TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Elektronický volební systém

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Petr Vejvoda**
Vedoucí práce: prof. Ing. Zdeněk Plíva, Ph.D.

Liberec 2015

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Electronic voting system

Diploma thesis

Study programme:	N2612 – Electrotechnology and informatics
Study branch:	1802T007 – Information technology
Author:	**Bc. Petr Vejvoda**
Supervisor:	prof. Ing. Zdeněk Plíva, Ph.D.

Liberec 2015

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2014/2015**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Petr Vejvoda**
Osobní číslo:	**M12000335**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Informační technologie**
Název tématu:	**Elektronický volební systém**
Zadávající katedra:	**Ústav informačních technologií a elektroniky**

Zásady pro vypracování:

1. Prostudujte informace o dostupných volebních aplikacích a vytvořte metodiku bezpečné identifikace voliče.
2. Prostudujte metody identifikace a možnosti využití identifikace v prostředí TUL. Prověřte a popište právní rámec celé aplikace.
3. Vytvořte a zprovozněte aplikaci volebního systému.
4. Na různých skupinách "voličů" vyzkoušejte funkčnost a bezpečnost systému.

Rozsah grafických prací: **Dle potřeby dokumentace**

Rozsah pracovní zprávy: **cca 40 až 50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] **Berger J. Elektronický systém voleb a hlasování (E-voting). Diplomová práce ČVUT v Praze, FEL, 2012**

[2] **Huseby H., S. Zranitelný kód, Brno: Computer Press, 2006, ISBN 80-251-1180-6**

[3] **web: cisco. Patrikakis, Ch. - Masikos, M. - Zouraraki, O. Distributed Denial of Service Attacks**

[4] **http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-4/dos_attacks.html, stav k 9.10.2014**

[5] **web: http://www.lupa.cz/clanky/dozral-v-cesku-cas-na-elektronicke-volby-vlada-je-chce-zkusit-pres-datoveschranky/, stav k 9.10.2014**

Vedoucí diplomové práce: **prof. Ing. Zdeněk Plíva, Ph.D.**
Ústav informačních technologií a elektroniky

Konzultant diplomové práce: **Ing. Jiří Jeníček, Ph.D.**
Ústav informačních technologií a elektroniky

Datum zadání diplomové práce: **12. září 2014**

Termín odevzdání diplomové práce: **15. května 2015**

prof. Ing. Václav Kopecký, CSc.
děkan

Technická univerzita v Liberci L.S. - 14 -

prof. Ing. Zdeněk Plíva, Ph.D.
vedoucí ústavu

V Liberci dne 12. září 2014

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 15. 5. 2015

Podpis:

Abstrakt

Cílem této práce je navrhnout a implementovat elektronický volební systém pro potřeby Technické univerzity v Liberci.

Systém jsem implementoval jako Java Webovou aplikaci, která je rozdělena do čtyř oddělených komponent, jenž spolu komunikují pomocí webových služeb. Díky tomu je systém jednoduše nasaditelný na tři servery pro zajištění bezpečnosti.

V praxi jsem vytvořil elektronický volební systém, který zajišťuje kompletní servis organizace voleb, jak na straně administrace s vyhlášením voleb, přes samotnou volbu voličem, až po vyhodnocení voleb. Systém je schopen obsloužit více paralelně probíhajících voleb, kterých se mohou zúčastnit studenti a zaměstnanci celé Technické univerzity v Liberci.

Zavedením této aplikace do praxe dojde k zjednodušení volebního procesu a možnosti zvýšení volební účasti.

Klíčová slova: elektronický volební systém, elektronické volby, Java, RSA, webové služby

Abstract

The goal of the thesis is to design and implement electronic voting system to be used by Technical University of Liberec.

The system is implemented as a Java Web application that is made from four separate components. These components are all web services. That enables the system to be deployed to separate servers to improve security.

The implemented voting system is able to mediate the whole process of electronic elections. Starting with announcement of elections, continuing with user voting and ending with evaluating of the elections. The system is able to run several elections running in parallel. These elections are accessible by all students and employees of the Technical University of Liberec.

Introduction of this voting system will make the voting process easier and will enable more people to participate in elections.

Keywords: electronic voting system, e-vote, Java, RSA, web services

Poděkování

Rád bych poděkoval vedoucímu práce prof. Ing. Zdeňku Plívovi, Ph.D. za metodické vedení a ochotnou pomoc při vyřizování veškerých univerzitních záležitostí.

Děkuji také své rodině za podporu po celou dobu studia, Tínce za trpělivost kterou se mnou má, a všem, kteří mi pomáhali při tvorbě práce.

Obsah

Seznam obrázků

Seznam zkratek

TUL	Technická univerzita v Liberci
FM	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
OIS	Oddělení informačních systémů TUL
ASS	Univerzitní autentizační systém Shibboleth
XML	Extensible Markup Language

1 Úvod

Již od konce 20. století je společnost výrazně ovlivňována rychlým rozvojem nových informačních technologií, a to především internetu. Jeho objevení se stalo velkým milníkem historie a bývá srovnáváno s vynálezem knihtisku [12]. Internet ovlivnil způsob komunikace a stal se tak klíčovou komunikační a informační platformou současnosti. Nejen vědní disciplíny, ale také široká veřejnost byly nuceny reagovat na jeho rozvoj a internet se tak prosazuje ve všech činnostech běžného života a velkou mírou ovlivňuje jeho chod. Není divu, že ani volební systémy nezůstaly vůči tomuto fenoménu imunní.

V celé řadě zemí jsou dnes elektronické technologie, včetně internetu, využívány volebními úřady v různých fázích volebního procesu. Jako příklad můžeme uvést kalkulaci výsledků hlasování. Stále se však řeší otázka, zda nasadit celkový elektronický volební systém pro volby v České republice. Tato otázka je se vzrůstající technickou gramotností občanů čím dál více aktuální.

Téma diplomové práce jsem si vybral především pro jeho aktuálnost a potenciál do budoucna. Věřím, že při studování informací o zkušenostech jiných subjektů s elektronickým hlasováním se dozvím mnoho nových a zajímavých informací.

V první řadě bych se rád informoval o systému fungování klasických papírových voleb a následně si vyhledal a prostudoval jednotlivé práce, které se zabývají požadavky elektronických volebních systémů.

Další oblastí mého zájmu je prostudovat bezpečnostní rizika elektronického volebního systému, a to od zabezpečení jednotlivých serverů po samotné šifrování volebních lístků, kde bude využíváno nejrůznějších kryptografických schémat.

Další neméně důležitou otázkou je identifikace a ověření uživatele se zajištěním anonymity volebních hlasů.

Velkou motivací mi je potencionální využití výsledků mé práce. Nechtěl jsem půl roku pracovat na aplikaci, která bude následně uložena do „šuplíku“. Raději budu navrhovat systém, který by mohl být využit k zlepšení jednotlivých volebních procesů Technické univerzity v Liberci a zjednodušit tak jejich průběh. Pevně proto věřím ve využitelnost aplikace a to také díky faktu, že většina studentů a zaměstnanců přichází do kontaktu s informačními technologiemi denně a nestraní se jejich užívání. Výhledově by tak mohlo dojít nejen ke zvýšení volební účasti na důležitých volbách (např. do akademického senátu), ale také ke vzniku většího množství méně důležitých hlasování (např. spokojenost se stravováním v menze).

Před výběrem tématu bylo ověřeno, že žádný podobný systém není doposud v rámci Technické univerzity v Liberci vyvíjen. Také z tohoto důvodu pro mě byla tvorba elektorického volebního systému velkou výzvou.

V teoretické části bych se nejprve rád seznámil se zkušenostmi jednotlivých států s nasazením elektronických volebních systémů a s problémy, které při tomto procesu vyvstaly. Rád bych také vyhledal a prozkoumal funkčnost a technologie dostupných volebních systémů a seznámil se tak s celkovým průběhem volebního procesu.

Dále se zaměřím na požadavky Technické univerzity v Liberci, rozeberu jednotlivé možnosti řešení vzniklých variant s definováním výhod a nevýhod daného řešení. Z těchto metod pak vyberu ty, které budu ve své aplikaci implementovat.

V implementační části se již zaměřím na přesný popis řešení jednotlivých částí systému s ukázkami důležitých částí kódu a konfigurací.

2 Popis problému a specifikace cíle

2.1 Cíl práce

V současné době probíhají volby v České republice pomocí papírových volebních lístků. Stejně je tomu i na Technické univerzitě v Liberci (dále také TUL). Tento způsob volby je léty ověřený a voliči v něj mají důvěru. Problémem je ovšem nízká volební účast, která může být zapříčiněna nutností osobní účastí na volbách v dané volební místnosti.

Cílem mé práce je navrhnout a implementovat elektronický volební systém, který by bylo možné využít pro potřeby Technické Univerzity v Liberci. Pomocí tohoto systému bude možné pořádat méně důležitá hlasování různého typu, od průzkumů spokojenosti se stravováním, až po přímé volby např. do akademických senátů. Jednotlivá hlasování bude možné vytvořit pro různě definované skupiny voličů, od akademických pracovníků kateder, přes studenty jednotlivých fakult, až po kompletní seznam zaměstnanců univerzity.

Systém by měl být navržen tak, aby dokázal zvládnout obsloužení více paralelně probíhajících hlasování.

V závěru práce by měl být systém otestován na různých skupinách „voličů“ a měla by být prověřena jeho funkčnost a bezpečnost.

Vytvářená aplikace musí zajistit kompletní servis organizace voleb, jak na straně administrace s vyhlášením voleb, přes samotnou volbu voličem, až po vyhodnocení voleb. To vše s maximálním možným zabezpečením proti zneužití.

2.2 Všeobecné požadavky elektronického volebního systému

Elektronickým volebním systémem rozumíme systém, pomocí kterého mohou voliči odevzdat svůj volební lístek v elektronické formě bez nutnosti dostavení se do příslušné volební místnosti.

Elektronický volební systém tedy musí splňovat následující požadavky. Požadavky jsou vytvořeny na základě několika studií, které byly sjednoceny v práci [10].

- **Anonymita** - Po odevzdání hlasu nesmí být zpětně rozklíčovatelná vazba mezi voličem a volebním lístkem.
- **Nevynutitelnost** - Žádný volič nemusí dokazovat třetí straně jak volil.
- **Ověřitelnost** - Každý oprávněný volič musí mít možnost jednoduchého ověření správného započtení svého hlasu.
- **Způsobilost** - Ve volbách smí hlasovat pouze oprávnění voliči.
- **Jedinečnost** - Každému oprávněnému voliči je povoleno ve volbách hlasovat pouze jednou.
- **Bezchybnost** - Elektronický systém musí být bezchybný. Vyskytnutí se jakékoli chyby by mohlo vést ke zneplatnění voleb.
- **Jednoduchost** - Po voliči není požadováno žádné speciální vzdělání k provedení volby.

Pokud by byly všechny tyto požadavky splněny, vzniknul by idální volební systém. To většinou není možné a je nutné provést v jednotlivých bodech částečné ústupky, které se v jednotlivých implementacích řeší různě.

Mezi požadavky, které je složité dodržet patří především *anonymita*. Pokud bychom chtěli dodržet absolutní *anonymitu*, nesměl by se uživatel do aplikace vůbec přihlašovat a nastal by nám problém s *jedinečností* a *způsobilostí*.

Dalším těžko splnitelným požadavkem je *nevynutitelnost*. U elektronického volebního systému většinou nelze zaručit, že volič nevolí pod nátlakem, jelikož může volit ze kteréhokoli počítače připojeného na internet.

2.2.1 Volební protokoly

Volební protokoly jsou soubory pravidel, které nám mají pomoci k dosažení splnění jednotlivých požadavků elektronického volebního systému. Volební protokoly jsou většinou navrženy na základě homomorfní vlastnosti některého kryptografického schématu nebo na podepisování naslepo a anonymních komunikačních kanálech. Volebních protokolů je velké množství, ale žádný z nich není dostatečně jednoduchý a zároveň bezpečný.

Volební protokoly jsem nalezl podrobně popsané v diplomových pracích [11][10].

3 Existující elektronické volební systémy

3.1 Elektronické volby ve světě

Elektronické volby nejsou ve světě žádnou novinkou. Mnoho států se již snažilo nasadit elektronické volební systémy pro volby do krajských zastupitelstev, parlamentu i do senátu. Většinou se ale setkaly s problémy, které jejich další použití znemožnily a státy se vrátily zpět k volbám papírovým. Ve většině vyspělých zemí přesto stále probíhají pilotní projekty a testy možných systémů pro elektronické volby.

Myslím si, že s nastupující mladou generací a s rostoucí počítačovou gramotností budou elektronické volby nevyhnutelné a je jen otázkou času, kdy k nim přistoupí i Česká republika.

Po prostudování několika prací popisujících zkušenosti a poznatky jednotlivých států s používáním elektronických voleb, se pokusím sepsat několik nejzajímavějších informací a také problémy, se kterými se při tomto procesu státy setkaly. Poměrně stručný souhrn lze nalézt např. na [3] nebo v [12].

Primárním problémem je bezpečnost a napadnutelnost elektronických systémů, tou se ale v této práci podrobně zabývat nebudu. Jednotlivá zabezpečení systémů proti útoku jsou podrobně popsaná v diplomové práci [10].

3.1.1 Německo

První nasazení elektronických voleb v Německu bylo v roce 1999 při volbách do Evropského parlamentu a následně ve volbách do Spolkového sněmu. Od roku 2005 proběhlo několik soudních sporů ohledně bezpečnosti a spolehlivosti volebních přístrojů a v roce 2009 byly elektronické volby Spolkovým soudem zakázány.

Důvodem bylo nedostatečné zabezpečení přístrojů proti manipulaci a neumožnění kontroly správnosti hlasu samotným voličem. Použití volebních přístrojů podle výroku soudu bylo neslučitelné s ústavní zásadou veřejnosti voleb kde má každý volič právo na kontrolu průběhu každé fáze voleb.

3.1.2 Estonsko

Estonsko je jedním z mála států, který hodnotí projekt elektronických voleb za velmi úspěšný. První hlasování přes internet bylo umožněno v komunálních volbách v roce 2005, následně pak v roce 2007 v parlamentních volbách, v roce 2009 ve volbách do Evropského parlamentu a znovu v komunálních volbách. Před spuštěním elektronických voleb byl systém několikrát prověřen hackery proti elektronickému útoku a v průběhu voleb byly všechny servery hlídány policií pro zamezení jakékoli manipulace s hardwarem.

Každý občan v Estonsku má přidělenou svou elektronickou identitu, ve které jsou uloženy dva certifikáty. Jeden pro autentizaci voliče a druhý pro elektronický podpis. Tato skutečnost zásadně přispívá k jednodušší a hlavně bezpečnější identifikaci voliče v rámci internetu. Dále mají ve svém elektronickém volebním systému důmyslně propracovanou auditní komponentu, pomocí které je možné ověřit, jak byl započten hlas voliče. Pokud započten nebyl, je možné bezpečně zjistit, proč byl elektronický volební lístek zamítnut.

Podrobné informace o Estonském volebním systému nalezneme na stránkách [1].

3.2 Dostupné volební aplikace

3.2.1 Helios

Elektronický volební systém Helios, zveřejněný pod licencí open-source, je implementován nad frameworkem Django, který je napsaný v Pythonu. Helios je testován a plně funkční s implementací databázové vrstvy pomocí PostgreSQL i když by měl fungovat s jinými databázemi, které framework Django podporuje.

K aplikaci Helios je možné udělat si jakýkoli frontend, který si data o volbách získává z Heliosu pomocí HTTP požadavků. Odpovědi na ně jsou zasílány ve formátu JSON, který je zcela obecný a ideální pro přenos dat v libovolném programovacím nebo skriptovacím jazyce.

Helios verze 4 byl vydán v srpnu roku 2012 a od té doby až do dnes je zmíněno, že dokumentace ještě není kompletní. Vypadá to tedy, že se jeho vývoj tou dobou pozastavil. Veškeré informace jsou zjištěny z oficiálního webu [2].

3.2.2 Simply Voting

Simply Voting je komerční volební systém, ve kterém je nutné nejdříve projít registrací a až jako registrovaný uživatel můžeme vytvářet volby. Systém nabízí několik možných identifikací voličů. V nabídce je i identifikace pomocí Shibbolethu, která by byla výhodná pro použití v rámci TUL. Další výhodou tohoto řešení je, že nepotřebujeme žádný vlastní hardware, veškeré elektronické volby jsou realizované společností SimplyVoting. V rámci registrace je dostupná i telefonická podpora. To vše je bohužel vyváženo částkou, kterou se platí za každé vytvořené volby. Například za organizaci základních voleb pro 100 voličů si účtují částku 200$.

Veškeré informace jsou shromážděny z oficiálních webových stránek Simply Voting [8].

4 Požadavky v rámci Technické Univerzity v Liberci

4.1 Využitelnost systému v jednotlivých sférách TUL

Elektronický volební systém má být univerzální s možností pořádání elektronických voleb napříč všemi studenty i zaměstnanci TUL. Je tedy nutné aby byla navržena identifikace voliče taková, kterou mají automaticky všichni studenti i zaměstnanci TUL. Nikdo by tedy neměl být znevýhodněn tím, že se nebude moci účastnit elektronických voleb, protože nevlastní nějakou z věcí k dokončení volebního procesu.

Zároveň bude nutné vyřešit přístup k databázi všech studentů a zaměstnanců TUL pro získání potřebných údajů o voličích. Důležité tedy je, aby systém obsáhnul všechny voliče, kteří mají co dočinění s Technickou univerzitou v Liberci.

4.2 Metoda bezpečné identifikace voliče

Pro bezpečnou identifikaci voliče bylo zapotřebí navrhnout a implementovat dvoufázový ověřovací proces. První fází bude přihlášení do volební aplikace pomocí univerzitního autentizačního systému založeného na technologii Shibboleth (dále jen ASS). Druhá fáze ověření se spustí těsně před samotným odesláním hlasu. V úvahu připadaly následující možné varianty, z nichž bylo potřeba vybrat jednu a tu následně implementovat.

4.2.1 GUID uvedené na dokumentu Rozhodnutí o přijetí na TUL

GUID/UUID v4 je 32 hexa znaků (52736424-6b1c-31a2-6b43-11f56145d634) dlouhý jednoznačný identifikátor studenta, který je uložen ve STAGu a je tisknut na Rozhodnutí o přijetí. Pokud je uchazeč v témže roce přijat na více oborů, je mu pro všechny generován stejný GUID. Naopak, hlásí-li se uchazeč v různých letech, pro každý rok mu je generováno jiné GUID.

Druhá fáze ověření založená na zadání GUID pro potvrzení odeslání hlasu je pro studenta velmi nepohodlná. Musel by nalézt svoje Rozhodnutí o přijetí a teprve poté by mohl dokončit hlasování. Tato skutečnost by nejspíše vedla ke snížení účasti ve volbách a pro studenta by bylo jednodušší se k volbám dostavit osobně.

4.2.2 Elektronické čipové karty

Čipové karty by také poskytovaly dostatečně bezpečnou identifikaci voliče. K jejich využití by bylo nutné, aby každý volič vlastnil čtečku těchto karet, případně by musel využít univerzitního volebního PC. Dalším problémem by mohla být skutečnost, že ne všichni zaměstnanci a studenti tyto elektronické karty vlastní.

Druhá fáze ověření pomocí elektronických karet byla díky své složitosti také vyřazena z nabízejících se řešení.

4.2.3 Odeslání náhodného ověřovacího kódu nezávislou cestou

Další variantou bezpečného ověření jsme se nechali inspirovat u elektronického bankovnictní. K ověření se zde využívá odeslání náhodně vygenerovaného řetězce s předem danou dobou platnosti na mobilní telefon klienta. Telefonní číslo musí být zadáno přímo na pobočce banky osobně klientem, říkáme tedy, že je „ověřené“. Tento typ ověření nemusí být zasílán pouze na mobilní telefon, ale jako varianta se nabízí i využití univerzitní e-mailové adresy. V následujícím textu popíši výhody a možné problémy při využití ověřeného telefonního čísla, nebo univerzitní e-mailové schránky.

Odeslání na univerzitní e-mail

Každý student i zaměstnanec TUL vlastní univerzitní e-mailovou adresu. Valná většina studentů tuto e-mailovou adresu nepoužívá a univerzitní záležitosti řeší pomocí své privátní adresy. Svou privátní adresu si student či zaměstnanec může doplnit do univerzitní databáze, ale není to jeho povinnost, tudíž ji většina z nich vyplněnou nemá. Privátní adresu tedy nelze s jistotou využít.

Ověření pomocí univerzitního e-mailu je bohužel také nedostatečné, a to kvůli použití stejných autentizačních údajů do e-mailové schránky jako do aplikace elektronického volebního systému. V obou případech se autentizace provádí pomocí ASS. Pokud by se tedy potencionální útočník dostal k druhé fázi ověření, je jasné, že musel projít první fází, kterou je výše zmíněné přihlášení pomocí ASS.

Získání e-mailové zprávy s ověřovacím kódem by proto již nebyl problém a tato druhá ověřovací fáze by neměla smysl.

Ověření přes SMS na ověřené telefonní číslo

Řešením, které by odstranilo bezpečnostní problém výše popsané univerzitní e-mailové adresy, je odeslání ověřovacího kódu formou SMS zprávy na ověřený telefon voliče. Mobilní telefon vlastní valná většina studentů a jeho používání je na denním pořádku. Toto ověření bude tedy pro voliče pohodlné a nemělo by jim činit potíže.

Ve své diplomové práci jsem se pro výše popsané výhody a nevýhody jednotlivých řešení dvoufázového ověření voliče rozhodl použít zaslání kódu na ověřené telefonní číslo společně s využitím univerzitní e-mailové schránky.

Problémem je, že na TUL ještě neexistuje SMS brána, přes kterou by bylo možné odesílat SMS zprávy pro účely aplikace elektronického volebního systému. Na SMS bráně by se již mělo pracovat a v budoucnu by ji mělo být možné využívat. Elektronický volební systém tedy navrhnu tak, aby mohl být v budoucnu jednoduše rozšířen pro odesílání ověřovacího kódu pomocí SMS zpráv.

4.3 Vytváření seznamu voličů

4.3.1 Seznam voličů

Seznamem voličů rozumíme skupinu jedinců, kterým je povoleno hlasovat ve volbách.

Pro budoucí používání volebního systému napříč celou TUL je nutné navrhnout metodu vytváření seznamu voličů. Aplikace musí zajistit, aby všechny osoby, které mají oprávnění vytvářet nové volby, mohly jednoduše zadávat, který student či zaměstnanec TUL je v daných volbách oprávněn volit.

Tento seznam bude administrátor voleb definovat zadáváním různých kritérií, která byla navržena tak, aby bylo možné studenty a zaměstnance dělit co nejpodrobněji. Bude možné vytvořit volby například pro celou fakultu mechatroniky, nebo pouze pro pracovníky z určitého pracoviště. Volební administrátor má také možnost vytvořit seznam voličů pouze z přesně definovaných osob, které spolu vůbec nemusejí souviset.

4.3.2 Generování seznamu voličů

Při tvorbě nových voleb jsou důležitá dvě data. Prvním je datum vzniku volby a druhým spuštění samotného hlasování. Otázkou tedy je, při kterém z těchto dvou dat, by se měl seznam voličů vygenerovat.

Při vytvoření voleb

Pokud vytvoříme seznam voličů v době, kdy je vytvářena nová volba, přestože je její spuštění plánováno na vzdálenější datum, může nastat problém. Uveďme si imaginární příklad, kdy administrátor vytvoří volby, například 1. dubna, ale spuštěny budou až 1. května. Pokud však v tomto časovém období univerzita přijme nové studenty či zaměstnance, nebo stávající odejdou, nepromítne se tato změna do již vytvořeného seznamu voličů. Tento fakt je hlavní nevýhodou výše uvedené metody.

Metoda má však i své nesporné výhody. Administrátor má možnost při tvorbě volby zkontrolovat seznam voličů. Také samotný volič může být již při vytvoření volby informován o možnosti účastnit se volby, případně o konkrétním datu spuštění.

Při spuštění voleb

Druhou možností je generovat seznam voličů těsně před spuštěním voleb. Tím se vyřeší problém s nekonzistencí oprávněných voličů mezi dnem vytvoření seznamu a dnem spuštění voleb popsaný v předchozím odstavci.

Nevýhodou je, že volič je informován o možnosti účastnit se na hlasování velmi krátkou dobu před samotným spuštěním voleb. Tvorba seznamu voličů je také závislá na funkčnosti Kartového systému TUL a pokud tento nebude v daný okamžik dostupný, nemusí být celá volba spuštěna.

Aby měly jednotlivé komponenty aplikace vazby pouze na ty části systému, které jsou pro jejich funkčnost nezbytné (viz. diagram komponent 5.2), rozhodl jsem se pro první výše popsanou možnost. Tímto způsobem zajistíme maximální transparentnost aplikace a nezávislost jednotlivých komponent. Seznam voličů tedy bude generován okamžitě s vytvořením voleb.

4.3.3 Kartový systém TUL

Kartový systém pod správou pověřeného pracovníka oddělení informačních zdrojů (dále jen OIS) shromažďuje informace o veškerých studentech a zaměstnancích, včetně jejich závazků vůči TUL. Data se každý den aktualizují z několika vedlejších databází (Vema, STAG, ...) a sjednocují se do databáze jediné, kterou je Kartový systém TUL.

Nad touto databází jsme společně navrhli strukturu webové služby, kterou poté Ing. Kopetschke implementoval. Poté, co této webové službě předáme požadavek s přesnými kritérii, vrátí se nám seznam vyhovujících studentů a zaměstnanců.

Webovou službu kartového systému využívám pro načítání seznamů voličů. Podrobnější popis metod konfigurace webové služby kartového systému naleznete v kapitole 5.6.

4.4 Sčítání hlasů a zveřejnění výsledků voleb

Při prvních návrzích aplikace pro sčítání volebních hlasů a zobrazení výsledků voleb jsem narazil na nedůvěřivost voličů k elektronickému sčítání hlasů. Voliči se nelíbí, pokud proběhnou volby, ty jsou sečteny nějakým strojem a zobrazeny jsou jen výsledky. Nedůvěřivost občanů k elektronickým krabičkám mě vedla k úvaze, zda u výsledků voleb nezveřejnit všechny volební lístky se svými identifikátory a kandidátem, kterému jsou určeny.

Volič si poté může ověřit započtení svého volebního lístku pomocí jeho identifikátoru a případně si přepočítat všechny hlasy ve volební urně. Tím by se voličova nedůvěra ve sčítání rozplynula. Na tomto principu jsou založené i některé volební protokoly [11].

5 Implementace elektronického volebního systému

5.1 Základní struktura systému

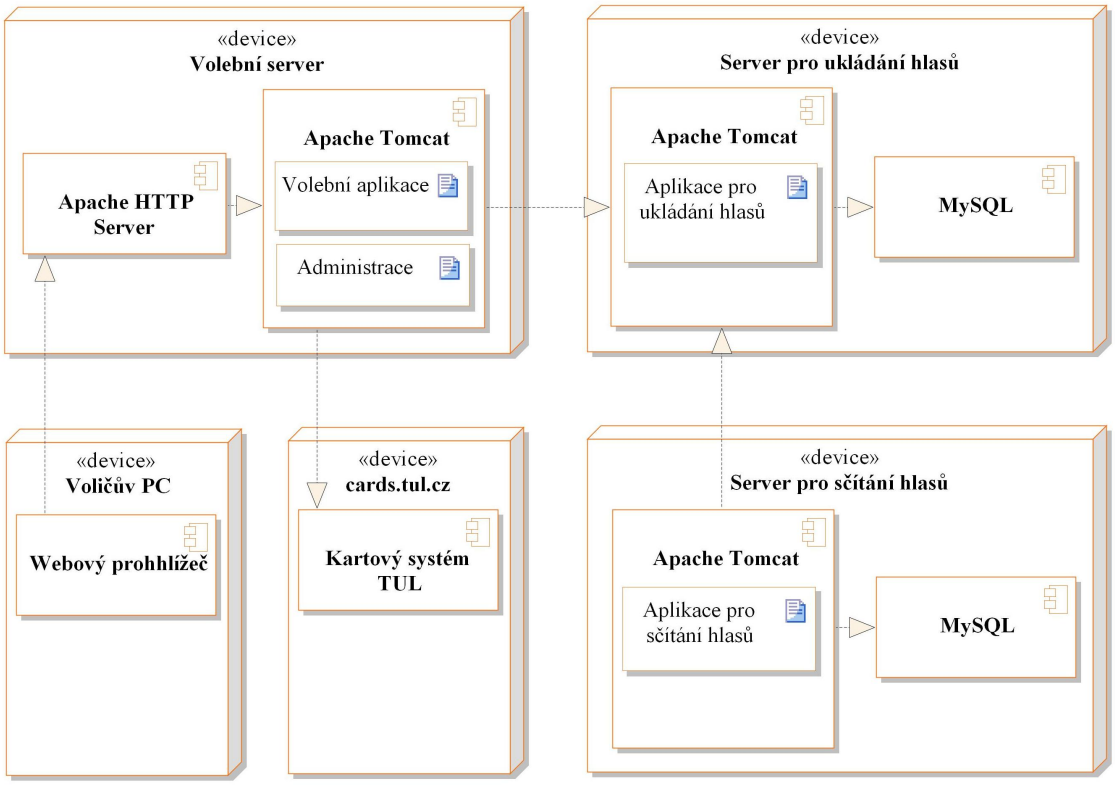
Základní strukturu elektronického volebního systému jsem navrhoval s ohledem na maximální možnou univerzálnost a s důrazem na zajištění všech požadavků elektronického volebního systému. Požadavky byly definovány v kapitole 2.2.

Aplikaci jsem tedy rozdělil do čtyř dílčích komponent znázorněných na obrázku 5.2 a pro jejich vzájemnou komunikaci jsem použil webových služeb, které jsou jednoduše implementovatelné ve valné většině programovacích jazyků a jsou proto velmi univerzální pro budoucí vývoj aplikace.

Pro zajištění bezpečnosi volebního systému by neměly být jednotlivé komponenty umístěny pouze na jednom serveru, ale měly by být distribuovámy mezi minimálně dva, nejlépe však tři servery. Při vývoji aplikace v rámci diplomové práce jsem měl k dispozici pouze jeden virtuální server, tudíž jsem veškeré komponenty elektronického volebního systému testoval na jednom virtuálním serveru. Každá z komponent je ovšem jednoduše konfigurovatelná a přenesitelná mezi různými servery. Při nasazení v praxi nebude tedy problém každou komponentu umístit na jiný server.

Ideální model nasazení jednotlivých částní systému je navržen na obrázku 5.1. Dané rozmístění komponent je koncipováno pro maximální bezpečnost elektronického volebního systému. Každý ze serverů plní pouze svou specifickou funkci a neměly by na něm běžet žádné další aplikace, které by mohly narušit jeho bezproblémový chod. Umístění aplikace pro ukládání hlasů a aplikace pro sčítání hlasů na rozdíl-

né servery, nám tak zvyšuje zabezpečení systému. Pokud by se případný útočník pokusil porušit anonymitu systému, musel by prolomit zabezpečení dvou serverů zároveň. Při prolomení pouze jednoho z nich, by nikdy nebyl schopen dekódovat volební lístky. Server pro ukládání hlasů by vůbec neměl být přístupný z veřejné internetové adresy a připojení bude akceptováno pouze z lokálních adres volebního serveru a serveru pro sčítání hlasů.

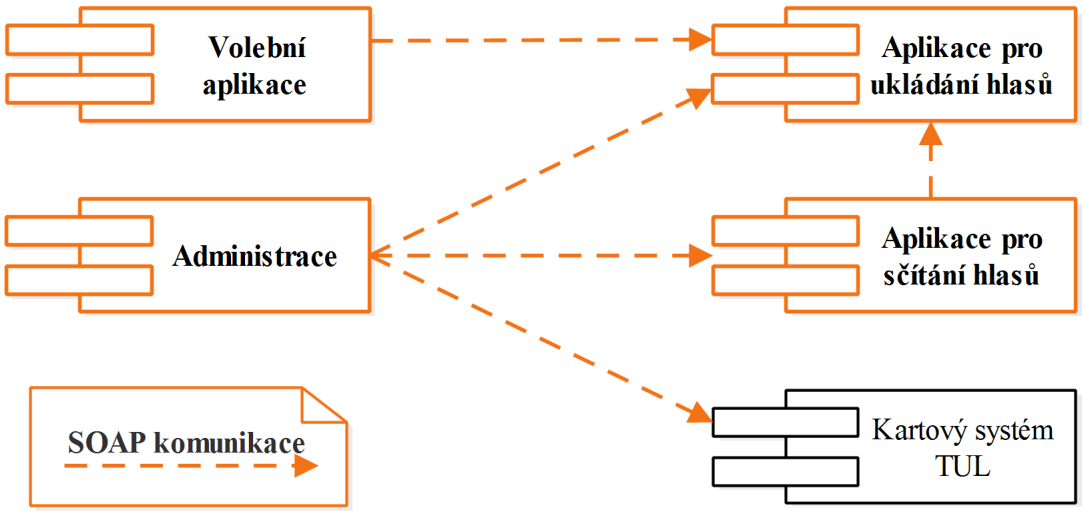


Obrázek 5.1: Diagram nasazení

5.1.1 Vzájemná komunikace jednotlivých komponent volebního systému

Zabezpečení komunikace

Zabezpečení komunikace mezi komponentami systému je realizováno ve dvou krocích:



Obrázek 5.2: Diagram komponent Elektronického volebního systému

1. **Firewall** - Konfigurace pravidel firewallu je důležitým krokem při nastavování všech serverů elektronického volebního systému. Výchozí politikou jsou všechny služby zakázány a povolíme jen ty, které jsou nutné pro správnou činnost systému. Povolovací pravidla doplníme o přesně definované IP adresy serverů, ze kterých mohou požadavky přicházet.

2. **Autorizační tokeny** - Je náhodně vygenerovaný řetězec dostatečné délky, který identifikuje klienta přistupujícího k webové službě. Token musí být bezpečnou cestou distribuován do obou komponent, které spolu komunikují.

5.2 Volební aplikace

5.2.1 Použité technologie

Volební aplikace je implementována jako Java Webová aplikace s využitím následujících frameworků a knihoven.

Spring MVC

Spring MVC je aplikační rámec pro vývoj webových aplikací ve stylu Model, View a Controller.

V MVC aplikaci jsou modelem reprezentovány informace, s nimiž aplikace pracuje a ty jsou následně předány do View, které se stará o jejich interaktivní prezentaci uživateli.

Veškeré požadavky ze strany klienta zpracovává Controller. Je definován jako třída s anotací `@Controller`. U jakékoli metody dané třídy můžeme využít anotaci `RequestMapping`, pomocí níž definujeme, jaký request ze strany klienta bude metoda zpracovávat. Návratovou hodnotou metody je řetězec, definující použitý View. Podrobné infomace o Srping MVC nalzeneme na stránkách oficiální dokumentace [9].

JavaServer Pages a JSTL

JavaServer Pages (dále jen JSP) je technologie pro vývoj dynamických HTML stránek. Při tvorbě se využívá primárně HTML a Java. Soubory mají přípony `.jsp` a nacházejí se ve složce `/WEB-INF/jsp/`.

JSTL je knihovna pomocí které můžeme v JSP využívat mnoho dalších funkcí pouze pomocí XML tagů. JSTL nabízí pět knihoven, kde každá obsahuje různé typy funkcí (Core, XML, Formatting, SQL, Functions).

JAX-WS

JAX-WS je API pro využívání webových služeb. Jelikož webové služby využívají serializaci objektů do XML, je potřeba ještě knihovna JAXB, která toto zajišťuje.

Implementace webové služby s využitím JAX-WS je velmi jednoduchá. Jako základ nám stačí vytvořit jednu třídu a přidat ji anotaci `WebService`. Následně veškeré metody, které chceme zpřístupnit skrze webovou službu oanotujeme jako `@WebMethod`.

Posledním krokem je vytvořit soubor `sun-jaxws.xml` ve složce `WEB-INF`, ve kterém definujeme koncový bod webové služby. Podrobné informace o JAX-WS nalezneme na [4], jednoduchý a přehledný tutoriál jsem nalezl na [5].

5.2.2 Identifikace voliče pomocí Shibbolethu TUL

Shibboleth je systém poskytující službu *Single Sign-On* (jednotné přihlášní). Uživatel může pomocí jednoho přihlášení využívat více síťových zdrojů. Shibboleth umožňuje získávat informace o úspěšné autentizaci uživatele a poskytovat o něm údaje, které mohou být využity i pro jeho autorizaci.

Systém Shibbolethu dělíme na dvě základní části:

- **Service provider** - dále jen SP, poskytovatel služby, vykonává proces SSO, musí být konfigurován na serveru, kde je nasazena Volební aplikace elektronicého volebního systému
- **Identity provider**- dále jen IdP, poskytovatel identit, ověřuje uživatele, je spravován centrálně v rámci TUL

Identifikace voliče za pomoci Shibbolethu je realizována instalací modulu podporujícího Shibboleth do Apache HTTP Serveru. Pokud se volič pokusí přistoupit do volební aplikace, jeho požadavek nejprve zpracuje Apache HTTP Server, který ověří ze SP, zda je volič úspěšně autentizován. Pokud není, přesměruje uživatele na přihlašovací formulář IdP. Pokud proběhlo přihlášení úspěšně, je uživatel přesměrován zpět do volební aplikace. Jeho požadavek o přístup k volební aplikaci opět zachytí Apache HTTP Server a jelikož je uživatel autentizován, vrátí mu požadovanou webovou stránku.

5.2.3 Struktura aplikace z pohledu voliče

Při návrhu volební aplikace jsem se snažil dbát na jednoduchost a intuitivnost ovládacích prvků a tím zjednodušit orientaci v aplikaci. Samotné hlasování jsem rozdělil do pěti následujících kroků, které by měly být intuitivní.

1. **výběr hlasování** - Fáze, do které se klient dostane po úspěšném přihlášení do volební aplikace pomocí Shibbolethu. Volič zde nalezne seznam jemu dostupných voleb s informací, od kdy jsou volby aktivní. V této fázi také nalezne seznam voleb, v kterých již hlasoval.

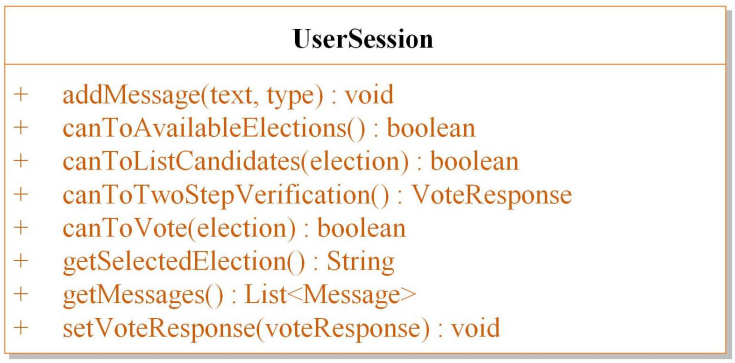
2. **volba kandidáta** - Voliči se zobrazí seznam všech kandidátů, které může volit.

3. **odeslání hlasu** - Při fázi odesílání hlasu se ve voličově počítači vygeneruje náhodný identifikátor, jenž se spolu se zvoleným kandidátem zašifruje veřejným klíčem sčítacího serveru a odešle na server volební aplikace. Nezakódovaný zvolený kandidát a vygenerovaný identifikátor se ještě pomocí JavaScriptu uloží do lokálního úložiště webového prohlížeče voliče.

4. **ověření voliče** - Pokud volič provádí hlasování ve volbách, které požadují dvoufázové ověření, je zobrazena stránka s informací, kam byl odeslán ověřovací kód (e-mail, sms) a políčky pro zadání kódu.

5. **souhrnné informace** - Pokud všechny předchozí body proběhly úspěšně, je voliči zobrazena informace o úspěšném hlasování. V případě, že se v některém z předchozích kroků vyskytla chyba, je volič automaticky vrácen do bodu 1 a je mu zobrazeno odpovídající chybové hlášení.

Při vstupu voliče do volební aplikace je vytvořen objekt `UserSession`, který se přenáší s voličem po celou dobu jeho práce s Volební aplikací. Hlavní funkcionalitou navrženého objektu je kontrola správného průchodu voliče volební aplikací. Pro zvýšení bezpečnosti aplikace není možné ručně měnit URL adresu, volič se může pohybovat pouze pomocí odkazů ve stránce. Pokud by si adresu odkazu změnil ručně a neprocházel podle osnovy definované výše, bude přesměrován na úvodní stránku aplikace a zobrazí se upozorňující hlášení. Metody definované v objektu UserSession jsou zobrazeny na obrázku 5.3.



Obrázek 5.3: Třída UserSession

5.2.4 Kontrola kompatibility použitého internetového prohlížeče

Při načtení volební aplikace do internetového prohlížeče si aplikace musí ověřit, zda voličův prohlížeč podporuje všechny potřebné funkce pro správné provedení hlasování. Při prvním spuštění se tedy provádí soubor testů, ve kterém se ověří, zda je voličův prohlížeč kompatibilní s aplikací. Pokud by jeden z testů neprošel, je voliči zobrazeno chybové hlášení a hlasování není umožněno. Soubor s testy se nachází ve složce `/resources/init-test.js` a obsahuje následující testy.

- **Ověření šifrování/dešifrování pomocí klíčového páru**
  Funkce `testEncryption()` obsahuje RSA klíčový pár uložený ve formátu Base64. Pomocí JavaScriptové utility JSEncrypt nejprve zakóduje a následně zpět dékóduje testovací text. Pokud se vstupní a výstupní text rovná, test proběhl úspěšně.
- **Local Storage** - Funkce `testLocalStorage()` zkontroluje, zda prohlížeč podporuje a má povolené ukládání proměnných do lokální paměti.

5.2.5 Jazyková lokalizace

Pro jayzkovou lokalizaci volební aplikace jsem využil JSP - Standard Tag Library, dále jen JSTL. Pomocí této knihovny lze jednoduše vytvořit multijazyčnou aplikaci.

Ukázka kódu 5.1: Inicializace jayzkové lokalizace

```
<fmt:setLocale value="${language}" scope="session" />
<fmt:setBundle basename="VoteForwardingServer.languages.text"
               scope="session" />
```

Ukázka kódu 5.1 je definována v souboru `/WEB-INF/tag/layout.tag` a inicializuje celou vícejazyčnou lokalizaci aplikace. Hodnota proměnné `${language}` definuje uživatelem zvolený jazyk a v parametr basename v elementu `fmt:setBundle` definuje v jakým balíku se nacházejí lokalizační soubory. Bude vybrán soubor s názvem `VoteForwardingServer.languages.text_${language}`.

Pokud chceme v aplikaci vypsat jakýkoli text, musíme to provést pomocí elementu `<fmt:message key="VALUE">` kde za hodnotu VALUE dosadíme klíč pod kterým je uložena požadovaná zpráva.

5.2.6 Šifrování volebního lístku a jeho odeslání

Šifrování volebního lístku se provádí pomocí utility JSEncrypt. Při načítání kandidátů z aplikace pro ukládání hlasů je společně s kandidáty předán i RSA veřejný klíč ve formátu Base64 pro šifrování hlasů v daných volbách. Privátní klíč pro dešifrování se v tu dobu nachází pouze na Serveru pro sčítání hlasů a není nikomu přístupný.

Pokud volič potvrdí zvoleného kandidáta, spustí se sekvence následujících operací.

Vytvoření volebního lístku

Vytvořený volební lístek je typu XML elementu a jeho ukázka je v ukázce kódu 5.2. Strukturu XML jsem zvolil pro její univerzálnost, přehlednost a hlavně její jednoduché parsování v aplikaci sčítacího serveru. Volební lístek obsahuje pouze dva údaje. Identifikační číslo zvoleného kandidáta a jednoznačný identifikátor volebního lístku který je náhodně vygenerován při vytváření volebního lístku.

Ukázka kódu 5.2: Dekodovaný volební lístek

```
<ballot uuid="a2s4e55w6" candidate="35" />
```

Uložení do paměti prohlížeče

Na konci volebního procesu je nutné voliči zobrazit, komu dal svůj hlas a jaký je vygenerovaný identifikátor jeho volební obálky. Nesmí být porušen požadavek volební anonymity, proto je nutné si tyto informace uložit pouze do webového prohlížeče. Aplikace si je musí uchovat napříč několika přesměrováními. Jako vhodné řešení se mi tedy jevilo využití Local Storage, kam je možné si uložit až 5MB dat a má podporu ve většině moderních prohlížečích. Seznam podpory Local Storage v jednotlivých prohlížečích je k nalezení na [7].

Šifrování

Volební lístek ve formátu XML elementu se pomocí JavaScriptové knihovny JSEncrypt zašifruje veřejným klíčem. Vznikne řetězec ve formátu Base64.

Odeslání

Zašifrovaný volební lístek se odešle na webový server volební aplikace, tam je podepsán privátním klíčem volební aplikace a odeslán do aplikace pro sčítání hlasů. Server volební aplikace lístek podepisuje pro zamezení změny volebního lístku při přenosu mezi volební aplikací a aplikací pro ukládání hlasů.

Ideální variantou by bylo, pokud by každý student měl svůj elektronický podpis a volební lístek podepsal již u sebe v počítači, na kterém provádí volbu. Studenti a zaměstnanci TUL ovšem vlastní elektronické podpisy nemají, tudíž toto zabezpečení zatím není možné.

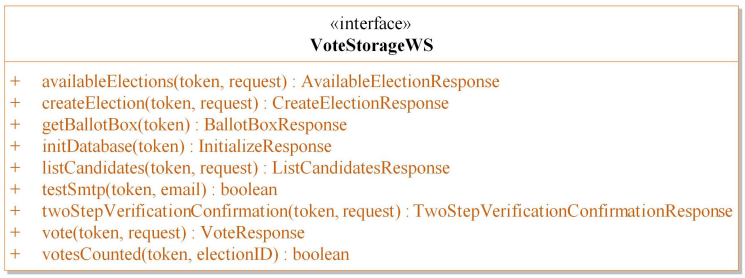
5.3 Aplikace pro ukládání hlasů

Aplikaci pro ukládání hlasů jsem navrhl jako java webovou aplikaci implementující jednu webovou službu obsahující metody znázorněné na obrázku 5.4.

5.3.1 Použité technologie

- **Hibernate** - Framework, který umožňuje objektově-relační mapování (ORM). Je jednou z implementací Java Persistance API (JPA). Mapování modelů jsem implementoval pomocí anotací u atributů objektů.
- **JAX-WS** - Popsáno již v kapitole 5.2.1.

5.3.2 Komunikační rozhraní aplikace



Obrázek 5.4: Rozhraní webové sluby Aplikace pro ukládání hlasů

K aplikaci pro ukládání hlasů lze přistupovat pouze pomocí webové služby viditelné na obrázku 5.4. Každé metodě musí být předán validní identifikační token (5.1.1), jinak metoda vyhodí výjimku. Webová služba obsahuje tři typy metod podle toho, jaká z komponent systému k dané metodě může přistupovat. Rozdělení metod je následující:

1. Volební aplikace
    - **availableElections** - Na základě předaného identifikátoru voliče metoda vrátí seznam dostupných voleb.

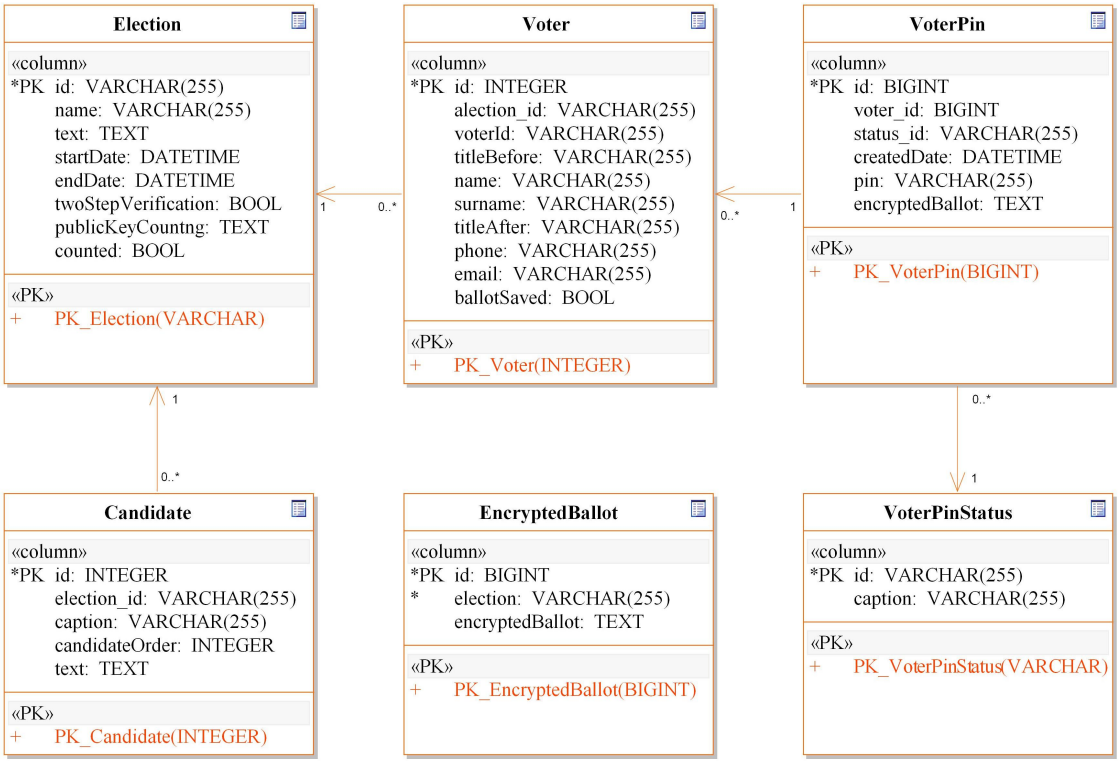
- **listCandidates** - Metoda vrátí plnou strukturu volby obsahující seznam kandidátů i veřejný klíč k šifrování volebního lístku.
- **vote** - Metodě se předává identifikátor voliče, identifikátor volby, zašifrovaný volební lístek a ještě je přiložen elektronický podpis zašifrovaného volebního lístku vytvořený pomocí privátního klíče Volební aplikace pro zvýšení bezpečnosti.

  V návratovém objektu je nejdůležitější hodnotou atribut `status`, podle ktrého volební aplikace pozná, zda je potřeba voliče dvoufázově ověřit, nebo zda byla volební obálka úspěšně uložena.
- **twoStepVerificationConfirmation** - Metoda pro zadání ověřovacího kódu při dvoufázovém ověření.

2. Aplikace pro sčítání hlasů
   - **getBallotBox** - Metoda vracející ukončené volby ke sčítání ve formě volební urny, je volána periodicky ze sčítací aplikace.
   - **votesCounted** - Označí volby za ukončené a sečtené, a dále se tak nebudou voličům zobrazovat ve volební aplikaci.
3. Administrace
   - **createElection** - Metodě se předává objekt se strukturou nových voleb a seznam voličů.
   - **initDatabase** - Metoda pro usnadnění naplnění databáze výchozími hodnotami.
   - **testSmtp** - Metoda pro spuštění testu komunikace se SMTP serverem. Pokusí se odeslat testovací e-mail na adresu, která je jí předána.

5.3.3 Struktura databáze

Strukturu databáze bylo potřeba navrhnout s ohledem na začlenění veškerých požadavků na elektronický volební systém. Přes snahu vytvořit databázové schéma v



Obrázek 5.5: Model databáze

co nejvyšší Normální formě jsem musel udělat částečné ústupky, a proto se pokusím vysvětlit problémy a zvolené řešení struktury databáze.

Tabulka Election

Tabulka Election obsahuje základní strukturu voleb. Každá volba má jednoznačný identifikátor ve formě UUID délky 32 znaků. Zamezí se tím možnému útočníkovi možnost jednoduchého odhadnutí identifikátoru jiných platných voleb.

Tabuka Candidate

Obsahuje informace o kandidátech. Každý kandidát je jednoznačně identifikován svým `id` a je ve vztahu N:1 s tabulkou Election.

Atribut `candidateOrder` slouží k nastavení pořadí při výpisu kandidátů ve volební aplikaci. Tento atribut by měl být nastavován pouze při průzkumech spokojenosti

a podobných typech hlasování. Při volbách, např. do akademických senátů, by se měl používat s opatrností, aby nevznikl problém zvýhodňování některých kandidátů. Proto by měl radši zůstat nastaven na `null`.

Tabulka Voter

Do tabulky Voter je ukládán seznam voličů. Každý řádek tabulky definuje oprávnění jednoho voliče volit v konkrétních volbách. Pokud má volič právo volit ve více volbách, tabulka bude obsahovat více řádků se stejným identifikátorem voliče a zároveň budou v každém z těchto řádků vyplněny voličovy osobní údaje, jako je *jméno*, *příjmení*, *tituly*, *telefon* a *email*. Tím vznikne v datech částečná redundance, kterou by bylo možné odstranit využitím nové spojovací tabulky a tím vytvoření vztahu M:N mezi tabulkami `Voter` a `Election`. Tuto redundanci jsem se však rozhodl akceptovat a vazbu M:N nevytvářet z následujícího důvodu.

Po vytvoření elektronických voleb je seznam voličů uzavřen a nemělo by se s ním před spuštěním voleb a při jejich běhu dále manipulovat ve smyslu, že se změní nějaké kontaktní nebo osobní údaje voliče. Pokud bychom využili vztahu M:N popsaného výše, mohla by zmíněná manipulace nastat v případě, kdy se vytvoří nové volby za běhu jiných voleb. Jelikož se vždy při vytváření nových voleb načítá aktuální seznam voličů z kartového systému, mohly by se přepsat staré kontaktní údaje v tabulce `Voter` za nové.

Při uložení, navrženém na obr. 5.5, akceptujícím danou redundanci, tato změna kontaktních údajů v seznamu voličů nenastane. Při vytvoření nových voleb, se vytvoří nový řádek v tabulce `Voter`.

Tabulka VoterPin

Pokud volič odeslal volební obálku a volby požadují dvoufázové ověření, spustí se následující životní cyklus záznamu v této tabulce.

1. Je vytvořen nový záznam, který obsahuje vygenerovaný ověřovací kód, který byl odeslán voliči, zašifrovaný volební lístek, datum vytvoření a je mu nastaven status `VALID`.

2. Po zadání ověřovacího kódu voličem se vyhledá záznam ve stavu `VALID`. Pokud vypršela platnost, změní se stav na `TIMEOUT`. Pokud byl rozdílný pin, je stav změněn na `FAILED`. Pokud projdou předchozí dvě podmínky, šifrovaný volební lístek z tohoto záznamu se uloží do tabulky EncryptedBallot a v tomto záznamu se nastaví na `null` a zároveň se změní status na `ACCEPTED`.

Tabulka EncryptedBallot

Obsahuje zašifrované volební lístky, které již nemají žádnou vazbu na voliče. Z této tabulky jsou poté exportovány do sčítací aplikace.

5.3.4 Zajištění dvoufázového ověření voliče

Dvoufázové ověření voliče může probíhat přes univerzitní e-mail nebo formou sms zprávy. Implementace těchto dvou metod se nachází ve třídách `votestorageserver.TwoStepVerificationEmailComponent` a `votestorageserver.TwoStepVerificationEmailComponent` v každé z těchto tříd stačí implementovat metodu `sendPIN(VoterPin voterPin)` která vykonává pouze samotné odeslání ověřovacího kódu. Návratová hodnota je typu boolean podle toho, zda se ověřovací pin podařilo odeslat.

Ověření pomocí e-mailu je v aplikaci již implementováno. Zaslání kódu pomocí SMS je nutné doimplementovat, až bude TUL disponovat funkční SMS bránou a systémem pro správu ověřeného telefonního čísla studentů a zaměstnanců.

Omezení dvoufázového ověření

Pro zvýšení bezpečnosti při dvoufázovém ověření se ve třídě `votestorageserver.TwoStepVerificationComponent` nachází metoda `canCreateTwoStepVerification(Voter voter)`. V této metodě lze přidávat nebo upravovat omezení pro dvoufázové ověření. Struktura databáze byla navržena tak, aby uchovávala všechny předchozí pokusy o dvoufázové ověření a uchovávala jejich výsledek. Z těchto záznamů si tedy můžeme vytvořit nejrůznější pravidla pro zajištění co největší bezpečnosti voleb. Základními pravidly jsou:

Obrázek 5.6: Rozhraní webové služby Aplikace pro sčítání hlasů

- Maximální počet špatných zadání ověřovacího kódu (3x)
- Maximální počet odeslání kódu za 30 minut (5x)

5.4 Aplikace pro sčítání hlasů

Při návrhu sčítací aplikace jsem vyšel z bezpečnostních požadavků definovaných v práci [10], do kterých jsem se pokusil zakomponovat určitá zjednodušení pro pohodlné využití na TUL. Při rozhodování, zda udělat sčítací aplikaci pro nasazení na serveru plně izolovaném od internetu, jsem dospěl k závěru, že tento způsob nebude pro potřeby TUL vhodný. Pokud se budou elektronické volby a hlasování provádět častěji a vždy by bylo pro administrátora složité jejich vytváření a výsledky by musel ručně přenést na paměťovém médiu do offline sčítací aplikace, nebylo by to výhodou, ale mohly by se spíše vyskytnout nečekané problémy.

Aplikaci pro sčítání hlasů jsem tedy navrhl tak, aby byla nasazena na samostatném serveru. V aplikaci je implementována webová služba znázorněná na obrázku 5.6, ke které má přístup pouze Administrační aplikace. Další částí aplikace je webové rozhraní, pomocí kterého si mohou uživatelé zobrazit výsledky všech ukončených voleb a ověřit si správné započtení jejich volebního lístku.

5.4.1 Analýza požadavků

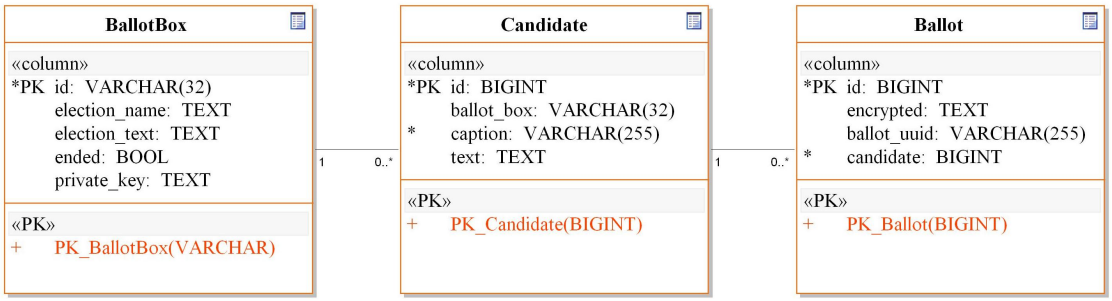
Funkční požadavky

- Generování klíčových párů pro šifrování/dešifrování volebních lístků

- Distribuce veřejného klíče aplikaci pro ukládání hlasů
- Ukládání struktury voleb a volební urny
- Dešifrování a sčítání volebních lístků
- Uživatelské rozhraní pro zobrazení výsledků voleb
- Uživatelské rozhraní pro ověření volebního lístku podle jeho UUID

5.4.2 Struktura databáze

Struktura databáze pro ukládání volebních uren je znázorněna na obrázku 5.7. Databáze je rozdělena do tří tabulek.



Obrázek 5.7: Model databáze aplikace pro sčítání hlasů

Tabulka BallotBox

Důležitým atributem tabulky BallotBox je `id`, které je generováno pomocí frameworku Hibernate na náhodný řetězec o 32 znacích. Identifikátory tedy netvoří posloupnost a jsou proto těžko odhadnutelné. V atributu `private_key` je uložen privátní klíč ve formátu Base64, který je využíván k dešifrování volebních lístků. Posledním zajímavým atributem je `ended`, který značí, zda byly volby už ukončeny a všechny volební lístky přeneseny do sčítací aplikace. Pokud je atribut `ended` nastaven na hodnotu `true`, jsou výsledky voleb zobrazeny ve webovém rozhraní aplikace a zpřístupněny všem voličům.

Tabulka Candidate

Tabulka Candidate uchovává informace o kandidátech. Ve sčítací aplikaci je navíc oproti aplikaci pro ukládání hlasů vazba 1:N mezi tabulkou `Candidate` a tabulkou `Ballot`.

Tabulka Ballot

Tabulka Ballot obsahuje atribut `encrypted`, kde je uložen zašifrovaný volební lístek, který byl přenesen z aplikace pro ukládání hlasů. Dále atribut `ballot_uuid`, který je již dešifrovaný z volebního lístku a je jeho jednoznačným identifikátorem, podle kterého si volič může ověřit, kterému kandidátovi byl volební lístek započten. Posledním důležitým je atribut `candidate`, který je cizím klíčem do tabulky `Candidate` a identifikuje zvoleného kandidáta.

5.4.3 Generování klíčového páru pro šifrování volebních hlasů

V aplikaci jsem implementoval třídu `EncryptionUtil`, nacházející se v balíku `votecountingserver.util`, ve kterém jsou implementovány všechny potřebné metody pro generování klíčů, šifrování a dešifrování, které aplikace potřebuje ke svému běhu.

Při vytváření nových voleb v administračním rozhraní nejprve administrační aplikace zavolá metodu `generateKey` implementovanou ve webové službě sčítací aplikace (obr. 5.6). Metoda vrátí ID nového hlasování a vygenerovaný veřejný klíč pro šifrování hlasů.

5.4.4 Přenos volebních lístků k sečtení

Sčítací aplikace periodicky stahuje volební urny ukončených voleb. Pro získání volebních uren je volána metoda `getBallotBox(token)` (obr. 5.4) která vrátí pole objektů typu `BallotBox`. Pokud je volební urna správně dekódována a uložena do databáze sčítací aplikace, je ukončena zavoláním metody `votesCounted(token, electionID)` (obr. 5.4). Pomocí této metody se daná volba

označí jako ukončená a sečtená a nebude se nadále zobrazovat voličům ve volební aplikaci.

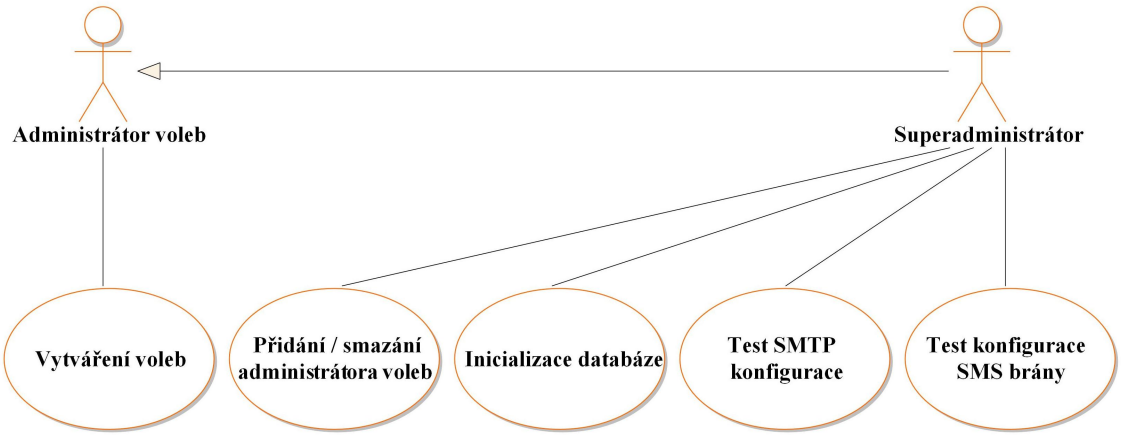
5.4.5 Ověření správného započtení volebního hlasu voličem

Ve sčítací aplikaci jsem implementoval webové rozhraní, ve kterém jsou zobrazeny výsledky voleb v přehledných grafech. Volič má možnost si ověřit správné započtení volebního lísku zadáním identifikátoru jeho volební obálky. Ukázka je na obrázku A.3.

5.5 Rozhraní pro správu systému

V administračním rozhraní jsem vytvořil dvě role uživatelů s rozdílnými oprávněními. Akce, které mohou jednotliví uživatelé provádět jsou znázorněny na obrázku diagramu užití 5.8.

V systému může být nastaveno libovolné množství uživatelů s rolí Administrátor voleb, ale pouze jeden uživatel smí mít roli Superadministrátor.



Obrázek 5.8: Diagram užití administračního rozhraní

5.5.1 Tvorba voleb a anket

Proces návrhu

Návrh nových anket jsem rozdělil do pěti sekcí (obr. A.4). Všechny editované hodnoty jsou ukládány do session a je tedy možné mezi sekcemi přecházet a vyplňovat je mimo dané pořadí.

- **Základní nastavení** - Definuje se základní struktura voleb a jejich kandidátů. Po uložení změn jsou data z formuláře odeslána do kontroleru a v něm uchovávána pro další změny. Kontroler má nastavenou životnost po dobu platnosti Session.

- **Kritéria studentů** - Zde se definují kritéria, která musí student splňovat, aby měl přístup k volbám. Kritérií může být definováno libovolné množství. Student vyhovující více kritériím je oprávněn volit pouze jednou.

  Studenty je možné filtrovat na základě následujících kritérií: *fakulta*, *forma studia*, *studijní program*, *stav studia*, *hlavní studium*, *rodné číslo*.

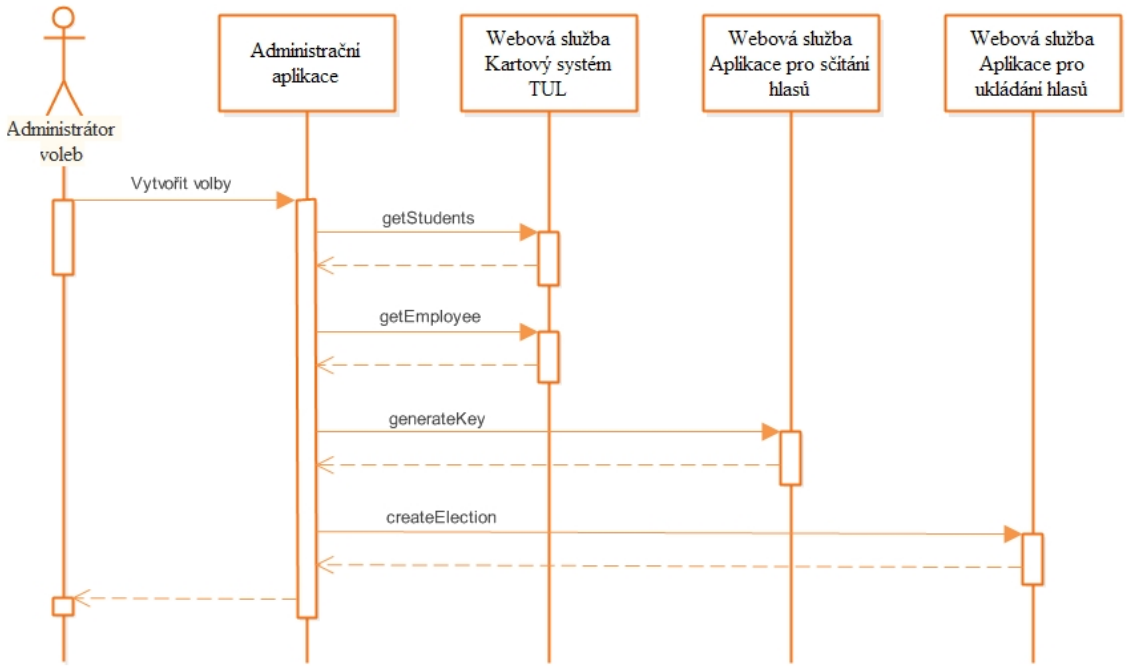
  Při zadávání kritéria, které omezuje studenty určitého studijního programu, lze využít zástupného znaku %, který zastupuje jakýkoliv řetězec. Kódy studijních programů jsou tvaru *BXXXX*, *NXXXX* a *PXXXX*, kde prefixy B, N a P rozlišují, zda jde o bakalářský, navazující nebo doktorský typ studia a XXXX je číslo identifikující daný obor. Pokud tedy chceme zadat kritérium, které bude filtrovat pouze studenty navazujících oborů, zadáme do políčka hodnotu `N%`.

- **Kritéria zaměstnanců** - Jsou stejná jako kritéria studentů pouze s jinými filtrovacími možnostmi. Zaměstnance lze filtrovat na základě těchto kritérií: *fakulta*, *pracoviště*, *procento úvazku* (je větší, je menší, je rovno), *zaměstnanci pouze s hlavním úvazkem*, *zaměstnanci jsou akademici*, *rodné číslo*.

- **Seznam voličů** - Tato sekce slouží pouze ke kontrole správné definice seznamu voličů. Zobrazí veškeré voliče, kteří budou oprávněni volit ve vytvářených volbách.
- **Souhrnné informace** - Administrátor voleb zde podrobně vidí sjednocené informace ze všech předchozích kroků a v této sekci se nachází tlačítko na finální vytvoření voleb.

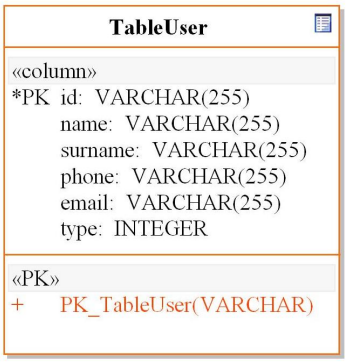
Proces vytvoření

Proces vytvoření volby a její distribuce do všech částí systému je zobrazena na obrázku 5.9 formou sekvenčního diagramu. Nejprve se z Kartového systému TUL sestaví list voličů. Následně se ze sčítacího serveru nechá vygenerovat veřejný klíč a v posledním kroku se struktura volby se seznamem voličů a veřejným klíčem přenese do Aplikace pro ukládání hlasů.



Obrázek 5.9: Sekvenční diagram vytvoření voleb

5.5.2 Struktura databáze



Obrázek 5.10: Struktura databáze Administrační aplikace

Struktura databáze administrační aplikace, znázorněna na obr. 5.10, obsahuje pouze jednu tabulku pro ukládání uživatelů, kteří mají oprávnění vytvářet volby. Primární klíč tabulky je nastaven na atribut `id`, a je do něj ukládán MD5 hash rodného čísla uživatele. Tento jednoznačný identifikátor nám totiž vrátí úspěšná autentizace pomocí ASS. Dále jsou zde uloženy dodatečné informace o uživateli jako je *jméno*, *příjmení*, *telefon*, *e-mail*.

Atribut `type` je zaveden pro použití do budoucna, pokud by bylo potřeba aplikaci rozšířit ještě o více rolí administrátorů s různými oprávněními.

5.5.3 Správa volebních administrátorů

Volební administrátory má právo vytvářet pouze uživatel s rolí Superadministrátor. Superadministrátor je definován v konfiguračním souboru který se obvykle nachází v `/etc/voting-system/administration/config.properties`. Proměnné `SUPERADMIN_IDENTIFICATOR` je nastaven MD5 hash rodného čísla vybraného superadministrátora.

5.6 Kartový systém

5.6.1 Připojení k webové službě

WSDL webové služby se nachází na adrese:

<https://cards.tul.cz:8443/WS_VOLBY/ElectionWS?wsdl>

Přístup k webové službě je omezen na základě dvojice IP adresa v rámci lokální sítě TUL a k ní platný autorizační token, který se předává jako parametr každé metodě webové služby.

5.6.2 Rozhraní webové služby

Struktura webové služby Kartového systému je znázorněna na obrázku 5.11 a níže je popsaná funkčnost jednotlivých metod.

Obrázek 5.11: Diagram komponent Elektronického volebního systému

Dostupné metody

- **getFaculties(token)** - Metoda vrací strukturu obsahující pole všech fakult TUL. Každá fakulta navíc obsahuje seznam všech pracovišť.
- **getFacultyByCode(token, code)** - Vrátí jednu fakultu podle zadaného identifikačního kódu.

- **getEmployee(token, filter, startPosition, recCount)** - Vrátí seznam zaměstnanců vyhovující předanému filtru. Pro velké množství položek se předávají proměnné `startPosition` a `recCount`, které slouží k načítání po částech.
- **getStudents(token, filter, startPosition, recCount)** - Stejná funkčnost jako metoda `getEmployee`, jen se předává jiná struktura filtru a návratovou hodnotou je pole studentů.

6 Testování aplikace

Pro testování aplikace bylo vytvořeno několik různých voleb a hlasování. Každé volby měly jiný seznam voličů, který jsem nejprve definoval pouze pro pár osob pomocí jejich rodných čísel a následně i na základě kritérií, která opravňovala k volbám všechny studenty fakulty Mechatroniky. Z nich jsem oslovil pouze pár osob k provedení volby.

Při průběhu těchto testovacích hlasování se nevyskytla žádná chyba. Veškeré lístky byly správně uloženy a po skončení voleb úspěšně přeneseny do sčítací aplikace, kde byly rozšifrovány a sečteny. Výsledky voleb byly zobrazeny ve webovém rozhraní sčítací aplikace.

Každému z voličů, který se přihlásil do volební aplikace, byly správně zobrazeny pouze volby, v kterých může volit a nebylo možné provést dvě hlasování v jedněch volbách.

Při uživatelském testování jsem se zaměřil kromě správného průběhu celého volebního procesu také na připomínky a návrhy voličů k intuitivnosti volební aplikace.

6.1 Intuitivnost volební aplikace

Průchod volební aplikací voličům nečinil problém, vyskytlo se pouze pár drobných připomínek, které jsem se pokusil sepsat v následujících kapitolách.

6.1.1 Záporné poznatky

- Při zadávání ověřovacího kódu není viditelné, kolik času zbývá do vypršení jeho platnosti.

- Na stránce s potvrzením volby a zobrazením identifikátoru volebního lístku by bylo dobré udělat tlačítko pro tisk identifikátoru volebního lístku nebo pro jeho uložení do počítače.

6.1.2 Kladné poznatky

- Panel se znázorněním jednotlivých fází volebního procesu celou volební aplikaci zpřehlední a volič se lépe orientuje.
- Pro voliče je zpříjemněno zadávání ověřovacího kódu, kurzor se automaticky posouvá mezi textovými poli.

6.2 Další testování

6.2.1 Testy zaměřené na výkon volebního systému

Před samotným zavedením elektronického volebního systému by bylo dobré aplikaci otestovat na její výkon, aby zvládla případný nápor voličů v době voleb. Takové testování jsem při své práci neprováděl, jelikož při vývoji byla aplikace nasazena na jednom virtuálním serveru. Pokud bude aplikace nasazena do ostrého provozu, měly by její jednotlivé komponenty běžet na více samostatných serverech. Díky této skutečnosti by testování na virtuálním serveru nemělo žádnou směrodatnou hodnotu.

Pro testování doby odezvy serveru při vyšším počtu souběžných požadavků je možné využít například nástroj Apache JMeter [6] vydávaný pod licencí Apache License 2.0.

7 Návrhy na vylepšení volebního systému

7.0.2 Generování studentských elektronických podpisů

Jako jedno z dalších zabezpečení by se nabízelo generování kvalifikovaného certifikátu studentům. Daný certifikát by jim byl předán buď při přijetí na školu nebo po zažádání studenta. Certifikát by sloužil k elektronickému podpisu důležité komunikace mezi studentem a TUL. Mohl by nahradit GUID generované na rozhodnutí o přijetí (viz. kapitola 4.2.1).

V elektronických volbách by se využil k podepsání volební obálky a tím pádem i zvýšení bezpečnosti.

7.0.3 Klientská aplikace jako Java applet

Klientská volební aplikace je nyní realizována jako webová aplikace. K její interpretaci je využíván webový prohlížeč na voličově počítači. Webových prohlížečů existuje velké množství a ne všechny prohlížeče podporují veškeré funkce potřebné ke správnému běhu volební aplikace (viz. kapitola 5.2.6). Mohly by tedy způsobit nefunkčnost volební aplikace.

Tento problém by byl vyřešen, pokud by byla volební aplikace realizována formou Java Appletu, který by byl stažen ze serveru a následně zobrazen na stránkách volební aplikace. Podepsáním appletu bychom navíc docílili prokázání původu aplikace a nenarušení kódu appletu při přenosu k uživateli. Tím by byla i zvýšena bezpečnost aplikace.

8 Závěr

Hlavním cílem mé diplomové práce bylo navrhnout a realizovat elektronický volební systém, který bude schopen pořádat jak běžné průzkumy mínění, tak i významné volby a to napříč různými skupinami voličů v rámci TUL. Velký důraz byl dán na nutnost zajištění jednoznačné identifikace voliče a zároveň také na zachování anonymity jím odevzdaného hlasu.

Výsledkem mé práce je systém, který zajistí kompletní servis volební procesu. Aplikace obsahuje administrační rozhraní pro jednoduché vyhlašování voleb, klientskou aplikaci pro zprostředkování hlasování a komponentu pro sčítání hlasů a zobrazení výsledků.

Volební systém je funkční a schopen realizovat bezpečné volby s dvoufázovým ověřením voliče. To je však nyní možné pouze pomocí univerzitní e-mailové adresy. Pro využití ověření pomocí SMS je nutné v rámci TUL vytvořit SMS bránu a systém správy ověřeného telefonního čísla studentů a zaměstnanců. Oba tyto systémy jsou již vyvíjeny a budou v dohledné době uvedeny do provozu. Elektronický volební systém je navržen tak, aby pro napojení na vzniklou SMS bránu stačil minimální zásah do kódu, který je popsaný v této práci.

Pro nasazení systému podle návrhu, obsaženém v mé práci, je zapotřebí vyhradit a nakonfigurovat dostačující dva, nejlépe však tři servery, na kterých aplikace poběží.

Pro potřeby volebního systému byla také navržena webová služba, pomocí které lze vyhledávat studenty či zaměstnance podle nejrůznějších kritérií. Tato webová služba má potencionální využitelnost v mnoha dalších aplikacích.

Věřím, že aplikace najde své uplatnění v portfoliu aplikací TUL a přispěje k zjednodušení volebních a hlasovacích procesů. V budoucnu by tak mohlo dojít ke

znatelnému nárůstu volební účasti a tím také k dynamickému rozvoji spokojenosti studentů a zaměstnanců TUL.

Literatura

[1] *Estonian National Electoral Committee* [online]. 2015. Dostupné z: <http://vvk.ee/voting-methods-in-estonia/>.

[2] *Helios Voting* [online]. 2015. Dostupné z: <https://vote.heliosvoting.org>.

[3] *INFORMACE O ELEKTRONICKÉM ZPŮSOBU HLASOVÁNÍ* [online]. 2012. Dostupné z: <http://www.komora.cz/download.aspx?dontparse=true&FileID=8822>.

[4] *JAX-WS* [online]. 2015. Dostupné z: <https://jax-ws.java.net/>.

[5] *DEPLOY JAXWS APPLICATION ON TOMCAT EXAMPLE* [online]. 2015. Dostupné z: <http://memorynotfound.com/deploy-jaxws-application-tomcat-example/>.

[6] *Apache JMeter* [online]. 2015. Dostupné z: <http://jmeter.apache.org>.

[7] *THE WORLD'S LARGEST WEB DEVELOPER SITE* [online]. 2015. Dostupné z: <http://www.w3schools.com/Html/html5_webstorage.asp>.

[8] *SimplyVoting* [online]. 2015. Dostupné z: <https://www.simplyvoting.com/>.

[9] *Spring Framework Reference Documentation* [online]. 2014. Dostupné z: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>.

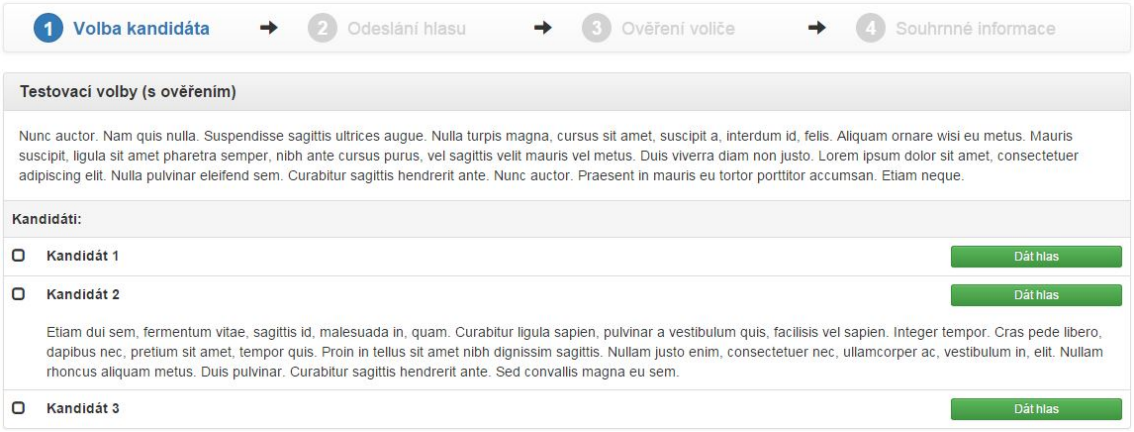
[10] BERGER, J. *Elektronický systém voleb a hlasování.* Praha : ČVUT, 2012. Dostupné z: <https://dip.felk.cvut.cz/browse/details.php?f=F3&d=K13136&y=2012&a=bergejos&t=dipl>.

[11] FIALíK, I. *Aplikace kryptografických primitiv.* Brno : Masarykova univerzita Fakulta informatiky, 2006. Dostupné z: <https://is.muni.cz/th/60488/fi_m/dp.pdf>.

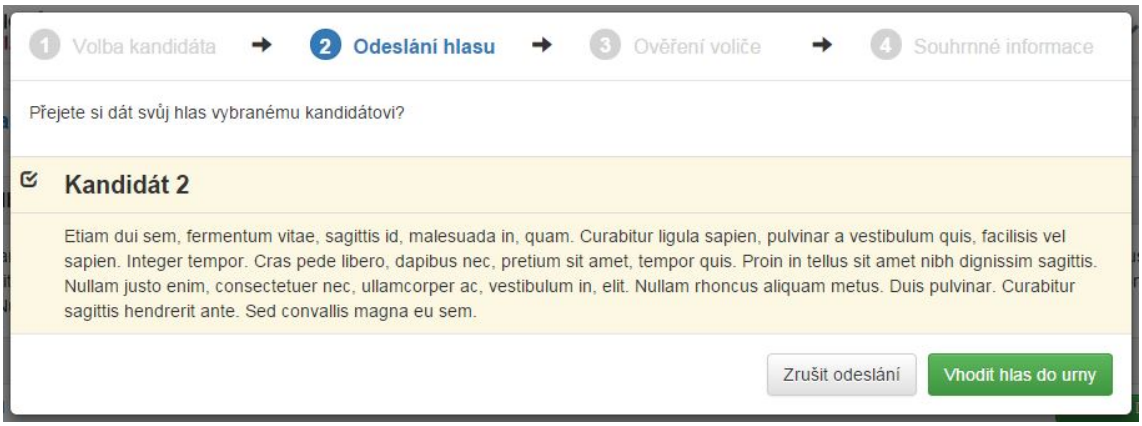
[12] MILOš BRUNCLíK, M. N. a. k. *Internetové volby: budoucnost, nebo slepá ulička demokracie?* Praha : Sociologické nakladatelství (SLON), 2014. ISBN 978-80-7419-168-8.

A Ukázka elektronického volebního systému

A.1 Volební aplikace

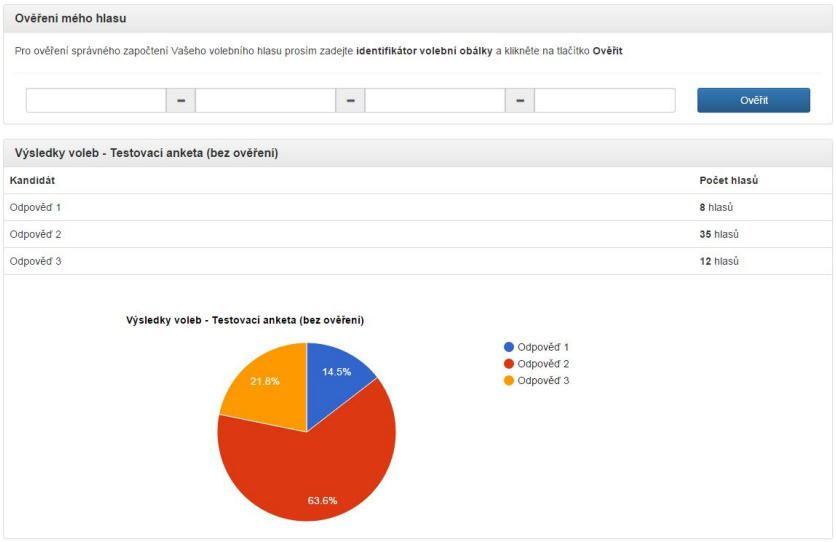


Obrázek A.1: Volba kandidáta



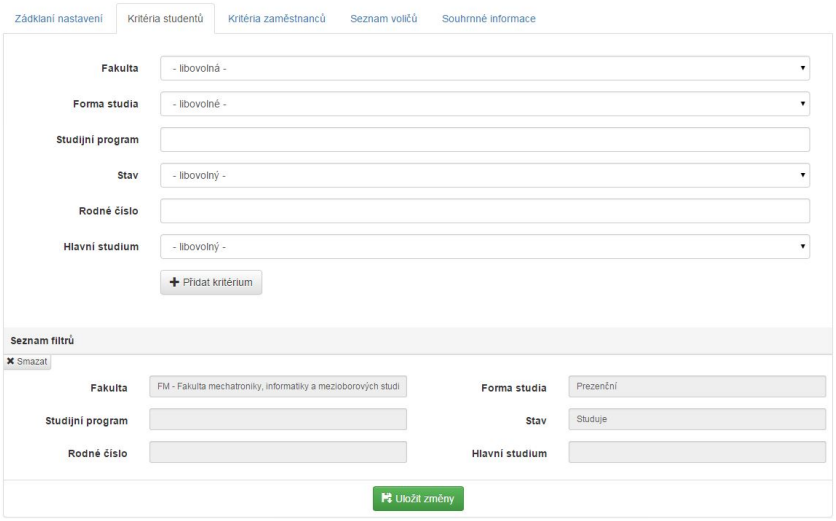
Obrázek A.2: Odeslání hlasu

A.2 Sčítací aplikace



Obrázek A.3: Výsledky voleb

A.3 Administrační rozhraní



Obrázek A.4: Definice kritérií studentů

B Konfigurační soubory aplikace

Konfigurační soubory jednotlivých komponent jsou v souborech s příponou `properties`. Jednotlivé konfigurační proměnné se zadávají ve formátu `KEY=value`. V následujících kapitolách popíši jednotlivé konfigurační proměnné a jejich možné hodnoty.

B.1 Aplikace pro ukládání hlasů

Umístění konfiguračního souboru:

`/etc/voting-system/storage/config.properties`

B.1.1 Konfigurace SMTP

- *MAIL_SMTP_HOST* - adresa SMTP serveru
- *MAIL_SMTP_PORT* - port SMTP serveru
- *MAIL_SMTP_AUTH* - zapnutí autorizace pro připojení k SMTP serveru
- *MAIL_SMTP_USER* - uživatelské jméno
- *MAIL_SMTP_PASSWORD* - heslo
- *MAIL_SMTP_TLS* - šifrování TLS
- *MAIL_SMTP_SSL* - šifrování SSL
- *MAIL_SMTP_FROM_ADDRESS* - e-mailová adresa odesílatele

B.1.2 Autorizační tokeny

Hodnota proměnných definujících autorizační tokeny může obsahovat buď cestu k souboru obsahujícího daný token nebo token přímo vložený ve formě řetězce jako hodnotu dané proměnné.

- *AUTH_TOKEN_FORWARDING* - autorizační token volební aplikace
- *AUTH_TOKEN_COUNTING* - autorizační token sčítací aplikace

- *AUTH_TOKEN_ADMIN* - autorizační token administrační aplikace
- *AUTH_TOKEN_STORAGE* - autorizační token aplikace pro ukládání hlasů

B.1.3 Dvoufázové ověření

- *TWO_STEP_VERIFICATION_METHOD* - Metoda pro dvoufázové ověření voliče (možné hodnoty: *EMAIL*, *PHONE*).
- *PIN_VALIDITY* - Platnost ověřovacího kódu pro dvoufázové ověření v sekundách (celé číslo, výchozí hodnota: 600).
- *ADMIN_EMAIL* - Kontaktní e-mailová adresa Superadministrátora. Je využita při spuštění testu správné konfigurace SMTP klienta z administračního rozhraní. Na tuto adresu se odešle testovací e-mailová zpráva.
- *ADMIN_PHONE* - Kontaktní telefonní číslo Superadministrátora. Je využita při spuštění testu správné konfigurace SMS brány z administračního rozhraní. Na tuto adresu se odešle testovací SMS zpráva.

B.2 Volební aplikace

Umístění konfiguračního souboru:

`/etc/voting-system/forwarding/config.properties`

B.2.1 Autorizační tokeny

Hodnota proměnných definujících autorizační tokeny může obsahovat buď cestu k souboru obsahujícího daný token nebo token přímo vložený ve formě řetězce jako hodnotu dané proměnné.

- *AUTH_TOKEN_FORWARDING* - autorizační token volební aplikace

B.2.2 Adresy ostatních částí systému

- *VOTE_STORAGE_SERVICE_WSDL* - URL adresa k WSDL souboru webové služby aplikace pro ukládání hlasů (viz. kapitola 5.3.2).

B.3 Aplikace pro sčítání hlasů

Umístění konfiguračního souboru:

`/etc/voting-system/counting/config.properties`

B.3.1 Autorizační tokeny

Hodnota proměnných definujících autorizační tokeny může obsahovat buď cestu k souboru obsahujícího daný token nebo token přímo vložený ve formě řetězce jako hodnotu dané proměnné.

- *AUTH_TOKEN_COUNTING* - autorizační token aplikace pro sčítání hlasů
- *AUTH_TOKEN_ADMIN* - autorizační token administrační aplikace

B.4 Administrační aplikace

Umístění konfiguračního souboru:

```
/etc/voting-system/administration/config.properties
```

B.4.1 Superadministrátor

Údaje o superadministrátoru se využívají při autorizaci do administrační aplikace a při vytváření testovacích voleb pomocí administrační aplikace se z těchto údajů vytvoří seznam voličů, který je opravňuje volit pouze superadministratora.

- *SUPERADMIN_IDENTIFICATOR* - identifikátor superadministrátora (MD5 Hash rodného čísla)
- *SUPERADMIN_EMAIL* - e-mailová adresa superadministrátora

B.4.2 Autorizační tokeny

Hodnota proměnných definujících autorizační tokeny může obsahovat buď cestu k souboru obsahujícího daný token nebo token přímo vložený ve formě řetězce jako hodnotu dané proměnné.

- *AUTH_TOKEN_ADMIN* - autorizační token administrační aplikace
- *AUTH_TOKEN_CARDS* - autorizační token Kartového systému TUL

B.4.3 Adresy ostatních částí systému

- *WSDL_CARDS* - adresa WSDL souboru Kartového systému TUL
- *WSDL_COUNTING* - adresa WSDL souboru aplikace pro čítání hlasů
- *WSDL_STORAGE* - adresa WSDL souboru aplikace pro ukládání hlasů