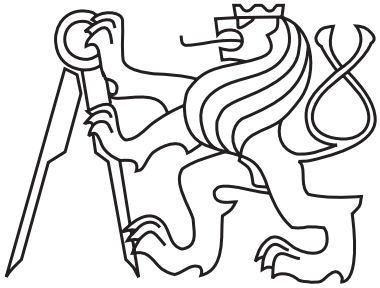
Zadání

Na základě existujícího prototypu navrhněte a realizujte 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

16. května 2011

Poděkování

Zde můžete napsat své poděkování, pokud chcete a máte komu děkovat.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém 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, které jsou označovány jako „Nová BSD licence“.

Nová BSD licence

V Chebu dne 15. 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

Seznam obrázků

Seznam tabulek

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 otázkách[1]. 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 elektronického 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 2.5.

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 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.

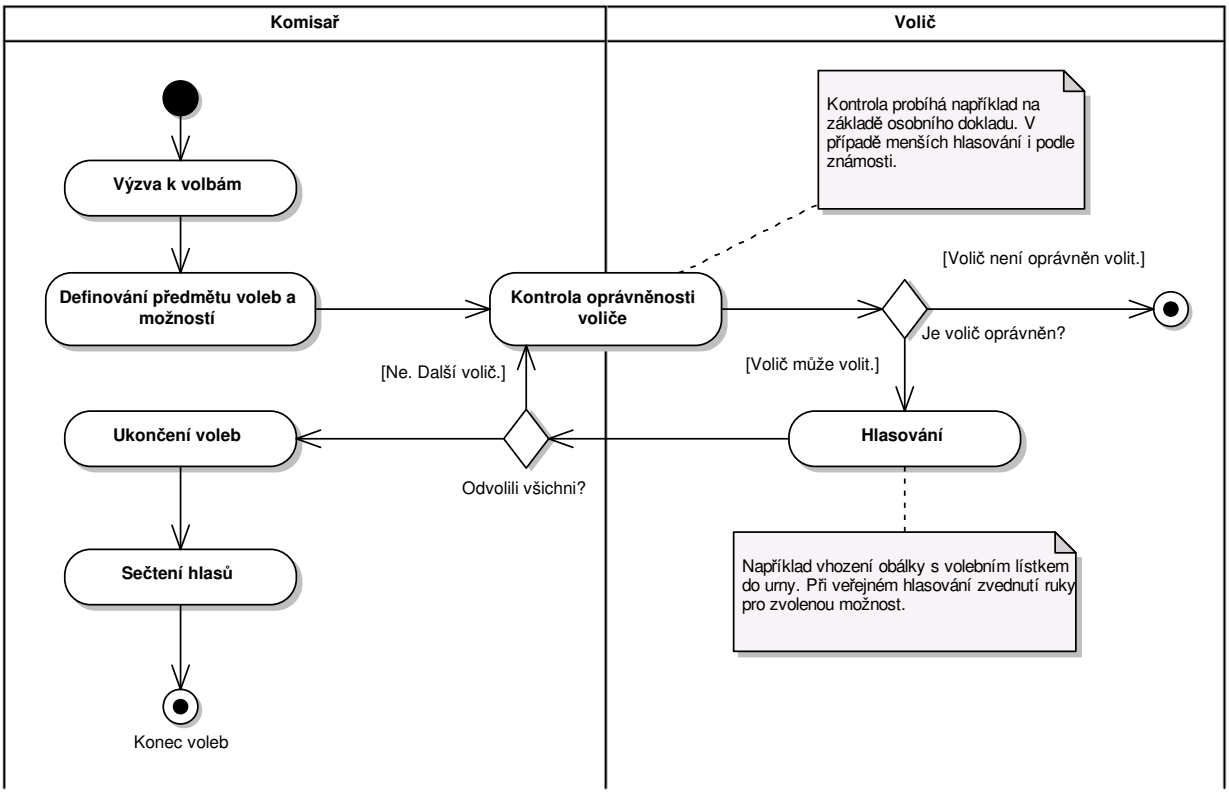
2.2 Rešerše

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 této práci bude systém rozšířen o možnost i nepersonálních voleb, kdy předmět voleb a možnosti budou definovány explicitně při jejich zakládání.



Obrázek 2.1: Diagram aktivit voleb bez elektronického systému.

V systému figurují tři role uživatelů: administrátor, komisař a volič. Pro lepší správu kandidátů přidává nový koncept další uživatelskou roli: kandidáta. Administrátor je v systému pouze jeden a má pravomoc založit volby a jmenovat jim komisaře. Jedny volby 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 nasazovány ve Spojených státech amerických, Německu, Velké Británii, Španělsku, Švýcarsku a Estonsku.

Zbytek práce se již orientuje více prakticky na návrh konceptuálního řešení distribuo-

vaného volebního schématu a stanovuje hypotézy pro experimentální ověření hypotéz. 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

Problematikou 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, speciálního 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, který si za vzor bere *Mobilní hlasovací zařízení* [5] Jakuba Valenty. Autor se sice zabývá problematikou hlasovacího zařízení v podobě mobilního telefonu s technologií Wi-Fi, 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, datová, 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 pro 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 Poláčkové 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, hrozících 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 průkazů[1]. 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 Nefunkční požadavky

V této sekci jsou stanoveny požadavky, které se netýkají funkčnosti, nýbrž jsou kladeny na celkový návrh, použité technologie a zpracování.

2.3.1.1 Obecné požadavky

Na celkový návrh řešení a dokumentaci jsou kladeny následující požadavky:

1. Systém bude rozdělen na serverovou a klientskou část.

[1] Obdoba našich občanských průkazů.

2. Instalace systému a jeho uvedení do provozu bude co nejjednodušší a nejrychlejší.
3. Systém bude multiplatformní díky běhovému prostředí JVM.
4. Systém bude používat open source nebo freeware technologie.
5. Systém zajistí bezpečnost hlasování, tj. nezfalšovatelnost hlasů a tajnost voleb vůči druhé i třetí osobě.
6. Součástí systému 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 a Java FX.

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. registrace voliče.
5. přiřazení voličů k volební události.
6. nominování kandidátů.
7. hlasování.
8. prezentaci výsledků.

2.4 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.

2.4.1 Registrace voličů

Před zahájením voleb je třeba do systému zanést seznam voličů s jejich nacionáliemi, 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 sejmutí otisku prstu a vygenerování přístupového PINu. Otisk prstu zabrání 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.

2.4.1.1 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 naimportovat celé seznamy voličů bez ohledu na to, zda již jsou v systému zaregistrování, nebo jejich předpřipravené účty zatím na registraci čekají.

2.4.2 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 voliče[2] a nastaví detaily voleb. Jedná-li se o volby personální, následují nominace popsané v kapitole 2.4.3. 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.

[2] Viz. kapitola 2.4.1.1

2.4.3 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.

2.4.4 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. Server hlas uloží bez informace, komu patří, a u voliče si nastaví příznak, že již odhlasoval.

2.4.5 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.

2.5 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 do pozadí 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.

2.5.1 Hrozby

Hrozby 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 neodhaleny.

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.

2.5.2 Zabezpečení

U voleb musí byt zajištěno nezfalšování 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 neevidování 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.

Kapitola 3

Analýza

3.1 Uživatelské role

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

3.1.1 Administrátor

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

3.1.2 Komisař

Komisař je jmenován administrátorem a je přiřazen k volbám. Komisařů pro jedny volby můžu být více, pak v důležitých momentech voleb rozhoduje nadpoloviční většina z nich. 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.

3.1.3 Volič

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

3.1.4 Kandidát

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

3.2 Případy užití

Případy užití jsou částečně popsané v předchozích kapitolách. Pro úplnost je na obrázku 3.1 znázorněn use case diagram. Oproti předchozím kapitolám jsou na něm zachyceny i systémové role v podobě uživatelů „Systém“ a „Čas“. Systémový uživatel představuje činnosti, které jsou v systému zautomatizovány a nahrazují lidskou práci. Uživatel Čas je abstraktní podobou toku času a znázorňuje možnost automatického spouštění daných činností v nadefinovanou dobu.

3.3 Technologie

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

3.3.1 Server

3.3.1.1 Java EE

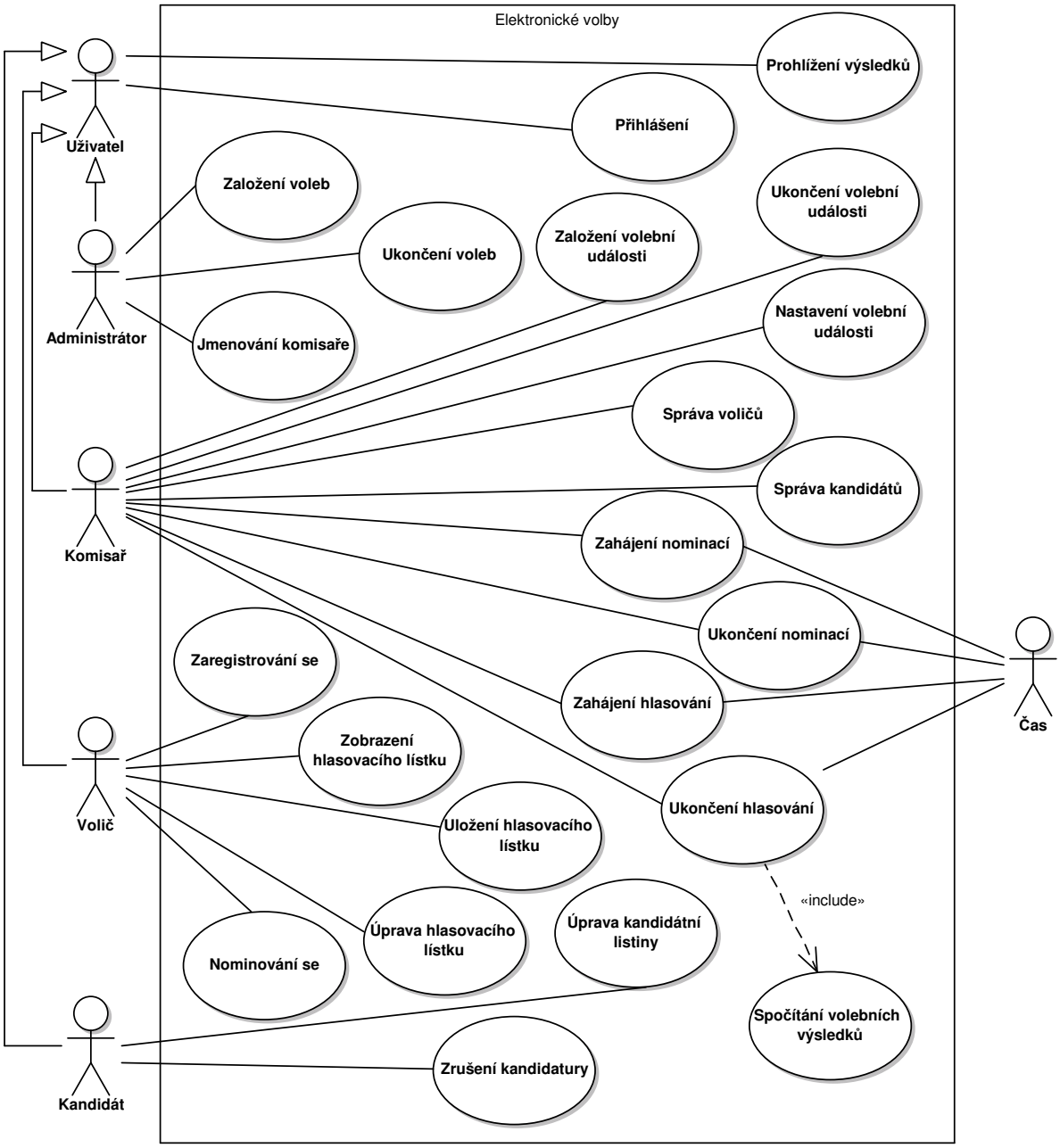
Java Enterprise Edition (`http://www.oracle.com/technetwork/java/javaee/overview/index.html`) je součást platformy Java určená pro vývoj podnikových aplikací a informačních systémů. Za jejím vývojem stojí společnost Sun Microsystems. Tuto technologii jsem zvolil pro její pokročilé možnosti a multiplatformní 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 3.3.1.2.

3.3.1.2 Aplikační server

Prototyp eVoleb běžel na aplikačním serveru Glassfish 2.1.1. Tato verze již ovšem zastarává a je na ní problematické testování [7]. 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.

3.3.1.3 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.



Obrázek 3.1: Diagram případů užití.

3.3.2 Klient

3.3.2.1 Java SE

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

3.3.3 Vývojové a testovací prostředí

Vývoj probíhal na operačním systém Ubuntu 11.04 a bylo použito vývojové prostředí NetBeans IDE ve verzi 7.0. Jedná se o komplexní prostředí, které poskytuje snadnou správu kódu s integrací SVN pro verzování. Zároveň umožňuje ovládání aplikačního serveru Glassfish a napojení JDBC ovladačů na databázový stroj.

Pro testování webového prostředí posloužil webový prohlížeč Firefox 4.0.1.

3.4 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 s 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 source 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.

3.4.1 Griaule Biometrics

Griaule Biometrics[1] 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.

3.4.2 Futronic

Společnost Futronic[2] 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.

[1] `http://www.griaulebiometrics.com`
[2] `http://www.futronic-tech.com`

3.4.3 Digital Persona

Jako nejvhodnější se ukázalo SDK od společnosti Digital Persona[3], 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.

3.4.4 Vybrané SKD 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 padla na něj především díky freewarovému SDK, které mají ostatní společnosti zpoplatněné, podpoře MS Windows a Linux a platformy Java.

[3] `http://www.digitalpersona.com`

Kapitola 4

Návrh řešení a realizace

Vzhledem k tomu, že systém vychází z prototypu systému eVolby, 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 Glassfishe 2.0.1 na Glassfish verze 3.1. K tomuto rozhodnutí mě vedla hlavně podpora nejnovějších technologií Javy Enterprise Edition 6[1]. 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 Glassfish 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 jednoduší, protože se provádí pouze jednou a nemusí se hlídat, v jakém pořadí se mají modulu nasadit.

[1] `http://glassfish.java.net/public/comparing_v2_and_v3.html`

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 uživatelích 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 odstraní příslušného persistentního unitu (PersistentUnit) a překonfigurování EJB tak, aby namísto něj používaly persistetní 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

4.2.3 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 na synchronizovat persistentní a datovou vrstvu zavoláním metody flush na manažeru entit (EntitManager).

4.2.4 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 kontant. Díky tomu bude v budoucnu jednoduší 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 a přes ty se následně iterovalo a hledaly se ty, které měly příznak „finished“ na hodnotě „false“.

4.2.4.1 Zabezpečení

Původní projekt využíval pro autentizaci uživatelů file realm Glassfishe. 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 nepřípustné. Nově přihlašovací formulář využívá metodu POST.

4.2.4.2 JSF 2.0

Jak již bylo zmíněno v první iteraci 4.1, nová verze Glassfishe 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 není neincludují hlavičku a patičku, jak tomu bylo původně, nýbrž využívají šablonovacího systému.

4.2.4.3 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, ke 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.

4.2.5 Výstup

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

4.3 Třetí iterace

4.3.1 Cíl

- přidání role kandidáta - aktér čas - Sheduler - desktopový klient

4.3.2 Postup

4.3.3 Výstup

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. Aplikace musí být srozumitelný nejen počítačově gramotným uživatelům, ale i uživatelům, kteří nemají zkušenosti ovládáním počítače.

5.1 Statická analýza kódu

5.2 Selenium

5.3 Zátěžové testy

5.3.1 Apache Benchmark

5.3.2 Doba SQL dotazů

Kapitola 6

Závěr

- Zhodnocení splnění cílů DP/BP a vlastního přínosu práce (při formulaci je třeba vzít v potaz zadání práce).
- Diskuse dalšího možného pokračování práce.

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

UML diagramy

Tato příloha není povinná a zřejmě se neobjeví v každé práci. Máte-li ale větší množství podobných diagramů popisujících systém, není nutné všechny umísťovat do hlavního textu, zvláště pokud by to snižovalo jeho čitelnost.

Příloha C

Instalační a uživatelská příručka

Tato příloha velmi žádoucí zejména u softwarových implementačních prací.

Příloha D

Obsah přiloženého CD