FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Systém pro elektronické zajištění voleb v Mensa ČR
Student:	Maksim Shchukin
Vedoucí:	Ing. Tomáš Nováček
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

Cílem práce je navrhnout a naimplementovat řešení projektu elektronického volebního systému Mensy ČR. Postupujte v těchto krocích:
- Seznamte se s problematikou realizace voleb přes internet, a to jak po technické, kryptografické, tak procesní stránce,
- prostudujte existující řešení pro elektronické hlasování Vědecké rady FIT ČVUT,
- seznamte se procesem voleb v organizaci Mensa ČR a analyzujte jeho odlišnosti od elektronického hlasování Vědecké rady FIT ČVUT,
- prodiskutujte s Mensou, jak upravit či přepracovat elektronické hlasování Vědecké rady FIT ČVUT tak, aby vyhovoval jejich potřebám,
- navrhněte, realizujte a zdokumentujte úpravu či přepracování elektronického hlasování Vědecké rady FIT ČVUT dle analýzy z předchozího bodu,
- ve spolupráci s Mensou proveďte akceptační test řešení.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 12. prosince 2019

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Systém pro elektronické zajištění voleb v Mensa ČR

Katedra softwarového inženýrství
Vedoucí práce: Ing. Tomáš Nováček

4. června 2020

Poděkování

Chtěl bych poděkovat své rodině za podporu během mého života a svým kamarádům, kteří ho zpříjemňují. A hlavně Ing. Tomáši Nováčkovi za podporu v nejtěžším čase a za víru v můj úspěch, *Mense ČR* za podnětné téma a Zuzce Polákové za věnovaný čas. A zvláštně pak Eleně Sychugové za neocenitelnou pomoc.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principu 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 předpisu. Dále prohlašuji, že jsem s Českým vysokým učením technickým v Praze uzavřel dohodu, na jejímž základě se ČVUT vzdalo práva na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ustanovení § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisu.

V Praze dne 4. června 2020 . . . . . .

Odkaz na tuto práci

Shchukin, Maksim. *Systém pro elektronické zajištění voleb v Mensa ČR*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Tato bakalářská práce se věnuje problematice voleb přes internet. Její součástí je implementace volebního systému pro *Mensu ČR*. Obsahem práce jsou analýza systému *Baletka*, který je využíván za účelem elektronického hlasování Vědeckou radou FIT ČVUT, analýza procesu voleb *Mensy ČR* a jeho porovnání se systémem *Baletka*. Výsledkem je volební webová aplikace s použitím jazyků Java s frameworkem Spring a JavaScript s frameworkem Vue.js. Následně proběhlo akceptační testování společně s *Mensou ČR*.

Klíčová slova internet volby, elektronické volby, internet hlasování, elektronické hlasování, webová aplikace, internetová bezpečnost

Abstract

This bachelor thesis deals with the issue of elections via the Internet. It includes the implementation of an voting system for *Mensa ČR*. The content of the thesis is the analysis of the *Baletka* system, which is used for electronic voting by the FIT CTU Scientific Council, the analysis of the *Mensy CR* election process and its comparison with the *Baletka* system. The result is an election web application using Java with the Spring framework and JavaScript with the Vue.js framework. After all, acceptance testing was performed with *Mensa ČR*.

Keywords internet elections, electronic elections, internetové voting, electronic voting, elections, voting, web application, internet security, e-voting

Obsah

Úvod **1**
Cíl práce 2

1 Analýza **3**
1.1 Problém realizace voleb přes internet 3
1.1.1 Kritika 5
1.1.2 Kontrola kvality elektronických volebních systémů . . . 5
1.2 Analýza zabezpečení webových aplikací 6
1.2.1 OWASP 7
1.3 Analýza systému *Baletka* 12
1.3.1 Technologie 12
1.3.2 Typy uživatelů 12
1.3.3 Přihlášení 13
1.3.4 Obnovování hesla 13
1.3.5 Typy hlasování 13
1.3.6 Možnosti administrátora 14
1.3.7 Hlasování 15
1.3.8 Bezpečnost 15
1.3.9 Problémy, na které jsem narazil 15
1.4 Analýza voleb *Mensy ČR* 15
1.4.1 Analýza procesu voleb Mensy CR 15
1.4.2 Role 16
1.4.3 Požadavky 16
1.5 Porovnání procesu hlasování *Mensy ČR* a procesu systému *Baletka* 18

2 Návrh **23**
2.1 Zpracování požadavků 23
2.1.1 Role 23

2.1.2 Diagram případů užití . . . 23
2.1.3 Pokrytí funkčních požadavků . . . 27
2.1.4 Proces voleb nového systému . . . 28
2.1.5 Doménový model . . . 28
2.1.6 Wireframy . . . 29
2.2 Návrh na vylepšení . . . 29
2.2.1 Návod . . . 29
2.2.2 Odmítnuté ideje . . . 30

3 Implementace **33**
3.1 Architektura . . . 33
3.2 Technologie . . . 33
3.3 Serverová část . . . 34
3.3.1 Datová vrstva . . . 34
3.3.2 Doménová vrstva . . . 34
3.3.3 Aplikační vrstva . . . 35
3.3.4 Autentifikace . . . 35
3.3.5 Ukládání hlasu voliče . . . 36
3.3.6 Generování výsledků . . . 36
3.3.7 Mensovní API . . . 36
3.4 Klientská část . . . 37
3.4.1 Komunikace se serverem . . . 37
3.4.2 Přihlášení . . . 37
3.4.3 Hlasování . . . 37
3.4.4 Nastavení systému . . . 38
3.4.5 Správa kandidátů . . . 38
3.4.6 Správa výsledků . . . 39
3.5 Testování . . . 39
3.6 Možné vylepšení . . . 40

Závěr **43**

Bibliografie **45**

A Seznam použitých zkratek **49**

B Obsah přiloženého CD **51**

Seznam obrázků

1.1 Diagram kritérií a podkritérií technologie pro elektronické hlasování 6
1.2 Diagram aktivit před volbami *Mensy ČR* 19
1.3 Diagram aktivit před hlasováním v systému *Baletka* 19
1.4 Diagram aktivit voleb *Mensy ČR* 20
1.5 Diagram aktivit hlasování v systému *Baletka* 20
1.6 Diagram aktivit po volbách *Mensy ČR* 21
1.7 Diagram aktivit po hlasování v systému *Baletka* 21

2.1 Diagram aktérů 24
2.2 Diagram případů užití 25
2.3 Tabulka pokrytí funkčních požadavků 27
2.4 Diagram aktivit nového systémů před internetovými volbami . . . 28
2.5 Diagram aktivit nového systémů během internetových voleb 29
2.6 Diagram aktivit nového systémů po internetových volbách 30
2.7 Doménový model 31

3.1 Diagram entit 35
3.2 Úvodní přihlašovací stránka. 38
3.3 Základní chybové oznámení. 39
3.4 Oznámeni o druhem pokusu se zúčastnit voleb. 39
3.5 Stránka pro zadaní kódu. 40
3.6 Stránka hlasování. 40
3.7 Stránka pro správu kandidátů. 41
3.8 Stránka pro správu výsledku voleb. 42

Seznam tabulek

1.1 Výsledné hodnocení hlasovacích systémů z [8] podle jejich metodologie 6

Úvod

Současnost neboli 21. století je věkem internetu. Jeho růst a vývoj dal lidstvu spoustu nových možností – okamžitý přístup k informacím/službám, které usnadňují každodenní život. V dnešní době se internet dostal do žárovek, konvic a dalších věcí, které používáme každý den. Přes internet můžete objednat pizzu a seznámit se s člověkem, který bydlí tisíce kilometrů daleko. Za 40 let své existence se internet dostal skoro do všech aspektů našeho života. Ne všechny oblasti jsou však online – například hlasování ve státních volbách.

Existující běžná řešení mají spoustu problémů. Potřebují váš čas, protože předat svou volbu lze jenom v určitých místech, vyžadují práci spousty lidí a obrovské množství státních peněz. A navíc nejsou vůbec bezpečné, přestože se pořád navrhují nové způsoby zabezpečení. Lidský faktor zvyšuje možnost chyby a snižuje bezpečnost.

Logickým řešením by bylo vynechat lidi z vnitřních procesů voleb použitím úplného elektronického systému. To by mohlo vyřešit tento problém a ušetřit čas i státní peníze. Bohužel není všechno tak snadné, i když to tak vypadá. Existuje spousta problémů, které je třeba vyřešit před reálným použitím podobných systémů: problém získání identity (jinak nelze identifikovat uživatele), zabezpečení serverů a dalších zařízení (aby nedocházelo ke ztrátě důvěry k uživateli).

Samostatně implementovat systémy pro řešení popsaných problémů není v rámci bakalářské práce možné. Ale díky *Mense ČR*, která je zákazníkem volebního systému pro své účely, mohu zkusit vytvořit vlastní systém, který se pokusí vyřešit zjednodušený problém elektronických voleb existujícími technologiemi. Nebude se hodit pro celý stát, ale bude vyhovovat potřebám menší společnosti.

Tato práce se skládá z 3 kapitol. První kapitola se věnuje analýze a skládá se z 5 částí. Úvodní část se věnuje problémům elektronických volebních systémů, druhá se věnuje analýze základu bezpečnosti webových aplikací, třetí část se věnuje systému *Baletka* (byl vytvořen Fakultou informačních technolo-

gií ČVUT v Praze pro volby do vědecké rady fakulty). Analýzou bezpečnostní strany tohoto systému se zabýval Petr Nohejl ve své bakalářské práce [1]. Čtvrtá část je zaměřena na analýzu procesu voleb *Mensy ČR*. V páté části jsou srovnávány procesy voleb *Mensy ČR* a systému *Baletka*. Ve druhé kapitole je návrh vlastního systému, který je výsledkem předchozí analýzy. Ve třetí kapitole jsou popsány jednotlivé části implementace.

Cíl práce

Hlavním cílem této bakalářské práce je implementace webové aplikace pro hlasování pro *Mensu ČR*. *Mensa ČR* je zákazníkem tohoto systému, a proto neoddělitelnou částí mého projektu je komunikace s ní a analýza jejích požadavků. V závěru ve spolupráci s *Mensou* provedu uživatelský a akceptační test řešení.

Dalším cílem je analýza existujícího systému *Baletka*. A další porovnání procesu voleb implementovaného v *Baletce* s procesem voleb *Mensy ČR*. Nedílnou součástí práce bude též analýza bezpečnosti webových aplikací a podobných systémů pro hlasování.

Kapitola 1

Analýza

Tato kapitola je věnována analýze nutné pro návrh a implementace výsledného systému. Je rozdělena na pět částí. První je věnována problematice realizace voleb přes internet. Druhá se zabývá seznámením se základy bezpečnosti webových aplikaci. Třetí sekce analyzuje systém *Baletka*. Ve čtvrté části jsou popsány požadavky *Mensy ČR* na nový volební systém. Pátá sekce porovnává požadavky *Mensy ČR* se systémem *Baletka*.

1.1 Problém realizace voleb přes internet

Elektronické hlasování je hlasování, během něhož se používají elektronické stroje pro sčítání hlasů voličů nebo pro celé hlasování. Tedy pro elektronické volby se používají speciální stroje (EVM[1]) nebo počítače připojené na internet. Zajímá mě druhý způsob realizace elektronických voleb. Takové volby budu nazývat internetovými volbami.

Ty mají několik výhod oproti standardním řešením s použitím fyzických hlasovacích lístků a spočítáním hlasů ručně nebo jiným způsobem.

- Snadno rozšiřitelný

  Není nutné vytvářet dedikovaná místa pro hlasování, protože všechny operace řeší stroj. Příklady jsou počítání hlasů nebo získání identity voliče. A proto možný počet účastníků voleb záleží na výkonu serveru, na kterém běží aplikace.

- Komfortní pro uživatele

  Uživatel se může zúčastnit voleb ze svého počítače, tedy kdykoli a kdekoli. Teoreticky to může zvýšit počet účastníků voleb, ale podle zkoumání ve Švýcarsku [2] to nebylo ovlivněno.

[1] Electronic voting machines

- Bezpečnost proti lidskému faktoru

  Všechny důležité operace spravuje stroj. Z toho plyne, že pokud je dobře implementován, počet chyb a zneužití[2] klesá k nule.

- Cena

  Velkou výhodou internetových voleb jsou jejich nejnižší náklady na uspořádání v porovnání s jinými způsoby voleb. Autoři [3] ve své analýze dosáhli výsledků, které tvrdí, že internetové hlasování je minimálně dvakrát levnější než ostatní způsoby voleb.

Elektronické hlasování na rozdíl od jiných citlivých systémů (např. bankovnictví) má několik vlastností. Z těch formulovaných v [4] bych rád vyčlenil následující:

- Vliv na společnost

  Volby byly vymyšleny, aby společnost mohla řešit spor bez násilí. Tudíž kompromitace volebního procesu má fatální následky, protože společnost spoléhá na integritu voleb a úspěšné ukládání každého hlasu.

- Identita

  Pro hlasování je důležité, komu patří hlas, protože jedním ze základních pravidel hlasování je účast každého voliče pouze jednou.

- Dostupnost

  Volby se konají v některé určité definované době. A útoky typu DDOS mohou zablokovat přístup k systému, tudíž oprávnění voliči nebudou mít možnost zúčastnit se voleb včas.

- Autentifikace a anonymita

  Ale paradoxně hlasovací systém potřebuje jednoznačně určit voliče a také ochránit jeho anonymitu.

- Monitorování a kontrola správnosti uložení hlasu

  Standardní papírové hlasování míní, že volič projeví svou volbu hlasovacím lístkem. To znamená, že hlasovací lístek má nějakým způsobem skrýt jeho volbu, ale zároveň zachytit fakt hlasování. V opačném případě může dojít k prodávání hlasů a donucení k určitému výběru. To platí také pro internetové volby. Takže kontrola správnosti uložení hlasu musí chránit identitu uživatele současně s kontrolou integrity hlasu.

[2] Ve smyslu zločinné manipulace s hlasovacími lístky a jiné pokusy ovlivnit výsledky hlasování.

1.1.1 Kritika

Velké množství specialistů kritizuje existující systémy a celkovou možnost elektronických voleb. Existuje spousta faktorů, které musí být zajištěny, aby hlasování bylo důvěryhodné. Stávající systémy nejsou dokonalé [5]. Jsou závislé na softwarové části, jejíž velikost může být několik set tisíc řádků kódů. Testování tak velkých programů je obrovskou prací, protože by mělo být zajištěno jejich dokonalé fungování, aby byly důvěryhodnými. Kromě toho by měla být zaručena skoro absolutní bezpečnost, ale pravděpodobnost, že bude opravdu zajištěna, je velmi nízká. Kromě toho nelze zapomenout na bezpečnost hardwaru, na kterém bude běžet software a jeho odolnost proti chybám. Avšak zajistit to, je mnohem složitější než pro software.

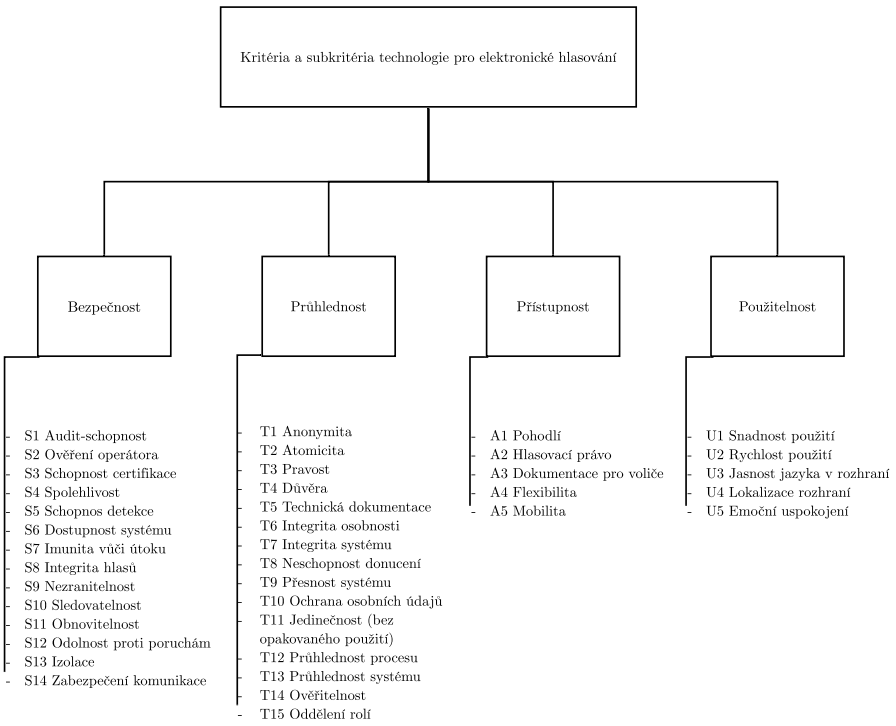
Ještě jedním důležitým faktorem je žádost zákazníků udržovat podobný systém. Je zřejmé, že se hlasovací systém nepoužívá každý den. Ale kvůli tomu, že má být co nejkvalitnější, potřebuje dlouhodobou údržbu aktuálnosti jak softwaru, tak hardwaru. Ale i při obezřetném dodržování všech faktorů může nastat nevypočitatelný jev, který může ovlivnit volby, jako v Belgii v roce 2003. Podle [6] bylo objeveno 4 096 hlasů navíc, ale žádný problém v systému nebyl komisí nalezen.

Všechny popsané komplikace ničí důležitou vlastnost internetových voleb, jako je cena, když se žádá o poskytnutí kvalitního systému. Implementace, návrh a testování takového systému bude drahé. A proto se objevuje tolik kritik existujících systémů – nejsou dost drahé. Avšak ve sporech o internetových volbách lidé zapomínají na jejich alternativu. Podle mého názoru není papírové hlasování vůbec dokonalé. Není tajemstvím, že je zneužíváno [7]. Takže někdy lidé ohrožují hlasování mnohem více než počítače. Jestli dát pozor na všechny podmínky důvěryhodného systému, když jejich dokonalá splnitelnost je stejně nerealizovatelná.

1.1.2 Kontrola kvality elektronických volebních systémů

Důležité a citlivé systémy potřebují seriózní kontrolu kvality. Samozřejmě to patří k volebním systémům. Jedna metodika byla propracována pro testování volebních systémů pro volby do portugalského parlamentu [8]. Diagram na obrázku 1.1 je přepracováním tabulky z [8]. Tato kritéria a podkritéria byla ohodnocena pomocí Saatyho metody[3] a každému podkritériu byla vypočtena jeho hmotnost. Pak lze vypočítat hodnocení každého z kritérií podle hodnocení jejich podkritérií. V rámci [8] bylo provedeno hodnocení čtyř různých systémů. Výsledky jsou shrnuty v tabulce 1.1. Důležité je říct, že systém Novabase patří k elektronickým volbám přes internet, zatímco ostatní k elektronickým volbám s použitím EVM.

[3] Analytical Hierarchy Process (AHP). Tato metoda je založena na porovnání důležitosti každé dvojice dílčích kritérií [9].



Obrázek 1.1: Diagram kritérií a podkritérií technologie pro elektronické hlasování

Tabulka 1.1: Výsledné hodnocení hlasovacích systémů z [8] podle jejich metodologie

	UNISYS	INDRA	MULTICERT	NOVABASE
Bezpečnost	4,2	4,1	2,6	3,6
Průhlednost	4,2	4,3	3,2	3,0
Použitelnost	4,2	3,9	2,7	3,8
Přístupnost	3,7	3,3	3,5	3,6

1.2 Analýza zabezpečení webových aplikací

Jednou z nejdůležitějších částí aplikace pro hlasování je její zabezpečení. Výsledkem této bakalářské práce by měla být webová aplikace a navíc jsem vyřešil rozebrat základy jejich zabezpečení.

1.2.1 OWASP

OWASP je zkratka *The Open Web Application Security Project*. To je fond, který se zabývá zvýšením bezpečnosti aplikací včetně webových. Pro dosažení tohoto cíle vytvaruje nástroje, projekty, dokumenty, pomocné články a uspořádá fóra. Jeho sloganem je „Společně zlepšíme bezpečnost softwaru.“[10] Za 19 let, kdy OWASP existuje, se stal standardem v oboru bezpečnosti. To dokazuje počet citací v odborných pracích [11].

Jedním z jeho projektů je OWASP Top Ten [12]. Popisuje deset nejdůležitějších bezpečnostních rizik pro webové aplikace.

Injection Injection neboli injekce je typ útoku, při kterém jsou nedůvěryhodná data předána interpretovi jako část příkazu. V takovém případě ho mohou škodlivá data oklamat a dojde ke spuštění ničivého kódu nebo k poskytnutí tajných dat neoprávněné osobě. Příklady injekce jsou SQL injekce, NoSQL injekce, OS injekce a LDAP injekce. Zabránit injekcím lze několika způsoby. Podle [13] jimi jsou:

- Oddělení dat od příkazu, buď pomocí bezpečného API, které nepoužívá interpret přímo, nebo použitím nástrojů pro ORM.
- Kontrola dat na serveru získaných od klientů.
- Výměna všech speciálních znaků v příkazu na speciální znaky interpreta.
- Omezení velikosti výsledku interpreta.

Tyto způsoby nezajistí úplnou bezpečnost proti injekčním útokům, pokud se nepoužívají spolu. Protože u různých interpretu existuji své výjimky.

Rozbitá autentifikace Často se stává, že funkce zodpovědné za autentifikace a řízení sezení[4] jsou nastaveny nesprávně, což může mít hrozivé následky. Hackeři mají přístup k milionům platných přihlašovacích údajů, které ve spojení s automatickým zkoušením mohou vést k přihlášení k cizímu účtu. Takže v případě, že účet patří správci systému, vede ke kompromitaci celého systému. Podle [14] lze takové útoky překazit:

- Zajistit vícefaktorovou autentifikaci.
- Vyhýbat se používání standardních přihlašovacích údajů.
- Omezit slabá hesla a používat politiku pro délku a složitost hesel.
- Používat stejné chybové oznámení pro všechny uživatele.

[4] Session management

- Omezit počet nesprávných pokusů přihlášení nebo jejich častost. Současně s tím logovat všechny takové pokusy a v případě detekce zneužití oznámit správci systému.
- Používat standardní bezpečné manažery sezení, které by se staraly o včasné změny a mazání *sessionId*.

Vystavení tajných dat Hodně webových aplikací a API nezabezpečují citlivá data správně. Útočníci je mohou ukrást nebo upravit a pak zneužít. Citlivá data mohou být kompromitována bez speciálního zabezpečení, jako šifrování a zvýšené opatření při komunikaci s klientem. Pro zabezpečení proti takovým útokům podle [15] by mělo být zajištěno:

- Klasifikace dat, se kterými pracuje aplikace. Zjistit, která data jsou citlivá, lze ze zákonu o ochraně soukromí, požadavku regulátoru, anebo z potřeb společnosti.
- Ukládání citlivých dat jenom, když je to nutné. A mazat je ze systému co nejrychleji.
- Šifrování citlivých dat.
- Zajištění používání moderních a silných algoritmů a protokolu.
- Pro šifrování předání dat používat moderní algoritmy jako TLS s šifrováním PFS. Protokol šifrování by měl být zvolen serverem a použita bezpečná nastavení. Vynutit používání šifrovacích direktiv jako HSTS.
- Vypnutí caching pro odpovědi, které obsahují citlivá data.
- Uložení hesel po hašovaní se solí, jako Argon2, script, bcript, PBKDF2.
- Nezávislost testování efektivity nastavení.

XML External Entities (XXE) Často se stává, že staré nebo špatné nastavení vyhodnocují odkazy na externí entity v rámci XML souboru. Externí entity mohou být využity k získání přístupu k vnitřním souborům pomocí správce URI souboru, ke sdílení vnitřních souborů, skenování vnitřních portů, spuštění kódu a DDOS útokům. Podle [16] lze takové útoky překazit:

- Používat méně komplikované formáty jako JSON, které usnadňují serializace.
- Používat aktuální verze softwaru.
- Vypnout vnější entity a zpracování DTD pro všechny XML parsery.
- Implementovat pozitivní filtrace vstupních dat.

- Používat SAST nástroje, které mohou detekovat XXE v zdrojových kódech.

Porušená kontrola přístupu Omezení přístupu uživatelů do různých částí systému je často špatně nakonfigurováno. Pomocí takových bezpečnostních chyb mohou útočníci získat přístup k funkcionalitám nebo datům, ke kterým by mít přístup neměli. Pro zabezpečení proti takovým útokům podle [17] by mělo být zajištěno:

- Implementace kontroly přístupu je pouze jedna a opakovaně je použita v celé aplikaci.
- Aplikace má jasně definovat vlastnictví dat a zakázat jejich změnu a jiné manipulace ostatními osobami.
- Unikátní omezení poskytovaná zákazníkem, by měla být zajištěna na úrovni doménového modelu.
- Zamezení přístupu k seznamu souborů webového serveru a kontrola, že žádná metadata nejsou ve veřejné složce.
- Oznámení správce systému o opakovaných neúspěšných pokusech přihlásit se.
- Omezení rychlosti zpracování API pro minimalizace rizik spojených s automatickým úrokovým softwarem.
- JWT[5] tokeny mají být anulovány po odhlášení uživatele.

Špatné nastavení bezpečnosti Je to jeden z nejčastěji se vyskytujících problémů. Častými příčinami jsou nebezpečné výchozí nastavení, nedokončené nebo místní nastavení, vystavené uchování dat, nesprávná konfigurace HTTP, podrobné chybové hlášení, které obsahuje citlivá data. Všechny softwarové části systému by měly být nejen bezpečné nastaveny, ale i často aktualizovány. Podle [18] lze takovým útokům zabránit:

- Zajistit opakovatelnost procesu zabezpečení pro snadné nasazení na různé prostředí.
- Zbavit se nepoužitých funkcí a knihoven.
- Dodržovat aktualitu nastavení zabezpečení a podle známých zranitelností.
- Rozdělit aplikace na různé části.

[5] JSON Web Token

- Použít security headers při komunikaci s klientem.
- Automatizovat proces kontroly efektivity bezpečnostního nastavení.

Cross-Site Scripting XSS Ohrožení typu XSS vzniká, když aplikace bez kontroly doplní novou webovou stránku nedůvěryhodnými daty, nebo upravuje už existující stránku pomocí API browseru, který může generovat HTML nebo JavaScript kód podle dat, které dostal od uživatele. To dovoluje spuštění škodlivého kódu, který může ukrást sezení uživatele, změnit uživatelské rozhraní stránky nebo přesměrovat na jiné webové stránky. Pro zvýšení zabezpečení proti takovým útokům podle [19] je vhodné zajistit:

- Použití frameworku, který je odolný proti XSS útokům na úrovni architektury.
- Omezit používání nedůvěryhodných dat.
- Použití citlivého prostředí ke kódování, když provádí úpravu stránky u klienta.
- Použití CSP jako *defense-in-depth*[6] zmírňující XSS útoky. Ale důležité je říct, že efektivita záleží na existenci jiných zranitelností aplikace, které povolí škodlivý kód nebo knihovnu části systému.

Nebezpečné deserializace Nebezpečné deserializace často vedou ke spuštění kódu vzdáleně. I když tomu tak není, stále mohou hrozit injekcemi, zvýšením práv přístupu nebo tzv. replay útoky. Jediná architekturní šablona, která může pomoct s takovým problémem, je odmítat objekty z nedůvěryhodných zdrojů nebo používat serializační média, jež povolují pouze primitivní datové typy. Jinak lze vyzkoušet instrukce z [21]:

- Implementovat integritní kontrolu všech serializovaných objektů.
- Izolovat kód, který provádí deserializace objektů, ve vlastním prostoru s nižším právem přístupů, pokud je to možné.
- Oznamovat o chybách deserializace.
- Kontrolovat nebo zakázat komunikace deserializačních služeb s internetem.
- Kontrolovat, jak často uživatel deserializuje, a v případě potenciálního nebezpečí oznamovat správci systému.

[6] Použití více bezpečnostních vrstev v aplikaci[20].

Používání softwaru, který obsahuje známé zranitelnosti Software jako knihovny, frameworky a další softwarové jednotky mají stejná přístupová práva jako aplikace, která běží a potřebuje je. Důsledky využití takových zranitelností jsou hrozivé. Konečná bezpečnost aplikace bude oslabena a existuje šance ztráty dat nebo poruchy serveru. Pro snížení rizika podobných problémů lze [22]:

- Odstranit nepoužívané závislosti, soubory, knihovny.
- Sledovat nové hrozby a zranitelnosti a dodržovat projekt v aktuálním stavu.
- Používat knihovny z důvěryhodných zdrojů. Pro nejlepší výsledek lze také kontrolovat, aby obsahovaly podpis pro kontrolu integrity používaných komponent.
- Kontrolovat poskytovatele komponent, jestli dodržují jejich bezpečnost v aktuálním stavu a vytváří bezpečnostní záplaty pro starší verze. V případě, že nějakou záplatu nelze instalovat, jedním z řešení bude monitorování známé zranitelnosti.

Nedostatečné logovaní a kontrolování Kvůli nedostatečnému logování a kontrolování současně s chybějícím nebo slabým začleněním do incidentu mohou útočníci provádět mnohem delší a hlubší útoky na více systémů, manipulovat, stahovat a mazat data. Častá zkoumání porušení ukazují, že doba, kdy došlo k porušení, je více než 200 dnů. Někdy porušení detekují externí strany místo interního logování a monitorování. V souladu s citlivostí a cenou dat, se kterými pracuje aplikace, lze dodržovat následující instrukce [23]:

- Zaručit, že každé oznámení o neúspěšné serverové validaci, přihlášení nebo kontrole práv přístupu je srozumitelné a umožňuje identifikovat nebezpečný účet a je uchováváno s dostatečnou dobou pro zjištění problému.
- Zaručit, že vygenerované oznámení může být zpracováno centrální logovací službou.
- Zaručit, že integrita důležitých transakcí je kontrolována.
- Zaručit, že monitorování je efektivní a umožňuje rychlou reakci v případě zneužití.
- Zaručit použití speciálních instrukcí, které popisují postup v případě nějakého incidentu.

1.3 Analýza systému *Baletka*

Baletka je systém vytvořený Fakultou informačních technologií ČVUT v Praze pro provádění elektronického hlasování členů vědecké rady. V této kapitole jsem analyzoval poslední stabilní verze systému 0.9.7. Nebyla mi poskytnuta dokumentace systému kromě postupu pro instalace, a proto informace o *Baletce* byly získány ze zdrojových kódů a z existující bakalářské práce [1].

1.3.1 Technologie

Pro implementaci serverové části systému byl použit jazyk Ruby s frameworkem *Ruby on Rails*. *Ruby on Rails* nebo prostě rails je framework pro vytváření webových aplikací v jazyce *Ruby*. Poskytuje rozhraní pro implementace databáze, webových služeb a webových stránek. Zároveň byly použity různé takzvané gemy. *Gemem* se označuje knihovna nebo balík pro *Ruby*, které poskytuje manager knihoven a balíků *RubyGems*.

Pro implementace klientské části *Baletky* byly použity *JavaScript*, *SCSS*, *CoffeeScript*, *JQuery*.

Zabezpečení připojení bylo zajištěno protokolem HTTPS. Pro autorizace zaměstnanců byl použit autorizační server FIT ČVUT, který je založen na protokolu *Oauth2* a je přístupný na `auth.fit.cvut.cz`. Pro jeho podporu v rámci Baletky byl použit gem *Devise* s modulem *Omniauthable*, který zajistí podporu *OmniAuth* – jedné z možných implementací *Oauth2*. Také gem *Devise* byl použit pro různé aspekty komunikace mezi uživatelem a serverem, buď k přihlášení, nebo obnovování hesla. Zabezpečení proti spamu a zneužití bylo zajištěno službou *reCAPTCHA*. *reCAPTCHA* je bezplatná služba, která chrání webové stránky před spamem a zneužitím. Používá analýzu rizik a dynamické úkoly, aby zabránila automatizovanému softwaru zapojit se do zneužívajících aktivit na webových stránkách, na kterých byl použit. Ale nechá k nim přístup platným uživatelům.

1.3.2 Typy uživatelů

V rámci aplikace existují tři typy uživatelů: Pověřené osoby nebo-li administrátoři, uživatel z ČVUT nebo-li interní volič a externí volič.

Administrátorem je uživatel, který může spravovat volby a jejich průběh. Uživatelská jména administrátorů jsou uložena v odpovídajícím konfiguračním souboru. Administrátor má přístup k následujícím funkcím: správa voleb, správa šablonu voleb, správa uživatelů. V dané implementaci *Baletky* (0. 9. 7) se administrátor nemůže zúčastnit voleb. Administrátor musí mít příslušnost k ČVUT.

Interní a externí uživatelé jsou uživatelé, kteří se mohou zúčastnit voleb, a to jejich hlavní funkce. Hlavní rozdíl mezi interním a externím uživatelem je příslušnost k ČVUT. Právě z toho plynou různé postupy přihlášení.

1.3.3 Přihlášení

V *Baletce* se používá dvoufaktorové přihlášení. První fáze se dělí na 2 typy. V případě, že uživatel má účet ČVUT, může použít tlačítko „Přihlášení pro zaměstnance ČVUT“, které ho přesměruje na stránky `https://auth.fit.cvut.cz`. To je adresa autentizačního serveru, který se používá pro přihlášení do většiny služeb FIT ČVUT. Pro přihlášení interního uživatele má administrátor vytvořit účet. Pro úspěšné dokončení této operace potřebuje administrátor předem vědět uživatelské jméno ČVUT a telefonní číslo uživatele. V případě, že uživatel nemá účet ČVUT, používá pro přihlášení email a heslo. Pro přihlášení externího uživatele má administrátor vytvořit účet. Pro úspěšné dokončení této operace potřebuje administrátor předem vědět email, plné jméno nového uživatele a jeho telefonní číslo. Telefonní číslo uživatelů pak bude použito pro druhou fázi přihlášení. Po úspěšném dokončení první fáze přihlášení, uživatel dostane SMS s kódem, který zadá do odpovídajícího pole. V případě, že kód byl zadán správně, se uživatel úspěšně přihlásí do systému.

1.3.4 Obnovování hesla

Přihlášený externí uživatel může změnit své heslo. K tomu potřebuje vybrat odpovídající položku menu v horním levém rohu stránky. Po zadání starého hesla a vyplnění *reCAPCHA* se uživatel dostane na stránku obnovování hesla. Aplikace kontroluje slabost hesla a také ji ukazuje uživateli pomocí barevného indikátoru. V případě, že uživatel zadal nové heslo dvakrát stejně, dostane se zpět na stránku běžících voleb.

Pokud uživatel své heslo zapomněl, může využít tlačítko, které ho převede na stránku, kde po vyplnění svého emailu a *reCAPCHA* dostane email s dalším postupem.

1.3.5 Typy hlasování

V *Baletce* je typ hlasování definovaný dvěma parametry:

- dle možnosti přerušení
  - standardní,
  - zrychlené.
- dle anonymity
  - tajné,
  - veřejné.

Pokud je hlasování standardní, nelze ho zastavit dříve než byl nastaven konec hlasování. Jinak je zrychlené. Tudíž ho lze zastavit před dosažením konce hlasování na stránkách spuštěných hlasování.

V případě, že je potřeba zjistit, jak někdo hlasoval, lze vybrat veřejné hlasování. Soubor s výsledky voleb pak bude obsahovat informace o rezultátech i hlasech voličů, zvlášť pro každý dotaz a seznam voličů, kteří se voleb zúčastnili. V případě, že je potřeba informace o hlasech voličů skrýt, lze v *Baletce* vytvořit tajné hlasování. Ve výsledném souboru pak bude jenom informace o výsledcích pro každou otázku a seznam voličů, kteří se zúčastnili voleb.

Avšak při zkoumání *Baletky* bylo zjištěno, že implementace serverové části se liší od klientské části a úmyslně omezuje zobrazení standardních hlasování. Z jakého důvodu tomu tak je, není známo.

1.3.6 Možnosti administrátora

V rámci *Baletky* má administrátor několik pravomocí. Nejdůležitějšími jsou správa hlasování a správa uživatelů. Na rozdíl od běžného uživatele se administrátor nemůže zúčastnit se voleb.

Správa uživatelů Aby uživatel se mohl zúčastnit voleb, administrátor ho potřebuje přidat do systému. Proto na stránce správy uživatelů jsou dva tlačítka pro přidání nových uživatelů. Administrátor rovněž může odstranit uživatele, avšak to není povoleno v době běhu jakéhokoliv hlasování. Také může upravovat údaje uživatelů. Například u externistů je povolená editace jména, emailu a telefonního čísla, ale u uživatelů z *ČVUT* jenom telefonní číslo. To je zřejmě, protože používají účet *ČVUT* pro přihlášení. Jeho úprava proběhá na `usermap.cvut.cz`

Správa hlasování *Baletka* byla vytvořena se zaměřením na automatizace. Administrátor potřebuje vytvořit pouze hlasování, systém ho pak spustí a zastaví samostatně.

Vytváření nového hlasování není komplikované. Pro rychlý přístup bylo přidáno tlačítko do horního menu. Hlasování v *Baletce* má další údaje: název, popis, čas spuštění a zastavení, zda je standardní nebo zrychlené, tajné nebo veřejné. Poslední část hlasování tvoří dotazy. Uložené hlasování lze upravovat, dokud ještě nebylo spuštěno. Během hlasování jsou všechny manipulace zakázány. Po dokončení hlasování může administrátor stáhnout PDF soubor s výsledky.

Počet dotazů nemůže být v hlasování menší než 1. Maximální počet dotazů není omezen. Hlasování má předdefinovaný formát. Změna formátu dotazu není možná bez úpravy zdrojových kódů.

Baletka podporuje možnost vytvořit šablonu pro hlasování. Pro uživatele to znamená, že mohou jednou vytvořit šablonu a na základě toho opakovaně vytvářet hlasování. V případě potřeby lze hlasování, vygenerované z šablony, před uložením upravit.

1.3.7 Hlasování

První stránka, která se otevře po přihlášení uživatele, je seznam aktuálních hlasování. V případě, že se uživatel hlasování nezúčastnil, může ho rozkliknout a zapojit se. Jinak se objeví hlášení, že už hlasoval.

Hlasování má předdefinovaný formát. Všechny dotazy jsou na jedné stránce. Každý dotaz má tři varianty odpovědí: kladný, záporný a zdržet se.

Divně vypadá řešení o zobrazení stejné první stránky po přihlášení administrátorů. K tomu jsou dva důvody. Za prvé se administrátor nemůže zúčastnit voleb. Za druhé je správa hlasování základní a nejdůležitější funkcionalita pro administrátora.

1.3.8 Bezpečnost

Analýzou bezpečnostní strany *Baletky* se zabýval Petr Nohejl ve své bakalářské práci [1]. Ve své bakalářské práci samozřejmě nestihnu rozebrat bezpečnost *Baletky* na stejné úrovni.

1.3.9 Problémy, na které jsem narazil

Pro analýzu *Baletky* mi fakulta poskytla její zdrojový kód. Při pokusu sestavit aplikace mi vyskočila chybová hlášení. Aktualizoval jsem gemy, ale některé nebyly kompatibilní, a proto jsem stále dostával chybová hlášení. Po několika hodinách pokusů a omylů se mi to podařilo. Zjistil jsem, že se to stalo, protože pro Baletku bylo doporučeno používat neaktuální verze Ruby 2.3, tedy podporu, která skončila 31. 3. 2019 včetně podpory bezpečnosti. Tato situace prokázala, že *Ruby* není nejlepší výběr pro systém, který by měl fungovat více než 5 let. Rovněž, že nasazení *Baletky* je komplikované pro lidi, kteří mají žádnou nebo malou zkušenost s programováním.

1.4 Analýza voleb *Mensy ČR*

V této části je uvedena analýza procesu voleb v *Mense ČR* a analýza jejích požadavků na nový systém, které byly zjištěny na schůzkách se zástupci Mensy.

1.4.1 Analýza procesu voleb Mensy CR

Celá tato sekce je shrnutím [24] schůzek se zákazníky. Volby se konají formou přímého, rovného a tajného hlasování jednou za dva roky. Volí se v něm kandidáti do Rady Mensy, Kontrolní komise a předseda Mensy.

Za organizaci, průběh a vyhlášení výsledků voleb zodpovídá Volební komise, kterou jmenuje Rada. Členové Volební komise nesmějí být členy stávajících orgánů Mensy ani kandidáty do nich. Volební komise se skládá ze čtyř členů a jednoho předsedy.

Libovolný člen Mensy může navrhnout kandidáta. Kandidátem může být libovolný zletilý člen Mensy, kromě členů Volební komise. Navržený kandidát, v případě, že chce kandidovat, má poslat svůj písemný souhlas Volební komisi. Současně může poskytnout volební proslov a svou fotografii, které budou zveřejněny prostřednictvím časopisu Mensy, webových stránek Mensy nebo přímým zasíláním. Ke dni přijetí kandidatury nesmí mít kandidát vůči Mense nevyrovnané závazky. Členové Mensy se mohou zúčastnit voleb různými způsoby, ale komise musí předat voličům hlasovací lístky i uvedené informace. Také se lze zúčastnit elektronicky přes internet. Právě realizace tohoto způsobu hlasování je cílem mé bakalářské práce.

Vyhlášení výsledků probíhá na Valné hromadě. Valná hromada je výroční jednání členů Mensy a je jejím nejvyšším orgánem. Den konání Valné hromady stanoví Rada Mensy. Voliči, kteří se nezúčastnili voleb dřív, mohou předat svůj hlas Volební komisi na Valné hromadě. Rada Mensy se skládá z 11 členů. Kontrolní komise má čtyři členy. Předseda může být jen jeden. Kandiduje-li na jednu funkci méně nebo tolik kandidátů kolik bylo požadováno, Volební komise vyhlásí tyto kandidáty za zvolené. V ostatních případech jsou zvoleni kandidáti, kteří získali nejvyšší počet hlasů.

1.4.2 Role

Ze schůzek se zástupci Mensy vyplynulo, že uživatelé budoucí aplikace se dělí na dvě skupiny. Administrátor je uživatelem s rozšířenou působností. Spravuje systém, připravuje ho k volbám a po skončení voleb má přístup k jejich výsledkům. Člen Mensy, dále jen ČM, je uživatelem systému, který se může zúčastnit voleb.

1.4.3 Požadavky

V této sekci jsou požadavky *Mensy ČR* na volební systém shrnuty ve formě funkčních a nefunkčních požadavků.

Funkční požadavky

F1. Přihlášení

- F1.1. ČM bude mít možnost se přihlásit do systému pomoci členského čísla a hesla do inranetu.
- F1.2. Jestli se po hlasování ČM pokusí ještě jednou přihlásit, dostane hlášení, že už hlasoval.
- F1.3. Přihlášený uživatel bude mít možnost odhlásit se.
- F1.4. Jestli se po hlasování ČM uživatel neodhlásí, systém ho přesměruje na přihlašovací stránku.

F2. Hlasování

F2.1. Přihlášený ČM bude mít možnost vybrat své kandidáty.

F2.2. Po hlasování se bude ČM ptát, zda souhlasí se svým výběrem.

F2.3. Po souhlasu se svým výběrem obdrží ČM poděkování za účast ve volbách.

F2.4. Kandidát může mít fotografii a krátký popis.

F2.5. Maximálně může hlasovat pro tolik kandidátů, kolik je míst v orgánu. Je-li počet kandidátů menší nebo stejný jako počet míst v orgánu, pak se hlasování do tohoto orgánu nevyžaduje.

F3. Back-office

F3.1. Administrátor bude mít možnost přidat kandidáta a informace o něm.

F3.2. Po volbách bude mít administrátor možnost stáhnout seznam lidí, kteří hlasovali.

F3.3. Po volbách bude mít administrátor možnost stáhnout výsledky voleb.

F3.4. Systém má podporovat několik administrátorů.

F3.5. Administrátor může přidat hlasy voličů na Valné hromadě.

F3.6. Administrátor může uzavřít seznam kandidátů pro úpravu.

F3.7. Administrátor bude mít možnost odebrat kandidáta ze systému.

F3.8. Administrátor může nastavit počet kandidátů do orgánů *Mensy ČR* a dobu konání hlasování.

F3.9. Počet administrátorů se rovná počtu členů Volební komise.

Nefunkční požadavky

NF 1. Systém má být snadno použitelný, aby ho mohl používat člověk, buď ČM, nebo administrátor, i bez znalostí informačních technologií.

NF 2. Systém nesmí ztratit hlasy voličů.

NF 3. Systém má být bezpečný.

NF 4. Systém má zajistit anonymitu ČM v rámci voleb.

NF 5. Aplikace má být webová a podporovat verze browserů, které byly vydány maximálně dva roky před psaním této práce[7].

[7]Google Chrome: >= 67.0.3396, Mozilla Firefox: >= 60.0, Safari: 11.1, Microsoft Edge: 42.17134, Enternet Explorer: 11.0.10240.16384

1.5 Porovnání procesu hlasování *Mensy ČR* a procesu systému *Baletka*

V této sekci se porovnávají procesy hlasování *Mensy ČR* a hlasování v systému *Baletka*. Pro lepší pochopení byly vytvářeny diagramy aktivity, viz obrázky 1.3, 1.2, 1.4, 1.5, 1.6, 1.7. Jsou rozdělené podle fází hlasování: aktivity před hlasováním, viz obrázky 1.3, 1.2, aktivity během hlasování, viz obrázky 1.4, 1.5, aktivity po hlasování, viz obrázky 1.6, 1.7. Důležité je říct, že jsem srovnával proces Mensy pouze se zrychleným hlasováním *Baletky*, protože proces standardního hlasování je velmi podobný procesu zrychleného hlasování. Liší se pouze procesem samotného hlasování. A také proces standardního hlasování je podobný procesu hlasování Mensy.

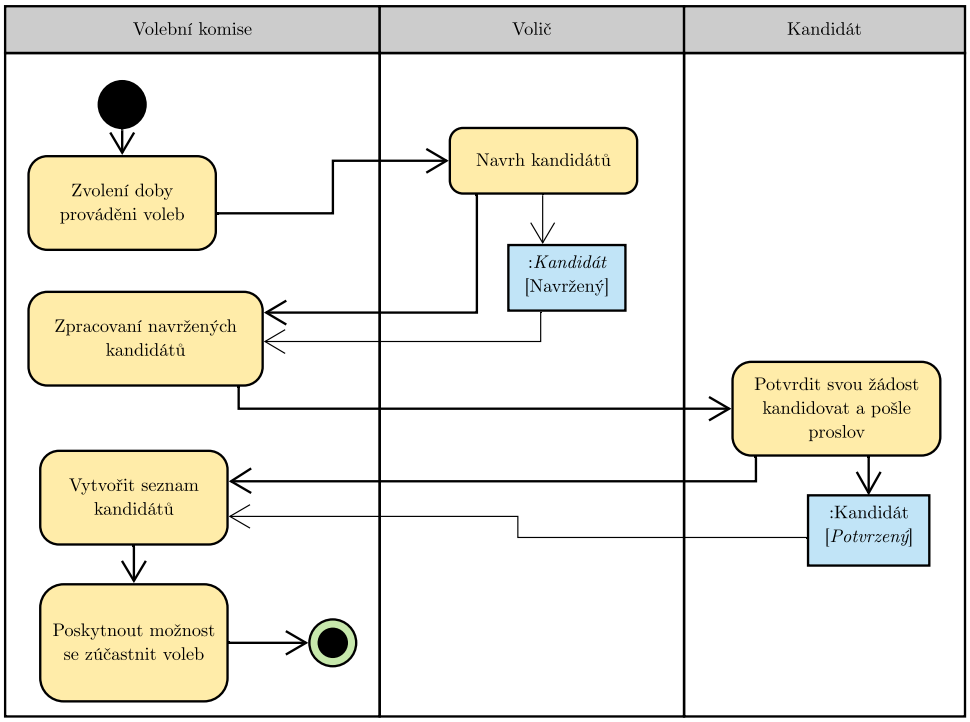
Baletka se používá pro hlasování VR FIT pro řešení důležitých dotazů. To znamená, že rozdíly mezi procesy jsou způsobeny tím, že *Baletka* se používá pro rozhodování VR FIT a Mensa volí členy mensovní správy [25].

Aktivity před hlasováním Příprava dotazu pro hlasování *Baletky* probíhá na jednání VR FIT. Je zřejmé, že to není část *Baletky*, proto není součástí diagramu aktivit, ale je důležité to zmínit[8]. K hlasování *Mensy ČR*je nutné připravit seznam kandidátů, kteří byli navrženi členy Mensy. Tato část procesu může trvat několik týdnů.

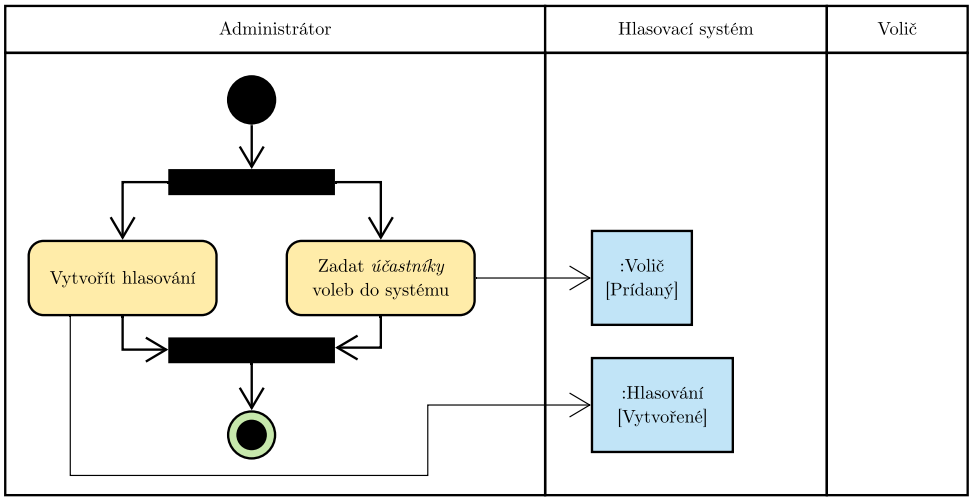
Aktivity během hlasování Odlišnost hlasování je způsobena tím, že *Baletka* je elektronický systém a Mensa používá také jiné způsoby hlasování. Proto možnost zastavit hlasování dřív, než bylo definováno před hlasováním, neexistuje. Mensa rovněž potřebuje sečíst hlasy ze všech zdrojů, což také ovlivní proces voleb.

Aktivity po hlasování *Baletka* po dokončení voleb umožňuje stáhnout výsledky v PDF souboru, který pak lze snadno použít jako oficiální dokument. Současně mensovní proces povoluje zúčastnit se voleb na Valné hromadě po zakončení základního hlasování. Nové přidané hlasy se sčítají s výsledkem základního hlasování.

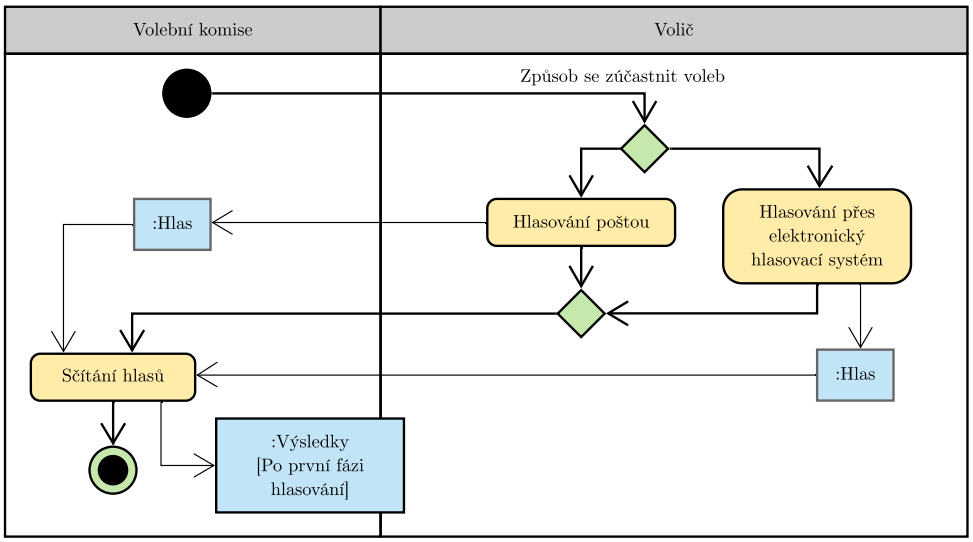
[8]Podrobněji o procesech VR FIT lze přečíst na stránkách VR FIT



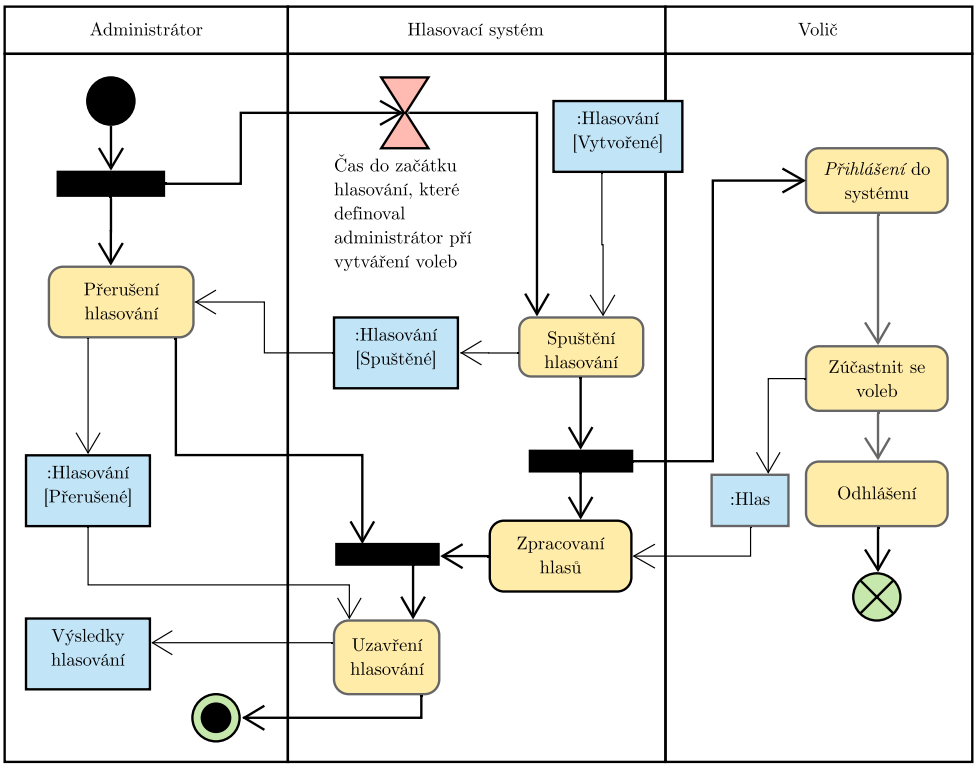
Obrázek 1.2: Diagram aktivit před volbami *Mensy ČR*



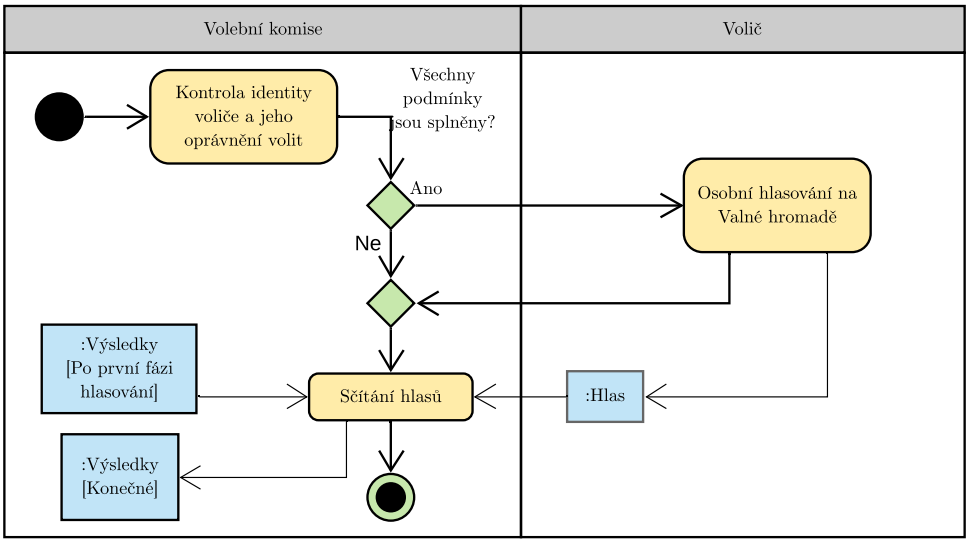
Obrázek 1.3: Diagram aktivit před hlasováním v systému *Baletka*



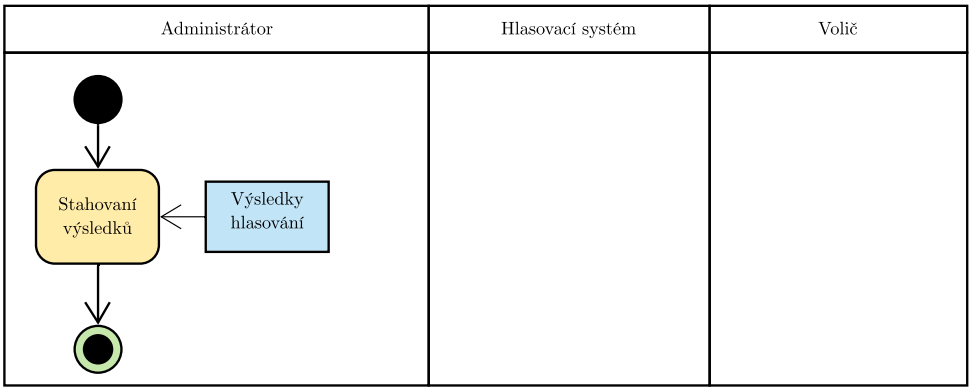
Obrázek 1.4: Diagram aktivit voleb *Mensy ČR*



Obrázek 1.5: Diagram aktivit hlasování v systému *Baletka*



Obrázek 1.6: Diagram aktivit po volbách *Mensy ČR*



Obrázek 1.7: Diagram aktivit po hlasování v systému *Baletka*

Kapitola 2

Návrh

Tato kapitola se věnuje návrhu vlastního hlasovacího systému podle analýzy, popsané v předchozí kapitole. Návrh systému probíhal ve spolupráci s *Mensou ČR*.

2.1 Zpracování požadavků

V této sekci jsem přepracoval analýzu procesu a požadavků *Mensy ČR*.

2.1.1 Role

Diagram na obrázku 2.1 je výsledkem zpracování sekce 1.4.2. V rámci systému jsem rozdělil uživatele do čtyř rolí.

Nepřihlášený uživatel Do této role patří libovolní uživatelé, kteří nejsou přihlášeni.

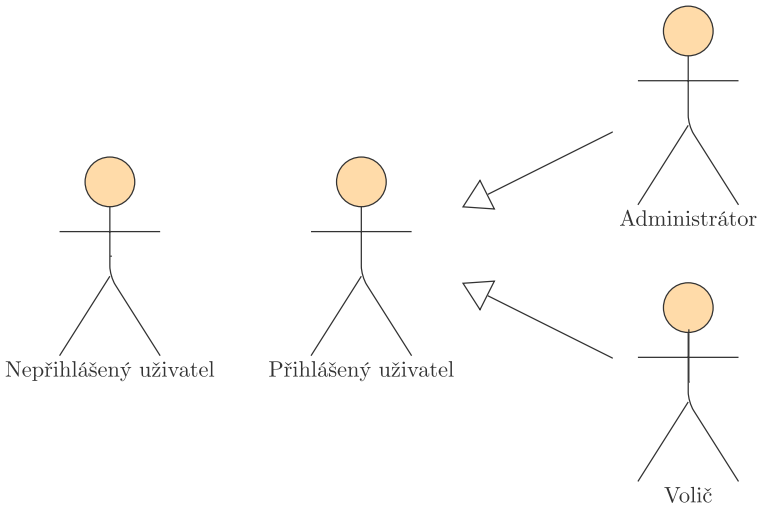
Přihlášený uživatel Do ní patří uživatelé, kteří se přihlásili do systémů.

Administrátor Řadí se do ní uživatelé, kteří mají právo spravovat systém a volby. S velkou pravděpodobností jsou členy Volební komise.

Volič. Tato role spojuje uživatele, kteří mají právo zúčastnit se voleb. To znamená, že jsou členy Mensy, zletilí a nemají s Mensou žádné nevyrovnané závazky.

2.1.2 Diagram případů užití

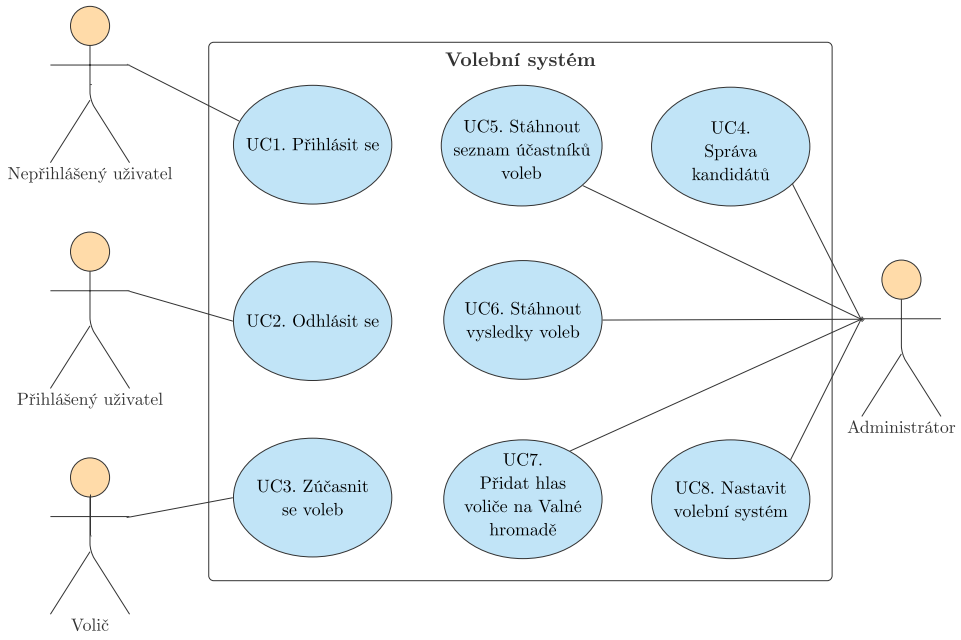
Z funkčních a nefunkčních požadavků (sekce 1.4.3) byl sestaven diagram případů užití, viz obrázek 2.2.



Obrázek 2.1: Diagram aktérů

UC1. Přihlásit se. Případ užití umožňuje uživateli přihlásit se do systému hlasování. Před spuštěním hlasování do systému se mohou přihlásit jenom administrátoři. Pro přihlášení zadají email, který byl zadán při nastavení systému, a heslo, které bylo vygenerováno při prvním nastavení systému. Pro přihlášení jsem nabídl použít dvoufaktorovou autentifikaci, neboť by to zlepšilo bezpečnost celé aplikace. Mnou byla nabídnuta autentifikace pomocí kódu, který by měl být zaslán uživateli v emailu nebo SMS. Po konzultaci se zástupci Mensy bylo vyřešeno použít emailové služby pro zasílání kódu. Takže po zadání správných údajů systém poprosí o zadání kódu pro úspěšné dokončení přihlášení. V průběhu voleb se mohou přihlásit do systému voliči. Proto musí zadat své členské číslo a heslo z intranetu. Pak stejně jako administrátoři potřebují zadat kód z emailu. Při dalším pokusu o přihlášení se zobrazí oznámení o tom, že se zúčastnil voleb. Po volbách se může přihlásit do systému jenom administrátor.

UC2. Odhlásit se. Případ užití umožňuje uživateli odhlásit se ze systému. Speciální požadavky k tomuto případu užití nebyly. Volič se odhlásí, když odsouhlasí dokončení hlasování a poslání svého hlasu hlas, pak bude zobrazeno tlačítko pro odhlášení. V případě, že se sám neodhlásí, systém ho odhlásí automaticky. Administrátor může použít speciální tlačítko pro odhlášení ze systému. Po odhlášení bude uživatel přesměrován na stránku přihlášení.



Obrázek 2.2: Diagram případů užití

UC3. Zúčastnit se voleb. Případ užití umožňuje uživateli zúčastnit se voleb. Je to nejdůležitější část aplikace. Po přihlášení se volič dostane na stránku voleb. Na ní může vybrat kandidáty. V případě zájmu může rozkliknout kandidáta a přečíst si jeho volební proslov. Pro dokončení hlasování uživatel potřebuje vybrat alespoň jednoho kandidáta. Nemůže také zvolit více kandidátů, než bylo nastaveno. Po souhlasu dokončit hlasování bude hlas voliče odeslán na server a volič bude odhlášen.

UC4. Správa kandidátů. Případ užití umožňuje administrátorovi spravovat kandidáty. Jeho základní podmínky jsou následující: administrátor je přihlášený do systému a hlasování ještě nebylo započato. V rámci tohoto případu užití jsou následující aktivity:

Přidat nového kandidáta. Na stránce správy kandidátů se administrátor po stisknutí speciálního tlačítka dostane na stránku přidání nového kandidáta. Administrátor může nastavit následující údaje, které budou ukázány voličům:

- povinné
  - plné jméno,
  - členské číslo,
  - volená pozice.

- nepovinné
    - telefonní číslo,
    - email,
    - doplňující informace,
    - proslov,
    - fotografie.

Po zadání údajů kandidáta je administrátor pomocí speciálního tlačítka může uložit. Nový kandidát se zobrazí v seznamu přidaných kandidátů.

Smazat přidaného kandidáta. Na stránce správy kandidátů může administrátor smazat kandidáta po stisknutí speciálního tlačítka a potvrzení své žádosti.

Uzavřít seznam kandidátů pro úpravu. Na stránce správy kandidátů může administrátor uzavřít seznam kandidátů pro úpravu a pak už nelze smazat kandidáta nebo přidat dalšího. Tuto operaci lze provést jen jednou a nelze ji zrušit.

UC5. Stáhnout seznam účastníků voleb. Případ užití umožňuje administrátorovi stáhnout seznam členů Mensy, kteří se zúčastnili voleb. Je přístupný po dokončení voleb na Valné hromadě.

UC6. Stáhnout výsledky voleb. Případ užití umožňuje administrátorovi stáhnout výsledky voleb. Jsou dostupné dvě varianty. První obsahuje výsledky elektronického hlasování. Lze ji stáhnout hned po elektronickém hlasování. Druhý je přístupný po dokončení voleb na Valné hromadě.

UC7. Přidat hlas voliče na Valné hromadě. Případ užití umožňuje administrátorovi přidat hlasy voličů, kteří hlasovali na Valné hromadě. V případě, že se ČM chce zúčastnit voleb, člen volební komise zkontroluje, zda nehlasoval elektronicky. Udělat to může přes administrátorskou část aplikace. Do speciálního pole zadá členské číslo a po stisknutí speciálního tlačítka dostane výsledek vyhledávání. V případě, že se ČM nezúčastnil voleb, stiskne tlačítko, pomocí kterého lze uložit členské číslo. Jakmile budou všichni voliči přidáni, administrátor zakáže přidávání nových voličů. Pak bude mít možnost přidat hlasy přidaných předem voličů do systému. Počet hlasů je omezen počtem voličů na Valné hromadě. Ve výsledku hlasy ČM, kteří hlasovali na Valné hromadě, budou uloženy v systému a sečteny s výsledky internetového hlasování.

UC8. Nastavit volební systém. Případ užití umožňuje provést základní nastavení systému. Patří k nim:

- Datum začátku voleb.
- Datum konce voleb.
- Datum Valné hromady.
- Počet volných míst v Radě *Mensy ČR*.
- Počet volných míst v Kontrolní komisi.
- Emailové adresy dalších administrátorů.

První přihlášený administrátor provede nastavení volebního systému. Bez něho je systém nepoužitelný. Po vyplnění nutných údajů a potvrzení jejich správnosti bude administrátor přesměrován zpět na přihlašovací stránku a systém bude připraven k použití.

2.1.3 Pokrytí funkčních požadavků

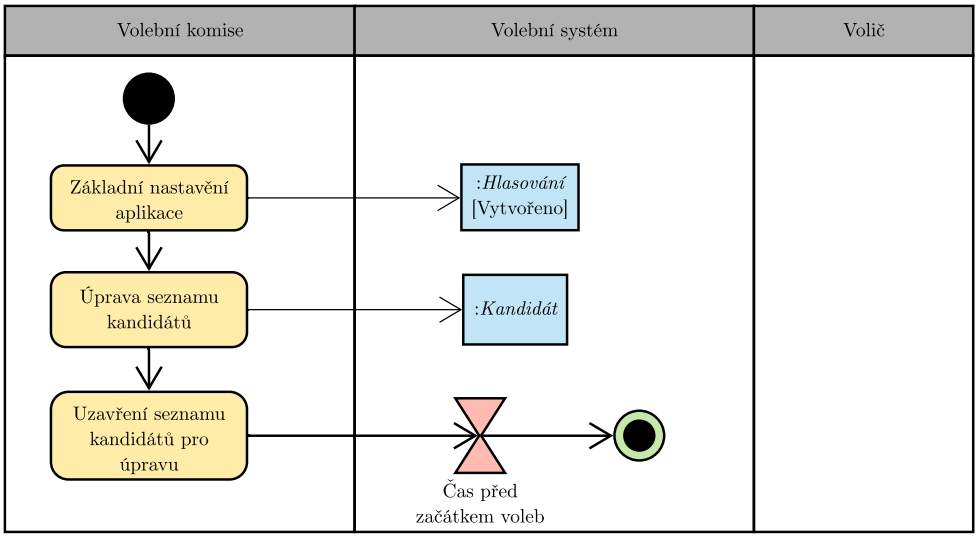
Zkontroloval jsem pokrytí požadavků pomocí tabulky pokrytí 2.3. Z ní vyplývá, že všechny funkční požadavky jsou pokryty navrženými případy užití.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
F1.1	+							
F1.2	+							
F1.3		+						
F1.4		+						
F2.1			+					
F2.2			+					
F2.3			+					
F2.4				+				
F2.5			+	+				
F3.1				+				
F3.2						+		
F3.3						+		
F3.4								+
F3.5							+	
F3.6				+				
F3.7				+				
F3.8								+
F3.9								+

Obrázek 2.3: Tabulka pokrytí funkčních požadavků

2.1.4 Proces voleb nového systému

Pro lepší pochopení procesu voleb nového systému jsem vytvořil diagramy aktivit, viz obrázky 2.4, 2.5, 2.5.



Obrázek 2.4: Diagram aktivit nového systémů před internetovými volbami

2.1.5 Doménový model

Na základě analýzy jsem vytvořil doménový model.

User Entita obsahuje údaje uživatele. V závislosti na jeho roli budou uloženy různé údaje. Systém bude ukládat hash hesel administrátorů, ale ne voličů.

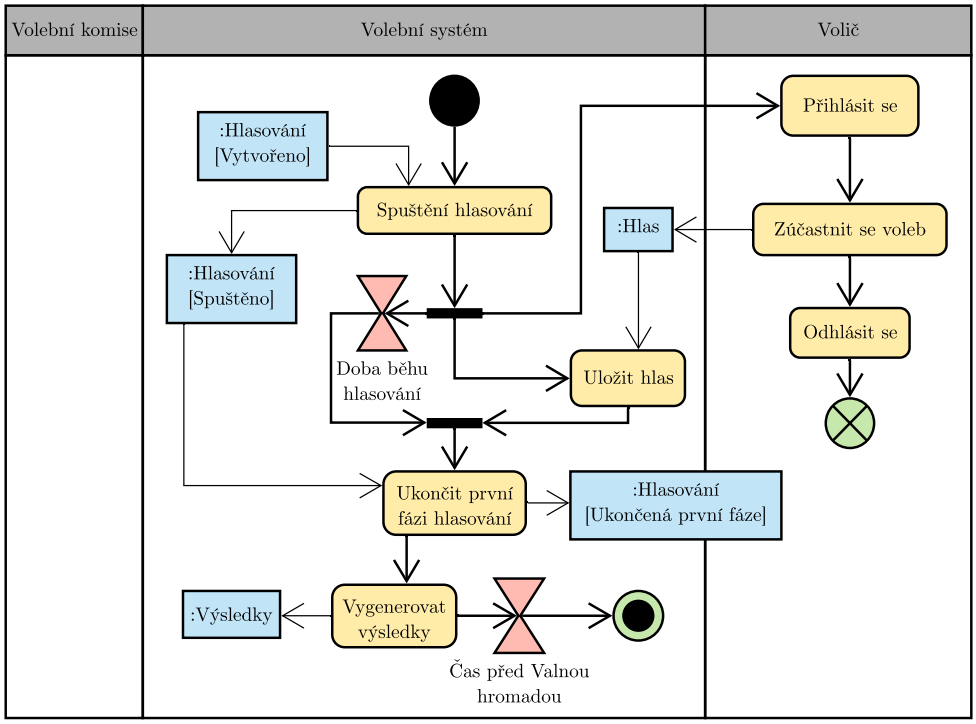
Role Entita obsahuje role, které mohou mít uživatelé. Předpokládá se, že jsou dvě: volič a administrátor.

Candidate Entita obsahuje údaje kandidátů. Povinné parametry jsou `mensaId`, `name`.

Vote Entita obsahuje informace o jednotlivých hlasech voličů.

Department Entita obsahuje informace o orgánech *Mensy ČR*.

Elections Entita obsahuje nastavení voleb. Smí obsahovat jenom jeden záznam.



Obrázek 2.5: Diagram aktivit nového systémů během internetových voleb

2.1.6 Wireframy

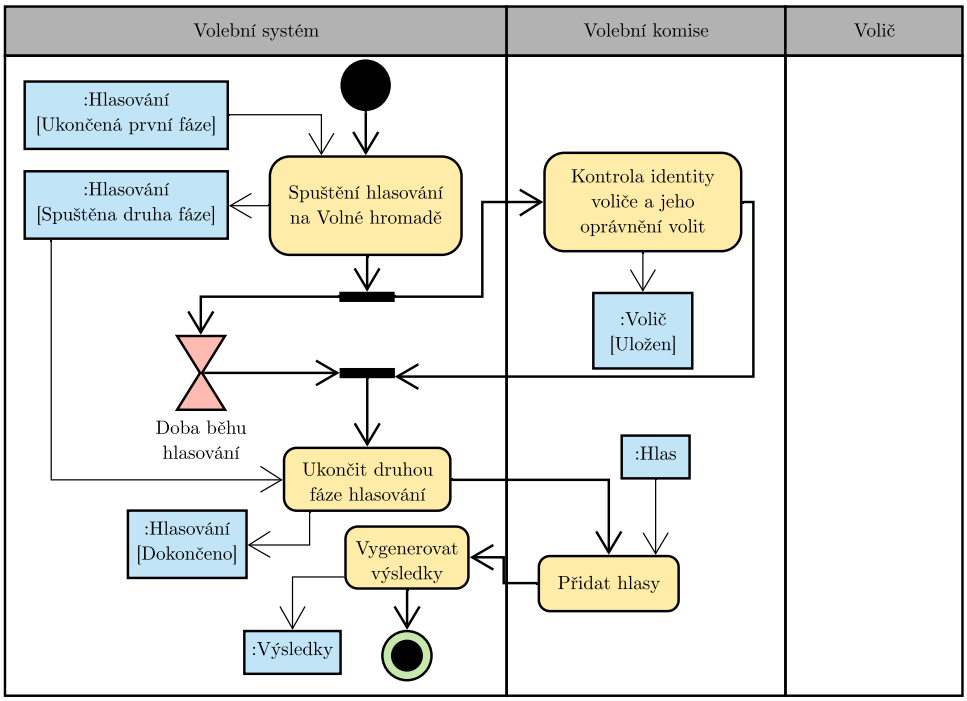
Pro vysvětlení svého návrhu zástupcům *Mensy ČR*, a ukázky moje vize budoucí aplikace, jsem vytvořil wireframy. Bylo provedeno několik schůzek se zástupci Mensy, na kterých jsem představoval svůj návrh. Po každé schůze byly provedeny úpravy návrhu, dokud s ním zákazník nebyl spokojen. Poslední verzi lze najít na SD karte, která je přílożena k práci.

2.2 Návrh na vylepšení

V této sekci je popsán V této sekci jsou popsány návrhy na dodatečné vylepšení systému, které nepatří mezi základní požadavky klienta.

2.2.1 Návod

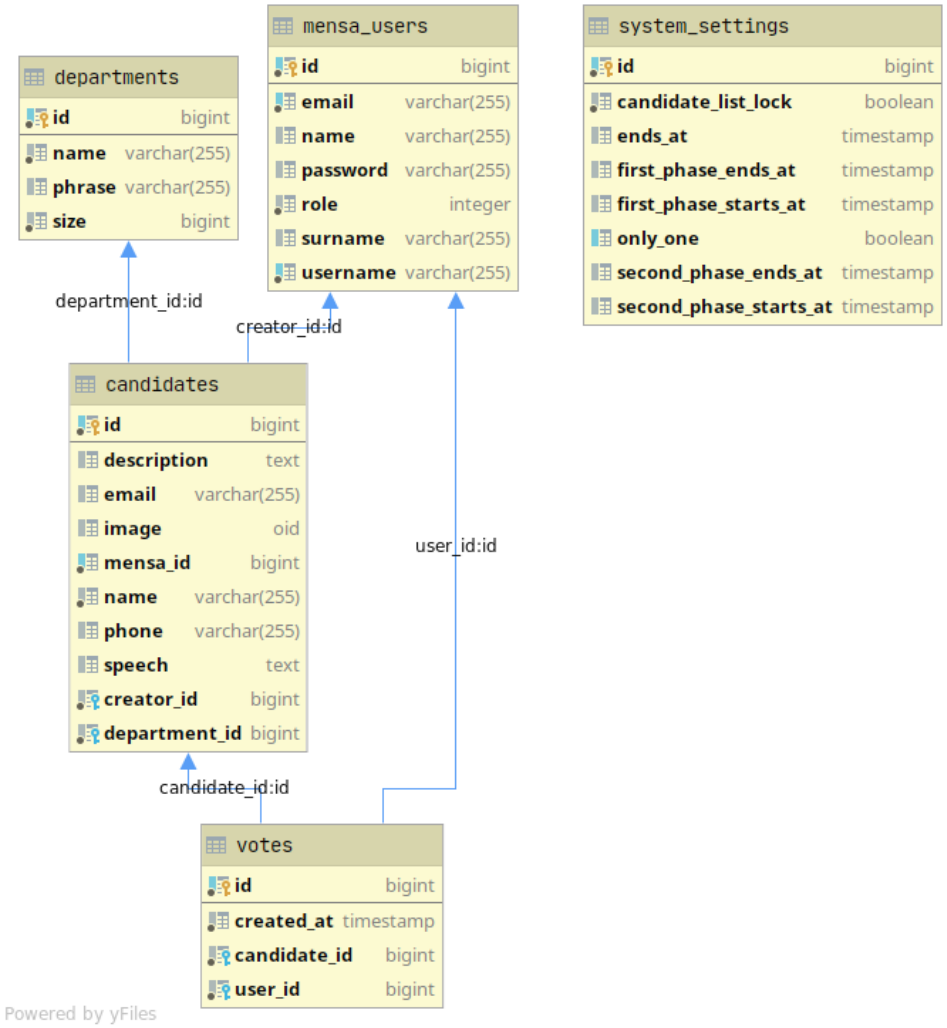
Zástupce Mensy zmínil, že velkou část ČM tvoří starší generace. Proto mne napadlo, že návod k volebnímu systému bude užitečnou funkcí, která pomůže voličům seznámit se s novým systémem. Na schůzi jsem ukázal svůj návrh zákazníkovi. Můj nápad se mu líbil, a proto odsouhlasil přidat ho do výsledného systému.



Obrázek 2.6: Diagram aktivit nového systémů po internetových volbách

2.2.2 Odmítnuté ideje

Při vytváření návrhu jsem měl několik idejí, které zákazník odmítl. Pro zvýšení odolnosti proti chybám a zneužití jsem nabídl omezit volnou úpravu seznamu kandidátů. Základní myšlenkou bylo zakázat přidání nebo mazaní kandidátů bez souhlasu všech administrátorů. Tímto by kompromitace skoro všech účtů administrátorů nebyla tak kritická pro fungování aplikace. Tento návrh má své problémy, protože komplikuje nastavení voleb a potřebuje účast všech administrátorů. To byl důvod k odmítnutí. Stejně byla odmítnuta idea použití certifikátu pro přihlášení administrátorů, protože komplikovala proces jejich voleb.



Obrázek 2.7: Doménový model

Kapitola 3

Implementace

Tato kapitola se věnuje implementaci vlastního hlasovacího systému na základě popsaného návrhu v předchozí kapitole a jeho testování.

3.1 Architektura

Aplikace sestává z klientské a serverové části. Klientská část komunikuje se serverovou částí pomocí REST API. Podpora REST služeb je poskytovaná ve velkém množství frameworků jak pro serverové, tak i pro klientské části. Navíc tato architektura dělá nezávislou serverovou a klientskou část, což umožňuje použít nejlepší technologii pro každou z nich. V případě potřeby lze přidat další klientské aplikace nebo vyměnit serverovou. Zvolil jsem takovou architekturu, protože dává flexibilitu při vývoji aplikace, avšak toto řešení není dokonalé, protože komplikuje architekturu a potřebuje od programátora vytvořit dvě aplikace.

3.2 Technologie

Pro serverovou část jsem zvolil jazyk Java, protože nabízí dlouhodobou podporu [26]. V případě, že Mensa bude požadovat přidání nové funkcionality, snadno najde programátora, který to udělá. Nepoužil jsem jen Javu, ale použil jsem jí společně s frameworkem Spring Boot, který je modulem frameworku Spring. Spring zjednoduší spoustu operací: komunikaci s databází, obsluhu připojení klientů a další. Díky architektuře frameworku Spring ho lze nakonfigurovat podle téměř libovolných potřeb. Dále jsem používal ještě jeden z modulu Spring – Spring Security, který nabízí mnoho možností pro zajištění bezpečnosti aplikace. Jako hlavní databázi jsem pro svou aplikaci použil PostgreSQL.

Pro klientskou část jsem vybral JavaScript s frameworkem Vue.js. Zvolil jsem je, protože dnes je JavaScript základem skoro všech webových aplikací a Vue.js je moderním *open source* frameworkem, který umožňuje snadné vy-

tváření webové aplikace. Také spolu s Vue.js používám Axios pro komunikaci se serverem, knihovnu Vuex pro uložení dat, Bulma pro nastavení CSS a CSS preprocesor Stylus, který umožňuje jednodušší psaní CSS kódu. Použití všech těchto technologií umožňuje snadný a rychlý vývoj moderních webových aplikací.

3.3 Serverová část

Pro serverové aplikace jsem použil vícevrstvou architekturu. Rozdělil jsem aplikace na 3 vrstvy:

- Datová vrstva.
- Doménová vrstva.
- Aplikační vrstva.

Výsledný diagram tříd serverové části lze najít na přiloženém CD.

3.3.1 Datová vrstva

Pro komunikaci s databází jsem použil standardní implementace ORM[9] v Spring Boot, která používá framework Hibernate.

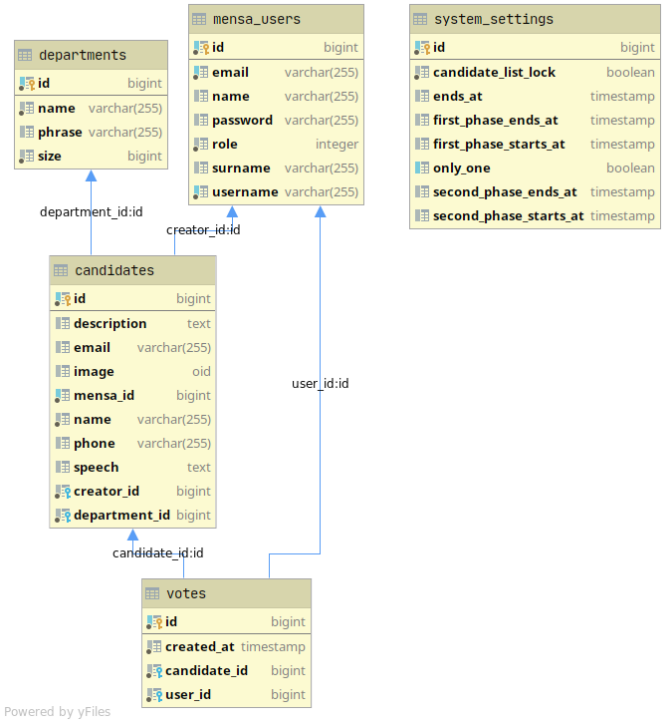
Diagram entit Na základě doménového modelu byl vytvořen diagram entit, viz obrázek 3.1, který pak byl použit pro implementaci tříd reprezentujících entity v rámci aplikace. Pro přístup k databázi použity implementaci rozhraní `CrudRepository`.

3.3.2 Doménová vrstva

Po získání entit z databází se konvertují do DTO objektu a dále používají jenom je. Pro konvertování byl použit mechanismus založený v Spring. Proto byly implementovány třídy pro každý směr konvertování, které implementují rozhraní `Converter`. Pak byl vytvořen Bean[10], který vrátí implementace rozhraní `ConversionService`, která obsahuje všechny konvertory. Dále v případě potřeby konvertování mezi entitou a DTO je nutné jenom vložit závislost na `ConversionService`. Zpracováním DTO se zabývají služby, které jsou závislé na `CrudRepository`. Nabízí metody pro získání dat v aplikační vrstvě.

[9] Object-relational mapping. Spojuje entity z databázi a třídy z aplikace.

[10] V terminologii Spring frameworku tak se jmenují objekty, které tvoří kostru aplikace a jsou řízeny Spring IoC [27].



Obrázek 3.1: Diagram entit

3.3.3 Aplikační vrstva

Aplikační vrstva obsahuje REST API, které odpovídá za zpracování requestů klientů a vytváření responsů. Podrobnější informace lze najít na stránkách [**web:spring:boot**].

3.3.4 Autentifikace

Pro autentifikace uživatelů jsem se rozhodl použít JWT[11], protože vyhovuje mým potřebám. Ale Spring Boot nemá podporu JWT, proto jsem ji potřeboval implementovat samostatně. Pro vlastní implementaci jsem vytvořil následující třídy:

`WebSecurityConfig` odpovídá za obecnou konfiguraci bezpečnosti celé aplikace. Rozšiřuje abstraktní `WebSecurityConfigurerAdapter`.

`JwtUtils` odpovídá za validace a zpracování tokenů.

`AuthEntryPointJwt` odpovídá za zpracování výjimek, které vznikly v procesu autentifikace. Implementuje rozhraní `AuthenticationEntryPoint`

[11] JSON Web Token

`AuthTokenFilter` odpovídá za integrace kontroly v procesu zpracování requestů. Rozšiřuje abstraktní třídu `OncePerRequestFilter`.

`UserDetailsImpl` třída, která reprezentuje uživatele v rámci kontextu autentifikace a obsahuje všechny potřebné údaje. Implementuje rozhraní `UserDetails`.

`UserDetailsServiceImpl` odpovídá za získání dat, potřebných pro autentifikace uživatele. Implementuje rozhraní `UserDetailsService`.

`AuthenticationManagerImpl` odpovídá za porovnání údajů uživatele, které byly částí reqestů pro přihlášení, a údajů, které byly nalezeny v databázi. Implementuje rozhraní `AuthenticationManager`.

Ve výsledku moje implementace umožnila používat JWT pro autentifikace uživatelů.

3.3.5 Ukládání hlasu voliče

Pro zajištění správného uložení hlasu voliče informace o něm se ukládá do speciální třídy `UserSessionDAO`, která obsahuje mapu uživatelských jmen a instance třídy `UserSession`. Třída `UserSession` ukládá informace o přihlášeném uživateli. Když volič odešle svůj hlas ke zpracování, informace o něm bude zpřístupněna přes `UserSessionDAO` a uložena s jeho hlasem současně jednou transakci. Takovým způsobem bude zajištěno, že jestli dojde k poruše serveru v moment ukládání hlasu, situace, kdy bude uložen jenom hlas nebo volič, nenastane.

3.3.6 Generování výsledků

Za generování výsledků voleb odpovídá rozhraní `ResultsService` a její implementace `ResultsServiceImpl`. Byly vymyšleny tak, aby nešlo vygenerovat několik různých souborů. V případě, že výsledky ještě nebyly vygenerovaný, aplikace je vygeneruje a uloži do speciálního souboru. Při dalších pokusech stáhnout výsledky budou vráceny vygenerované dříve soubory.

3.3.7 Mensovní API

Pro kontrolu údajů ČM Mensou bylo poskytnuto API, které existuje ve dvou verzích – produkční a testovací. Na vstup potřebuje členské číslo a heslo uživatele, a speciální klíč. Pak vrací údaje uživatele, včetně jména, příjmení a oprávnění volit, v případě, že jsou přihlašovací udaje platné. V opačném případě vrací chybový kód a její příčinu.

Na začátku se mi nedařilo připojit se na API, protože nepoužívají validní HTTPS certifikát. Ale po jeho vyměně a drobných úpravách v souboru `/etc/hosts` na vlastním počítači se mi to podařilo. Důležité je říci, že takové

komplikace vznikly jenom na testovacím API. Produkční API fungovalo bez závad.

3.4 Klientská část

Tato sekce popisuje jednotlivé častí implementace klientské části.

Výsledná aplikace částečně podporuje mobilní zařízení. Administrátorská část aplikace nepodporuje mobilní výhled.

Pro uložení JWT tokenu používám knihovnu Vuex, která ho ukládá na vnitřního úložiště. Pak přidávám token do hlavičky dotazu na server. Také ho používám pro omezení přístupu uživatelů. Při každém přesměrování se provádí kontrola role uživatele. V případě, že nemá oprávnění bude přesměrován na přihlašovací stránku.

3.4.1 Komunikace se serverem

Pro komunikaci se serverem byl použit framework Axios. Jednotlivé metody byly vyčleněny do složky `services`, a pak rozděleny podle kontrolérů, které dotazují. Většinou jedna služební třída odpovídá jednomu kontroléru ze serverové aplikace.

3.4.2 Přihlášení

Přihlášení je úvodní stránka, viz obrázek 3.2. Po odhlášení se uživatel vrátí sem. Také pří pokusu se dostat na stránky, ke kterým uživatel nemá oprávnění přistoupit, vrátí se na stránku přihlášení. V případě, že token uživatele není validní, přesměruje ho na úvodní stránku.

Jestli uživatel zadal nesprávné přihlašovací údaje dostane chybovou hlásku, viz obrázek 3.3. Také dostane hlásku v případě, že nastal nějaký problém se serverovou aplikaci. V případě, že se uživatel zúčastnil voleb, ale zkouší se přihlásit znovu, dostane další oznámení, viz obrázek 3.4.

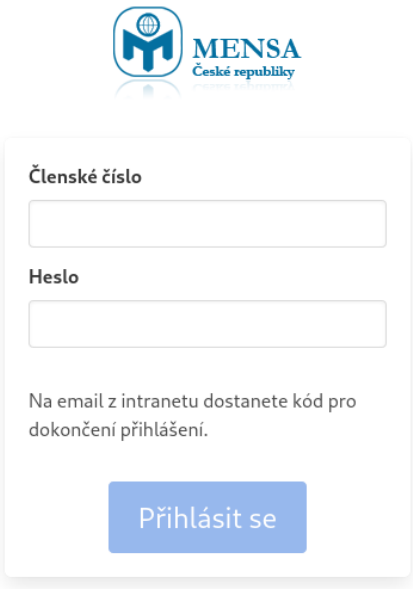
Po zadání správných přihlašovacích údajů se uživatel dostane na stránku pro zadání kódu, který dostal na email, viz obrázek 3.5. v Případě, že ho nedostal, může požádat o nový kód.

V případě, že uživatel je administrátorem, pak se dostane na stránky administrátorské konzole. V případě, že je voličem, dostane se na stránku návodu k volebnímu systému. Tam se uživatel seznámí s funkcionalitou systému.

3.4.3 Hlasování

Hlasování je základní aktivita pro uživatele, viz obrázek 3.6. Hlasování je založeno na následujících pravidlech:

- Uživatel musí někoho vybrat. Bez toho nemůže dokončit hlasování.



Obrázek 3.2: Úvodní přihlašovací stránka.

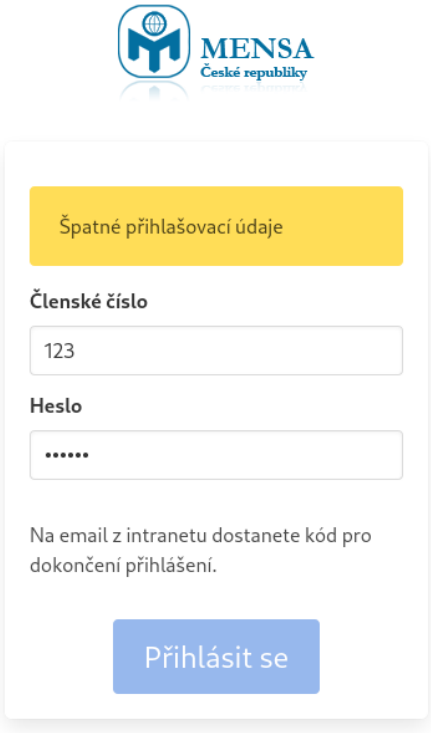
- Uživatel nemůže vybrat víc kandidátu do jedeného orgánu, než nastavil administrátor. V případě pokusu vybrat kandidáta nad nastavený limit se zobrazí oznámení.
- Uživatel musí odsouhlasit dokončení svého hlasování.
- V případě, že nestihne udělat to včas, bude odhlášen a potřebuje začít proces znovu. To je uděláno z bezpečnostních důvodů.

3.4.4 Nastavení systému

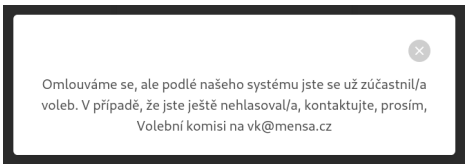
Nastavení systému probíhá pomocí konfiguračního souboru `config.properties`. Obsahuje emaily a hesla administrátorů, a nastavení pro orgány *Mensy ČR*. Jestli systém bude spuštěn v režimu `prod`, pak po inicializace systému soubor bude smazán.

3.4.5 Správa kandidátů

Po nastavení systémů administrátory mohou přidávat nové kandidáty, a v případě potřeby je odebírat. Uzavřít úpravu seznamu kandidátu lze pomoci speciálního tlačítka, a zrušit tuto akci nelze. Pro přidání kandidátů byla vytvořena speciální stránka, viz obrázek 3.7.



Obrázek 3.3: Základní chybové oznámení.



Obrázek 3.4: Oznámeni o druhem pokusu se zúčastnit voleb.

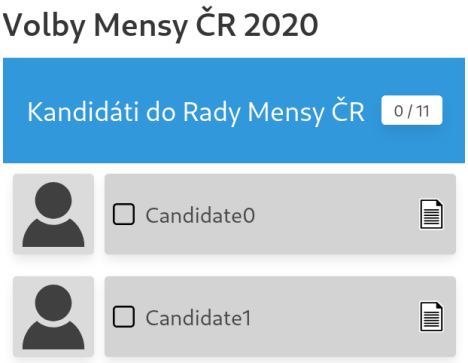
3.4.6 Správa výsledků

Na této stránce lze stáhnout výsledky a zkontrolovat, jestli se volič zúčastnil voleb, viz obrázek 3.8. Zda kandidát hlasoval, lze ho přidat do seznamu účastníků voleb a pak přidat jeho hlas.

3.5 Testování

V této sekci jsem chtěl popsat pokusy uživatelského a akceptačního testování. Jsem zkoušel provést společné s *Mensou ČR*. Byly provedeny dva pokusy, ale žádný nebyl úspěšným. Pokoušel jsem se provést testování společně s *Mensou ČR*. Ale to, co se mi podařilo ukázat členům Mensy bylo ohodnoceno dobře.

Obrázek 3.5: Stránka pro zadaní kódu.

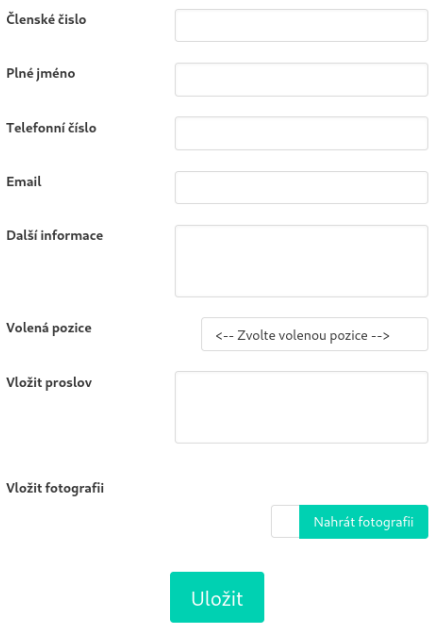


Obrázek 3.6: Stránka hlasování.

3.6 Možné vylepšení

Výsledný systém lze vylepšit. Systém není hotov, protože mi vývoj šel pomaleji, než jsem čekal, kvůli výběru technologií. Věci, které jsem nestihl:

- Fáze voleb.
- Dvoufactorová autentifikce není integrována.
- HTTPS.
- Nastaveni systému.



Obrázek 3.7: Stránka pro správu kandidátů.

- Nebylo provedeno testování.
- Validace dat, získaných od uživatele.

Ke konci implementace jsem dospěl k následujícím závěrům:

- Komplikace, které nastaly kvůli rozdělení serverové a klientské části nebyly oceněny dobře. Nedostatečná zkušenost s technologiemi pro klientskou a serverovou části zvlášť při současném použití obou technologii mohou vytvořit doplňující komplikace.
- Vyčlenění mechanismu přihlášení a řízení autentifikace do další aplikace možné by bylo vhodným řešením. Ve výsledku by to povolilo snadnou úpravu mechanismu autentifikace. Ale je zřejmé, že to ještě zkomplikuje výsledný systém.
- Java velmi komplikuje některé částí vývoje. Například v Kotlinu na rozdíl od Javy se *gettery* a *settery* generují pří kompilaci. Výsledkem je klesání počtu drobných úprav kódu.

Obrázek 3.8: Stránka pro správu výsledku voleb.

Závěr

Cílem této práce byla implementace volebního systému pro *Mensu ČR*. Jejími nedílnými částmi bylo také seznámit se s problematikou voleb přes internet, analyzovat požadavky *Mensy ČR* a volební systém Fakulty informačních technologií ČVUT v Praze pro provádění elektronického hlasování členů vědecké rady. Navíc byla uskutečněna analýza základů bezpečnosti webových aplikací.

Návrh systému byl projednán a následně schválen zástupci Mensy. Pro implementaci serverové části byl použit jazyk Java s frameworkem Spring. Pro implementaci klientské části byl použit jazyk JavaScript s frameworkem Vue.js.

Mensa poskytla vlastní testovací server a testovací API pro kontrolu údajů uživatele. Ale kvůli nedokončené implementaci uživatelské a akceptační testování nebylo úspěšné. To, co bylo ukázáno na testování bylo akceptováno s drobnými úpravami.

Bibliografie

1. PETR, Nohejl. *Zabezpečení hlasovací aplikace Baletka*. Praha, 2018. Dostupné také z: `http://hdl.handle.net/10467/76794`. Bakaláská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, Katedra počítačových systémů. Vedoucí práce: Starosta Štěpán.
2. GERMANN, Micha; SERDÜLT, Uwe. Internet voting and turnout: Evidence from Switzerland. *Electoral Studies*. 2017, roč. 47, s. 1–12. ISSN 0261-3794. Dostupné z DOI: `https://doi.org/10.1016/j.electstud.2017.03.001`.
3. KRIMMER, Robert; DUENAS-CID, David; KRIVONOSOVA, Iuliia; VINKEL, Priit; KOITMAE, Arne. How Much Does an e-Vote Cost? Cost Comparison per Vote in Multichannel Elections in Estonia. In: KRIMMER, Robert; VOLKAMER, Melanie; CORTIER, Véronique; GORÉ, Rajeev; HAPSARA, Manik; SERDÜLT, Uwe; DUENAS-CID, David (ed.). *Electronic Voting*. Cham: Springer International Publishing, 2018, s. 117–131. ISBN 978-3-030-00419-4.
4. 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, č. 4, s. 729–744.
5. SPRINGALL, Drew; FINKENAUER, Travis; DURUMERIC, Zakir; KITCAT, Jason; HURSTI, Harri; MACALPINE, Margaret; HALDERMAN, J. Alex. Security Analysis of the Estonian Internet Voting System. In: *Proceedings of the 21st ACM Conference on Computer and Communications Security*. 2014.
6. *18/05/2003: Rapport concernant les élections du 18 mai 2003[18. 5. 2003: Zpráva o volbách ze dne 18. května 2003]* [online]. Pour une Éthique du Vote Automatisé [cit. 2020-05-15]. Dostupné z: `https://www.poureva.be/article.php3?id_article=32&lang=fr`.

7. RAIN, TV. Veselie vybory. Podborka vbrosov [Vesele volby. Kolekce zneužít]. In: *Youtube* [online]. 2015 [cit. 2020-04-15]. Dostupné z: https://www.youtube.com/watch?v=JTvQXQLoq8Q. Kanál uživatele TV Rain.
8. FALCÃO E CUNHA, João; LEITÃO, Mário Jorge; FARIA, João Pascoal; PIMENTA MONTEIRO, Miguel; CARRAVILLA, Maria Antónia. A methodology for auditing e-voting processes and systems used at the elections for the portuguese parliament. In: KRIMMER, Robert (ed.). *Electronic Voting 2006 – 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC*. Bonn: Gesellschaft für Informatik e.V., 2006, s. 145–154.
9. WIKIPEDIE. *Saatyho metoda – Wikipedie: Otevřená encyklopedie* [online]. 2020 [cit. 2020-04-15]. Dostupné z: %5Curl%7Bhttps://cs.wikipedia.org/w/index.php?title=Saatyho_metoda&oldid=18066048%7D.
10. *About the OWASP Foundation* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/about/.
11. *Vyhledávač Summon* [online]. Serials Solutions [cit. 2020-05-14]. Dostupné z: http://80.cvut.summon.serialssolutions.com.dialog.cvut.cz/#!/search?ho=t&l=en&q=owasp.
12. *OWASP Top Ten* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/.
13. *A1:2017-Injection* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A1-Injection.
14. *A2:2017-Broken Authentication* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication.
15. *A3:2017-Sensitive Data Exposure* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure.
16. *A4:2017-XML External Entities (XXE)* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A4-XML_External_Entities_(XXE).
17. *A5:2017-Broken Access Control* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A5-Broken_Access_Control.
18. *A6:2017-Security Misconfiguration* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A6-Security_Misconfiguration.

19. *A7:2017-Cross-Site Scripting (XSS)* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_(XSS).

20. WIKIPEDIA CONTRIBUTORS. *Defense-in-depth (computing) – Wikipedia, The Free Encyclopedia* [online]. 2020 [cit. 2020-05-28]. Dostupné z: %5Curl%7Bhttps://en.wikipedia.org/w/index.php?title=Defense_in_depth_(computing)&oldid=950656380%7D.

21. *A8:2017-Insecure Deserialization* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A8-Insecure_Deserialization.

22. *A9:2017-Using Components with Known Vulnerabilities* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities.

23. *A10:2017-Insufficient Logging & Monitoring* [online]. OWASP [cit. 2020-05-14]. Dostupné z: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A10-Insufficient_Logging%5C%252526Monitoring.

24. *Stanovy Mensy ČR* [online]. Mensa ČR, 2016 [cit. 2020-04-15]. Dostupné z: http://www.mensa.cz/mensa/stanovy/.

25. *Vědecká rada* [online]. ČVUT FIT, 2020 [cit. 2020-05-28]. Dostupné z: https://fit.cvut.cz/cs/veda-a-vyzkum/zazemi/vedecka-rada.

26. *Oracle Java SE Support Roadmap* [online]. Oracle, 2020 [cit. 2020-05-28]. Dostupné z: https://www.oracle.com/java/technologies/java-se-support-roadmap.html.

27. *Core Technologies* [online]. Spring, 2020 [cit. 2020-05-28]. Dostupné z: https://docs.spring.io/spring/docs/current/spring-framework-reference/core.html#beans-introduction.

Příloha A

Seznam použitých zkratek

AHP Analytical Hierarchy Process

API Application Programming Interface

DDOS Distributed denial of service

DTO Data transfer object

EVM Electronic voting machines

GUI Graphical user interface

HSTS HTTP Strict Transport Security

HTTP Hypertext Transfer Protocol

JWT JSON Web Token

LDAP Lightweight Directory Access Protocol

NoSQL non SQL

OS Operating system

ORM Object-relational mapping

OWASP The Open Web Application Security Project

PDF Portable Document Format

PFS Perfect Forward Secrecy

SAST Source Code Analysis Tools

SQL Structured Query Language

TLS Transport Layer Security

VR FIT Vědecká rada Fakulty informačních technologií ČVUT

XML Extensible markup language

XSS Cross-Site Scripting

XXE XML External Entities

Příloha B

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `jar` ..... adresář se spustitelnou formou implementace
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `BP_Shchukin_Maksim_2020.pdf` ..... text práce ve formátu PDF