

Zadání

Na základě existujícího prototypu navrhnete a realizujete systém pro elektronické volby, jehož účelem bude zjednodušení a zrychlení stávajících volebních procesů a kompletní nahrazení „obálkového“ hlasování. Systém bude umožňovat hlasování za pomoci dotykových terminálů. Před začátkem hlasování voliči zaregistrují svůj otisk prstu do systému a obdrží číselný PIN. Systém zajistí tajnost a nezfalšovatelnost hlasování. Po ukončení hlasování systém spočítá hlasy a prezentuje výsledky.

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

Elektronické volby

Tomáš Čerevka

Vedoucí práce: Ing. Martin Komárek

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Softwarové inženýrství

25. května 2011

Poděkování

Především bych rád poděkoval své rodině a přátelům, kteří mě podporovali během celého studia a dodávali mi kuráž tam, kde já sám jsem již propadal beznaději.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon). Veškeré nakládání s touto prací je omezeno pravidly stanovenými níže, která jsou označována jako „Nová BSD licence“.

Nová BSD licence

Copyright © 2011 Tomáš Čerevka Všechna práva vyhrazena.

Redistribuce a použití zdrojových i binárních forem díla, v původním i upravovaném tvaru, jsou povoleny za následujících podmínek:

- Šířený zdrojový kód musí obsahovat výše uvedenou informaci o copyrightu, tento seznam podmínek a níže uvedené zřeknutí se odpovědnosti.
- Šířený binární tvar musí nést výše uvedenou informaci o copyrightu, tento seznam podmínek a níže uvedené zřeknutí se odpovědnosti ve své dokumentaci a/nebo dalších poskytovaných materiálech.
- Ani jméno vlastníka práv, ani jména přispěvatelů nemohou být použita při podpoře nebo právních aktech souvisejících s produkty odvozenými z tohoto software bez výslovného písemného povolení.

TENTO SOFTWARE JE POSKYTOVÁN DRŽITELEM LICENCE A JEHO PŘÍSPĚVATELI „JAK STOJÍ A LEŽÍ“ A JAKÉKOLIV VÝSLOVNÉ NEBO PŘEDPOKLÁDANÉ ZÁRUKY VČETNĚ, ALE NEJEN, PŘEDPOKLÁDANÝCH OBCHODNÍCH ZÁRUK A ZÁRUKY VHODNOSTI PRO JAKÝKOLIV ÚČEL JSOU POPŘENY. DRŽITEL, ANI PŘÍSPĚVATELÉ NEBUDOU V ŽÁDNÉM PŘÍPADĚ ODPOVĚDNI ZA JAKÉKOLIV PŘÍMÉ, NEPŘÍMÉ, NÁHODNÉ, ZVLÁŠTNÍ, PŘÍKLADNÉ NEBO VYPLÝVAJÍCÍ ŠKODY (VČETNĚ, ALE NEJEN, ŠKOD VZNIKLYCH NARUŠENÍM DODÁVEK ZBOŽÍ NEBO SLUŽEB; ZTRÁTOU POUŽITELNOSTI, DAT NEBO ZISKŮ; NEBO PŘERUŠENÍM OBCHODNÍ ČINNOSTI) JAKKOLIV ZPŮSOBENÉ NA ZÁKLADĚ JAKÉKOLIV TEORIE O ZODPOVĚDNOSTI, AŽ UŽ PLYNOUCÍ Z JINÉHO SMLUVNÍHO VZTAHU, URČITÉ ZODPOVĚDNOSTI NEBO PŘEČINU (VČETNĚ NEDBALOSTI) NA JAKÉMKOLIV ZPŮSOBU POUŽITÍ TOHOTO SOFTWARE, I V PŘÍPADĚ, ŽE DRŽITEL PRÁV BYL UPOZORNĚN NA MOŽNOST TAKOVÝCH ŠKOD.

V Praze dne 26.5.2011

.....

Abstract

This work deals with the concept and implementation of electronic election system which uses local voting terminals. The system provides easy and quick deployment, secure voting and post-processing results. It also allows the presentation of election results. Main emphasis are security and user-friendly enviroment.

Abstrakt

Tato práce se zabývá návrhem a implementací systému pro správu elektronických voleb za využití lokálních hlasovacích terminálů. Systém zajišťuje snadné a rychlé uvedení do provozu, bezpečné hlasování a následné zpracování výsledků. Zároveň umožňuje prezentaci výsledků voleb. Důraz je kladen na zabezpečení a uživatelskou přívětivost.

Obsah

1	Úvod	1
2	Popis problému, specifikace cíle	3
2.1	Motivace	3
2.2	Rešerše	3
2.2.1	Prototyp systému eVolby	3
2.2.2	Experimentální ověření distribuovaného volebního schématu	4
2.2.3	Hlasovací systém pro samosprávné orgány vysokých škol	5
2.2.4	Mobilního hlasovacího zařízení	5
2.2.5	Rešerše algoritmů pro snímání a zpracování otisku prstů	5
2.2.6	Electronic Voting in Europe - Technology, Law, Politics and Society	6
2.2.7	Elektronické volby u nás a ve světě	6
2.3	Požadavky	6
2.3.1	Obecné požadavky	6
2.3.1.1	Společné požadavky	6
2.3.1.2	Server	7
2.3.1.3	Terminálový klient	7
2.3.2	Funkční požadavky	7
3	Analýza	9
3.1	Fáze elektronických voleb	9
3.1.1	Registrace voličů	9
3.1.1.1	Seznamy voličů	9
3.1.2	Založení voleb	10
3.1.3	Nominace	10
3.1.4	Hlasování	10
3.1.5	Výsledky	10
3.1.6	Společné schvalování	10
3.2	Bezpečnostní rizika	11
3.2.1	Hrozby	11
3.2.2	Zabezpečení	11
3.3	Uživatelské role	11
3.3.1	Administrátor	11
3.3.2	Komisař	12
3.3.3	Volič	12

3.3.4	Kandidát	12
3.4	Případy užití	12
3.5	Scénáře	12
3.5.1	Založení voleb	12
3.5.2	Přiřazení komisaře k volbám	14
3.5.3	Odstranění komisaře z voleb	14
3.5.4	Založení volební události	15
3.5.5	Přiřazení voliče k volební události	15
3.5.6	Odstranění voliče z volební události	15
3.5.7	Zahájení nominování	16
3.5.8	Ukončení nominování	16
3.5.9	Zahájení hlasování	17
3.5.10	Ukončení hlasování	17
3.5.11	Nominování kandidáta	17
3.5.12	Zamítnutí nominace	18
3.5.13	Hlasování	18
3.6	Technologie	18
3.6.1	Server	18
3.6.1.1	Java EE	18
3.6.1.2	Aplikační server	19
3.6.1.3	Databáze	19
3.6.2	Klient	19
3.6.2.1	Java SE	19
3.6.3	Vývojové a testovací prostředí	19
3.7	Snímač otisků prstů	19
3.7.1	Griaule Biometrics	20
3.7.2	Futronic	20
3.7.3	Digital Persona	20
3.7.4	Vybrané SKD a snímač otisků prstů	20
4	Návrh řešení a realizace	21
4.1	První iterace	21
4.1.1	Cíl	21
4.1.2	Postup	21
4.1.2.1	Sdílené knihovny	21
4.1.2.2	Entity	22
4.1.2.3	Databáze	22
4.1.3	Výstup	22
4.2	Druhá iterace	22
4.2.1	Cíl	22
4.2.2	Postup	22
4.2.2.1	Synchronizace persistentní vrstvy	22
4.2.2.2	SQL dotazy	23
4.2.2.3	Zabezpečení	23
4.2.2.4	JSF 2.0	23
4.2.2.5	Volební applet	23

4.2.3	Výstup	24
4.3	Třetí iterace	24
4.3.1	Cíl	24
4.3.2	Postup	24
4.3.2.1	Grafické úpravy	24
4.3.2.2	Odebrání komisaře	24
4.3.2.3	Multijazyčnost	24
4.3.2.4	Datový model	24
4.3.2.5	Role kandidáta	25
4.3.2.6	Terminálový klient	25
4.3.3	Výstup	25
5	Testování	27
5.1	Selenium	27
5.2	Apache Benchmark	27
6	Závěr	29
A	Seznam použitých zkratk	33
B	Instalační a uživatelská příručka	35
B.1	Potřebné programy	35
B.2	Instalace	35
B.2.1	Glassfish Server Open Source Edition 3.1	35
B.2.2	MySQL Server	36
B.2.3	Netbeans 7.0	36
B.3	Nastavení	36
B.3.1	MySQL Server	36
B.3.2	Glassfish Server Open Source Edition 3.1	36
B.3.2.1	Fronty	37
B.3.2.2	E-mail	37
B.3.2.3	Security Realm	37
B.4	Spuštění	38
C	Obsah přiloženého CD	39

Seznam obrázků

2.1	Diagram aktivit voleb bez elektronického systému.	4
3.1	Diagram případů užití.	13
4.1	Aktuální datový model v projektu.	25

Seznam tabulek

3.1	Tabulka rizik a opatření	12
5.1	Výsledky zátěžového testování pomocí Apache Benchmark	28

Kapitola 1

Úvod

Na schůzích a zasedáních dochází často k rozhodování mezi několika možnostmi. Může se jednat o zvolení kandidáta do funkce, nebo se může například hlasovat o nepersonálních otázkách¹. Jelikož ne všichni jsou vždy stejného názoru a je třeba rozhodnout kolektivně, přistupuje se k hlasování. Hlasování může být veřejné, tj. ví se, jak kdo hlasoval, nebo tajné, kdy je každý hlas anonymní. V obou případech je ovšem třeba zajistit, aby mohly hlasovat jen k tomu oprávněné osoby.

Tématika elektronických voleb je poslední dobou stále častěji zmiňována v souvislosti s rozvojem moderních technologií a internetu. Modernizace stávajícího volebního systému je základním pilířem tzv. systému elektronické vlády, e-governmentu. Přínosy elektronického hlasování jsou podrobně popsány v kapitole 2.1, ale ruku v ruce s nimi jde i řada technických a organizačních otázek, které je třeba řešit a jejichž základní nástin nabízí kapitola 3.2.

Cílem této práce je jejich zodpovězení a vytvoření funkčního systému, který umožní realizovat elektronické volby za pomoci dotykových terminálů ve volební místnosti. Centrální server bude od terminálů přijímat jednotlivé hlasy a na konci hlasování prezentuje výsledky.

¹ Například využití peněz z bytového fondu, kudy povede nová dálnice, jak vyřešit stávku etc.

Kapitola 2

Popis problému, specifikace cíle

2.1 Motivace

V současné době probíhá většina voleb bez jakéhokoliv elektronického systému. Při veřejném hlasování voliči pouze zvednou ruku a pověřená osoba je spočítá. Tajné volby bývají realizovány za pomoci volební urny a obálek. Na začátku se každý volič prokáže osobním dokladem, obdrží orazítkovanou volební obálku s volebními lístky a vydá se za plentu, kde z volebních lístků jeden vybere, vloží ho do obálky a tu následně vhodí do urny. Když takto odhlasují všichni, pověřená osoba urnu rozpečetí a ručně sečte všechny hlasy. Na obrázku 2.1 je diagram aktivit znázorňující průběh voleb bez elektronického systému.

Tento způsob voleb je sice funkční a léty prověřený, ovšem jeho realizace je časově a personálně náročná na přípravu, na samotný průběh voleb a následné sčítání hlasů. Jeho slabinou je možnost selhání lidského faktoru, na kterém je tento proces příliš závislý. To by mohlo zapříčinit jak neúmyslné chyby, tak zcela cílevědomé ovlivnění výsledků.

2.2 Rešerše

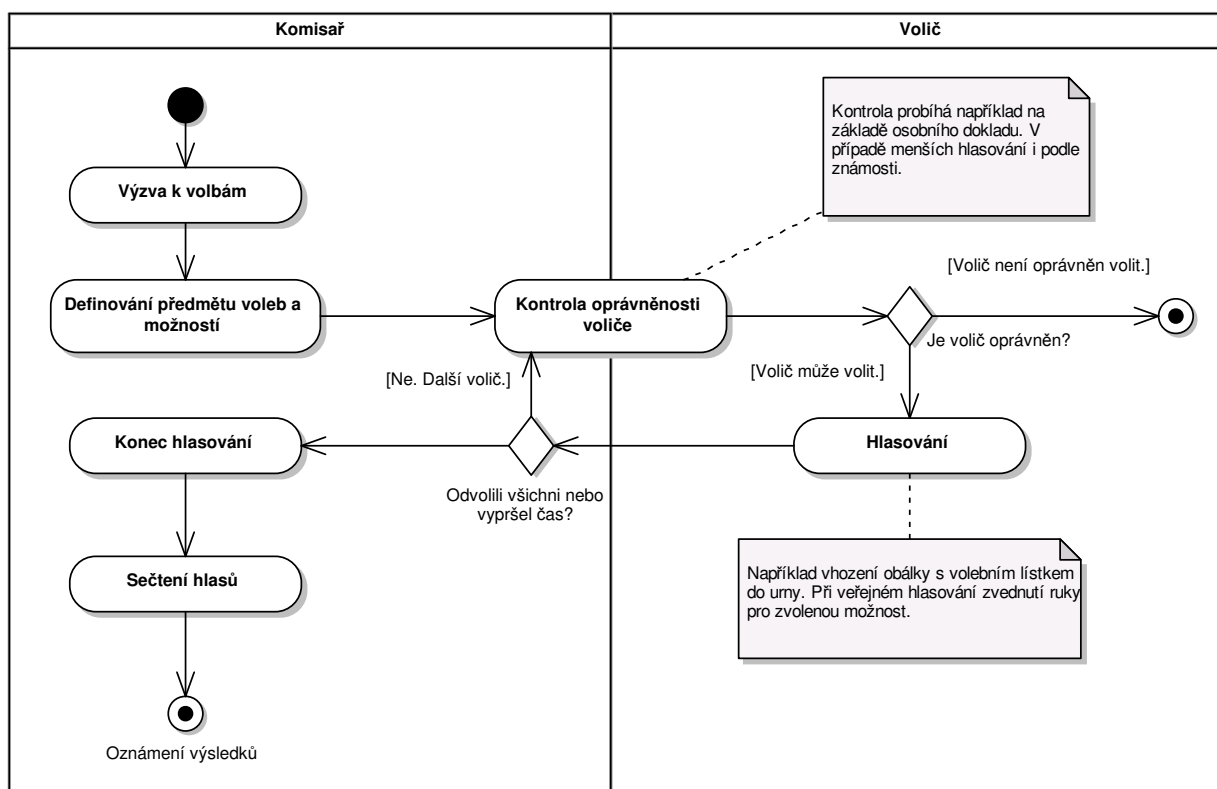
Elektronické volby jsou tématem, které již bylo mnohokrát zpracováváno. Následující odstavce rozebírají několik zdrojů, ze kterých tato práce vychází, a stručně popisují jejich obsah.

2.2.1 Prototyp systému eVolby

Implementační část této bakalářské práce vychází z prototypu systému internetových elektronických voleb pro Akademický senát ČVUT, který vznikl v rámci předmětů Y36SI2 a Y36SI3 [7]. Tento systém se sestává pouze ze serveru napsaném v Enterprise Javě, který zajišťuje jak administraci voleb, tak samotné hlasování voličů přes internet.

Prototyp je koncipován pouze pro personální volby, kdy se nejprve z řad voličů nominují kandidáti. Po ukončení nominací následuje hlasování, kdy voliči dávají své hlasy jednotlivým kandidátům.

V systému figurují tři role uživatelů: administrátor, komisař a volič. Administrátor je v systému pouze jeden a má pravomoc založit volby a jmenovat jim komisaře. Jedny volby



Obrázek 2.1: Diagram aktivit voleb bez elektronického systému.

mohou mít více komisařů a v případě důležitého rozhodnutí je třeba potvrzení alespoň nadpoloviční většiny. Komisaři pod volbami zakládají volební události představující jednotlivá kola voleb. K těm přiřazují oprávněné voliče a schvalují či vyřazují nominované kandidáty.

2.2.2 Experimentální ověření distribuovaného volebního schématu

Konceptuální návrh podobného systému zpracoval ve své disertační práci *Experimentální ověření distribuovaného volebního schématu* Radek Šilhavý [4]. Tato práce se zabývá hlavně teoretickou rovinou a popisu procesů a požadavků na realizaci elektronických volebních systémů a dává k dispozici ucelený pohled na problematiku. Důraz je kladen hlavně na internetové volební systémy umožňující vzdálené hlasování bez nutnosti fyzické účasti ve volební místnosti.

V úvodu se autor zabývá teoretických rámcem od legislativní problematiky v prostředí České republiky až po praktické využití ve světě. Popisuje, v jaké podobě byly elektronické volby testovány a nasazovány ve Spojených státech amerických, Německu, Velké Británii, Španělsku, Švýcarsku a Estonsku.

Zbytek práce se již orientuje více prakticky na návrh konceptuálního řešení distribuovaného volebního schématu a stanovuje hypotézy pro experimentální ověření. První hypotéza předpokládá, že elektronizace volebního procesu za využití internetového volebního systému je možná a že je přínosem pro voliče. Druhá hypotéza prohlašuje využití webové aplikace

jako základního řešení za výhodné, protože každému uživateli je přístupná a dostupná v čase a místě jemu vyhovujícím.

2.2.3 Hlasovací systém pro samosprávné orgány vysokých škol

Problematickou elektronických voleb se zabývá i diplomová práce Michala Petříka *Hlasovací systém pro samosprávné orgány vysokých škol* [2]. Zde se autor zaměřuje hlavně na potřeby samosprávných orgánů vysokých škol, speciálně Akademického senátu FIT ČVUT.

Tento systém je navržen tak, aby dokázal zautomatizovat online hlasování, správu jednotlivých hlasování v rámci zasedání, generování usnesení ze zasedání a prezentaci vybraných informací na webové stránky. Je zde propracován systém oprávnění, který umožňuje přidělovat libovolná administrátorská práva uživateli nebo skupině uživatelů. Autentizace uživatelů je velmi rozsáhlá, neboť poskytuje několik různých cest: přes uživatelské účty v databázi LDAP, pomocí identifikační karty či samotným systémem.

2.2.4 Mobilního hlasovacího zařízení

Hlavním účelem elektronických voleb je snaha o zpříjemnění a zrychlení práce se samotnými hlasy. Proto je také důležité zpracování terminálového klienta, které si za vzor beru *Mobilní hlasovací zařízení* [5] Jakuba Valenty. Autor se sice zabývá problematikou hlasovacího zařízení v podobě mobilního telefonu s technologií Wi-Fi, čímž se tato práce nezabývá, ale podobnost s terminálovým klientem z hlediska analýzy je značná. Kromě toho rozebírá síťovou komunikaci posíláním zpráv v jazyce XML přes protokol HTTP.

Na implementaci je zajímavá vícevrstvá architektura dělená na prezentační, aplikační, datovou, persistentní a síťovou vrstvu. Serverová část v sobě také zahrnuje modul pro generování výsledků, jehož studium v sobě skrývá potenciál využitelný při zpracovávání hlasů a prezentaci výsledků.

2.2.5 Rešerše algoritmů pro snímání a zpracování otisku prstů

Při volbách se voliči budou přihlašovat za pomoci otisku prstu a PINu. Pro tyto účely je třeba zvolit správnou čtečku otisků prstů, která zajistí spolehlivost, dlouhou životnost a cenovou dostupnost. *Rešerše algoritmů pro snímání a zpracování otisku prstů* [9] od Zuzany Poláckové poskytuje podrobný náhled do oblasti daktyloskopie. Kromě jiného pojednává o základních biometrických systémech, zpracování otisků prstů a o algoritmech používaných v praxi. Nejdůležitější je pro mě kapitola o snímačích otisků prstů, kde porovnává výhody a nevýhody optoelektrických, kapacitních, teplotních, elektroluminiscenčních a ultrazvukových snímačů.

Jako nejlepší volba se jeví elektroluminiscenční snímač, jelikož pořizuje kvalitní výstup v digitální podobě, nemá problémy se snímáním extrémně suché pokožky a je cenově dostupný. Mezi nevýhody sice patří menší odolnost a citlivost na prach a vodu, ale oproti ostatním snímačům, které trpí závažnějšími neduhy, je to zanedbatelné.

2.2.6 Electronic Voting in Europe - Technology, Law, Politics and Society

Ve dnech 7. a 9. července 2004 se konal první mezinárodní workshop *Electronic Voting in Europe - Technology, Law, Politics and Society* [1] v Schloß Hofenenu v Rakousku. Z tohoto workshopu vznikla stejnojmenná kniha v podobě sborníku prací. Kniha rozebírá problematiku elektronických voleb ze všech možných úhlů a představuje systémy používané v Evropě. Dotýká se témat bezpečnosti, hrožících rizik a jejich řešení, biometrických systémů, technických požadavků a legislativních principů.

2.2.7 Elektronické volby u nás a ve světě

Na portále www.internetprovsechny.cz vyšel rozsáhlý článek [8] pojednávající o elektronických volbách v Estonsku. Estonsko je pokládáno za internetově velmi vyspělou zemi. Už jen skutečnost, že jejich ústava zaručuje od roku 2000 všem jejich občanům přístup na internet je toho dostatečným důkazem.

Důležitou skutečností, která výrazně pomohla zavedení elektronických voleb, bylo zavedení elektronických identifikačních průkazů¹. ID karta obsahuje nejen strojově čitelný kód, ale i mikročip, který má na sobě uloženy dva digitální certifikáty a související privátní klíče chráněné PIN kódy. Jeden certifikát slouží k autentizaci, druhý k elektrickému podpisu, přičemž je každý může používat bez omezení. Nejsou tedy vázány jen na komunikaci se státem.

Zajímavostí modelu hlasování je zabezpečení hlasů. Funguje na tzv. principu dvou obálek. Hlasovací lístek se zašifruje klíčem voleb, což tvoří vnitřní obálku. Vnější obálku představuje elektronický podpis vnitřní obálky. Díky tomuto systému je možné, aby hlas nebyl spojen s voličem, ale zároveň volič může hlasovat vícekrát, kdy se započítá jen poslední hlas. Tento koncept byl zvolen kvůli eliminaci hlasování pod tlakem. U klasických voleb přistupuje volič za plentu samostatně, ovšem u voleb přes internet nelze zajistit jeho soukromí. Kdyby někdo voliče nutil hlasovat jinak, než on sám chce, může tak učinit, ale pak svůj hlas může změnit.

Portál www.lupa.cz [6] uveřejnil článek, který se více orientuje na situaci v tuzemsku. Nejdříve ovšem popisuje výhody a nevýhody elektronických voleb a zkušenosti v zahraničí, načež informuje o tom, že ČSÚ je již přepraven na jejich modernizaci [10].

2.3 Požadavky

2.3.1 Obecné požadavky

V této sekci jsou stanoveny požadavky, které se netýkají funkčnosti, nýbrž jsou kladeny na celkový návrh, použité technologie a zpracování. Požadavky jsou děleny zvlášť na serverovou a klientskou část.

2.3.1.1 Společné požadavky

Na celkový návrh řešení a dokumentaci jsou kladeny následující požadavky:

¹Obdoba našich občanských průkazů.

1. Systém bude rozdělen na serverovou a klientskou část.
2. Instalace systému a jeho uvedení do provozu bude co nejjednodušší, aby ji zvládl i zaškolený běžný uživatel počítače.
3. Systém bude multiplatformní díky prostředí JVM.
4. Systém bude používat open license nebo freeware technologie.
5. Systém zajistí bezpečnost hlasování, tj. nefalšovatelnost hlasů a tajnost voleb vůči druhé i třetí osobě.
6. Součástí práce bude podrobná dokumentace.

2.3.1.2 Server

Serverová část aplikace řídí celý průběh voleb, zpracovává požadavky od klientů a poskytuje jim data. Server:

1. zajistí práci s databází a validaci dat.
2. definuje rozhraní klientským aplikacím.
3. bude využívat technologii Java EE.

2.3.1.3 Terminálový klient

Samostatná aplikace běžící na dotykovém terminálu, která umožňuje hlasování. Vyžaduje připojení k serverové části aplikace. Klient:

1. umožní voličům hlasovat.
2. umožní voličům registraci.
3. si neukládá žádná data. Vše předává přes protokol HTTPS serverové části.
4. bude používat technologii Java SE.

2.3.2 Funkční požadavky

Tato sekce se zabývá požadavky na funkčnost systému. Systém umožní:

1. založení voleb.
2. založení volební události.
3. přiřazení komisaře k volbám.
4. registraci voliče.
5. přiřazení voličů k volební události.

6. nominování kandidátů.
7. hlasování.
8. prezentaci výsledků.
9. hromadné vložení voličů.

Kapitola 3

Analýza

3.1 Fáze elektronických voleb

Životní cyklus voleb je neměnný a čítá několik fází, které je třeba důkladně ošetřit. Pro zvýšení efektivity práce se systémem je třeba průběh každé fáze co nejvíce uživatelsky zjednodušit.

3.1.1 Registrace voličů

Před zahájením voleb je třeba do systému zanést seznam voličů, kteří jsou oprávněni se zaregistrovat. Díky tomuto opatření se minimalizuje riziko, že se do voleb zaregistruje neoprávněná osoba a zároveň bude evidováno, kteří voliči se k registraci ještě nedostavili. Tato skutečnost může být důležitá pro volby, kdy je třeba přítomnost pevně dané části voličů, aby volby byly usnášeníschopné.

Na průběh registrace voličů bude dohlížet pověřená osoba, která bude voliče připouštět k registraci po kontrole dokladu totožnosti, přičemž vždy na příslušném registračním terminálu vyvolá registraci voliče z předdefinovaného seznamu.

Registrace bude obnášet sejmутí otisku prstu a vygenerování přístupového PINu. Otisk prstu zabráni případnému hlasování v zastoupení, kdy by volič sdělil svůj PIN neoprávněné osobě. PIN slouží jako druhotný autentizační prvek, který vyloučí případnou chybu čtečky otisků prstů.

3.1.1.1 Seznamy voličů

Dá se předpokládat, že systém bude používán na více než jedny volby, přičemž seznam voličů se mezi jednotlivými volbami bude lišit jen minimálně, ne-li vůbec. Správce voleb si tedy může jednotlivé voliče zařadit do seznamu, přičemž jeden volič může být obsažen ve více seznamech, ovšem nemusí být v žádném. Při zakládání hlasování pak může voliče přidávat jednotlivě nebo hromadně vložit celé seznamy voličů bez ohledu na to, zda již jsou v systému zaregistrováni, nebo jejich předpřipravené účty zatím na registraci čekají.

3.1.2 Založení voleb

Když jsou v systému připraveni uživatelé, ať již zaregistrovaní či nezaregistrovaní, musí se založit volby. To učiní administrátor a deleguje volbám komisaře z databáze uživatelů. Komisař může být zároveň volič, ale nemusí jím být. Systém předpokládá, že jedny volby mohou mít více kol. Proto komisař pod volbami založí volební událost, ke které přiřadí oprávněné voliče¹ a nastaví detaily voleb. Jedná-li se o volby personální, následují nominace popsané v kapitole 3.1.3. U voleb nepersonálních je tato fáze vynechána a komisaři nadefinují možnosti hlasování explicitně. Poté zahájí hlasování. Pokud je k daným volbám přiřazeno více komisařů, je pro zahájení hlasování třeba potvrzení alespoň nadpoloviční většiny z nich.

3.1.3 Nominace

Nominace se týkají pouze voleb personálních a jsou zahajovány alespoň nadpoloviční většinou komisařů. V této fázi se může volič nominovat a stát se tak zároveň kandidátem. Součástí kandidatury je kandidátní listina s volebním programem a fotografií kandidáta. Kandidát má možnost kandidátní listinu upravovat nebo svou kandidaturu zrušit, dokud nejsou nominace ukončeny. Komisaři mají právo kandidáta do systému zadat ručně nebo kandidaturu zrušit. Pro ukončení nominací je třeba souhlas alespoň nadpoloviční většiny komisařů.

3.1.4 Hlasování

K hlasování voliči používají terminálového klienta, za jeho pomoci se identifikují otiskem prstu a následně se autentizují svým PINem. Systém si nechá potvrdit voličovu totožnost a zobrazí mu předmět voleb s možnostmi, pro které může hlasovat. Volič odhlasuje a jeho volební lístek se odešle ke zpracování na server.

3.1.5 Výsledky

Hlasování ukončuje komisař, v případě více komisařů nadpoloviční většina z nich. Celé volby uzavírá administrátor, pokud jsou všechna hlasování v nich ukončena. Systém zpracuje hlasy a vyhodnotí výsledky.

3.1.6 Společné schvalování

Některé úkoly jsou v životním cyklu volební události přelomové a nevratné. Konkrétně se jedná o zahájení a ukončení nominací a zahájení a ukončení hlasování. Proto je třeba k jejich uskutečnění souhlas nadpoloviční většiny komisařů, kteří danou volební událost mají na starosti.

¹Viz. kapitola 3.1.1.1

3.2 Bezpečnostní rizika

Jak již bylo zmíněno v kapitole 2.1, volby jsou provázeny mnoha faktory, které mohou nepříznivě ovlivnit jejich výsledky. Elektronická verze hlasování potlačuje lidský faktor a minimalizuje tak jeho negativní dopad, ovšem objevují se nové hrozby, jejichž dopad by mohl být mnohem horší. Těm se ale můžeme efektivně bránit.

3.2.1 Hrozby

Hrozby[1] se dají obecně rozdělit do dvou kategorií:

- manipulační
- sabotující

Ke zmanipulování voleb může dojít ze strany komisaře, který využije svých práv v systému a upraví výsledky, nebo ze strany voliče, který se pokusí hlasovat vícekrát, případně hlasovat cizím jménem. Výsledky mohou také být ovlivněny hackerským útokem modifikujícím hlasy. Všechny tyto snahy o manipulaci s daty se snaží nevzbudit žádné podezření a zůstat tak neodhalené.

Na druhé straně jsou hrozby, které mohou zabránit průběhu hlasování. Ty se dají dále dělit na úmyslné a neúmyslné. Mezi neúmyslné může například patřit přerušení dodávky elektřiny, či selhání hlasovacího systému vinou neodhalené chyby. Naopak mezi úmyslné hrozby patří DDoS útok [3] na server, úmyslné přerušení dodávky elektřiny či neoprávněná manipulace s hardwarem a softwarem. Výčet rizik a jejich opatření je uveden v tabulce 3.1.

3.2.2 Zabezpečení

U voleb musí být zajištěna nezfalšovatelnost hlasů, proto je třeba, aby všechny volební terminály komunikovaly se serverem pomocí protokolu HTTPS a veškerá komunikace mezi nimi byla zašifrovaná. Dále je třeba zajistit, aby volba každého voliče byla tajná. Toho se docílí neevიდování vazby mezi hlasem a voličem. Jakmile tedy volič odhlasuje, přičte se ke zvolené možnosti hlas, systém si uloží, že tento volič již hlasoval, a znemožní mu další hlasování v daném kole voleb. Těmito postupy je možné zabránit rizikům odcizení citlivých údajů praktikami vyjmenovanými v kapitole 3.2.1.

3.3 Uživatelské role

Volebního procesu se účastní mnoho osob, které se dělí do rolí, které při volbách představují. Pro snazší orientaci uvádím jejich výčet.

3.3.1 Administrátor

Úkolem administrátora je zakládání a ukončování voleb a jmenování komisařů. Na samotný průběh voleb nemá žádný vliv.

Riziko	Opatření
Hackerský útok	Uzavřená datová síť.
Výpadek systému	Zálohovaná data a použití transakčního zpracování.
Hardwarová závada	Připravený záložní hardware.
Vynucené hlasování	Voliči hlasují samostatně za plentou.
Únik informací o hlasování	Veškerá data jsou přenášena po HTTPS protokolu.
Podvržení hlasu	Veškerá data jsou přenášena po HTTPS protokolu.
Ztráta hlasu	Server potvrzuje přijetí.

Tabulka 3.1: Tabulka rizik a opatření

3.3.2 Komisař

Komisař je jmenován administrátorem a je přiřazen k volbám. Komisařů pro jedny volby může být více, pak v důležitých momentech voleb rozhoduje nadpoloviční většina z nich (viz. kapitola 3.1.6). Komisař zakládá v rámci voleb jednotlivé volební události, definuje jejich nastavení a přiřazuje k nim oprávněné voliče. Jeho úkolem je zahajování a ukončování nominací a hlasování.

3.3.3 Volič

Volič modifikuje a odevzdává volební lístek ve volbách, ve kterých je oprávněn hlasovat. Také se může nominovat na kandidáta.

3.3.4 Kandidát

Kandidát je nominovaný volič a může být zvolen. Má možnost upravovat svou kandidátní listinu a stáhnout kandidátku, dokud jsou volby ve fázi nominací.

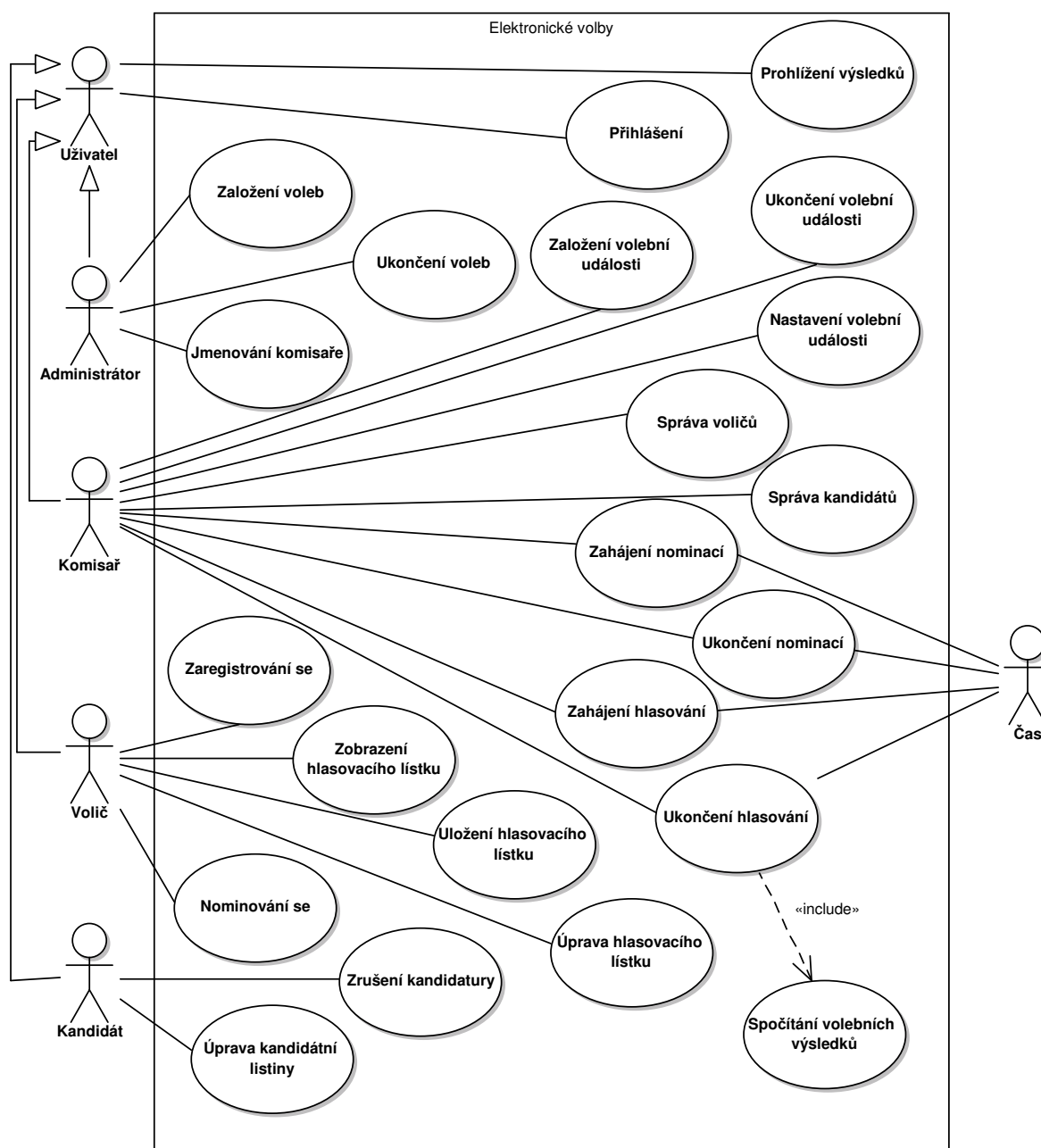
3.4 Případy užití

Případy užití jsou částečně popsány v předchozích kapitolách. Pro úplnost je na obrázku 3.1 znázorněn diagram případů užití. Oproti předchozím kapitolám je na něm zachycena i systémová role „Čas“. Aktér Čas je abstraktní podobou toku času a znázorňuje možnost automatického spouštění daných činností v nadefinovanou dobu.

3.5 Scénáře

3.5.1 Založení voleb

- Aktéři: Administrátor
- Vstup: -



Obrázek 3.1: Diagram případů užití.

- Výstup: Založené volby.
- Kroky:
 1. Uživatel požádá o vytvoření nových voleb.
 2. Systém zobrazí formulář pro vytvoření nových voleb.
 3. Uživatel vyplní formulář (název a typ voleb).

4. Uživatel odešle formulář.
5. Systém provede validaci, že je vyplněn název voleb.
6. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
7. Systém založí volby.

3.5.2 Přiřazení komisaře k volbám

- Aktéři: Administrátor
- Vstup: Volby.
- Výstup: Volby s přiřazenými komisaři.
- Kroky:
 1. Uživatel požádá o zobrazení detailu voleb.
 2. Systém zobrazí detail voleb.
 3. Uživatel zvolí z menu přidání komisaře.
 4. Systém zobrazí formulář se seznamem komisařů.
 5. Uživatel vybere komisaře a odešle formulář.
 6. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 4.
 7. Systém přiřadí komisaře k volbám.

3.5.3 Odstranění komisaře z voleb

- Aktéři: Administrátor
- Vstup: Volby s přiřazeným komisařem.
- Výstup: Volby s odebraným komisařem.
- Kroky:
 1. Uživatel požádá o zobrazení detailu voleb.
 2. Systém zobrazí detail voleb.
 3. Uživatel zvolí odstranění komisaře ze seznamu přiřazených komisařů.
 4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
 5. Systém odstraní komisaře z voleb.

3.5.4 Založení volební události

- Aktéři: Komisař
- Vstup: Volby s přiřazeným komisařem.
- Výstup: Založená volební událost.
- Kroky:
 1. Uživatel požádá o zobrazení detailu voleb.
 2. Systém zobrazí detail voleb.
 3. Uživatel zvolí založení volební události.
 4. Systém zobrazí formulář pro založení volební události.
 5. Uživatel vyplní formulář (název volební události, popis volební události) a odešle ho.
 6. Systém provede validaci, že je vyplněn název volební události.
 7. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 4.
 8. Systém založí volební událost.

3.5.5 Přiřazení voliče k volební události

- Aktéři: Komisař
- Vstup: Volební událost.
- Výstup: Volební událost s přiřazeným voličem.
- Kroky:
 1. Uživatel požádá o zobrazení detailu volební události.
 2. Systém zobrazí detail volební události.
 3. Uživatel zvolí přidání voliče.
 4. Systém zobrazí formulář se seznamem voličů.
 5. Uživatel vybere voliče a odešle formulář.
 6. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 4.
 7. Systém přiřadí voliče k volební události.

3.5.6 Odstranění voliče z volební události

- Aktéři: Komisař
- Vstup: Volební událost s přiřazeným voličem.
- Výstup: Volební událost s odebraným voličem.

- Kroky:

1. Uživatel požádá o zobrazení detailu volební události.
2. Systém zobrazí detail volební události.
3. Uživatel zvolí odstranění voliče ze seznamu přiřazených voličů.
4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
5. Systém odstraní voliče z volební události.

3.5.7 Zahájení nominování

- Aktéři: Komisař

- Vstup: Volební událost s voliči.

- Výstup: Volební událost se zahájenými nominacemi.

- Kroky:

1. Uživatel požádá o zobrazení detailu volební události.
2. Systém zobrazí detail volební události.
3. Uživatel zvolí zahájení nominací.
4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
5. Systém zahájí nominace.

3.5.8 Ukončení nominování

- Aktéři: Komisař

- Vstup: Volební událost s voliči a kandidáty se zahájenými nominacemi.

- Výstup: Volební událost s ukončenými nominacemi.

- Kroky:

1. Uživatel požádá o zobrazení detailu volební události.
2. Systém zobrazí detail volební události.
3. Uživatel zvolí ukončení nominací.
4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
5. Systém ukončí nominace.

3.5.9 Zahájení hlasování

- Aktéři: Komisař
- Vstup: Založená volební s voliči s kandidáty s ukončeným nominováním.
- Výstup: Volební událost se zahájeným hlasováním.
- Kroky:
 1. Uživatel požádá o zobrazení detailu volební události.
 2. Systém zobrazí detail volební události.
 3. Uživatel zvolí zahájení hlasování.
 4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
 5. Systém zahájí hlasování.

3.5.10 Ukončení hlasování

- Aktéři: Komisař
- Vstup: Volební událost s voliči s kandidáty se zahájeným hlasováním.
- Výstup: Ukončená volební událost.
- Kroky:
 1. Uživatel požádá o zobrazení detailu volební události.
 2. Systém zobrazí detail volební události.
 3. Uživatel zvolí ukončení hlasování.
 4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
 5. Systém ukončí hlasování.

3.5.11 Nominování kandidáta

- Aktéři: Volič
- Vstup: Volič přiřazený do volební události se zahájenými nominacemi.
- Výstup: Nominovaný volič (kandidát) ve volební události.
- Kroky:
 1. Uživatel požádá o zobrazení nominačního formuláře.
 2. Systém zobrazí formulář na nominování se seznamem dostupných volebních událostí.
 3. Uživatel zvolí volební událost, ve které chce být nominován, a odešle formulář.
 4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
 5. Systém nominuje voliče jako kandidáta.

3.5.12 Zamítnutí nominace

- Aktéři: Komisař
- Vstup: Volič nominovaný do volební události.
- Výstup: Zamítnutá kandidatura.
- Kroky:
 1. Uživatel požádá o zobrazení detailu volební události.
 2. Systém zobrazí detail volební události.
 3. Uživatel zvolí odstranění kandidáta se seznamu kandidátů.
 4. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 2.
 5. Systém odstraní kandidáta z volební události.

3.5.13 Hlasování

- Aktéři: Volič
- Vstup: Volič přiřazený do volební události, která má zahájené hlasování.
- Výstup: Odeslaný hlasovací lístek.
- Kroky:
 1. Uživatel požádá o zobrazení volebních událostí, ve kterých může hlasovat.
 2. Systém zobrazí seznam volebních událostí, ve kterých může voliče hlasovat.
 3. Uživatel vybere volební událost a formulář odešle.
 4. Systém zobrazí hlasovací lístek s kandidáty.
 5. Uživatel zvolí kandidáty, jimž chce udělit hlas, a lístek odešle.
 6. ALTERNATIVA Systém hlásí chybu. NÁVRAT NA KROK 4.
 7. Systém odešle hlasovací lístek.

3.6 Technologie

Při výběru technologií byly voleny open license či freeware řešení, které zajistí multiplatformnost systému.

3.6.1 Server

3.6.1.1 Java EE

Java Enterprise Edition (<http://www.oracle.com/technetwork/java/javaee/overview/index.html>) je součástí platformy Java určená pro vývoj podnikových aplikací a informačních systémů. Za jejím vývojem stojí společnost Sun Microsystems. Tuto technologii jsem zvolil pro její pokročilé možnosti a multiplatformní nasazení díky JVM. Navíc prototyp eVoleb již byl napsán v Javě EE. Java EE vyžaduje ke svému běhu navíc aplikační server, jehož volba je popsána v kapitole 3.6.1.2.

3.6.1.2 Aplikační server

Prototyp systému eVolby[7] běžel na aplikačním serveru Glassfish 2.1.1. Tato verze již ovšem zastarává a je na ní problematické testování. Proto jsem zvolil aplikační server Glassfish 3.1 (<http://glassfish.java.net>), který přináší mnohá zlepšení. Glassfish jsem zvolil proto, že se jedná o referenční implementaci Javy EE6 přímo společností Sun Microsystems. Glassfish je open source projekt, ale existuje i jeho komerční verze, jenž se jmenuje Oracle Glassfish 3.1. Tyto dvě verze se od sebe v podstatě neliší, rozdíl je hlavně v podpoře a aktualizacích.

3.6.1.3 Databáze

Java EE disponuje sadou různých databázových konektorů díky technologii JDBC, které umožňují databázový stroj kdykoliv změnit. Při vývoji bude používána databáze MySQL (<http://www.mysql.com>). Pro snadné nasazení je ale možné použití JavaDB (<http://www.oracle.com/technetwork/java/javadb/overview/index.html>), což je vestavěná databáze. Jedná se vlastně jen o knihovnu napsanou v jazyce Java, která umožňuje přístup k datům uloženým na pevném disku za pomoci SQL dotazů.

3.6.2 Klient

3.6.2.1 Java SE

Java SE slouží k tvorbě desktopových aplikací, které běží pod JVM. Je tedy opět multiplatformní.

3.6.3 Vývojové a testovací prostředí

Vývoj probíhal na operačním systému Ubuntu 11.04 a bylo použito vývojové prostředí NetBeans IDE ve verzi 7.0. Jedná se o komplexní prostředí, které umožňuje ovládání aplikačního serveru Glassfish a napojení JDBC ovladačů na databázový stroj. Pro verzování projektu jsem používal GIT s klientem SmartGIT (<http://www.syntevo.com/smartgit/>) a repozitář projektu je veřejně dostupný na adrese <https://github.com/cerevka/eVoting>.

Pro testování webového prostředí posloužil webový prohlížeč Firefox 4.0.1.

3.7 Snímač otisků prstů

V kapitole 2.2.5 jsem rozebíral výhody a nevýhody jednotlivých technologií pro snímání otisků prstů. Pro vývoj klientské části, která má umožňovat autentizaci na základně otisku prstu, je třeba vybrat konkrétní snímač s kompatibilním SDK, které umožňuje práci s Javou. Využitím biometrických údajů se zabývá mnoho společností a většina z nich nabízí i vlastní SDK určené právě pro jejich čtečky, což omezuje výběr.

V úvahu je také třeba vzít ceny licencí SDK. V požadavcích v kapitole 2.3.1.1 je stanoveno, že budou využity open license či freeware technologie. Proto se klíčovým atributem pro výběr snímače otisků prstů stala cena licence SDK společně s podporou platformy Java. Při výběru

jsem dbal i na to, aby SDK a snímače bylo možné používat pod operačními systémy MS Windows a Linux.

Pro detailnější prozkoumání jsem se zaměřil na produkty následujících společností, které nabízí jak samotné snímače otisků prstů, tak i k nim kompatibilní SDK.

3.7.1 Griaule Biometrics

Griaule Biometrics² nabízí velmi robustní Fingerprint SDK s podporou více než 40 snímačů, ovšem při detailním zkoumání se ukázalo, že jejich verze SDK pro Javu má tento výběr omezen na zhruba čtvrtinové množství.

SDK spolupracuje s mnoha platformami (C++ .NET, C# .NET, Delphi 6-7, Visual Basic 6, VB .NET, ASP.net, Java, Java Applet) a je k dispozici i pro MS Windows i pro Linux.

Dále v neprospěch tohoto SDK hovoří i licenční politika, která vyžaduje uhrazení licence pro každý vývojářský počítač, na kterém bude Fingerprint SDK nainstalováno. Cena jedné licence se pohybuje kolem \$100.

3.7.2 Futronic

Společnost Futronic³ vyvinula pro svůj snímač otisků prstů Futronic FS80 vlastní SDK, které podporuje MS Windows, Linux i MacOS a může se pochlubit spoluprací s platformami VB6, VC6, Borland Delphi, VB.NET a VC.NET, VB.2005, VC.2005 a samozřejmě Javou.

Bohužel toto SDK není veřejně dostupné a pro více informací či jeho získání je třeba kontaktovat přímo společnost Futronic.

3.7.3 Digital Persona

Jako nejvhodnější se ukázalo SDK od společnosti Digital Persona⁴, které je nabízeno jako freeware a k němuž je nabízen placený doplněk pro rychlejší porovnávání otisků prstů. Základní verze SDK však zcela postačuje, je k dispozici ve verzi pro MS Windows i pro Linux a spolupracuje s platformami Java, C, C++ a .NET.

Digital Persona k tomuto SDK nabízí i svůj vlastní snímač otisků prstů pod označením U.are.U 4500 Fingerprint Reader v ceně \$99.

3.7.4 Vybrané SDK a snímač otisků prstů

Pro vývoj jsem tedy zvolil SDK od společnosti Digital Persona a jejich snímač otisků prstů U.are.U 4500 Fingerprint Reader. Volba na něj padla především díky volně dostupnému SDK, které mají ostatní společnosti zpoplatněné, podpoře MS Windows a Linux a platformy Java.

²<http://www.griaulebiometrics.com>

³<http://www.futronic-tech.com>

⁴<http://www.digitalpersona.com>

Kapitola 4

Návrh řešení a realizace

Vzhledem k tomu, že systém vychází z prototypu systému eVolby[7], o kterém hovoří řešerše v kapitole 2.2.1 a který již má své řešení navržené, budu v této práci popisovat jednotlivé iterace, ve kterých jsem ho modifikoval, přičemž v každé iteraci se zaměřím na problémy, se kterými jsme se setkal a popíši postup, kterým jsem se dopracoval k funkčnímu řešení.

4.1 První iterace

4.1.1 Cíl

Jako první jsem si stanovil za cíl migraci z Glassfish 2.0.1 na Glassfish verze 3.1. K tomuto rozhodnutí mě vedla hlavně podpora nejnovějších technologií Javy Enterprise Edition 6¹. S tím se pojí i možnost unit testů v EJB, jak uvádí i závěrečná zpráva z vývoje prototypu eVolby[7].

4.1.2 Postup

4.1.2.1 Sdílené knihovny

Zatímco Glassfish 2.0.1 byl certifikován pro Java EE 5, jeho novější verze 3.1 je již certifikována pro Java EE 6. S tím souvisí změna některých standardů, která měla na svědomí problémy při migraci. Zjistil jsem, že při nasazení aplikace na Glassfish 3.1 dochází k chybě kvůli tomu, že EJB moduly neobsahují žádné EJB. To bylo způsobováno tím, že Glassfish vynechával ty EJB, ve kterých narazil na jakoukoliv chybu. Problém tkvěl ve sdílené knihovně POJO objektů, které slouží jako přepravní kontejnery mezi EJB moduly. Jakmile remote interface dané EJB obsahoval referenci na takový sdílený POJO objekt, Glassfish 3.1 zahlásil, že takový objekt nezná.

Tento problém se mi po dařilo odstranit zapouzdřením všech tří EJB modulů spolu s webovým modulem pod jednu enterprise aplikaci, která se pak nasazuje na aplikační server jako celek namísto nasazování každého modulu zvlášť. Díky tomu je i nasazování aplikace

¹http://glassfish.java.net/public/comparing_v2_and_v3.html

jednoduší, protože se provádí pouze jednou a nemusí se hlídat, v jakém pořadí se mají moduly nasadit.

4.1.2.2 Entity

Dále se objevily chyby v definici entit, kdy některé entity neměly pojmenované spojovací sloupce přes anotaci `@JoinColumn`. To způsobovalo nevygenerování tabulek do databáze.

U vazeb mezi entitami bylo třeba doplnit kaskádní chování při jejich ukládání (`CascadeType.PERSIST`), jinak docházelo k přerušení transakcí, aby se zabránilo nekonzistenci dat.

4.1.2.3 Databáze

V původním návrhu je EJB modul Controller napojen na dvě databáze - první databáze obsahuje data o volebních událostech a druhá data o uživateli systému. Jelikož uživatelská databáze měla pouze jednu tabulku, sloučil jsem tyto databáze do jedné, což vede k zjednodušení konfigurace při nasazení na server. Sloučení spočívalo v odstranění příslušného persistentního unitu (`PersistentUnit`) a překonfigurování EJB tak, aby namísto něj používaly persistentní unit pro databázi kontroleru.

4.1.3 Výstup

Po skončení této iterace jsem měl funkční nasazení aplikace na Glassfishi 3.1 využívající pouze tři databáze namísto čtyř s korektně vygenerovanými tabulkami.

4.2 Druhá iterace

4.2.1 Cíl

Druhým krokem bylo zprovoznění základní funkcionality a její otestování společně se zlepšením zabezpečení.

4.2.2 Postup

4.2.2.1 Synchronizace persistentní vrstvy

Při zakládání voleb a volebních událostí docházelo k chybě, která se na první pohled neprojevovala. Byla způsobena až v na úrovni modulů `Validator` a `Counter`, do nichž se reflektovalo založení v modulu `Controller`.

Chyba byla způsobována tím, že těsně po uložení entity do databáze se předpokládalo, že entita již bude v databázi i fyzicky. Tomu tak ovšem nebylo a proto se namísto přiděleného ID předával `null`. Před samotnou další prací s entitou je třeba synchronizovat persistentní a datovou vrstvu zavoláním metody `flush` na manažeru entit (`EntityManager`).

4.2.2.2 SQL dotazy

Logickým krokem byla oprava a optimalizace SQL dotazů nad databází. Odstranil jsem tedy všechny NativeQuery a nahradil je NamedQuery, které jsem nadefinoval přímo u entit a pojmenoval jsem je za pomoci konstant. Díky tomu bude v budoucnu jednodušší refaktoring entit a i další vývoj, kdy je seznam všech NamedQuery dostupný přes našeptávač IDE a nemůže tak dojít k překlepům.

Stejným způsobem jsem změnil i výběr neukončených volebních událostí, které se vybíraly zcela neefektivně - položil se dotaz na databázi pro výběr všech událostí, přes ty se následně iterovalo a hledaly se ty, které měly příznak „finished“ na hodnotě „false“.

4.2.2.3 Zabezpečení

Původní projekt využíval pro autentizaci uživatelů file realm Glassfish. Toto nicméně vedlo k duplikaci dat, neboť každý uživatel musel být jak v tomto file realmu, tak v databázi uživatelů. Nakonfiguroval jsem tedy v Glassfishi JDBC realm, který přes JNDI přistupuje do interní databáze uživatelů a zajišťuje jejich autentizaci na základě dat v uvedených tabulkách.

V rámci zabezpečení jsem také nastavil, aby i přihlašovací formulář byl zabezpečen HTTPS spojením. Původní formulář navíc využíval metodu GET, která parametry předává přes adresní řádku, což je v případě předávání hesla nepřipustné. Nově přihlašovací formulář využívá metodu POST.

4.2.2.4 JSF 2.0

Jak již bylo zmíněno v první iteraci 4.1, nová verze Glassfish 3.1 s sebou přináší nejnovější technologie. To v sobě zahrnuje i přechod z JSF 1.2 na verzi JSF 2.0. Původní prototyp systému eVolby[7] využíval střídavě technologií JSP i tohoto zastaralého JSF.

Předělal jsem všechny webové stránky tak, aby jednotně používaly JSF 2.0 za využití systému šablon. Stránky tedy nyní do sebe nevkládají společnou hlavičku a patičku, jak tomu bylo původně, nýbrž využívají šablonovacího systému.

Dalším přínosem šablon je i skutečnost, že nyní jsou menu pro jednotlivé role na jednom místě a nejsou na každé stránce definovány znovu. V původním konceptu by přidání jedné položky do menu obnášelo aktualizaci všech stránek.

4.2.2.5 Volební applet

Applet napsaný v Javě FX se při testování prototypu systému eVolby[7] ukázal jako nepoužitelný. Hlavním problémem byla verze nainstalované Javy na klientském počítači, kde se applet spouštěl, společně s problematickou komunikací s hostitelským serverem.

Na doporučení ze závěrečné zprávy jsem tento applet z projektu vyřadil a nahradil ho klasickým formulářem. Ten zajišťuje mnohem pohodlnější, rychlejší a hlavně funkční hlasování.

4.2.3 Výstup

Na konci druhé iterace je projekt ve své základní funkční podobě, umožňuje snadnější přidávání uživatelů a bezproblémové hlasování. Prozatím není implementována nová funkcionality pro roli kandidáta a aktéra času.

4.3 Třetí iterace

4.3.1 Cíl

V poslední části jsem si vzal za úkol vylepšení stávající funkcionality a zpříjemnění používání aplikace běžnému uživateli.

4.3.2 Postup

4.3.2.1 Grafické úpravy

Jedním z takových kroků byly i úpravy CSS stylů a odladění grafického vzezření některých sekcí, které by mohly být zavádějící. S tím se pojí i zlepšení navigace po aplikaci.

4.3.2.2 Odebrání komisaře

Administrátorovi byla přidána možnost odebrat komisaře z voleb.

4.3.2.3 Multijazyčnost

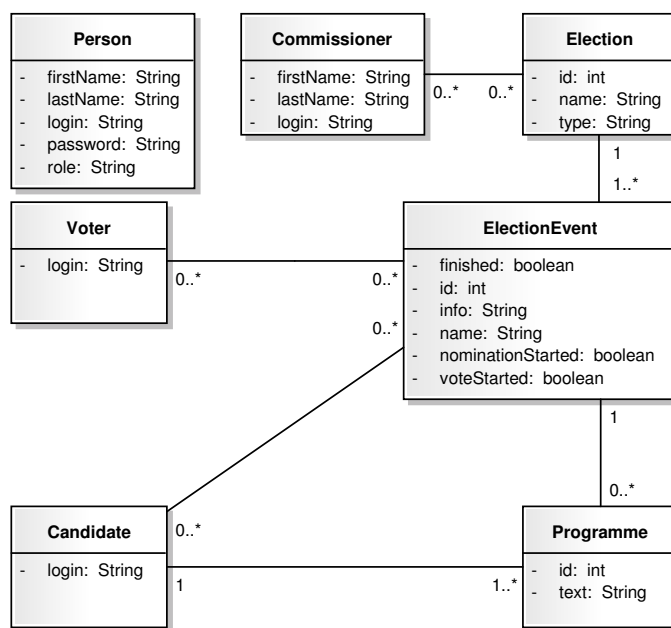
Aby práce s aplikací byla co nejpříjemnější pro co největší množství uživatelů, integroval jsem do ní podporu multijazyčnosti. Všechny řetězce, které byly napevno vepsány do zdrojového kódu jsem nahradil značkami, které se nahrazují překlady dle zvoleného jazyka. Aktuálně aplikace podporuje angličtinu a češtinu.

Všechny překlady jsou umístěny ve speciálním properties souboru a pro vytvoření nové jazykové mutace stačí přeložit pouze tento jeden soubor. Ten je přímo zakompilován do aplikace a Java EE jej obsluhuje za pomoci vestavěných nástrojů, tedy práce s ním nijak nezpomaluje chod aplikace.

4.3.2.4 Datový model

Diagram 4.1 zachycuje aktuální podobu datového modelu. Od původního návrhu datového modelu z prototypu systému eVolby[7] se poměrně liší a je značně nelogicky uspořádan. Hlavním problémem vidím v tom, že entity jsou spolu špatně navázány na úrovni objektového návrhu a následně se s nimi v projektu špatně pracuje, případně některé změny jsou neuskutečnitelné.

Předpokládám, že tento problém vznikl až při skutečné implementaci objektů z důvodu nedostatečné znalosti toho, jak funguje persistentní vrstva v Java EE. To vedlo na návrh, kdy se jednotlivé objekty na sebe váží přes své ID a nepoužívají persistentních vztahů. Takový datový model se dá sice dobře používat při klasickém použití relační databáze, nikoliv však při objektově relačním mapování.



Obrázek 4.1: Aktuální datový model v projektu.

4.3.2.5 Role kandidáta

Pokusy o přidání role kandidáta selhaly kvůli špatně navrženému datovému modelu aplikace (viz. kapitola 4.3.2.4), který nepodporuje možnost, aby jeden uživatel měl více rolí. Vyplývá to z diagramu 4.1, kde je vidět, že ke každému účtu patří právě jedna role. Navíc tento návrh porušuje normálovou formu, jelikož zde dochází k redundanci dat, na které by se mělo odkazovat pomocí klíčů.

4.3.2.6 Terminálový klient

Kvůli problémům s objednáním a následným doručením snímače otisků prstů, který byl vybrán v kapitole 3.7.4 a který nebyl k dispozici ani v době dokončení práce, nebyl terminálový klient na platformně Java SE realizován. Z časových důvodů již nebyl možný ani vznik jeho zjednodušené verze bez snímače.

4.3.3 Výstup

Po dokončení těchto úprav se již nedochází ke grafickým defektům a aplikace plně podporuje multijazyčné rozhraní, do kterého se dá kdykoliv přidat další jazyková mutace. Výstupem jsou i závěry, kvůli kterým nemohly být realizovány všechny požadavky navržené v kapitole 3.

Kapitola 5

Testování

Celý systém musí být řádně otestován, jelikož každá chyba v průběhu ostrých voleb, může mít zásadní dopad na jejich průběh i jejich výsledek. Hrozby elektronických voleb popisuje kapitola 3.2.1. Aplikace musí být srozumitelná nejen počítačově gramotným uživatelům, ale i uživatelům, kteří nemají zkušenosti s ovládáním počítače.

5.1 Selenium

Při testování webových aplikací je třeba otestovat i průchod uživatele skrz webové stránky. K tomu jsem použil testovací nástroj Selenium. Nabízí se dvě metody testování - buďto za pomoci pluginu pro prohlížeč Firefox, nebo za pomoci speciálního serveru Selenia. Já jsem si zvolil druhý způsob, jelikož umožňuje integraci do vývojového prostředí (pro Netbeans doplněk „Selenium Plugin for Ant Projects“). Samotné testy jsou pak velice podobné unit testování za pomoci JUnitu.

Po spuštění testu se začnou předávat příkazy serveru Selenia, který simuluje pohyb uživatele po webu a dokáže zacílit kterýkoliv element a provést na něm řadu akcí. Lze takto vytvořit sadu testů, která vždy otestuje, že novými zásahy nedošlo k poškození stávající funkcionality. Testy je nutné spouštět na nezabezpečeném spojení, jelikož Selenium si neumí poradit s varovným hlášením o nedůvěryhodném certifikátu, který Glassfish používá, pokud se do něj nenainstaluje důvěryhodný certifikát podepsaný certifikační autoritou.

Tvorba testů pro Selenium se dá zjednodušit zmiňovaným doplňkem pro Firefox. Ten totiž umožňuje nahrávat akce a výsledné záznamy znovu spouštět či vyexportovat to mnoha programovacích jazyků právě pro vzdálené volání za pomoci serveru Selenia.

5.2 Apache Benchmark

Pro zátěžové testování jsem zvolil nástroj Apache Benchmark, který dokáže produkovat požadavky na server. Stačí mu nastavit, kolik požadavků má být celkem posláno a kolik z nich má odejít zároveň.

Testování jsem prováděl jen pro zobrazení přihlašovací stránky, jelikož zbytek aplikace je přístupný až po zadání přihlašovacích údajů, což nelze při tomto druhu testování simulovat. Výsledky testů jsou zaznamenány v tabulce 5.1.

Požadavků	Současných požadavků	Čas na odpověď [ms]	Čas celkem [s]
100	1	5.149	0.515
1 000	1	5.015	5.015
1 000	10	3.127	3.127
10 000	1	4.904	49.039
10 000	10	3.026	30.257
10 000	100	2.922	29.223

Tabulka 5.1: Výsledky zátěžového testování pomocí Apache Benchmark

Testy ukazují, že Glassfish 3.1 nemá nejmenší problémy se zpracováním obrovského náporu požadavků a že časy jeho odpovědí jsou v řádech milisekund.

Kapitola 6

Závěr

Cílem práce bylo upravit stávající prototyp systému eVolby[7] pro lokální hlasování za použití terminálů se snímači otisků prstů. Během vývoje jsem uskutečnil celou řadu úprav, které mezi sebe zahrnují hlavně migraci ze zastaralého aplikačního serveru Glassfish 2 na Glassfish 3.1, který již podporuje nejnovější platformu Java Enterprise Edition 6. To s sebou neslo nemalé překážky, které tkvěly hlavně ve změnách specifikace. Nicméně migrace se i přesto plně zdařila.

Další úpravy se zaměřily na vyladění aplikace a opravu chyb, které způsobovaly její pád, což velmi často obnášelo i aktualizaci zastaralých postupů, které byly v aplikaci používány. Konkrétně se například jednalo o používání technologie JSP a JSF 1.2, která je nyní nahrazena verzí 2.0. V posledních fázích došlo k přidávání nové funkcionality. Jako nejvýznamnější přínos v oblasti uživatelsky přívětivého prostředí považuji integraci multijazyčnosti, díky které je možné kdykoliv si přepnout jazyk, ve kterém aplikace komunikuje. Samozřejmostí je snadné přidání další jazykové mutace.

Do doby dokončení této práce se bohužel nepodařilo obstarat snímač otisků prstů, se kterým by bylo možné uvedení klientského terminálu do provozu tak, jak bylo zamýšleno. Jelikož se stav prototypu ukázal jako nevyhovující, zaměřil jsem se hlavně na úpravy serverové části, jejichž rozsah předčil očekávání.

Práce v sobě také shrnuje nedostatky, které se v aplikaci vyskytují a které komplikují její další vývoj. Jedná se hlavně o nedokonalý datový model, který plně nevyužívá možností persistence, což vede ke špatné manipulaci s daty. Pro další vývoj serverové části bych doporučil přehodnocení celého datového modelu a čistou implementaci odpovídající business logiky.

Literatura

- [1] Alexander Prosser, Robert Krimmer. *Electronic Voting in Europe - Technology, Law, Politics and Society*. Gesellschaft für Informatik, 2004.
- [2] Bc. Michal Petřík. Hlasovací systém pro samosprávné orgány vysokých škol. *Diplomová práce ČVUT FEL*, 2011.
- [3] Filip Weber. DoS a DDoS útoky a ochrana proti nim.
<http://www.svetsiti.cz/view.asp?rubrika=Technologie&clanekID=324>, stav z 21.03.2011.
- [4] Ing. Radek Šilhavý. Experimentální ověření distribuovaného volebního schématu. *Disertační práce UTB FAI*, 2011.
- [5] Jakub Valenta. Mobilní hlasovací zařízení. *Bakalářská práce ČVUT FEL*, 2010.
- [6] Jan Ambrož. E-volby jsou v ČR vzdálenou budoucností.
<http://www.lupa.cz/clanky/e-volby-jsou-v-cr-vzdalenou-budoucnosti>, stav z 23.01.2011.
- [7] Tomáš Čerevka, Jan Zahula, Pavel Valenta, Rado Murin. Prototyp systému eVolby.
<http://code.google.com/p/evolby/>, stav z 21.03.2011.
- [8] Václav Rada. Internet v praxi: Komunální volby v Estonsku, dočkáme se i u nás?
<http://www.internetprovsechny.cz/internet-v-praxi-komunalni-volby-v-estonsku-dockame-se-i-u-nas>, stav z 23.01.2011.
- [9] Zuzana Poláčková. Rešerše algoritmů pro snímání a zpracování otisku prstů. *Bakalářská práce ČVUT FEL*, 2008.
- [10] ČTK. V Česku by se za šest let mohlo volit přes internet.
http://www.czso.cz/csu/redakce.nsf/i/7_4_2008_v_cesku_by_se_za_sest_let_mohlo_volit_pres_internet, stav z 30.01.2011.

Příloha A

Seznam použitých zkratek

ČSÚ	Český statistický úřad
ČTK	Česká tisková kancelář
ČVUT	České vysoké učení technické v Praze
DDoS	Distributed Denial of Service
EJB	Enterprise Java Bean
FAI	Fakulta aplikované informatiky
FEL	Fakulta elektrotechnická
FIT	Fakulta informačních technologií
Java EE	Java Enterprise Edition
Java SE	Java Standart Edition
JDBC	Java Database Connectivity
JSF	Java Server Faces
JSP	Java Server Pages
JVM	Java Virtual Machine
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
LDAP	Lightweight Directory Access Protocol
POJO	Plain Old Java Object
SDK	Software Development Kit

SQL Structured Query Language

UML Unified Modeling Language

UTB Univerzita Tomáše Bati ve Zlíně

XML eXtensible Markup Language

Y36SI2 Řízení SW projektů

Y36SI3 Realizace programových systémů

Příloha B

Instalační a uživatelská příručka

Zde je popsán postup, jak je správně aplikaci nainstalovat a jak nastavit aplikační server Glassfish 3.1, aby aplikace fungovala.

B.1 Potřebné programy

- Java JRE a JDK (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- Glassfish Server Open Source Edition 3.1 (<http://glassfish.java.net/downloads/3.1-final.html>)
- MySQL Server (<http://www.mysql.com/downloads/mysql/>)
- Netbeans 7.0 (<http://netbeans.org/>) s pluginy:
 - Java SE
 - Java Web and EE
 - JUnit
 - JSF
 - Selenium Plugin for Ant Projects
- Internetový prohlížeč

B.2 Instalace

B.2.1 Glassfish Server Open Source Edition 3.1

1. Stáhnout a instalovat podle pokynů instalátoru.
2. Zvolené heslo bude třeba při pozdějším použití.

B.2.2 MySQL Server

1. Stáhnout a instalovat podle pokynů instalátoru.
2. Zvolené heslo bude třeba při pozdějším použití.

B.2.3 Netbeans 7.0

1. Stáhnout a instalovat podle pokynů instalátoru.
2. Instalovat potřebné pluginy přes *Tools -> Plugins -> Available Plugins -> Označit vybrané -> Install*.
3. Restartovat Netbeans.

B.3 Nastavení

B.3.1 MySQL Server

1. Vytvořit databázového uživatele **eVoting** s heslem **eVoting**.
2. Vytvořit tři databáze se jmény:
 - **eVotingController**
 - **eVotingCounter**
 - **eVotingValidator**
3. Databázovému uživateli **eVoting** přidělit plná práva k těmto databázím.
4. V Netbeans v záložce **Services** připojit MySQL server pod databázovým rootem.
5. U všech vytvořených databází změnit uživatele na **eVoting** a zadat jeho heslo **eVoting**.
6. U jednotlivých EJB modulů upravit *Configuration Files -> persistence.xml (New Data Source)*:
 - **eVotingController -> mysql/eVotingController**
 - **eVotingCounter -> mysql/eVotingCounter**
 - **eVotingValidator -> mysql/eVotingValidator**

B.3.2 Glassfish Server Open Source Edition 3.1

Následující nastavení se provádí v administrační konzoli aplikačního serveru Glassfish 3.1, která je po jeho zapnutí dostupná defaultně na adrese <http://localhost:4848/>. Všechny požadované položky lze nalézt přes levé rozbalovací menu.

B.3.2.1 Fronty

1. *Resources -> JMS Resources -> Connection Factories -> New...*
 - *Pool name:* `jms/validatedVotesFactory`
 - *Resource Type:* `javax.jms.QueueConnectionFactory`
2. *Resources -> JMS Resources -> Destination Resources -> New...*
 - *JNDI Name:* `jms/validatedVotes`
 - *Physical Destination Name:* `ValidatedVotesQueue`
 - *Resource Type:* `javax.jms.Queue`

B.3.2.2 E-mail

1. *Resources -> JavaMail Sessions -> New...*
 - *JNDI Name:* `mail/evolbyMailSession`
 - *Mail Host:* dle mailového serveru
 - *Default User:* dle mailového serveru
 - *Default Sender Address:* dle mailového serveru
 - *Transport Protocol:* `smtps`
 - *Transport Protocol Class:* `com.sun.mail.smtp.SMTPSSLTransport`
2. Do Additional Properties přidat:
 - `mail-smtps-auth -> true`
 - `mail-smtps-password -> heslo k e-mailu`

B.3.2.3 Security Realm

1. *Configurations -> server-config -> Security Realms -> New...*
 - *Name:* `eVotingRealm`
 - *Class Name:* `com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm`
 - *JAAS Context:* `jdbcRealm`
 - *JNDI:* `mysql/eVotingController`
 - *User Table:* `person`
 - *User Name Column:* `login`
 - *Password Column:* `password`
 - *Group Table:* `person`
 - *Group Name Column:* `personGroup`
 - *Digest Algorithm:* `none`

B.4 Spuštění

1. Spustit Glassfish 3.1.
2. Na projektu `eVoting` zvolit v kontextovém menu *Deploy* v Netbeans.
3. V databázi *eVotingController* přidat do tabulky *person* uživatele. Možné role jsou:
 - `administrators`
 - `commissioners`
 - `voters`
4. Aplikace je připravena na adrese `https://localhost:8181/evoting/`.

Příloha C

Obsah přiloženého CD

```
cd/
|-- abstraktCZ.txt
|-- abstraktEN.txt
|-- bin
|   '-- eVoting.ear
|-- doc
|   |-- eVotingController
|   |-- eVotingCounter
|   |-- eVotingLibrary
|   |-- eVotingValidator
|   '-- eVotingWeb
|-- readme.txt
|-- src
|   |-- build
|   |-- build.xml
|   |-- dist
|   |-- eVotingController
|   |-- eVotingCounter
|   |-- eVotingLibrary
|   |-- eVotingValidator
|   |-- eVotingWeb
|   |-- nbproject
|   '-- src
'-- txt
    |-- csplainnat.bst
    |-- diagrams
    |-- ea
    |-- evoting.aux
    |-- evoting.bbl
    |-- evoting.blg
    |-- evoting.dvi
    |-- evoting.lof
```

```
|-- evoting.log  
|-- evoting.lot  
|-- evoting.pdf  
|-- evoting.ps  
|-- evoting.tex  
|-- evoting.toc  
|-- figures  
|-- hyphen.tex  
|-- k336_thesis_macros.sty  
|-- Makefile  
|-- reference.bib  
'-- texput.log
```