [cit. 2021-06-03]. Dostupné z: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>.

31. Článek 17 EU obecné nařízení o ochraně osobních údajů "Právo na výmaz ("právo být zapomenut")" [online]. SecureDataService [cit. 2021-06-27]. Dostupné z: <https://www.privacy-regulation.eu/cs/17.htm>.
32. Twig - The flexible, fast, and secure template engine for PHP [online]. Paris: SensioLabs, ©2010-2021 [cit. 2021-04-20]. Dostupné z: <https://twig.symfony.com/>.
33. MariaDB [online]. Espoo: MariaDB Corporation, ©2021 [cit. 2021-04-20]. Dostupné z: <https://mariadb.com/>.
34. Dockerfile reference [online]. Palo Alto: Docker, ©2013-2021 [cit. 2021-06-21]. Dostupné z: <https://docs.docker.com/engine/reference/builder/>.
35. MORENO, Gerardo Fernández. Ger86/symfony-docker [online]. San Francisco: GitHub, ©2021 [cit. 2021-06-21]. Dostupné z: <https://github.com/ger86/symfony-docker>.
36. ZAVERI, Meet. What is boilerplate and why do we use it? Necessity of coding style guide [online]. San Francisco: Free Code Camp Inc. [cit. 2021-06-21]. Dostupné z: <https://www.freecodecamp.org/news/whats-boilerplate-and-why-do-we-use-it-let-s-check-out-the-coding-style-guide-ac2b6c814ee7/>.
37. Nginx [online]. San Francisco: Nginx [cit. 2021-06-21]. Dostupné z: <https://nginx.org/en/>.
38. Networking in Compose [online]. Palo Alto: Docker, ©2013-2021 [cit. 2021-06-21]. Dostupné z: <https://docs.docker.com/compose/networking/>.
39. What Is Load Balancing? [Online]. Seattle: F5, Inc. [cit. 2021-06-21]. Dostupné z: <https://www.nginx.com/resources/glossary/load-balancing/>.
40. ZAMMIT, Adam. Adamzammit/limesurvey-docker [online]. San Francisco: GitHub, ©2021 [cit. 2021-06-21]. Dostupné z: <https://github.com/adamzammit/limesurvey-docker>.
41. Symfony/recipes [online]. San Francisco: GitHub, ©2021 [cit. 2021-06-21]. Dostupné z: <https://github.com/symfony/recipes>.
42. CHRISTENSSON, Per. MVC Definition [online]. ©2021 [cit. 2021-06-25]. Dostupné z: <https://techterms.com/definition/mvc>.
43. What is CRUD? [Online]. New York City: Codecademy, ©2021 [cit. 2021-06-22]. Dostupné z: <https://www.codecademy.com/articles/what-is-crud>.
44. Skloňování jmen: oslovení ve správném pádu [online]. Morkovice-Slížany: Jan Kozák [cit. 2021-06-23]. Dostupné z: <https://www.sklonovani-jmen.cz/>.

45. SLABON, Jan. *Setasign/FPDF* [online]. San Francisco: GitHub, ©2021 [cit. 2021-06-21]. Dostupné z: <https://github.com/Setasign/FPDF>.
46. SLABON, Jan; KRESSE, Maximilian. *Setasign/FPDI* [online]. San Francisco: GitHub, ©2021 [cit. 2021-06-21]. Dostupné z: <https://github.com/Setasign/FPDI>.
47. *Titillium* [online]. Wilmington (Delaware): Font Squirrel, ©2021 [cit. 2021-06-23]. Dostupné z: <https://www.fontsquirrel.com/fonts/titillium>.
48. KURKA, Milan. *Jednotný vizuální styl Studentské unie ČVUT* [online]. 2019 [cit. 2021-06-23]. Dostupné z: <https://old.su.cvut.cz/cs/filebrowser/download/3950776>.
49. NERAVATH, Sachin. *Sachinchoolur/lightGallery* [online]. San Francisco: GitHub, ©2021 [cit. 2021-06-21]. Dostupné z: <https://github.com/sachinchoolur/lightGallery>.
50. MANNINO, Miro. *Miromannino/Justified-Gallery* [online]. San Francisco: GitHub, ©2021 [cit. 2021-06-21]. Dostupné z: <https://github.com/miromannino/Justified-Gallery>.
51. *VPN* [online]. Praha: GitHub [cit. 2021-06-27]. Dostupné z: <https://help.fit.cvut.cz/vpn/index.html>.
52. *Laboratoř pro testování uživatelského rozhraní (UsabilityLab)* [online]. Praha: Fakulta informačních technologií ČVUT, ©2021 [cit. 2021-06-25]. Dostupné z: <https://fit.cvut.cz/cs/veda-a-vyzkum/zazemi/laboratore/8359-laborator-pro-testovani-uzivatelskeho-rozhrani-usabilitylab>.
53. *What is Regression Testing? Definition, Tools, Method, and Example* [online]. Software Testing Help, ©2021 [cit. 2021-06-25]. Dostupné z: <https://www.softwaretestinghelp.com/regression-testing-tools-and-methods/>.
54. *JavaScript End to End Testing Framework | cypress.io* [online]. Atlanta: Cypress.io, ©2021 [cit. 2021-06-25]. Dostupné z: <https://www.cypress.io/>.





## Seznam použitých zkratk

<b>ČVUT</b>	České vysoké učení technické
<b>ISC</b>	International Student Club
<b>CI</b>	Continuous Integration
<b>CD</b>	Continuous Delivery
<b>WYSIWYG</b>	What You See Is What You Get
<b>HTML</b>	Hypertext Markup Language



---

## Obsah přiloženého média

database_design.mwb	databázový model ve formátu MySQL Workbench
diagrams.eapx	návrhové diagramy ve formátu Enterprise Architect Repository
FIT+_web_(Responses).xlsx	odpovědi dotazníkového šetření pro analýzu požadavků
readme.txt	stručný popis obsahu média
thesis	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
user_testing	podklady k uživatelskému testování
_ videos	adresář s audio-vizuálními záznamy uživatelského testování
_ user_testing_scenarios.pdf	dokument se scénáři použitý během uživatelského testování
web-pro-klub-fitpp	lokální kopie repozitáře se zdrojovými kódy implementace